TP Unix – Gestion des Users

Objectif : Mettre à disposition un environnement à des utilisateurs, gérer les accès aux fichiers, les droits, l' environnement de travail, gérer les groupes..

Contexte : pour cet exercice, on imagine une entreprise dont deux services sont connectés au réseau : le service ***Commercial****, et le service* ***Comptable****. L'ambiance au sein du service de comptabilité est assez conviviale, et bien que toutes les informations traitées soient absolument secrètes (et donc interdites en accès à toute autre personne qu'un comptable), ces personnes ont l'habitude de partager leurs données. Concernant les commerciaux, au contraire, tous leurs travaux sont absolument secrets (échanges commerciaux, ristournes et remise, chiffre d'affaires, arrangements divers...)*

Pré-requis :

Documentation et exercice de Jean Gourdin :

http://www.linux-kheops.com/doc/cours/jgourdin/comptes-utilisateurs.html

Documentation technique :

la séquence de log : Lorsqu'un utilisateur se loggue, tout est géré par le programme login : Ce programme va obéir aux PAM (Pluggable Authentification Module), qui vont lui indiquer une suite d'étapes à valider. Le fichier **/etc/passwd** sera certainement lu (si on utilise une authentification standard et locale), et diverses informations concernant l'utilisateur seront ainsi récupérées : Son nom, son home, son programme de connexion.)

Exercice :
Affichez le contenu de /etc/passwd.
Quel est l'UID de root ? Quel est son shell de connexion ?
Qui est 1000 ?
Affichez le contenu de /etc/group
Combien y a t'il de groupes ? Qui est le 101 ?
Affichez le contenu de /etc/shadow
Que contient il ? Quel est le password du root ?
Quels sont les droits de ces 3 fichiers ?
Pourquoi ?
Les droits sur /etc/passwd semblent un peu larges. Supprimez l'accès en

lecture à tout le monde. Essayez de vous connecter en tant que simple utilisateur. Si cela fonctionne, listez ensuite le détail (ls -l) du contenu de votre home. Que remarquez vous ?
Remettez les droits normaux à /etc/passwd.
Vérifiez avec le simple utilisateur la différence

Le contenu du fichier **/etc/profile** est exécuté (réglages génériques pour tous les utilisateurs). Puis ensuite, pour une connexion distant, un sous-shell, ou un xterm, alors **/etc/bash.bashrc** puis ~/**.bashrc** seront exécutés

Exercices :
Décodez le contenu de ces fichiers sur votre machine.
Expliquez pourquoi Debian n'offre pas la visualisation en couleurs des fichiers (ls). Dans votre bash, lancez un nouveau bash (en tapant simplement bash). Testez le ls.. Que se passe t il ?

Amélioration du bash :

Peut être avez vous remarqué le fichier **/etc/bash_completion.**
Vous connaissez la complétion, mais celle ci est encore supérieure : connectez-vous sur deux sessions (afin de comparer la différence), et sur l'une d'elles, activez la complétion étendue :
tapez : **. /etc/bash_completion**
Attention au point tout seul : très important pour que le script demandé s'exécute dans l'environnement courant, et non dans un fils)

Testez la complétion étendue avec cd (tab), avec des commandes (aptitude i(tab)), ssh (tab), man ls(tab).
Que remarquez vous ?
Comment faire pour bénéficier de tout ceci ?
Consultez vos propres fichiers de connexion (.bash_profile, et .bashrc). Modifiez .bash_profile afin qu'il exécute .bashrc (ainsi, .bashrc est exécuté toujours, en shell de login ou autre). Puis, dans .bashrc, dé-commentez l'accès à /etc/bash_completion.
Testez...

/etc/skel (profil par défaut)

Extension de ces réglages à tous les nouveaux utilisateurs (ou 'du bon usage de **/etc/skel**')
Dans le répertoire **/etc/skel**, vous trouverez le squelette de tous les nouveaux comptes : Quel est le contenu de ce répertoire ?

