Administration des utilisateurs

Il a été vu dans le chapitre Gestion des droits utilisateurs tout ce qui concernait les droits des utilisateurs. L'approche de ce paragraphe concerne plus spécialement le travail d'administration des comptes utilisateurs. L'administration des comptes utilisateurs, outre le fait qu'elle occupe dans une bonne part du temps de l'administrateur, consiste à s'occuper des tâches de création, de suppression, de modification et de consultation. Il faut évidemment être connecté en `root`...

1. Principes, commandes et tâches

La gestion des utilisateurs sous Linux/Ubuntu a pour principe l'indépendance, c'est-à-dire protéger l'utilisateur contre les accès indésirables des autres et lui octroyer un cadre de travail cohérent.

a. Principales commandes

Voici sous forme de liste récapitulative les principales commandes concernant la gestion des utilisateurs :

- **gestion des comptes utilisateurs :**
  - `adduser` : ajouter un utilisateur (préféré à `useradd` en ligne de commande).
  - `usermod` : modifier un utilisateur.
  - `deluser` : supprimer un utilisateur (préféré à `userdel` en ligne de commande).
- **gestion des groupes :**
  - `addgroup` : ajouter un groupe (préféré à `groupadd` en ligne de commande).
  - `groupmod` : modifier la définition d'un groupe.
  - `delgroup` : supprimer un groupe (préféré à `groupdel` en ligne de commande).
  - `groups` : afficher le groupe d'appartenance de l'utilisateur.
- **administration des utilisateurs :**
  - `passwd` : changer le mot de passe de l'utilisateur.
  - `chfn` : modifier les informations de l'utilisateur (champ numéro 5 du fichier `/etc/passwd`).
  - `chsh` : changer le shell (interpréteur de commandes) de l'utilisateur.
  - `id` : affiche l'identifiant de l'utilisateur (`whoami` affiche le nom de connexion).
  - `last` : afficher la liste des connexions utilisateurs.
  - `su` : passer sous l'identité de l'administrateur (`root`).
  - `sudo` : obtenir des droits étendus.
  - `who` : montrer qui est connecté.

Rappel : il existe deux types d'utilisateurs sous Linux, les utilisateurs systèmes (virtuels) ou propriétaires de processus et les utilisateurs humains dont le `root` (l'administrateur) en est un type particulier.

Les commandes préférées le sont car elles suivent la charte Debian alors que les autres correspondent à la norme POSIX (*Portable Operating System for Computer Environment*), standard UNIX. Par contre, l'administrateur utilise les commandes POSIX plutôt dans les scripts SHELL d'automatisation de création/modification d'utilisateurs.

b. Exemples d'utilisation

Création d'un nouvel utilisateur

Être utilisateur signifie être connu du poste local, de pouvoir s'y connecter, d'avoir un accès complet sur son répertoire personnel et de disposer de certains droits (réseaux ou autres). Syntaxe de la commande :

```
adduser login_utilisateur
```

Avec pour effet :

- la création du répertoire personnel `/home/login_utilisateur`,
- tout ce qui concerne la gestion et l'authentification des utilisateurs est inscrit dans un seul fichier `/etc/passwd`,
- la gestion des groupes est assurée par `/etc/group`,
- les mots de passe cryptés sont placés dans `/etc/shadow`, par sécurité lisible seulement par le `root`.

Pour les options, vous vous référerez au manuel en ligne.

Rappel : la maîtrise de la commande `adduser` (commande POSIX) est indispensable pour écrire des scripts de génération automatique de comptes.

Exemple :

```
useradd toto -u 1200 -p moi -g 1000 -s /bin/bash
```

Cette commande crée en une seule fois l'utilisateur `toto` avec le numéro d'identifiant 1200, le mot de passe `moi`, le numéro de groupe 1000 et le shell `/bin/bash`.

Rappel : la gestion des mots de passe cryptés par l'installation (courante maintenant) de la `shadow-suite` fait que le fichier `/etc/shadow` ne contient plus que le mot de passe crypté et pour des raisons de sécurité ne peut être ouvert en lecture que par son propriétaire le `root`.

Ajout/modification d'un mot de passe

Par défaut, la simple commande `useradd` crée un compte sans mot de passe et sans répertoire personnel (à la différence de `adduser`). Pour le créer ou le modifier, on aura :

```
passwd login_utilisateur
```

Suivi de deux demandes (l'autre pour la vérification). Il est possible d'avoir des messages si votre mot de passe est trop court, trop simple ("azerty") ou basé sur uniquement des lettres...

