Version : **2024.01**

Dernière mise-à-jour : 2024/03/08 08:28

LDF501 - Gestion des Utilisateurs

Contenu du Module

Présentation

A faire : Afin de mettre en pratique les exemples dans ce cours, vous devez vous connecter à votre système en tant que root grâce à la commande **su -** et le mot de passe **fenestros**.

La bonne gestion des utilisateurs passe par une bonne stratégie de groupes. En effet, chaque utilisateur est affecté à un groupe **principal** mais il peut aussi être membre d'un ou de plusieurs groupes secondaires.

Comme dans d'autres systèmes d'exploitation, sous Linux il est préférable de donner les droits d'accès aux groupes et non aux utilisateurs individuels.

Les bases de données utilisées pour stocker les informations des utilisateurs et des groupes sont stipulées dans le fichier **/etc/nsswitch.conf**. Dans notre cas les entrées passwd, shadow et group indique le mot clef **files**. Ceci indique l'utilisation des fichiers suivants en tant que base de données :

- **/etc/passwd**,
- **/etc/shadow**,
- **/etc/group**.

/etc/nsswitch.conf

```
root@debian11:~# cat /etc/nsswitch.conf
# /etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch functionality.
# If you have the `glibc-doc-reference' and `info' packages installed, try:
```

```
# `info libc "Name Service Switch"' for information about this file.

passwd:         files systemd
group:          files systemd
shadow:         files
gshadow:        files

hosts:          files mdns4_minimal [NOTFOUND=return] dns
networks:       files

protocols:      db files
services:       db files
ethers:         db files
rpc:            db files

netgroup:       nis
```

Pour les entrées **passwd**, **group**, **shadow** et **gshadow** :

- **files** implique l'utilisation des fichiers locaux dans le répertoire **/etc**,
- **systemd** implique l'utilisation du plugin **nss-systemd** de la fonctionalité **Name Service Switch** (NSS) de la bibliothèque **GNU C Library** (glibc).

Il est aussi possible de trouver une autre valeur pour ces entrées :

- **sss** implique l'utilisation du **System Security Services Daemon** (SSSD).
  - SSSD trouve ses origines dans le projet opensource **FreeIPA** (Identity, Policy and Audit) et offre aux réseaux Linux/Unix des fonctionalités similaires à celles fournies aux réseaux Windows™ par les Microsoft Active Directory Domain Services,
  - Pour plus d'informations, consultez **cette page**.

Interrogation des Bases de Données

La commande **getent** est utilisée pour interroger les bases de données. Elle prend la forme suivante :

```
getent base-de-données clef
```

Par exemple pour rechercher l'utilisateur dans la base de données des utilisateurs, il convient d'utiliser la commande suivante :

```
root@debian11:~# getent passwd trainee
trainee:x:1000:1000:trainee,,,:/home/trainee:/bin/bash
```

Pour rechercher quels utilisateurs appartiennent à quels groupes, il convient d'utiliser la commande suivante :

```
root@debian11:~# getent group mail
mail:x:8:
```

L'utilisation de la commande getent sans spécifier une clef imprime à l'écran le contenu de la base de données :

```
root@debian11:~# getent passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
```

```
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-network:x:101:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:102:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:109::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:104:110:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
usbmux:x:105:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
rtkit:x:106:113:RealtimeKit,,,:/proc:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
avahi:x:108:114:Avahi mDNS daemon,,,:/run/avahi-daemon:/usr/sbin/nologin
speech-dispatcher:x:109:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/false
pulse:x:110:116:PulseAudio daemon,,,:/run/pulse:/usr/sbin/nologin
saned:x:111:119::/var/lib/saned:/usr/sbin/nologin
colord:x:112:120:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
lightdm:x:113:121:Light Display Manager:/var/lib/lightdm:/bin/false
trainee:x:1000:1000:trainee,,,:/home/trainee:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
sshd:x:114:65534::/run/sshd:/usr/sbin/nologin
```

Les Fichiers /etc/group et /etc/gshadow

Pour lister les groupes existants sur le système, saisissez la commande suivante :

```
root@debian11:~# cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
```

```
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:
dialout:x:20:
fax:x:21:
voice:x:22:
cdrom:x:24:trainee
floppy:x:25:trainee
tape:x:26:
sudo:x:27:
audio:x:29:pulse,trainee
dip:x:30:trainee
www-data:x:33:
backup:x:34:
operator:x:37:
list:x:38:
irc:x:39:
src:x:40:
gnats:x:41:
shadow:x:42:
utmp:x:43:
video:x:44:trainee
sasl:x:45:
plugdev:x:46:trainee
staff:x:50:
games:x:60:
users:x:100:
nogroup:x:65534:
systemd-journal:x:101:
systemd-network:x:102:
systemd-resolve:x:103:
input:x:104:
kvm:x:105:
```

```
render:x:106:
crontab:x:107:
netdev:x:108:trainee
messagebus:x:109:
systemd-timesync:x:110:
ssh:x:111:
ssl-cert:x:112:
rtkit:x:113:
avahi:x:114:
lpadmin:x:115:trainee
pulse:x:116:
pulse-access:x:117:
scanner:x:118:saned,trainee
saned:x:119:
colord:x:120:
lightdm:x:121:
trainee:x:1000:
systemd-coredump:x:999:
mlocate:x:122:
```

Important : Notez que la valeur du GID du groupe root est toujours de 0. Notez que sous Debian 11, les GID des utilisateurs normaux commencent à **1000** et les GID des comptes système sont inclus entre 1 et 99 et entre 201 et 999.

Dans ce fichier, chaque ligne est constituée de 4 champs :

- Le nom **unique** du groupe,
- Le mot de passe du groupe. Une valeur de **x** dans ce champs indique que le système utilise le fichier **/etc/gshadow** pour stocker les mots de passe. Une valeur de **!** indique que le groupe n'a pas de mot passe et que l'accès au groupe via la commande **newgrp** n'est pas possible,
- Le GID. Une valeur unique utilisée pour déterminée les droits d'accès aux fichiers et aux répertoires,
- La liste des membres ayant le groupe comme groupe **secondaire**.

Pour consulter le fichier **/etc/gshadow**, saisissez la commande suivante :

```
root@debian11:~# cat /etc/gshadow
root:*::
daemon:*::
bin:*::
sys:*::
adm:*::
tty:*::
disk:*::
lp:*::
mail:*::
news:*::
uucp:*::
man:*::
proxy:*::
kmem:*::
dialout:*::
fax:*::
voice:*::
cdrom:*::trainee
floppy:*::trainee
tape:*::
sudo:*::
audio:*::pulse,trainee
dip:*::trainee
www-data:*::
backup:*::
operator:*::
list:*::
irc:*::
src:*::
gnats:*::
shadow:*::
utmp:*::
video:*::trainee
```

```
sasl:*::
plugdev:*::trainee
staff:*::
games:*::
users:*::
nogroup:*::
systemd-journal:!::
systemd-network:!::
systemd-resolve:!::
input:!::
kvm:!::
render:!::
crontab:!::
netdev:!::trainee
messagebus:!::
systemd-timesync:!::
ssh:!::
ssl-cert:!::
rtkit:!::
avahi:!::
lpadmin:!::trainee
pulse:!::
pulse-access:!::
scanner:!::saned,trainee
saned:!::
colord:!::
lightdm:!::
trainee:!::
systemd-coredump:!*::
mlocate:!::
```

Chaque ligne est constituée de 4 champs :

- Le nom du groupe. Ce champs est utilisé pour faire le lien avec le fichier **/etc/group**,

- Le mot de passe **crypté** du groupe s'il en existe un. Une valeur **vide** dans ce champs indique que seuls les membres du groupe peuvent exécuter la commande **newgrp**. Une valeur de **!**, de **x** ou de * indique que personne ne peut exécuter la commande **newgrp** pour le groupe,
- L'administrateur du groupe s'il en existe un,
- La liste des membres ayant le groupe comme groupe **secondaire**.

