Version : **2024.01**

Dernière mise-à-jour : 2024/10/08 09:03

RH12406 - Gestion des Utilisateurs

Contenu du Module

        - Utiliser des Mots de Passe Complexe
        - Configuration
    - su et su -
    - sudo

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 sous RHEL 9

```
[root@redhat9 ~]# cat /etc/nsswitch.conf
# Generated by authselect on Wed Sep 25 12:09:11 2024
# Do not modify this file manually.

# If you want to make changes to nsswitch.conf please modify
```

```
# /etc/authselect/user-nsswitch.conf and run 'authselect apply-changes'.
#
# Note that your changes may not be applied as they may be
# overwritten by selected profile. Maps set in the authselect
# profile takes always precedence and overwrites the same maps
# set in the user file. Only maps that are not set by the profile
# are applied from the user file.
#
# For example, if the profile sets:
#     passwd: sss files
# and /etc/authselect/user-nsswitch.conf contains:
#     passwd: files
#     hosts: files dns
# the resulting generated nsswitch.conf will be:
#     passwd: sss files # from profile
#     hosts: files dns  # from user file

passwd:     files sss systemd
group:      files sss systemd
netgroup:   sss files
automount:  sss files
services:   sss files

# Included from /etc/authselect/user-nsswitch.conf

#
# /etc/nsswitch.conf
#
# Name Service Switch config file. This file should be
# sorted with the most-used services at the beginning.
#
# Valid databases are: aliases, ethers, group, gshadow, hosts,
# initgroups, netgroup, networks, passwd, protocols, publickey,
# rpc, services, and shadow.
```

```
#
# Valid service provider entries include (in alphabetical order):
#
#       compat                  Use /etc files plus *_compat pseudo-db
#       db                      Use the pre-processed /var/db files
#       dns                     Use DNS (Domain Name Service)
#       files                   Use the local files in /etc
#       hesiod                  Use Hesiod (DNS) for user lookups
#
# See `info libc 'NSS Basics'` for more information.
#
# Commonly used alternative service providers (may need installation):
#
#       ldap                    Use LDAP directory server
#       myhostname              Use systemd host names
#       mymachines              Use systemd machine names
#       mdns*, mdns*_minimal    Use Avahi mDNS/DNS-SD
#       resolve                 Use systemd resolved resolver
#       sss                     Use System Security Services Daemon (sssd)
#       systemd                 Use systemd for dynamic user option
#       winbind                 Use Samba winbind support
#       wins                    Use Samba wins support
#       wrapper                 Use wrapper module for testing
#
# Notes:
#
# 'sssd' performs its own 'files'-based caching, so it should generally
# come before 'files'.
#
# WARNING: Running nscd with a secondary caching service like sssd may
#          lead to unexpected behaviour, especially with how long
#          entries are cached.
#
# Installation instructions:
```

```
#
# To use 'db', install the appropriate package(s) (provide 'makedb' and
# libnss_db.so.*), and place the 'db' in front of 'files' for entries
# you want to be looked up first in the databases, like this:
#
# passwd:    db files
# shadow:    db files
# group:     db files

# In order of likelihood of use to accelerate lookup.
shadow:     files
hosts:      files dns myhostname

aliases:    files
ethers:     files
gshadow:    files
# Allow initgroups to default to the setting for group.
# initgroups: files
networks:   files dns
protocols:  files
publickey:  files
rpc:        files
```

Dans ce fichier :

- **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**.
- **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).

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@redhat9 ~]# 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@redhat9 ~]# getent group mail
mail:x:12:
```

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

```
[root@redhat9 ~]# getent passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:65534:65534:Kernel Overflow User:/:/sbin/nologin
systemd-coredump:x:999:997:systemd Core Dumper:/:/sbin/nologin
```

```
dbus:x:81:81:System message bus:/:/sbin/nologin
polkitd:x:998:996:User for polkitd:/:/sbin/nologin
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin
tss:x:59:59:Account used for TPM access:/dev/null:/sbin/nologin
colord:x:997:993:User for colord:/var/lib/colord:/sbin/nologin
clevis:x:996:992:Clevis Decryption Framework unprivileged user:/var/cache/clevis:/usr/sbin/nologin
rtkit:x:172:172:RealtimeKit:/proc:/sbin/nologin
sssd:x:995:991:User for sssd:/:/sbin/nologin
geoclue:x:994:990:User for geoclue:/var/lib/geoclue:/sbin/nologin
libstoragemgmt:x:988:988:daemon account for libstoragemgmt:/:/usr/sbin/nologin
systemd-oom:x:987:987:systemd Userspace OOM Killer:/:/usr/sbin/nologin
setroubleshoot:x:986:986:SELinux troubleshoot server:/var/lib/setroubleshoot:/sbin/nologin
pipewire:x:985:984:PipeWire System Daemon:/var/run/pipewire:/sbin/nologin
flatpak:x:984:983:User for flatpak system helper:/:/sbin/nologin
gdm:x:42:42::/var/lib/gdm:/sbin/nologin
cockpit-ws:x:983:982:User for cockpit web service:/nonexisting:/sbin/nologin
cockpit-wsinstance:x:982:981:User for cockpit-ws instances:/nonexisting:/sbin/nologin
gnome-initial-setup:x:981:980::/run/gnome-initial-setup/:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/usr/share/empty.sshd:/sbin/nologin
chrony:x:980:979:chrony system user:/var/lib/chrony:/sbin/nologin
dnsmasq:x:979:978:Dnsmasq DHCP and DNS server:/var/lib/dnsmasq:/sbin/nologin
tcpdump:x:72:72::/:/sbin/nologin
trainee:x:1000:1000:trainee:/home/trainee:/bin/bash
```

Les Fichiers /etc/group et /etc/gshadow

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

