Administration Système Réseau Linux
Notes de cours

Gestion des utilisateurs

Présentation

Le système, dès son installation, avant même la première connexion au système, a créé des utilisateurs système. Un utilisateur n'est donc pas uniquement une personne physique. Le système a besoin d'utilisateurs pour sa gestion interne, notamment comme propriétaire des divers processus. La commande

```
$ ps aux | less
```

montre qu'avant toute connexion d'utilisateur humain, root a lancé init, et la plupart des services, crond, inetd, lpd, smbd, ... , avant de lancer les connexions utilisateurs dans les consoles, y compris éventuellement la sienne!

Les principales commandes

Utilisez les pages manuel de ces commandes pour avoir un descriptif complet

- Gestion des comptes utilisateur : useradd, usermod, userdel
- Gestion des groupes : groupadd, groupmod, groupdel
- Vérification des fichiers : pwck, grpck
- Changement du mot de passe d'un utilisateur : passwd
- utilitaires divers : chfn, id, groups, chsh

Gestion des comptes

Créer un compte pour un nouvel utilisateur

Cela signifie lui permettre d'être connu du poste local, s'y loguer, avoir un accès complet sur son répertoire personnel. Mais aussi dans une configuration réseau, de pouvoir se connecter à son compte par telnet ssh et ftp, et de pouvoir bénéficier de services réseau de partage distant.

Créer l'utilisateur toto

```
#useradd toto
```

Cette commande crée le répertoire personnel /home/toto, portant par défaut le nom du compte. une nouvelle entrée dans les 2 fichiers fondamentaux /etc/passwd et /etc/group.

Attribuer un mot de passe à l'utilisateur toto

```
#passwd toto
```

Saisir deux fois le mot de passe. Attention aucun caractère n'apparaît à l'écran.

Supprimer le compte utilisateur toto (non connecté)

```
# userdel -r toto
```

L'option -r supprime aussi le répertoire personnel et les fichiers de l'utilisateur La commande supprime toute trace de l'utilisateur dans le fichier de configuration : /etc/passwd y compris dans les groupes d'utilisateurs.

Modifier le compte utilisateur toto

```
# usermod [options] toto
```

Les options sont les mêmes que useradd.

Les groupes

Un groupe est pour Linux un ensemble d'utilisateurs qui partagent les mêmes fichiers et répertoires.

Chaque utilisateur doit faire partie au moins d'un groupe, son groupe primaire. Celui-ci est défini au moment de la création du compte, et par défaut, l'utilisateur appartient à un nouveau groupe créé, portant son nom.

Ainsi, dans /etc/passwd chaque utilisateur possède un groupe par défaut, précisé par son identifiant gid dans ce fichier. L'appartenance au groupe primaire n'étant pas exclusive, tout utilisateur peut faire partie de plusieurs autres groupes, appelés ses groupes secondaires. Mais le rôle joué par le groupe primaire demeure prépondérant.

Lister tous les groupes (primaire et secondaires) de l'utilisateur toto

```
# groups toto
```

Créer un nouveau groupe tata

```
# groupadd tata
```

Supprimer un groupe tata

```
# groupdel tata.
```

Le groupe est supprimé du fichier /etc/group.

Ajouter un utilisateur à un groupe

Le plus simple est d'éditer le fichier /etc/group et d'ajouter une liste d'utilisateurs (séparés par des virgules) sur la ligne du groupe.

Comment ça marche ?

Tout ce qui concerne la gestion et l'authentification des utilisateurs est inscrit dans un seul fichier /etc/passwd
La gestion des groupes est assurée par un fichier /etc/group
Les mots de passe cryptés sont placés dans un fichier /etc/shadow, par sécurité lisible seulement par root.

Structure de /etc/passwd

Ce fichier comprend 7 champs, séparés par le symbole « : »

1 Nom de connexion
2 Ancienne place du mot de passe crypté
3 Numéro d'utilisateur uid, sa valeur est le véritable identifiant pour le système Linux. L'uid de root est 0. Le système attribue conventionnellement un uid à partir de 1000 aux comptes créés.
4 Numéro de groupe gid, dans lequel se trouve l'utilisateur par défaut. Le gid de root est 0. Le système attribue conventionnellement un gid à partir de 1000 aux groupes d'utilisateurs créés.
5 Nom complet, il peut être suivi d'une liste de renseignements personnels. Le champ est écrit de la façon suivante: « Nom et prénom, numéro de bureau; numéro de téléphone travail, numéro de téléphone personnel, autres commentaires ». Chaque champ est séparé par le symbole « , »
6 Répertoire personnel (c'est également le répertoire de connexion).
7 Programme à lancer au démarrage (programme de base), généralement un interpréteur de commande (shell). La durée de vie de ce processus correspond à celle de la session utilisateur, c'est à dire que la session de l'utilisateur se terminera avec le processus. Il est possible de préciser ici tout type de programme, ce qui permet de limiter le champ d'action d'un utilisateur en le connectant directement au programme qu'il doit utiliser, par exemple.

