UNIX

DROITS ET UTILISATEURS – DEFINITIONS

Table des matières

1 Permissions UNIX

1.1 Notion d'utilisateur (user)

Toute entité (personne physique ou programme particulier) devant interagir avec un système UNIX est authentifié sur cet ordinateur par un utilisateur ou user. Ceci permet d'identifier un acteur sur un système UNIX. Un utilisateur est reconnu par un nom unique et un numéro unique (la correspondance nom/numéro est stockée dans le fichier /etc/passwd).

Tous les utilisateurs UNIX n'ont pas les mêmes droits d'accès à l'ordinateur (ils ne peuvent pas tous faire la même chose), et ceci simplement pour des raisons de sécurité et d'administration. Par exemple, pour éviter tout problème sur Internet, l'utilisateur qui gère le serveur HTTP n'a pas le droit d'exécuter des commandes localement, pour éviter que le serveur ne puisse le faire.

Certains utilisateurs ne peuvent en effet pas s'authentifier sur l'ordinateur et accéder à un interpréteur de commandes. Cela ne veut toutefois pas dire qu'ils ne peuvent rien faire sur l'ordinateur : il leur est possible de lire ou écrire des fichiers mais cela nécessite que le super-utilisateur (voir plus bas) démarre un programme pour cet utilisateur. Ce mécanisme est généralement utilisé pour les démons : le super utilisateur démarre le démon et pour éviter que ce dernier ne puisse faire tout et n'importe quoi sur la machine, il est par exemple attribué à l'utilisateur bin.

Sur tout système UNIX, il y a un super-utilisateur, généralement appelé root, qui a tous les pouvoirs. Il peut accéder librement à toutes les ressources de l'ordinateur, y compris à la place d'un autre utilisateur, c'est-à-dire sous son identité. En général, du moins sur les systèmes de production, seul l'administrateur système possède le mot de passe root. L'utilisateur root porte le numéro 0.

1.2 Groupe

Un utilisateur UNIX appartient à un ou plusieurs groupes. Les groupes servent à rassembler des utilisateurs afin de leur attribuer des droits communs. Par exemple, sur un système doté d'une carte son, il y a souvent un groupe audio qui regroupe les utilisateurs autorisés à en faire usage.

1.3 Propriétaire

Tout fichier UNIX possède un propriétaire. Au départ, c'est l'utilisateur qui a créé le fichier mais "root" peut le « donner » à un autre utilisateur. Seul le propriétaire du fichier et le super utilisateur (root) peuvent changer les droits.

Traditionnellement, les Unix "System V" permettent au propriétaire de changer le possesseur d'un fichier, mais dans la tradition BSD et sous Linux, seul "root" peut changer le propriétaire d'un fichier, notamment pour éviter qu'un utilisateur n'échappe aux quotas disque en donnant ses fichiers à un autre utilisateur.

Un fichier UNIX appartient aussi à un groupe. Ceci donne pleinement son sens à la notion de groupe. On définit ainsi les actions du groupe sur ce fichier. Ce groupe est souvent le groupe d'appartenance du propriétaire, mais ce n'est pas obligatoire. Tout dépend en fait de ce qu'on veut faire. On peut imaginer un scénario de délégation d'administration : le super utilisateur est propriétaire d'un fichier de configuration, mais autorise tous les utilisateurs du groupe admin (les administrateurs) à modifier

ce fichier. Le fichier en question aura donc root comme propriétaire et appartiendra au groupe admin.

Rappelons que les répertoires sous UNIX sont aussi des fichiers. Les droits sur les répertoires (mais aussi les périphériques, etc.) fonctionnent exactement de la même façon que sur des fichiers ordinaires.

1.4 Droits d'accès à un fichier

À chaque fichier est associée une liste de permissions, qui déterminent ce que chaque utilisateur a le droit de faire du fichier.

