TP13 : Les utilisateurs et les droits

Sommaire

1 – La gestion des utilisateurs

→ On vérifie si les utilisateurs **daemon** et **luke** existent en regardant leur uid, gid et groupes si c'est le cas avec **id** :

```
root@DEB12Server: ~#id daemon
uid=1(daemon) gid=1(daemon) groupes=1(daemon)
root@DEB12Server: ~#id luke
id: « luke » : utilisateur inexistant
```

→ On créé les groupes **jedi** et **rebelles** avec **useradd** :

```
root@DEB12Server: ~#groupadd jedi
root@DEB12Server: ~#groupadd rebelles
```

→ On consulte le manuel d'utilisation de la commande **useradd** grâce à la commande **man** :

```
root@DEB12Server: ~#man useradd_
```

```
USERADD(8)              Commandes de gestion du système              USERADD(8)

NOM
       useradd - créer un nouvel utilisateur ou modifier les informations par
       défaut appliquées aux nouveaux utilisateurs

SYNOPSIS
       useradd [options] LOGIN

       useradd -D

       useradd -D [options]

DESCRIPTION
       useradd is a low level utility for adding users. On Debian,
       administrators should usually use adduser(8) instead.

       When invoked without the -D option, the useradd command creates a new
       user account using the values specified on the command line plus the
       default values from the system. Depending on command line options, the
       useradd command will update system files and may also create the new
       user's home directory and copy initial files.

       By default, a group will also be created for the new user (see -g, -N,
       -U, and USERGROUPS_ENAB).
```

3

→ On créé les utilisateurs **luke**, **vador** et **solo** en les ajoutant dans des goupes pricipaux ou secondaires avec **-g** et **-G** puis on les visualise :

```
root@DEB12Server: ~#useradd -g jedi -G rebelles -m luke
root@DEB12Server: ~#useradd -g jedi -m vador
root@DEB12Server: ~#useradd -g rebelles -m solo
root@DEB12Server: ~#id luke
uid=1002(luke) gid=1002(jedi) groupes=1002(jedi),1003(rebelles)
root@DEB12Server: ~#id vador
uid=1003(vador) gid=1002(jedi) groupes=1002(jedi)
root@DEB12Server: ~#id solo
uid=1004(solo) gid=1003(rebelles) groupes=1003(rebelles)
```

→ On affiche les dernières lignes des fichiers **/etc/passwd** et **/etc/group** :

```
root@DEB12Server: ~#tail -3 /etc/passwd
luke:x:1002:1002::/home/luke:/bin/sh
vador:x:1003:1002::/home/vador:/bin/sh
solo:x:1004:1003::/home/solo:/bin/sh
root@DEB12Server: ~#tail -2 /etc/group
jedi:x:1002:
rebelles:x:1003:luke
```

*** Les comptes utilisateurs et les groupes qu'on vient de créer apparaissent.**

→ On modifie le **mot de passe** du compte **luke** avec la commande **passwd** :

```
root@DEB12Server: ~#passwd luke
Nouveau mot de passe :
Retapez le nouveau mot de passe :
passwd : mot de passe mis à jour avec succès
```

4

→ On ouvre une seconde console avec le raccourci **CTRL + ALT + F2** et on se connecte au compte **luke** :

```
Debian GNU/Linux 12 DEB12Server tty2

DEB12Server login: luke
Password:
Linux DEB12Server 6.1.0-26-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.112-1 (2024-09-30) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
$ _
```

*** On observe uniquement le signe $ dans le prompt.**

→ On se déconnecte du compte luke et on retourne sur la première console avec **CTL + ALT + F1** pour remplacer le **shell sh** du compte **luke** par **bash** :

```
root@DEB12Server: ~#usermod -s /bin/bash luke
```

→ On se reconnecte sur le compte **luke** et on observe que le **prompt** a été modifié. On utilise ensuite la commande **id** sans arguments pour l'utiliser directement sur le compte actuel :

```
luke@DEB12Server:~$ id
uid=1002(luke) gid=1002(jedi) groupes=1002(jedi),1003(rebelles)
```