Structure de /etc/group

Ce fichier comprend 4 champs, séparés par le symbole « : »

1 Nom du groupe
2 x pour remplacer un mot de passe non encore attribué
Numéro de groupe, c'est à dire l'identifiant gid
3 La liste des membres du groupe

Structure de /etc/shadow

Le problème des fichiers passwd traditionnels est qu'ils sont lisibles par tous ! C'est à dire que n'importe qui sur le système peut lire le fichier passwd. Ceci permet aux programmes d'y extraire des informations au sujet d'un utilisateur, telles que son nom complet. Ceci veut donc dire que tout le monde peut lire le mot de passe crypté dans le second champ. Aussi, tout le monde peut-il copier le champ ``mot de passe" d'un utilisateur et essayer des milliards de mots de passe jusqu'à ce qu'un des mots cryptés ainsi composé corresponde au mot de passe crypté de l'utilisateur. Si vous avez une centaine d'utilisateurs sur votre système, il se peut que plusieurs d'entre eux choisissent un mot du dictionnaire comme mot de passe. Les attaques par le dictionnaire consiste à tester les 80.000 mots d'anglais courant pour trouver une correspondance. Si vous pensez qu'il est astucieux d'ajouter un nombre en préfixe à un mot du dictionnaire, sachez que les algorithmes de craquage des mots de passe savent ça aussi, comme c'est d'ailleurs le cas pour d'autres astuces. C'est pour résoudre ce problème que les mots de passe cryptés ont été inventés. Le fichier des mots de passe cryptés est exclusivement utilisé lors de l'étape d'authentification, en vérifiant que l'utilisateur est le véritable détenteur du compte. Il n'est pas lisible par tous et aucun utilisateur normal n'a le droit de voir le champ ``mot de passe chiffré". Il n'y a pas d'information dans ce fichier nécessaire aux programmes. Les champs du fichier/etc/shadow sont séparés par des doubles points comme c'est le cas du fichier /etc/passwd. :

Ce fichier comprend 9 champs, séparés par le symbole « : »

« username:passwd:last:may:must:warn:expire:disable:reserved »

1 username : Le nom de connexion
2 passwd : Le mot de passe encodé. C'est le mot de passe chiffré résultant d'une transformation irréversible d'un mot de passe à 8 caractères. Un algorithme mathématique est appliqué au mot de passe de manière à produire un résultat unique pour chaque mot de passe. Pour retrouver le mot de passe à partir du mot chiffré, il faut essayer chaque mot possible. Bien qu'une attaque brutale de ce type soit considérée comme coûteuse du point de vue de la charge de calcul, elle n'est pas impossible. Pour vérifier si un mot de passe est bon, il suffit de lui appliquer l'algorithme permettant de vérifier la correspondance avec le mot de passe crypté. C'est comme cela que fonctionne la commande login. Si vous voyez :
   - un signe « * » à la place du mot chiffré, cela signifie que le compte a été désactivé.
   - un signe "!" à la place du mot chiffré, cela signifie que le compte est bloqué.
3 last : Date de la dernière modification (en nombre de jours depuis le 1er janvier 1970).
4 may : Nombre de jours avant que le mot de passe puisse être modifié. Usuellement, la valeur est = 0. Ce champ n'est pas souvent utilisé.
5 must : Nombre de jours avant que le mot de passe ne doive être modifié. Ce champ est rarement utilisé. Par défaut, sa valeur est 99999.

6 warn : Nombre de jours durant lesquels l'utilisateur est prévenu de l'expiration de son mot de passe. 7 Expire : Nombre de jours entre l'expiration du mot de passe et la fermeture du compte.

8 Disable : Date de la fermeture du compte (en nombre de jours depuis le 1er janvier 1970). -1 est utilisé pour indiquer un nombre infini de jours (ce qui veut dire que cette propriété du compte n'est pas considérée).
9 reserved : Ce champ est réservé pour une utilisation future.

Quelques utilitaires.

Pour avoir plus de renseignements sur les commandes suivantes, n'hésitez pas à utiliser les pages manuel (man).

Décrire un utilisateur.

chfn: Cette commande permet d'indiquer dans le champ numéro 5 du fichier /etc/passwd différentes informations sur un utilisateur, son nom complet, son bureau, ses numéros de téléphone (séparées par des virgules).

« chsh » Changer l'interpréteur de commandes.

Cette commande permet de changer l'interpréteur de commande d'un utilisateur, entre autre quand on ne veut pas laisser la possibilité à un utilisateur de se connecter en mode console sur la station ; l'on met alors /bin/false.

« passwd » Changer le mot de passe.

Cette commande est chargée du cryptage du mot de passe dans /etc/shadow
Syntaxe : **`passwd [option] nom-login`**

Options :

- -d , pour supprimer le mot de passe, l'utilisateur pourra se connecter sans!
- -l , pour verrouiller le compte et empêcher sa connexion.
- -u , pour déverrouiller.

