Plongée dans l'abîme Linux

Résumé du cours

(Par Kacper, Nicolas et Pierre)

I Table des matières

Chapitre 1 : Les outils essentiels de base

Il existe 7 commandes essentielles sous la majorité des distributions Linux:

- **"whoami"** : Elle vous donne votre login actuel.
- **"hostname"** : Elle vous donne le nom de la machine sur laquelle vous travaillez.
- **"date"** : Elle vous affiche la date actuelle.
- **"uname"** : Elle vous donne des informations sur le système actuel.
- **"passwd"** : Elle permet de changer le mot de passe d'un l'utilisateur.
- **"touch"** : Elle crée un fichier vide ou met à jour la date de modification d'un fichier existant.
- **"last"** : Elle vous donne la liste des utilisateurs qui se sont connectés récemment sur le système.

La plupart des commandes peuvent être accompagnées d'un attribut qui modifient certains critères de celle-ci selon l'attribut utilisé.

Les attributs Linux ne possèdent qu'un seul tiret.
Les attributs Unix quant à eux possèdent deux tirets.
L'attribut "–help" sert à afficher une aide sur la commande en question.

La commande "man" sert à ouvrir le manuel de la distribution Linux sur une commande demandée. Le manuel a plusieurs sections qui correspondent à différents aspects de chaque commande. Par exemple, "man 1 passwd" vous ouvrira le manuel pour la commande passwd à la section 1 qui est la section des commandes utilisateur. L'attribut -a permettra à "man" d'afficher toutes les sections du manuel pour la commande en question.

Le Shell possède un outil "autocomplete", ce qui permet d'auto-achever des morceaux de commandes en appuyant sur la touche TAB s'il en existe commençant par ce que vous avez tapé. Tapez 2x sur TAB pour avoir une liste réduite de commandes possibles.

La commande "history" référence un fichier qui conserve les logs des commandes utilisées par l'utilisateur. (Très pratique pour retrouver toutes les commandes tapées sur un serveur par un utilisateur kamikaze)

En Linux, il y a 3 canaux principaux:

- Le **STDIN** : C'est l'input vers la distribution (en général, c'est un clavier).
- Le **STDOUT** : C'est l'output vers l'utilisateur (en général, un écran).
- Le **STDERR** : C'est le canal d'erreur (en général, vers un fichier).

Il est possible de diriger la sortie standard vers un fichier plutôt que vers l'écran.
Par exemple, la commande "history » /home/admin/Documents/test" vous renverra les logs des commandes récentes dans un fichier nommé "test" qui lui sera dans "Documents", qui lui-même se trouve dans "admin", et "admin" se trouvant dans "home".

Les "Tuyaux" (pipe) « | » :
Le "tuyau" (pipe) permet de rediriger la sortie d'une commande dans l'entrée d'une seconde pour que la deuxième commande effectue un traitement sur le résultat de la première.

Chapitre 2 : Les outils pour la gestion de fichiers

Les arborescences de fichiers en Linux prennent toujours naissance avec le root directory ou le "/"
Depuis le "/" l'arborescence se dessine autour de dossiers fondamentaux pour le fonctionnement du système.

Il est recommandé d'isoler certains dossiers sur différents stockages, par exemple le "/home" car il prend souvent beaucoup de place.

Quelques commandes basiques:

→ **"mount"** sert à connecter une partie du système de fichier sur un stockage physique particulier.
→ **"man hier"** vous permet d'obtenir toutes les informations sur le système de fichier.
→ **"ls"** vous renvoie la liste des fichiers et dossiers présents dans le répertoire courant.

- **"ls" avec argument "-l"**: renvoie aussi les propriétés de ceux-ci.
- **"ls" avec l'argument "-a"**: renvoie TOUS les fichiers/dossiers présents dans le répertoire courant.
- **"ls" avec l'argument "-lrt"**: renvoie la liste des fichiers/dossiers classés en fonction du temps de la dernière modif.