Afin de vérifier les fichiers **/etc/group** et **/etc/gshadow** pour des erreurs éventuelles, saisissez la commande suivante :

```
root@debian11:~# grpck -r
root@debian11:~#
```

Dans le cas où vos fichiers ne comportent pas d'erreurs, vous vous retrouverez retourné au prompt.

Important : L'option **-r** permet la vérification des erreurs sans le modifier.

Dans le cas où il est nécessaire de régénérer un des deux fichiers, il convient d'utiliser une des deux commandes suivantes :

- **grpconv**
  - permet de régénérer le fichier **/etc/gshadow** à partir du fichier **/etc/group** et éventuellement du fichier **/etc/gshadow** existant
- **grpunconv**
  - permet de régénérer le fichier **/etc/group** à partir du fichier **/etc/gshadow** et éventuellement du fichier **/etc/group** existant puis supprime le fichier **/etc/gshadow**

Les Fichiers /etc/passwd et /etc/shadow

Important : Notez que la règle la plus libérale concernant les noms d'utilisateurs sous Linux limite la longueur à 32 caractères et permet l'utilisation de majuscules, de minuscules, de nombres (sauf au début du nom) ainsi que la plupart des caractères de ponctuation. Ceci dit, certains utilitaires, tel **useradd** interdisent l'utilisation de majuscules et de caractères de ponctuation mais permettent l'utilisation des caractères **_**, **.** ainsi que le caractère **$** à la fin du nom (**ATTENTION** : dans le cas de samba, un nom d'utilisateur se terminant par **$** est considéré comme un compte **machine**). Qui plus est, certains utilitaires limitent la longueur du nom à **8** caractères.

Pour lister les comptes utilisateur existants sur le système, saisissez la commande suivante :

```
root@debian11:~# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-network:x:101:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:102:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:109::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:104:110:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
usbmux:x:105:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
rtkit:x:106:113:RealtimeKit,,,:/proc:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
avahi:x:108:114:Avahi mDNS daemon,,,:/run/avahi-daemon:/usr/sbin/nologin
speech-dispatcher:x:109:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/false
pulse:x:110:116:PulseAudio daemon,,,:/run/pulse:/usr/sbin/nologin
saned:x:111:119::/var/lib/saned:/usr/sbin/nologin
colord:x:112:120:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
```

```
lightdm:x:113:121:Light Display Manager:/var/lib/lightdm:/bin/false
trainee:x:1000:1000:trainee,,,:/home/trainee:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
sshd:x:114:65534::/run/sshd:/usr/sbin/nologin
```

Important : Notez que la valeur de l'UID de root est toujours de 0. Notez que sous Debian 11, les UID des utilisateurs normaux commencent à **1000** et les UID des comptes système sont inclus entre 1 et 99 et entre 201 et 999.

Chaque ligne est constituée de 7 champs :

- Le nom d'utilisateur
- Le mot de passe. Une valeur de **x** dans ce champs indique que le système utilise le fichier **/etc/shadow** pour stocker les mots de passe.
- L'UID. Une valeur unique qui est utilisée pour déterminée les droits aux fichiers et aux répertoires.
- Le GID. Une valeur indiquant le groupe **principal** de l'utilisateur
- Le nom complet. Ce champs optionnel est aussi appelé **GECOS**
- Le répertoire personnel de l'utilisateur
- Le shell de l'utilisateur.

Pour consulter le fichier **/etc/shadow**, saisissez la commande suivante :

```
root@debian11:~# cat /etc/shadow
root:$y$j9T$3oULwcP4KCW0crXb9zLB90$Tqr6eSITrKaEnKecir1vRGXpa1OdRRi3/Q.gLwLPph/:19107:0:99999:7:::
daemon:*:19107:0:99999:7:::
bin:*:19107:0:99999:7:::
sys:*:19107:0:99999:7:::
sync:*:19107:0:99999:7:::
games:*:19107:0:99999:7:::
man:*:19107:0:99999:7:::
lp:*:19107:0:99999:7:::
mail:*:19107:0:99999:7:::
news:*:19107:0:99999:7:::
uucp:*:19107:0:99999:7:::
```

```
proxy:*:19107:0:99999:7:::
www-data:*:19107:0:99999:7:::
backup:*:19107:0:99999:7:::
list:*:19107:0:99999:7:::
irc:*:19107:0:99999:7:::
gnats:*:19107:0:99999:7:::
nobody:*:19107:0:99999:7:::
_apt:*:19107:0:99999:7:::
systemd-network:*:19107:0:99999:7:::
systemd-resolve:*:19107:0:99999:7:::
messagebus:*:19107:0:99999:7:::
systemd-timesync:*:19107:0:99999:7:::
usbmux:*:19107:0:99999:7:::
rtkit:*:19107:0:99999:7:::
dnsmasq:*:19107:0:99999:7:::
avahi:*:19107:0:99999:7:::
speech-dispatcher:!:19107:0:99999:7:::
pulse:*:19107:0:99999:7:::
saned:*:19107:0:99999:7:::
colord:*:19107:0:99999:7:::
lightdm:*:19107:0:99999:7:::
trainee:$y$j9T$iKJ5MC8LmULY.yq58DCjw1$WsIdItQEonaSeCFZil61bk4YPxSp1.5aFg0uDhwIbZC:19107:0:99999:7:::
systemd-coredump:!*:19107::::::
sshd:*:19107:0:99999:7:::
```

Chaque ligne est constituée de 8 champs :

- Le nom de l'utilisateur. Ce champs est utilisé pour faire le lien avec le fichier **/etc/passwd**,
- Le mot de passe **crypté** de l'utilisateur. Le cryptage est à sens **unique**. Ce champ peut aussi contenir une des trois valeurs suivantes :
  - **!!** - Le mot de passe n'a pas encore été défini et l'utilisateur ne peut pas se connecter,
  - * - L'utilisateur ne peut pas se connecter,
  - **vide** - aucun mot de passe sera demandé pour l'utilisateur concerné,
- Le nombre de jours entre le **01/01/1970** et le dernier changement du mot de passe,
- Le nombre de jours que le mot de passe est encore valide. Une valeur de **0** dans ce champs indique sue le mot de passe n'expire jamais,

- Le nombre de jours après lequel le mot de passe doit être changé,
- Le nombre de jours avant la date de modification forcée que l'utilisateur recevra un avertissement,
- Le nombre de jours après l'expiration du mot de passe que le compte sera désactivé,
- Le **numéro** du jour après le **01/01/1970** que le compte a été désactivé.

Afin de vérifier les fichiers **/etc/passwd** et **/etc/shadow** pour des erreurs éventuelles, saisissez la commande suivante :

```
root@debian11:~# pwck -r
user 'lp': directory '/var/spool/lpd' does not exist
user 'news': directory '/var/spool/news' does not exist
user 'uucp': directory '/var/spool/uucp' does not exist
user 'www-data': directory '/var/www' does not exist
user 'list': directory '/var/list' does not exist
user 'irc': directory '/run/ircd' does not exist
user 'gnats': directory '/var/lib/gnats' does not exist
user 'nobody': directory '/nonexistent' does not exist
user '_apt': directory '/nonexistent' does not exist
pwck: no changes
```

Important : Les erreurs ci-dessus ne sont pas importantes. Elles sont dues au fait que les répertoires de connexion de certains comptes systèmes ne sont pas créés par le système lors de la création des comptes et ceci justement pour éviter la possibilité qu'un pirate ou un hacker puisse se connecter au système en utilisant le compte concerné. Encore une fois, l'option **-r** permet la vérification des erreurs dans sans le modifier.