« id » Connaître les identifiants.

Cette commande permet de déterminer l'identifiant utilisateur et l'identifiant groupe de l'utilisateur courant ou d'un utilisateur précis. De plus, elle affiche les groupes auxquels appartient cet utilisateur.

« groups » Déterminer les groupes d'appartenance d'un utilisateur.

Cette commande permet d'afficher les groupes auxquels appartient un utilisateur.

Gestion d'utilisateurs en nombre.

Les commandes de gestion des utilisateurs peuvent devenir fastidieuses et difficile à manipuler dans le cas de gestion d'utilisateurs en nombre. Dans ce cas il devient plus facile de modifier directement les fichiers /ets/passwd et /etc/group. Dans une entreprise, la liste des utilisateurs existe. Il est donc possible de générer ces fichiers à l'aide de script. Ces deux fichiers étant de simples fichiers texte composés de champs séparés par le symbole « : », ils peuvent être aussi modifiés à l'aide d'un tableur. C'est à vous de choisir la méthode qui vous convient le mieux.

Voici les étapes à réaliser :

- Création du fichier /etc/passwd.
- Création du fichier /etc/group.
- Création de l'arborescence correspondante.
- Création des mots de passe.
- Tests de connexion avec chaque utilisateur créé.

Sauvegarde ou exportation de la gestion des utilisateurs

La sauvegarde de la gestion des utilisateurs se limite à la sauvegarde de trois fichiers :

- /etc/passwd
- /etc/group
- /etc/shadow

Pour exporter la gestion des utilisateurs, il faut ajouter les lignes correspondant aux utilisateurs créés des fichiers /etc/passwd, /etc/group et /etc/shadows dans les fichiers correspondants de la machine réceptrice. N'oubliez pas de créer l'arborescence correspondante

NFS

NFS c'est quoi ?

NFS (Network File System) est un protocole qui permet le partage des fichiers entre des ordinateurs utilisant les systèmes d'exploitation de type Unix.

Un serveur NFS donne la possibilité de partager un disque dur, un espace de stockage avec de nombreux postes clients. Ces clients voient le partage comme une ressource locale, l'intégrant parfaitement au système. C'est une architecture client / serveur.

NFS est aussi utilisé dans le cas de clients légers, machines sans disque dur. Le système d'exploitation, les logiciels et les données sont localisés sur un serveur distant et montés au démarrage du client léger via le protocole NFS.

Comment fonctionne NFS ?

Deux versions de NFS sont actuellement en vigueur. La version 2 de NFS (NFSv2) utilisée depuis plusieurs années, est largement supportée. La version 3 de NFS (NFSv3) apporte d'autres fonctions, y compris un traitement de fichiers de taille variable et un meilleur rapportage d'erreurs, mais n'est pas entièrement compatible avec les clients NFSv2.

NFSv2 utilise le protocole UDP (User Datagram Protocol) pour fournir une connexion réseau sans état entre le client et le serveur. NFSv3 peut utiliser soit UDP soit TCP (Transmission Control Protocol) en cours d'exécution sur un réseau IP.

La connexion sans déclaration UDP réduit le trafic réseau, puisque le serveur NFS envoie au client un cookie qui l'autorise à accéder au volume partagé. Ce cookie est une valeur aléatoire stockée du côté serveur et transmis en même temps que les requêtes RPC du client. Non seulement le serveur NFS peut être redémarré sans affecter le client mais le cookie restera intact. Cependant, vu que UDP est sans état, si le serveur s'arrête inopinément, les clients UDP continueront à saturer le réseau de requêtes. Pour cette raison, TCP est le protocole préféré lors de la connexion sur un serveur NFSv3.

NFS n'effectue d'authentification que lorsqu'un système client tente de monter une ressource NFS partagée. Pour limiter l'accès au service NFS, des enveloppeurs TCP sont employés. Ceux-ci lisent les fichiers /etc/hosts.allow et /etc/hosts.deny pour déterminer si un client particulier doit se voir refuser ou accorder l'accès au service NFS.

Après l'autorisation d'accès du client permise par les enveloppeurs TCP, le serveur NFS se réfère à son fichier de configuration, /etc/exports, pour déterminer si le client peut monter l'un des systèmes de fichiers exportés. Après avoir autorisé l'accès, l'utilisateur peut effectuer toute opération de fichiers ou de répertoires.

Les privilèges de montage NFS sont accordés à l'hôte client et non pas à l'utilisateur. Ainsi, tout utilisateur sur un hôte client ayant des permissions d'accès, peut accéder aux systèmes de fichiers exportés. Lors de la configuration de partages NFS, faites particulièrement attention aux hôtes qui obtiennent les permissions de lecture/écriture (rw).