Exercice : Modifiez le contenu de /etc/skel afin d'y déposer un .bashrc « aux petits oignons » (avec par exemple, un alias ll pour ls -al). Créez un nouvel

utilisateur (adduser toto). Connectez vous avec toto, et vérifiez ses réglages.

Imaginons maintenant que nous voulons que tous les nouveaux utilisateurs se retrouvent avec une structure de home standardisée (*par exemple, avec un fichier 'mode d'emploi du réseau' et un sous répertoire public_html déjà prêt pour la publication de pages web)* :

Créez ce répertoire et un texte 'mode_d_emploi_du_reseau.txt' dans le répertoire /etc/skel.
Créez deux nouveaux utilisateurs (foo et bar)
Observez leurs homes.

Gestion des droits : Vous avez certainement remarqué la commande umask. Cette commande définit les droits standards dont seront affublés vos fichiers. Les droits normaux sont 666 pour un fichier, et 777 pour un répertoire. **Umask** vient en soustraction pour le calcul des droits. L'emploi **d'umask** seul permet d'afficher la valeur d'umask, et umask XXXX permet de définir l'umask à XXXX.

Exercice
Définissez votre umask à 0000 : Créez des fichiers et des répertoires et vérifiez les droits obtenus. Définissez votre umask pour être le seul à pouvoir voir vos fichiers et répertoires... Vérifiez.

Ajout de comptes

Dans le contexte décrit, on peut proposer de résoudre le problème par la création de deux groupes (commerciaux et comptables). Il semble préférable de créer le home de chaque utilisateur dans /home/NomDuGroupe, dans un souci de bonne gestion.
Le comportement par défaut de création des comptes est piloté par **/etc/adduser.conf.**
Selon la taille de la population d'utilisateurs à gérer, on pourra modifier ce fichier pour adapter la gestion à nos besoins. Dans notre cas, on utilisera des groupes : On créera des groupes, et on créera des utilisateurs dans ces groupes.

Modification du fichier /etc/adduser.conf

Ce fichier est suffisamment documenté pour que vous puissiez vous débrouiller seul. On peut y définir le shell de connexion proposé par défaut, le nom du répertoire contenant les home directories, l'endroit des squelettes, etc...

Exercices :
Expliquez ce que sont les LETTERHOMES, le rôle des directives commençant

par FIRST. Comment faire pour que les homes soit créées dans un sous-répertoire de home portant le nom du groupe ? (tous les homes des comptables dans un sous répertoire /home/comptables)
L'ajout de groupes et d'utilisateurs se fait respectivement par les commandes addgroup et adduser. Consultez le man de ces commandes. Faites le réglage du adduser.conf correspondant, et testez l'ajout de deux groupes (testprofs et testetudiants), puis de quelques utilisateurs (profs et étudiants)
Testez ensuite le bon fonctionnement, en vous connectant en tant que certains de ces utilisateurs.
Supprimez ensuite tous ces utilisateurs, ainsi que leurs répertoires (man userdel)
Droits d'accès, et multigroupes

Les commandes **chmod, chown** et **chgrp** permettent d'attribuer, de modifier des droits sur les objets du file system (fichiers et répertoires)
(faire un **man**)
D'autre part, un utilisateur peut appartenir à plusieurs groupes (un groupe principal, et d'autres additionnels)
Cela permet un certain souplesse dans les droits, bien que seule l'utilisation des ACLs puisse permettre de tout gérer (au prix d'une dangereuse complexité).
Pour ajouter un utilisateur dans un groupe additionnel, utilisez **adduser user groupAdditionnel.**
Vous pourrez alors donner des droits à ce groupe, et l'OS évaluera les droits de chaque utilisateur par rapport à l'ensemble de ses groupes

Exercice :
Créez l'ensemble des comptes selon les régles définies en début de cet exercice : Les commerciaux (Bill, Bob, Carlos, Richard, Laura) et les comptables (Raymond, Georgette, Carlotta, Paula).
Ces utilisateurs peuvent avoir un site web (pensez à créer le public_html). Le système de gestion de courrier demande à avoir un répertoire MailDir dans chacun des homes.
Les chefs de services (Bill et Raymond) ont la possibilité d'alimenter le site web (création de pages...) tandis que les utilisateurs ne peuvent que consulter.