Dans le cas où il est nécessaire de régénérer un des deux fichiers, il convient d'utiliser une des deux commandes suivantes :

- **pwconv**
  - permet de régénérer le fichier **/etc/shadow** à partir du fichier **/etc/passwd** et éventuellement du fichier **/etc/shadow** existant
- **pwunconv**
  - permet de régénérer le fichier **/etc/passwd** à partir du fichier **/etc/shadow** et éventuellement du fichier **/etc/passwd** existant puis supprime le fichier **/etc/shadow**

Commandes

Groupes

groupadd

Cette commande est utilisée pour créer un groupe.

Les options de cette commande sont :

```
root@debian11:~# groupadd --help
Usage: groupadd [options] GROUP

Options:
  -f, --force                   exit successfully if the group already exists,
                                and cancel -g if the GID is already used
  -g, --gid GID                 use GID for the new group
  -h, --help                    display this help message and exit
  -K, --key KEY=VALUE           override /etc/login.defs defaults
  -o, --non-unique              allow to create groups with duplicate
                                (non-unique) GID
  -p, --password PASSWORD       use this encrypted password for the new group
  -r, --system                  create a system account
  -R, --root CHROOT_DIR         directory to chroot into
  -P, --prefix PREFIX_DIR       directory prefix
```

Important : Il est possible de créer plusieurs groupes ayant le même GID.

Important : Notez l'option **-r** qui permet la création d'un groupe système.

groupdel

Cette commande est utilisée pour supprimer un groupe.

Les options de cette commande sont :

```
root@debian11:~# groupdel --help
Usage: groupdel [options] GROUP

Options:
  -h, --help                    display this help message and exit
  -R, --root CHROOT_DIR         directory to chroot into
  -P, --prefix PREFIX_DIR       prefix directory where are located the /etc/* files
  -f, --force                   delete group even if it is the primary group of a user
```

groupmod

Cette commande est utilisée pour modifier un groupe existant.

Les options de cette commande sont :

```
root@debian11:~# groupmod --help
Usage: groupmod [options] GROUP

Options:
  -g, --gid GID                 change the group ID to GID
```

```
  -h, --help                    display this help message and exit
  -n, --new-name NEW_GROUP      change the name to NEW_GROUP
  -o, --non-unique              allow to use a duplicate (non-unique) GID
  -p, --password PASSWORD       change the password to this (encrypted)
                                PASSWORD
  -R, --root CHROOT_DIR         directory to chroot into
  -P, --prefix PREFIX_DIR       prefix directory where are located the /etc/* files
```

newgrp

Cette commande est utilisée pour modifier le groupe de l'utilisateur qui l'invoque.

Les options de cette commande sont :

```
root@debian11:~# newgrp --help
Usage: newgrp [-] [group]
```

gpasswd

Cette commande est utilisée pour administrer le fichier **/etc/group**.

Les options de cette commande sont :

```
root@debian11:~# gpasswd --help
Usage: gpasswd [option] GROUP

Options:
  -a, --add USER                add USER to GROUP
  -d, --delete USER             remove USER from GROUP
  -h, --help                    display this help message and exit
  -Q, --root CHROOT_DIR         directory to chroot into
```

```
  -r, --remove-password         remove the GROUP's password
  -R, --restrict                restrict access to GROUP to its members
  -M, --members USER,...        set the list of members of GROUP
  -A, --administrators ADMIN,...
                                set the list of administrators for GROUP
Except for the -A and -M options, the options cannot be combined.
```

Utilisateurs

useradd

Cette commande est utilisée pour ajouter un utilisateur.

Les codes retour de la commande useradd sont :

Code Retour	Description
1	Impossible de mettre à jour le fichier /etc/passwd
2	Syntaxe invalide
3	Option invalide
4	L'UID demandé est déjà utilisé
6	Le groupe spécifié n'existe pas
9	Le nom d'utilisateur indiqué existe déjà
10	Impossible de mettre à jour le fichier /etc/group
12	Impossible de créer le répertoire personnel de l'utilisateur
13	Impossible de créer le spool mail de l'utilisateur

Les options de cette commande sont :

```
root@debian11:~# useradd --help
Usage: useradd [options] LOGIN
       useradd -D
```

```
       useradd -D [options]

Options:
      --badnames                do not check for bad names
  -b, --base-dir BASE_DIR       base directory for the home directory of the
                                new account
      --btrfs-subvolume-home    use BTRFS subvolume for home directory
  -c, --comment COMMENT         GECOS field of the new account
  -d, --home-dir HOME_DIR       home directory of the new account
  -D, --defaults                print or change default useradd configuration
  -e, --expiredate EXPIRE_DATE  expiration date of the new account
  -f, --inactive INACTIVE       password inactivity period of the new account
  -g, --gid GROUP               name or ID of the primary group of the new
                                account
  -G, --groups GROUPS           list of supplementary groups of the new
                                account
  -h, --help                    display this help message and exit
  -k, --skel SKEL_DIR           use this alternative skeleton directory
  -K, --key KEY=VALUE           override /etc/login.defs defaults
  -l, --no-log-init             do not add the user to the lastlog and
                                faillog databases
  -m, --create-home             create the user's home directory
  -M, --no-create-home          do not create the user's home directory
  -N, --no-user-group           do not create a group with the same name as
                                the user
  -o, --non-unique              allow to create users with duplicate
                                (non-unique) UID
  -p, --password PASSWORD       encrypted password of the new account
  -r, --system                  create a system account
  -R, --root CHROOT_DIR         directory to chroot into
  -P, --prefix PREFIX_DIR       prefix directory where are located the /etc/* files
  -s, --shell SHELL             login shell of the new account
  -u, --uid UID                 user ID of the new account
  -U, --user-group              create a group with the same name as the user
```

```
  -Z, --selinux-user SEUSER     use a specific SEUSER for the SELinux user mapping
```

Important : Il est possible de créer plusieurs utilisateurs ayant le même UID.

Important : Notez l'option **-r** qui permet la création d'un compte système. Dans ce cas la commande useradd ne crée pas de répertoire personnel.

userdel

Cette commande est utilisée pour supprimer un utilisateur.

Les options de cette commande sont :

```
root@debian11:~# userdel --help
Usage: userdel [options] LOGIN

Options:
  -f, --force                   force removal of files,
                                even if not owned by user
  -h, --help                    display this help message and exit
  -r, --remove                  remove home directory and mail spool
  -R, --root CHROOT_DIR         directory to chroot into
  -P, --prefix PREFIX_DIR       prefix directory where are located the /etc/* files
  -Z, --selinux-user            remove any SELinux user mapping for the user
```

Important : Notez que lors de la suppression d'un utilisateur, l'UID associé avec ce compte peut être réutilisé. Le nombre maximum de comptes était de **65 536** avec le noyau **2.2.x**. Avec les noyaux récents, cette limite passe à plus de 4,2 Milliards.

usermod

Cette commande est utilisée pour modifier un utilisateur existant.

