Book Initiation — V1

Club*NIX

https://clubnix.fr

Table des matières

1 Linux

1.1 GNU-Linux

1.1.1 UNIX

Unix est un **systeme d'exploitation** multi-tâches et multi- utilisateurs créé en **1969**. Il repose sur un interprétateur (le shell) et sur plusieurs petits utilisateurs (avec des actions spécifiques selon son groupe).

C'est une **marque deposée de l'OpenGroup**. Son nom est un dérivé de "Unics" (*Uniplexed Information and Computing Service*). C'est un jeu de mot avec "Multics" (un autre noyau) qui vise à offrir simultanément plusieurs services à un ensemble d'utilisateurs.

Unix a donné naissance à une **famille de systèmes** comme BSD, GNU/LINUX, iOS ou encore MacOS, eux-même divisés en variantes de systèmes d'exploitation aux **normes POSIX**. Cette norme technique standardise les interfaces de programmation des logiciels.

Il faut savoir que la quasi-totalité des systèmes PC ou mobiles (à l'exception des systèmes Windows) sont basés sur le noyau de UNIX (y compris ceux de la marque Apple). Comme dit précédemment, Unix est un noyau, **il doit donc être utilisé avec d'autres logiciels** car il lui faut un système d'exploitation.

1.1.2 GNU : "GNU's Not UNIX"

GNU est un **projet des années 1990** lancé par Richard STALLMAN. C'est un **système d'exploitation libre** compatible avec le noyau UNIX. Il utilise des logiciels libres issus d'autres projets tels que le noyaux Linux (*voir GNU/LINUX*). Sa **philosophie** est de maintenir intacts les **tradictions hackers de partage** dans un monde de plus en plus marqué par l'empreinte du droit d'auteur. Il se bat donc pour une **libre diffusion des connaissances**. "GNU vise à ne laisser l'homme devenir ni l'esclave de la machine ni de ceux qui auraient l'exclusivité de sa programmation".

Il est **utilisable et partageable librement par tous**, ainsi chacun complète petit a petit l'architecture initiale de GNU pour le rendre meilleur. C'est dans ce contexte que le projet invite la communauté de hackers à le rejoindre et à participer à son développement. Il faut savoir qu'il est composé :

— d'un éditeur de texte (emacs),
— d'un compilateur très performant (gcc),
— d'un débogueur (gdb),
— d'un langage de script (bash),
— de bibliothèques de systèmes (glibc),
— (plus tard d'un noyau ramené par le projet Linux).

Malgré tous ces composants, GNU est toujours **incomplet** : son noyau est immature et incompatible avec certains de ses composants.

1.1.3 GNU/LINUX

GNU/LINUX a été créé en **1991**. Il est **initié par le projet Debian** et la naissance du noyau Linux. Il crédite donc à la fois Linux et GNU *mais l'usage de Linux est plus connu au grand public.*

Il est alors toujours basé sur le **mouvement du logiciel libre et du mode opératoire du hacker**. Cette association a eu lieu pour combler le vide causé par le développement inachevé de GNU (*voir GNU*).

Il est utilisé sur la plupart des téléphones portables comme sur les super-ordinateurs. Ce projet eut un grand impact dans le monde des serveurs informatiques. GNU/-LINUX veut casser le fait qu'à l'origine il fallait des connaissances solides en informatique pour utiliser un système d'exploitation (*pas d'interface graphique et besoin d'installer toutes les applications soi-même*).

Il a donc été un important vecteur de **popularisation du mouvement de l'open source**. Il a eu des **centaines de milliers de redistributions** avec des versions différentes pour plaire à tous les goûts (en fonction des besoins, configurations, sécurité, ...*voir différents OS*). GNU/LINUX a remplacé d'autres systèmes de type Unix et/ou évite l'achat d'une licence Windows (qui est très chère à l'achat).

Aujourd'hui on peut retrouver tous les équivalents des logiciels/applications qu'il y a sous Windows mais en Open Source.

1.2 Différents OS

En informatique, un système d'exploitation est un ensemble de programmes qui dirige l'utilisation des capacités d'un ordinateur par des logiciels applicatifs.

