CHAPITRE 2: Gestion locale des utilisateurs

1. Principe

Du point de vue du système, un utilisateur (**user**) est une entité autorisée à utiliser toutes ou parties des ressources matérielles et logicielles de la machine par le biais du système d'exploitation.

Les systèmes d'exploitations multi-utilisateurs nécessitent une authentification préalable de l'utilisateur qui veut utiliser toutes ou parties des ressources matérielles et logicielles de la machine.

De plus, ils doivent fournir des mécanismes de contrôle d'accès à ces ressources, pour que chaque utilisateur puisse protéger ses données et ses ressources privées.

La gestion locale des utilisateurs consiste à mettre en place, pour chaque utilisateur, un espace privé de stockage de données dans le(s) système(s) de fichiers et à configurer les dispositifs d' authentification et de contrôle de ces utilisateurs.

Les concepts présentés dans ce chapitre sont hérités du système UNIX et sont applicables au système Linux.

La gestion **centralisée** des comptes des utilisateurs sur des machines en réseau **TCP/**IP, en utilisant le service **NIS** (Network Information Service) sera présentée ultérieurement dans le module optionnel **ASR1** du semestre **S4** pour la formation initiale ou dans le module **ASR-Adr** pour l'Année Spéciale.
Sa mise en œuvre nécessite de maîtriser les concepts présentés dans ce chapitre.

2. Attributs d'un utilisateur

Chaque utilisateur est caractérisé par des attributs dont les principaux sont :

- un numéro d'identification **UID** (User IDentification ≥ 0)
- un numéro d'identification de groupe **GID** (Group IDentification ≥ 0)
- un nom symbolique (**login name**)
- un mot de passe (**password**)
- un répertoire privé (**home directory**) appelé aussi répertoire de connexion
- un nom de programme exécutable qui sera lancé lors de la connexion (un **shell** par exemple)

Un utilisateur peut être une personne ou un groupe de personnes utilisant le même **login name**.
Une personne ou groupe de personnes peut posséder plusieurs **login name**.

Certains utilisateurs spéciaux sont créés automatiquement lors de la phase d'installation du système pour exécuter des tâches d'administration ou pour sécuriser certaines ressources.

Ces utilisateurs ne correspondent pas forcement à un individu en particulier (ex : **root**, **lp**, …) ou sont des entités virtuelles (ex : **sshd**, **telnetd**, **www-data**, ….).

3. Fichiers de configuration

Les informations utilisées pour la gestion **locale** des utilisateurs sont stockées dans des fichiers du répertoire/**etc**.
Les principaux fichiers de configuration sont :

- le fichier **`/etc/group`** qui contient la liste des groupes existants
- le fichier **`/etc/passwd`** qui contient la liste des utilisateurs existants
- le fichier **`/etc/shadow`** qui contient la liste des mots de passe **cryptés** des utilisateurs existants
- le fichier **`/etc/gshadow`** qui contient la liste des mots de passe **cryptés** des groupes existants

3.1. Le fichier /etc/group

Pour sécuriser l'accès aux objets du système de fichiers, le système distingue 3 classes d'utilisateurs :

- la classe du **propriétaire** (**user**), qui correspond à l'utilisateur qui a créé le fichier ou qui en a hérité
- la classe des utilisateurs appartenant à un **groupe** existant (**group**)
- la classe des **autres utilisateurs** (**others**) qui ne font pas partie de 2 classes précédentes

Lors de la création d'un compte utilisateur, on doit obligatoirement inscrire cet utilisateur dans un groupe existant qui est appelé **groupe principal** (primary group) de cet utilisateur.

Par la suite, l'administrateur en étant **root**, peut inscrive cet utilisateur dans un ou plusieurs autres groupes existants, qui seront appelés **groupes supplémentaires** (secondary group) de cet utilisateur.
Ceci permettra à cet utilisateur d'accéder aux objets du système dont ces groupes sont propriétaires.