Le Shell possède des **"Wildcards"**. C'est-à-dire qu'il est capable d'interpréter des symboles de remplacement dans les commandes. Voici certains de ces symboles:

- "*" remplace plusieurs caractères inconnus.
- "**?**" remplace un caractère inconnu.
- "**a-9**" remplace un caractère par un des caractères du "range" défini.

La commande "**cp**" permet de copier un fichier.
Exemple:

- "cp [source] [destination]" pour un fichier.
- "cp -R [source] [destination]" pour un dossier.

La commande "**cd**" vous permet de vous déplacer dans le système. Exemple :

- "cd [chemin]"
- "cd " équivaut à "cd /home".
- "cd ." permet de rester dans le répertoire courant.
- "cd .." permet de remonter dans le répertoire parent.

La commande "**mkdir**" permet de créer un dossier dans le système.
La commande "**rmdir**" permet de supprimer un dossier dans le système.
La commande "**mv**" permet de déplacer un fichier.
La commande "**rm**" permet de supprimer un fichier.

- "-r" permet de supprimer un dossier.
- "-f" permet de forcer la suppression du fichier ou dossier sans confirmation.

[!] Attention: "**rm -rf**" peut être extrêmement dangereux si utilisé avec les droits "root".

Un chemin absolu commence à la racine du système de fichier, donc avec un backslash("/").
Un chemin relatif commence à la position actuelle dans le système de fichier.

Hard-Link et Symbolic-Link

Le Hard-Link référence un inode qui lui même référence un block sur le stockage.
Le Symbolic-Link référence un Hard-Link.
La commande "**ln**" crée un Hard-Link.
La commande "**ln -s**" crée un Symbolic-Link.
La commande "**find**" permet de trouver un fichier.

Chapitre 3 : Traitement de texte

VIM permet d'éditer des fichiers et faire du traitement de texte, il fonctionne sous 3 modes:

- mode "**command**" permet de sauvegarder, rechercher, quitter.
- mode "**insert**" permet d'éditer le texte.
- mode "**visual**" permet de faire des sélections dans le texte.

La commande "**less**" organise le texte en page pour le Shell.
La commande "**cat**" permet de lire un fichier directement dans le shell sans mettre le texte en page.
La commande "**tac**" fonctionne de la même façon que "cat" mais en commençant par la fin du fichier.
La commande "**head**" permet de lire les n premières lignes d'un fichier.
La commande "**tail**" permet de lire les n dernières lignes d'un fichier.

La commande "**grep**" permet de trouver une correspondance entre une expression et un texte contenu dans un fichier.
La commande "**awk**" permet de découper un texte en fonction de ses délimiteurs.
La commande "**sort**" permet de trier un texte dans l'ordre alphabétique ou numérique.
La commande "**tr**" permet de traduire certains caractères du texte en d'autres.

Chapitre 4 : Se connecter à un serveur

Il existe deux types d'utilisateur sur Linux:

1. Les "*local user*" avec certaines autorisations, sans accès au noyau.
2. Le superuser "root" qui est tout puissant sur le système et a accès au noyau.

Notez qu'il est possible de se connecter en tant que "root" à condition de connaître son mot de passe. Il est également préférable de ne travailler sur un serveur en "root" qu'en cas de nécessité et si l'on est sûr de ce que l'on fait, car un accident ou une mauvaise manipulation en tant que root peut être fatale pour un serveur.

La commande "**su**"*(Switch User)* permet de changer de session utilisateur.
La commande "**su -**" permet de se connecter au Shell avec les droits de "root".
La commande "**sudo**"*(SuperUser Do)* permet de lancer une commande en tant que "superuser".

Il est possible de se connecter à distance sur un Linux à l'aide telnet ou ssh.
Le ssh étant beaucoup plus moderne et sécurisé que telnet, nous utiliserons ssh.
Exemple: ssh [Username]@[Adresse IP]

Chapitre 5 : La gestion des utilisateurs

Pourquoi les utilisateurs sont-ils nécessaires ?
En simple, il est impossible de se connecter à une distribution Linux sans utilisateur, tout processus démarré dans le système appartient à un utilisateur. Pour accéder à une ressource, il faut un compte utilisateur enregistré dans le système. Chaque fichier ou dossier doit appartenir à un utilisateur.