1.2.1 Différents types d'OS basés sur différents noyaux

- MacOS (série d'interfaces graphiques basée sur l'opération des systèmes développé par Apple pour leur Macintosh),
- iOS (système d'opérations pour mobile développé et distribué par Apple pour iPhone et iPod),
- **Linux** (c'est un Unix ; système d'opérations pour ordinateur assemblé sous le modèle du "Free and Open Source Software" ou *FOSS*),
- Android (c'est un dérivé de Linux ; système d'opération désigné principalement pour les mobiles tactiles comme les smartphones et les tablettes, initialement développé par Android),
- Microsoft Windows (série d'interfaces graphiques développée et commercialisée par Microsoft),
- BSD/OS (réputé pour réhabiliter le rôle des serveurs, organisé par les programmeurs d'Unix, utilisé pour un utilisateur personnel du web).

1.2.2 différentes familles d'OS Linux

- **Debian/Ubuntu** : *1993*

  - développé par SPI (*Software in the Public Interest*),
  - caractère non commercial et mode de gouvernance coopératif,
  - déjà installé avec son noyau, ses pilotes, son programme d'installation de distribution, ses logiciels "utiles" (pour le WiFi, un navigateur web, ...) ,
  - réunit une dizaine de sous-familles Debian avec le même noyau mais avec une architecture différente, dont les plus connues sont : KaliLinux, Kubuntu, Raspbian, Ubuntu, Xubuntu, etc...

- **Red Hat/Cent OS/Fedora** : *1994*
  - développé et distribué par l'entreprise Red Hat (entreprise, rachetée par IBM en 2018, dédiée aux logiciels Open source + distributeur de système d'exploitation Linux),
  - plusieurs grosses distributions sont issues de ce dev : Fedora, Enigma, ...
  - principalement destinées aux serveurs des entreprises,
  - voulait faire passer doucement les utilisateurs Windows sous Linux.

- **Arch** : *2002*
  - accent sur la simplicité et légèreté : parfait pour les utilisateurs avancés,
  - contribution ouverte (Open Source) tant que cela respecte sa philosophie.

- **Suse/OpenSuse** : *1994*
  - distribution communautaire et commerciale,
  - destinée à l'utilisation en entreprise mais toujours en Open Source,
  - cycle de développement long mais cycle de vie long,
  - disponible à la vente (licence et mise à jour).

2 Ecrire un document texte

2.1 LaTeX

2.1.1 Histoire et Principe

LaTeXa été créé en 1983, c'est un **langage et un système de composition de documents**. Il sert principalement à de la mise en page simple de documents et a pour but de séparer le fond et la forme.

LaTeXest devenu le langage privilégié pour les documents scientifiques du fait de sa simplicité.

Pour rédiger du LaTeX, il faut uniquement se concentrer sur la structure logique du document et son contenu car le logiciel s'occupe de la mise en page automatiquement. Vous pouvez écrire en LaTeXgrâce à différents logiciels d'éditeurs de texte dédiés comme TexMaker, Texlive, TeXworks, TexMacs, ... Dans ces éditeurs, il est possible de voir directement la mise en page en format PDF lorsqu'il est compilé. Mais il est également possible de manipuler LaTeXsimplement à partir d'un terminal ou de gedit par exemple.

L'évolution de LaTeXest assurée par la communauté d'utilisateurs qui regroupe des étudiants et des professeurs de mathématiques ou de physique, comme des musiciens et ingénieurs en informatique notamment.

2.1.2 Différences avec Word

Il faut d'abord savoir que les deux logiciels ne se comportent et ne s'utilisent pas de la même façon.

- La mise en page (images, figures, légendes, formules mathématiques, dessins, tableaux, ...) sur Word est une rude manipulation qui fait perdre du temps. LaTeXle fait tout seul, mais son interface austère fait "peur" aux débutants. Pourtant il suffit de lui dire quel type de documents on souhaite obtenir pour obtenir quelque chose de lisible et adapté avec les normes éditoriales ;
- LaTeXest gratuit (et même libre), contrairement à Word ;
- Les formules mathématiques sont simples d'écriture sur LaTeX ;
- Tout est modifiable et paramétrable avec LaTeXà n'importe quel moment (*si au bout de la 100 ème page vous vous rendez compte que vous voulez changer de police, uniquement sur tous les titres par exemple*) ;
- La gestion de documents longs est intuitive sur LaTeX, contrairement à la complexité sur Word lorsqu'il faut gérer la mise en page identique ;
- LaTeXpeut générer automatiquement des bibliothèques ou tables de matières beaucoup plus facilement que sur Word ;
- Accéder à la création des PDF rapidement sur LaTeX ;

2.1.3 Utilisation

Le système de balisage est assez semblable au langage HTML. Il est donc possible de **créer ou modifier des macro-commandes** afin d'ajouter des raccourcis. *Par exemple pour regrouper plusieurs instructions en une seule.* Comme ce langage a été

créé avant le Unicode, tous les caractères peuvent s'écrire en ASCII (mais ce n'est plus nécessaire).