```
[root@redhat9 ~]# cat /etc/group
root:x:0:
bin:x:1:
daemon:x:2:
sys:x:3:
```

```
adm:x:4:
tty:x:5:
disk:x:6:
lp:x:7:
mem:x:8:
kmem:x:9:
wheel:x:10:
cdrom:x:11:
mail:x:12:
man:x:15:
dialout:x:18:
floppy:x:19:
games:x:20:
tape:x:33:
video:x:39:
ftp:x:50:
lock:x:54:
audio:x:63:
users:x:100:
nobody:x:65534:
utmp:x:22:
utempter:x:35:
input:x:999:
kvm:x:36:
render:x:998:
systemd-journal:x:190:
systemd-coredump:x:997:
dbus:x:81:
polkitd:x:996:
printadmin:x:995:
ssh_keys:x:994:
avahi:x:70:
tss:x:59:clevis
colord:x:993:
```

```
clevis:x:992:
rtkit:x:172:
sssd:x:991:
geoclue:x:990:
sgx:x:989:
libstoragemgmt:x:988:
systemd-oom:x:987:
setroubleshoot:x:986:
brlapi:x:985:
pipewire:x:984:
flatpak:x:983:
gdm:x:42:
cockpit-ws:x:982:
cockpit-wsinstance:x:981:
gnome-initial-setup:x:980:
sshd:x:74:
chrony:x:979:
slocate:x:21:
dnsmasq:x:978:
tcpdump:x:72:
trainee:x:1000:
screen:x:84:
```

Important : Notez que la valeur du GID du groupe root est toujours de 0. Notez que sous RHEL 9 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@redhat9 ~]# cat /etc/gshadow
root:::
bin:::
daemon:::
sys:::
adm:::
tty:::
disk:::
lp:::
mem:::
kmem:::
wheel:::
cdrom:::
mail:::
man:::
dialout:::
floppy:::
games:::
tape:::
video:::
ftp:::
lock:::
audio:::
users:::
nobody:::
utmp:!::
utempter:!::
input:!::
```

```
kvm:!::
render:!::
systemd-journal:!::
systemd-coredump:!::
dbus:!::
polkitd:!::
printadmin:!::
ssh_keys:!::
avahi:!::
tss:!::clevis
colord:!::
clevis:!::
rtkit:!::
sssd:!::
geoclue:!::
sgx:!*::
libstoragemgmt:!*::
systemd-oom:!*::
setroubleshoot:!::
brlapi:!::
pipewire:!::
flatpak:!::
gdm:!::
cockpit-ws:!::
cockpit-wsinstance:!::
gnome-initial-setup:!::
sshd:!::
chrony:!::
slocate:!::
dnsmasq:!::
tcpdump:!::
trainee:!::
screen:!::
```

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@redhat9 ~]# grpck -r
[root@redhat9 ~]#
```

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@redhat9 ~]# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:65534:65534:Kernel Overflow User:/:/sbin/nologin
systemd-coredump:x:999:997:systemd Core Dumper:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
polkitd:x:998:996:User for polkitd:/:/sbin/nologin
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin
tss:x:59:59:Account used for TPM access:/dev/null:/sbin/nologin
colord:x:997:993:User for colord:/var/lib/colord:/sbin/nologin
clevis:x:996:992:Clevis Decryption Framework unprivileged user:/var/cache/clevis:/usr/sbin/nologin
rtkit:x:172:172:RealtimeKit:/proc:/sbin/nologin
sssd:x:995:991:User for sssd:/:/sbin/nologin
geoclue:x:994:990:User for geoclue:/var/lib/geoclue:/sbin/nologin
libstoragemgmt:x:988:988:daemon account for libstoragemgmt:/:/usr/sbin/nologin
```

```
systemd-oom:x:987:987:systemd Userspace OOM Killer:/:/usr/sbin/nologin
setroubleshoot:x:986:986:SELinux troubleshoot server:/var/lib/setroubleshoot:/sbin/nologin
pipewire:x:985:984:PipeWire System Daemon:/var/run/pipewire:/sbin/nologin
flatpak:x:984:983:User for flatpak system helper:/:/sbin/nologin
gdm:x:42:42::/var/lib/gdm:/sbin/nologin
cockpit-ws:x:983:982:User for cockpit web service:/nonexisting:/sbin/nologin
cockpit-wsinstance:x:982:981:User for cockpit-ws instances:/nonexisting:/sbin/nologin
gnome-initial-setup:x:981:980::/run/gnome-initial-setup/:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/usr/share/empty.sshd:/sbin/nologin
chrony:x:980:979:chrony system user:/var/lib/chrony:/sbin/nologin
dnsmasq:x:979:978:Dnsmasq DHCP and DNS server:/var/lib/dnsmasq:/sbin/nologin
tcpdump:x:72:72::/:/sbin/nologin
trainee:x:1000:1000:trainee:/home/trainee:/bin/bash
```