→ On ajoute le compte **leia** en tant que **root** et on le visualise ;

```
root@DEB12Server: ~#useradd leia
root@DEB12Server: ~#id leia
uid=1005(leia) gid=1005(leia) groupes=1005(leia)
```

*** On observe que par défaut, un nouveau compte créé est mis dans un nouveau groupe du même nom.**

5

→ On observe les répertoires personnels :

```
root@DEB12Server: ~#ls -l /home
total 20
drwx------ 5 guest guest    4096 21 déc.  15:50 guest
drwxr-xr-x 2 luke  jedi     4096 22 déc.  14:50 luke
drwx------ 2 noah  noah     4096  6 oct.  20:31 noah
drwxr-xr-x 2 solo  rebelles 4096 22 déc.  14:50 solo
drwxr-xr-x 2 vador jedi     4096 22 déc.  14:50 vador
```

*** Le répertoire personnel du compte leia n'a pas été créé.**

→ On affecte l'utilisateur **leia** au groupe **rebelles** :

```
root@DEB12Server: ~#usermod -G rebelles leia
root@DEB12Server: ~#id leia
uid=1005(leia) gid=1005(leia) groupes=1005(leia),1003(rebelles)
```

→ On l'affecte maintenant au groupe **jedi** :

```
root@DEB12Server: ~#usermod -G jedi leia
root@DEB12Server: ~#id leia
uid=1005(leia) gid=1005(leia) groupes=1005(leia),1002(jedi)
```

*** L'utilisateur quitte automatiquement son ancien groupe secondaire quand on l'affecte à un autre.**

→ On l'affecte aux groupes **jedi** et **rebelles :**

```
root@DEB12Server: ~#usermod -G jedi,rebelles leia
root@DEB12Server: ~#id leia
uid=1005(leia) gid=1005(leia) groupes=1005(leia),1002(jedi),1003(rebelles)
```

*** L'utilisateur peut maintenant être dans les groupes puisqu'on l'a affecté aux deux groupes dans la même commande.**

→ On **retire** tous les groupes secondaires de l'utilisateur **leia** :

```
root@DEB12Server: ~#usermod -G "" leia
root@DEB12Server: ~#id leia
uid=1005(leia) gid=1005(leia) groupes=1005(leia)
```

→ On utilise l'option **-a** pour affecter un utilisateur qui a déjà un groupe secondaire à un **autre groupe secondaire** sans lui faire quitter l'ancien :

```
root@DEB12Server: ~#usermod -G jedi leia
root@DEB12Server: ~#usermod -aG rebelles leia
root@DEB12Server: ~#id leia
uid=1005(leia) gid=1005(leia) groupes=1005(leia),1002(jedi),1003(rebelles)
```

→ On **supprime** le compte **leia** :

```
root@DEB12Server: ~#userdel leia
root@DEB12Server: ~#id leia
id: « leia » : utilisateur inexistant
```

→ On recréé le compte **leia** en lui ajoutant cette fois-ci un **répertoire de connexion** grâce à **-m** :

```
root@DEB12Server: ~#useradd -m leia
```

→ On se déplace dans le **répertoire personnel** de **leia** puis on créé un **répertoire** ainsi qu'un **fichier** à partir du compte utilisateur :

```
root@DEB12Server: ~#cd /home/leia
root@DEB12Server: /home/leia#su - leia
$ mkdir rep1
$ cd rep1
$ touch fichier1
$ ls -l
total 0
-rw-r--r-- 1 leia leia 0 22 déc.  15:39 fichier1
$ exit
```