Le fichier **`/etc/group`** est un fichier texte ASCII dont chaque ligne contient la définition d'un groupe.
Une ligne est structurée en enregistrement composé de **4 champs** séparés par "**:**" et contenant :

- le nom du groupe
- le mot de passe crypté du groupe ou rien dans les anciennes versions d'UNIX.
  Ce champ est en général vide car on utilise très rarement un mot de passe crypté pour un groupe.
  dans les versions récentes d'UNIX/Linux, ce champ contient un "**x**" . Si un mot de passe crypté pour ce groupe est utilisé, ce qui est rarement utilisé, il est stocké dans le fichier **`/etc/gshadow`** qui n'est accessible que par **root** car le fichier **`/etc/group`** est lisible par tout utilisateur.
- le **GID** qui est un entier compris entre **0** et **65535**
  le GID **0** est réservé au groupe **root** auquel appartient le super-utilisateur **root**.
- la liste des utilisateurs appartenant au groupe
  Il n'est pas nécessaire d'inscrire les utilisateurs dont ce groupe est le **groupe principal** car leur GID est stocké dans le fichier **`/etc/passwd`**. Seuls les utilisateurs pour lesquels ce groupe est un **groupe supplémentaire** doivent obligatoirement être inscrits.

Attention :

- ce fichier ne doit pas contenir de lignes vides
- si une ligne comporte une erreur, toutes les lignes suivantes seront ignorées par le système
- chaque groupe doit avoir un nom et un GID unique

Exemple : extrait d'un fichier /etc/group

root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:
dialout:x:20:
cdrom:x:24:
floppy:x:25:
tape:x:26:
sudo:x:27:
audio:x:29:
www-data:x:33:

……

3.2. Le fichier /etc/passwd

Le fichier **`/etc/passwd`** est un fichier texte ASCII dont chaque ligne contient la définition d'un utilisateur.
Une ligne est structurée en enregistrement composé de **7 champs** séparés par "**:**" et contenant :

- le nom de connexion (login) de l'utilisateur
- le mot de passe **crypté** de l'utilisateur dans les anciennes versions d'UNIX
  dans les versions récentes d'UNIX/Linux, ce champ contient un "**x**" . Pour parer aux attaques par dictionnaire (cracking), le mot de passe crypté est stocké dans le fichier **`/etc/shadow`** qui n'est accessible que par **root** car le fichier **`/etc/passwd`** est lisible par tout utilisateur.
- l'**UID** de l'utilisateur qui est un entier compris entre **0** et **65535**
  l'UID 0 est réservé au super-utilisateur **root**
- le **GID** du groupe **principal** de l'utilisateur
- le nom propre, la fonction de l'utilisateur ou un commentaire ou rien
- le répertoire de connexion de l'utilisateur
- la commande à exécuter à l'ouverture de la session (un **shell** en général)

Attention :

- ce fichier ne doit pas contenir de lignes vides
- si une ligne comporte une erreur, toutes les lignes suivantes seront ignorées par le système
- chaque utilisateur doit avoir un nom de login et un UID unique

Exemple : extrait d'un fichier /etc/passwd

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
Debian-exim:x:102:102::/var/spool/exim4:/bin/false
identd:x:100:65534::/var/run/identd:/bin/false
sshd:x:101:65534::/var/run/sshd:/bin/false
telnetd:x:104:104::/nonexistent:/bin/false
ftp:x:103:65534::/home/ftp:/bin/false

…….

3.3. Le fichier /etc/shadow

Le fichier **`/etc/shadow`** est un fichier texte ASCII dont chaque ligne contient les caractéristiques du mot de passe d'un utilisateur.
Une ligne est structurée en enregistrement composé de **7 champs** séparés par "**:**" et contenant :

- le nom de connexion (login) de l'utilisateur
- le mot de passe crypté de l'utilisateur. C'est une chaîne de 13 à 24 caractères ASCII en fonction de la méthode de cryptage utilisée (md5,). Si la chaîne n'est pas le résultat d'une méthode de cryptage, l'utilisateur ne pourra pas se connecter. Les caractères génériques "**!**" et "*****" des shell ne peuvent pas figurer dans la chaîne. Pour bloquer un compte, il suffit en étant **root**, de placer un "**!**" ou une "*****" dans ce champ.
- le nombre de jours écoulés depuis le 1er janvier 1970 jusqu'au dernier changement de mot de passe
- le nombre de jours à attendre avant de pouvoir changer le mot de passe
- le nombre de jours après lesquels le mot de passe doit être changé
- le nombre de jours avant la fin de validité du mot de passe impliquant l'avertissement de l'utilisateur
- le nombre de jours après la fin de validité provoquant la désactivation du compte
- le numéro du jour depuis le 1er janvier 1970 à partir duquel le compte est désactivé
- champ réservé

