

Laboratoire d'administration système

## Partie « Linux »

par Louis SWINNEN

---

Ce cours est soumis aux droits d'auteur. Ainsi, il ne peut être reproduit, traduit, ou transmis sous quelque forme que ce soit sans l'autorisation préalable et écrite de l'auteur. Conformément aux règlements en vigueur à HELMo, une licence gratuite est concédée à tout étudiant inscrit et suivant régulièrement ce cours, tant que l'enseignant garde, dans ses attributions, le cours concerné.

---

Copyright © Louis SWINNEN, Tous droits réservés, 2015

---

Septembre 2015

## 1. Introduction

Dans cette partie du cours, nous allons aborder des notions importantes en administration système sous GNU/Linux. Pour les aspects pratiques, nous utiliserons le logiciel VMware Workstation<sup>1</sup> ainsi que le système d'exploitation GNU/Linux CentOS 7<sup>2</sup>.

Suite logique des cours de *systèmes d'exploitation et réseaux*, des notions importantes dans ces domaines sont requises pour pouvoir aborder le cours d'administration système sereinement.

Dans ce cours, nous aborderons les bases du système (gestion des utilisateurs, administration du disque dur, configuration réseau, scripting PERL) ainsi que les principales tâches dédiées aux serveurs Linux : serveur Web (hébergement de sites web PHP avec MySQL), serveur FTP, serveur DNS, accès à distance, firewall et sécurisation du système.

### 1.1 Pourquoi CentOS 7 ?

Le choix d'une distribution Linux est, avant tout, *une question de goût*. Il existe de nombreuses distributions : certaines très à la mode comme Ubuntu, d'autres commerciales comme RedHat, d'autres totalement libres et un peu moins accessibles comme Debian. Le choix de CentOS est, avant tout, un choix de raison : cette distribution est **gratuite**, orientée **serveur** et **est compatible avec les versions RedHat® Enterprise Linux**.

Le choix s'impose donc assez aisément puisque cette distribution est très proche d'une version commerciale, disponible avec support et assez répandue dans le milieu professionnel.

Enfin, passer d'une distribution à une autre demande un peu de travail même si les bases restent les mêmes. Ainsi chaque distribution propose son système de *packages* (distribution de logiciels compilés), ses fichiers de configuration et sa sélection de programmes. En effet, sous Linux, il n'est pas rare de trouver beaucoup de programmes répondant aux mêmes attentes (par exemple le serveur FTP : il y a *ftpd*, *proftpd*, *wu-ftp* ou encore *vsftpd*). La distribution propose, en standard, une version du service avec lequel la configuration est proposée.

### 1.2 Cours, Notes, ECTS

Le cours d'*administration système et réseau* est fractionné en 2 parties distinctes pour le parcours *standard* : la première reprenant les notions d'administration sous Linux et la seconde reprenant les notions d'administration sous Windows Server.

Pour les étudiants inscrits au parcours Salto, le cours est limité aux notions d'administration sous Linux.

#### 1.2.1 Le parcours standard

Le cours fait **70h** et **6** crédits ECTS. La partie *administration système Linux* reprendra :

---

<sup>1</sup> Une version plus limitée mais gratuite, appelée VMware View est disponible sur le site de VMware

<sup>2</sup> A l'instar de nombreuses distributions Linux, CentOS GNU/Linux est disponible gratuitement sur internet.

- Un cours de **35h** (uniquement au laboratoire)
- Une note comptant **pour la moitié des points** dans le cours complet.
  - Administration Système
    - Partie Windows, pondération 50 %
    - Partie Linux, pondération 50 %
- La proportion note année, note examen est la suivante : **pas de note année + 100 % note examen.**

### 1.2.2 Le parcours Salto

Le cours fait **35h** et **3** crédits ECTS. La partie *administration système Linux* reprendra :

- Un cours de **35h** (uniquement au laboratoire)
- Administration Système
  - Partie Linux, pondération 100 %
- La proportion note année, note examen est la suivante : **pas de note année + 100 % note examen.**

Il convient de se référer à la **fiche UE officielle** du cours pour ce point précis.

### 1.3 Examen

L'examen d'administration système, partie Linux a lieu, en **1<sup>ère</sup> session**, en **janvier**. La seconde session est programmée en août.

Il s'agit d'un **examen pratique à cours ouvert** dans lequel une configuration particulière vous sera demandée et vous disposerez d'un temps défini pour la réaliser. La configuration demandée peut reprendre n'importe quel aspect du cours et inclura l'écriture de scripts.

La seule manière de vous entraîner à ce cours est d'y participer activement en réalisant les exercices proposés à chaque séance, ne prenant pas de retard par rapport à la théorie et en posant des questions si le besoin s'en fait ressentir.

La matière est conséquente et une bonne préparation est vraiment nécessaire à la maîtrise des concepts vus ici.

## Leçon 1 : Introduction à Linux

Dans cette leçon, nous allons découvrir le système Linux. Beaucoup d'information seront données dans cette leçon, certaines seront des révisions du laboratoire de système d'exploitation.

Avant de commencer, une référence intéressante pour cette leçon est :

[RHELC] T. BARTOLONE, Red Hat Enterprise Linux CentOS – Mise en production et administration de serveurs, 2<sup>ème</sup> édition, Editions ENI, février 2015

### 1.1 Déploiement de la machine virtuelle

Une machine virtuelle toute prête est disponible. Elle est déjà installée et certains outils ont déjà été ajoutés pour faciliter l'apprentissage. Le lien vers la machine virtuelle se trouve sur la page web décrivant le cours.

Pour l'utiliser dans VMware, il faut décompresser le fichier dans le dossier c:\admsys. Une fois décompressée, il faut démarrer VMware et choisir les options **File > Open** et sélectionner la machine virtuelle décompressée.

Avant de démarrer la machine virtuelle, n'oubliez pas de démarrer **le firewall pfSense** afin de connecter votre machine Linux à internet.

#### 1.1.1 Connexion au réseau

Avant de commencer, il est nécessaire de bien comprendre la connexion au réseau au travers de VMware.

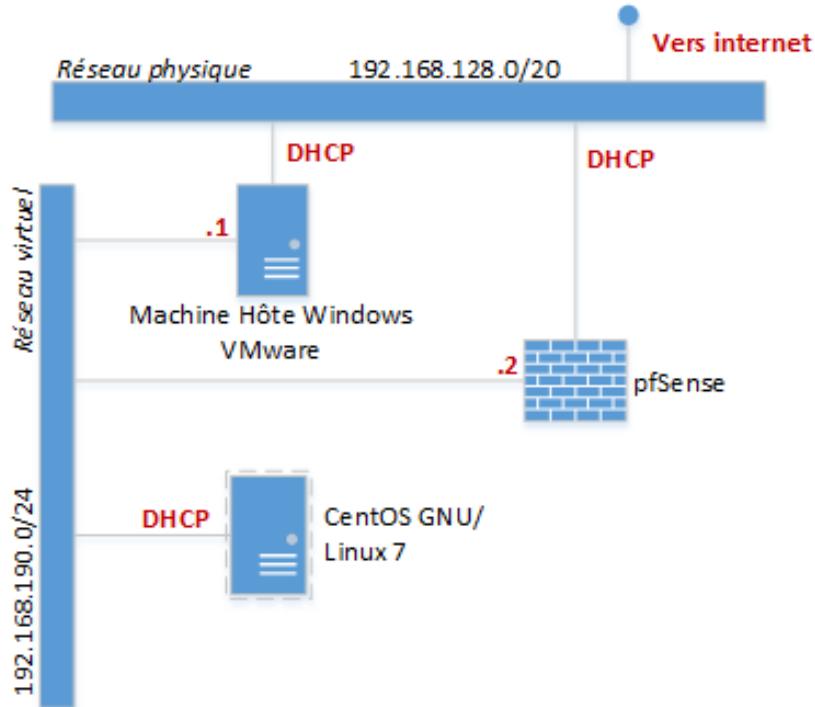


Figure 1.1 : Le réseau virtuel

Comme nous pouvons le voir sur la figure 1.1, il y a *le réseau physique* connecté au réseau de la haute école (ou de votre domicile), lui-même relié à Internet. Lors de l'installation de VMware, celui-ci ajoute des *réseaux virtuels*. Le réseau virtuel peut être utilisé pour interconnecter les machines virtuelles entre-elles et les relier à Internet.

Ainsi, sur le réseau configuré sur le campus Guillemins, nous avons le réseau virtuel **192.168.190.0/24** qui est dédié aux machines virtuelles. Il est configuré **de la même manière sur chaque machine qui tourne VMware** et cela ne pose aucun souci puisque ce réseau est *privé* et contenu sur la machine exécutant VMware.

Afin de connecter les machines virtuelles à Internet, un firewall gratuit virtuel, appelé pfSense, est installé. Il permet de relier le réseau virtuel au réseau physique. Il assure **également une complète isolation du réseau virtuel et du réseau physique**. Ainsi, si vous faites des erreurs de configuration sur vos machines virtuelles, cela n'aura pas d'impact sur le réseau du Campus Guillemins.