→ On **supprime** le compte utilisateur en utilisant cette fois le paramètre **-r** qui permet de supprimer tout les fichiers de son répertoire :

```
root@DEB12Server: ~#userdel -r leia
userdel : leia spool de courrier /var/mail/leia non trouvé
root@DEB12Server: ~#ls -l /home/leia
ls: impossible d'accéder à '/home/leia': Aucun fichier ou dossier de ce type
root@DEB12Server: ~#id leia
id: « leia » : utilisateur inexistant
```

7

→ On recréé le compte **leia** à l'identique en spécifiant son **uid** et son **gid** puis on lui définit un mot de passe :

```
root@DEB12Server: ~#groupadd -g 1007 leia
root@DEB12Server: ~#useradd -u 1007 -g leia -m -s /bin/bash leia
root@DEB12Server: ~#id leia
uid=1007(leia) gid=1007(leia) groupes=1007(leia)
root@DEB12Server: ~#passwd leia
Nouveau mot de passe :
Retapez le nouveau mot de passe :
passwd : mot de passe mis à jour avec succès
```

→ On créé le compte **toor** en lui attribuant les mêmes droits que **root** :

```
root@DEB12Server: ~#useradd -u 0 -o -d /root -s /bin/bash toor
useradd warning: toor's uid 0 outside of the UID_MIN 1000 and UID_MAX 60000 rang
e.
root@DEB12Server: ~#id toor
uid=0(root) gid=1008(toor) groupes=0(root)
```

→ On se connecte au compte **toor** sur une nouvelle console :

```
DEB12Server login: toor
Password:
Linux DEB12Server 6.1.0-26-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.112-1 (2024-09-30) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Dec 22 14:35:57 CET 2024 on tty1
root@DEB12Server: ~#
```

→ On ajoute un nouvel utilisateur **palpatine** sans aucuns arguments pour utiliser la méthode de création de compte par défaut de Debian :

```
root@DEB12Server: ~#adduser palpatine
Ajout de l'utilisateur « palpatine » ...
Ajout du nouveau groupe « palpatine » (1005) ...
Ajout du nouvel utilisateur « palpatine » (1005) avec le groupe « palpatine » (1005) ...
Création du répertoire personnel « /home/palpatine » ...
Copie des fichiers depuis « /etc/skel » ...
Nouveau mot de passe :
Retapez le nouveau mot de passe :
passwd : mot de passe mis à jour avec succès
Modifier les informations associées à un utilisateur pour palpatine
Entrer la nouvelle valeur, ou appuyer sur ENTER pour la valeur par défaut
        NOM []:
        Numéro de chambre []:
        Téléphone professionnel []:
        Téléphone personnel []:
        Autre []:
Cette information est-elle correcte ? [O/n]o
Ajout du nouvel utilisateur « palpatine » aux groupes supplémentaires « users » ...
Ajout de l'utilisateur « palpatine » au groupe « users » ...
```

→ On affiche les caractéristiques de l'utilisateur local **luke** et du groupe local **rebelles** :

```
root@DEB12Server: ~#grep luke /etc/passwd
luke:x:1002:1002::/home/luke:/bin/bash
root@DEB12Server: ~#grep rebelles /etc/group
rebelles:x:1003:luke
```

*** On voit que l'utilisateur luke appartient au groupe secondaire rebelles**

2 - La gestion des droits

→ On créé une arborescence de fichiers :

```
root@DEB12Server: ~#mkdir /home/etoilenoire
root@DEB12Server: ~#cd /home/etoilenoire
root@DEB12Server: /home/etoilenoire#echo "voici les plans" > plans
root@DEB12Server: /home/etoilenoire#echo "c'est ouvert" > entree_secrete
```

→ On modifie les **caractéristiques** du répertoire **etoilenoire** :

```
root@DEB12Server: /home/etoilenoire#ls -ld /home/etoilenoire
drwxr-xr-x 2 root toor 4096 22 déc.  16:01 /home/etoilenoire
```

- On utilise **chown** pour changer le **propriétaire** du répertoire :

```
root@DEB12Server: /home/etoilenoire#chown luke /home/etoilenoire
```