Il faut tout d'abord configuration les packages à utiliser pour déterminer la langue d'écriture, les marges, polices d'écriture, couleur et taille de l'écriture, style du document, etc ... Tous ces packages **peuvent être trouvés documentés sur Internet**, vous pouvez donc simplement les importer.

Voici les principales instructions qui vous serviront :
— Partie : $\backslash part\{nom\_\ de\_\ la\_\ partie\}$
— Section : $\backslash section\{nom\_\ de\_\ la\_\ partie\}$
— Sous-section : $\backslash subsection\{nom\_\ de\_\ la\_\ partie\}$
— Paragraphe : $\backslash paragraph\{nom\_\ de\_\ la\_\ partie\}$
— Saut de ligne : $\backslash newlineou\ \backslash\ \backslash$
— Liste : $commencerpar\backslash begin\{itemize\}puispourchaquetiret\backslash itemetconclurepar\backslash end\{itemize\}$ *NB : pour modifier les puces il suffit de mettre la puces souhaitées entre [] apres* $\backslash item$
pour des listes numérotées il faut écrire $\backslash begin\{enumerate\}$
— très très petite écriture : $\backslash scriptsize$
— très petite écriture : $\backslash footnotesize$
— petite écriture : $\backslash small$
— grande écriture : $\backslash large$
— très grande écriture : $\backslash LARGE$
— très très grande écriture : $\backslash huge$
— Gras : $\backslash textbf\{\}$
— Italique : $\backslash textit\{\}$
— Penché : $\backslash textsl\{\}$
— Machine à écrire : $\backslash texttt\{\}$
— Exposant : $\backslash textsuperscript\{\}$
— Encadré : $\backslash fbox\{\}$
— Souligné : $\backslash ul\{\}$
— Barré : $\backslash st\{\}$

2.2 Markdown

2.2.1 Histoire

Le Markdown est un **langage de balisage** crée en 2004. Il est facile à manipuler, donc simple à écrire et à lire sans connaître les balises. Le Markdown peut être écrit sur n'importe quel éditeur de texte, il suffit, lorsque le document est prêt a être enregistré de nommer le document avec l'extension : "**.md**".

2.2.2 Instructions

Les instructions sont **très simples et peuvent être combinées**. On va voir quelques instructions de base :

Mise en forme du texte
— titre : # (rajouter des # par sous niveaux de paragraphes)
— commencer un paragraphe : mettre 4 espaces

— délimiter un paragraphe : sauter une ligne
— retour à la ligne : deux espaces à la fin de phrase

Police d'un texte

— Italique : encadrer le(s) mot(s) désiré(s) par * ou _
— Gras : le(s) mot(s) désiré(s) par **
— Souligner : encadrer le(s) mot(s) désiré(s) par ___
— Barré : encadrer le(s) mot(s) désiré(s) par ~~
— souligner les titres : mettre sur une ligne en dessous des = ou -

Ajout d'éléments au texte

— bloc de code : encadrer le(s) mot(s) désiré(s) par “‘
— citation : commencer par >
— liste : commencer par * ou - ou +
— liste ordonnée : commencer par 1. 2. 3. ...
— cases a cocher : [ ] ou [x]
— tableau :
  — délimiter les colonnes par : |
  — délimiter les titres des autres lignes par : :—— :
— lien :
  — en hypertexte : l'encadrer en <et >
  — en bouton : [*nom du bouton*](#).btn
— mage :![*texte*](*url de l'image*)

3 Terminal

3.1 Terminal

3.1.1 Quoi et pourquoi ?

Le terminal est un programme lançant une console permettant d'exécuter des commandes.

Les commandes permettent en une ligne de texte d'effectuer des opérations qui peuvent s'avérer très longues avec l'interface graphique. Par exemple, modifier les droits d'accès ou d'écriture à un fichier s'effectue en une dizaine de clics avec l'interface graphique, alors que la commande *chmod* avec les droits voulus et le nom du fichier le fait instantanément.

3.1.2 Ouvrir une console et s'en servir

Pour ouvrir une console de terminal, on peut :
— chercher terminal dans la barre de recherche ;
— avec le raccourci clavier disponible sur la plupart des environnements de bureau avec Ctrl + Alt + T.

Une première ligne apparaît, et est comme ceci :
— *nom d'utilisateur"@"nom du pc" :~$*

Tapez alors votre ligne de commande puis *Enter* pour l'exécuter.