Important : Notez que la valeur de l'UID de root est toujours de 0. Notez que sous RHEL 9, 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@redhat9 ~]# cat /etc/shadow
root:$6$AbCPA3HFsB/NDBkA$q2T8XLo83bPWCid/WHo.i0m9dgjdfQWiZAkqD/ajOjZ8gHdKFYX5y8kuTvIYY/qMjmu9beCk3BZV8ewL/Q15D1::
```

```
0:99999:7:::
bin:*:19347:0:99999:7:::
daemon:*:19347:0:99999:7:::
adm:*:19347:0:99999:7:::
lp:*:19347:0:99999:7:::
sync:*:19347:0:99999:7:::
shutdown:*:19347:0:99999:7:::
halt:*:19347:0:99999:7:::
mail:*:19347:0:99999:7:::
operator:*:19347:0:99999:7:::
games:*:19347:0:99999:7:::
ftp:*:19347:0:99999:7:::
nobody:*:19347:0:99999:7:::
systemd-coredump:!!:19649::::::
dbus:!!:19649::::::
polkitd:!!:19649::::::
avahi:!!:19649::::::
tss:!!:19649::::::
colord:!!:19649::::::
clevis:!!:19649::::::
rtkit:!!:19649::::::
sssd:!!:19649::::::
geoclue:!!:19649::::::
libstoragemgmt:!*:19649::::::
systemd-oom:!*:19649::::::
setroubleshoot:!!:19649::::::
pipewire:!!:19649::::::
flatpak:!!:19649::::::
gdm:!!:19649::::::
cockpit-ws:!!:19649::::::
cockpit-wsinstance:!!:19649::::::
gnome-initial-setup:!!:19649::::::
sshd:!!:19649::::::
chrony:!!:19649::::::
```

```
dnsmasq:!!:19649::::::
tcpdump:!!:19649::::::
trainee:$6$RTR0r5su3SinU2DK$dt/TI6LByO3SKM04nopxI33O7.eE62rPQ0DlO2HRH2PUtPM4c1pvh3koznv6nE6Z0oCoM0Fq7IUdt8cbjXUMh
0::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@redhat9 ~]# pwck -r
[root@redhat9 ~]#
```

> **Important** : Les erreurs éventuelles concernant es répertoires de connexion de certains comptes systèmes ne sont pas importantes. Elles sont dues au fait que les répertoires ne sont pas créés par le système lors de la création des comptes. 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.

Options de la commande

```
[root@redhat9 ~]# 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_DI        directory prefix
```

```
  -U, --users USERS             list of user members of this group
```

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.

Options de la commande

```
[root@redhat9 ~]# 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.

Options de la commande

```
[root@redhat9 ~]# groupmod --help
Usage: groupmod [options] GROUP

Options:
  -a, --append                  append the users mentioned by -U option to the group
                                without removing existing user members
  -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
  -U, --users USERS             list of user members of this group
```

newgrp

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

Options de la commande

```
[root@redhat9 ~]# newgrp --help
Usage: newgrp [-] [group]
```

gpasswd

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

Options de la commande

```
[root@redhat9 ~]# 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, --delete-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

Code Retour	Description
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

Options de la commande

```
[root@redhat9 ~]# useradd --help
Usage: useradd [options] LOGIN
       useradd -D
       useradd -D [options]

Options:
      --badname                 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.

Options de la commande

```
[root@redhat9 ~]# userdel --help
Usage: userdel [options] LOGIN

Options:
  -f, --force                   force some actions that would fail otherwise
                                e.g. removal of user still logged in
                                or files, even if not owned by the 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.

Options de la commande

```
[root@redhat9 ~]# usermod --help
Usage: usermod [options] LOGIN
```

```
Options:
  -b, --badname                 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.