Le protocole NFS4 apporte de nombreuses améliorations dans divers domaines. Nous nous focaliserons uniquement dans ce paragraphe sur les nouveaux mécanismes de sécurité. Le protocole s'appuie toujours sur l'échange de messages RPC pour la communication entre le client et le serveur. NFS4 dispose des méthodes d'authentification des versions précédentes et en propose une nouvelle intitulée RPCSEC_GSS. Cependant cette nouvelle méthode ne se contente pas de réaliser l'authentification des messages, c'est un mécanisme de sécurité à part entière qui repose sur la GSSAPI (Generic Security Service Application Program). Il existe de nombreux procédés permettant de sécuriser les transmissions entre deux intervenants :

certains utilisent à la fois la cryptographie à clé publique et la cryptographie à clé privée, d'autres uniquement la cryptographie à clé privée. L'objectif de la GSSAPI est de fournir une interface de programmation générique qui encapsule plusieurs mécanismes de sécurité. Toute session RPC de type RPCSEC_GSS est constituée de 3 phases :

- une phase de création de contexte. Le client et le serveur se mettent d'accord sur le mécanisme de sécurité qu'ils vont utiliser, les algorithmes permettant de mettre en œuvre ce mécanisme et le type de service à mettre en place :
  - service d'authentification : les entêtes RPC sont authentifiées avec l'algorithme sélectionné lors de la phase précédente,
  - service d'intégrité : les entêtes RPC sont authentifiées. L'intégrité des données est vérifiée.
  - service de confidentialité : inclus le service d'intégrité et chiffrement des données RPC.
- une phase d'échange de données correspondant à la communication entre le client et le serveur,
- une phase de destruction de contexte.

Mise en place du serveur

Liste des paquets à installer pour le fonctionnement commun du serveur NFS.

- portmap : gère les appels de procédures distantes ou Remote Procedure Call (RPC).
- nfs-common : gère les états des transactions NFS.
- nfs-kernel-server : serveur NFS.

Vérification de la présence ou non des paquets

```
serveur:~# dpkg -l portmap nfs-common nfs-kernel-server
Souhait=inconnU/Installé/suppRimé/Purgé/H=à garder

| État=Non/Installé/fichier-Config/dépaqUeté/échec-conFig/H=semi-installé/W=attend-traitement-déclenchements

|/ Err?=(aucune)/H=à garder/besoin Réinstallation/X=les deux (État,Err: majuscule=mauvais)

||/ Nom                          Version                      Description

+++-==============================-============================-

================================================================================

ii  nfs-common                   1:1.1.2-6lenny2              NFS support files common to client and server
ii  nfs-kernel-server            1:1.1.2-6lenny2              support for NFS kernel server

ii  portmap                      6.0-9                        RPC port mapper
```

ou

```
serveur:~# dpkg-query -l | grep nfs

ii  libnfsidmap2                 0.20-1                   An nfs idmapping library

ii  nfs-common                   1:1.1.2-6lenny2          NFS support files common to client and serve
ii  nfs-kernel-server            1:1.1.2-6lenny2          support for NFS kernel server
```

Détermination de la version du protocole utilisée.

Sur les distributions récentes le serveur nfs est intégré au noyau (kernel). Les paramètres dépendent de la configuration du kernel. On les trouve dans le fichier /boot/config-x

```
serveur# cat /boot/config-2.6.26-2-xen-amd64 | grep NFS
CONFIG_NFS_FS=m

CONFIG_NFS_V3=y
CONFIG_NFS_V3_ACL=y
CONFIG_NFS_V4=y
CONFIG_NFSD=m
CONFIG_NFSD_V2_ACL=y
CONFIG_NFSD_V3=y
CONFIG_NFSD_V3_ACL=y
CONFIG_NFSD_V4=y
CONFIG_NFS_ACL_SUPPORT=m
CONFIG_NFS_COMMON=y
CONFIG_NCPFS_NFS_NS=y
```

Installation du serveur NFS (si non présent)

```
#apt-get install nfs-kernel-server
```

Contrôle de l'état des services :

```
serveur:~# cat /etc/services | grep nfs

nfs          2049/tcp                    # Network File System

nfs          2049/udp                    # Network File System
```

Le port d'écoute est le 2049 pour les protocoles UDP et TSP

```
serveur:~# cat /etc/services | grep portmap

sunrpc              111/tcp              portmapper    # RPC 4.0 portmapper
sunrpc              111/udp              portmapper

rpc2portmap 369/tcp

rpc2portmap 369/udp                                 # Coda portmapper
```

Le port d'écoute est le 111 pour les protocoles UDP et TSP

Démarrer arrêter le service portmap

Après toute modification du fichier de configuration vous devez redémarrer le service nfs

```
# /etc/init.d/portmap stop
# /etc/init.d/portmap start
```

Démarrer arrêter le service NFS

Après toute modification du fichier de configuration vous devez redémarrer le service nfs

```
# /etc/init.d/nfs-kernel-server stop
# /etc/init.d/nfs-kernel-server start
```

Configuration du service NFS

On édite le fichier /etc/exports

Structure et syntaxe

Chaque ligne définit un répertoire partagé avec la syntaxe suivante:
dossier_partagé machine_cliente1(options) machine_cliente2 (options)...