Les multiples réglages

Il est possible d'augmenter le niveau de sécurité, ou tout simplement de diversifier votre système en introduisant des comportements qui vous sont spécifiques, permettant ainsi de troubler tout attaquant. Il est possible par exemple de n'autoriser les connexions root que sur les écrans pairs. Dans ce cas, un attaquant disposant du bon mot de passe mais n'ayant pas l'information sur les écrans pairs peut tout à fait penser qu'il n'a pas récuperé

la bonne information lorsqu'il se fait interdire l'accès à un écran impair (pour peu que le message d'erreur soit sibyllin, comme « accès refusé »)

Exercices :
Interdisez les connexions root sur les 3 premiers écrans.
A quoi sert le fichier /etc/nologin ? Interdisez la machine à tout utilisateur en le prévenant qu'une maintenance est en cours. Ré-ouvrez ensuite l'accès. A quoi sert le script /etc/init.d/rmnologin ?
A quoi sert le fichier /etc/login.defs ? Modifiez ce fichier afin de raccourcir la durée pendant laquelle login attend le mot de passe d'un compte (5 secondes). Limiter la durée de vie d'un mot de passe (15 jours)
Lister les différents shell de connexion du fichier /etc/passwd
Quel est le rôle de cette information, et à quoi pourrait elle servir ?
Modifier le programme de connexion afin d'invalider l'accès d'un utilisateur. Vérifier. Pourquoi est-ce préférable à la suppression du compte (pensez à des stagiaires par exemple) ?
Les PAM : (Attention, c'est très sensible) : Lire tout d'abord http://www.debian.org/doc/manuals/debian-reference/ch04.fr.html. Ensuite, essayez de forcer les utilisateurs à utiliser un mot de passe de plus de 5 caractères. Débrouillez vous pour les utilisateurs d'un certain groupe puissent se connecter malgré la présence d'un fichier /etc/nologin.Interdisez la commande su à un groupe d'utilisateurs.

Les limites du système

Ce système de droits à des limites. On peut vouloir utiliser un système ou les combinaisons sont infinies, malgré le danger que cela représente (erreurs de conceptions des accès)... On peut même parfois y être obligé : (par exemple, dans nos classes d'ESIPE : On veut que tous les étudiants aient un home distincts, mais on veut que chaque binôme dispose d'un répertoire commun sous /var/www, afin d'y déposer ses pages web et php. Ici, la seule solution, peu efficace, serait de créer autant de groupes qu'il y a de binôme, et de donner à chacun de ses groupes le droit de lecture écriture sur le répertoire qui leur correspond)

Exercice :
Recherchez des informations sur l'installation des ACLs sous Linux
Débrouillez vous pour installer les ACLS sur votre machine.
Créez ensuite les comptes de tous les étudiants présents dans votre salle, qui appartiendront tous à un seul groupe : ESIPE.
Créez un répertoire pour chaque binôme repéré dans /var/www (par exemple, /var/www/B1 pour le binôme1, /var/www/B2 pour le binôme 2, etc)
Donnez les droits afin que seuls ces deux personnes puissent lire et écrire, que le groupe www-data puisse lire, et que personne d'autre ne puisse y

aller
Ajouter le droit de lecteur à tous les membres du groupe profs.

Scripting : Écrire un script qui prend en argument 2 noms et un numéro de binôme, et qui créé les comptes pour ses deux personnes (dans le groupe ESIPE), crée le répertoire pour le binome dans /var/www, et enfin donne les droits sur ce répertoire.

A explorer : les quotas Disk !!