Voici un résumé des connexions prévues :

Machine	Réseau physique	Réseau virtuel
Machine Windows exécutant VMware	IP via DHCP	192.168.190.1
Firewall pfSense	IP via DHCP	192.168.190.2
Serveur virtuel CentOS	<b>Aucune connexion</b>	IP via DHCP

### Configuration de VMware à domicile

Si vous souhaitez travailler ou faire fonctionner les machines virtuelles chez vous, il faut reproduire la configuration virtuelle. Pour ce faire, il y a 2 étapes à accomplir :

1. Via l'outil Virtual Network Editor installé avec VMware (accessible depuis le menu **Edit** de VMware Workstation), il faut vérifier les paramètres pour **VMnet1** :

- VMnet Information : Host-Only
- Vérifier que l'option *Connect a host virtual adapter to this network* est **activée**
- Vérifier que l'option *Use local DHCP service to distribute IP address to VMs* est **désactivée**
- Configurer les paramètres réseaux comme suit : *Subnet IP* : **192.168.190.0** et *Subnet mask* : **255.255.255.0**

Une fois cette étape terminée, votre configuration devrait ressembler à celle présentée sur la figure 1.2.

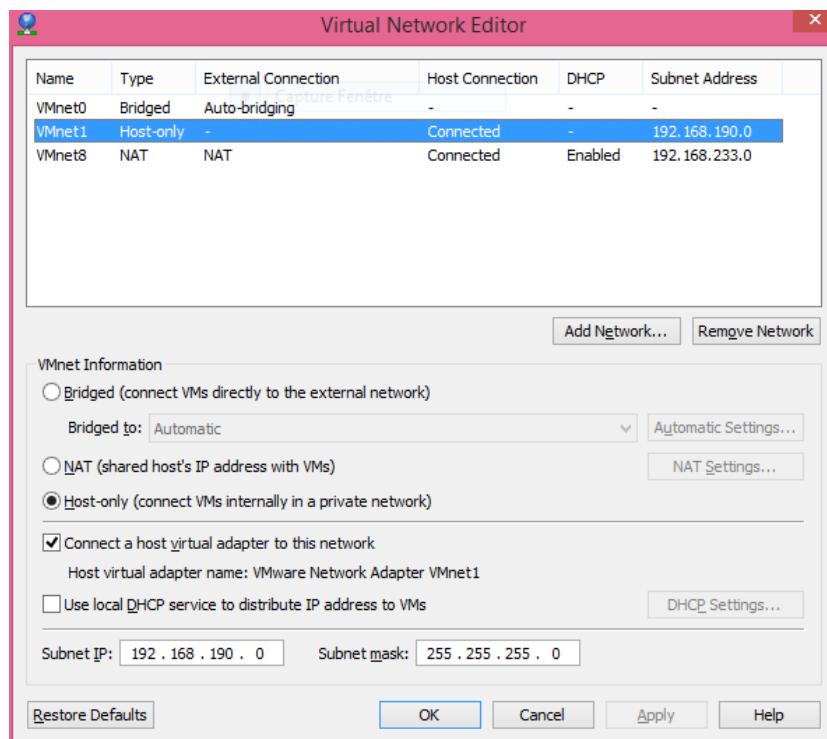


Figure 1.2 : Outil Virtual Network Editor

- Pour la seconde étape, il faut ouvrir **le centre de réseau et de partage** et choisir l'option à gauche **Modifier les paramètres de la carte**, et faire un clic-droit sur l'interface **VMware Network Adapter VMnet1** et choisir **Propriétés**.

Il faut ensuite choisir **Protocol Internet version 4 (TCP/IPv4)** et choisir **Propriétés**.

Il faut alors vérifier que :

- Utiliser l'adresse IP suivante est bien sélectionnée.
  - Adresse IP : 192.168.190.1
  - Masque de sous-réseau : 255.255.255.0

Ce sont les seuls paramètres qui doivent être configurés

Une fois ces 2 étapes achevées, la configuration du réseau virtuel devrait être fonctionnelle et votre serveur virtuel devrait fonctionner aussi bien sur les machines de l'école que sur votre machine personnelle.

### 1.1.2 Accès au firewall pfSense

Le firewall pfSense dispose d'une interface web permettant de configurer celui-ci. Nous aurons, dans la suite du cours, l'occasion de revenir sur ses paramètres. Notons surtout que pfSense permet une isolation complète entre le réseau virtuel et le réseau du campus.

Pour y accéder, il faut simplement se connecter à l'adresse : <http://192.168.190.2>

- Login : admin
- Mot de passe : rootroot

### Ouverture de ports dans pfSense

Pour ouvrir des ports dans pfSense, il faut se connecter à l'interface web de configuration du firewall et puis aller dans le menu **Firewall > NAT**.

Supposons que nous souhaitions ouvrir et rediriger le port 21 (=FTP) et tous les ports entre 15000 et 15500 vers notre machine virtuelle. Nous allons devoir ajouter 2 règles<sup>3</sup> pour permettre la connexion sur ces ports depuis le réseau de l'école :

1. Interface : WAN ; Protocol : TCP ; Destination : WAN address ; Destination port range from : (other) 15000 to (other) 15500 ; Redirect target IP : <ip de votre serveur Linux> ; Redirect target port : 15000 ; Cliquer sur SAVE
  
2. Interface : WAN ; Protocol : TCP ; Destination : WAN address ; Destination port range from : FTP to FTP ; Redirect target IP : <ip de votre serveur Linux> ; Redirect target port : FTP ; Cliquer sur SAVE

Cliquer sur **Apply Changes**

Les ports en question sont désormais redirigé vers la machine virtuelle. Ainsi, toute connexion sur un de ces ports en utilisant l'adresse IP extérieur provoquera la connexion vers la machine virtuelle sur ces mêmes ports.

## 1.2 Présentation de Linux

Linux est un système d'exploitation particulier : en effet, il s'agit d'un système d'exploitation *libre*. C'est à dire que, parmi les droits qui sont concédés aux utilisateurs, il y a la *liberté de modifier* le système et de l'adapter comme souhaité. Linux est d'ailleurs présent dans beaucoup d'environnements embarqués. Ainsi, beaucoup de « box internet » utilisent Linux pour interconnecter votre réseau personnel à celui de votre fournisseur internet (FAI ou ISP).

Outre la liberté, Linux est également un système extrêmement stable. Ce qui le rend particulièrement adapté à l'exécution de programmes serveurs. Ainsi, beaucoup de services critiques sur Internet, comme le DNS ou le mail, utilisent abondamment des serveurs Linux.

On trouve également beaucoup de serveurs LAMP (acronyme de Linux Apache MySQL Php) permettant de faire tourner des sites web dynamiques.

Pour **trouver de l'aide** sous Linux, Internet est une véritable mine d'information. Il ne faut pas négliger non plus **les pages de manuels** qui présentent toutes les commandes et les options :

```
$ man ls
```

Affiche la page de manuel pour la commande `ls`. La navigation dans le manuel est simple : `<espace>` permet de passer à la page suivante, `b` à la page précédente et `q` permet de quitter. Les flèches du clavier peuvent également être utilisées pour naviguer dans le manuel.

---

<sup>3</sup> Les ports proposés sont donnés à titre d'exemple (21 et ceux compris entre 15000 et 15500)

### 1.2.1 Disque, partition, LVM et système de fichiers

Un disque dur est un espace disponible pour le système. On trouve aujourd’hui des disques durs jusqu’à 4 To, bon marché. Afin d’exploiter efficacement cet espace, il est possible de *créer des partitions* (i.e. découpe logique du disque en plusieurs unités). Comme tout périphérique, le disque dur est renseigné dans le dossier `/dev`. Le premier disque dur SATA ou SCSI porte le nom `/dev/sda`, le second `/dev/sdb`, et ainsi de suite.

La gestion des partitions se fait soit en utilisant MBR (Master Boot Record - implémenté en 1983) ou GPT (GUID Partition Table - développé début 2000). Les deux technologies sont toujours présentes aujourd’hui même si l’antique MBR tend à disparaître avec l’apparition des BIOS compatibles UEFI et les disques durs de plus en plus gros. Ce qu’il faut simplement retenir est qu’il est possible d’utiliser l’un ou l’autre système et que la manière dont Linux interprète les informations est simple.

En mode MBR, on peut découper un disque en partition primaire (4 maximum) ou étendue (3 primaires et 1 étendue). Une partition étendue peut contenir autant de partitions logiques que nécessaire. De plus, il est possible d’installer Linux sur n’importe quel type de partition. La taille maximale du disque est de 2 To. Ainsi, en Linux, la 1<sup>ère</sup> partition principale sera renseignée par `/dev/sda1` (si elle se situe sur le 1<sup>er</sup> disque), `/dev/sda2` pour la seconde et ainsi de suite. Les partitions logiques commencent avec `/dev/sda5`.

En mode GPT, il est possible de créer autant de partition que souhaitée. Il n’y a pas de limite quant au nombre, ni à l’espace alloué. Les disques modernes (de plus de 2 To) sont supportés sans problème. Avec ce mécanisme, les partitions sont identifiées à l’aide d’un identifiant global et sont ensuite numérotées par le système `/dev/sda1`, `/dev/sda2`, ....

