

La gestion des utilisateurs

Objectifs : Dans ce chapitre, les futurs administrateurs Linux vont apprendre comment :

- ✓ ajouter, supprimer ou modifier un **groupe** ;
- ✓ ajouter, supprimer ou modifier un **utilisateur** ;
- ✓ connaître la syntaxe des fichiers associés à la gestion des groupes et des utilisateurs ;
- ✓ changer le *propriétaire* ou le *groupe propriétaire* d'un fichier ;
- ✓ sécuriser les comptes utilisateurs ;
- ✓ changer d'identité.

utilisateurs

Connaissances :

Complexité :

Temps de lecture : 30 minutes

Généralités

Chaque utilisateur est membre d'au moins un groupe : **c'est son groupe principal**.

Plusieurs utilisateurs peuvent faire partie d'un même groupe.

Les utilisateurs peuvent appartenir à d'autres groupes. Ces utilisateurs sont *invités* dans ces **groupes secondaires**.

Note

Chaque utilisateur a un groupe primaire et peut être invité dans un ou plusieurs groupes secondaires.

Les groupes et utilisateurs se gèrent par leur identifiant numérique unique `GID` et `UID`.

Les fichiers de déclaration des comptes et groupes se trouvent dans `/etc`. `UID` : *User IDentifier*. Identifiant unique d'utilisateur. `GID` : *Group IDentifier*. Identifiant unique de groupe.

Danger

Il est recommandé d'utiliser les commandes d'administration au lieu de modifier manuellement les fichiers.

Gestion des groupes

Fichiers modifiés, ajout de lignes :

- `/etc/group`
- `/etc/gshadow`

La commande `groupadd`

La commande `groupadd` permet d'ajouter un groupe au système.

```
groupadd [-f] [-g GID] groupe
```

Exemple :

```
$ sudo groupadd -g 1012 GroupeB
```

Option

Description

`-g GID`

`GID` du groupe à créer.

`-f`

Le système choisit un `GID` si celui précisé par l'option `-g` existe déjà.

`-r`

Crée un groupe système avec un `GID` compris entre `SYS_GID_MIN` et `SYS_GID_MAX`. Ces deux variables sont définies dans `/etc/login.defs`.

Règles de nommage des groupes :

- Pas d'accents, ni caractères spéciaux ;
- Différents du nom d'un utilisateur ou fichier système existant.

Note

Sous **Debian**, l'administrateur devrait utiliser, sauf dans les scripts destinés à être portables sur toutes les distributions Linux, les commandes `addgroup` et `delgroup` spécifiées dans le `man` :

`man addgroup` DESCRIPTION `adduser` et `addgroup` ajoutent des utilisateurs ou des groupes au système en fonction des options fournies en ligne de commande et des informations contenues dans le fichier de configuration `/etc/adduser.conf`. Ce sont des interfaces plus conviviales que les programmes `useradd` et `groupadd`. Elles permettent de choisir par défaut des UID ou des GID conformes à la charte Debian, de créer un répertoire personnel configuré suivant un modèle (squelette), d'utiliser un script sur mesure, et d'autres fonctionnalités encore.

La commande `groupmod`.

La commande `groupmod` permet de modifier un groupe existant sur le système.

```
groupmod [-g GID] [-n nom] groupe
```

Exemple :

```
$ sudo groupmod -g 1016 GroupeP $ sudo groupmod -n GroupeC GroupeB
```

Option

Description

`-g GID`

Nouveau `GID` du groupe à modifier.

`-n nom`

Nouveau nom.

Il est possible de modifier le nom d'un groupe, son `GID` ou les deux simultanément.

Après modification, les fichiers appartenant au groupe ont un `GID inconnu`. Il faut leur réattribuer le nouveau `GID`.

```
$ sudo find / -gid 1002 -exec chgrp 1016 {} \;
```

La commande `groupdel`

La commande `groupdel` permet de supprimer un groupe existant sur le système.

```
groupdel groupe
```

Exemple :

```
$ sudo groupdel GroupeC
```

Tip

Pour être supprimé, un groupe ne doit plus contenir d'utilisateurs.

