Utilisateurs, groupes, profils et mécanismes de droits

Gorgoumack SAMBE

Université de Ziguinchor

Version 1.0[1]

[1] Décembre 2012

Objectifs

A l'issue de ce chapitre, l'apprenant doit être capable de :

- distinguer les types de comptes d'utilisateurs sur Linux
- personnaliser un profil utilisateur
- distinguer les groupes d'utilisateurs sur Linux
- distinguer et modifier les droits liés à un fichier
- configurer les droits par défaut des fichiers

Plan

Plan

Utilisateur

Sur Linux ,il existe trois types de comptes :

1. compte de l'administrateur du système(super-utilisateur) : Il a tous les droits sur le système et est appelé par défaut root.
2. comptes d'utilisateurs : ils utilisent les ressources mis à leur disposition par le système :
   Exemple : un compte assane.
3. comptes systèmes : compte lié à une application(pour des raisons techniques) :
   - un utilisateur vboxuser lié au serveur virtualbox
   - un utilisateur mysql lié au serveur mysql.

Répertoire personnel(home)

- Tout utilisateur possède un répertoire personnel appelé home directory.
  Il stocke
  - les données de l'utilisateur.
  - le profil de l'utilisateur : fichiers de configuration de son environnement (couleur des fenêtres, thème,favoris internet. . .).
    - Configuration à travers les fichiers cachés qui sont dans le home(.profile, .bashrc,. . .).
- Exemples :
  - /root pour le root
  - /home/assane pour l'utilisateur assane.

Profil utilisateur

- Changement de profil
  - su -omar
  - su -root $\Longleftrightarrow$ su
  - su assane (ne charge pas son profil)
- Lancer une commande sans devenir root de manière permanente
  - sudo rm /root/test
  - il faut faire partie du **groupe sudo**(pas disponible pour toutes les distributions).

Groupe

- Catégorie d'utilisateurs à laquelle on peut associer une politique d'accès.
  - un groupe est défini à partir d'une liste d'utilisateur éventuellement vide.
  - un utilisateur doit faire partie au moins d'un groupe, son **groupe primaire**.
    - porte généralement le même nom que l'utilisateur.
  - un utilisateur peut appartenir à plusieurs groupes.
- Exemples
  - le groupe assane : groupe par défaut de l'utilisateur assane.
  - le groupe sudo : groupe possédant plus ou moins de droit d'administration à travers la commande sudo.
- groups : affiche les groupes d'appartenance de l'utilisateur

Quelques commandes

- useradd (adduser) : créer un nouvel utilisateur.
- userdel : supprimer un compte utilisateur et les fichiers associés.
- usermod : modifier un compte utilisateur.
- groupadd, groupdel, groupmod : ???
- passwd : modifier le mot de passe d'un utilisateur.
- id : affiche les identifiants d'utilisateur et de groupes.

Plan

Droits sur les fichiers et répertoires

- La gestion des droits d'accès est une caractéristique du système de fichier linux (Droits POSIX sur le VFS et natif sur ext).
- Les informations stockées dans un inode sont :
  - Utilisateur propriétaire
  - Groupe propriétaire
  - ...

La commande ls -l permet d'avoir les informations sur les propriétaires du fichier.

[savary@iut0039 CoursSE]$ ls -al cours/
total 1236
drwxr-xr-x 2 savary Enseignants 4096 jan 11 16:05 .
drwxr-x–x 7 savary Enseignants 4096 jan 11 15:19 ..
-rw-r–r– 1 savary Enseignants 329033 jan 11 10:56 cours1
...

lrwxrwxrwx 1 savary Enseignants 13 jan 11 15:07 doc -> ../doc-linux/

Droits sur les fichiers et répertoires

Pour un fichier donné, on classifie les utilisateurs en 3 catégories

1. Le propriétaire: **u** (user)
2. Les membres du groupe propriétaire: **g** (group)
3. Les autres: **o** (others)
4. . . . et tous les utilisateurs **a** (all):

3 opérations élémentaires sont contrôlées.

- La lecture: **r** (**4**) pour la lecture (Read)
- L'écriture: **w** (**2**) pour l'écriture (Write)
- L'exécution: **x** (**1**) pour l'exécution (eXecute)

La commande ls -l permet d'avoir les informations sur les droits d'accès.

	user			group			other		
	r	w	x	r	w	x	r	w	x
	4	2	1						
Modes spéciaux			s			s			t

Signification des droits

	Fichier	Répertoire
r	Lecture autorisée. On peut lire son contenu avec cat, more, gedit, etc.	Lecture de la totalité du répertoire possible. On peut lancer la commande ls pour voir les entrées du répertoire.
w	Ecriture autorisée. On peut modifier son contenu.	On peut y créer ou supprimer des fichiers ou répertoires.
x	Exécution autorisée. On peut lancer le programme contenu dans le fichier.	En absence de ce droit, aucun accès au répertoire et à la sous arborescence issue du répertoire n'est possible. On peut s'y positionner avec la commande cd.

- Toutes les combinaisons de ces droits sont possibles.
- Par exemple, on peut lire le contenu d'un fichier dans un répertoire que l'on n'a pas le droit de lire.