#### *Et LVM dans tout ça ...*

Logical Volume Manager ou LVM est un mécanisme sous Linux permettant de dissocier un système de fichiers de son emplacement sur un ou plusieurs disques ou partitions. En fait, LVM introduit la notion de *volume physique (PV)* qui représente un disque ou une partition. La seconde notion importante est *le volume group (VG)* qui peut regrouper plusieurs volumes physiques. Grâce à ces VG, il est possible de créer un espace allant au-delà d’un seul volume physique.

Enfin, la dernière notion importante est *le volume logique (LV)* qui est l’espace destiné à un système de fichiers. Le LV est localisé sur un VG (qui lui peut recouvrir plusieurs disques ou partitions). On voit ainsi apparaître une grande flexibilité : il est possible d’agrandir un volume logique en lui dédiant un nouveau disque que l’on ajoute au VG configuré.

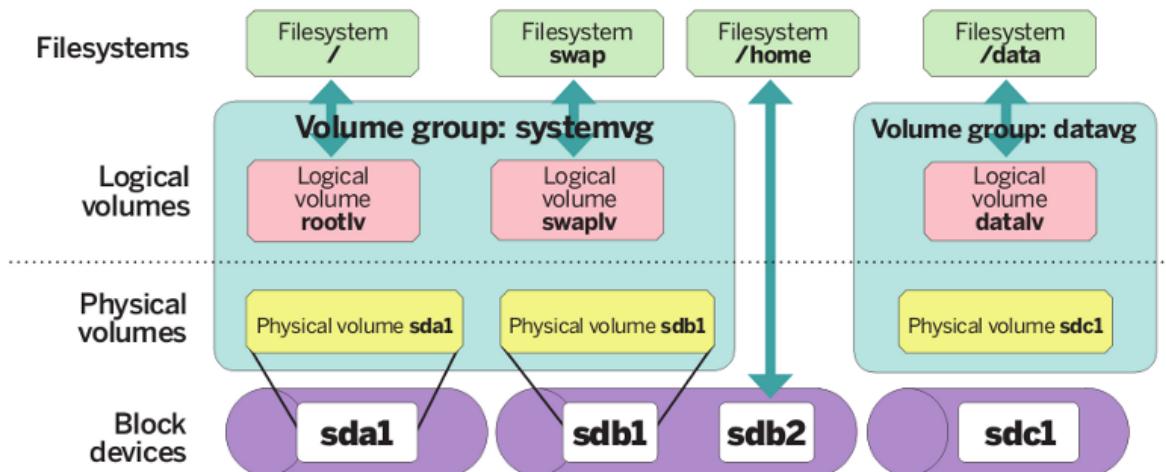


Figure 1.3 : Exemple d'utilisation de LVM<sup>4</sup>

Sur la figure 1.3, nous avons 3 disques durs SATA ou SCSI nommés `sda`, `sdb` et `sdc`. Sur le 1<sup>er</sup> et le dernier disque, il y a une seule partition (`sda1` et `sdc1`) tandis que le second disque contient deux partitions (`sdb1` et `sdb2`).

Dans cet exemple, 3 volumes physiques sont configurés pour utiliser LVM: `sda1` et `sdb1` d'une part et `sdc1` d'autre part. Enfin, 2 volumes groups sont ajoutés : le 1<sup>er</sup> `systemvg` utilise les volumes physiques `sda1` et `sdb1`, le second, `datavg` utilise le dernier volume physique `sdc1`.

Enfin, les volumes logiques sont créés : `rootlv` et `swaplv` sur le VG `systemvg`, `datalv` sur le VG `datavg`. Les systèmes de fichiers sont attachés au LV : la racine (le `/`) sur `rootlv`, la mémoire virtuelle sur `swaplv` et le dossier `/data` sur `datalv`. Il reste le dossier `/home` qui est directement contenu dans la partition `/dev/sdb2` (sans passer par LVM donc).

Sur notre machine virtuelle, MBR et LVM sont utilisés conjointement :

`$ fdisk /dev/sda`

Cette commande permet de gérer les partitions MBR sur le disque dur.

```

$ pvdisplay
$ vgdisplay
$ lvdisplay

```

Toutes ces commandes permettent de visualiser les informations concernant les PV, VG et LV configurés. Dans notre système, il y a 1 seul PV qui est `sda2`. Il y a un volume group reprenant le PV configuré (et se nommant `centos`).

Sur ce VG, il y a 3 volumes logiques (LV) nommés `root`, `swap` et `home`. Le premier reprend le système de fichiers racine (i.e. le `/`), le second est utilisé pour la mémoire virtuelle et le dernier reprend le stockage des dossiers personnels (i.e. `/home`). Ils sont accessibles au travers des noms de périphériques particuliers : `/dev/mapper/centos-root`, `/dev/mapper/centos-swap` et `/dev/mapper/centos-home`.

<sup>4</sup> La figure est extraite de : <http://www.tuxradar.com/content/lvm-made-easy>

## 1.2.2 Les répertoires

Les systèmes UNIX proposent une structure de répertoire assez commune. Ainsi, on trouve généralement les dossiers suivants :

Dossiers	Utilisation habituelle
/etc	Dossier contenant les fichiers de configuration du système (configuration réseau, système, mais aussi les différents services installés)
/dev	Dossier pointant vers les périphériques attachés au système. Il est possible ainsi de désigner un disque, une partition, un port d'E/S (USB par exemple) ou encore des périphériques particuliers (appelé pseudo-périphérique) comme /dev/zero (qui retourne uniquement des zeros) ou /dev/random (qui retourne des nombres aléatoires) ou /dev/null (qui absorbe tout et ne retourne rien).
/media	Dossier vide qui peut être utilisé par l'administrateur pour <i>monter</i> manuellement un périphérique (lecteur DVD, clé USB, dossier réseau distant, ...)
/home	Dossier standard pour contenir l'ensemble des répertoires personnels des utilisateurs (excepté pour l'utilisateur root). Il arrive souvent que son contenu soit stocké sur une autre partition ou sur un serveur distant.
/root	Répertoire personnel de l'administrateur (utilisateur root)
/var	Dossier contenant des informations <i>en cours de traitement</i> . Ainsi, on y trouve les files d'attente pour l'impression (/var/spool/lpd) ou les boîtes aux lettres des utilisateurs (/var/spool/mail), les pages web – si le serveur web est installé – (dans /var/www), les bases de données MySQL (/var/lib/mysql), ...
/bin & /sbin /usr/bin /usr/sbin	Dossiers contenant les programmes exécutables installés sur le système. Les dossiers bin contiennent les programmes disponibles pour tous les utilisateurs alors que les dossiers sbin contiennent plutôt les programmes qui peuvent être lancés par l'administrateur
/tmp	Dossier temporaire
/proc	Dossier particulier (il s'agit d'un système de fichier monté) contenant des informations sur les processus en cours d'exécution mais aussi sur le matériel et la configuration logicielle du système d'exploitation. Ce dossier est automatiquement généré par le système. Il ne faut donc rien y ajouter. Ex : /proc/cpuinfo identifie le processus du système

## 1.2.3 Administration du serveur par le Web

Un outil particulier est installé pour permettre l'administration du serveur par une interface web. Cet outil, nommé *Webmin*, permet d'ajouter, adapter, supprimer des éléments de configuration comme : ajouter des utilisateurs ou des groupes, administrer le disque dur, voir/planifier des tâches, configurer certains services, adapter les paramètres réseaux, ...

Cependant, dans notre étude du système Linux, **nous préférerons souvent la ligne de commande** qui a le grand avantage de permettre de scripter facilement toutes ces tâches.

Un second outil, *Usermin*, est également installé. Il permet la modification des paramètres personnels de chaque utilisateur. Voici comment accéder à ces outils :

- Webmin : <https://localhost:10000>
- Usermin : <https://localhost:20000>

### 1.2.4 Les utilisateurs globaux et locaux

Comme tous les systèmes d'exploitations actuels, Linux supporte à la fois des utilisateurs *locaux* (i.e. définit *localemement* sur la machine) et les utilisateurs *globaux* (i.e. définit dans un annuaire partagé entre plusieurs machines).

Les utilisateurs **globaux** sont utilisés lorsqu'il faut permettre à des utilisateurs de se connecter sur n'importe quelle machine du réseau. On définit alors ces utilisateurs dans un annuaire LDAP et on renseigne cet annuaire au niveau des différentes machines. Ainsi, l'authentification est *déléguée* à l'annuaire (et les informations concernant ces utilisateurs sont centralisées en 1 point) qui vérifie si le nom d'utilisateur et le mot de passe sont corrects.

A l'inverse, les utilisateurs **locaux** sont utilisés pour permettre l'accès au serveur considéré. Ainsi, il est possible de permettre la connexion sur ce seul serveur puisque l'utilisateur est seulement connu par celui-ci.