À noter :

- l'option `-d` pour supprimer le mot de passe
- l'option `-l` pour le verrouiller
- l'option `-u` pour le déverrouiller

Supprimer un utilisateur

Supprimer le compte d'un utilisateur comporte l'obligation que celui-ci ne soit pas connecté :

```
userdel [-r] login_utilisateur
```

L'option `-r` supprime aussi le répertoire personnel (non effacé par défaut), les fichiers de l'utilisateur et toute trace de l'utilisateur dans les fichiers de configuration.

Modifier un utilisateur

La commande de modification d'un compte utilisateur s'applique en fonction des options désirées.

Exemples :

```
usermod -G nom_groupe login_utilisateur
usermod -L login_utilisateur
usermod -e MM/JJ/AA login_utilisateur
```

La première ligne ajoute `login_utilisateur` dans le groupe (existant bien sûr !). La deuxième bloque le compte de l'utilisateur (voir le résultat dans le fichier `/etc/passwd`). La dernière change la date d'expiration du compte. Voir le manuel pour les autres options.

Gestion des groupes

Un groupe comporte un ensemble d'utilisateurs partageant les mêmes fichiers et répertoires et ce, avec des droits d'accès. Chaque utilisateur fait partie d'au moins un groupe, dit groupe primaire. Sur Ubuntu, celui-ci est automatiquement identique au `login` de l'utilisateur par défaut, par souci de sécurité. Un utilisateur peut faire partie de plusieurs autres groupes, appelés groupes secondaires.

Explications sur les commandes principales :

- pour créer un nouveau groupe : `addgroup nom_du_groupe`
- pour lister tous les groupes d'un utilisateur : `groups nom_du_groupe`
- pour supprimer un groupe : `delgroup nom_du_groupe`
- pour ajouter un utilisateur à un groupe : `usermod nom_du_groupe` (ou `groupmod` avec l'option `-n`).

2. Gestion avancée des utilisateurs

a. Utilisateur modèle

Le répertoire `/etc/skel` (et son contenu) sert de modèle pour les utilisateurs lors d'une création. Pour examiner les valeurs par défaut appliquées par `useradd`, on a la commande :

```
useradd -D
```

L'édition du fichier `/etc/default/useradd` donne le même résultat, soit pour Ubuntu :

```
GROUP=100                   # identifiant du groupe primaire
HOME=/home                  # racine des répertoires personnels
INACTIVE=-1                 # nombre de jours avant destruction du
                            # compte
EXPIRE=                     # nombre de jours avant expiration du mot
                            # de passe (vide par défaut)
SHELL=/bin/bash             # shell de connexion attribué au compte
SKEL=/etc/skel              # fichiers recopiés par défaut dans
                            # chaque répertoire personnel
```

Le répertoire `/etc/skel` contient, on le voit, les fichiers recopiés par défaut dans chaque répertoire personnel. Il suffit pour l'administrateur d'effectuer une modification dans ce répertoire pour qu'elle se répercute à chaque création future d'utilisateur.

Le fichier `/etc/login.defs` contient toutes les informations spécifiques à l'ordinateur pendant le processus de connexion (comme par exemple lors de la commande `/bin/login`). Se reporter au manuel en ligne et au fichier pour connaître et comprendre ses différents paramètres.

b. Utilisation des quotas de disque

Installer des quotas de disque consiste à fixer des limites de capacité de stockage pour chaque utilisateur. On **ne peut utiliser** les quotas sur une partition `root`, ou plus exactement une partition qui contient par exemple `/proc`. Le plus simple pour affecter des quotas utilisateurs, c'est d'avoir une **partition `/home` spécifique**.

Première étape : installation du paquetage

```
aptitude install quota
```

Deuxième étape : modification du fichier `/etc/fstab` avec `usrquota` pour la partition `/home`

```
UUID=votre_numéro    /home      ext3  relatime,usrquota    0    2
```

- Redémarrez le système.

Troisième étape : initialisation de la table de quotas

```
touch /home/quota.user
```

```
chmod 600 /home/quota.user
```

```
quotacheck -cu /home
```

Le fichier `quota.user` a été créé dans `/home` et contient la table des quotas.