La suppression du dernier utilisateur d'un groupe éponyme entraînera la suppression de ce groupe par le système.

Tip

Chaque groupe a un `GID` unique. Un groupe peut être dupliqué. Par convention, les GID des groupes systèmes vont de 0 (root) à 999.

Tip

Un utilisateur faisant obligatoirement partie d'un groupe, il est préférable de créer les groupes avant d'ajouter les utilisateurs. Par conséquent, un groupe peut ne pas avoir de membres.

Le fichier `/etc/group`

Ce fichier contient les informations de groupes (séparées par `:`).

```
$ sudo tail -1 /etc/group GroupP: x: 516: patrick (1) (2)(3) (4)
```

- 1: Nom du groupe.
- 2: Mot de passe (`x` si défini dans `/etc/gshadow`).
- 3: GID.
- 4: Membres invités (séparés par des virgules, ne contient pas les membres principaux).

Note

Chaque ligne du fichier `/etc/group` correspond à un groupe. Les utilisateurs dont ce groupe est leur groupe principal ne sont pas listés à ce niveau. Cette information d'appartenance est en fait déjà fournie par le fichier `/etc/passwd`...

Le fichier `/etc/gshadow`

Ce fichier contient les informations de sécurité sur les groupes (séparées par `:`).

```
$ sudo grep GroupA /etc/gshadow GroupA: $6$2, 9, v... SBn160: alain: rockstar (1)  
(2) (3) (4)
```

- 1: Nom du groupe.
- 2: Mot de passe chiffré.

- 3: Administrateur du groupe.
- 4: Membres invités (séparés par des virgules, ne contient pas les membres principaux).

Warning

Pour chaque ligne du fichier `/etc/group` doit correspondre une ligne du fichier `/etc/gshadow`.

Un `!` au niveau du mot de passe indique que celui-ci est bloqué. Ainsi aucun utilisateur ne peut utiliser le mot de passe pour accéder au groupe (sachant que les membres du groupe n'en ont pas besoin).

Gestion des utilisateurslink")

Définition

Un utilisateur se définit comme suit dans le fichier `/etc/passwd` :

- 1: Login ;
- 2: Mot de passe ;
- 3: UID ;
- 4: GID du groupe principal ;
- 5 : Commentaires ;
- 6: Répertoire de connexion ;
- 7: Shell (`/bin/bash`, `/bin/nologin`, ...).

Il existe trois types d'utilisateurs :

- **root** : Administrateur du système ;
- **utilisateur système** : Utilisé par le système pour la gestion des droits d'accès des applications ;
- **utilisateur ordinaire** : Autre compte permettant de se connecter au système.

Fichiers modifiés, ajout de lignes :

- `/etc/passwd`
- `/etc/shadow`

La commande `useradd`.

La commande `useradd` permet d'ajouter un utilisateur.

```
useradd [-u UID] [-g GID] [-d repertoire] [-s shell] login
```

Exemple :

```
$ sudo useradd -u 1000 -g 1013 -d /home/GroupeC/carine carine
```

Option

Description

`-u` `UID`

`UID` de l'utilisateur à créer.

`-g` `GID`

`GID` du groupe principal.

`-d` répertoire

Répertoire de connexion.

`-s` `shell`

Interpréteur de commandes.

`-c`

Ajoute un commentaire.

`-U`

Ajoute l'utilisateur à un groupe portant le même nom créé simultanément.

`-M`

Ne crée pas le répertoire de connexion.

À la création, le compte ne possède pas de mot de passe et est verrouillé.

Il faut assigner un mot de passe pour déverrouiller le compte.

Règles de nommage des comptes :

- Pas d'accents, de majuscules ni caractères spéciaux ;
- Différents du nom d'un groupe ou fichier système existant ;
- Définir les options `-u`, `-g`, `-d` et `-s` à la création.

Warning

L'arborescence du répertoire de connexion doit être créée à l'exception du dernier répertoire.

Le dernier répertoire est créé par la commande `useradd` qui en profite pour y copier les fichiers de `/etc/skel`.

