Version: **2024.01**

Last update: 2024/11/26 13:43

# RH12406 - User Management

## Contents

        - Using Complex Passwords
        - Configuration
    - su and su -
    - sudo

# Presentation

> **To do**: In order to put the examples in this course into practice, you need to connect to your system as root using the command **su -** and the password **fenestros**.

Good user management involves a good group strategy. In fact, each user is assigned to a **primary** group but can also be a member of upto 15 secondary groups.

As in other operating systems, under Linux it is preferable to give access rights to groups and not to individual users.

The databases used to store user and group information are stipulated in the **/etc/nsswitch.conf** file. In our case, the passwd, shadow and group entries indicate the **files** keyword. This indicates the use of the following files as a database:

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

## /etc/nsswitch.conf

```
[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 always takes 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
```

In this file:

- **sss** indicates the use of the **System Security Services Daemon** (SSSD).
  - SSSD has its origines in **FreeIPA** (Identity, Policy and Audit) and provides Linux/Unix networks with similar functionalities as those provided by Microsoft Active Directory Domain Services to Windows™ networks,
  - For more information, consult **this page**.
- **files** indicates the use of local text files in **/etc**,
- **systemd** indicates the use of the **nss-systemd** plugin which uses the **Name Service Switch** (NSS) function found in the **GNU C Library** (glibc).

## Database query

The **getent** command is used to query databases. It takes the following form:

```
getent database key
```

For example, to search for a user in the user database, use the following command:

```
[root@redhat9 ~]# getent passwd trainee
trainee:x:1000:1000:trainee:/home/trainee:/bin/bash
```

To find out which users belong to which groups, use the following command:

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

Using the getent command without specifying a key prints the contents of the database to STDOUT:

```
[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
```

## The /etc/group and /etc/gshadow Files

To list the existing groups on the system, enter the following command:

```
[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**: Note that the root group GID value is always 0. Note that under RHEL 9, normal user GIDs start at **1000** and system account GIDs are included between 1 and 99 and between 201 and 999.

In this file, each line consists of 4 fields:

- The **unique** name of the group,
- The group password. A value of **x** in this field indicates that the system uses the **/etc/gshadow** file to store passwords. A value of **!** indicates that

the group has no password and that access to the group via the **newgrp** command is not possible,
- The GID. A unique value used to determine access rights to files and directories,
- The list of members who have the group as their **secondary** group.

To view the **/etc/gshadow** file, enter the following command:

```
[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:!::
```

Each line consists of 4 fields:

- The unique group name from the **/etc/group** file,
- The encrypted group password. If this is empty, only the users who are members of the group can use the newgrp command. If the field contains a !, an x or a * it indicates that noone can use the newgrp command,
- The group administrator if one exists,
- The list of users who have the group as a secondary group.

To check the **/etc/group** and **/etc/gshadow** files for possible errors, enter the following command:

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

> **Important**: The **-r** option allows error checking without modifying anything.

In the event that it is necessary to regenerate one of the two files, one of the following two commands should be used:

- **grpconv**
  - regenerates the **/etc/gshadow** file from the **/etc/group** file and possibly the existing **/etc/gshadow** file.
- **grpunconv**
  - regenerates the **/etc/group** file from the **/etc/gshadow** file and possibly from the existing **/etc/group** file, then deletes the **/etc/gshadow** file.

## The /etc/passwd and /etc/shadow Files

> **Important**: Note that the most liberal rule regarding usernames under Linux limits the length to 32 characters and allows the use of upper case, lower case, numbers (except at the beginning of the name) as well as most punctuation characters. However, some utilities, such as **useradd** prohibit the use of upper case and punctuation characters but allow the use of **_**, **.** and the **$** character at the end of the name (**WARNING**: in the case of

samba, a user name ending in **$** is considered to be a **machine** account). What's more, some utilities limit the length of the name to **8** characters.

To list existing user accounts on the system, enter the following command:

```
[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**: Note that the root UID value is always 0. Note that under RHEL 9, normal user UIDs start at **1000** and system account UIDs are included between 1 and 99 and between 201 and 999.

Each line consists of 7 fields:

- The user name
- The password. A value of **x** in this field indicates that the system uses the **/etc/shadow** file to store passwords.
- THE UID. A unique value used to determine rights to files and directories.
- GID. A value indicating the user's **primary** group.
- The full name. This optional field is also called **GECOS**.
- The user's home directory
- The user's shell.