**Il faut bien remarquer que dans l'annuaire, il n'y a que les utilisateurs globaux alors que dans les fichiers de configuration de la machine, il n'y a que les utilisateurs locaux.**

#### Les utilisateurs et groupes locaux

Les utilisateurs **locaux** (uniquement) sont renseignés dans le fichier `/etc/passwd`. Chaque ligne de ce fichier texte décrit un utilisateur existant sur le système. Les informations suivantes sont renseignées pour chaque utilisateur, dans cet ordre (séparé par le symbole « : ») :

Information	Explication
<b>login</b>	Nom utilisé pour la connexion sur le serveur, c'est l'identifiant texte. Il doit être unique.
<b>Mot de passe</b>	Le mot de passe n'est pas précisé dans ce fichier mais dans le fichier <code>/etc/shadow</code> sous un format haché. C'est pourquoi celui-ci est remplacé par <code>x</code> dans ce fichier.
<b>UID</b>	Identifiant numérique unique associé à l'utilisateur. L'utilisateur <code>root</code> a toujours l'UID 0. Les utilisateurs locaux ont un UID $\geq 1000$ . Notons que les identifiants $< 500$ sont réservés pour <b>des comptes systèmes</b> .
<b>GID</b>	Identifiant numérique du groupe principal de l'utilisateur. Précisons que le groupe <code>root</code> a le GID 0. Le groupe <code>users</code> a le GID 100.
<b>Informations utilisateur</b>	Le champ suivant reprend les informations sur l'utilisateur : son nom, son prénom, son téléphone, son bureau, ...
<b>Dossier personnel</b>	L'avant dernier champ mentionne le dossier personnel de l'utilisateur. A l'exception des comptes systèmes et de l'utilisateur <code>root</code> , les dossiers des utilisateurs sont stockés dans <code>/home</code> . Ainsi, l'utilisateur <code>lsw</code> a son répertoire personnel dans <code>/home/lsw</code> . L'utilisateur <code>root</code> a son répertoire dans <code>/root</code> .
<b>Le shell</b>	Le dernier argument est le shell à lancer lors de la connexion de l'utilisateur. Nous utiliserons toujours <code>bash</code> comme shell en précisant : <code>/bin/bash</code> comme shell.

Les commandes pour gérer les utilisateurs sont :

**useradd**

Cette commande permet d'ajouter un utilisateur sur le système. Par défaut sur les systèmes RedHat/CentOS, cette commande crée un groupe du nom de l'utilisateur et intègre l'utilisateur dans ce groupe principal. Comme ce comportement n'est pas souhaité, il convient de préciser les arguments suivants :

Exemple :

```
$ useradd -g users superswila
```

Cette commande ajoute l'utilisateur (dont le login est `superswila`) dans la liste des utilisateurs locaux du système. Cet utilisateur est placé dans le groupe principal `users`. Par défaut, le système lui attribue le shell `bash` et détermine le chemin vers le dossier personnel dans `/home/superswila`. Consulter la page de manuel pour plus d'information<sup>5</sup>.

Le compte de l'utilisateur **est désactivé tant qu'aucun mot de passe n'est précisé**.

#### **usermod**

Cette commande permet de modifier les paramètres d'un utilisateur local déjà créé (par exemple, l'ajout dans un groupe secondaire, ...). Consulter la page de manuel pour plus d'information<sup>5</sup>.

#### **userdel**

Cette commande permet de supprimer un utilisateur local existant. Sans option particulière, le dossier personnel de l'utilisateur est conservé. Consulter la page de manuel pour plus d'information<sup>5</sup>.

#### **chfn**

Cette commande permet de changer le nom de l'utilisateur, préciser son bureau, et toutes les informations utilisateurs qui lui sont attachées. Consulter la page de manuel pour plus d'information<sup>5</sup>.

#### **passwd**

Sans paramètre, cette commande permet **de changer le mot de passe de l'utilisateur courant**. L'administrateur peut préciser le login d'un utilisateur pour changer ou fixer le mot de passe de celui-ci. Quand le mot de passe est mentionné pour la 1<sup>ère</sup> fois, le compte est automatiquement activé.

#### **id**

Sans paramètre, cette commande permet de connaître le nom d'utilisateur courant. Il précise également les groupes (principaux et secondaires) de cet utilisateur. Il est possible de préciser le login d'un utilisateur, la commande retourne alors les informations de cet utilisateur.

#### **chsh**

Cette commande permet de changer le shell d'un utilisateur.

Les groupes **locaux** sont mentionnés dans le fichier `/etc/group`. Le format est analogue à celui des utilisateurs et il indique :

Information	Explication
-------------	-------------

<sup>5</sup> Pour rappel : `man useradd (<espace> page suivante, b page précédente, q quitter)`.

<b>groupname</b>	Nom textuel utilisé pour identifier ou afficher le groupe
<b>x</b>	Ce champ ne sert pas ici
<b>GID</b>	L'identifiant numérique associé au groupe. Le groupe <code>root</code> utilise l'identifiant 0, le groupe <code>users</code> utilise l'identifiant 100.
<b>liste des utilisateurs séparés par « , »</b>	Precise les utilisateurs ayant <b>comme groupe secondaire le groupe en question</b> . Ainsi, si l'utilisateur lsw a comme groupe principal le groupe <code>users</code> et comme groupe secondaire <code>audio</code> , nous aurons : l'identifiant 100 précisé comme GID dans le fichier <code>passwd</code> pour cet utilisateur et, dans les membres du groupe <code>audio</code> , le login <code>lsw</code> apparaîtra. Par contre, il n'apparaîtra pas pour le groupe <code>users</code> (ce n'est pas un groupe secondaire).

Les commandes pour gérer les groupes sont :

#### groupadd

Cette commande permet d'ajouter un groupe sur le système. Consulter la page de manuel pour plus d'information.

#### groupmod

Cette commande permet de modifier un groupe déjà existant. Consulter la page de manuel pour plus d'information.

#### groupdel

Cette commande permet de supprimer un groupe existant. Consulter la page de manuel pour plus d'information.

#### groups

Cette commande permet de connaître les groupes auxquels l'utilisateur appartient. Consulter la page de manuel pour plus d'information.

### Les utilisateurs et groupes globaux

Les utilisateurs et groupes **globaux** sont précisés dans l'annuaire LDAP. L'annuaire est déjà installé sur le serveur CentOS virtuel. Il est possible d'accéder, en mode graphique, à ce dernier comme suit :

\$ 389-console

Cette commande permet de démarrer l'interface d'administration graphique en Java. Il faut préciser les informations de connexion suivantes :

- User ID : `cn=Directory Manager`
- Password : `rootroot`
- Administration URL : `http://localhost:9830`

L'interface est alors démarrée. Si l'on déploie l'arbre, on voit apparaître deux éléments importants :

- *Administration Server* : cette option reprend tous les paramètres de configuration **du serveur d'administration**. Il n'est pas nécessaire d'y accéder.
- *Directory Server (localhost)* : cette option reprend la configuration et le contenu **de l'annuaire LDAP**. C'est cette partie que nous allons explorer.

Si l'on l'explore l'élément *Directory Server*, on voit apparaître la fenêtre de gestion de l'annuaire comportant plusieurs onglets. L'onglet *Task* permet de relancer le service, gérer les certificats, etc.

L'onglet *Configuration* permet de déterminer comment se connecter à l'annuaire (activation de SSL/TLS), configurer la réPLICATION, etc. Ensuite, l'onglet *Directory* permet de parcourir le contenu de l'annuaire. Enfin, l'onglet *Status* regroupe toutes les informations concernant l'état du système, les fichiers journaux, etc.

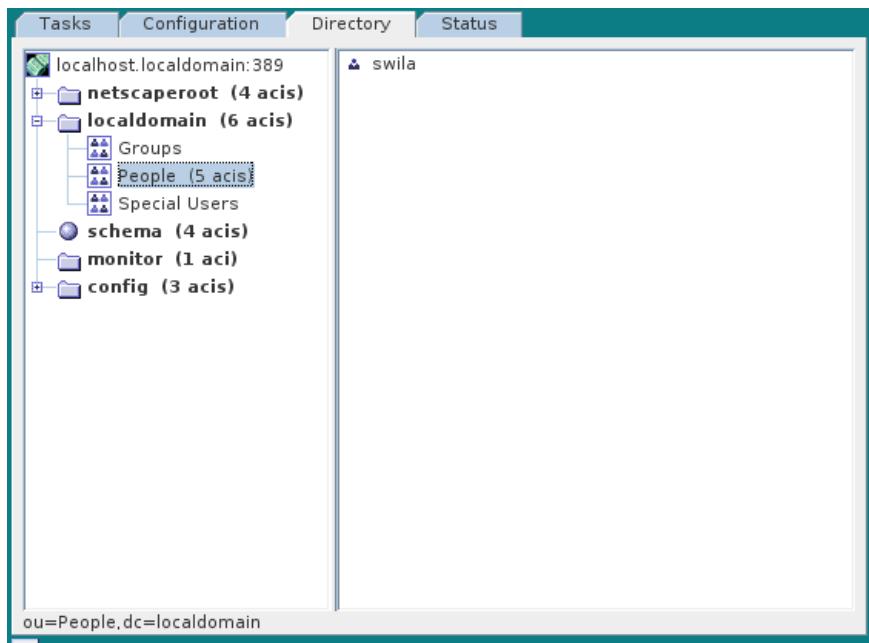


Figure 1.4 : Le contenu de l'annuaire LDAP

Comme on peut le voir sur la figure 1.4, nous pouvons consulter le contenu de l'annuaire. Il y a déjà plusieurs *unités d'organisation* qui existent : Groups, People et Special Users. Ces unités permettent de structurer l'annuaire de sorte à ce qu'il reproduise l'organisation de l'entreprise. Ainsi, il n'est pas rare de créer des unités organisationnelles (abrégées en OU) pour chaque service de l'entreprise. Une OU est un conteneur pouvant accueillir des utilisateurs ou des groupes globaux. Ainsi, dans la OU People, nous avons déjà l'utilisateur swila qui est présent. Il s'agit donc d'un utilisateur global.