Un utilisateur peut faire partie de plusieurs groupes en plus de son groupe principal.

Pour les groupes secondaires, il faut utiliser l'option `-G`.

Exemple :

```
$ sudo useradd -u 1000 -g GroupeA -G GroupeP,GroupeC albert
```

Note

Sous Debian, il faudra spécifier l'option `-m` pour forcer la création du répertoire de connexion ou positionner la >variable `CREATE_HOME` du fichier `/etc/login.defs`. Dans tous les cas, l'administrateur devrait privilégier, sauf dans des >scripts ayant la vocation d'être portables sur toutes les distributions Linux, les commandes `adduser` et `deluser` comme précisé >dans le `man` :

```
$ man useradd DESCRIPTION      **useradd** est un utilitaire de bas niveau pour ajouter des utilisateurs. Sur Debian, les administrateurs devraient généralement utiliser **adduser(8)** à la place.
```

Valeur par défaut de création d'utilisateur.

Modification du fichier `/etc/default/useradd`.

```
useradd -D [-b repertoire] [-g groupe] [-s shell]
```

Exemple :

```
$ sudo useradd -D -g 1000 -b /home -s /bin/bash
```

Option

Description

`-D`

Définit les valeurs par défaut de création d'utilisateur.

`-b repertoire`

Définit le répertoire de connexion par défaut.

`-g groupe`

Définit le groupe par défaut.

```
-s shell
```

Définit le shell par défaut.

```
-f
```

Nombre de jours suivant l'expiration du mot de passe avant que le compte ne soit désactivé.

```
-e
```

Date à laquelle le compte sera désactivé.

La commande `usermod`.

La commande `usermod` permet de modifier un utilisateur.

```
usermod [-u UID] [-g GID] [-d repertoire] [-m] login
```

Exemple :

```
$ sudo usermod -u 1044 carine
```

Options identiques à la commande `useradd`.

Option

Description

```
-m
```

Associé à l'option `-d`, déplace le contenu de l'ancien répertoire de connexion vers le nouveau.

```
-l login
```

Nouveau nom.

```
-e AAAA-MM-JJ
```

Date d'expiration du compte.

```
-L
```

Verrouille le compte.

```
-U
```


Déverrouille le compte.

```
-a
```

Empêche la suppression de l'utilisateur d'un groupe secondaire lors de l'ajout dans un autre groupe secondaire.

```
-G
```

Précise plusieurs groupes secondaires lors de l'ajout.

Avec la commande `usermod`, le verrouillage d'un compte se traduit par l'ajout de `!` devant le mot de passe dans le fichier `/etc/shadow`.

Tip

Pour être modifié, un utilisateur doit être déconnecté et ne pas avoir de processus en cours.

Après modification de l'identifiant, les fichiers appartenant à l'utilisateur ont un `UID` inconnu. Il faut leur réattribuer le nouvel `UID`.

```
$ sudo find / -uid 1000 -exec chown 1044: {} \;
```

Où `1000` est l'ancien `UID` et `1044` le nouveau.

Il est possible d'inviter un utilisateur dans un ou plusieurs groupes secondaires avec les options `-a` et `-G`.

Exemple :

```
$ sudo usermod -aG GroupeP,GroupeC albert
```

La commande `usermod` agit en modification et non en ajout.

Pour un utilisateur invité dans un groupe par l'intermédiaire de cette commande et déjà positionné comme invité dans d'autres groupes secondaires, il faudra indiquer dans la commande de gestion de groupe tous les groupes dont il fait partie sinon il disparaîtra de ceux-ci.

L'option `-a` modifie ce comportement.

Exemples :

- Invite `albert` dans le groupe `GroupeP`.

```
$ sudo usermod -G GroupeP albert
```

- Invite `albert` dans le groupe `GroupeG`, mais le supprime de la liste des invités de `GroupeP`.

```
$ sudo usermod -G GroupeG albert
```

- Donc soit :

```
$ sudo usermod -G GroupeP,GroupeG albert
```

- Ou :

```
$ sudo usermod -aG GroupeG albert
```

La commande `userdel`.