Les options de cette commande sont :

```
root@debian11:~# usermod --help
Usage: usermod [options] LOGIN

Options:
  -b, --badnames                allow bad names
  -c, --comment COMMENT         new value of the GECOS field
  -d, --home HOME_DIR           new home directory for the user account
  -e, --expiredate EXPIRE_DATE  set account expiration date to EXPIRE_DATE
  -f, --inactive INACTIVE       set password inactive after expiration
                                to INACTIVE
  -g, --gid GROUP               force use GROUP as new primary group
  -G, --groups GROUPS           new list of supplementary GROUPS
  -a, --append                  append the user to the supplemental GROUPS
                                mentioned by the -G option without removing
                                the user from other groups
  -h, --help                    display this help message and exit
  -l, --login NEW_LOGIN         new value of the login name
  -L, --lock                    lock the user account
  -m, --move-home               move contents of the home directory to the
                                new location (use only with -d)
  -o, --non-unique              allow using duplicate (non-unique) UID
  -p, --password PASSWORD       use encrypted password for the new password
```

```
  -R, --root CHROOT_DIR         directory to chroot into
  -P, --prefix PREFIX_DIR       prefix directory where are located the /etc/* files
  -s, --shell SHELL             new login shell for the user account
  -u, --uid UID                 new UID for the user account
  -U, --unlock                  unlock the user account
  -v, --add-subuids FIRST-LAST  add range of subordinate uids
  -V, --del-subuids FIRST-LAST  remove range of subordinate uids
  -w, --add-subgids FIRST-LAST  add range of subordinate gids
  -W, --del-subgids FIRST-LAST  remove range of subordinate gids
  -Z, --selinux-user SEUSER     new SELinux user mapping for the user account
```

Important : Notez l'option **-L** qui permet de verrouiller un compte.

passwd

Cette commande est utilisée pour créer ou modifier le mot de passe d'un utilisateur.

Les options de cette commande sont :

```
root@debian11:~# passwd --help
Usage: passwd [options] [LOGIN]

Options:
  -a, --all                     report password status on all accounts
  -d, --delete                  delete the password for the named account
  -e, --expire                  force expire the password for the named account
  -h, --help                    display this help message and exit
  -k, --keep-tokens             change password only if expired
  -i, --inactive INACTIVE       set password inactive after expiration
                                to INACTIVE
```

```
  -l, --lock                    lock the password of the named account
  -n, --mindays MIN_DAYS        set minimum number of days before password
                                change to MIN_DAYS
  -q, --quiet                   quiet mode
  -r, --repository REPOSITORY   change password in REPOSITORY repository
  -R, --root CHROOT_DIR         directory to chroot into
  -S, --status                  report password status on the named account
  -u, --unlock                  unlock the password of the named account
  -w, --warndays WARN_DAYS      set expiration warning days to WARN_DAYS
  -x, --maxdays MAX_DAYS        set maximum number of days before password
                                change to MAX_DAYS
```

Important : Notez l'option **-l** qui permet de verrouiller un compte en plaçant le caractère **!** devant le mot de passe crypté.

chage

La commande chage modifie le nombre de jours entre les changements de mot de passe et la date du dernier changement. Ces informations sont utilisées par le système pour déterminer si un utilisateur doit changer son mot de passe.

Les options de cette commande sont :

```
root@debian11:~# chage --help
Usage: chage [options] LOGIN

Options:
  -d, --lastday LAST_DAY        set date of last password change to LAST_DAY
  -E, --expiredate EXPIRE_DATE  set account expiration date to EXPIRE_DATE
  -h, --help                    display this help message and exit
  -i, --iso8601                 use YYYY-MM-DD when printing dates
  -I, --inactive INACTIVE       set password inactive after expiration
```

```
                                to INACTIVE
  -l, --list                    show account aging information
  -m, --mindays MIN_DAYS        set minimum number of days before password
                                change to MIN_DAYS
  -M, --maxdays MAX_DAYS        set maximum number of days before password
                                change to MAX_DAYS
  -R, --root CHROOT_DIR         directory to chroot into
  -W, --warndays WARN_DAYS      set expiration warning days to WARN_DAYS
```

Configuration

La commande **useradd** est configurée par le fichier **/etc/default/useradd**. Pour consulter ce fichier, saisissez la commande suivante :

```
root@debian11:~# cat /etc/default/useradd
# Default values for useradd(8)
#
# The SHELL variable specifies the default login shell on your
# system.
# Similar to DSHELL in adduser. However, we use "sh" here because
# useradd is a low level utility and should be as general
# as possible
SHELL=/bin/sh
#
# The default group for users
# 100=users on Debian systems
# Same as USERS_GID in adduser
# This argument is used when the -n flag is specified.
# The default behavior (when -n and -g are not specified) is to create a
# primary user group with the same name as the user being added to the
# system.
# GROUP=100
#
```

```
# The default home directory. Same as DHOME for adduser
# HOME=/home
#
# The number of days after a password expires until the account
# is permanently disabled
# INACTIVE=-1
#
# The default expire date
# EXPIRE=
#
# The SKEL variable specifies the directory containing "skeletal" user
# files; in other words, files such as a sample .profile that will be
# copied to the new user's home directory when it is created.
# SKEL=/etc/skel
#
# Defines whether the mail spool should be created while
# creating the account
# CREATE_MAIL_SPOOL=yes
```

Dans ce fichier, nous trouvons les directives suivantes :

- **SHELL** - renseigne le shell de l'utilisateur,
- **GROUP** - identifie le groupe principal par défaut de l'utilisateur quand l'option **-N** est utilisée avec la commande **useradd**. Dans le cas contraire le groupe principal est soit le groupe spécifié par l'option **-g** de la commande, soit un nouveau groupe au même nom que l'utilisateur,
- **HOME** - indique que le répertoire personnel de l'utilisateur sera créé dans le répertoire **home** lors de la création du compte si cette option a été activée dans le fichier **/etc/login.defs**,
- **INACTIVE** - indique le nombre de jours d'inactivité après l'expiration d'un mot de passe avant que le compte soit verrouillé. La valeur de -1 désactive cette directive,
- **EXPIRE** - sans valeur, cette directive indique que le mot de passe de l'utilisateur n'expire jamais,
- **SKEL** - indique le répertoire contenant les fichiers qui seront copiés vers le répertoire personnel de l'utilisateur, si ce répertoire est créé lors de la création de l'utilisateur,
- **CREATE_MAIL_SPOOL** - indique si oui ou non une boite mail interne au système sera créée pour l'utilisateur.

Il est aussi possible de renseigner la valeur de la directive suivante :

- **UMASK** - indique l'umask de l'utilisateur. Cette valeur, si présente, prend le dessus sur la valeur indiqué dans le fichier **/etc/login.defs**.

Ces mêmes informations peuvent être visualisées en exécutant la commande **useradd** avec l'option **-D** :

```
root@debian11:~# useradd -D
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/sh
SKEL=/etc/skel
CREATE_MAIL_SPOOL=no
```

Pour consulter la liste des fichiers dans **/etc/skel**, saisissez la commande suivante :

```
root@debian11:~# ls -la /etc/skel
total 20
drwxr-xr-x   2 root root 4096 Apr 25 06:29 .
drwxr-xr-x 112 root root 4096 Jun  2 16:42 ..
-rw-r--r--   1 root root  220 Aug  4  2021 .bash_logout
-rw-r--r--   1 root root 3526 Aug  4  2021 .bashrc
-rw-r--r--   1 root root  807 Aug  4  2021 .profile
```

Important : Notez que sous Debian le fichier **.profile** remplace le fichier **.bash_profile**.