La commande "**useradd**" permet de créer un utilisateur:
Exemple: useradd [options] [login]

- "**useradd –help**" apporte une courte aide sur l'utilisation de la commande.
- "**useradd -m [login]**" permet de créer un /home directory en même temps que la création de l'utilisateur.
- "**useradd -g [groupe] [login]**" permet d'attribuer un groupe primaire à l'utilisateur créé, mais attention: le groupe doit être déjà existant.
- "**useradd -G [groupe] [login]**" permet d'attribuer un groupe secondaire à l'utilisateur créé(ça ne crée pas le groupe non plus).

La commande "**groupadd**" permet de créer un groupe:

- "**groupadd [options] [groupe]**" permet de créer un groupe.
- "**groupadd -g [groupe]**" permet de choisir le Group ID du groupe créé.

Exemple: "groupadd -g 2000 compta"

La commande "**usermod**" permet d'attribuer un groupe à un utilisateur déjà existant:

- "**usermod" -g [nouveau groupe primaire] [login]**" permet de changer le groupe primaire de l'utilisateur.
- "**usermod" -G [nouveaux groupes secondaires] [login]**" permet d'écraser la liste de groupes secondaires de l'utilisateur pour les remplacer par ceux cités.
- "**usermod" -aG [nouveaux groupes secondaires] [login]**" permet d'ajouter des groupes secondaires à ceux déjà existants chez l'utilisateur.

La commande "**userdel**" permet de supprimer un utilisateur.

- "**userdel -r**" permet de supprimer un utilisateur ainsi que son répertoire personnel.

La commande "**groupdel**" permet de supprimer un groupe.

- "**groupdel -f**" permet de forcer la suppression d'un groupe même si celui-ci contient des utilisateurs.

La commande "**groupmod**" permet de modifier les propriétés d'un groupe.

La commande "**chage**" permet approximativement les mêmes actions que passwd, en plus lisible.
Le fichier passwd contient la liste d'utilisateurs, est en accès non restreint et contient 7 champs:

- **Login**
- **Password**
- **UID**
- **GID**
- **comment**
- **homedir**
- **shell**

Le fichier group contient la définition des groupes, la liste de ses utilisateurs et contient 4 champs:

- **Group**
- **password**
- **GID**
- **users**

Il est possible de créer un utilisateur entièrement à la main en travaillant sur les fichiers car ce sont des fichiers plats, mais c'est vivement déconseillé.
Le fichier "**shadow**" contient, entre autre, les mots de passes cryptés des utilisateurs.

Chapitre 6 : Les permissions et les quotas

La commande "**chmod [permissions] [chemin]**" permet de changer les permissions sur un fichier ou dossier.
Il est possible de travailler avec le modèle **UGO** *(User-Group-Other)* ou en octal *(de 0 à 7)*

La commande "**setfacl**" permet de définir une access list.
Exemple: "setfacl -Rm [modif permission] [fichier/chemin de dossier]"

C'est par l'intermédiare de cette commande que nous allons orienter le type d'access list configuré:
- Normal *(appliqué sur les fichiers déjà existants).*
- Default *(appliqué sur les fichiers qui seront nouvellement créés).*

La commande "**getfacl [fichier/chemin de dossier]**" permet de connaître le statut ACL du fichier/dossier.

Les permissions permettent d'accorder des droits aux utilisateurs sur certains fichiers et dossiers, les attributs étendus quant à eux ne concernent que les fichiers et se posent comme une couche supplémentaire de sécurité pour le système de fichiers.
La commande "**lsattr**" permet de lister les attributs étendus des fichiers.
La commande "**chattr**" permet de changer les attributs étendus des fichiers.

Chapitre 7 : Configurer les éléments réseaux

Les quotas permettent des placer une limite d'espace disponible pour un utilisateur ou un groupe.
Il existe deux types de limites: les blocks et les inodes.