To view the **/etc/shadow** file, enter the following command:

```
[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:::
```

Each line consists of 8 fields:

- The user's name. This field is used to make the link with the **/etc/passwd** file,
- The user's **encrypted** password. Encryption is **one-way**. This field can also contain one of the following three values:
  - **!!** - The password has not yet been set and the user cannot log in,
  - * - The user cannot log in,
  - **empty** - No password will be requested for this user,
- The number of days between **01/01/1970** and the last password change,
- The number of days that the password is still valid. A value of **0** in this field indicates that the password never expires,
- The number of days after which the password must be changed,
- The number of days before the forced change date that the user will receive a warning,
- The number of days after the password expires that the account will be deactivated,
- The **number** of the day after **01/01/1970** that the account was deactivated.

To check the **/etc/passwd** and **/etc/shadow** files for errors, enter the following command:

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

> **Important**: Any errors concerning the login directories of certain system accounts are not important. They are due to the fact that the directories are not created by the system when the accounts are created. Once again, the **-r** option allows errors to be checked in without modifying anything.

In the event that it is necessary to regenerate one of the two files, one of the following two commands should be used:

- **pwconv**
  - regenerates the **/etc/shadow** file from the **/etc/passwd** file and possibly the existing **/etc/shadow** file.
- **pwunconv**
  - regenerates the **/etc/passwd** file from the **/etc/shadow** file and possibly from the existing **/etc/passwd** file, then deletes the

**/etc/shadow** file.

# Commands

## Groups

### groupadd

This command is used to create a group.

#### Command Line Switches

```
[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**: It is possible to create several groups with the same GID.

**Important**: Note the **-r** option which allows the creation of a system group.

### groupdel

This command is used to delete a group.

#### Command Line Switches

```
[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

This command is used to modify an existing group.

#### Command Line Switches

```
[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

This command is used to change the group of the user invoking it.

#### Command Line Switches

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

### gpasswd

This command is used to administer groups.

#### Command Line Switches

```
[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.
```

## Users

### useradd

This command is used to add a user.

The return codes for the useradd command are:

| Return code | Description |
|---|---|
| 1 | Unable to update /etc/passwd file |
| 2 | Invalid syntax |

| Return code | Description |
|---|---|
| 3 | Invalid option |
| 4 | The requested UID is already in use |
| 6 | The specified group does not exist |
| 9 | The specified user name already exists |
| 10 | Unable to update the /etc/group file |
| 12 | Unable to create the user's home directory |
| 13 | Unable to create the user's mail spool |

**Command Line Switches**

```
[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, --no-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**: It is possible to create multiple users with the same UID.

**Important**: Note the **-r** option which allows a system account to be created. In this case the useradd command does not create a home directory.

### userdel

This command is used to delete a user.

#### Command Line Switches

```
[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**: Note that when deleting a user, the UID associated with that account can be reused. The maximum number of accounts was **65,536** with the **2.2.x** kernel. With recent kernels, this limit rises to more than 4.2 Billion.

### usermod

This command is used to modify an existing user.

#### Command Line Switches

```
[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**: Note the **-L** option that locks an account.

### passwd

This command is used to create or change a user's password.

#### Command Line Switches

```
[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**: Note the **-l** option, which locks an account by placing the character **!** in front of the encrypted password.

### chage

The chage command changes the number of days between password changes and the date of the last change. This information is used by the system to determine whether a user should change their password.

#### Command Line Switches

```
[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

The **useradd** command is configured by the **/etc/default/useradd** file. To view this file, enter the following command:

```
[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
```

In this file we find the following directives:

- **GROUP** - identifies the user's default primary group when the **-N** option is used with the **useradd** command. Otherwise, the primary group is either the group specified by the **-g** option of the command, or a new group with the same name as the user,
- **HOME** - indicates that the user's home directory will be created in the **home** directory when the account is created if this option has been enabled in the **/etc/login.defs** file,
- **INACTIVE** - indicates the number of days of inactivity after a password has expired before the account is locked. A value of -1 disables this directive,
- **EXPIRE** - with no value, this directive indicates that the user's password never expires,
- **SHELL** - specifies the user's shell,
- **SKEL** - indicates the directory containing the files that will be copied to the user's home directory, if this directory is created when the user is created,
- **CREATE_MAIL_SPOOL** - indicates whether or not an internal mailbox will be created for the user.