Pour connaître l'UID, le GID et l'appartenance aux groupes d'un utilisateur, il convient d'utiliser la commande **id**. Saisissez la commande suivante :

```
root@debian11:~# id trainee
uid=1000(trainee) gid=1000(trainee)
groups=1000(trainee),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),108(netdev),115(lpadmin),118(sc
anner)
```

Pour seulement connaître les groupes d'un utilisateur, il convient d'utiliser la commande **groups**. Saisissez la commande suivante :

```
root@debian11:~# groups trainee
trainee : trainee cdrom floppy audio dip video plugdev netdev lpadmin scanner
```

Les valeurs minimales de l'UID et du GID utilisés par défaut lors de la création d'un utilisateur sont stipulées dans le fichier **/etc/login.defs** :

```
...
#
# Min/max values for automatic uid selection in useradd
#
UID_MIN                  1000
UID_MAX                 60000
# System accounts
#SYS_UID_MIN              100
#SYS_UID_MAX              999

#
# Min/max values for automatic gid selection in groupadd
#
GID_MIN                  1000
GID_MAX                 60000
# System accounts
#SYS_GID_MIN              100
#SYS_GID_MAX              999
...
```

LAB #1 - Gérer les Utilisateurs et les Groupes

Créez maintenant trois groupes **groupe1**, **groupe2** et **groupe3**. La valeur du GID du groupe **groupe3** doit être de **1807** :

```
root@debian11:~# groupadd groupe1; groupadd groupe2; groupadd -g 1807 groupe3
```

Créez maintenant trois utilisateurs **fenestros1**, **fenestros2** et **fenestros3**. Les trois utilisateurs ont pour groupe principal **groupe1**, **groupe2** et **groupe3** respectivement. **fenestros2** est aussi membre des groupes **groupe1** et **groupe3**. **fenestros1** à un GECOS de **tux1** :

```
root@debian11:~# useradd -g groupe2 fenestros2; useradd -g 1807 fenestros3; useradd -g groupe1 fenestros1

root@debian11:~# usermod -G groupe1,groupe3 fenestros2

root@debian11:~# usermod -c "tux1" fenestros1
```

En consultant la fin de votre fichier **/etc/passwd**, vous obtiendrez un résultat similaire à celui-ci :

```
root@debian11:~# tail /etc/passwd
pulse:x:110:116:PulseAudio daemon,,,:/run/pulse:/usr/sbin/nologin
saned:x:111:119::/var/lib/saned:/usr/sbin/nologin
colord:x:112:120:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
lightdm:x:113:121:Light Display Manager:/var/lib/lightdm:/bin/false
trainee:x:1000:1000:trainee,,,:/home/trainee:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
sshd:x:114:65534::/run/sshd:/usr/sbin/nologin
fenestros2:x:1001:1002::/home/fenestros2:/bin/sh
fenestros3:x:1002:1807::/home/fenestros3:/bin/sh
fenestros1:x:1003:1001:tux1:/home/fenestros1:/bin/sh
```

En regardant la fin de votre fichier **/etc/group**, vous obtiendrez un résultat similaire à celui-ci :

```
root@debian11:~# tail /etc/group
scanner:x:118:saned,trainee
saned:x:119:
colord:x:120:
lightdm:x:121:
trainee:x:1000:
systemd-coredump:x:999:
mlocate:x:122:
groupe1:x:1001:fenestros2
```

```
groupe2:x:1002:
groupe3:x:1807:fenestros2
```

Créez le mot de passe **fenestros** pour le **groupe3** :

```
root@debian11:~# gpasswd groupe3
Changing the password for group groupe3
New Password: fenestros
Re-enter new password: fenestros
```

Important : Notez que les mots de passe saisis ne seront **pas** visibles.

Consultez la fin de votre fichier **/etc/gshadow** :

```
root@debian11:~# tail /etc/gshadow
scanner:!::saned,trainee
saned:!::
colord:!::
lightdm:!::
trainee:!::
systemd-coredump:!*::
mlocate:!::
groupe1:!::fenestros2
groupe2:!::
groupe3:$6$QEb2./zgnba/w1$FZ6hbzzcsZmoraLn2tNUhXd3F8K8yUI14.u7cVp2GPteBLJ6h0iNcjnhCsjS5/Wp8u8pe8nXpn/9UFsOXEGJ90:
:fenestros2
```

Important : Notez la présence du mot de passe crypté pour le **groupe3**.

Nommez maintenant **fenestros1** administrateur du **groupe3** :

```
root@debian11:~# gpasswd -A fenestros1 groupe3
```

Consultez la fin de votre fichier **/etc/gshadow** de nouveau :

```
root@debian11:~# tail /etc/gshadow
scanner:!::saned,trainee
saned:!::
colord:!::
lightdm:!::
trainee:!::
systemd-coredump:!*::
mlocate:!::
groupe1:!::fenestros2
groupe2:!::
groupe3:$6$QEb2./zgnba/w1$FZ6hbzzcsZmoraLn2tNUhXd3F8K8yUI14.u7cVp2GPteBLJ6h0iNcjnhCsjS5/Wp8u8pe8nXpn/9UFsOXEGJ90:
fenestros1:fenestros2
```

Important : L'utilisateur **fenestros1** peut maintenant administrer le groupe **groupe3** en y ajoutant ou en y supprimant des utilisateurs à condition de connaître le mot de passe du groupe.

Essayez maintenant de supprimer le groupe **groupe3** :

```
root@debian11:~# groupdel groupe3
groupdel: cannot remove the primary group of user 'fenestros3'
```

Important : En effet, vous ne pouvez pas supprimer un groupe tant qu'un utilisateur le possède comme son groupe principal.

Supprimez donc l'utilisateur **fenestros3** :

```
root@debian11:~# userdel fenestros3
```

Ensuite essayez de supprimer le groupe **groupe3** :

```
root@debian11:~# groupdel groupe3
```

Important : Notez que cette fois-ci la commande est exécutée sans erreur.

Le fait de supprimer un utilisateur **sans** l'option **-r** implique que le répertoire personnel de l'utilisateur demeure sur la machine.

Sous Debian les répertoires personnels des utilisateurs n'ont pas été créés parce que les directives n'ont pas été activées dans le fichier **/etc/default/useradd** et que nous n'avons pas spécifier la création du répertoire lors de l'utilisation de la commande **useradd** avec l'option **-m** :

```
root@debian11:~# cat /etc/default/useradd
# Default values for useradd(8)
#
# The SHELL variable specifies the default login shell on your
# system.
# Similar to DSHELL in adduser. However, we use "sh" here because
# useradd is a low level utility and should be as general
# as possible
SHELL=/bin/sh
#
# The default group for users
# 100=users on Debian systems
# Same as USERS_GID in adduser
# This argument is used when the -n flag is specified.
# The default behavior (when -n and -g are not specified) is to create a
# primary user group with the same name as the user being added to the
```

```
# system.
# GROUP=100
#
# The default home directory. Same as DHOME for adduser
# HOME=/home
#
# The number of days after a password expires until the account
# is permanently disabled
# INACTIVE=-1
#
# The default expire date
# EXPIRE=
#
# The SKEL variable specifies the directory containing "skeletal" user
# files; in other words, files such as a sample .profile that will be
# copied to the new user's home directory when it is created.
# SKEL=/etc/skel
#
# Defines whether the mail spool should be created while
# creating the account
# CREATE_MAIL_SPOOL=yes
```

Otez donc le caractère **#** devant les lignes suivantes :

```
root@debian11:~# vi /etc/default/useradd

root@debian11:~# cat /etc/default/useradd
# Default values for useradd(8)
#
# The SHELL variable specifies the default login shell on your
# system.
# Similar to DSHELL in adduser. However, we use "sh" here because
# useradd is a low level utility and should be as general
# as possible
```

```
SHELL=/bin/sh
#
# The default group for users
# 100=users on Debian systems
# Same as USERS_GID in adduser
# This argument is used when the -n flag is specified.
# The default behavior (when -n and -g are not specified) is to create a
# primary user group with the same name as the user being added to the
# system.
GROUP=100
#
# The default home directory. Same as DHOME for adduser
HOME=/home
#
# The number of days after a password expires until the account
# is permanently disabled
INACTIVE=-1
#
# The default expire date
EXPIRE=
#
# The SKEL variable specifies the directory containing "skeletal" user
# files; in other words, files such as a sample .profile that will be
# copied to the new user's home directory when it is created.
SKEL=/etc/skel
#
# Defines whether the mail spool should be created while
# creating the account
CREATE_MAIL_SPOOL=yes
```