Il existe de nombreux outils dans le terminal, que nous allons voir ici :

Arrêter une commande Il est possible de lancer une commande puis de l'arrêter manuellement sans attendre qu'elle se termine.

Par exemple, vous avez lancé une commande ping pour tester votre réseau. La commande ping ne s'arrête que si on lui demande. On peut alors taper la commande à travers le raccourci clavier suivant : *Ctrl + C* pour l'arrêter. Attention cependant. Même si sur la commande ping l'arrêt de la commande n'a pas d'impact, ce n'est pas le cas pour toutes les commandes. **Ctrl + C est une façon relativement moche d'arrêter un programme.**

Copier-coller Copier-coller une ligne de commande depuis un terminal est possible, mais pas avec les raccourcis claviers classiques. En effet, Ctrl + C est déjà un raccourci clavier du terminal. Il faut donc faire *Ctrl + Shift + C* pour copier une ligne et *Ctrl + Shift + V* pour coller. Vous pouvez aussi, pour les ordinateurs en disposant, sélectionner la ligne "en bleu" et cliquer sur le clic du milieu (ou la molette de la souris) afin de coller la ligne dans le terminal.

Il faut faire attention toutefois avec le copier coller depuis les forums. En effet, si vous copiez collez une suite de commandes avec des retours à la ligne comme par exemple :
— *ls*
— *cd dossier*
— *cat fichier*

Le terminal exécutera les deux premières commandes car elles sont séparées par un retour à la ligne. Cela peut être très pratique mais aussi dangereux.

3.1.3 L'auto-complétion

Certaines lignes peuvent être longues à taper. Le terminal met à disposition une touche permettant de compléter seul la fin de la commande. C'est la touche Tab. Après avoir tapé 3 lettres, vous pouvez demander l'auto-complétion. C'est le cas par exemple pour un nom de fichier très long ou de paquets. Il suffit alors de taper "cd début + Tab " et le terminal finira à votre place. Lorsque plusieurs fichiers ont le même début de nom, le terminal vous les proposera alors en dessous de votre ligne de commande.

Le manuel La plupart des commandes disposent d'un manuel, qui renseigne sur les paramètres de la commande, son utilité, ou encore comment l'utiliser. Pour ouvrir le manuel d'une commande, on tape dans le terminal *man + 'nom de la commande'*.

Retrouver une commande déjà tapée précédemment Pour retrouver une commande déjà tapée, on peut cliquer sur la flèche du haut. Un clic remonte d'une commande.

De ce fait, si vous souhaitez taper une commande très longue et que vous avez déjà tapée il y a quelques temps, cliquez sur la flèche du haut autant de fois que nécessaire pour la retrouver.

Il existe aussi une commande, plus lourde, *history* qui affiche les 500 dernières commandes tapées. Elle est lourde, mais très pratique couplée avec *grep*.

3.1.4 Les droits SUDO

Pour exécuter certaines commandes, notamment installer des paquets ou redémarrer la machine, le terminal a besoin des droirts super-utilisateur. Ce sont les droits SUDO, pour Super Utilisateur DO. Les droits sudo concernent les commandes administrateurs systèmes. Quiconque détiendrait les droits sudo pourrait passer des commandes de bas niveau capables de modifier gravement la configuration même, donc influer sur le comportement de la machine.

En accordant les droits sudo à une commande, elle est alors capable de tout. Par exemple d'installer un programme : *sudo apt install nom_ du_ programme*, modifier un fichier de configuration ...

Pour lancer une commande administrateur système, il faut taper *sudo + nom_ de_ la_ commande*. Le terminal va alors vous demander votre mot de passe administrateur avant de lancer la commande. Si le mot de passe ne s'affiche pas, ni même des astérisques ou autre, c'est normal, c'est pour renforcer la sécurité car personne ne saura ne serait-ce que la taille de ce mot de passe. Attention, vous êtes le seul responsable de votre machine et lancer des commandes sudo sans être sûr de leur impact pourraient complètement détruire votre machine.

3.1.5 Les questions des commandes

Certaines commandes vous posent des questions, comme par exemple lors de l'installation d'un paquet ces questions sont de la forme :

— 'question' [Y/N]. Il faut alors taper Y (pour yes) ou N (pour no) puis entrer afin de répondre à la question.

— 'question' [Y/n]" ou "'question' [y/N]" C'est une variante dans laquelle vous pouvez toujours taper y ou n mais aussi directement entrée. La réponse prise en compte sera celle en majuscule.