Options de la commande

```
[root@redhat9 ~]# passwd --help
Usage: passwd [OPTION...] <accountName>
  -k, --keep-tokens       keep non-expired authentication tokens
  -d, --delete            delete the password for the named account (root only); also removes password lock if
any
  -l, --lock              lock the password for the named account (root only)
  -u, --unlock            unlock the password for the named account (root only)
  -e, --expire            expire the password for the named account (root only)
  -f, --force             force operation
  -x, --maximum=DAYS      maximum password lifetime (root only)
  -n, --minimum=DAYS      minimum password lifetime (root only)
  -w, --warning=DAYS      number of days warning users receives before password expiration (root only)
  -i, --inactive=DAYS     number of days after password expiration when an account becomes disabled (root only)
  -S, --status            report password status on the named account (root only)
      --stdin             read new tokens from stdin (root only)

Help options:
  -?, --help              Show this help message
      --usage             Display brief usage message
```

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.

Options de la commande

```
[root@redhat9 ~]# 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@redhat9 ~]# cat /etc/default/useradd
```

```
# useradd defaults file
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel
CREATE_MAIL_SPOOL=yes
```

Dans ce fichier, nous trouvons les directives suivantes :

- **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,
- **SHELL** - renseigne le shell de l'utilisateur,
- **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.

Cette même information peut être visualisée en exécutant la commande **useradd** avec l'option **-D** :

```
[root@redhat9 ~]# useradd -D
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel
CREATE_MAIL_SPOOL=yes
```

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

```
[root@redhat9 ~]# ls -la /etc/skel
total 24
drwxr-xr-x.   3 root root   78 Sep 25 11:52 .
drwxr-xr-x. 134 root root 8192 Sep 26 14:57 ..
-rw-r--r--.   1 root root   18 Feb 15  2024 .bash_logout
-rw-r--r--.   1 root root  141 Feb 15  2024 .bash_profile
-rw-r--r--.   1 root root  492 Feb 15  2024 .bashrc
drwxr-xr-x.   4 root root   39 Oct 19  2023 .mozilla
```

Important : Notez que sous RHEL 9 le fichier **.bash_profile** remplace le fichier **.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@redhat9 ~]# id trainee
uid=1000(trainee) gid=1000(trainee) groups=1000(trainee)
```

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

```
[root@redhat9 ~]# groups trainee
trainee : trainee
```

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(8)
#
UID_MIN                  1000
UID_MAX                 60000
```

```
# System accounts
SYS_UID_MIN               201
SYS_UID_MAX               999
# Extra per user uids
SUB_UID_MIN                100000
SUB_UID_MAX             600100000
SUB_UID_COUNT               65536

#
# Min/max values for automatic gid selection in groupadd(8)
#
GID_MIN                  1000
GID_MAX                 60000
# System accounts
SYS_GID_MIN               201
SYS_GID_MAX               999
# Extra per user group ids
SUB_GID_MIN                100000
SUB_GID_MAX             600100000
SUB_GID_COUNT               65536
...
```

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@redhat9 ~]# 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@redhat9 ~]# useradd -g groupe2 fenestros2; useradd -g 1807 fenestros3; useradd -g groupe1 fenestros1
```

```
[root@redhat9 ~]# usermod -G groupe1,groupe3 fenestros2

[root@redhat9 ~]# usermod -c "tux1" fenestros1
```

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

```
[root@redhat9 ~]# tail /etc/passwd
gnome-initial-setup:x:981:980::/run/gnome-initial-setup/:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/usr/share/empty.sshd:/sbin/nologin
chrony:x:980:979:chrony system user:/var/lib/chrony:/sbin/nologin
dnsmasq:x:979:978:Dnsmasq DHCP and DNS server:/var/lib/dnsmasq:/sbin/nologin
tcpdump:x:72:72::/:/sbin/nologin
trainee:x:1000:1000:trainee:/home/trainee:/bin/bash
fenestros2:x:1001:1002::/home/fenestros2:/bin/bash
fenestros3:x:1002:1807::/home/fenestros3:/bin/bash
fenestros1:x:1003:1001:tux1:/home/fenestros1:/bin/bash
```

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

```
[root@redhat9 ~]# tail /etc/group
chrony:x:979:
slocate:x:21:
dnsmasq:x:978:
tcpdump:x:72:
trainee:x:1000:
screen:x:84:
groupe1:x:1001:fenestros2
groupe2:x:1002:
groupe3:x:1807:fenestros2
```

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

```
[root@redhat9 ~]# 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 le fichier **/etc/gshadow** :

```
[root@redhat9 ~]# tail /etc/gshadow
chrony:!::
slocate:!::
dnsmasq:!::
tcpdump:!::
trainee:!::
screen:!::
groupe1:!::fenestros2
groupe2:!::
groupe3:$6$EjmZ4ucT6i/QPvkm$NvJ6r8ytCgdzDfZBSVCXdxUTJJL7RE/gjTsOYiV3UjKuoZp67OoxsERcBCAB71WXF4JcYLRjGGZxTxRg5kgiB
.::fenestros2
```

