# Rapport : Laboratoire 7 – Account Management


## Auteurs

- Harun Ouweis
- Mouti Amir


---

## 1. Introduction

Ce rapport présente les étapes réalisées dans le cadre du Laboratoire 7 portant sur la gestion des comptes utilisateurs sous Linux. L'objectif est de se familiariser avec les outils `useradd`, `groupadd`, `usermod`, `userdel` et les mécanismes liés aux groupes, à `sudo`, et aux fichiers systèmes `/etc/passwd`, `/etc/group` et `/etc/sudoers`.

Chaque tâche est documentée avec les **commandes exécutées**, **les résultats obtenus**, et **les explications nécessaires** pour une relecture claire et complète.

---

## 2. Étape 0 – Analyse du compte utilisateur courant

### 2.1. Commandes exécutées

```
id
cat /etc/passwd | grep tobioo
cat /etc/group | grep tobioo
```

---

### 2.2. Sorties obtenues

**Commande `id`**

```
uid=1000(tobioo) gid=1000(tobioo)
groups=1000(tobioo),4(adm),20(dialout),24(cdrom),25(floppy),27(sudo),29(audio),30(
dip),44(video),46(plugdev),116(netdev),998(ollama),1001(docker),1002(proj_a),1003(
proj_b)
```

**Commande `cat /etc/passwd | grep tobioo`**

```
tobioo:x:1000:1000:,,,:/home/tobioo:/bin/bash
```

**Commande `cat /etc/group | grep tobioo`**

```
adm:x:4:syslog,tobioo
dialout:x:20:tobioo
cdrom:x:24:tobioo
floppy:x:25:tobioo
sudo:x:27:tobioo
audio:x:29:tobioo
dip:x:30:tobioo
video:x:44:tobioo,ollama
plugdev:x:46:tobioo
netdev:x:116:tobioo
tobioo:x:1000:
docker:x:1001:tobioo
ollama:x:998:tobioo
proj_a:x:1002:tobioo
proj_b:x:1003:tobioo
```

---

## 2.3. Explications

- L'utilisateur `tobioo` possède l'UID **1000** et le GID principal **1000**, correspondant au groupe `tobioo`.
- Son **shell de connexion est** `/bin/bash` et son **répertoire personnel est** `/home/tobioo`, comme visible dans `/etc/passwd`.
- Il est membre de plusieurs **groupes secondaires**, dont :
  - `sudo` : lui donnant temporairement les droits administrateur via la commande `sudo`.
  - `docker`, `ollama`, `proj_a`, `proj_b` : probablement liés à des projets ou des outils spécifiques.
  - `adm`, `audio`, `video`, etc. : groupes système standard permettant l'accès à certains périphériques ou fichiers log.
- Ces informations proviennent de deux fichiers critiques :
  - `/etc/passwd` : décrit chaque utilisateur (UID, GID, home, shell).
  - `/etc/group` : associe des utilisateurs à des groupes (GID, membres).
- La commande `id` résume en une ligne l'ensemble des informations utiles sur les groupes, UID et GID liés à l'utilisateur courant.

## 2.4. Fichiers skeleton copiés

### Commandes exécutées

```
ls -la /home/tobioo
ls -la /etc/skel
```

### Sortie typique obtenue

```
# /etc/skel
.bash_logout
.bashrc
.profile
```

```
# /home/tobioo
.bash_logout
.bashrc
.profile
...
```

**Explication**

Lors de la création d'un utilisateur avec l'option `-m`, le contenu du répertoire `/etc/skel` est copié dans le répertoire personnel. On retrouve ainsi les fichiers `.bashrc`, `.profile`, `.bash_logout`, qui servent à initialiser l'environnement du shell bash à la première connexion.

---

## 2.5. Réponses aux questions

**Q1. Quel est le groupe principal du compte utilisateur courant ?**

**Réponse :** Le groupe principal du compte `tobioo` est le groupe `tobioo`, ayant le GID 1000. Cela est confirmé par la commande :

```
id
```

Sortie :

```
uid=1000(tobioo) gid=1000(tobioo) ...
```

---

**Q2. À quels autres groupes le compte appartient-il ?**

**Réponse :** Le compte `tobioo` est également membre des groupes suivants :

- `adm`, `dialout`, `cdrom`, `floppy`, `sudo`, `audio`, `dip`, `video`, `plugdev`, `netdev`, `ollama`, `docker`, `proj_a`, `proj_b`

Ces groupes peuvent être affichés avec :

```
id
```

---

**Q3. Quel est l'UID du compte et le GID de son groupe principal ?**

**Réponse :**

- UID : `1000`
- GID (groupe principal) : `1000` Cela est affiché par la commande :

```
id
```

---

**Q4. Quels fichiers skeleton ont été copiés dans le répertoire personnel ?**

**Réponse :** Les fichiers suivants ont été copiés depuis `/etc/skel` vers `/home/tobioo` lors de la création du compte :

- `.bashrc`
- `.bash_logout`
- `.profile`

Vérifié avec :

```
ls -la /etc/skel
ls -la /home/tobioo
```

---

# 3. Étape 1 – Création de comptes et groupes

---

## 3.1. Création des groupes `jedi` et `rebels`

**Commandes exécutées**

```
# Vérifier si les groupes existent déjà
getent group jedi
getent group rebels

# Créer les groupes s'ils n'existent pas
sudo groupadd jedi
sudo groupadd rebels
```

**Explication**

- `getent group <nom>` interroge la base des groupes pour éviter un doublon.