- dossier_partagé : chemin du dossier partagé.
- machine_cliente : machine cliente qui peut accéder à ce partage.
- options : spécifient les options de partage. Pour connaître toutes les options utilisez faites man exports.

Attention les utilisateurs doivent avoir le même UID sur les machines clientes que sur la machine hébergeant le service NFS. Il en est de même pour les GID des groupes.

OPTIONS	SIGNIFICATIONS
root_squash	Interdiction à l'utilisateur root des machines spécifiées d'être assimilé au root du serveur, en transformant leur UID/GID en ceux de l'utilisateur nobody (UID 65534 eq. -2) Cette option est active par défaut
no_root_squash	Annulation de l'option par défaut (root_squash)
ro	read-only. Le client peut seulement lire dans le répertoire exporté par conséquent, monter ce répertoire en lecture seule.
rw	read-write. Le client peut lire et écrire dans le répertoire exporté par conséquent, monter ce répertoire en lecture et écriture.
intr	Autorise les signaux à interrompre un appel NFS. Ceci est utile lorsque le serveur ne répond pas.
timeo=n	indique le temps (en dixièmes de seconde) pendant lequel le client nfs attendra qu'une requête aboutisse. La valeur par défaut est 7 donc 0.7s

Exemple du contenu d'un fichier /etc/exports

```
# repertoire   liste-machines  (liste-options)

/home/toto    m1(ro) m2(rw)

/home/tata    192.168.0.1(rw)

/home/tata/doc (ro)
```

- Le répertoire /home/toto sera accessible en lecture seule pour la machine m1 et lecture écriture pour la machine m2
- Le répertoire /home/tata sera accessible en lecture écriture pour la machine ayant l'adresse IP 192.168.0.1
- Le répertoire /home/tata sera accessible en lecture seule par toutes les machines

Pour valider un changement opéré dans ce fichier de configuration, faire appel à la commande :

```
# exportfs -a
```

Le fichier /etc/hosts.deny liste les machines clientes dont on veut interdire l'accès au service NFS. Le fichier /etc/hosts.allow liste les machines clientes dont on veut permettre l'accès au service NFS. Cinq fichiers interviennent dans le fonctionnement du serveur NFS, en plus du fichier /etc/exports.

- /etc/mtab contient la liste des fichiers montés par le noyau par la commande mount.
- /var/lib/nfs/rmtab contient un ensemble de lignes, avec sur chaque ligne le nom d'un client et le système de fichiers qu'il importe de ce serveur.
- /var/lib/nfs/etab contient la liste de systèmes de fichiers partagés par NFS. Ce fichier est lu par mountd dès qu'un client distant demandera l'accès pour le montage d'une arborescence de fichier. Il est créé par exportfs
- /proc/fs/nfs/exports contient la liste des clients reconnus par le noyau.
- /var/lib/nfs/xtab contient la liste des noms de machines reconnues par le noyau

Quand un client souhaite accéder à un répertoire par nfs, il commence par le demander à mountd. Celui-ci recherche alors dans etab si la requête est accessible. Il vérifie aussi auprès du noyau que la requête du client est légitime par l'intermédiaire des fichiers hosts.allow et hosts.deny ou des règles de pare-feu. Cette vérification lui permet, entre autres, de mettre à jour etab si nécessaire. Si dans ce fichier, le répertoire exporté est autorisé à un groupe auquel appartient le client, mountd restreint alors la requête au client et en informe le noyau qui met à jour xtab avec ce nouvel hôte.

Mise en place du client NFS.

Liste des paquets à installer pour le fonctionnement commun du client NFS.

- portmap : gère les appels de procédures distantes ou Remote Procedure Call (RPC).
- nfs-common : gère les états des transactions NFS.

Vérification de la présence ou non des paquets

```
client:~#    dpkg    -l    portmap    nfs-common
Souhait=inconnU/Installé/suppRimé/Purgé/H=à garder

| État=Non/Installé/fichier-Config/dépaqUeté/échec-conFig/H=semi-installé/W=attend-traitement-déclenchements
```

```
|/ Err?=(aucune)/H=à garder/besoin Réinstallation/X=les deux (État,Err: majuscule=mauvais)
||/ Nom                          Version                  Description
+++-============================-========================-
================================================================
ii  nfs-common                   1:1.1.1-13               NFS support files common to client and server
ii  portmap                      6.0-5                    RPC port mapper
```

La commande système "mount"

Comme le système de fichiers Linux se concentre dans une seule arborescence de fichiers, l'accès et l'utilisation de systèmes extérieurs (partage, disques, disquettes, cd ..) doit s'effectuer par intégration de ces systèmes de fichiers dans le système fondamental "racine". Ce mécanisme d'intégration, souple et paramétrable, s'appelle le montage. Le montage permet d'accéder à des systèmes de fichiers (filesystem), ceux-ci peuvent être des partitions locales ou distantes, des système USB, des cdrom, etc... Elle doit être exécutée en tant que root. Techniquement, l'opération de montage consiste à mettre en relation un fichier de périphérique situé dans le répertoire /dev (qui permet la communication physique avec les données du périphérique) avec un noeud d'insertion dans l'arborescence, appelé son point de montage .