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

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

```
[root@redhat9 ~]# gpasswd -A fenestros1 groupe3
```

Consultez le fichier **/etc/gshadow** de nouveau :

```
[root@redhat9 ~]# tail /etc/gshadow
```

```
chrony:!::
slocate:!::
dnsmasq:!::
tcpdump:!::
trainee:!::
screen:!::
apache:!::
groupe1:!::fenestros2
groupe2:!::
groupe3:$6$EjmZ4ucT6i/QPvkm$NvJ6r8ytCgdzDfZBSVCXdxUTJJL7RE/gjTsOYiV3UjKuoZp67OoxsERcBCAB71WXF4JcYLRjGGZxTxRg5kgiB
.: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@redhat9 ~]# 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@redhat9 ~]# userdel fenestros3
```

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

```
[root@redhat9 ~]# 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 le système.

Saisissez la commande suivante sous RHEL 9 pour vérifier :

```
[root@redhat9 ~]# ls -ld /home/fenestros3
drwx------. 3 1002 1807 78 Sep 27 13:59 /home/fenestros3
```

Pour supprimer les fichiers de cet utilisateur, il convient de saisir la commande suivante :

```
[root@redhat9 ~]# find /home -user 1002 -exec rm -rf {} \;
find: ‘/home/fenestros3’: No such file or directory

[root@redhat9 ~]# ls -ld /home/fenestros3
ls: cannot access '/home/fenestros3': No such file or directory
```

> **Important** : La commande **find** est lancée d'une manière itérative. L'erreur est normale car quand la commande **find** ne trouve plus de fichiers à supprimer, elle s’arrête avec un code retour de 2.

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

```
[root@redhat9 ~]# passwd fenestros1
```

```
Changing password for user fenestros1.
New password: fenestros1
BAD PASSWORD: The password contains the user name in some form
Retype new password: fenestros1
passwd: all authentication tokens updated successfully.

[root@redhat9 ~]# passwd fenestros2
Changing password for user fenestros2.
New password: fenestros2
BAD PASSWORD: The password contains the user name in some form
Retype new password: fenestros2
passwd: all authentication tokens updated successfully.
```

Important : Notez que les règles gouvernant l'utilisation des mots de passe ne sont pas appliqués aux utilisateurs 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

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@redhat9 ~]# ls -l /etc/pam.d
total 128
-rw-r--r--. 1 root root 272 Apr  4  2022 atd
-rw-r--r--. 1 root root 192 Feb  8  2024 chfn
-rw-r--r--. 1 root root 192 Feb  8  2024 chsh
```

```
-rw-r--r--. 1 root root 910 Apr  2 05:27 cockpit
-rw-r--r--. 1 root root 232 Feb 12  2024 config-util
-rw-r--r--. 1 root root 322 Feb 15  2019 crond
-r--r--r--. 1 root root 146 Jun 19 11:00 cups
lrwxrwxrwx. 1 root root  32 Sep 25 12:09 fingerprint-auth -> /etc/authselect/fingerprint-auth
-rw-r--r--. 1 root root 622 Jul 23  2021 gdm-autologin
-rw-r--r--. 1 root root 561 Jul 23  2021 gdm-fingerprint
-rw-r--r--. 1 root root 307 Jul 23  2021 gdm-launch-environment
-rw-r--r--. 1 root root 787 Jul 23  2021 gdm-password
-rw-r--r--. 1 root root 800 Jul 23  2021 gdm-pin
-rw-r--r--. 1 root root 553 Jul 23  2021 gdm-smartcard
-rw-r--r--. 1 root root 676 Feb  8  2024 login
-rw-r--r--. 1 root root 154 Feb 12  2024 other
-rw-r--r--. 1 root root 168 Aug 10  2021 passwd
lrwxrwxrwx. 1 root root  29 Sep 25 12:09 password-auth -> /etc/authselect/password-auth
-rw-r--r--. 1 root root 155 Dec  5  2022 polkit-1
lrwxrwxrwx. 1 root root  25 Sep 25 12:09 postlogin -> /etc/authselect/postlogin
-rw-r--r--. 1 root root 640 Feb  8  2024 remote
-rw-r--r--. 1 root root 143 Feb  8  2024 runuser
-rw-r--r--. 1 root root 138 Feb  8  2024 runuser-l
-rw-r--r--. 1 root root  36 Jan  4  2022 screen
lrwxrwxrwx. 1 root root  30 Sep 25 12:09 smartcard-auth -> /etc/authselect/smartcard-auth
-rw-r--r--. 1 root root 727 Jul  3 11:56 sshd
-rw-r--r--. 1 root root 214 Jan 12  2024 sssd-shadowutils
-rw-r--r--. 1 root root 566 Feb  8  2024 su
-rw-r--r--. 1 root root  97 Jan 18  2024 subscription-manager
-rw-r--r--. 1 root root 154 Jan 24  2024 sudo
-rw-r--r--. 1 root root 178 Jan 24  2024 sudo-i
-rw-r--r--. 1 root root 137 Feb  8  2024 su-l
lrwxrwxrwx. 1 root root  27 Sep 25 12:09 system-auth -> /etc/authselect/system-auth
-rw-r--r--. 1 root root 414 Jul 18 13:00 systemd-user
-rw-r--r--. 1 root root  84 Jun 21  2023 vlock
-rw-r--r--. 1 root root 159 Dec  4  2023 vmtoolsd
-rw-r--r--. 1 root root 163 Jan 19  2024 xserver
```