- On utilise **chgrp** pour changer le **groupe** du répertoire :

```
root@DEB12Server: /home/etoilenoire#chgrp jedi /home/etoilenoire
```

- On utilise **chmod** pour changer les **droits** des utilisateurs sur le répertoire :

```
root@DEB12Server: /home/etoilenoire#chmod 750 /home/etoilenoire
```

```
root@DEB12Server: /home/etoilenoire#ls -ld /home/etoilenoire
drwxr-x--- 2 luke jedi 4096 22 déc.  16:01 /home/etoilenoire
```

*** Le propriétaire du répertoire est maintenant luke et son groupe est jedi.**

Le propriétaire a gardé tout les droits sur le répertoire, le groupe a gardé les droits de lecture et d'accès mais le droit d'accès a été retiré aux autres.

→ On modifie les **caractéristiques** des **fichiers** :

- On donne le droit de **lecture** au groupe et on retire tout les droits aux **autres** :

```
root@DEB12Server: /home/etoilenoire#chmod g=r,o=- /home/etoilenoire/*
```

- On **affilie** le fichier **plans** au groupe **jedi** et le fichier **entree_secrete** au groupe **rebelles** :

```
root@DEB12Server: /home/etoilenoire#chgrp jedi /home/etoilenoire/plans
root@DEB12Server: /home/etoilenoire#chgrp rebelles /home/etoilenoire/entree_secrete
```

```
root@DEB12Server: /home/etoilenoire#ls -l /home/etoilenoire/
total 8
-rw-r----- 1 root rebelles 13 22 déc.  16:01 entree_secrete
-rw-r----- 1 root jedi     16 22 déc.  16:01 plans
```

→ On se connecte au compte **luke** et on vérifie ses **permissions :**

```
root@DEB12Server: ~#su - luke
luke@DEB12Server:~$ ls /home/etoilenoire/
entree_secrete  plans
luke@DEB12Server:~$ cat /home/etoilenoire/entree_secrete
c'est ouvert
luke@DEB12Server:~$ cal > /home/etoilenoire/fichier
luke@DEB12Server:~$ ls /home/etoilenoire/
entree_secrete  fichier  plans
luke@DEB12Server:~$ rm /home/etoilenoire/fichier
luke@DEB12Server:~$ ls /home/etoilenoire/
entree_secrete  plans
luke@DEB12Server:~$ echo "===" >> /home/etoilenoire/plans
-bash: /home/etoilenoire/plans: Permission non accordée
luke@DEB12Server:~$ exit_
```

*** L'utilisateur a tout les droits sur le répertoire etoilenoire car il est propriétaire, il peut lire les fichiers plans et entree_secrete car il est membre de groupe jedi et rebelles mais il ne peut pas les modifier car seul root y est autorisé.**

→ On teste les **permissions** du compte **vador** :

```
root@DEB12Server: ~#su - vador
$ ls /home/etoilenoire
entree_secrete  plans
$ rm /home/etoilenoire/plans
rm : supprimer '/home/etoilenoire/plans' qui est protégé en écriture et est du t
ype « fichier » ? y
rm: impossible de supprimer '/home/etoilenoire/plans': Permission non accordée
$ cal > /home/etoilenoire/fichier
-sh: 3: cannot create /home/etoilenoire/fichier: Permission denied
$ cat /home/etoilenoire/plans
voici les plans
$ cat /home/etoilenoire/entree_secrete
cat: /home/etoilenoire/entree_secrete: Permission non accordée
$ echo "====" >> /home/etoilenoire/plans
-sh: 6: cannot create /home/etoilenoire/plans: Permission denied
$ exit
```

*** Il peut lister le répertoire etoilenoire car il est membre du groupe jedi et a accès aux fichiers qu'il contient mais il n'a pas d'autres droits dessus. Il peut aussi lire le fichier plans car il est dans le groupe jedi.**

