TP 2
Le système Linux : Gestion des utilisateurs

Introduction à la gestion des utilisateurs

Le compte d'un utilisateur est représenté par le *login* et un *mot de passe* associé. Les informations sur les comptes utilisateurs disponibles sur une machine Unix sont regroupées dans le fichier `/etc/passwd`. Chaque ligne de ce fichier correspond à un compte. Une ligne est composée de 7 champs séparés par des `:`. Les champs sont les suivants :

Champs	***Contenu***
Uname	c'est le login
Mot de passe	Il s'agit du mot de passe crypté. Quand le système *shadow password* est utilisé ce champ contient le caractère `x`.
UID	Numéro **unique** d'identification de l'utilisateur.
GID	Numéro d'identification du groupe principal auquel appartient l'utilisateur
Gecos	Ce champ est de type informatif. Il est à la disposition de l'administrateur qui l'utilise habituellement pour indiquer le nom et le prénom de l'utilisateur.
Homedir	répertoire personnel de l'utilisateur
Login shell	Shell initial exécuté à la connexion.

Les informations concernant les groupes sont conservés dans le fichier */etc/group.* Ce fichier contient une entrée par groupe. Chaque entrée est composée de 4 champs:

Champs	***Contenu***
Gname	Nom du groupe
Mot de passe	Rarement utilisé
GID	Identificateur du groupe
Liste des membres	c'est la liste des login des utilisateurs membres du groupe. Les utilisateurs dont c'est le groupe principale n'ont pas besoin d'apparaître dans cette liste.

Bien que les mots de passes soient sauvegardés d'une manière cryptée dans les fichiers `/etc/passwd` et `/etc/group`, des problèmes de sécurité se sont posés. En effet, ces deux fichiers doivent être en mode lecture pour tout le monde, ce qui implique que tous les utilisateurs peuvent avoir accès aux mots de passes, cryptés certes, des autres. Si le cryptage empêche une lisibilité directe, son efficacité n'est pas sûre à 100%. Il existe aujourd'hui beaucoup des logiciels pour décoder les mots de passes cryptés. Seul des mots de passes longs et complexes nécessiteront

un décryptage beaucoup plus long. La solution proposée par Linux (et d'autres système Unix) est de stocker les mots des passes dans un fichier où seul l'administrateur système a le droit de lire et de le modifier. Sous Linux, les mots de passe des utilisateurs sont stockés dans */etc/shadow*. et ceux des groupes dans */etc/gshadow*. En plus de sauvegarde des mots de passe, Linux offre avec le fichier *shadow* la possibilité de gérer un système de vieillissement des mots de passe. Autrement dit, l'administrateur peut fixer une age limite des mots de passe après laquelle l'utilisateur est invité à changer son mot de passe sous peine de ne plus avoir accès à son compte.

Une ligne dans le fichier */etc/shadow* est composé de neuf champs séparés par le caractère `:`.

Champ	***Contenu***
Le login	
Mot de passe	Une * dans ce champ indique le compte ne peut être connecté (cas du compte bin par exemple). Un mot de passe commençant par !! indique que le compte est verrouillé.
Age	Le nombre de jour écoulé depuis le 1er janvier 1970 et la date de mise à jour du mot de passe.
Période de changement	Le nombre minimum de jours entre deux changement de mots de passe. Un 0 indique que l'utilisateur peut changer le mot de passe à n'importe quel moment.
Durée de validité	Le nombre maximum de jours pendant lesquels le mots de passe est valide. Le valeur 99999 indique que le mot de passe est toujours valide.
Durée de validité restant	Nombre de jours avant l'expiration.
Durée d'invalidation	Nombre de jour après l'expiration provoquant la désactivation du compte. Un champ vide indique qu'il n'y a aucune désactivation
Date d'expiration	Exprimée en nombre de jour depuis la date de référence (1/1/70)
Champs réservé	

Exercice 1. En utilisant l'éditeur vi., modifier les fichiers nécessaires afin d'ajouter deux groupes : *gtr1* et un groupe correspondant à votre groupe de TP.

Une méthode naïve pour l'ajout des utilisateurs consiste à éditer les fichiers `/etc/passwd`. Et */etc/shadow.* Selon cette méthode l'ajout d'un utilisateur passe par les étapes suivantes :

1. Ajout d'une entrée (ligne) correspondant à l'utilisateur dans le fichier `/etc/passwd`. et une autre dans `/etc/shadow`
2. Positionnement du mot de passe initial
3. Création du répertoire personnel de l'utilisateur
4. Changement de propriétaire pour ce répertoire
5. Copie des fichiers d'environnement (les fichiers contenus dans `/etc/skel`).