Attention :

Mêmes remarques que pour le fichier /etc/group

3.4. Le fichier /etc/gshadow

Le fichier **/etc/gshadow** est un fichier texte ASCII dont chaque ligne contient les informations de sécurité d'un groupe existant dans le fichier **/etc/group.**
Une ligne est structurée en enregistrement composé de **4 champs** séparés par "**:**" et contenant :

- le nom du groupe
- le mot de passe crypté du groupe ou un des caractères "**!**" ou "*****" si on n'utilise pas de mot de passe.
- la liste des utilisateurs autorisés à administrer les permissions du groupe avec la commande **gpasswd**.
  Seul **root** peut ajouter ou supprimer des logins dans cette liste. Comme le mot de passe crypté pour un groupe est rarement utilisé, ce champ est presque toujours vide.

- la liste des utilisateurs pour lesquels ce groupe est un **groupe supplémentaire**.
  Cette liste est identique à celle du 4ème champ du fichier **/etc/group**.

Le fichier **/etc/gshadow** est utilisé par la commande **newgrp** pour autoriser ou interdire à un utilisateur (autre que **root**) à changer de groupe pendant sa session.
Si ce groupe est le groupe principal ou un des groupes supplémentaires de l'utilisateur, il sera autorisé à appartenir à ce groupe en donnant le mot de passe crypté du groupe s'il y en a un ou sans contrôle si le 2ème champ contient un des caractères "**!**" ou "*****".

4. Commandes d'administration

En étant root, on peut créer ou supprimer des groupes et des utilisateurs en éditant les 4 fichiers décrits au paragraphe précédent puis créer ou supprimer les répertoires de connexions de ces utilisateurs.
D'une part, cette méthode n'est utilisable que pour administrer un très petit nombre de comptes utilisateurs, d'autre part elle est très dangereuse car la moindre erreur de manipulation peut bloquer le système.

Les interfaces graphiques fenêtrées (KDE, GNOME, ...) fournissent des fonctions conviviales de gestion des comptes utilisateurs mais qui n'ont d'intérêt que pour administrer un très petit nombre de comptes.

Le système Linux fournit des commandes sécurisées qui contrôlent la cohérence des opérations et empêchent les erreurs de manipulation. Ces commandes utilisées dans des shell scripts permettent d'automatiser la gestion de très grand nombre de comptes :

- **groupadd** permet de créer un nouveau groupe
- **groupdel** permet de supprimer un groupe
- **groupmod** permet de modifier les attributs d'un groupe existant
- **useradd** permet de créer un nouveau compte utilisateur
- **userdel** permet de supprimer un compte utilisateur
- **usermod** permet de modifier les attributs d'un compte utilisateur existant
- **pwck** et **grpck** permettent de vérifier l'intégrité des fichiers **passwd, shadow, group** et **gshadow**

Exemples :

- **`groupadd -g 2000 adminS2a`** création du groupe **`adminS2a`** avec le GID **`2000`**
- **`groupdel adminS2a`** suppression du groupe **`adminS2a`**
- **`usermod -G hotplug jpense`** ajout de l'utilisateur **`jpense`** au groupe **`hotplug`**
- **`useradd -u 2010 -g adminS2a -m -d /home/jpense -s /bin/bash \ -p $pass jpense`**

  création du compte de l'utilisateur **`jpense`** du groupe **`gadminS2a`** avec l'UID **`2010`**
  répertoire de connexion **`/home/jpense ,`** shell de session **`/bin/bash`**
  mot de passe crypté préalablement stocké dans la variable **`$pass`** , l'option **`-m`** entraînera la copie des fichiers du répertoire **`/etc/skel`** dans le répertoire de connexion **`/home/jpense`**

- **`userdel -r jpense`** suppression du compte de l'utilisateur **`jpense`**
  l'option **`-r`** supprime le répertoire de connexion et la boite à lettre locale

5. Fichiers de configuration complémentaires

5.1. Le répertoire /etc/skel/

Lors de la création du compte d'un utilisateur, il est nécessaire de copier dans son répertoire de connexion des fichiers utilisés par le shell pour gérer sa session de travail après sa connexion.