Pour tester la configuration, créez un utilisateur test :

```
root@debian11:~# useradd -m test
```

Vérifiez que l'utilisateur test a un répertoire personnel :

```
root@debian11:~# ls -l /home
total 8
drwxr-xr-x  2 test    test    4096 Jun  5 13:01 test
drwxr-xr-x 17 trainee trainee 4096 Jun  3 17:39 trainee
```

Créez maintenant les répertoires personnels de fenestros1 et fenestros2 :

```
root@debian11:~# mkdir /home/fenestros1 /home/fenestros2
```

Copiez le contenu du répertoire **/etc/skel** dans les répertoires **/home/fenestros1** et **/home/fenestros2** :

```
root@debian11:~# cp -r /etc/skel/.[a-zA-Z]* /home/fenestros1
root@debian11:~# cp -r /etc/skel/.[a-zA-Z]* /home/fenestros2
```

Modifiez le propriétaire et le groupe pour les répertoires **/home/fenestros1** et **/home/fenestros2** :

```
root@debian11:~# chown -R fenestros1:groupe1 /home/fenestros1
root@debian11:~# chown -R fenestros2:groupe2 /home/fenestros2
```

Créez maintenant les mots de passe pour **fenestros1** et **fenestros2**. Indiquez un mot de passe identique au nom du compte :

```
root@debian11:~# passwd fenestros1
New password: fenestros1
Retype new password: fenestros1
passwd: password updated successfully
root@debian11:~# passwd fenestros2
New password: fenestros2
Retype new password: fenestros2
passwd: password updated successfully
```

Important : Notez que les règles concernant la création de mots de passe ne sont pas appliqués aux mots de passe créés par root. Notez aussi que les mots de passe saisis ne seront **PAS** visibles.

LAB #2 - Forcer l'utilisation des mots de passe complexe avec PAM

2.1 - Présentation de PAM

PAM (*Pluggable Authentification Modules* ou Modules d'Authentification Enfichables) est une architecture modulaire permettant à l'administrateur système de définir une politique d'authentification pour les logiciels prenant en charge PAM.

Les fichiers de configuration se trouvent dans le répertoire **/etc/pam.d** :

```
root@debian11:~# ls -l /etc/pam.d
total 104
-rw-r--r-- 1 root root  384 Feb  7  2020 chfn
-rw-r--r-- 1 root root   92 Feb  7  2020 chpasswd
-rw-r--r-- 1 root root  581 Feb  7  2020 chsh
-rw-r--r-- 1 root root 1208 Apr 25 06:49 common-account
-rw-r--r-- 1 root root 1214 Apr 25 06:49 common-auth
-rw-r--r-- 1 root root 1660 Apr 25 06:49 common-password
-rw-r--r-- 1 root root 1146 Apr 25 06:49 common-session
-rw-r--r-- 1 root root 1154 Apr 25 06:49 common-session-noninteractive
-rw-r--r-- 1 root root  606 Feb 22  2021 cron
-rw-r--r-- 1 root root   69 May 27  2021 cups
-rw-r--r-- 1 root root 1354 Feb  3  2020 lightdm
-rw-r--r-- 1 root root 1428 Feb  3  2020 lightdm-autologin
-rw-r--r-- 1 root root  493 Feb  3  2020 lightdm-greeter
-rw-r--r-- 1 root root 4126 Feb  7  2020 login
-rw-r--r-- 1 root root   92 Feb  7  2020 newusers
-rw-r--r-- 1 root root  520 Jan 30  2021 other
```

```
-rw-r--r-- 1 root root   92 Feb  7  2020 passwd
-rw-r--r-- 1 root root  270 Jan 13 20:32 polkit-1
-rw-r--r-- 1 root root  168 Jan  7  2021 ppp
-rw-r--r-- 1 root root  143 Jan 20 20:55 runuser
-rw-r--r-- 1 root root  138 Jan 20 20:55 runuser-l
-rw-r--r-- 1 root root 2133 Mar 13  2021 sshd
-rw-r--r-- 1 root root 2259 Jan 20 20:55 su
-rw-r--r-- 1 root root   95 Feb 27  2021 sudo
-rw-r--r-- 1 root root  137 Jan 20 20:55 su-l
```

Ces fichiers ont une structure spécifique et sont nommés d'après le service ou l'application qu'ils contrôlent. Leur contenu fait appel à des modules qui se trouvent dans le répertoire **/lib/x86_64-linux-gnu/security** :

```
root@debian11:~# ls /lib/x86_64-linux-gnu/security
pam_access.so  pam_faildelay.so      pam_issue.so      pam_loginuid.so   pam_permit.so     pam_sepermit.so
pam_time.so       pam_usertype.so
pam_debug.so   pam_faillock.so       pam_keyinit.so    pam_mail.so       pam_pwhistory.so  pam_setquota.so
pam_timestamp.so  pam_warn.so
pam_deny.so    pam_filter.so         pam_lastlog.so    pam_mkhomedir.so  pam_rhosts.so     pam_shells.so
pam_tty_audit.so  pam_wheel.so
pam_echo.so    pam_ftp.so            pam_limits.so     pam_motd.so       pam_rootok.so     pam_stress.so
pam_umask.so      pam_xauth.so
pam_env.so     pam_gnome_keyring.so  pam_listfile.so   pam_namespace.so  pam_securetty.so  pam_succeed_if.so
pam_unix.so
pam_exec.so    pam_group.so          pam_localuser.so  pam_nologin.so    pam_selinux.so    pam_systemd.so
pam_userdb.so
```

Les modules les plus importants sont :

Module	**Description**
pam_access.so	Ce module est utilisé pour interdire l'accès aux services sécurisés par des hôtes non-autorisés.
pam_echo.so	Ce module présente le contenu du fichier passé en argument à tout utilisateur lors de sa connexion.
pam_limits.so	Ce module implémente les limites des ressources détaillées dans le fichier **/etc/security/limits.conf** et dans les fichiers ***.conf** trouvés dans le répertoire **/etc/security/limits.d/**.

Module	Description
pam_listfile.so	Ce module est utilisé pour consulter un fichier spécifique pour vérifier les authiorisations. Par exemple, le service ftp utilise ce module pour consulter le fichier **/etc/ftpusers** qui contient une liste d'utilisateurs qui ne sont **pas** autorisés à se connecter au serveur ftp.
pam_nologin.so	Ce module interdit les connexions d'utilisteurs, autre que root, dans le cas où le fichier **/etc/nologin** est présent.
pam_securetty.so	Ce module interdit des connexions de root à partir des périphériques tty qui ne sont pas listés dans le fichier **/etc/securetty**.
pam_unix.so	Ce module est utilisé pour vérifier les informations suivantes ; expire, last_change, max_change, min_change, warn_change.