Pour créer une OU, il faut simplement faire un **clic-droit** dans l'arbre à gauche, dans le conteneur parent et choisir **New > Organizational Unit**. Il faut préciser le nom (qui est obligatoire) et éventuellement une description.

Il est également possible de créer cette OU par la ligne de commande. Il faut commencer par créer un fichier texte reprenant l'unité organisationnelle à créer (par exemple : /tmp/ou.ldif):

```
dn: ou=Student, ou=People, dc=localdomain
ou: Student
description: Tous les etudiants
objectClass: top
objectClass: organizationalUnit
```

Nous pouvons ensuite importer les informations dans l'annuaire avec la commande :

```
$ ldapadd -D 'cn=Directory Manager' -f /tmp/ou.ldif -x -W
```

Le mot de passe est demandé. L'unité Student est ensuite créée dans l'annuaire dans l'unité People.

Pour créer un utilisateur dans une OU, il faut simplement faire un clic-droit dans le conteneur souhaité et choisir l'option **New > User**. Dans les options à préciser, il faut remplir *le first name*, *le last name*, *le common name*, *le user ID* (= login) et *le password*. Il faut également remplir les informations **Posix User** pour permettre la connexion de cet utilisateur. Ainsi, il faut spécifier son *UID Number* (= son identifiant numérique, idéalement  $\geq 5000$  pour éviter tout conflit avec les utilisateurs locaux), *GID Number* (= identifiant numérique du groupe principal – peut être 100 si le groupe principal est `users`), *le Home directory* (= le chemin vers son dossier personnel – idéalement `/home/<login>` qu'il faut créer à la main), et le *Login Shell* (= `/bin/bash`).

Pour créer un utilisateur global par ligne de commande, il faut créer un fichier texte (par exemple :

```
/tmp/user.ldif):
dn: uid=ldapswila,ou=Student,ou=People,dc=localdomain
objectClass: top
objectClass: inetorgperson
objectClass: posixAccount
cn: Louis SWINNEN
sn: SWINNEN
givenname: Louis
userPassword: rootroot
gidNumber: 100
uidNumber: 5001
homeDirectory: /home/ldapswila
loginShell: /bin/bash
```

L'ajout dans l'annuaire se fait par la commande :

```
$ ldapadd -D 'cn=Directory Manager' -f /tmp/user.ldif -x -W
```

Cette commande provoque l'ajout de l'utilisateur `ldapswila` dans l'OU `Student` de l'annuaire LDAP.

Pour créer un groupe, il faut faire un clic-droit sur le conteneur et choisir **New > Group**. Il faut mentionner le *Group Name* et, éventuellement une description. Il faut également remplir les informations **Posix Group** pour indiquer le *GID Number* (= l'identifiant numérique du groupe), et inclure les utilisateurs ayant ce groupe comme groupe secondaire.

Pour créer un groupe global par ligne de commande, il faut créer un fichier texte (par exemple :

```
/tmp/group.ldif):
dn: cn=bac1,ou=Groups,dc=localdomain
description: bac1
objectClass: top
objectClass: groupOfUniqueNames
objectClass: posixGroup
gidNumber: 5000
```

L'ajout dans l'annuaire se fait par la commande :

```
$ ldapadd -D 'cn=Directory Manager' -f /tmp/group.ldif -x -W
```

Cette commande provoque l'ajout du groupe `bac1` dans l'OU `Groups`.

Pour ajouter un utilisateur dans un groupe existant (`/tmp/addmember.ldif`) :

```
dn: cn=bac1,ou=Groups,dc=localdomain
changetype: modify
```

```
add: memberuid  
memberuid: ldapswila
```

L'ajout dans l'annuaire se fait par la commande :

```
$ ldapadd -D 'cn=Directory Manager' -f /tmp/addmember.ldif -x -W
```

Cette commande provoque l'ajout de l'utilisateur `ldapswila` dans le groupe `bac1`.

## 1.3 Les commandes de base<sup>6</sup>

### 1.3.1 Les commandes sur les dossiers

#### ls

Cette commande permet de lister les répertoires et les fichiers.

```
ls [-l|-a|-h] [fichier...]
```

Exemples:

```
$ ls  
$ ls -l  
$ ls -a -l est équivalent à ls -al  
$ ls -la ~
```

#### cd

Cette commande permet de changer le répertoire courant

```
cd [répertoire]
```

Exemples:

```
$ cd /tmp  
$ pwd  
$ cd ..  
$ cd -  
$ cd
```

#### mkdir

Permet de créer un répertoire

```
mkdir [-m mode|-p] répertoire...
```

Exemples:

```
$ mkdir rep1 rep2  
$ mkdir -p rep3/test/exemple doc/louis/linux
```

#### rmdir

Permet de supprimer un répertoire vide

```
rmdir [-p] répertoire...
```

Exemples:

```
$ rmdir rep2  
$ rmdir -p rep3/test/exemple
```

<sup>6</sup> Cette partie est extraite du cours *d'introduction à Linux* du NamurLUG (<http://www.namurlug.org>).

### 1.3.2 Les commandes sur les fichiers

#### **mv**

Cette commande permet de changer le nom d'un répertoire ou d'un fichier. Cette commande permet également de déplacer un fichier ou un répertoire.

```
mv [-i] source dest  
mv [-i] source... répertoire
```

Exemples:

```
$ mv repl jef  
$ mv jef doc/
```

#### **cp**

Permet de copier un fichier ou un répertoire.

```
cp [-i | -r] source dest  
cp [-i | -r] source... répertoire
```

Exemples:

```
$ cp -r /etc doc/jef  
$ cp /etc/nsswitch.conf doc/louis
```

#### **rm**

Permet de supprimer un fichier ou un répertoire

```
rm [-i | -f | -r] fichier...
```

Exemples:

```
$ rm -rf doc/jef  
$ rm -i doc/louis/nsswitch.conf
```

#### **file**

Cette commande permet de connaître le type d'un fichier

```
file fichier...
```

Exemples:

```
$ file /usr/bin/man  
$ file /etc/nsswitch.conf  
$ file doc
```

#### **more et less**

Ces commandes permettent de visualiser le contenu d'un fichier texte, page par page.

```
more fichier  
less fichier
```

Exemples:

```
$ more /etc/nsswitch.conf  
$ less /etc/nsswitch.conf
```

## cat

Cette commande permet de fusionner plusieurs fichiers et afficher le résultat sur la sortie standard (écran, par défaut). Cette commande est également utilisée pour afficher le contenu d'un fichier à l'écran.

```
cat fichier...
```

Exemples:

```
$ cat /etc/nsswitch.conf /etc/fstab
```

## find

Cette commande permet d'effectuer une recherche d'un fichier.

```
find répertoire... expression
```

Exemples:

```
$ find / -name man  
$ find / -name emacs -exec file {} \;
```

## grep

Cette commande permet de faire une recherche à l'intérieur d'un fichier texte. Elle affiche les lignes contenant au moins une occurrence de l'expression recherchée.

```
grep [-r] [-i] "chaîne" fichiers...
```

Exemples:

```
$ grep "files" /etc/nsswitch.conf
```

### 1.3.3 Les redirections

Le système Linux définit 3 flux standards :

- *stdin* : l'entrée standard, par défaut attachée au clavier
- *stdout* : la sortie standard, par défaut attachée à l'écran
- *stderr* : la sortie pour les erreurs, par défaut attachée à l'écran

Chaque commande *peut* utiliser ces 3 flux standards. Par exemple, lors d'un *printf* en C, le résultat est par défaut envoyé vers la sortie standard. Il est possible de rediriger ces flux vers des fichiers.

#### Redirection de la sortie standard

Le résultat est alors envoyé dans un fichier.

commande > fichier      *le fichier est créé ou écrasé*  
commande >> fichier      *le résultat est ajouté au fichier*

Exemples:

```
$ ls > out  
$ ls -l >> out
```

### **Redirection du standard d'erreur**

Les messages d'erreurs sont alors envoyés dans un fichier.

commande 2> fichier	<i>Redirige le standard d'erreur</i>
commande &> fichier	<i>Redirige la sortie standard + erreur</i>

Exemples:

```
$ find / -name emacs 2> find-error
```

### **Redirection de l'entrée standard**

Les données ne sont plus demandées au clavier et sont prises à partir d'un fichier.

commande < fichier

Exemples:

```
$ mail root < /etc/inittab
```

### **Liens entre plusieurs commandes**

L'intérêt majeur des redirections est de pouvoir lier la sortie d'une commande à l'entrée d'une autre. Les systèmes UNIX permettent de réaliser cela en une seule opération. Il est alors possible de chaîner plusieurs commandes réalisant des opérations ciblées.

commande1 | commande2

Exemples:

```
$ cat /etc/fstab | less
$ cat /etc/nsswitch.conf | grep "files"
$ cat /etc/aliases | grep "news"
```

### **1.3.4 Commandes sur la gestion des partitions**

La description des différentes partitions du système est concentrée dans le fichier `/etc/fstab`. On trouve, pour chaque partition le point de montage associée, le système de fichier utilisé et les options utilisées lors du montage (nous verrons les différentes options lors d'une prochaine séance).