Quatrième étape : fixation des quotas

La commande `edquota -u nom_utilisateur` sert à fixer par Nano les limites `soft` et `hard` (espace disque et nombre de fichiers pour les deux).

Exemple avec un utilisateur nouvellement créé :

```
Filesystem     blocks     soft     hard     inodes     soft      hard
/dev/sda2      20         8182     10240    6          512       1024
```

L'exemple montre pour une limite d'espace disque à 8 Mo en `soft` et 10 Mo en `hard` et un nombre de fichiers à 512 en `soft` et 1024 en `hard`. Un dépassement de la limite `soft` affiche un message d'alerte et l'utilisateur ne peut dépasser la limite `hard`.

D'autres commandes existent :

- la génération de rapports : `repquota -a`
- l'affichage d'information pour un utilisateur : `quota -u utilisateur`
- le paramètre `grace` qui fixe la période possible du dépassement de la limite soft : `edquota -t`

c. Accorder des droits supplémentaires avec sudo

Rappel : la commande `su nom_utilisateur` démarre un nouveau processus avec un nouveau shell avec l'identité de l'utilisateur, après bien sûr demande et validation du mot de passe approprié. Une utilisation courante consiste, en étant un utilisateur ordinaire, à passer en `root` afin d'effectuer une tâche d'administration. Dans ce cas, la commande `su` sans le nom de l'utilisateur suffit.

Une autre possibilité existe avec la commande `sudo`, on l'a vu (commande `sudo -i`). Cette commande permet à des utilisateurs indiqués dans le fichier `/etc/sudoers` de lancer des commandes de superutilisateur. Pour accorder des droits, il faut modifier le fichier `/etc/sudoers` par la commande `visudo` et bien sûr en étant en `root`.

Configuration de sudo

```
visudo
```

```
# /etc/sudoers
# This file MUST be edited with the 'visudo' command as root.
# See the man page for details on how to write a sudoers file.

Defaults    env_reset
# Host alias specification
# User alias specification
# Cmnd alias specification

# User privilege specification
root    ALL=(ALL) ALL

# Uncomment to allow members of group sudo to not need a password
# (Note that later entries override this, so you might need to move
# it further down
# %sudo ALL=NOPASSWD: ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL
```

Ce fichier contient deux types d'entrées : les **alias** ou plus simplement les variables et les spécifications des utilisateurs. En cas de multiples entrées pour un utilisateur, les règles s'appliquent dans l'ordre de lecture. La syntaxe des entrées ou règles vient de la forme EBNF (*Extended Backus-Naur Form*), nom barbare pour une lecture assez simple : en gros sous la forme de paires "champs = valeurs".

L'étude complète de cette syntaxe `sudo` ne présente pas véritablement d'intérêt. Aussi, son apprentissage se limite à l'étude de quelques exemples.

Exemples :

```
donald     ALL = (ALL) ALL
```

L'utilisateur `donald` peut lancer toutes les commandes sur toutes les machines.

```
Donald     ALL = NOPASSWD: ALL
```

L'utilisateur `donald` peut lancer toutes les commandes sur toutes les machines sans demande de mot de passe.

```
Host_Alias        RESEAU = 192.168.3.0/255.255.255.0
donald            RESEAU = (ALL) ALL
```

L'utilisateur `donald` peut lancer toutes les commandes sur toutes les machines du réseau donné par l'alias.

```
%info          ALL = (ALL) ALL
```

Tous les utilisateurs du groupe `info` peuvent lancer toutes les commandes sur toutes les machines.

```
Cmnd_Alias         REBOOT = /sbin/reboot
donald             ALL = ALL, !REBOOT
```

L'utilisateur `donald` ne peut relancer le système. Notez la syntaxe qui autorise tout d'abord toutes les commandes pour ensuite restreindre spécifiquement le `reboot`.

```
User_Alias         WEB = donald, riri, fifi, loulou
WEB                srvweb = (www-data) ALL, (root) /bin/su www-data
```

Sur la machine `srvweb`, tous les utilisateurs listés dans WEB peuvent lancer toutes les commandes sous l'identité `www-data` (c'est le propriétaire des pages Web sous Apache) ou simplement se transférer sous ce compte.

```
donald   ALL=NOPASSWD: /usr/bin/vim /etc/passwd
```

Donne à `donald` le droit d'éditer et de modifier le fichier `/etc/passwd` sans demande de mot de passe.