Serval-Concept

Apprenez autrement

Linux Security

Votre formateur
Beirnaert Jacques
Consultant Sécurité
Serval-Concept
E-mail : jacques.beirnaert@serval-concept.com

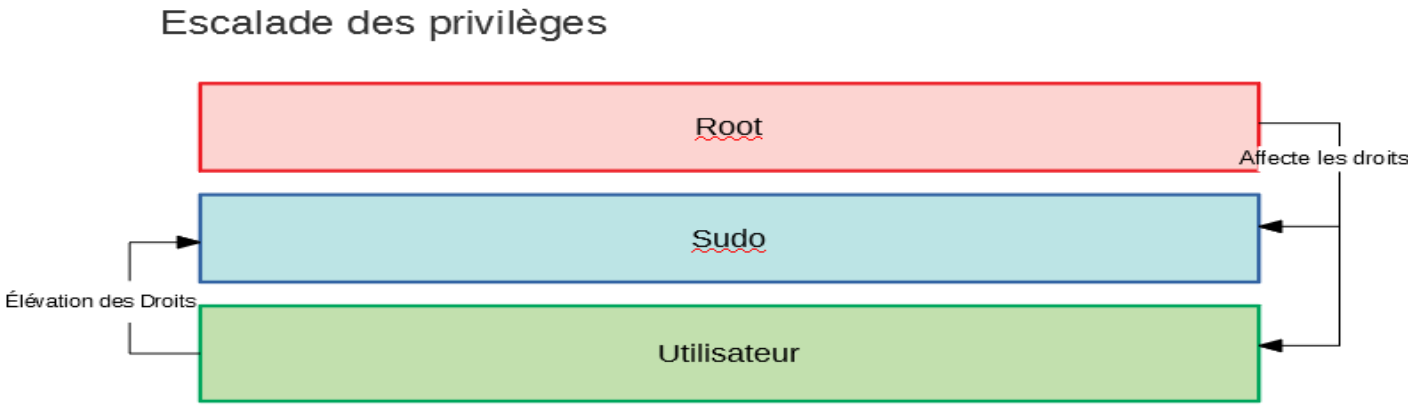
www.serval-concept.com

Les droits et permissions

Les droits et permissions

- Sur Linux on peut déterminer 3 niveaux hiérarchiques :
  - Utilisateur : c'est le plus bas niveau, il ne possède aucun droit pouvant mettre en danger le système. Il se contente d'utiliser sa machine pour les tâches métiers.
  - Sudo : c'est un utilisateur avec des droits supérieurs permettant d'agir de façon limitée sur le système.
  - Root : c'est le chef du système. Il est capable de manipuler entièrement le système, de faire toutes les actions pouvant modifier le fonctionnement de la machine et ses composants.

Les droits et permissions



Les droits et permissions

- Repérer le mode :
  - En utilisateur simple
    - user@debian:~$
    - Remarque : Pour passer du user en root : su + mot de passe du root
  - En root
    - root@debian:/home/user#

Les droits et permissions

- Unix est un système multi-utilisateurs. Les utilisateurs y sont rassemblés par groupes. Chaque utilisateur est donc identifié dans le système par :
  - Son login = numéro unique : l'uid
  - Son groupe = numéro unique : le gid

- Le système gère la correspondance entre identifiants symbolique et numérique via des fichiers textes :
  - Login et uid via le fichier /etc/passwd
  - Groupe et gid via le fichier /etc/group

Les droits et permissions

- Chaque fichier :
  - Appartient à un utilisateur (son propriétaire) et à un groupe
    - La modification se fait avec la commande chown
    - La syntaxe est la suivante :
      - chown user:group

  - Possède des droits d'utilisation applicables :
    - A son propriétaire (u : user)
    - Aux utilisateurs du groupe (g : group)
    - Aux utilisateurs n'appartenant pas à son groupe (o : other)

Les droits et permissions

- Pour chacune des catégories, il existe trois types de droits :
  - Lecture : autoriser la lecture du contenu
  - Écriture : autoriser la modification du contenu
  - Exécution : autoriser l'exécution du contenu

- L'option -l de ls permet de voir les droits d'accès d'un fichier. Pour chacun des trois cas d'application les droits sont affichés par une chaîne de caractères avec la représentation suivante :
  - r : read , lecture
  - w : write, écriture
  - x : execute , exécution
  - - : indique qu'il n'y a pas de droit attribué.

Les droits et permissions

- La modification se fait avec la commande chmod en mode root.
- La syntaxe est la suivante :
  - chmod <mode> <fichier>
- Il y a deux façons pour réaliser le changement :
  - Sous la forme symbolique
  - Sous la forme numérique

Les droits et permissions

- Chmod (forme symbolique)
  La syntaxe est la suivante chmod <personne><action><accès> fichier

<personne>		<action>		<accès>	
u	propriétaire	+	ajouter	r	lecture
g	groupe	-	enlever	w	écriture
o	autres	=	initialiser	x	exécution
a	tous				

  - Par exemple : chmod a+r fichier.txt

Les droits et permissions

- Chmod (forme numérique)
  La syntaxe est la suivante chmod <octal> fichier

symbolique	binaire	Octal
---	000	0
--x	001	1
-w-	010	2
-wx	011	3
r--	100	4
r-x	101	5
rw-	110	6
rwx	111	7

  - Par exemple : chmod 700 fichier.txt

Les droits et permissions

- L ‘attribution des droits est une chose essentielle en terme de sécurité.
- Il faut être bien conscient des droits que l’on affecte aux différents utilisateurs ainsi que l’impact occasionné.
- Il est interdit de donner les droits 777 à un fichier.