Changement des droits

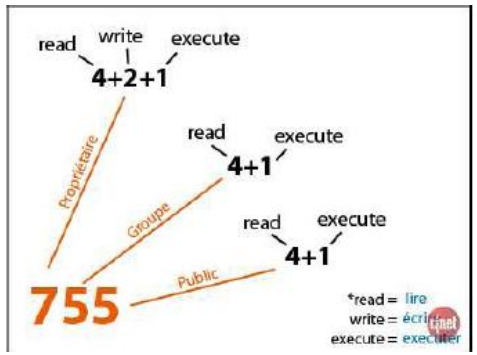
Changement ponctuel de droit sur un fichier: commande chmod

- Mode symbolique

chmod u+x script.sh	+ eXéc. pour u
chmod g-w image.jpg	- Write. à g
chmod o-rwx image.jpg	- tous les droits à o
chmod ug=rwx monrep	= tous les droits pour u et g
chmod a=rw image.jpg	= Read et Write pour tous (a=all=ugo)
chmod ug+x, o-r monrep	+ eXéc. pour u et g, - lect. pour o
chmod a-rwx test.sh	- tous les droits à tous (a=all=ugo)

- Mode numérique (codage en octal de bits de permission).

chmod 755 image.jpg



Exemples

Command	Object	Meaning
chmod 400	file	To protect a file against accidental overwriting/
chmod 500	directory	To protect yourself from accidentally removing, renaming or moving files from this directory.
chmod 600	file	A private file only changeable by the user who entered this command.
chmod 644	file	A publicly readable file that can only be changed by the issuing user.
chmod 660	file	Users belonging to your group can change this file, others don't have any access to it at all.
chmod 700	file	Protects a file against any access from other users, while the issuing user still has full access.
chmod 755	directory	For files that should be readable and executable by others, but only changeable by the issuing user.
chmod 775	file	Standard file sharing mode for a group.
chmod 777	file	Everybody can do everything to this file.

Droits par défaut : umask

- Des fichiers sans droit n'existent pas sous Unix
- À chaque création de fichier, des droit par défaut sont appliqués.
- Ces droits par défaut sont définis grâce à la commande umask

- La valeur actuelle du masque est obtenu en tapant umask.
  - $ umask
    0022

Principe d'umask

1. À chaque création de fichier (téléchargement à partir d'internet, sauvegarde, etc.), une fonction de création de fichier est lancée. en donnant les permissions suivantes:

   Répertoire: droits 777 ou rwxrwxrwx
   Fichier: droits 666 ou rw-rw-rw-

2. Application du masque:
   - La valeur (les droits) umask est soustraite(en réalité AND NOT) des droits ci-dessus.

		777	Valeur avant masque
Répertoire:	-	022	masque
		755	Droits par défaut du répertoire

		666	Valeur avant masque
Fichier:	-	022	masque
		644	Droits par défaut du fichier

Droits spéciaux : SUID et SGID

- Les utilisateurs souhaitent pouvoir changer de mot de passe ou de shell sans le demander à l'administrateur système.
- Les noms des utilisateurs, leur shell et leur mot de passe sont stockés dans /etc/passwd(ou /etc/shadow) dont les droits d'accès sont restreints.
  - $ ls -l /etc/passwd
    -rw-r–r– 1 root root 1817 2011-12-29 14:32 /etc/passwd
- **Solution** : donner une permission spéciale au programme /usr/bin/passwd. passwd est exécuté avec les droits de l'utilisateur ou du groupe possédant le programme et non les droits de l'utilisateur l'ayant lancé.

Droits spéciaux : SUID et SGID

- Le suid ou le sgid est indiqué par un "s" à la place du droit x pour l'user ou le group, selon l'emplacement du symbole.
  - $ which passwd
    /usr/bin/passwd
    $ ls -l /usr/bin/passwd

    -rwsr-xr-x 1 root root 37140 2011-02-14 22:11 /usr/bin/passwd

- Le suid ou le sgid est aussi établi de deux manières

chmod u+s programme	#	#	chmod g+s programme
	# OU	#	
chmod 4000 programme	#	#	chmod 2000 programme

Le 4 au début indique le suid ## Le 2 indique le sgid

Droits spéciaux : Sticky Bit

- L'administrateur système peut souhaiter avoir des répertoires partagés dans lesquels chaque utilisateur a des droits d'écriture.
- Il ne serait pas souhaitable de laisser un utilisateur supprimer ou renommer un fichier ne lui appartenant pas.
  - Solution: sticky bit.

  Le sticky bit est indiqué par un "t" à la place du droit x pour les others.
- >$ ls -ld /var/tmp
  drwxrwxrwt 2 root root 4096 2011-12-29 09:55 /var/tmp
- Le sticky bit indique que seul le propriétaire du répertoire et le propriétaire d'un objet qui s'y trouve ont le droit de supprimer cet objet.

- Le sticky bit est établi de deux manières
  chmod o+t Répertoire
  chmod 1777 Répertoire # le 1 au début indique le sticky bit

Changement de propriétaire

Changer le propriétaire et le groupe:

- Vous pouvez "donner" un fichier vous appartenant à un autre utilisateur, c'est à dire qu'il deviendra propriétaire du fichier, et que vous n'aurez plus que les droits que le nouveau propriétaire voudra bien vous donner sur le fichier.
  - chown nouveau-propriétaire nom-fichier
- Dans le même ordre d'idée vous pouvez changer le groupe.
  - chgrp nouveau-groupe nom-fichier
- Commandes utilisables que si on est propriétaire du fichier.
  - NB: Sur certains UNIX, on interdit leur usage (des raisons de sécurité)