La commande `userdel` permet de supprimer le compte d'un utilisateur.

```
$ sudo userdel -r carine
```

Option

Description

`-r`

Supprime le répertoire de connexion et les fichiers contenus.

Tip

Pour être modifié, un utilisateur doit être déconnecté et ne pas avoir de processus en cours.

`userdel` supprime la ligne de l'utilisateur dans les fichiers `/etc/passwd` et `/etc/gshadow`.

Le fichier `/etc/passwd`.

Ce fichier contient les informations des utilisateurs (séparées par `:`).

```
$ sudo head -1 /etc/passwd root:x:0:0:root:/root:/bin/bash (1)(2)(3)(4)(5) (6) (7)
```

- 1: Login.
- 2: Mot de passe (`x` si défini dans `/etc/shadow`).
- 3: UID.
- 4: GID du groupe principal.
- 5: Commentaire.
- 6: Répertoire de connexion.
- 7: Interpréteur de commandes.

Le fichier `/etc/shadow`.

Ce fichier contient les informations de sécurité des utilisateurs (séparées par `:`).

```
$ sudo tail -1 /etc/shadow root:$6$. . . :15399:0:99999:7::: (1) (2) (3) (4) (5) (6) (7,8,9)
```

- 1: Login.
- 2: Mot de passe chiffré.
- 3: Date du dernier changement.
- 4: Durée de vie minimale du mot de passe.
- 5: Durée de vie maximale du mot de passe.
- 6: Nombre de jours avant avertissement.
- 7: Délai avant désactivation du compte après expiration.
- 8: Délai d'expiration du compte.
- 9: Réserve pour une utilisation future.

Danger

Pour chaque ligne du fichier `/etc/passwd` doit correspondre une ligne du fichier `/etc/shadow`.

Les propriétaires des fichiers.

Danger

Tous les fichiers appartiennent forcément à un utilisateur et à un groupe.

Le groupe principal de l'utilisateur qui crée le fichier est, par défaut, le groupe propriétaire du fichier.

Commandes de modifications.

La commande `chown`.

La commande `chown` permet de modifier les propriétaires d'un fichier.

```
chown [-R] [-v] login[:groupe] fichier
```

Exemples :

```
$ sudo chown root monfichier $ sudo chown albert:GroupeA monfichier
```

Option

Description

`-R`

Modifie les propriétaires du répertoire et de son contenu.

```
-v
```

Affiche les modifications exécutées.

Pour ne modifier que l'utilisateur propriétaire :

```
$ sudo chown albert fichier
```

Pour ne modifier que le groupe propriétaire :

```
$ sudo chown :GroupeA fichier
```

Modification de l'utilisateur et du groupe propriétaire :

```
$ sudo chown albert:GroupeA fichier
```

Dans l'exemple suivant le groupe attribué sera le groupe principal de l'utilisateur précisé.

```
$ sudo chown albert: fichier
```

La commande `chgrp`.

La commande `chgrp` permet de modifier le groupe propriétaire d'un fichier.

```
chgrp [-R] [-v] groupe fichier
```

Exemple :

```
$ sudo chgrp groupe1 fichier
```

Option

Description

```
-R
```

Modifie les groupes propriétaires du répertoire et de son contenu (récursivité).

```
-v
```

Affiche les modifications exécutées.

Note

Il est possible d'appliquer à un fichier un propriétaire et un groupe propriétaire en prenant comme référence ceux d'un autre fichier :

```
chown [options] --reference=RRFICHER FICHER
```

Par exemple :

```
chown --reference=/etc/groups /etc/passwd
```

Gestion des invités.

La commande `gpsswd`.

La commande `gpsswd` permet de gérer un groupe.

```
gpsswd [-a login] [-A login] [-d login] [-M login] groupe
```

Exemples :

```
$ sudo gpsswd -A alain GroupeA [alain]$ gpsswd -a patrick GroupeA
```

Option

Description

```
-a login
```

Ajoute l'utilisateur au groupe.

```
-A login
```

Définit l'administrateur du groupe.

```
-d login
```

Retire l'utilisateur du groupe.

```
-M login
```

Définit la liste exhaustive des invités.