#### **fdisk**

Cette commande permet de modifier les partitions du système (à utiliser avec précaution !). Elle permet également de connaître l'état des partitions. Il suffit d'utiliser l'option p du menu pour connaître les différentes partitions du système.

`fdisk periphérique`

Exemples:

```
$ fdisk /dev/sda
```

#### **mount**

Cette commande permet de monter une partition dans le système de fichier. Par cette commande, on rend accessible un système de fichier. Il peut s'agir d'une clé ou d'un disque USB, d'un DVD (`/dev/sr0`), d'une partition sur un disque.

```
mount [-t système_fichier] périphérique point_de_montage/
```

Exemples:

```
$ mount /home  
$ mount /dev/sr0 /media  
$ mount -t vfat /dev/sdb1 /media
```

### umount

Cette commande permet de supprimer l'association entre le système de fichier et le périphérique. Il est indispensable de démonter un périphérique avant de l'emporter. Cela assure que toutes les opérations ont bien été effectuées sur le disque.

```
umount périphérique  
umount point_de_montage/
```

Exemples:

```
$ umount /home  
$ umount /media  
$ umount /dev/sdb1
```

### e2fsck

Cette commande permet de vérifier la cohérence et l'intégrité du système *ext2/ext3/ext4*. Elle est lancée par le système lors d'un arrêt brutal (coupure de courant, problème matériel, ...). Dans certains cas, lorsque les erreurs sont graves, le système demande d'exécuter cette commande manuellement.

**Cette commande ne peut être exécutée que sur des systèmes non montés !**

Exemples:

```
$ umount /dev/mapper/centos-home  
$ e2fsck -c /dev/mapper/centos-home  
$ e2fsck -f /dev/mapper/centos-home  
$ e2fsck -y /dev/mapper/centos-home
```

Dans ces exemples, nous réparons le système de fichier attaché au volume *centos-home*. Il est, en effet, formaté en *ext4*. Sans option particulière, le programme vérifie le système de manière interactive (si des problèmes sont rencontrés, l'utilisateur doit prendre une décision). L'option *-c* permet de lancer une vérification des secteurs défectueux, l'option *-f* force la vérification du système même si l'état est cohérent, l'option *-y* suppose la réponse *yes* à toutes les questions (correction des erreurs).

### xfs\_repair

Cette commande permet de vérifier la cohérence et l'intégrité des systèmes *xfs*. Comme la commande précédente, elle est lancée en cas d'arrêt brutal du système. Il est parfois nécessaire de lancer cette commande manuellement.

**Cette commande ne peut être exécutée que sur des systèmes non montés !**

Exemples :

```
$ umount /dev/sda1  
$ xfs_repair -n /dev/sda1  
$ xfs_repair /dev/sda1
```

Dans ces exemples, nous réparons le système de fichier présent sur la 1<sup>ère</sup> partition du disque. Il est, en effet, formaté en *xfs*. L'option *-n (no modify)* n'effectue aucune correction sur le système de fichier, il permet de savoir si celui-ci est corrompu et s'il doit être réparé. Sans option, le système tente de réparer les erreurs rencontrées.

## 1.4 Exercices

1. Au moyen de la ligne de commande :
  - a. Créer un utilisateur local correspondant à votre login HELMo et placer le dans le groupe principal *users*. Précisez également vos informations (nom, prenom, ...) et fixez le mot de passe.
  - b. Créer un groupe local *friends*
  - c. Créer un compte pour votre voisin dans le groupe principal *users* et dans le groupe secondaire *friends*.
  - d. Modifier (*usermod*) votre compte pour qu'il appartienne également au groupe secondaire *friends*
  - e. Vérifier avec les commandes et en consultant les fichiers adéquats que tout est correct. Tentez de vous connecter !
2. Au moyen de la console d'administration LDAP :
  - a. Créez une OU *student* avec, à l'intérieur, une autre unité *BM*
  - b. Dans l'OU *Groups*, créer 4 groupes globaux : *biomed1*, *biomed2*, *biomed3* et *étudiants*
  - c. Dans l'OU *BM*, créer 3 utilisateurs globaux : *bm1*, *bm2*, et *bm3* appartenant au groupe principal *étudiants* et au groupe *biomedX* correspondant.
3. Au moyen de la ligne de commande, insérer dans LDAP :
  - a. Une OU *Info* à l'intérieur de la OU *student*
  - b. Dans l'OU *Groups*, les 3 groupes globaux : *info1*, *info2* et *info3*
  - c. Dans l'OU *Info*, créer 3 utilisateurs globaux *inf1*, *inf2* et *inf3* appartenant au groupe principal *étudiants* et au groupe *infoX* correspondant.

Pour ce faire, créer les fichiers LDIF et importez-les.

# Leçon 2 : Scripting sous Linux

Dans la version CentOS 7, la version de *Python* installée est la version 2.x. Il faut, dès lors, commencer par installer Python 3. Pour ce faire, il faut entrer la commande suivante (en *root*) :

```
# yum -y install python3
```

Une fois ce package installé, il est possible de créer un script *Python* simplement :

```
#  
1  #! /usr/bin/python3  
2  print("Hello World !")
```

Script 2.1 : le programme helloworld.py

La 1<sup>ère</sup> ligne indique que le script doit être exécuté avec l'interpréteur *python3*. A partir de la 2<sup>ème</sup> ligne, on trouve le script en lui-même.

Pour **exécuter** le script sous Linux, il faut commencer par lui donner la permission en exécution (à faire une seule fois) :

```
$ chmod +x helloworld.py  
$ ./helloworld.py
```

## Exercices

1. Créer un script *Python*, nommé `mklist.py`, qui va **produire à l'écran** le fichier CSV auquel on aura ajouté 2 colonnes supplémentaires : les logins et mots de passe.
  - a. **Les logins** pour les utilisateurs suivront la règle : en fonction de la section de l'étudiant, la 1<sup>ère</sup> lettre sera a, b, i ou t. Nous aurons ensuite 4 chiffres représentant une séquence. Ainsi, a0001 représente le 1<sup>ier</sup> étudiant de la section automatique, i0054, le 54<sup>ème</sup> étudiant de la section informatique...
  - b. Pour générer le mot de passe de chaque utilisateur, utilisez la commande `mkpasswd`<sup>1</sup>. Le mot de passe doit faire 12 caractères et comporter des lettres (majuscules et minuscules) et chiffres (reportez-vous à la documentation pour les paramètres à passer à `mkpasswd`).

Testez votre script et vérifiez si l'information à l'écran est correcte. **Ensuite seulement**, lancez votre script comme suit :

```
$ mklist.py > liste-login-pass.csv
```

Vous obtiendrez ainsi **un nouveau fichier CSV** contenant en plus les logins et mots de passe.

2. Créer un script *Python* nommé `mkuser.py` qui va, en se basant sur le fichier CSV obtenu à l'étape précédente, créer les comptes locaux des différents étudiants. On vous demande également de fixer, par ce script, les mots de passe et les données (noms et prénoms) de ces utilisateurs.

On vous demande de placer les étudiants dans le groupe principal (et local) *users* et dans un groupe secondaire correspondant à leur classe (en minuscule).

<sup>1</sup> `mkpasswd` est une commande système existante. Vous pouvez d'ailleurs voir son résultat :  
\$ `mkpasswd`

- Créer un script *Python* permettant d'ajouter facilement un utilisateur dans le LDAP. Pour ce faire l'utilisateur sera ajouté dans le `OU=People`, il sera configuré comme utilisateur POSIX, aura comme groupe principal, le groupe `users` (`gid=100`) et comme shell `bash` (`/bin/bash`). Le nom, le prénom, l'UID, le login et le mot de passe seront donnés en argument de votre script.