- `groupadd` crée un groupe système.
- Ces groupes sont utilisés comme **groupes principaux** (`-g`) ou **secondaires** (`-G`) pour les utilisateurs.

---

## 3.2. Création des utilisateurs `luke`, `vader` et `solo`

### Commandes exécutées

```
# Vérifier que les utilisateurs n'existent pas
getent passwd luke
getent passwd vader
getent passwd solo

# Si aucune ligne n'est retournée, les utilisateurs peuvent être créés

# Créer l'utilisateur luke, groupe principal jedi, groupe secondaire rebels
sudo useradd -m -s /bin/bash -g jedi -G rebels luke

# Créer l'utilisateur vader, groupe principal jedi
sudo useradd -m -s /bin/bash -g jedi vader

# Créer l'utilisateur solo, groupe principal rebels
sudo useradd -m -s /bin/bash -g rebels solo
```

### Explication

- `-m` crée automatiquement le répertoire personnel `/home/<utilisateur>`.
- `-s` définit le shell à utiliser (ici `/bin/bash`).
- `-g` précise le groupe principal, `-G` les groupes secondaires.
- Par défaut, sans `-s`, le shell est `/bin/sh`.

---

## 3.3. Définition du mot de passe pour `luke`

### Commande exécutée

```
sudo passwd luke
```

→ Mot de passe défini manuellement (exemple : `1234`).

### Explication

- Un utilisateur sans mot de passe ne peut pas se connecter via `su -`.
- On peut aussi définir les mots de passe de `vader` et `solo` si nécessaire pour les tests.

---

## 3.4. Tests d’accès et de permissions

#### Commandes exécutées

```
# Se connecter en tant que luke
su - luke

# Créer un fichier dans son home
touch test.txt
ls -l test.txt
ls -ld .

# Vérifier l'accès au fichier sensible
cat /etc/shadow
```

#### Sorties obtenues

```
-rw-r--r-- 1 luke jedi 0 May 16 00:46 test.txt
drwxr-x--- 2 luke jedi 4096 May 16 00:47 .
cat: /etc/shadow: Permission denied
```

#### Explication

- Le fichier `test.txt` est créé avec succès, montrant que l'environnement utilisateur fonctionne.
- Les permissions du répertoire personnel de `luke` sont `drwxr-x---`, ce qui permet théoriquement l'accès aux membres du groupe principal (`jedi`).
- Ce comportement est spécifique à **WSL**, car dans une distribution Linux native, on aurait généralement `drwx------`.

---

### 3.5. Accès au home de `luke` via `vader`

#### Commandes exécutées

```
sudo passwd vader
su - vader
ls /home/luke
```

#### Sortie obtenue

```
test.txt
```

#### Explication

- Dans une distribution Linux classique, cela devrait échouer (`Permission denied`) à cause des permissions strictes sur `/home/luke`.
- Sous **WSL**, les permissions par défaut sont plus permissives, ce qui explique pourquoi `vader` peut lire le contenu du répertoire.

---

## 3.6. Réponses aux questions

**Q1. Quelle option faut-il spécifier pour que `useradd` crée un répertoire personnel (home) ?**

**Réponse :** L'option `-m`

```
sudo useradd -m <nom>
```

---

**Q2. Quel est le shell par défaut pour les utilisateurs créés avec `useradd` ?**

**Réponse :** Le shell par défaut est `/bin/sh`.

---

**Q3. Quelle commande devons-nous utiliser pour changer le shell par défaut de `/bin/sh` à `/bin/bash` ?**

**Réponse :** À la création du compte :

```
sudo useradd -s /bin/bash <nom>
```

Après la création :

```
sudo chsh -s /bin/bash <nom>
```

---

## 3.7. Vérification

**Commandes exécutées**

```
# Vérifier les utilisateurs créés
getent passwd luke
getent passwd vader
getent passwd solo

# Vérifier les groupes et l'appartenance
id luke
id vader
id solo
```