Il existe un poly regroupant les principales commandes terminal disponibles sur le GitHub du club.

3.2 Bash

3.2.1 Bash et scripts

Le bash est un interpréteur de script natif aux systèmes d'exploitation sous environnement Unix.

A quoi ça sert ? Le bash permet l'exécution de scripts, petits morceaux de code, souvent exécutés au démarrage, afin d'effectuer une tâche précise. Par exemple, l'ensemble des applications s'ouvrant au démarrage sont regroupées dans un script. Bash permet l'interprétation des commandes shell. Une commande shell est une chaîne de caractères en minuscules qui peut être invoquée au travers d'un invite de commande ou d'un script. Des options et des arguments peuvent la compléter.

Le bash propose différents traitements des commandes :

— Les commandes simultanées : com1 & com2 & com3 & ... & comN. Les commandes com1 à comN sont exécutées simultanéments ;

— Les commandes successives : com1 ; com2 ; com3 ; ... ; comN. Les commandes com 1 à comN sont exécutées les unes après les autres ;

— etc,...

Il existe deux modes de fonctionnement du bash :

— Le mode interactif : Le terminal fonctionne avec le bash. lorsque l'on tape une commande manuellement dans le terminal, on crée en réalité un mini script d'une ligne, interprété par bash puis executé ;

— Le mode batch : bash exécute automatiquement un script contenu dans un fichier texte contenant les commandes à utiliser. Ce fichier doit être exécutable (réglable avec chmod).

L'interprétation d'une ligne de commande Chaque interprétation d'une ligne de commande d'un script repose sur des codes bien stricts :

— Le premier mot de la ligne est interprété comme le nom de la commande ;

— Chaque mot est séparé par un ou plusieurs caractères de séparation (espace, tabulation, tiret, underscore ...) ;

— La fin de la ligne se termine par un ' ;', comme dans la plupart des langages de programmation, ou un retour à la ligne

Pour plus d'informations sur les lignes de commandes bash, il existe un poly sur les commandes du terminal disponible sur le GitHub du club.

4 SSH

4.0.1 Définition

SSH signifie "Secure Shell", c'est un protocole sécurisé pour les communications à distance créé en 1995.

Pour rappel, un Shell va permettre de dialoguer avec une machine ou un serveur (grâce au terminal qui est une application graphique du Shell) via l'exécution de différentes commandes qui retourneront des informations. Il existe toutes sortes de Shell mais le plus utilisé reste bash.

Un shell va donc nous permettre d'administrer nos serveurs Linux, en local, c'est-à-dire lorsque l'on se trouve physiquement en face de notre serveur, mais aussi à distance ! Notamment grâce au Secure Shell (SSH).

L'administration à distance est aujourd'hui vitale lorsque l'on gère un seul serveur, comme des milliers qui sont la plupart du temps difficiles d'accès car ils sont stockés dans des datacenter et isolés géographiquement.

Autrefois, d'autres protocoles étaient utilisés pour accéder à distance à un serveur Linux. Le protocole Telnet a pendant longtemps été utilisé, il permet également d'accéder à distance à une machine Linux, mais Telnet est aujourd'hui délaissé au profit de SSH et cela pour une raison très simple : son manque de sécurité. A l'inverse, SSH va créer un tunnel sécurisé entre l'utilisateur et la machine (il va crypter les données).

Comme dit précédemment nous utilisons SSH surtout pour faire le lien entre nos machines et nous.

4.0.2 Les Commandes De Bases

- Toutes les commandes du terminal sont applicables
- Connexion à la machine distante avec le login john 'ssh john@remotehost.example.com' ou 'ssh -l john remotehost.example.com'
- Copie de la clé publique sur la machine distante 'ssh-copy-id -i ~/.ssh/id_dsa.pub john@remotehost.example.com'
- la commande **scp** va permettre de copier des fichiers entre le serveur et le client ssh de manière sécurisée **scp my_file john@remotehost.example.com :/home/john/a_folder**
  pour un seul fichier et **scp -r my_folder john@remotehost.example.com :/home/john/another_folder** pour un dossier (ajout de l'argument de récursivité). Il est indispensable de préciser le dossier de destination (après les ' :').

5 Exemples de logiciels libres

Tout logiciel libre est défini par la **licence publique générale GNU** (*appelé GPL*) qui rend les logiciels indépendants de tous éditeurs en les encourageant à l'entraide et au partage.