Exercice 2. Appliquer la méthode naïve décrite ci-haut pour créer deux comptes pour les membres du binôme. Essayer d'accéder à la machine en fournissant le *login* de l'utilisateur après chaque étape (donc essayer 5 fois) et noter les messages rendus par le système. Préciser dans votre compte-rendu les modifications apportées aux fichiers de configuration.
Exercice 3. L'administrateur peut-il connaître le mot de passe d'un utilisateur ? Peut-il le changer ?
Exercice 4. Que fait la commande `id` ?
Exercice 5. Donner les étapes nécessaires fermer le compte d'un utilisateur.

Commandes de gestion des utilisateurs

Le système Linux offre un ensemble de commandes qui permettent de faciliter la gestion des utilisateurs. Les principales commandes sont :

- `groupadd` : pour créer un nouveau groupes
- `groupdel` : pour détruire un groupe.
- `useradd` : pour créer un nouvel utilisateur
- `userdel` : pour effacer un utilisateur

Exercice 6. Consulter les pages de manuels de ces commandes et utiliser les pour créer un nouveau compte d'un utilisateur puis le détruire.

Gestion des droit d'accès

(Rappel) La commande `chmod` permet de changer les droits d'accès sur un fichier ou un répertoire. Pour chaque fichier on peut attribuer trois types de droits : lecture (r), écriture (w) et exécution (x) et ceci pour trois groupes d'acteurs : l'utilisateur propriétaire, les utilisateurs du même groupe que le propriétaire puis pour tous les autres utilisateurs.

Deux méthodes sont possibles pour positionner les droits sur un fichier (ou répertoire) : 1)en calculant le masque octal associé au fichier ou 2) par l'utilisation de la notation symbolique.

- le maque octal est constitué de trois chiffres octals ; un chiffre pour désigner les droits associés au (dans l'ordre) le propriétaire, le groupe et les autres utilisateurs. Un chiffre octale s'exprime sur 3 bits qui représente (dans l'ordre) les droits *read*, *wrire* et *execute* associé à un acteur. Ainsi la commande `chmod 400 file` donne le droit de lecture au propriétaire seulement.
- La notation symbolique est de la forme `chmod` *`CibleOperationDroit`* `file` où `Cible` désigne l'acteur cible (`u` pour me propriétaire, `g` pour le groupe, `o` pour les autres et `a` pour tous).Il est possible de combiner les plusieurs cible en même commande. `L'opération` peut être une des trois suivantes : + pour ajouter – pour retirer et = pour installer exactement les droits désignés. `Les droits` sont désigné par les symboles `r, w, et x.` Ainsi la commande `chmod u=r file`. permet de préciser que le propriétaire a le droit de lecture seulement sur file.

Exercice 7. Connectez-vous au système comme un un des utilisateur créé précédément. et créer un fichier vide nommé file dans le répertoire de connexion.
a) Quels sont les droits des autres sur ce fichier ?
b) Modifier les droits sur files de sorte à permette au groupe de le lire et le modifier. Les autres utilisateurs n'ont aucun droit sur ce fichier. (Donner les deux versions des commandes : masque octal et notation symbolique).
c) Appliquer la commande les séries des commandes suivantes et noter la différence :
Série 1) chmod 200 file ; chmod u=r file
Série 2) chmod 200 file ; chmod u+r file
Série 3) chmod 220 file , chmod u=r file
ci) Donner au propriétaire le droits de lire et écrire le fichier, au groupe le droit de lire t l'exécuter et aux autres le droit de l'exécuter.
cii) Quel est le rôle du droit d'exécution sur un répertoire ? Donner un scénario d'exemple qui illustre droit.

Les droits d'accès peuvent être gérés de deux manières complémentaires : *a priori* ou *a posteriori*. Le commande *chmod* s'applique sur des fichiers existants. Il peut devenir fastidieux de changer systématiquement les droits initiaux quand ils ne sont pas conformes aux droits souhaités. La commande *umask* résout ce problème. La commande *umask* ne change pas les droits sur les fichiers existants mais sur les fichier à créer. Cette commande permet de fixer un filtre (souvent exprimé en trois chiffres (en base 8) à l'instar de la commande `chmod`) qui permet de changer les droits demandées par le logiciel qui crée un nouveau fichier. Dans le filtre un bit à 0 laisse passer le droit demandé et un 1 l'inhibe. Ainsi les résultats des droits est fonction du filtre et les droit demandés.

Exercice 8 Utiliser la commande `umask` pour donner le droit de lecture seulement aux membres de votre groupe (linux) et aucun droit aux autres.

A la connexion, un utilisateur est identifié par son UID et le GID de son groupe principal qui devient le groupe actif durant la session. L'utilisateur peut changer temporairement son groupe en exécutant la commande : `newgrp`. Le changement de groupe ne modifie pas les droits de l'utilisateur sur les fichiers. Seules les informations que l'utilisateur donnera sur le nouveau fichier qui va créer seront modifiées.

Exercice 9. Consulter les pages de manuels de la commande `newgrp` et donner un exemple de son utilisation. Comment vérifier que le groupe actif de l'utilisateur a bien changer ?