→ On teste aussi les **permissions** du compte **solo** :

```
root@DEB12Server: ~#su - solo
$ ms /home/etoilenoire
-sh: 1: ms: not found
$ cal > /home/etoilenoire/fichier
-sh: 2: cannot create /home/etoilenoire/fichier: Permission denied
$ rm -f /home/etoilenoire/entree_secrete
rm: impossible de supprimer '/home/etoilenoire/entree_secrete': Permission non a
ccordée
$ cat /home/etoilenoire/entree_secrete
cat: /home/etoilenoire/entree_secrete: Permission non accordée
$ exit
```

*** Il n'a aucun droits sur le répertoire etoilenoire car il fait partie de other.**

→ On **supprime** le **droit d'éxuction** de la commande **uptime** à **other** de façon temporaire et on teste les conséquences depuis le compte **luke** :

```
root@DEB12Server: ~#whereis uptime
uptime: /usr/bin/uptime /usr/share/man/man1/uptime.1.gz
root@DEB12Server: ~#whatis uptime
uptime (1)           - Indiquer depuis quand le système a été mis en route
root@DEB12Server: ~#uptime
 17:08:04 up  2:40,  3 users,  load average: 0,00, 0,00, 0,00
root@DEB12Server: ~#ls -l /usr/bin/uptime
-rwxr-xr-x 1 root root 14648 19 déc.   2022 /usr/bin/uptime
root@DEB12Server: ~#chmod o-x /usr/bin/uptime
root@DEB12Server: ~#su - luke
luke@DEB12Server:~$ uptime
-bash: /usr/bin/uptime: Permission non accordée
luke@DEB12Server:~$ exit_
root@DEB12Server: ~#chmod o+x /usr/bin/uptime
root@DEB12Server: ~#ls -l /usr/bin/uptime
-rwxr-xr-x 1 root root 14648 19 déc.   2022 /usr/bin/uptime
root@DEB12Server: ~#su - luke
luke@DEB12Server:~$ uptime
 17:09:49 up  2:42,  3 users,  load average: 0,00, 0,00, 0,00
luke@DEB12Server:~$ exit
```

3 - La gestion des droits, compléments

→ On ajoute des droits **SGID** et **sticky-bit** au répertoire **etoilenoire** puis on créé des fichiers dans celui-ci pour vérifier l'impact des droits :

```
root@DEB12Server: ~#chmod 3770 /home/etoilenoire/
root@DEB12Server: ~#ls -ld /home/etoilenoire
drwxrws--T 2 luke jedi 4096 22 déc.  17:13 /home/etoilenoire
root@DEB12Server: ~#echo "fichier un" > /home/etoilenoire/f1
root@DEB12Server: ~#su - luke
luke@DEB12Server:~$ echo "bonjour" > /home/etoilenoire/f2
luke@DEB12Server:~$ exit
```

```
root@DEB12Server: ~#su - vador
$ echo "bonjour" > /home/etoilenoire/f3
$ exit
root@DEB12Server: ~#ls -l /home/etoilenoire/f?
-rw-r--r-- 1 root  jedi 11 22 déc.  17:13 /home/etoilenoire/f1
-rw-r--r-- 1 luke  jedi  8 22 déc.  17:13 /home/etoilenoire/f2
-rw-r--r-- 1 vador jedi  8 22 déc.  17:19 /home/etoilenoire/f3
```

*** L'ensemble des fichiers sont affiliés au groupe jedi à cause du droit SGID**

→ On essaye de supprimer le fichier de **luke** à partir du compte **vador** avec le droit **sticky-bit** :

```
root@DEB12Server: ~#su - vador
$ rm /home/etoilenoire/f2
rm : supprimer '/home/etoilenoire/f2' qui est protégé en écriture et est du type
 « fichier » ? y
rm: impossible de supprimer '/home/etoilenoire/f2': Opération non permise
$ exit
```