La licence générale est définie par 4 lois principales :
— pouvoir utiliser un logiciel sans restrictions,
— pouvoir étudier le logiciel,
— pouvoir modifier pour l'adapter aux besoins des utilisateurs,
— pouvoir redistribuer sous certaines conditions précises.

Cela a beaucoup d'avantages comme la possibilité de **corrigé rapidement des bogues et des failles de sécurité**.

Attention, un logiciel libre n'est pas nécessairement gratuit et inversement un logiciel gratuit n'est pas forcement libre !

5.1 Git

5.1.1 Définition

Git est le logiciel de gestion de version décentralisé (pas forcément besoin d'être en ligne) le plus couramment utilisé dans le monde (il en existe d'autres comme SVN mais beaucoup moins utilisés), il fût développé en 2005 par Linus Torvalds (créateur du noyau linux).

Git est un outil de versionnement, c'est à dire que lorsque plusieurs personnes sont sur un même projet, Git va garder les différentes versions dans le temps.

En réalité, Git va conserver les modifications effectuées, ce qui empêche de prendre trop de place.

Un projet constitue plusieurs fichiers amenés à évoluer dans le temps. Git va conserver les informations sur Qui a modifié Quoi, Quand et Pourquoi. Cet outil est donc primordial durant la conception d'un projet à plusieurs et même seul (lorsque l'on revient sur du code écrit il y a 3 mois on ne sait pas toujours à quoi il correspond).

Attention, il ne faut pas confondre Git avec GitHub et GitLab. GitHub et GitLab vont se servir de Git mais ils vont aussi rajouter une interface graphique et vont permettre le stockage sur un serveur distant. De plus il ne faut pas confondre Github et Gitlab, Github est géré par une entreprise privée qui entre temps a été rachetée par Microsoft (entreprise vouée à l'échec) et l'autre est une entreprise qui fourni aussi son logiciel en Open Source.

5.1.2 Commandes de base

Voir le poly Git du club qui est sur le Github.

5.2 F-Droid

F-Droid est un magasin d'application (*au même titre que le PlayStore ou l'AppleStore*) pour smartphones Android qui met à disposition plus de **12 000 appli-**

cations libres et gratuites.

Il a été créé en 2010 et est promu par la Free SoftWare Foundation Europe. Son architecture de sécurité est basée sur le modèle de Debian. Comme toute construction Open Source, F-Droid est tenu et développé par un grand nombre de contributeurs faisant partie de la communauté. Ainsi, chaque personne peut créer sa propre application et la mettre à disposition gratuitement sur ce magasin. F-Droid assure tout de même un niveau de sécurité, les applications publiées sont vérifiées.

L'avantage d'utiliser F-Droid plutôt que le magasin d'application par défaut est principalement la préservation de vos données. Mais aussi la **sécurité** de vos applications et le fait qu'il n'est pas nécessaire de s'identifier pour pouvoir télécharger des nouvelles applications.

Ce magasin facilite la découverte et l'installation de multiples applications. De plus, contrairement à la plupart des applications, vous n'êtes pas obligé de faire les mises à jour, et pouvez ainsi garder une ancienne version.

6 Infrastructure réseau

6.1 LDAP

6.1.1 Le LDAP, ou *Lightweigth Directory Acess Protocol*

Lorsqu'un utilisateur tente de se connecter à un ordinateur, le LDAP reçoit une combinaison login/mot de passe et donne ensuite ou non l'autorisation de se connecter.

6.1.2 Introduction

Le LDAP est un protocole créé en 1995, succédant au protocole DAP et permettant l'accès et la modification de base de données sur les utilisateurs d'un reseau. Un LDAP sert notement à se connecter ou se déconnecter d'un serveur hébergant le LDAP afin d'y être identifié, mais aussi à chercher des informations, les comparer, ajouter des utilisateurs...

6.2 Le LDAP du club Nix ou de l'école

Le LDAP du Club*Nix ou de l'école permet de se connecter et de s'identifier afin de récupérer ses informations, quelle que soit la machine utilisée.

6.2.1 Les entrées

Un LDAP définit l'accès aux entrées (ou utilisateurs la plupart du temps). Le LDAP ne peut gérer que des entrées. Une entrée peut être un nom d'utilisateur, un périphérique ou encore des paramètres. Il existe 2 types d'entrées, les entrées normales et les entrées opérationnelles :

- Les entrées classiques, telles que le nom d'utilisateur ou la date d'anniversaire sont des entrées dites classiques.
- Les entrées opérationnelles, tels que les paramètres, les dates de modification, qui ne sont accessibles et utilisables uniquement par le serveur.