Par exemple, si le shell de session est **`/bin/bash`**, il faut copier dans son répertoire de connexion, les fichiers de base **`.bash_profile`** et **`.bashrc`** qu'il pourra par la suite personnaliser à sa convenance.

La commande **`useradd`** copiera automatiquement dans le répertoire de connexion de l'utilisateur, tous les fichiers contenus dans le répertoire **`/etc/skel/`** si l'option **`-m`** est utilisée.

5.2. Le fichier /etc/login.defs

Ce fichier contient des valeurs par défaut qui sont utilisées par les commandes de configuration des comptes lorsque certains paramètres ne sont pas précisés.

On y trouve par exemple les valeurs par défaut concernant la durée de vie du mot de passe dans le fichier **`/etc/shadow`**, mais également les valeurs minimales et maximales pour l'UID et le GID ou un booléen qui précise si le répertoire de connexion doit être créé ou pas.

6. Gestion des mots de passe

6.1. Changer un mot de passe et verrouiller un compte

La commande **passwd** sans arguments permet interactivement à un utilisateur de changer son mot de passe après avoir saisi son mot de passe actuel.

Exécutée par **root**, elle permet de fixer la politique d'authentification d'un utilisateur par mot de passe.
Les principales options sont :

- **passwd *login*** permet interactivement d'affecter un nouveau mot de passe à l' utilisateur ***login*** la plupart du temps parce qu'il à oublié le sien
- **passwd -l *login*** permet de verrouiller (**l**ock) le compte de l' utilisateur ***login*** qui désormais ne pourra plus se connecter
- **passwd -u *login*** permet de déverrouiller (**u**nlock)le compte de l' utilisateur ***login***
- **passwd -d *login*** permet à l' utilisateur ***login*** de se connecter sans mot de passe (à n'utiliser que occasionnellement pour des tests ou du dépannage !)

6.2. Changer la duré de validité du mot de passe

Le fichier **/etc/shadow** contient des paramètres qui définissent la duré de validité du mot de passe.
La commande **chage** exécutée par **root**, permet de modifier ces paramètres en fonction des options utilisées.

7. Protection du compte root

Le stockage du mot de passe de **root** dans le fichier **/etc/shadow** est un premier niveau de sécurité mais qui est insuffisant lorsque la machine est accessible sur le réseau local ou pire sur le réseau public.
Si un pirate parvenait à "cracker" le mot de passe de **root** (c'est tout à fait faisable avec les moyens actuels) il pourrait depuis sa machine ouvrir une session sous **root** sur notre machine et en prendre l'entier contrôle.

La solution est d'interdire la connexion en **root** par le réseau depuis tout autre machine. Dans ce cas il faut être sur la console de la machine pour ouvrir une session sous **root**.
Cette solution est très contraignante car l'administrateur lui même ne peut plus administrer la machine à distance.

L'utilisation des services réseaux sécurisés tels que **ssh** permettent de lever cette contrainte en autorisant une connexion par **ssh** à un utilisateur privilégié défini par l'administrateur mais différent de **root**.
Une fois connecté avec le login de cet utilisateur privilégié, l'administrateur qui connaît le mot de passe de **root**, pourra avec la commande **su** (substitute user) devenir **root**.

Sous Linux, il existe un module nommé **PAM** (Pluggables Authentification Modules) qui permet de contrôler ces différents types d'accès.

Les fichiers de configuration des services d'authentification gérés par **PAM** sont archivés dans **/etc/pam.d**.
Par exemple :

- le fichier **/etc/pam.d/ssh** définit quels sont les utilisateurs autorisés à se connecter par **ssh**
- le fichier **/etc/pam.d/su** définit quels sont les utilisateurs autorisés à exécuter la commande **su**