This same information can be viewed by executing the **useradd** command with the **-D** option:

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

To view the list of files in **/etc/skel**, enter the following command:

```
[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**: Note that under RHEL 9 the **.bash_profile** file replaces the **.profile** file.

To find out a user's UID, GID and group membership, use the **id** command. Enter the following command:

```
[root@redhat9 ~]# id trainee
uid=1000(trainee) gid=1000(trainee) groups=1000(trainee)
```

If you only want to know a user's groups, you should use the **groups** command. Enter the following command:

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

The minimum UID and GID values used by default when creating a user are stipulated in the **/etc/login.defs** file:

```
...
#
# 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 - Manage Users and Groups

Now create three groups **group1**, **group2** and **group3**. The GID value of group **group3** must be **1807**:

```
[root@redhat9 ~]# groupadd group1; groupadd group2; groupadd -g 1807 group3
```

Now create three users **fenestros1**, **fenestros2** and **fenestros3**. The three users have **group1**, **group2** and **group3** as their primary group respectively. **fenestros2** is also a member of **group1** and **group3**. **fenestros1** has a GECOS of **tux1**:

```
[root@redhat9 ~]# useradd -g group2 fenestros2; useradd -g 1807 fenestros3; useradd -g group1 fenestros1
```

```
[root@redhat9 ~]# usermod -G group1,group3 fenestros2

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

If you check your **/etc/passwd** file, you'll get a result similar to this:

```
[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
```

Looking at your **/etc/group** file, you'll get a result similar to this:

```
[root@redhat9 ~]# tail /etc/group
chrony:x:979:
slocate:x:21:
dnsmasq:x:978:
tcpdump:x:72:
trainee:x:1000:
screen:x:84:
group1:x:1001:fenestros2
group2:x:1002:
group3:x:1807:fenestros2
```

Create the password **fenestros** for **group3**:

```
[root@redhat9 ~]# gpasswd group3
Changing the password for group groupe3
```

```
New Password: fenestros
Re-enter new password: fenestros
```

**Important**: Note that the passwords entered will **not** be visible.

Check the **/etc/gshadow** file:

```
[root@redhat9 ~]# tail /etc/gshadow
chrony:!::
slocate:!::
dnsmasq:!::
tcpdump:!::
trainee:!::
screen:!::
group1:!::fenestros2
group2:!::
group3:$6$EjmZ4ucT6i/QPvkm$NvJ6r8ytCgdzDfZBSVCXdxUTJJL7RE/gjTsOYiV3UjKuoZp67OoxsERcBCAB71WXF4JcYLRjGGZxTxRg5kgiB.
::fenestros2
```

**Important**: Note the presence of the encrypted password for **group3**.

Now name **fenestros1** administrator of **group3**:

```
[root@redhat9 ~]# gpasswd -A fenestros1 group3
```

Check the **/etc/gshadow** file again:

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

```
chrony:!::
slocate:!::
dnsmasq:!::
tcpdump:!::
trainee:!::
screen:!::
apache:!::
group1:!::fenestros2
group2:!::
group3:$6$EjmZ4ucT6i/QPvkm$NvJ6r8ytCgdzDfZBSVCXdxUTJJL7RE/gjTsOYiV3UjKuoZp67OoxsERcBCAB71WXF4JcYLRjGGZxTxRg5kgiB.
:fenestros1:fenestros2
```

> **Important**: The **fenestros1** user can now administer the **group3** group by adding or deleting users.

Now try deleting the **group3** group:

```
[root@redhat9 ~]# groupdel group3
groupdel: cannot remove the primary group of user ‘fenestros3’
```

> **Important**: In effect, you cannot remove a group as long as a user has it as their primary group.

So delete the user **fenestros3**:

```
[root@redhat9 ~]# userdel fenestros3
```

Next try deleting the **group3** group:

```
[root@redhat9 ~]# groupdel group3
```

**Important**: Note that this time the command is executed without error.

Deleting a user **without** the **-r** option implies that the user's home directory remains on the system.

Enter the following command under RHEL 9 to check:

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

To delete the files for this user, enter the following command:

```
[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**: The **find** command is run iteratively. The error is normal because when the **find** command finds no more files to delete, it stops with a return code of 2.

Now create the passwords for **fenestros1** and **fenestros2**. Specify a password identical to the account name:

```
[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**: Note that the rules governing the use of passwords are not applied to users created by root. Also note that passwords entered will **NOT** be visible.

## LAB #2 - Forcing the use of complex passwords with PAM

**PAM** ( *Pluggable Authentication Modules* ) is a modular architecture allowing the system administrator to define an authentication policy for software supporting PAM.

The configuration files can be found in the **/etc/pam.d** directory:

```
[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 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
```

These files have a specific structure and are named after the service or application they control. Their contents call on modules found in the **/lib64/security** directory:

```
[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
```

The most important modules are:

| Module | Description |
|---|---|
| pam_access.so | This module is used to prohibit access to secure services by unauthorised hosts. |
| pam_echo.so | This module presents the contents of the file passed as an argument to any user when they connect. |
| pam_limits.so | This module implements the resource limits detailed in the **/etc/security/limits.conf** file and in the ***.conf** files found in the **/etc/security/limits.d/** directory. |
| pam_listfile.so | This module is used to consult a specific file to check authentications. For example, the ftp service uses this module to consult the **/etc/ftpusers** file, which contains a list of users who are **not** authorised to connect to the ftp server. |
| pam_nologin.so | This module prohibits connections from users other than root if the **/etc/nologin** file is present. |
| pam_pwquality.so | This module is used to check the quality of a user's password. |
| pam_securetty.so | This module prevents root connections from tty devices that are not listed in the **/etc/securetty** file. |
| pam_unix.so | This module is used to check the following information; expire, last_change, max_change, min_change, warn_change. |

Each file in /etc/pam.d contains the PAM rules used during authentication. Open the **login** file:

```
[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
```

The first line of this file is a comment that specifies that the file conforms to the PAM 1.0 specification.

This file, like the others, is then structured as follows:

- One module per line,
- Four fields separated by a space in each rule, the first three of which are mandatory.

The **first field** is the ***module type***. There are four types:

| **Type** | |
|---|---|
| **auth** | Used to authenticate a user or system prerequisites ( for example /etc/nologin ) |
| **account** | Used to check whether the user can authenticate (e.g. account validity). |
| **password** | Used to check whether the user has the rights to update the authentication mechanism. |
| **session** | Used to manage the session after authentication (for example, mount a directory). |

The **second field** is the ***Control-flag***. There are four of them:

| **Control-flag** | **Description** |
|---|---|
| **required** | Successful completion of this module is essential. The failure of a *required* module is not communicated to the application until all modules with a *control-flag* of **required** have been checked. |
| **requisite** | This module must succeed. Failure of a *requisite* module is immediately communicated to the application. |
| **sufficient** | Success is sufficient to authorise authentication. If no previous *required* test has failed, verification stops. If a previous *required* test failed, the *sufficient* test is ignored. The failure of a *sufficient* test has no consequence if all the *required* tests succeed. |
| **optional** | The success or failure of this module is irrelevant, **unless** it is the only module to be executed. |
| **include** | This control flag is used to include all lines of the same *module type* in the file specified as an argument. |

The **third field** specifies the ***module*** associated with the rule. Without an absolute path, the file is assumed to be in the **/lib/security** directory. To include a module outside this directory, its absolute path must be specified.

The **fourth field** may contain the **arguments**.

PAM also offers a solution for all applications that do not have their own PAM configuration files. This solution takes the form of the **/etc/pam.d/other** file:

```
[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
```

### Using Complex Passwords

### Configuration

Some PAM modules can be configured using files in the **/etc/security** directory:

```
[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
```

Among the files listed are those that can be used to configure the following modules:

| **File*/Directory*** | **Description** |
|---|---|
| **access.conf** | Used by the pam_access.so module |
| ***console.apps*** | Used by the pam_console.so module |
| **console.perms** | Used by the pam_console.so module |
| ***console.perms.d*** | Used by the pam_console.so module |
| **group.conf** | Used by the pam_group.so module |
| **limits.conf** | Used by module pam_limits.so |
| **pam_env.conf** | Used by the pam_env.so module |
| **pwquality.conf** | Used by pam_pwquality.so module |
| **time.conf** | Used by the pam_time.so module |

Password complexity is managed by the **pam_pwquality.so** module. In order to set up a complex password policy, the **/etc/security/pwquality.conf**

file needs to be modified:

```
[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 and su -

You will now become **fenestros2**, first without the **fenestros2** environment and then with the **fenestros2** environment.

Check your current working directory:

```
[root@redhat9 ~]# pwd
/root
```

To become **fenestros2 without** its environment, enter the following command:

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

Check your current working directory:

```
[fenestros2@redhat9 root]$ pwd
/root
```

You will notice that you are still in the **/root** directory. This indicates that you have kept the **root** environment.

> **Important**: A user's environment therefore includes, among other things, the user's home directory and the value of the **PATH** system variable.

Enter the following command to become **root** again:

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

Enter the following command to become **fenestros2** again:

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

Check your current working directory:

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

You will notice that you are now in the **/home/fenestros2** directory. This indicates that you have the **fenestros2** environment.

> **Important**: Note that **root** can become any user **without** needing to know their password.

# sudo

The **sudo** command allows an authorised user to execute a command as **root** or as another user. When the command is executed, the effective and real UID and GID are those of the target user's identity. Using the **sudo** command is a simple way of delegating administrative tasks to other users without disclosing the **root** password and without setting a SUID bit on the executable. The **sudo** command is configured using the **/etc/sudoers** file.

Enter the following command:

```
[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**: Note the presence of the line **%wheel ALL=(ALL) ALL**. This line has the format **Who Where = (As who) What**. The line therefore implies that members of the **wheel** group can execute all system commands from any host and as any role. In this file, a group is referenced by a **%**. A name without this character is user. To edit the **/etc/sudoers** file, it is **necessary** to use the **visudo** command.

---

Copyright © 2024 Hugh Norris.