Une entrée est définie par son nom, ou DN pour *Distinguished Name*, composé d'une série de clés et de valeurs de ces clés. Par exemple, la clé *uid* définit le nom d'utilisateur et la clé *cn* définit le nom. Pour ces deux clés, une entrée serait sous la forme : *uid=utilisateur,cn=nix*

6.2.2 L'arborescence

Un serveur hébergeant un LDAP est organisé selon une arborescence, comme un système de fichiers, dans lequel chaque branche correspond à une entrée. Une branche située à la racine sera appelée racine ou root en anglais. Le schéma d'une clé correspond à l'ensemble des valeurs des attributs ou valeurs attribuées aux clés. Les annuaires LDAP répondent à certaines règles de structure :

- Un annuaire est un arbre d'entrées ;
- Une entrée est constituée d'un ensemble d'attributs ;
- Un attribut possède un nom, un type et une ou plusieurs valeurs ;
- Les attributs sont définis dans des schémas.

6.2.3 Chercher des informations d'un LDAP

Le protocole LDAP fournit un ensemble de fonctions permettant d'interroger le serveur sur lequel est hébergé le serveur LDAP afin de modifier, ajouter ou supprimer des entrées. On peut citer notamment *add* pour ajouter une entrée, *delete* pour la supprimer, ou *rename* pour la renommer, afin de modifier l'arborescence du LDAP.

On utilise essentiellement les LDAP grâce à des interfaces graphiques ou à d'autres logiciels qui utilisent un LDAP.

6.3 NFS

6.3.1 Introduction

Un NFS, ou *Network File Systeme*, à traduire par *Système de fichiers en réseau*, est un protocole permettant de partager et récupérer des données via un réseau. Les réseaux utilisant un NFS permettent l'utilisation des fichiers sauvegardés sur le serveur hébergeant le NFS à distance, tant que la connexion est établie. En pratique, on peut récupérer ses données sauvegardées sur n'importe quelle machine.

6.3.2 Le NFS du Club*Nix ou de l'ESIEE

Le Club*Nix et l'ESIEE possèdent un NFS, permettant de se connecter à partir de son identifiant LDAP (voir chapitre LDAP), et de récupérer ses données personnelles, quelle que soit la machine utilisée.

6.3.3 Architecture d'un NFS

Un NFS suit un modèle d'architecture classique réseau : c'est à dire que chaque client (utilisateurs) se voit accordé un espace de stockage maximal, exactement comme pour un système de ficher local. Par la suite, les données sont sauvegardées sur le serveur hébergeant le NFS, disposant d'une bien plus grosse capacité de stockage. Chaque demande d'accès ou de sauvegarde de fichier passe par le réseau.

6.3.4 NFS et VFS

Sous Linux, un protocole permet de prendre en charge plusieurs systèmes de fichiers différents sur un même serveur : c'est le VFS. Le système VFS détermine le stockage auquel une demande est effectuée. Lorsqu'une demande s'avère être destinée au NFS, VFS la transmet au noyau (appelé kernel) du NFS. Le NFS interprète alors la requête d'entrée ou sortie et la traduit en procédure exécutable par le protocole NFS. On peut notamment citer les procédures suivantes :

— OPEN
— ACCESS
— CREATE
— READ
— CLOSE
— REMOVE
— ...

Le serveur NFS va alors satisfaire la demande de l'utilisateur en lui "renvoyant" ou "effectuant" sa demande. En soi, NFS n'est pas un système de fichier au sens propre mais un protocole réseau permettant d'accéder à des fichiers à distance via un réseau.

6.3.5 La sécurité du NFS

Il existe actuellement 4 versions de NFS. Dans les 3 premières versions, le protocole NFS n'était pas sécurisé et permettait l'utilisation en local, comme dans une école ou au Club Nix par exemple. La dernière version (v.4) est doté d'un système de chiffrement comprenant la négociation du niveau de sécurité entre client et serveur, une sécurisation simple mais efficace, ainsi qu'un chiffrement des communications. La version 4.1 du système est actuellement en cours de développement et n'est pas prévue avant plusieurs années.

6.4 Fichiers logs

6.4.1 Définition

Dans le domaine informatique, le terme log désigne un type de fichier, ou une entité équivalente, dont la mission principale consiste à stocker un historique des événements. Diminutif de logging, le terme peut être traduit en français par "journal". Le log s'apparente ainsi à un journal de bord horodaté, qui ordonne les différents événements qui se sont produits sur un ordinateur, un serveur, etc. Il permet ainsi d'analyser heure par heure, voire minute par minute, l'activité interne d'un processus.