## Ressources

- Lire un fichier ligne par ligne en *python* : <https://www.geeksforgeeks.org/read-a-file-line-by-line-in-python/>
- Utiliser des expressions régulières en *Python* : <https://docs.python.org/fr/3/library/re.html>
- Exécuter un programme et récupérer le résultat en *Python* :  
<https://stackoverflow.com/questions/4760215/running-shell-command-and-capturing-the-output> ou encore <https://stackoverflow.com/questions/1410976/equivalent-of-bash-backticks-in-python>
- Formater des nombres (ajouter des *0* en tête) :  
<https://stackoverflow.com/questions/134934/display-number-with-leading-zeros>
- Récupérer les arguments de la ligne de commande en *Python* avec `sys.argv` :  
<https://www.geeksforgeeks.org/how-to-use-sys-argv-in-python/>

## Leçon 3 : Administration du disque dur

### 3.1 Les entrées dans le système de fichiers

Sous Unix, il existe plusieurs types d'entrée différente. La commande ls permet de montrer les détails concernant ces « fichiers » particuliers. Le 1<sup>er</sup> caractères désigne le type d'entrée. Ainsi, on trouve :

- Des *fichiers* (-) ou des *dossiers* (d) : ls -l /etc, montre notamment :
 

```
-rw-r--r--. 1 root root      970 Mar  9 2015 yum.conf
drwxr-xr-x. 2 root root     4096 Jul 28 11:58 yum.repos.d
```
- Des *liens symboliques* (l) : ls -l /bin
 

```
lrwxrwxrwx. 1 root root 7 Jul 27 00:55 /bin -> usr/bin
```
- Des *périphériques blocs* (b) ou *caractères* (c) : ls -l /dev/sda /dev/tty
 

```
brw-rw---- 1 root disk 8, 0 Sep 15 22:16 /dev/sda
crw-rw-rw- 1 root tty  5, 0 Sep 15 22:05 /dev/tty
```
- Des *sockets locaux* (s) : ls -l /dev/log
 

```
srw-rw-rw- 1 root root 0 Sep 15 22:05 /dev/log
```

### 3.2 Les liens

Les liens sont des entrées particulières qui permettent de *pointer* vers un autre endroit du disque dur. Ainsi, un *lien symbolique* permet de créer une entrée pointant vers un dossier ou un fichier ailleurs. L'intérêt principal étant de faire apparaître, dans des dossiers différents, les mêmes entrées.

Attention ! Il ne s'agit pas d'une copie mais bien d'un *pointeur*. Quand on ouvre un lien, c'est l'élément pointé qui est ouvert.

### 3.3 Les droits

Les droits dans les systèmes UNIX sont de différents types : il y a les *droits classiques* et les *ACLs*.

#### 3.3.1 Les droits classiques

Les droits classiques sont ceux qui sont affichés lorsqu'on exécute la commande ls -l. Nous pourrions obtenir le résultat suivant sur le dossier /home :

```
$ ls -l /home
total 28
drwx----- 2 root  root  16384 Jul 27 00:54 lost+found
drwx----- 15 lsw   users  4096 Aug 13 22:12 lsw
drwxrwxr-x+ 2 root  root  4096 Aug 11 23:27 public
drwx----- 18 swila users  4096 Aug 14 00:01 swila
```

Nous y voyons des combinaisons étranges des lettres r, w, x et du symbole -. Les permissions représentées sont : *l'autorisation en lecture (r)*, *l'autorisation en écriture (w)* et *l'autorisation en exécution / accès (x)*.

Ces permissions sont présentes à la fois pour les dossiers ou les fichiers. Elles sont présentes sous la forme d'un trio, répété trois fois (ex : rwxrwxr-x).

Chaque trio doit être interprété comme suit :

Permissions	Explication
rwx	Autorisation en lecture, en écriture, en accès / exécution

<b>r-x</b>	Autorisation en lecture, en exécution / accès
<b>r--</b>	Autorisation en lecture
<b>--x</b>	Autorisation en exécution / accès
<b>rwx</b>	Autorisation en lecture, en écriture

Ainsi, la présence de la lettre dans le trio indique que l'autorisation correspondante est présente alors qu'un tiret (-) indique que l'autorisation correspondante est absente.

Comme le trio est répété trois fois, cela indique que les permissions s'adressent à des groupes d'utilisateurs différents. Ainsi dans l'exemple du dossier `/home` présenté ci-avant, nous avons :

```
drwx----- 15 lsw    users  4096 Aug 13 22:12 lsw
```

Le premier trio (`rwx`) indique que les autorisations mentionnées sont accordées **à l'utilisateur propriétaire** qui dans notre cas, est `lsw`. Le second trio (`---`) indique que les autorisations mentionnées (aucun droit ici) sont accordées **au groupe propriétaire** qui dans cet exemple est `users`. Enfin, le dernier trio (`---`) indique que les autorisations mentionnées (aucun droit ici) sont accordées **aux autres utilisateurs** (i.e. qui ne sont ni l'utilisateur propriétaire, ni membres du groupe propriétaire).

```
drwxrwxr-x+ 2 root   root   4096 Aug 11 23:27 public
```

Le premier trio (`rwx`) indique que les autorisations mentionnées sont accordées **à l'utilisateur propriétaire** qui dans notre cas, est `root`. Le second trio (`rwx`) indique que les autorisations mentionnées sont accordées **au groupe propriétaire** qui dans cet exemple est `root`. Enfin, le dernier trio (`r-x`) indique que les autorisations mentionnées sont accordées **aux autres utilisateurs** (i.e. qui ne sont ni l'utilisateur propriétaire, ni membres du groupe propriétaire).

Le symbole +, présent à la fin des permissions, informe que des ACLs sont définies également (voir plus loin).

### Les permissions sur les fichiers

L'autorisation en lecture (`r`) sur un fichier indique *qu'il est permis de lire le contenu du fichier*. L'autorisation en écriture (`w`) sur un fichier indique *qu'il est permis de modifier le contenu du fichier*. L'autorisation en exécution (`x`) sur un fichier indique *que ce fichier peut être exécuté*.

### Les permissions sur les dossiers

L'autorisation en lecture (`r`) sur un dossier indique qu'il est permis de lire le contenu du répertoire (les fichiers qui s'y trouvent). L'autorisation en écriture (`w`) sur un dossier indique qu'il est permis de modifier le contenu du répertoire (ajouter / supprimer des fichiers ou dossiers à l'intérieur). Enfin, l'autorisation en exécution (`x`) sur un dossier indique qu'il est permis de traverser le répertoire.

Un exemple particulier :

```
drwx--x--x. 15 lsw    users  4096 Aug 13 22:12 lsw
```

Dans cet exemple, le dossier `lsw` peut être lu, modifié et traversé par l'utilisateur `lsw`. Par contre, les membres du groupe `users` et les autres peuvent seulement traverser (i.e. entrer dans) le répertoire sans pouvoir lire son contenu.

Cette particularité est parfois intéressante pour *donner une autorisation plus grande sur un dossier à l'intérieur*.

### Modification des droits

La modification des *droits classiques* est assez simple. Cette modification peut prendre à la fois la forme d'un changement de permission ou d'un changement de propriétaire (utilisateur ou groupe propriétaire). Les commandes suivantes sont utilisées :

#### chmod

La commande `chmod` permet de modifier les permissions sur un dossier ou un fichier. Pour exprimer la permission souhaitée, il est possible d'utiliser une forme symbolique ou numérique.

La méthode numérique consiste à interpréter le trio `rwx` sous une forme binaire. Ainsi le droit `r-x` se traduit, en binaire par `101`, c'est-à-dire `5`. Ecrire `755` représente la permission `rwxr-xr-x`.

La méthode textuelle consiste à décrire les droits souhaités en utilisant les raccourcis suivants : `u` pour l'utilisateur propriétaire, `g` pour le groupe propriétaire et `o` pour les autres. Ainsi, écrire `u=rwx, g=rx, o---` représente la permission `rwxr-x---`.

```
$ chmod 755 mondossier  
$ chmod u=rwx,go=rx mondossier
```

#### chown

La commande `chown` permet de changer l'utilisateur et le groupe propriétaire d'un dossier ou fichier.

Par exemple :

```
$ chown lsw:users mondossier
```

#### chgrp

La commande `chgrp` permet de modifier le groupe propriétaire d'un dossier ou d'un fichier.

## 3.4 Les ACLs

La limitation des droits classiques a conduit à l'introduction des *ACLs*. Une ACL est un droit spécifique, mentionnant des permissions pour un utilisateur ou un groupe donné. L'ACL est attachée à un dossier ou un fichier. Dans certains cas, l'ACL est transmise aux dossiers ou fichiers contenus.

Ainsi, grâce à une ACL il est possible de préciser que l'utilisateur `lsw` peut lire ou modifier un fichier donné.

Bien sûr, il est possible d'attacher *plusieurs ACLs* à un fichier ou dossier pour permettre ainsi à l'utilisateur `lsw` mais aussi à l'utilisateur `swila` d'accéder au dossier, par exemple. Les ACLs doivent être combinées avec les droits classiques.

Certains systèmes de fichier nécessite d'activer le support des ACLs. Par défaut sur les distributions CentOS 7, les ACLs sont actives sur les systèmes de fichier `ext4` et `xfs`. Dans notre cas, cela signifie que les ACLs peuvent être utilisées n'importe où.

### 3.4.1 Les ACLs classiques et les ACL par défaut

Il y a 2 types d'ACLs : les ACLs classiques et les ACLs par défaut. Les ACLs classiques peuvent être définies sur un fichier ou un dossier et elles portent uniquement sur celui-ci. Ainsi, **elles ne sont pas transmises ou héritées** si un nouveau fichier est ajouté dans le dossier.