Quelques fichiers spéciaux et les périphériques associés:

- /dev/hda : disque maître sur le premier port IDE
- /dev/hda2 : seconde partition primaire sur ce disque
- /dev/tty1 : première console virtuelle
- /dev/lp0 : troisième port parallèle (imprimante)
- /dev/sda : premier disque dur SCSI.(les clés USB sont reconnues en tant que disques SCSI)
- /dev/sda3 : troisième partition sur ce disque
- /dev/video0 : Acquisition vidéo

On pourrait continuer ainsi de suite, la liste complète occuperait plusieurs pages. Il suffit de visualiser le résultat de la commande **`#ls /dev`** pour s'en rendre compte!

Naturellement le montage fondamental est celui du répertoire racine / du système. Celui-ci se trouve dans une partition automatiquement montée au démarrage. Ce processus est fondamental sur les systèmes de la famille UNIX, il conditionne tout accès à une ressource externe, en particulier à des ressources réseau à d'autres disques Linux

Sa syntaxe habituelle est :

```
#mount -t type_de_système_de_fichier -o options /dev/type_de_periphérique /mnt/répertoire_de_montage
```

à condition que le type de système de fichier soit supporté par le noyau et que le répertoire d'accrochage /mnt/répertoire_de_montage existe déjà.

- L'argument "-t"(type) sert a spécifier le type de système de fichier, les plus courants sont:
  - ext2 (type par défaut): c'est le filesystem linux par défaut
  - msdos:
    - vfat: FAT16 ou FAT32 de Win95 ou Win98
    - ntfs: NT4, Win2000 ou XP
  - ISO9660: pour les CDROM
  - nfs: (network file system) distant se trouvant sur une autre machine
- L'argument "-o" (options), sert à spécifier les options voulues. Sans cet argument on a les options par défaut suivantes:
  - rw: lecture et écriture
  - suid: prise en compte du bits Set-UID (identifiant utilisateur) ou Set-GID (identifiant groupe) des fichiers se trouvant sur le système monté. Il s'agit d'un dispositif de sécurité essentiel qui autorise un utilisateur quelconque (par rapport à la commande) à bénéficier de droits plus étendus que les siens (souvent ceux de root). Par exemple la commande sudo sous Ubuntu.
  - dev: permet l'utilisation des fichiers de périphériques.
  - exec: permet l'exécution des fichiers binaires.

Quelques exemples:

Avant de monter un support amovible vous devez l'insérer ou le connecter.

Avant de retirer le support amovible vous devez le démonter

```
#mount -t iso9660 /dev/cdrom /mnt/cdrom
```

Cette commande permet de monter le lecteur de CDROM identifié par /dev/cdrom dans le répertoire /mnt/ cdrom.

```
#mount -t nfs serv1:/partage /home/partage_serv1
```

Cette commande permet de monter la ressource partagée /partage du serveur distant nommé serv1 sur le répertoire d'accrochage /home/partage_serv1.

```
#mount -t vfat -o uid=5001,gid=5000,umask=000 /dev/hda1 /home/hdd_win
```

Cette commande permet de monter la partition Windows identifiée par /dev/hda1 dans le répertoire d'accrochage /home/hdd_win. L'utilisateur dont l' identifiant utilisateur (uid) est 5001 et le groupe dont l' identifiant groupe (gid) 5000, seront propriétaires d'accrochage de tous les fichiers. Lors de la création d'un fichier, celui-ci aura les droits maximum

qui correspondent à un umask de 000 en notation octale.

Pour démonter ce système de fichier, il suffit de taper en root :

```
#umount /dev/type_de_périphérique
```

ou :

```
#umount /mnt/répertoire_de_montage
```

Par contre, n'importe quel utilisateur peut taper la commande mount tout court pour savoir quels sont les systèmes de fichiers montés à l'instant d'exécution de la commande.

Les fichiers /etc/fstab et /etc/mtab

Le premier processus init (exécuté au démarrage), après chargement du noyau, se charge de monter les systèmes de fichiers conformément aux spécifications du fichier /etc/fstab et effectue leur éventuel montage automatique.

En effet, ce fichier /etc/fstab constitue une véritable "table de montage". Il fait l'inventaire des divers systèmes de fichiers que le noyau Linux est susceptible de gérer, précise la façon de les monter. S'ils doivent l'être au démarrage, ou plus tard à la demande des utilisateurs etc ..

Le fichier /etc/mtab est dynamique et décrit à tout moment l'état des montages des partitions disques et des périphériques.