Chaque action d'un système informatique (ouverture d'une session, installation d'un programme, navigation sur Internet...) produit un fichier log. Diminutif de logging, le terme peut être traduit en français par "journal" . Ces fichiers textes listent chronologiquement les événements exécutés. Ils s'avèrent utiles pour comprendre la provenance d'une erreur en cas de bug. Ils permettent également d'établir des statistiques de connexions à un site Web ou à un serveur.

S'agissant d'un serveur Web, un fichier log va par exemple enregistrer la date et l'heure de la tentative d'accès, l'adresse IP du client, le fichier cible, le système d'exploitation utilisé, le navigateur, la réponse du serveur à cette requête, éventuellement le type d'erreur rencontré...

Les fichiers logs peuvent contenir des informations confidentielles
- L'heure et date de consultation
- Le nombre de consultations
- La durée de la session
- L'adresse IP et le nom d'hôte de l'utilisateur
- Les informations sur le client demandeur (en général le navigateur)
- Le moteur de recherche utilisé, dont les requêtes
- Le système d'exploitation utilisé
- Une entrée classique d'un fichier log d'un serveur Web se présente comme ci-dessous : *183.121.143.32 - - [18/Mar/2003 :08 :04 :22 +0200] "GET /images/logo.jpg HTTP/1.1" 200 512 "http ://www.wikipedia.org/" "Mozilla/5.0 (X11 ; U ; Linux i686 ; de-DE ;rv :1.7.5)"*

6.5 Virtual Machine

6.5.1 Définitions

La Virtualisation Au sens large, la virtualisation consiste à simuler l'existence de plusieurs machines informatiques en utilisant une seule. Ceci permet en particulier de diminuer les coûts d'achat de matériel informatique et de rentabiliser leur utilisation. Exemples de logiciels de virtualisation : VMware, VirtualBox...

Un Hyperviseur

— assure le contrôle du processeur et des ressources de la machine hôte
— Alloue à chaque machine virtuelle (VM) les ressources dont elle a besoin
— S'assure que ces VM n'interfèrent pas l'une avec l'autre
Il existe deux types d'hyperviseurs : le type 1 et le type 2

Définition d'une machine virtuelle Une machine virtuelle est un fichier informatique, généralement appelé image, qui se comporte comme un ordinateur réel.

En d'autres termes, il s'agit d'un ordinateur créé à l'intérieur d'un ordinateur. Elle s'exécute dans une fenêtre, comme tout autre programme, en offrant à l'utilisateur final une expérience identique à celle qu'il aurait sur le système d'exploitation hôte.

La machine virtuelle est placée dans un « bac à sable » qui l'isole du reste du système, de sorte que les logiciels installés sur la machine virtuelle ne peuvent ni s'échapper, ni modifier l'ordinateur hôte. Cela produit un environnement idéal pour tester d'autres systèmes d'exploitation, dont des versions bêta, l'accès à des données infectées par des virus, la création de sauvegardes de système d'exploitation et l'exécution de logiciels ou d'applications sur des systèmes d'exploitation auxquels ils ne sont pas destinés à l'origine. Il est possible d'exécuter plusieurs machines virtuelles simultanément sur un même ordinateur physique. Pour les serveurs, les divers systèmes d'exploitation fonctionnent côte à côte, avec un composant logiciel appelé hyperviseur (logicielle de virtualisation) pour les gérer, alors que les ordinateurs de bureau classiques n'utilisent qu'un seul système d'exploitation pour exécuter d'autres systèmes d'exploitation dans des fenêtres de programme qui leur sont propres.

Chaque machine virtuelle demande son propre matériel virtuel, à savoir les processeurs, la mémoire, les disques durs, les interfaces réseau et les autres périphériques nécessaires. Le matériel virtuel est ensuite mappé au matériel réel sur la machine physique, ce qui permet de réaliser des économies en réduisant le besoin de disposer de systèmes matériels physiques, ainsi que les coûts de maintenance associés, tout en réduisant la demande en alimentation et refroidissement.

VirtualBox est une application de virtualisation x86 qui facilite la création de machines virtuelles mais pas très conseillé d'utilisation, il est préférable d'utiliser Docker(lxd) qui est très léger et permet d'éviter une surcharge Sinon nous avons KVM qui lui est un peu plus lourd mais va plutôt se rapprocher de virtual Box.