1.5 Les différents droits

Les droits sur un fichier UNIX s'attribuent sur trois « actions » différentes possibles :

- la lecture (r) : on peut par exemple lire le fichier avec un logiciel. Lorsque ce droit est alloué à un répertoire, il autorise l'affichage du contenu du répertoire (la liste des fichiers présents à la racine de ce répertoire).
- l'écriture (w) : on peut modifier le fichier et le vider de son contenu. Lorsque ce droit est alloué à un répertoire, il autorise la création, la suppression et le changement de nom des fichiers qu'il contient, quels que soient les droits d'accès des fichiers de ce répertoire (même s'ils ne possèdent pas eux-mêmes le droit en écriture). Néanmoins le droit spécial sticky bit permet de passer outre ce comportement.
- l'exécution (x) : on peut exécuter le fichier s'il est prévu pour, c'est-à-dire si c'est un fichier exécutable. Lorsque ce droit est attribué à un répertoire, il autorise l'accès (ou ouverture) au répertoire.

On appelle parfois r, w et x des « flags » ou « drapeaux ». Sur un fichier donné, ces trois flags doivent être définis pour son propriétaire, son groupe, mais aussi les autres utilisateurs (différents du propriétaire et n'appartenant pas au groupe).

Seuls root et le propriétaire d'un fichier peuvent changer ses permissions d'accès.

1.6 Représentation des droits

Cet ensemble de trois droits sur trois entités se représente généralement de la façon suivante : on écrit côte à côte les droits r, w puis x respectivement pour le propriétaire (u), le groupe (g) et les autres utilisateurs (o). Les codes u, g et o (u comme user, g comme group et o comme others) sont utilisés par les commandes UNIX qui permettent d'attribuer les droits et l'appartenance des fichiers. Lorsqu'un flag est attribué à une entité, on écrit ce flag (r, w ou x), et lorsqu'il n'est pas attribué, on écrit un '-'. Par exemple,

```
rwx r-x r—
①  ②   ③
```

③ droits des autres utilisateurs (o)

② droits des utilisateurs appartenant au groupe (g)

① droits du propriétaire (u)

signifie que le propriétaire peut lire, écrire et exécuter le fichier, mais que les utilisateurs du groupe attribué au fichier ne peuvent que le lire et l'exécuter, et enfin que les autres utilisateurs ne peuvent que lire le fichier.

Une autre manière de représenter ces droits est sous forme binaire grâce à une clef numérique fondée sur la correspondance entre un nombre décimal et son expression binaire :

- 0 = 000
- 1 = 001
- 2 = 010
- 3 = 011
- 4 = 100
- 5 = 101
- 6 = 110
- 7 = 111

À l'expression binaire en trois caractères sont associés les 3 types de droits (r w x) ; il suffit donc de déclarer pour chacune des catégories d'utilisateur (user, group, others) un chiffre entre 0 et 7 auquel correspond une séquence de droits d'accès. Par exemple :

- 777 donne 111 111 111 soit r w x r w x r w x
- 605 donne 110 000 101 soit r w - - - - r - x
- 644 donne 110 100 100 soit r w - r - - r - -
- 666 donne 110 110 110 soit r w - r w - r w -

Une astuce permet d'associer rapidement une valeur décimale à la séquence de droits souhaitée. Il suffit d'attribuer les valeurs suivantes pour chaque type de droit :

- lecture (r) correspond à 4
- écriture (w) correspond à 2
- exécution (x) correspond à 1

Puis on additionne ces valeurs selon qu'on veuille ou non attribuer le droit en correspondant.

Ainsi, rwx « vaut » 7 (4+2+1), r-x « vaut » 5 (4+1) et r-- « vaut » 4. Les droits complets (rwxr-xr--) sont donc équivalent à 754. Une manière directe d'attribuer les droits est de les écrire sous cette forme et d'utiliser le code à 3 chiffres résultant avec chmod (voir ci-après).