La commande `gpsswd -M` agit en modification et non en ajout.

```
# gpsswd GroupeA New Password : Re-enter new password :
```

La commande `id`.

La commande `id` affiche les noms des groupes d'un utilisateur.

```
id login
```

Exemple :

```
$ sudo id alain uid=1000(alain) gid=1000(GroupeA) groupes=1000(GroupeA),1016(GroupeP)
```

La commande `newgrp`.

La commande `newgrp` permet d'utiliser temporairement un groupe secondaire pour la création de fichiers.

```
newgrp [groupesecondaire]
```

Exemple :

```
[alain]$ newgrp GroupeB
```

Note

Après utilisation de cette commande, les fichiers seront créés avec le `GID` de son groupe secondaire.

La commande `newgrp` sans paramètre réaffecte le groupe principal.

Sécurisation.

La commande `passwd`.

La commande `passwd` permet de gérer un mot de passe.

```
passwd [-d] [-l] [-S] [-u] [login]
```

Exemples :

```
$ sudo passwd -l albert $ sudo passwd -n 60 -x 90 -w 80 -i 10 patrick
```

Option

Description

```
-d
```

Supprime le mot de passe.

`-l`

Verrouille le compte.

`-S`

Affiche le statut du compte.

`-u`

Déverrouille le compte.

`-e`

Fait expirer le mot de passe.

`-n jours`

Durée de vie minimale du mot de passe.

`-x jours`

Durée de vie maximale du mot de passe.

`-w jours`

Délai d'avertissement avant expiration.

`-i jours`

Délai avant désactivation lorsque le mot de passe expire.

Avec la commande `passwd`, le verrouillage d'un compte se traduit par l'ajout de `!!` devant le mot de passe dans le fichier `/etc/shadow`.

L'utilisation de la commande `usermod -U` ne supprime qu'un seul des `!`. Le compte reste donc verrouillé.

Exemple :

- Alain change son mot de passe :

```
[alain]$ passwd
```

- root change le mot de passe d'Alain :

```
$ sudo passwd alain
```

Note

La commande `passwd` est accessible aux utilisateurs pour modifier leur mot de passe (l'ancien mot de passe est demandé). L'administrateur peut modifier les mots de passe de tous les utilisateurs sans restriction.

Ils devront se soumettre aux restrictions de sécurité.

Lors d'une gestion des comptes utilisateurs par script shell, il peut être utile de définir un mot de passe par défaut après avoir créé l'utilisateur.

Ceci peut se faire en passant le mot de passe à la commande `passwd`.

Exemple :

```
$ sudo echo "azerty,1" | passwd --stdin philippe
```

Warning

Le mot de passe est saisi en clair, `passwd` se charge de le chiffrer.

La commande `chage`.

La commande `chage` permet de gérer la stratégie de compte.

```
chage [-d date] [-E date] [-I jours] [-l] [-m jours] [-M jours] [-W jours] [login]
```

Exemple :

```
$ sudo chage -m 60 -M 90 -W 80 -I 10 alain
```

Option

Description

`-I jours`

Délai avant désactivation, mot de passe expiré (i majuscule).

`-l`

Affiche le détail de la stratégie (l minuscule).

`-m jours`

Durée de vie minimale du mot de passe.

`-M jours`

Durée de vie maximale du mot de passe.

```
-d AAAA-MM-JJ
```

Dernière modification du mot de passe.

```
-E AAAA-MM-JJ
```

Date d'expiration du compte.

```
-W jours
```

Délai d'avertissement avant expiration.

La commande `chage` propose également un mode interactif.

L'option `-d` force la modification du mot de passe à la connexion.

Exemples :

```
$ sudo chage philippe $ sudo chage -d 0 philippe
```

Note

En l'absence d'utilisateur précisé, la commande concernera l'utilisateur qui la saisit.

Gestion du compte utilisateur avec `chage`

Gestion avancée.

