

# Gestion des utilisateurs

Abdelali SAIDI

abdelali.saidi@gmail.com

# Plan

- 1 Gestion des utilisateurs
- 2 Gestion des groupes
- 3 Gestion de permissions
- 4 Gestion des ACLs

# Plan

- 1 Gestion des utilisateurs
- 2 Gestion des groupes
- 3 Gestion de permissions
- 4 Gestion des ACLs

# Gestion des utilisateurs

## Le fichier /etc/passwd

Chaque utilisateur du système doit avoir une ligne de paramètres au niveau de ce fichier. Ces paramètres représentent :

- Le login: L'identifiant de l'utilisateur
- Le mot de passe: Ce champs concerne le mot de passe et peut avoir plusieurs significations :
  - \*: impossible d'ouvrir une session avec ce compte
  - !!: le compte est désactivé
  - x ou !: le mot de passe est chiffré et se trouve sur le fichier /etc/shadow
- UID: L'identifiant unique de l'utilisateur et doit être supérieur à 100
- UIG: L'identifiant unique du groupe principale de l'utilisateur
- Commentaire: des informations à propos de l'utilisateur
- Le répertoire personnel: Sous la forme \$HOME/nomUser
- Le shell: Le shell par défaut de l'utilisateur

# Gestion des utilisateurs

## Le fichier /etc/shadow

- Le fichier /etc/passwd est accessible par tous les utilisateurs.
- /etc/shadow peut être lu seulement par le super-utilisateur et contient le haché des mots de passe

# Gestion des utilisateurs

## Le fichier /etc/shadow

Les paramètres de ce fichier sont :

- Le login de l'utilisateur
- Le mot de passe haché
- La dernière modification du mot de passe (en jours depuis le 1 janvier 1970)
  - 0 signifie que l'utilisateur devrait modifier son mot de passe à la prochaine ouverture de session
  - le vide signifie que le mot de passe est valable pour toujours
- Le nombre de jours minimal que l'utilisateur n'aura pas le droit de modifier son propre mot de passe
- Le nombre de jours maximal après lesquels l'utilisateur sera obligé de modifier son mot de passe
- Période d'avertissement (à propos de l'expiration du mot de passe)
- Le nombre de jours d'inactivité après lesquels le compte sera désactivé
- La date d'expiration du compte (en jours depuis le 1 janvier 1970)

# Plan

- 1 Gestion des utilisateurs
- 2 Gestion des groupes**
- 3 Gestion de permissions
- 4 Gestion des ACLs

# Gestion des groupes

## Le fichier /etc/group

Le fichier /etc/group contient les informations qui concernent les groupes existants

- Groupe : Le nom du groupe
- Mot de passe : si la valeur est x cela veut dire que le mot de passe est chiffré et se trouve sur le fichier /etc/gshadow
- GID : L'identifiant unique du groupe
- Membres: Les utilisateurs appartenant au groupe

# Gestion des groupes

## Le fichier /etc/gshadow

Chaque ligne correspond à un groupe et contient les informations de telle sorte :

- Le nom du groupe
- Le mot de passe
- L'administrateur du groupe
- Les membres du groupe

# Les commandes

En ce qui concerne la gestion des utilisateurs

- useradd
- passwd
- usermod
- userdel
- su
- chage

# Les commandes

En ce qui concerne la gestion des groupes

- groupadd
- groupdel
- groupmod
- gpasswd
- newgrp

# Plan

- 1 Gestion des utilisateurs
- 2 Gestion des groupes
- 3 Gestion de permissions**
- 4 Gestion des ACLs

# Le propriétaire

## ls -l

La commande ls munit de l'option -l permet d'afficher plusieurs informations en champs avec le listing du contenu d'un répertoire donnée.

- -rwxr-xr-x 1 root admin 32464 27 May 09:35 test.c

Dans cet exemple, nous avons :

- Root est l'utilisateur propriétaire du fichier
- Admin est le groupe propriétaire du fichier

# Le propriétaire

## Terminologies

- L'utilisateur propriétaire : l'utilisateur détermine l'identité des comptes existants sur la machine. Chaque utilisateur possède un UID (User IDentifier). L'utilisateur propriétaire est désigné (souvent) celui qui a créer le fichier
- Le groupe propriétaire : un groupe est un ensemble d'utilisateurs auxquels on voudrait attribuer les mêmes droits. Chaque groupe possède un GID (Group IDentifier)
- Les autres : tout autre utilisateur

## Définition

La notion de propriété à propos d'un fichier détermine la façon avec laquelle il peut être traité par un utilisateur donné.

# Le propriétaire

## La commande chown

La commande chown permet de changer l'utilisateur propriétaire d'un fichier. Elle ne peut être exécuté que par l'utilisateur root.

- La syntaxe : `chown utilisateur[:groupe] fichier`

## chgrp

La commande chgrp permet de changer le groupe propriétaire d'un fichier. Elle peut être exécuté par l'utilisateur propriétaire du fichier.

- La syntaxe : `chgrp groupe fichier`

NB: L'utilisateur doit appartenir au nouveau groupe

# Types de permissions

## Présentation

Sur un SE GNU/Linux, nous avons trois permissions pour chaque fichier:

- r : lire (read)
- w : écrire (write)
- x : exécuter (execute)

Les utilisateurs du système sont désignés par:

- u : utilisateur propriétaire
- g : groupe propriétaire
- o : les autres

# Types de permissions

## La commande chmod

La commande chmod permet de modifier les permissions d'un fichier.