Les droits et permissions

- Pour les fichiers :
  - Quand un fichier est exécutable par son propriétaire, il peut de plus être setuid. Cela signifie que lorsqu'il est exécuté, il l'est avec les droits de son propriétaire.
    - Par exemple : passwd
  - De la même façon, un exécutable peut être setgid, et s'exécuter avec les droits du groupe auquel il appartient.
  - Enfin, un exécutable peut être "sticky": cela signifie qu'il reste en mémoire même après la fin de son exécution, pour pouvoir être relancé plus rapidement.

- Pour les dossiers :
  - Pas de setuid
  - Quand un répertoire est setgid, tous les fichiers créés dans ce répertoire appartiennent au même groupe que le répertoire.
  - En positionnant le sticky bit sur un répertoire un utilisateur ne peut effacer que les fichier qui lui appartiennent.

Les droits et permissions

- Dans la présentation des droits en texte, ces droits particuliers seront représentés comme suit:
  - pour le suid: un « s » à la place du « x » du propriétaire comme dans « rwsr-xr-x ».
  - pour le sgid: un « s » à la place du « x » du groupe comme dans « rwxr-sr-x ».
  - pour le sticky bit: un « t » à la place du dernier « x » comme dans « drwxrwxrwt ».

    Remarque : s'il n'y avait pas de droit d'exécution « x » avant d'appliquer ces droits, les lettres « s » et « t » seront mises en majuscule

- Dans la présentation en nombre octal, ces droits correspondent à un 4ème chiffre octal situé à gauche, ce qui donnera, par exemple sur la base d'un « 755 » = « rwxr-xr-x »
  - « 4755 » pour « rwsr-xr-x » avec un suid
  - « 2755 » pour « rwxr-sr-x » avec un sgid
  - « 6755 » pour « rwsr-sr-x » avec un suid + un guid
  - « 1755 » pour « rwxr-xr-t » avec un stiky bit

Les droits et permissions

- Quand vous créer un fichier, par défaut celui-ci possède certains droits.
  - Ce sont 666 pour un fichier (-rw-rw-rw-) et 777 pour un répertoire (-rwxrwxrwx), ce sont les droits maximum.
  - Vous pouvez faire en sorte de changer ces paramètres par défaut. La commande umask est là pour ça.

- Pour un fichier :
  - Si vous tapez umask 022, vous partez des droits maximum 666 et vous retranchez 022, on obtient donc 644, par défaut les fichiers auront comme droit 644 (-rw-r-r--).
- Pour un répertoire :
  - Si vous tapez umask 022, vous partez des droits maximum 777 et vous retranchez 022, on obtient donc 755, par défaut les dossiers auront comme droit 755 (-rwxr-xr-x).

Les droits et permissions

- Par défaut, la gestion des droits sous Linux est relativement basique (bien que souvent suffisante).
- En effet, il n'est possible de permettre ou d'interdire la lecture, l'écriture et l'exécution de fichiers que pour trois catégories d'utilisateurs :
  - le propriétaire du fichier
  - le groupe auquel appartient le propriétaire
  - et le reste du monde.
- Cette gestion des droits permet de configurer les accès aux fichiers dans la plupart des situations simples mais peut s'avérer limitée, par exemple, dans un contexte où plusieurs utilisateurs doivent accéder à une ressource partagée alors qu'ils n'ont pas de groupe commun.
  - Les ACL permettent de pallier ce manque grâce à une gestion avancée des droits. Ainsi, il devient possible d'autoriser un utilisateur tiers à effectuer des opérations sur un fichier (dossier) sans autoriser tout un groupe ou tout le reste du monde.
- L'attribution des droits se fait grâce à la commande setfacl, la lecture des droits avec getfacl
  - Un fichier dont les ACL auront été spécifiés verra s'ajouter un +, visible avec la commande « ls -l fichier »

Travaux pratique

- Développer un script bash aidant l'administrateur à intégrer le nouveau personnel à l'aide d'un menu
  - Création d'utilisateur (non interactif) s'il n'est pas déjà présent
    - Création de leur arborescence personnel (Bureau, Documents, Images, …)
    - Import des documents (file0.txt, …, file100.txt)
  - Suppression d'utilisateurs (non interactif)
  - Ajout au service (administratif, comptable, informatique, employé)
  - Suppression du service
- Remarque :
  - Le personnel administratif aura accès au dossier/home/administratif et son contenu
  - Le personnel comptable aura accès au dossier /home/compta et son contenu
  - Le personnel informatique aura accès à tous les dossiers
  - Le dossier /home/public est accessible à tous les employés
  - L'utilisateur stagiaire aura l'accès au dossier /home/administratif/stagiaire mais pas /home/administratif
  - Les fichiers présents dans les dossiers partagés appartiendront au groupe
  - Les utilisateurs ne pourront supprimer que leurs propres documents