#### Sorties obtenues

```
luke:x:1001:1004::/home/luke:/bin/bash
vader:x:1002:1004::/home/vader:/bin/bash
solo:x:1003:1005::/home/solo:/bin/bash

uid=1001(luke) gid=1004(jedi) groups=1004(jedi),1005(rebels)
uid=1002(vader) gid=1004(jedi) groups=1004(jedi)
uid=1003(solo) gid=1005(rebels) groups=1005(rebels)
```

#### Explication

- Tous les utilisateurs sont créés avec un UID distinct, un répertoire personnel, et le shell `/bin/bash`.
- Les affectations de groupes sont correctes et conformes à l'énoncé.
-

---

# 4. Étape 2 – Modification de l'appartenance aux groupes

## 4.1. Création de `leia` sans groupe spécifié

#### Commande exécutée

```
sudo useradd -m leia
id leia
```

##### Sortie obtenue

```
uid=1004(leia) gid=1006(leia) groups=1006(leia)
```

#### Explication

En l'absence d'option `-g`, un groupe du **même nom que l'utilisateur** est automatiquement créé et assigné comme groupe principal (comportement standard sur les systèmes Linux avec configuration `USERGROUPS_ENAB yes` dans `/etc/login.defs`).

---

## 4.2. Ajout de `leia` au groupe `rebels` (groupe secondaire)

#### Commande exécutée

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

**Sortie obtenue**

```
uid=1004(leia) gid=1006(leia) groups=1006(leia),1005(rebels)
```

**Explication**

- `-aG` signifie : ajouter (`-a`) l'utilisateur à un ou plusieurs groupes secondaires (`-G`) **sans retirer ceux déjà existants**.
- Sans `-a`, l'utilisateur perdrait tous ses groupes secondaires précédents.

---

## 4.3. Retrait du groupe `rebels` et ajout au groupe `jedi`

**Commandes exécutées**

```
# Retirer leia du groupe rebels
sudo gpasswd -d leia rebels

# Ajouter leia au groupe jedi
sudo usermod -aG jedi leia

id leia
```

**Sortie obtenue**

```
Removing user leia from group rebels
uid=1004(leia) gid=1006(leia) groups=1006(leia),1004(jedi)
```

**Explication**

- `gpasswd -d` permet de retirer proprement un utilisateur d'un groupe existant.
- On utilise à nouveau `usermod -aG` pour ajouter un groupe secondaire sans perte d'autres appartenances.

---

## 4.4. Retrait de tous les groupes secondaires

**Commande exécutée**

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

**Sortie obtenue**

```
uid=1004(leia) gid=1006(leia) groups=1006(leia)
```

**Explication**

- En fournissant une chaîne vide à `-G`, tous les groupes secondaires de l'utilisateur sont retirés.
- Le groupe principal reste inchangé (dans ce cas : `leia`).

---

## 4.5. Réponses aux questions

**Q1. Quel groupe principal a été attribué automatiquement à `leia` ?**

**Réponse :** Un groupe du même nom (`leia`) a été automatiquement créé et défini comme groupe principal. Cela est visible avec :

```
id leia
```

sortie obtenue:

```
uid=1004(leia) gid=1006(leia) groups=1006(leia)
```

---

**Q2. Comment rendre `leia` membre du groupe `rebels` (groupe secondaire) ?**

**Réponse :** Utiliser la commande :

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

---

**Q3. Comment retirer `leia` du groupe `rebels` et l'ajouter au groupe `jedi` ?**

**Réponse :**

```
sudo gpasswd -d leia rebels
sudo usermod -aG jedi leia
```

**Q4. Comment retirer `leia` de tous ses groupes secondaires ?**

**Réponse :**

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

---

# 5. Étape 3 – Droits `sudo` pour un utilisateur

---

## 5.1. Ligne dans `/etc/sudoers` donnant les droits sudo au groupe `sudo`

**Commande exécutée**

```
sudo cat /etc/sudoers | grep -E '^%sudo'
```

**Sortie obtenue :**

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

**Explication**

- Cette ligne autorise tous les membres du groupe `sudo` à exécuter **n'importe quelle commande** en tant que **n'importe quel utilisateur** (ALL) et groupe (ALL), avec élévation via `sudo`.

---

## 5.2. Variante sans mot de passe (non recommandée)

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

**Explication**

- Ajouter `NOPASSWD:` signifie que les membres du groupe `sudo` **n'auront pas à saisir leur mot de passe** pour utiliser `sudo`.
- **Ce n'est pas recommandé en environnement réel** car cela réduit la sécurité du système.

---

## 5.3. Donner à `luke` les droits sudo

**Commande exécutée**

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

**Vérification**

```
id luke
```

**Sortie obtenue :**

```
uid=1001(luke) gid=1004(jedi) groups=1004(jedi),27(sudo),1005(rebels)
```

Confirme que `luke` fait désormais partie du groupe `sudo`.