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 **/lib64/security** :

```
[root@redhat9 ~]# ls -l /lib64/security
total 1724
-rwxr-xr-x. 1 root root  19560 Feb 12  2024 pam_access.so
-rwxr-xr-x. 1 root root  15136 Jul 12  2023 pam_cap.so
-rwxr-xr-x. 1 root root  15288 Feb 12  2024 pam_chroot.so
-rwxr-xr-x. 1 root root  15008 Apr  2 05:45 pam_cockpit_cert.so
-rwxr-xr-x. 1 root root  32112 Feb 12  2024 pam_console.so
-rwxr-xr-x. 1 root root  15344 Feb 12  2024 pam_debug.so
-rwxr-xr-x. 1 root root  15040 Feb 12  2024 pam_deny.so
-rwxr-xr-x. 1 root root  15368 Feb 12  2024 pam_echo.so
-rwxr-xr-x. 1 root root  19568 Feb 12  2024 pam_env.so
-rwxr-xr-x. 1 root root  23536 Feb 12  2024 pam_exec.so
-rwxr-xr-x. 1 root root  15304 Feb 12  2024 pam_faildelay.so
-rwxr-xr-x. 1 root root  23632 Feb 12  2024 pam_faillock.so
drwxr-xr-x. 2 root root     24 Sep 25 11:56 pam_filter
-rwxr-xr-x. 1 root root  19472 Feb 12  2024 pam_filter.so
-rwxr-xr-x. 1 root root  32600 Aug 26  2021 pam_fprintd.so
-rwxr-xr-x. 1 root root  15296 Feb 12  2024 pam_ftp.so
-rwxr-xr-x. 1 root root  15328 Jan 18  2024 pam_gdm.so
-rwxr-xr-x. 1 root root  31960 Jul  9 15:38 pam_gnome_keyring.so
-rwxr-xr-x. 1 root root  19456 Feb 12  2024 pam_group.so
-rwxr-xr-x. 1 root root  15336 Feb 12  2024 pam_issue.so
-rwxr-xr-x. 1 root root  15464 Feb 12  2024 pam_keyinit.so
-rwxr-xr-x. 1 root root  19632 Feb 12  2024 pam_lastlog.so
-rwxr-xr-x. 1 root root  27648 Feb 12  2024 pam_limits.so
-rwxr-xr-x. 1 root root  15352 Feb 12  2024 pam_listfile.so
-rwxr-xr-x. 1 root root  15344 Feb 12  2024 pam_localuser.so
-rwxr-xr-x. 1 root root  15360 Feb 12  2024 pam_loginuid.so
-rwxr-xr-x. 1 root root  19416 Feb 12  2024 pam_mail.so
-rwxr-xr-x. 1 root root  15296 Feb 12  2024 pam_mkhomedir.so
-rwxr-xr-x. 1 root root  15376 Feb 12  2024 pam_motd.so
```

```
-rwxr-xr-x. 1 root root  44264 Feb 12  2024 pam_namespace.so
-rwxr-xr-x. 1 root root  15344 Feb 12  2024 pam_nologin.so
-rwxr-xr-x. 1 root root  15328 Feb 12  2024 pam_permit.so
-rwxr-xr-x. 1 root root  15296 Feb 12  2024 pam_postgresok.so
-rwxr-xr-x. 1 root root  27624 Feb 12  2024 pam_pwhistory.so
-rwxr-xr-x. 1 root root  15840 Aug 10  2021 pam_pwquality.so
-rwxr-xr-x. 1 root root  15296 Feb 12  2024 pam_rhosts.so
-rwxr-xr-x. 1 root root  15352 Feb 12  2024 pam_rootok.so
-rwxr-xr-x. 1 root root  15360 Feb 12  2024 pam_securetty.so
lrwxrwxrwx. 1 root root     15 Feb 12  2024 pam_selinux_permit.so -> pam_sepermit.so
-rwxr-xr-x. 1 root root  27720 Feb 12  2024 pam_selinux.so
-rwxr-xr-x. 1 root root  19472 Feb 12  2024 pam_sepermit.so
-rwxr-xr-x. 1 root root  19424 Feb 12  2024 pam_setquota.so
-rwxr-xr-x. 1 root root  15328 Feb 12  2024 pam_shells.so
-rwxr-xr-x. 1 root root  27928 Apr  2 05:45 pam_ssh_add.so
-rwxr-xr-x. 1 root root  36216 May 17 03:59 pam_sss_gss.so
-rwxr-xr-x. 1 root root  69264 May 17 03:59 pam_sss.so
-rwxr-xr-x. 1 root root  19528 Feb 12  2024 pam_stress.so
-rwxr-xr-x. 1 root root  19520 Feb 12  2024 pam_succeed_if.so
-rwxr-xr-x. 1 root root 514384 Jul 18 13:01 pam_systemd.so
-rwxr-xr-x. 1 root root  19456 Feb 12  2024 pam_time.so
-rwxr-xr-x. 1 root root  27696 Feb 12  2024 pam_timestamp.so
-rwxr-xr-x. 1 root root  15344 Feb 12  2024 pam_tty_audit.so
-rwxr-xr-x. 1 root root  15296 Feb 12  2024 pam_umask.so
lrwxrwxrwx. 1 root root     11 Feb 12  2024 pam_unix_acct.so -> pam_unix.so
lrwxrwxrwx. 1 root root     11 Feb 12  2024 pam_unix_auth.so -> pam_unix.so
lrwxrwxrwx. 1 root root     11 Feb 12  2024 pam_unix_passwd.so -> pam_unix.so
lrwxrwxrwx. 1 root root     11 Feb 12  2024 pam_unix_session.so -> pam_unix.so
-rwxr-xr-x. 1 root root  56824 Feb 12  2024 pam_unix.so
-rwxr-xr-x. 1 root root  19472 Feb 12  2024 pam_userdb.so
-rwxr-xr-x. 1 root root  15384 Feb 12  2024 pam_usertype.so
-rwxr-xr-x. 1 root root  15344 Feb 12  2024 pam_warn.so
-rwxr-xr-x. 1 root root  15352 Feb 12  2024 pam_wheel.so
-rwxr-xr-x. 1 root root  27632 Feb 12  2024 pam_xauth.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/**.
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_pwquality.so	Ce module est utilisé pour vérifier la qualité du mot de passe d'un utilisateur
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@redhat9 ~]# cat /etc/pam.d/login
#%PAM-1.0
auth       substack     system-auth
auth       include      postlogin
account    required     pam_nologin.so
account    include      system-auth
password   include      system-auth
# pam_selinux.so close should be the first session rule
session    required     pam_selinux.so close
session    required     pam_loginuid.so
# pam_selinux.so open should only be followed by sessions to be executed in the user context
session    required     pam_selinux.so open
session    required     pam_namespace.so
session    optional     pam_keyinit.so force revoke
session    include      system-auth
session    include      postlogin
```

```
-session   optional     pam_ck_connector.so
```