Structure des tables

Sur chaque ligne on trouve la description du montage d'un système, avec 6 champs :

- nom du fichier "device"
- chemin vers le point de montage
- le type de fichiers : ext2 (Linux), swap, msdos, vfat (Win9x), ntfs (NT), iso9660 (Cd-rom), nfs
- liste d'options de montage, séparées par des virgules. Liste non exhaustive des options:
  - auto: mount automatique lors d'un appel à « mount -a ».
  - defaults: équivalent à rw,suid,dev,exec,auto,nouser,async.
  - dev: interprète les fichiers device ;
  - exec: permet l'exécution de programmes .
  - nosuid: les bits SetUID et SetGID ne sont pas pris en compte .
  - nouser: seul root a la permission de faire un mount.
  - ro: read only : accès en lecture seulement.
  - rw: read write : accès en lecture et écriture.
  - setuid: les bits SetUID et SetGID sont pris en compte
  - user: permission accordée aux utilisateurs ordinaires de faire un mount.
- paramètre pour dump (commande de sauvegarde) Une valeur 0 signifie que le système de fichiers ne sera pas sauvegardé lors d'un dump. Une valeur 1 pour les fichiers concernés par une sauvegarde au moyen de l'utilitaire dump.
- paramètre pour la vérification des fichiers. Il indique l'ordre dans lequel fsck devra vérifier les fichiers. fsck est l'utilitaire permettant de vérifier l'intégrité d'un système de fichiers. Une valeur 1 signifie une priorité (c'est normalement la partition racine /).
  - Une valeur 2 signifie pas de priorité.
  - Une valeur 0 signifie pas de vérification par fsck.

Le fichier possède autant de lignes qu'il y a de file system à monter.

Exemple de /etc/fstab et /etc/mtab du même serveur

Le signe "#" en début de ligne indique que la ligne sera reconnue comme une remarque

/etc/fstab:

```
# /etc/fstab: static file system information.
#

#     <file system><mount point><type> <options>     <dump> <pass>
      proc   /proc  proc   defaults      0      0

      /dev/hda2     /      ext3   defaults,errors=remount-ro 0      1

      /dev/hda3     /home   ext3  defaults      0      2
      /dev/hdb1     /home/b1      ext3   defaults      0      2
      /dev/hdc1     /home/c1      vfat   defaults      0      2
      /dev/hda1     none   swap   sw     0      0
      /dev/hdd      /media/cdrom0iso9660       ro,user,noauto      0      0
```

/etc/mtab:

```
      /dev/hda2     /      ext3   rw,errors=remount-ro    0        0
      proc   /proc  proc   rw     0      0
      devpts /dev/pts      devpts rw,gid=5,mode=620   0   0
      tmpfs  /dev/shm      tmpfs  rw     0      0
      /dev/hda3     /home  ext3   rw     0      0
      /dev/hdb1     /home/b1   ext3      rw     0       0
```

```
/dev/hdc1    /home/c1   vfat      rw     0       0
usbfs   proc/bus/usb usbfs  w      0      0
```

On peut remarquer que les supports amovibles ne sont pas présents dans le serveur car le système de fichier /dev/hdd n'est pas présent dans le fichier /etc/mtab. On peut remarquer que l'on voit apparaître une ligne de plus qui est la ligne qui va permettre la gestion des périphériques USB. Le répertoire /proc/bus/usb est le répertoire ou se trouve les fichiers permettant la gestion de ces périphériques.

SSH

Qu'est ce que SSH?

Secure Shell (SSH) est à la fois un programme informatique et un protocole de communication sécurisé. Le protocole de connexion impose un échange de clés de chiffrement en début de connexion. Par la suite toutes les trames sont chiffrées. Il devient donc impossible d'utiliser un sniffer pour voir ce que fait l'utilisateur.

Nature du problème

Installer un serveur SSH permet aux utilisateurs d'accéder au système à distance, en rentrant leur login et leur mot de passe (ou avec un mécanisme de clés). Cela signifie aussi qu'un pirate peut essayer d'avoir un compte sur le système (pour accéder à des fichiers sur le système ou pour utiliser le système comme une passerelle pour attaquer d'autres systèmes) en essayant plein de mots de passe différents pour un même login (il peut le faire de manière automatique en s'aidant d'un dictionnaire électronique). On appelle ça une attaque en force brute.

Il y a donc quatre contraintes majeures pour garder un système sécurisé après avoir installé un serveur SSH :

- Avoir un serveur SSH à jour au niveau de la sécurité, ce qui doit être le cas si vous faites consciencieusement les mises à jour de sécurité.
- Les mots de passe de TOUS les utilisateurs doivent être suffisamment complexes pour résister à une attaque en force brute.
- Utiliser un jeux de deux clés (clé privée , clé publique).
- Surveiller les connexions en lisant régulièrement le fichier de log /var/log/auth.log.

Choisir des mots de passe complexes

Un mot de passe complexe est un mot de passe qui ne veut rien dire, qui n'est pas dans le dictionnaire et qui comporte au moins 10 caractères, de préférence avec un mélange de lettres minuscules, de lettres majuscules, de chiffres et de caractères de ponctuation.