*** L'opération est refusée à cause de celui-ci**

→ On réessaye sans le droit **sticky-bit** :

```
root@DEB12Server: ~#chmod -t /home/etoilenoire/
root@DEB12Server: ~#ls -ld /home/etoilenoire/
drwxrws--- 2 luke jedi 4096 22 déc.  17:19 /home/etoilenoire/
root@DEB12Server: ~#su - vador
$ rm /home/etoilenoire/f2
rm : supprimer '/home/etoilenoire/f2' qui est protégé en écriture et est du type
 « fichier » ? y
$ ls -l /home/etoilenoire/f2
ls: impossible d'accéder à '/home/etoilenoire/f2': Aucun fichier ou dossier de c
e type
$ exit_
```

*** Sans le droit sticky-bit la permission est accordée.**

→ On regarde qui peut formater la partition **/dev/sda1** :

```
root@DEB12Server: ~#ls -l /dev/sda1
brw-rw---- 1 root disk 8, 1 22 déc.  14:27 /dev/sda1
```

*** Seul root et les membres du groupe disk peuvent le faire.**

→ On copie les fichiers du répertoire **etoilenoire** dans **/tmp** en conservant leurs attributs en tant que **root** :

```
root@DEB12Server: ~#cp -p /home/etoilenoire/* /tmp
root@DEB12Server: ~#ls -l /tmp/plans /tmp/entree_secrete
-rw-r----- 1 root rebelles 13 22 déc.  16:01 /tmp/entree_secrete
-rw-r----- 1 root jedi     16 22 déc.  16:01 /tmp/plans
```

*** Le paramètre -p permet que les fichiers copiés gardent leur groupes propriétaire.**

→ On donne le fichier **entree_secrete** au compte **luke** :

```
root@DEB12Server: ~#chown luke /tmp/entree_secrete
root@DEB12Server: ~#ls -l /tmp/entree_secrete
-rw-r----- 1 luke rebelles 13 22 déc.  16:01 /tmp/entree_secrete
```

→ On teste les accès au fichier **/tmp/entree_secrete** :

Sur le compte luke :

```
root@DEB12Server: ~#su - luke
luke@DEB12Server:~$ cat /tmp/entree_secrete
c'est ouvert
luke@DEB12Server:~$ echo "======" >> /tmp/entree_secrete
luke@DEB12Server:~$ cat /tmp/entree_secrete
c'est ouvert
======
luke@DEB12Server:~$ /tmp/entree_secrete
-bash: /tmp/entree_secrete: Permission non accordée
luke@DEB12Server:~$ exit
```

Sur le compte solo :

```
root@DEB12Server: ~#su - solo
$ cat /tmp/entree_secrete
c'est ouvert
======
$ echo "++++++" >> /tmp/entree_secrete
-sh: 2: cannot create /tmp/entree_secrete: Permission denied
$ exit
```

Sur le compte root :

```
root@DEB12Server: ~#cat /tmp/entree_secrete
c'est ouvert
======
root@DEB12Server: ~#echo "+=+=+=+=" >> /tmp/entree_secrete
-bash: /tmp/entree_secrete: Permission non accordée
root@DEB12Server: ~#cat /tmp/entree_secrete
c'est ouvert
======
root@DEB12Server: ~#/tmp/entree_secrete
-bash: /tmp/entree_secrete: Permission non accordée
```

*** Ils ont tous les droits d'affichage et d'écriture mais pas d'éxecution.**

→ On visualise les droits du fichier **shadow** ainsi que de la commande **passwd** :

```
root@DEB12Server: ~#ls -l /etc/shadow
-rw-r----- 1 root shadow 1322 22 déc.  15:55 /etc/shadow
root@DEB12Server: ~#ls -l /usr/bin/passwd
-rwsr-xr-x 1 root root 68248 23 mars   2023 /usr/bin/passwd
```

*** Tout le monde peut éxecuter la commande passwd.**

Les utilisateurs récupèrent les droits de root grâce au droit SUID et donc peuvent accéder au fichier shadow.