- Syntaxe : chmod typeUtilisateur opérateur permission fichier
  - typeUtilisateur : u/g/o/a
  - opérateur : +/=/-
  - permission : r/w/x

## Exemples

- chmod g-rw file
- chmod a+r file

# Types de permissions

## La commande chmod

Cette commande peut être utilisée d'une manière numérique. Pour cela, un numéro de trois chiffres décrit les permissions de tous les utilisateurs. Chaque chiffre est la somme des permissions que chaque type d'utilisateur possède en ce qui concerne un fichier donnée. Les valeurs d'affectation sont :

- r = 4
- w = 2
- x = 1

## Exemples

- chmod 751 file
  - 7 = 4 + 2 + 1 (lecture, écriture et exécution pour l'utilisateur propriétaire)
  - 5 = 4 + 1 (lecture et exécution pour le groupe propriétaire)
  - 1 = 1 (exécution pour les autres)

# Types de permissions

## Exercice

Soit commande.sh un fichier qui possède les permissions rw-rw-r-

- Ajouter le droit d'exécution à tous les utilisateurs
- Enlever le droit d'écriture au groupe propriétaire
- Enlever le droit de lecture aux autres utilisateurs

# La commande umask

## Présentation

A la création d'un objet, le système met des valeurs de permissions automatiquement. La commande umask permet de modifier ces valeurs pour une session courante.

- umask x : x est un nombre exprimé sous forme octale qui déterminera les permissions par complétion (AND) de 0666 pour les fichiers et de 0777 pour les répertoires qui seront créés ultérieurement

## Exercice

Avec quels droits seront créés les fichiers après l'exécution de la commande umask 022 ?

# Les droits spéciaux

## SUID

Normalement, l'exécution d'une commande hérite les permissions de l'utilisateur qui l'a lancée. La permission SUID permet à une commande d'hériter les permissions de son propriétaire lors de son exécution par n'importe quel autre utilisateur

## Caractéristiques

- Le SUID est représenté par "s" et 4000 numériquement
- Il prend place de la permission "x" de l'utilisateur propriétaire
- Activation : chmod u+s fichier

Exemple : Vérifiez les permissions de la commande /bin/ping

# Les droits spéciaux

## SGID

Normalement, à la création d'un fichier donnée, son groupe propriétaire sera le groupe principal de son utilisateur propriétaire. Le SGID sur un répertoire donné affectera le groupe propriétaire de ce dernier comme groupe propriétaire de tous les fichiers qui seront créés dedans

## Caractéristiques

- Activation chmod g+s répertoire
- Numériquement il est représenté par 2000

# Les droits spéciaux

## Le sticky bit

Le sticky bit empêche à tous les utilisateurs d'écrire sur des fichiers qui ne leur appartiennent pas même s'ils se trouvent sur des répertoires sur lesquels ils ont droit d'écrire.

## Caractéristiques

- Il est idéal pour les répertoires à usage public (/tmp)
- Activation chmod +t répertoire
- Numériquement il est représenté par 1000

# Plan

- 1 Gestion des utilisateurs
- 2 Gestion des groupes
- 3 Gestion de permissions
- 4 Gestion des ACLs

# Présentation

## Limitation de la gestion classique des permissions

La gestion classique des permissions est suffisante pour les situations simples. Qu'en est-t-il lorsqu'on a plusieurs utilisateurs qui doivent accéder à une même ressource avec les mêmes droits mais ils n'ont pas de groupe commun?

## Définition

Les ACLs permettent d'autoriser un utilisateur ou un groupe tiers (qui ne sont pas propriétaires) avec des permissions spécifiques.

# Présentation

## Activation des ACLs sur une partition

L'utilisation des ACLs nécessite l'ajout de l'option de montage "acl" sur la partition concernée. Il faut alors:

- Soit monter la partition avec l'option acl
- Soit l'indiquer sur le fichier /etc/fstab et remonter la partition

## Remarque

Si la partition est de type ext4, les ACLs sont activés par défaut. Sinon, si on voudrait les désactiver, nous avons l'option "noacl"

# Présentation

## Terminologie

- Le masque d'une ACL : Le masque ACL précise si les ACLs doivent être prises en compte ou pas sur un fichier donné
- Les ACL par défaut : Pour ne pas définir les ACLs fichier par fichier, une ACL par défaut est une ACL qu'on définit sur un répertoire et tout fichier créé dedans héritera ces ACLs

# Les commandes de gestion

## La commande setfacl

La syntaxe de cette commande

- `setfacl -m ACL[,...] fichier ...`

Une ACL est définie comme suivant

- `<type_d_ACL>:[<valeur>]:<droits>`
- L'option `-d` permet de définir les ACLs par défaut (pour les répertoires)

## Exemples

- `user:pierre:rw-` : Pierre a les droits de lecture et d'écriture (r et w)
- `group:g1:r—` : Le groupe g1 a le droit de lecture (r)
- `mask::rwx` : Les ACL sont totalement pris en compte
- `mask::—` : Les ACL sont désactivés
- `mask::r-` : Seul le droit de lecture est pris en compte dans les ACL
- `default:user:paul:rw-` : Exemple d'ACL par défaut (pour un répertoire)

# Les commandes de gestion

## La commande getfacl

La syntaxe de cette commande

- `getfacl nom_fichier`
- l'option `--omit-header` n'affiche pas les trois premières lignes

## Exemples

```
$ getfacl fichier
# file: fichier
# owner: utilisateur
# group: utilisateur
user::rwx
user:utilisateur1:rw-
user:utilisateur2:r-
group::r-
mask::rwx
other::—
```