---

## 5.4. Test des droits sudo

**Commandes exécutées**

```
su - luke
sudo cat /etc/shadow
```

**Sortie obtenue :**

```
# Un prompt demandant le mot de passe
[sudo] password for luke:

# Puis le contenu du fichier si le mot de passe est correct
root:*:xxxxx:x:xxxxx:x:::
...
```

**Explication**

- Le fait que `luke` puisse lire `/etc/shadow` prouve que `sudo` fonctionne.
- Sans `sudo`, ce fichier est inaccessible à tout utilisateur non-root.

---

## 5.5. Retrait des droits sudo de `luke`

**Commande exécutée**

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

**Sortie obtenue**

```
Removing user luke from group sudo
```

**Vérification finale**

```
id luke
```

Doit **ne plus inclure** `sudo` dans les groupes :

```
uid=1001(luke) gid=1004(jedi) groups=1004(jedi),1005(rebels)
```

---

### 5.6. Réponses aux questions

---

**Q1. Quelle ligne dans `/etc/sudoers` donne aux membres du groupe `sudo` le droit d'exécuter n'importe quelle commande ?**

**Réponse :**

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

---

**Q2. Comment modifier cette ligne pour autoriser `sudo` sans mot de passe ?**

**Réponse :**

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

Cette configuration n'est **pas recommandée** en environnement de production, mais peut être utile à des fins de test ou de scripts automatisés.

---

## 6. Étape 4 – Suppression de compte utilisateur

---

## 6.1. Suppression du compte `leia` (sans effacer le répertoire home)

**Commande exécutée**

```
sudo userdel leia
```

**Explication**

- La commande `userdel` supprime l'entrée de l'utilisateur dans `/etc/passwd`, mais **ne supprime pas son répertoire personnel**.
- Cela permet de préserver les fichiers de l'utilisateur en cas de besoin.

---

## 6.2. Observation du répertoire home après suppression

**Commande exécutée**

```
ls -l /home
```

**Sortie obtenue**

```
drwxr-x---  2   1004   1006 4096 May 16 01:08 leia
```

**Explication**

- Le répertoire `/home/leia` est toujours présent.
- Il appartient encore à l'utilisateur supprimé (l'UID de `leia` est conservé sur les fichiers), même si ce compte n'existe plus.

---

## 6.3. Recherche des fichiers appartenant à `leia` dans tout le système

**Commande exécutée**

```
sudo find / -user leia 2>/dev/null
```

**Sortie obtenue**

```

```

**Explication**

- Cette commande permet de repérer tous les fichiers appartenant à l'ancien utilisateur `leia`.
- Elle est utile pour **nettoyer manuellement** les fichiers oubliés ou résiduels.
- Ici, aucun fichier n'a été trouvé, ce qui est normal car nous n'avons pas créé de fichiers supplémentaires sous `leia` après sa création.

---

## 6.4. Suppression manuelle du répertoire personnel

**Commande exécutée**

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

**Explication**

- Cela supprime tous les fichiers restants appartenant à l'ancien utilisateur.
- Alternative : on aurait pu faire cette suppression automatiquement avec :

```
sudo userdel -r leia
```

Mais dans cette étape, on voulait **observer la persistance du home**, donc la suppression est volontairement faite en deux temps.

---

## 6.5. Vérification finale

**Commandes exécutées**

```
getent passwd leia
ls -l /home
```

**Sortie obtenue**

- `getent passwd leia` → **ne retourne rien**, preuve que l'utilisateur est supprimé.
- `ls -l /home` → **le répertoire `/home/leia` a été supprimé**.

---

## 6.6. Réponses aux questions du laboratoire

---

**Q1. Inspecter le répertoire personnel (`/home/leia`). Que constate-t-on ?**

**Réponse :** En inspectant le répertoire avec :

```
ls -l /home
```

On obtient :

```
drwxr-x---  2   1004   1006 4096 May 16 01:08 leia
```

Cela indique que :

- Le répertoire existe toujours.
- Le **nom de l'utilisateur `leia` a disparu**, remplacé par son **UID `1004`**.
- Le groupe principal reste également référencé par son GID (`1006`).
- Cela confirme que le compte est supprimé, mais les fichiers **conservent leurs UID/GID d'origine**.

---

**Q2. Supposons que `leia` ait créé d'autres fichiers ailleurs sur le système, mais que nous ne sachions pas où. Comment les retrouver systématiquement ?**

**Réponse :** Il faut utiliser la commande :

```
sudo find / -user leia 2>/dev/null
```

Cela permet de :

- Parcourir **tout le système de fichiers**,
- Filtrer selon les **fichiers appartenant à l'utilisateur supprimé** (`UID`),
- En ignorant les erreurs d'accès (`2>/dev/null`).

Si le compte est supprimé, on peut aussi utiliser son UID directement :

```
sudo find / -uid 1004 2>/dev/null
```

---