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/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@redhat9 ~]# cat /etc/pam.d/other
#%PAM-1.0
auth     required       pam_deny.so
account  required       pam_deny.so
password required       pam_deny.so
session  required       pam_deny.so
```

Utiliser des Mots de Passe Complexes

Configuration

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

```
[root@redhat9 ~]# ls /etc/security
access.conf  console.apps      console.perms    faillock.conf  limits.conf  namespace.conf  namespace.init
pam_env.conf    pwquality.conf    sepermit.conf
chroot.conf  console.handlers  console.perms.d  group.conf     limits.d     namespace.d     opasswd
pwhistory.conf  pwquality.conf.d  time.conf
```

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

Fichier/*Répertoire*	**Description**
access.conf	Utilisé par le module pam_access.so
console.apps	Utilisés par le module pam_console.so
console.perms	Utilisé par le module pam_console.so
console.perms.d	Utilisé par le module pam_console.so
group.conf	Utilisés par le module pam_group.so
limits.conf	Utilisé par le module pam_limits.so

Fichier/*Répertoire*	**Description**
pam_env.conf	Utilisé par le module pam_env.so
pwquality.conf	Utilisé par le module pam_pwquality.so
time.conf	Utilisé par le module pam_time.so

La complexité des mots de passe est gérée par le module pam_pwquality.so. Afin de mettre en place une politique de mots de passe complexe, il convient de modifier le fichier **/etc/security/pwquality.conf** :

```
[root@redhat9 ~]# vi /etc/security/pwquality.conf

[root@redhat9 ~]# 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
```

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@redhat9 ~]# pwd
/root
```

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

```
[root@redhat9 ~]# su fenestros2
```

Contrôlez votre répertoire courant de travail :

```
[fenestros2@redhat9 root]$ 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 donc, entre autre, le répertoire personnel de l'utilisateur ainsi que la valeur de la variable système **PATH**.

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

```
[fenestros2@redhat9 root]$ exit
exit
```

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

```
[root@redhat9 ~]# su - fenestros2
```

Contrôlez votre répertoire courant de travail :

```
[fenestros2@redhat9 ~]$ pwd
/home/fenestros2
```

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

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

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 :

```
[fenestros2@redhat9 ~]$ exit
logout

[root@redhat9 ~]# cat /etc/sudoers
## Sudoers allows particular users to run various commands as
## the root user, without needing the root password.
##
## Examples are provided at the bottom of the file for collections
## of related commands, which can then be delegated out to particular
## users or groups.
##
## This file must be edited with the 'visudo' command.