Tester la complexité des mots de passe

Pour vérifier que les mots de passe des utilisateurs du système sont vraiment complexes, le root peut les soumettre à un cracker de mots de passe et voir combien de temps ils résistent !

Les mots de passe des utilisateurs sont stockés dans le fichier /etc/shadow. Seul l'utilisateur root peut lire ce fichier. Pour tester la complexité des mots de passe, le root peut donc installer le programme john et le lancer sur le fichier /etc/shadow :

```
# apt-get install john
# john /etc/shadow
```

Quand john a trouvé un mot de passe, il l'affiche avec le login associé.

Attention, john utilisera le processeur à 100 % ! Il est donc conseillé de lui donner une priorité faible (commande nice ou renice) si la machine doit être utilisée pendant ce temps. Plus le nombre d'utilisateurs est grand, plus il faudra laisser tourner john longtemps pour que le test soit significatif.

Installation du serveur SSH

Installation du serveur SSH

Pour pouvoir vous connecter à distance sur une machine, vous devez installer un serveur SSH :

```
# apt-get install openssh-server
```

L'installation comporte une étape de génération des clefs de cryptage. Finalement, le serveur SSH se lance.

Configuration du serveur SSH

Le fichier de configuration du serveur SSH est /etc/ssh/sshd_config. À ne pas confondre avec le fichier /etc/ssh/sshd_config, qui est le fichier de configuration du client SSH. La grande majorité des lignes sont commentées.

Les lignes les plus importantes de ce fichier de configuration sont:

- Port X : Signifie que le serveur SSH écoute sur le port X. Le port 22 est le port par défaut de SSH. Vous pouvez le faire écouter sur un autre port en changeant cette ligne. Vous pouvez aussi le faire écouter sur plusieurs ports à la fois en rajoutant des lignes similaires.
- PermitRootLogin yes: Signifie que vous pouvez vous logguer en root par SSH. Vous pouvez changer et mettre "no", ce qui signifie que pour vous connecter en root à distance, vous devrez d'abord vous connecter par SSH en tant que simple utilisateur, puis utiliser la commande su pour devenir root. Sans cela, un pirate n'aurait qu'à trouver le mot de passe du compte root, alors que là, il doit trouver votre login et votre mot de passe en plus!
- X11Forwarding yes: Signifie que vous allez pouvoir travailler en export display par SSH.
- PasswordAuthentication no : Signifie que vous interdisez l'authentification par mot de passe.
- ForwardAgent yes : Permet le transport de l'agent d'authentification.

Si vous avez modifié le fichier de configuration du serveur, il faut relancer le serveur SSH pour que les informations soient prises en compte:

```
# /etc/init.d/ssh reload
```

En tant qu'administrateur réseau, vous devez faire des taches répétitives sur un groupe machines Linux, DSH est là pour vous aider. Dsh (Distributed SHell) permet de lancer une commande sur des groupes de machines. Par exemple il peut-être intéressant d'éteindre toutes les machines simultanément.

Mise en place de DSH

- Installer ssh sur toutes les machines à administrer du réseau.
  #apt-get install ssh

- Installer dsh sur notre machine administratrice.
  #apt-get install dsh

Pour ne pas être obligé d'entrer votre mot de passe pour chaque machine il est raisonnable d'utiliser l'authentification par certificat. Il faut donc générer une clé ssh sur la machine administratrice.

- Génération du couple clé publique (id_dsa.pub) clé privé (id_dsa) sur la machine administratrice.
  $ ssh-keygen -t dsa

- Copie de la clé publique sur les machines administrées, dans le fichier ~/.ssh/id_dsa.pub, jusqu'à la dernière machine.
  $ ssh-copy-id -i /home/user/.ssh/id_dsa.pub user@ip_machine_administree_1

  $ ssh-copy-id -i /home/user/.ssh/id_dsa.pub user@ip_machine_administree_2

Configuration de DSH

Le fichier /etc/dsh/dsh.conf est le fichier de configuration de DSH.

Le fichier /etc/dsh/machines.list contient la liste des machines. Il contient les adresses ip ou les noms dns des machines administrées. Le format est une machine par ligne, sans ligne vide.

Le dossier group permet de créer des groupes de machines. Pour ce faire, vous devez créer un fichier contenant les adresses ip ou les noms dns des machines administrées de ce groupe et lui donner un nom. Pour des raisons de rapidité d'écriture de la commande dsh choisissez un nom court, voir un caractère.

Exemple :

```
$ cat /etc/dsh/group/t
192.168.0.2

192.168.0.3

192.168.0.3
```

Utilisation de DSH

Arrêt des machines du groupe « t » :

```
# dsh -g t -cM poweroff
```

L'option -c permet de réaliser la commande simultanément, l'option -M spécifie le nom des machines.

Mise à jour de toutes les machines :

```
# dsh -g all -cM apt-get update
```

Utiliser la page manuel pour plus de renseignements.