Chaque fichier dans /etc/pam.d contient les règles PAM utilisées pendant l'authentification. Ouvrez le fichier **login** :

```
root@debian11:~# cat /etc/pam.d/login
#
# The PAM configuration file for the Shadow `login' service
#

# Enforce a minimal delay in case of failure (in microseconds).
# (Replaces the `FAIL_DELAY' setting from login.defs)
# Note that other modules may require another minimal delay. (for example,
# to disable any delay, you should add the nodelay option to pam_unix)
auth       optional   pam_faildelay.so  delay=3000000

# Outputs an issue file prior to each login prompt (Replaces the
# ISSUE_FILE option from login.defs). Uncomment for use
# auth       required   pam_issue.so issue=/etc/issue

# Disallows other than root logins when /etc/nologin exists
# (Replaces the `NOLOGINS_FILE' option from login.defs)
auth       requisite  pam_nologin.so

# SELinux needs to be the first session rule. This ensures that any
# lingering context has been cleared. Without this it is possible
# that a module could execute code in the wrong domain.
# When the module is present, "required" would be sufficient (When SELinux
# is disabled, this returns success.)
```

```
session [success=ok ignore=ignore module_unknown=ignore default=bad] pam_selinux.so close

# Sets the loginuid process attribute
session    required     pam_loginuid.so

# Prints the message of the day upon successful login.
# (Replaces the `MOTD_FILE' option in login.defs)
# This includes a dynamically generated part from /run/motd.dynamic
# and a static (admin-editable) part from /etc/motd.
session    optional   pam_motd.so motd=/run/motd.dynamic
session    optional   pam_motd.so noupdate

# SELinux needs to intervene at login time to ensure that the process
# starts in the proper default security context. Only sessions which are
# intended to run in the user's context should be run after this.
# pam_selinux.so changes the SELinux context of the used TTY and configures
# SELinux in order to transition to the user context with the next execve()
# call.
session [success=ok ignore=ignore module_unknown=ignore default=bad] pam_selinux.so open
# When the module is present, "required" would be sufficient (When SELinux
# is disabled, this returns success.)

# This module parses environment configuration file(s)
# and also allows you to use an extended config
# file /etc/security/pam_env.conf.
#
# parsing /etc/environment needs "readenv=1"
session       required   pam_env.so readenv=1
# locale variables are also kept into /etc/default/locale in etch
# reading this file *in addition to /etc/environment* does not hurt
session       required   pam_env.so readenv=1 envfile=/etc/default/locale

# Standard Un*x authentication.
@include common-auth
```

```
# This allows certain extra groups to be granted to a user
# based on things like time of day, tty, service, and user.
# Please edit /etc/security/group.conf to fit your needs
# (Replaces the `CONSOLE_GROUPS' option in login.defs)
auth       optional   pam_group.so

# Uncomment and edit /etc/security/time.conf if you need to set
# time restraint on logins.
# (Replaces the `PORTTIME_CHECKS_ENAB' option from login.defs
# as well as /etc/porttime)
# account    requisite  pam_time.so

# Uncomment and edit /etc/security/access.conf if you need to
# set access limits.
# (Replaces /etc/login.access file)
# account  required       pam_access.so

# Sets up user limits according to /etc/security/limits.conf
# (Replaces the use of /etc/limits in old login)
session    required   pam_limits.so

# Prints the last login info upon successful login
# (Replaces the `LASTLOG_ENAB' option from login.defs)
session    optional   pam_lastlog.so

# Prints the status of the user's mailbox upon successful login
# (Replaces the `MAIL_CHECK_ENAB' option from login.defs).
#
# This also defines the MAIL environment variable
# However, userdel also needs MAIL_DIR and MAIL_FILE variables
# in /etc/login.defs to make sure that removing a user
# also removes the user's mail spool file.
# See comments in /etc/login.defs
session    optional   pam_mail.so standard
```

```
# Create a new session keyring.
session    optional   pam_keyinit.so force revoke

# Standard Un*x account and session
@include common-account
@include common-session
@include common-password
```

La première ligne de ce fichier est un commentaire qui spécifie que le fichier est conforme à la spécification PAM 1.0.

Ce fichier, tout comme les autres, est ensuite structuré de la façon suivante :

- Une module par ligne,
- Quatre champs séparés par un espace dans chaque règle dont les trois premières sont obligatoires.

Le **premier champs** est le ***type de module***. Il en existe quatre :

Type	Description
auth	Utilisé pour authentifier un utilisateur ou les pré-requis système (par exemple /etc/nologin)
account	Utilisé pour vérifier si l'utilisateur peut s'authentifier (par exemple la validité du compte)
password	Utilisé pour vérifier si l'utilisateur dispose des droits pour mettre le mécanisme d'authentification à jour
session	Utilisé pour gérer la session après l'authentification (par exemple monter un répertoire)

Le **deuxième champs** est le ***Control-flag***. Il en existe quatre :

Control-flag	Description
required	La réussite de ce module est indispensable. L'échec d'un module *required* n'est communiqué à l'application qu'après la vérification de tous les modules ayant un *control-flag* de **required**
requisite	La réussite de ce module est indispensable. L'échec d'un module *requisite* est immédiatement communiqué à l'application
sufficient	La réussite de ce module est suffisant pour authoriser l'authentification. Si aucun test *required* précédent est en échec, la vérification s'arrête. Si un test *required* précédent était en échec, le test *sufficient* est ignoré. L'échec d'un test *sufficient* n'a pas de conséquence si tous les tests *required* réussissent.
optional	La réussite ou l'échec de ce module est sans importance, **sauf** s'il s'agit du seul module à exécuter
@include	Ce control-flag permet d'inclure toutes les lignes du même *type de module* se trouvant dans le fichier spécifié en argument

Le **troisième champs** stipule le ***module*** associé à la règle. Sans chemin absolu, le fichier est supposé être dans le répertoire **/lib/x86_64-linux-gnu/security**. Pour inclure un module en dehors de ce répertoire il convient donc de stipuler son chemin absolu.

Le **quatrième champs** contient éventuellement les **arguments**.

Dernièrement, PAM propose une solution pour toutes les applications ne disposant pas de leurs propres fichiers de configuration PAM. Cette solution prend la forme du fichier **/etc/pam.d/other** :

```
root@debian11:~# cat /etc/pam.d/other
#
# /etc/pam.d/other - specify the PAM fallback behaviour
#
# Note that this file is used for any unspecified service; for example
#if /etc/pam.d/cron  specifies no session modules but cron calls
#pam_open_session, the session module out of /etc/pam.d/other is
#used.  If you really want nothing to happen then use pam_permit.so or
#pam_deny.so as appropriate.

# We fall back to the system default in /etc/pam.d/common-*
#

@include common-auth
@include common-account
@include common-password
@include common-session
```

2.2 - Configuration des modules

Certains modules de PAM peuvent être configurés grâce aux fichiers présents dans le répertoire **/etc/security** :

```
root@debian11:~# ls -l /etc/security
total 48
-rw-r--r-- 1 root root 4564 Aug 26  2021 access.conf
```

```
-rw-r--r-- 1 root root 2234 Aug 26  2021 faillock.conf
-rw-r--r-- 1 root root 3635 Aug 26  2021 group.conf
-rw-r--r-- 1 root root 2161 Aug 26  2021 limits.conf
drwxr-xr-x 2 root root 4096 Aug 26  2021 limits.d
-rw-r--r-- 1 root root 1637 Aug 26  2021 namespace.conf
drwxr-xr-x 2 root root 4096 Aug 26  2021 namespace.d
-rwxr-xr-x 1 root root 1016 Aug 26  2021 namespace.init
-rw------- 1 root root    0 Apr 25 06:30 opasswd
-rw-r--r-- 1 root root 2971 Aug 26  2021 pam_env.conf
-rw-r--r-- 1 root root  419 Aug 26  2021 sepermit.conf
-rw-r--r-- 1 root root 2179 Aug 26  2021 time.conf
```

Parmi les fichiers cités on note ceux qui peuvent être utilisés pour configurer les modules suivants :

Fichier/*Répertoire*	**Module**
access.conf	pam_access.so
faillock.conf	pam_faillock.so
group.conf	pam_group.so
limits.conf	pam_limits.so
namespace.conf	pam_namespace.so
pam_env.conf	pam_env.so
sepermit.conf	pam_sepermit.so
time.conf	pam_time.so