La commande "**repquota -a**" donne l'état actuel des quotas.

Il existe deux types de configurations réseau Linux:
1 - La configuration "**run time**" conçue pour monitorer les paramètres et effectuer des tests.
2 - La configuration "**persistante**" conçue pour fournir à votre machine un accès permanent et fiable au réseau.

Pour travailler en mode "run time" il faut utiliser la commande "**ip [options] [objets]**".
Pour voir les interfaces réseau disponibles sur le système, utilisez la commande "**ip link**".
Pour voir les interfaces réseau ainsi que leur adresse, utilisez la commande "**ip address show**".
Pour voir la passerelle par défaut de vos interfaces, utilisez la commande "**ip route show**".
Pour créer une configuration réseau persistante, utilisez la commande "nmcli". La commande étant très complète, il est nécessaire d'utiliser «TAB» pour auto-achever et construire instinctivement les commandes de configuration.
Nous travaillons donc principalement dans "**nmcli connection**".

Afin de modifier une connexion, il faudra compléter par un "**modify**", comme ceci: "**nmcli connection modify**". Ensuite, il faudra choisir un type de connexion. En filaire, par exemple: "**nmcli connection modify Connexion filaire 1 [type IP] (auto complete)**".
Pour activer l'interface de la configuration réseau, utilisez la commande "**nmcli connection up [nom de la connexion]**".

Afin de tester la connectivité d'un réseau, utilisez la commande "**ping [adresse ip/nom d'hôte]**". Cela va envoyer des paquets et en demander en retour. Si le serveur renvoie les paquets correctement, la connexion est établie.

Pour vérifier l'activité du serveur dns et la validité d'un hostname, utilisez la commande "**dig [nom d'hôte]**".

Le firewall est une couche de sécurité supplémentaire, il est principalement installé sur les routeurs afin d'empêcher des connexions entrantes non désirées sur le serveur.
Il est géré par Netfilter et est directement intégré au kernel.
Netfilter va donc filtrer les "**input**"*(paquets entrants)*, "**output**"*(paquets sortants)*, "**forward**"*(dans un routeur pour les paquets transférés)*.
Afin de manipuler Netfilter, il faut utiliser la commande "**iptables**". Cette commande est extrêmement riche et est donc très difficile à maîtriser sans l'utiliser régulièrement.

Il existe des ports bien connus à connaître pour chaque protocole, voici une liste des plus utilisés:

- **HTTP** (Web) = port 80
- **HTTPS** (Web sécurisé) = port 443
- **FTP** (Transfert de fichier) = port 20/21
- **SFTP** (Transfert de fichier sécurisé) = port 22
- **DNS** (Résolution de nom de domaine) = port 53
- **SSH** (Connexion à distance sécurisée) = port 22

Pour connaître les règles de trafic en vigueur sur votre réseau, utilisez la commande "**iptables -L -v**".
"**firewall-cmd**" est une surcouche de iptables et permet de gérer les input et output sur votre système.

Il est important de connaître le statut des autorisations de trafic sur le système pour firewall-cmd avec la commande "**firewall-cmd –list-all**".
La configuration de firewall-cmd est basée sur l'autorisation des services, ceux-ci sont visibles grâce à la commande "**firewall-cmd –get-services**".

Dans ces fichiers on retrouve l'ensemble des éléments nécessaires pour configurer le firewall.

Pour accorder l'autorisation d'un service et l'autoriser à communiquer à travers du firewall, il faut utiliser la commande "**firewall-cmd –add-service ssh**" pour la version run time et "**firewall-cmd –add-service ssh –permanent**" pour la version persistante.

Pour supprimer l'utilisation d'un service il faut remplacer add par remove.
Exemple: "firewall-cmd –remove-service ssh –permanent" pour la version persistante.

Après avoir imposé une nouvelle règle, il faut relancer le service avec reload.
Exemple: "firewall-cmd -reload".

Fin du résumé