1.7 Utilisation

Pour voir quels droits sont attribués à un fichier, il suffit de taper la commande ls -l nom_du_fichier :

```
ls -l toto
-rwxr-xr--    1 user     group     12345 Nov 15 09:19 toto
```

La sortie signifie que le fichier toto (de taille 12345) appartient à « user », qu'on lui a attribué le groupe « group », et que les droits sont rwxr-xr--. On remarque qu'il y a en fait 10 caractères sur la zone de droits. Le premier - n'est pas un droit, c'est un caractère réservé pour indiquer le type de fichier. Il peut prendre les valeurs suivantes :

- d : répertoire
- l : lien symbolique
- c : périphérique de type caractère
- b : périphérique de type bloc
- p : fifo
- s : socket
- - : fichier classique

Le changement de droits s'effectue avec la commande chmod ; le changement de propriétaire ou de groupe, à l'aide de la commande chown.

Changer les droits peut s'effectuer également simplement à partir du nombre à 3 chiffres calculé comme précédemment. Ainsi, pour attribuer les droits r-xr-xr-x (i.e. 555), il suffit d'exécuter :

```
chmod 555 nom_du_fichier
```

2 Qui est utilisateur ?

Le système, dès son installation, avant même la première connexion au système a créé des users système.

Un utilisateur n'est donc pas uniquement une personne physique, le système a besoin d'utilisateurs pour sa gestion interne, notamment comme propriétaire des divers processus.

La commande ps aux | less montre qu'avant toute connexion d'utilisateur humain (repérée par les lignes login --user), root a lancé init, et la plupart des services, crond, inetd, lpd, smbd, ... , avant de lancer les connexions utilisateurs dans les consoles, y compris éventuellement la sienne !

Les principales commandes

- useradd, usermod, userdel : gestion des comptes utilisateur
- groupadd, groupmod, groupdel : gestion des groupes
- pwck, grpck : vérification des fichiers
- passwd : changer le mot de passe d'un utilisateur
- chfn, id, groups, finger : utilitaires divers

3 Gestion des comptes utilisateur

3.1 Créer un compte pour un nouvel utilisateur

Cela signifie lui permettre d'être connu du poste local, s'y loguer, avoir un accès complet sur son répertoire personnel.

Mais aussi dans une configuration réseau, de pouvoir se connecter à son compte par telnet et ftp, et de pouvoir bénéficier de services réseau de partage distant (sous Linux par NFS et sous Windows par SMB).

Pour créer l'utilisateur tux, root passe la commande :

```
useradd tux
```

Ceci crée :

- le répertoire personnel /home/tux, portant par défaut le nom du compte
- une nouvelle entrée dans les 2 fichiers fondamentaux /etc/passwd et /etc/group.

3.2 Attribuer un mot de passe

Pour lui attribuer le mot de passe tux :

```
passwd tux
```

Saisir 2 fois tux

3.3 Supprimer le compte d'un utilisateur (non connecté)

```
userdel [-r] tux
```

L'option -r supprime aussi le répertoire personnel et les fichiers de l'utilisateur.

La commande supprime toute trace de l'utilisateur dans le fichier de configuration : /etc/passwd y compris dans les groupes d'utilisateurs.

3.4 Modifier le compte de l'utilisateur toto

```
usermod [options] toto
```

Les options sont les mêmes que useradd

`usermod -G stagiaire,prof toto` ajoute toto dans les 2 groupes stagiaire et prof (qui doivent exister)

4 Les groupes

Un groupe est, aussi pour Linux, un ensemble d'utilisateurs qui partagent les mêmes fichiers et répertoires. Nous verrons que les fichiers accordent des droits d'accès réglables à ces groupes.

Chaque utilisateur doit faire partie au moins d'un groupe, son groupe primaire. Celui-ci est défini au moment de la création du compte, et par défaut, l'utilisateur appartient à un nouveau groupe créé, portant son nom.

Ainsi, dans /etc/passwd chaque utilisateur possède un groupe par défaut, précisé par son identifiant gid dans ce fichier.

L'appartenance au groupe primaire n'étant pas exclusive, tout utilisateur peut faire partie de plusieurs autres groupes, appelés ses groupes secondaires.