2.3 - Utiliser des Mots de Passe Complexes

La complexité des mots de passe est gérée par le module **pam_pwquality.so**. Commencez par installer **libpam-pwquality** :

```
root@debian11:~# apt-get -y install libpam-pwquality
```

Vérifiez la présence du module :

```
root@debian11:~# ls /lib/x86_64-linux-gnu/security | grep quality
pam_pwquality.so
```

L'installation du module a aussi installé un fichier de configuration :

```
root@debian11:~# ls -l /etc/security/pwquality.conf
-rw-r--r-- 1 root root 2674 Dec 17  2020 /etc/security/pwquality.conf
```

Afin de mettre en place une politique de mots de passe complexe, il convient de modifier le fichier **/etc/security/pwquality.conf** :

```
root@debian11:~# vi /etc/security/pwquality.conf
root@debian11:~# cat /etc/security/pwquality.conf
# Configuration for systemwide password quality limits
# Defaults:
#
# Number of characters in the new password that must not be present in the
# old password.
# difok = 1
#
# Minimum acceptable size for the new password (plus one if
# credits are not disabled which is the default). (See pam_cracklib manual.)
# Cannot be set to lower value than 6.
minlen = 8
#
# The maximum credit for having digits in the new password. If less than 0
# it is the minimum number of digits in the new password.
dcredit = -2
#
# The maximum credit for having uppercase characters in the new password.
# If less than 0 it is the minimum number of uppercase characters in the new
# password.
ucredit = -1
#
# The maximum credit for having lowercase characters in the new password.
```

```
# If less than 0 it is the minimum number of lowercase characters in the new
# password.
lcredit = -1
#
# The maximum credit for having other characters in the new password.
# If less than 0 it is the minimum number of other characters in the new
# password.
ocredit = -1
#
# The minimum number of required classes of characters for the new
# password (digits, uppercase, lowercase, others).
minclass = 4
#
# The maximum number of allowed consecutive same characters in the new password.
# The check is disabled if the value is 0.
# maxrepeat = 0
#
# The maximum number of allowed consecutive characters of the same class in the
# new password.
# The check is disabled if the value is 0.
# maxclassrepeat = 0
#
# Whether to check for the words from the passwd entry GECOS string of the user.
# The check is enabled if the value is not 0.
gecoscheck = 1
#
# Whether to check for the words from the cracklib dictionary.
# The check is enabled if the value is not 0.
dictcheck = 1
#
# Whether to check if it contains the user name in some form.
# The check is enabled if the value is not 0.
usercheck = 1
#
```

```
# Length of substrings from the username to check for in the password
# The check is enabled if the value is greater than 0 and usercheck is enabled.
# usersubstr = 0
#
# Whether the check is enforced by the PAM module and possibly other
# applications.
# The new password is rejected if it fails the check and the value is not 0.
enforcing = 1
#
# Path to the cracklib dictionaries. Default is to use the cracklib default.
# dictpath =
#
# Prompt user at most N times before returning with error. The default is 1.
# retry = 3
#
# Enforces pwquality checks on the root user password.
# Enabled if the option is present.
# enforce_for_root
#
# Skip testing the password quality for users that are not present in the
# /etc/passwd file.
# Enabled if the option is present.
# local_users_only
```

LAB #3 - su et su -

Vous allez maintenant devenir **fenestros2**, d'abord sans l'environnement de **fenestros2** puis avec l'environnement de **fenestros2**.

Contrôlez votre répertoire courant de travail :

```
root@debian11:~# pwd
/root
```

Pour devenir **fenestros2 sans** son environnement, saisissez la commande suivante :

```
root@debian11:~# su fenestros2
```

Contrôlez votre répertoire courant de travail :

```
$ pwd
/root
```

Vous noterez que vous êtes toujours dans le répertoire **/root**. Ceci indique que vous avez gardé l'environnement de **root**.

Important : L'environnement d'un utilisateur inclut, entre autre, le répertoire personnel de l'utilisateur ainsi que la valeur de la variable système **PATH**.

Saisissez la commande suivante pour redevenir **root** :

```
$ exit
root@debian11:~#
```

Saisissez la commande suivante pour redevenir **fenestros2** :

```
root@debian11:~# su - fenestros2
```

Contrôlez votre répertoire courant de travail :

```
$ pwd
/home/fenestros2
```

Vous noterez que vous êtes maintenant dans le répertoire **/home/fenestros2**. Ceci indique que vous avez l'environnement de **fenestros2**.

Saisissez la commande suivante pour redevenir **root** :

```
$ exit
root@debian11:~#
```

Important : Notez que **root** peut devenir n'importe quel utilisateur **sans** avoir besoin de connaître son mot de passe.

LAB #4 - sudo

La commande **sudo** permet à un utilisateur autorisé d'exécuter une commande en tant que **root** ou en tant qu'un autre utilisateur. Lors de l'exécution de la commande, l'UID et le GID éffectifs et réels sont ceux de l'identité de l'utilisateur cible. L'utilisation de la commande **sudo** est une façon simple de déléguer des tâches administratives à d'autres utilisateurs sans communiquer le mot de passe de **root** et sans placer un SUID bit sur l'exécutable. La commande **sudo** est configurée grâce au fichier **/etc/sudoers**.

Saisissez la commande suivante :

```
root@debian11:~# cat /etc/sudoers
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"

# Host alias specification
```

```
# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

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

# See sudoers(5) for more information on "@include" directives:

@includedir /etc/sudoers.d
```

Important : Notez la présence de la ligne **%sudo ALL=(ALL:ALL) ALL**. Cette ligne possède le format **Qui Hôte = (Utilisateur:Groupe) Commande(s)**. La ligne implique donc que les membres du groupe **sudo** peuvent exécuter à partir de n'importe quel hôte et en tant que n'importe quel utilisateur de n'importe quel groupe, toutes les commandes du système. Dans ce fichier donc, un groupe est référencé par un **%**. Un nom sans ce caractère est forcément un utilisateur. Pour éditer le fichier **/etc/sudoers**, il est **nécessaire** d'utiliser la commande **visudo**.

Devenez l'utilisateur **trainee** :

```
root@debian11:~# exit
logout
trainee@debian11:~$
```

Saisissez la commande suivante :

```
trainee@debian11:~$ sudo su -

We trust you have received the usual lecture from the local System
```

```
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for trainee:
```

Saisissez le mot de passe de **trainee** :

```
[sudo] password for trainee: trainee
trainee is not in the sudoers file.  This incident will be reported.
```

Important : Notez que la commande a échoué car l'utilisateur **trainee** n'est pas référencé dans le fichier **/etc/sudoers**.

Mettez maintenant l'utilisateur **trainee** dans le groupe **sudo** :

```
trainee@debian11:~$ su -
Password: fenestros

root@debian11:~# usermod -aG sudo trainee

root@debian11:~# groups trainee
trainee : trainee cdrom floppy sudo audio dip video plugdev netdev lpadmin scanner

root@debian11:~# exit
logout
```

Constatez que vous ne faites pas parti du groupe **sudo** :

```
trainee@debian11:~$ groups
```

```
trainee cdrom floppy audio dip video plugdev netdev lpadmin scanner
```

Rejoignez le groupe **sudo** :

```
trainee@debian11:~$ newgrp sudo

trainee@debian11:~$ groups
sudo cdrom floppy audio dip video plugdev netdev lpadmin scanner trainee
```

Essayez de nouveau la commande sudo :

```
trainee@debian11:~$ sudo su -
[sudo] password for trainee: trainee
root@debian11:~#
```

Important : Notez que la commande a réussi.