## Host Aliases
## Groups of machines. You may prefer to use hostnames (perhaps using
## wildcards for entire domains) or IP addresses instead.
# Host_Alias     FILESERVERS = fs1, fs2
```

```
# Host_Alias     MAILSERVERS = smtp, smtp2

## User Aliases
## These aren't often necessary, as you can use regular groups
## (ie, from files, LDAP, NIS, etc) in this file - just use %groupname
## rather than USERALIAS
# User_Alias ADMINS = jsmith, mikem



## Command Aliases
## These are groups of related commands...

## Networking
# Cmnd_Alias NETWORKING = /sbin/route, /sbin/ifconfig, /bin/ping, /sbin/dhclient, /usr/bin/net, /sbin/iptables,
/usr/bin/rfcomm, /usr/bin/wvdial, /sbin/iwconfig, /sbin/mii-tool

## Installation and management of software
# Cmnd_Alias SOFTWARE = /bin/rpm, /usr/bin/up2date, /usr/bin/yum

## Services
# Cmnd_Alias SERVICES = /sbin/service, /sbin/chkconfig, /usr/bin/systemctl start, /usr/bin/systemctl stop,
/usr/bin/systemctl reload, /usr/bin/systemctl restart, /usr/bin/systemctl status, /usr/bin/systemctl enable,
/usr/bin/systemctl disable

## Updating the locate database
# Cmnd_Alias LOCATE = /usr/bin/updatedb

## Storage
# Cmnd_Alias STORAGE = /sbin/fdisk, /sbin/sfdisk, /sbin/parted, /sbin/partprobe, /bin/mount, /bin/umount

## Delegating permissions
# Cmnd_Alias DELEGATING = /usr/sbin/visudo, /bin/chown, /bin/chmod, /bin/chgrp

## Processes
```

```
# Cmnd_Alias PROCESSES = /bin/nice, /bin/kill, /usr/bin/kill, /usr/bin/killall

## Drivers
# Cmnd_Alias DRIVERS = /sbin/modprobe

# Defaults specification

#
# Refuse to run if unable to disable echo on the tty.
#
Defaults   !visiblepw

#
# Preserving HOME has security implications since many programs
# use it when searching for configuration files. Note that HOME
# is already set when the the env_reset option is enabled, so
# this option is only effective for configurations where either
# env_reset is disabled or HOME is present in the env_keep list.
#
Defaults    always_set_home
Defaults    match_group_by_gid

# Prior to version 1.8.15, groups listed in sudoers that were not
# found in the system group database were passed to the group
# plugin, if any. Starting with 1.8.15, only groups of the form
# %:group are resolved via the group plugin by default.
# We enable always_query_group_plugin to restore old behavior.
# Disable this option for new behavior.
Defaults    always_query_group_plugin

Defaults    env_reset
Defaults    env_keep =  "COLORS DISPLAY HOSTNAME HISTSIZE KDEDIR LS_COLORS"
Defaults    env_keep += "MAIL PS1 PS2 QTDIR USERNAME LANG LC_ADDRESS LC_CTYPE"
Defaults    env_keep += "LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT LC_MESSAGES"
```

```
Defaults    env_keep += "LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER LC_TELEPHONE"
Defaults    env_keep += "LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET XAUTHORITY"

#
# Adding HOME to env_keep may enable a user to run unrestricted
# commands via sudo.
#
# Defaults   env_keep += "HOME"

Defaults    secure_path = /sbin:/bin:/usr/sbin:/usr/bin

## Next comes the main part: which users can run what software on
## which machines (the sudoers file can be shared between multiple
## systems).
## Syntax:
##
##      user    MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)       ALL

## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS

## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)       ALL

## Same thing without a password
# %wheel        ALL=(ALL)       NOPASSWD: ALL

## Allows members of the users group to mount and unmount the
```

```
## cdrom as root
# %users  ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom

## Allows members of the users group to shutdown this system
# %users  localhost=/sbin/shutdown -h now

## Read drop-in files from /etc/sudoers.d (the # here does not mean a comment)
#includedir /etc/sudoers.d
```

Important : Notez la présence de la ligne **%wheel ALL=(ALL) ALL**. Cette ligne possède le format **Qui Où = (En tant que qui) Quoi**. La ligne implique donc que les membres du groupe **wheel** peuvent exécuter à partir de n'importe quel hôte et en tant que n'importe quel rôle, 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**.