Fichiers de configuration : * `/etc/default/useradd` * `/etc/login.defs` * `/etc/skel`

Note

L'édition du fichier `/etc/default/useradd` se fait grâce à la commande `useradd`.

Les autres fichiers sont à modifier avec un éditeur de texte.

Fichier `/etc/default/useradd`.

Ce fichier contient le paramétrage des données par défaut.

Tip

Lors de la création d'un utilisateur, si les options ne sont pas précisées, le système utilise les valeurs par défaut définies dans `/etc/default/useradd`.

Ce fichier est modifié par la commande `useradd -D` (`useradd -D` saisie sans autre option affiche le contenu du fichier `/etc/default/useradd`).

Valeur

Commentaire

GROUP

Groupe par défaut.

HOME

Chemin dans lequel le répertoire de connexion au nom de l'utilisateur sera créé.

INACTIVE

Nombre de jours suivant l'expiration du mot de passe avant que le compte ne soit désactivé.

EXPIRE

Date d'expiration du compte.

SHELL

Interpréteur de commandes.

SKEL

Répertoire squelette du répertoire de connexion.

CREATE_MAIL_SPOOL

Création de la boîte aux lettres dans `/var/spool/mail`.

Warning

Sans l'option `-g`, la commande `useradd` crée un groupe du nom de l'utilisateur et l'y place.

Pour que la commande `useradd` récupère la valeur du champ `GROUP` du fichier `/etc/default/useradd`, il faut préciser l'option `-N`.

Exemple :

```
$ sudo useradd -u 501 -N GroupeA
```

Fichier `/etc/login.defs`.

Ce fichier contient de nombreux paramètres par défaut utiles aux commandes de création ou de modification d'utilisateurs. Ces informations sont regroupées par paragraphe en fonction de leur utilisation :

- Boîtes aux lettres ;
- Mots de passe ;
- UID et GID ;
- Umask ;
- Connexions ;
- Terminaux.

Fichier `/etc/skel`.

Lors de la création d'un utilisateur, son répertoire personnel et ses fichiers d'environnement sont créés.

Ces fichiers sont copiés automatiquement à partir du répertoire `/etc/skel`.

- `.bash_logout`
- `.bash_profile`
- `.bashrc`

Tous les fichiers et répertoires placés dans ce répertoire seront copiés dans l'arborescence des utilisateurs lors de leur création.

Changement d'identité.

La commande `su`.

La commande `su` permet de modifier l'identité de l'utilisateur connecté.

```
su [-] [-c command] [login]
```

Exemples :

```
$ sudo su - alain [albert]$ su -c "passwd alain"
```

Option

Description

-

Charge l'environnement complet de l'utilisateur.

-c command

Exécute la commande sous l'identité de l'utilisateur.

Si le login n'est pas spécifié, ce sera `root`.

Les utilisateurs standards devront taper le mot de passe de la nouvelle identité.

Tip

Il y a création de 'couches' successives (un empilement d'environnement `bash`). Pour passer d'un utilisateur à un autre, il faut d'abord taper la commande `exit` pour reprendre son identité puis la commande `su` pour prendre une autre identité.

Chargement du profil.

`root` endosse l'identité de l'utilisateur `alain` avec `su` :

```
... /home/GroupA/alain/.bashrc /etc/bashrc ...
```

`root` endosse l'identité de l'utilisateur `alain` avec `su -` :

```
... /home/GroupA/alain/.bash_profile /home/GroupA/alain/.bashrc /etc/bashrc ...
```

Un utilisateur peut endosser temporairement (pour une autre commande ou une session entière) l'identité d'un autre compte.

Si aucun utilisateur n'est précisé, la commande concernera `root` (`su -`).

Il est nécessaire de connaître le mot de passe de l'utilisateur dont l'identité est endossée sauf si c'est `root` qui exécute la commande.

Un administrateur peut ainsi travailler sur un compte utilisateur standard et n'utiliser les droits du compte `root` que ponctuellement.

Dernière mise à jour: 9 janvier 2022

Revision #2

Created 3 October 2022 10:07:42 by edoppel

Updated 6 October 2022 12:35:53 by edoppel