Mais le rôle joué par le groupe primaire demeure prépondérant, comme nous le verrons dans le système des permissions des fichiers.

4.1 Lister tous les groupes (primaire et secondaires) d'un utilisateur

```
groups toto
```

4.2 Créer un nouveau groupe

```
groupadd stagiaire
```

4.3 Supprimer un groupe

```
groupdel stagiaire
```

Le groupe est supprimé du fichier /etc/group.

4.4 Ajouter un utilisateur à un groupe

Le plus simple est d'éditer le fichier /etc/group et d'ajouter une liste d'utilisateurs (séparés par des virgules) sur la ligne du groupe (ou utiliser Linuxconf).

5 Commandes su, sudo

5.1 su (switch user)

Il y a plusieurs manières d'obtenir les droits root ou ceux d'un autre utilisateur sous linux. Il faut remarquer qu'à ce titre, Ubuntu se distingue des autres distributions dans le sens où il n'y a pas d'utilisateur root mais que les droits root sont donnés par défaut à un utilisateur.

```
su
```

La principale commande pour changer d'utilisateur ou devenir root est su.

Elle s'utilise comme ceci:

```
$ su utilisateur
```

ou

```
$ su root
```

Dans le cas de root, il n'est pas nécessaire de le spécifier. su tout court suffit.

Une petite particularité, si vous faites suivre la commande su ci-dessus d'un '-':

```
$ su - toto
```

ou

```
$ su -
```

Vous changez d'utilisateur et vous passez sur le home et les variables d'environnement définies par le nouvel utilisateur.

Pour sortir de ce mode, tapez :

```
$ exit
```

5.2 sudo (abréviation de substitute user do)

sudo exécute une commande sous un autre utilisateur. Pour qu'il puisse le faire, il faut qu'il en ait le droit. Pour cela, il doit se trouver dans une liste d'utilisateurs autorisés. Cette liste se trouve dans le fichier /etc/sudoers.

C'est cette méthode qu'utilise Ubuntu. Il n'y a pas d'utilisateur root mais n'importe quel utilisateur peut être mis dans la liste de ceux autorisés à obtenir des droits root. Dans le fichier /etc/sudoers, on trouve une ligne comme celle-ci:

```
%admin ALL=(ALL) ALL
```

Cela signifie que tous les utilisateurs qui font partie du groupe admin peuvent avoir des droits root. Autrement dit si sur votre machine vous voulez autoriser l'utilisateur toto à pouvoir utiliser sudo

pour obtenir des droits root, il vous suffit d'ajouter toto au groupe d'utilisateurs admin.

```
$ sudo commande
```

exécute la commande avec les droits root. Votre mot de passe vous sera demandé pour vous authentifier.

```
$sudo -u toto commande
```

exécute la commande avec les droits de toto.

Attention, sudo ne passe pas sur le home du nouvel utilisateur et ne switche pas sur ses variables d'environnement.

6 Visite des coulisses

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 chiffrés sont maintenant placés dans **/etc/shadow**, par sécurité lisible seulement par root.

Structure de /etc/passwd

Ce fichier comprend 7 champs, séparés par le symbole :

1. nom de connexion
2. ancienne place du mot de passe chiffré
3. numéro d'utilisateur uid, sa valeur est le véritable identifiant pour le système Linux; l'uid de root est 0, le système attribue conventionnellement un uid à partir de 500 aux comptes créés (1000 sous Ubuntu).
4. numéro de groupe gid, dans lequel se touve l'utilisateur par défaut; le gid de root est 0, des groupes d'utilisateurs au delà de 500
5. nom complet, il peut être suivi d'une liste de renseignements personnels (cf chfn)
6. répertoire personnel (c'est également le répertoire de connexion)
7. shell, interprétateur de commandes (par défaut /bin/bash)

Structure de /etc/group

Ce fichier comprend 4 champs, séparés par le symbole :

1. nom du groupe
2. x pour remplacer un mot de passe non attribué maintenant
3. numéro de groupe, c-à-d l'identifiant gid
4. la liste des membres du groupe