Les ACLs par défaut sont par contre celles **qui seront héritées** lorsqu'un fichier ou dossier sera créé. Grâce à ces ACLs par défaut, il est possible de transmettre des droits déterminés.

Ainsi, si mon dossier `testACL` dispose des ACLs suivantes :

```
user:lsw:rwx
user:swila:rwx
default:user:lsw:rwx
```

Les utilisateurs `lsw` et `swila` disposent personnellement des permissions `rwx` sur le dossier `testACL` (ils peuvent donc ajouter / supprimer / traverser le dossier). Par contre, si un dossier y est ajouté, seule la permission par défaut (`user:lsw:rwx`) sera héritée par ce nouveau dossier.

### 3.4.2 Manipuler les ACLs

Pour manipuler les ACLs, il y a 2 commandes principales : `getfacl` et `setfacl`. La commande `getfacl` liste les droits sur un dossier ou fichier en y incluant les ACLs qui seraient présentes :

```
$ getfacl testACL
# file: testACL/
# owner: root
# group: root
user::rwx
user:lsw:rwx
user:swila:rwx
group::r-x
mask::rwx
other::r-x
default:user::rwx
default:user:lsw:rwx
default:group::r-x
default:mask::rwx
default:other::r-x
```

Nous pouvons voir dans cet exemple plusieurs choses importantes. Tout d'abord, un rappel de l'utilisateur (`owner`) et du groupe (`group`) propriétaire. Ensuite, nous avons la liste des ACLs définie qu'il faut comprendre comme suit :

ACL	Explication
<code>user::rwx</code>	L'utilisateur propriétaire (ici <code>root</code> ) dispose des autorisations <code>rwx</code> sur le dossier <code>testACL</code>
<code>user:lsw:rwx</code> <code>user:swila:rwx</code>	L'utilisateur renseigné ( <code>lsw</code> et <code>swila</code> ) dispose des autorisations <code>rwx</code> sur le dossier <code>testACL</code>
<code>group::r-x</code>	Le groupe propriétaire (ici <code>root</code> ) dispose des autorisations <code>r-x</code> sur le dossier <code>testACL</code>
<code>other::r-x</code>	Les autres utilisateurs disposent des autorisations <code>r-x</code> sur le dossier <code>testACL</code>
<code>default:user::rwx</code> <code>default:group::r-x</code> <code>default:other::r-x</code>	Les ACLs <code>default</code> définissent les permissions qui seront héritées. De manière assez logique, les autorisations pour l'utilisateur propriétaire, le groupe propriétaire et les autres (i.e. les droits classiques donc) sont mentionnés
<code>default:user:lsw:rwx</code>	Cette ACL précise que si un dossier est créé dans le dossier <code>testACL</code> ,

	<p>les ACLs suivantes sont automatiquement associées :</p> <pre>user:lsw:rwx default:user:lsw:rwx</pre> <p>Nous remarquons, par contre, que les ACLs concernant l'utilisateur swila ne font l'objet d'aucun héritage (car elles ne sont pas mentionnées aussi en mode <code>default</code>).</p>
--	--

La commande `setfacl` permet, quant à elle, de fixer les ACLs sur un fichier ou un dossier. Ainsi, à titre d'exemple, si nous souhaitons fixer les ACLs pour le dossier `testACL`, voici la commande :

```
$ setfacl -m u:lsw:rwx -m u:swila:rwx -m d:u:lsw:rwx testACL
```

Comme nous pouvons le voir, l'option `-m` permet de modifier ou d'ajouter des ACLs. Cette option peut être répétée autant de fois que souhaité. Ainsi, écrire `u:lsw:rwx` précise que l'utilisateur (`u`) dont le login est `lsw` doit se voir attribuer les autorisations `rwx`. Il en va de même pour l'utilisateur `swila`. Pour la dernière, `d:u:lsw:rwx` précise une permission par défaut qui sera héritée automatiquement par tous les objets enfants.

### 3.5 Le fichier `fstab`

Le système présente, dans le dossier `/etc`, un fichier texte nommé `fstab`. Ce fichier décrit toutes les partitions que le système connaît et détermine les options utilisées lors du *montage*<sup>7</sup> de celle-ci.

Ce fichier texte est important et **toute modification doit être apportée soigneusement**. En effet, une erreur dans le fichier peut conduire le système à ne plus démarrer correctement.

```
# /etc/fstab
# Created by anaconda on Mon Jul 27 00:54:49 2015
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
/dev/mapper/centos-root / ext4 defaults 1 1
UUID=53796dc0-62e5-4048-8a2b-0dedc1425f42 /boot xfs defaults 0 0
/dev/mapper/centos-home /home ext4 defaults 1 2
/dev/mapper/centos-swap swap defaults 0 0
```

Comme nous pouvons le voir ci-dessus, le contenu du fichier `/etc/fstab` est divisé en plusieurs colonnes. Ainsi la 1<sup>ère</sup> colonne décrit la **partition concernée** : celle-ci peut être désignée par son entrée dans le répertoire `/dev` ou par un identifiant comme le UUID (identifiant unique). La 2<sup>ème</sup> colonne mentionne le **point de montage** : nous avons dans l'extrait ci-dessus `/`, `/boot`, `/home` et `swap` (qui représente la mémoire virtuelle). Cela signifie donc que lorsqu'on écrit un dossier ou un fichier dans le dossier `/home`, par exemple, on travaille sur une autre partition.

La 3<sup>ème</sup> colonne définit le format du système de fichier. Ainsi, les partitions `/` et `/home` sont formatées en mode `ext4` (mode courant sous Linux), tandis que la partition `/boot` est formatée en mode `xfs` (type de partition plus récent). Le format est déterminé à l'installation du système.

---

<sup>7</sup> La mise à disposition du système. Au cours de cette opération, le système vérifie la cohérence des données.

La 4<sup>ème</sup> colonne détermine les options que l'on peut préciser lors du *montage* de la partition. Nous remarquons qu'actuellement, l'option `defaults` est la seule qui est précisée. Les options suivantes sont possibles (sans que cette liste ne soit exhaustive, consultez la page de manuel de `fstab` pour plus de précisions) :

Option	Explication
<code>sync</code> <code>async</code>	Les opérations sur le système de fichiers sont faites de manière synchrone ( <code>sync</code> ) ou asynchrone ( <code>async</code> ).
<code>auto</code> <code>noauto</code>	La partition est ( <code>auto</code> ) ou n'est pas ( <code>noauto</code> ) montée automatiquement au démarrage du système.
<code>exec</code> <code>noexec</code>	Cette option autorise ( <code>exec</code> ) ou interdit ( <code>noexec</code> ) l'exécution de programme depuis cette partition.
<code>suid</code> <code>nosuid</code>	Cette option autorise ( <code>suid</code> ) ou interdit ( <code>nosuid</code> ) le prise en compte de la permission setuid <sup>8</sup>
<code>rw</code> <code>ro</code>	Cette option permet de <i>monter</i> le système en lecture seule ( <code>ro</code> ) ou en lecture-écriture ( <code>rw</code> ).
<code>user</code> <code>nouser</code> <code>users</code>	Cette option particulière permet de contrôler si le système de fichier correspondant peut être <i>monté/démonté</i> par un utilisateur. Ainsi, les options <code>user</code> et <code>users</code> permettent cette particularité.  Par contre, l'option <code>nouser</code> réserve le droit de <i>monter / démonter</i> le système de fichier à l'administrateur ( <code>root</code> ) seulement.
<code>defaults</code>	Il s'agit d'un raccourci pour les options suivantes : <code>rw</code> , <code>suid</code> , <code>dev</code> , <code>exec</code> , <code>auto</code> , <code>nouser</code> et <code>async</code>
<code>grpquota</code>	Cette option active la prise en charge des <i>quotas groupes</i> collectif.
<code>usrquota</code>	Cette option active la prise en charge des <i>quotas utilisateurs</i> .

### 3.6 Les quotas

Les systèmes Linux proposent une gestion des quotas disques. Un quota est actif *sur une partition entière*. Grâce aux quotas, il est possible de limiter une ressource, qui peut être partagée entre plusieurs utilisateurs, de manière équitable.

Les quotas peuvent être définis par utilisateur ou collectivement pour un groupe (le quota s'appliquant ainsi au groupe, dans son ensemble).

Avant de pouvoir utiliser les quotas, **il est indispensable de les activer sur le système de fichier concerné.**

Ainsi, si nous souhaitons activer les quotas sur la partition `/home`, il faut éditer le fichier `fstab` pour ajouter les options suivantes :

```
/dev/mapper/centos-home /home ext4 defaults,grpquota,usrquota 1 2
```

Ces modifications doivent être apportées prudemment.

---

<sup>8</sup> Cette permission particulière n'a pas été abordée dans le cadre de ce cours. Simplement, elle permet de changer temporairement d'utilisateur pour l'exécution d'une commande ou d'un programme. Cette particularité est utilisée pour permettre à des utilisateurs d'effectuer des tâches d'administration, par exemple. Ainsi, la commande `mount (/bin/mount)` utilise cette permission particulière.

### 3.6.1 Stratégie de mise en place des quotas

La stratégie de mise en place des quotas dans les systèmes Linux suit toujours le même schéma :

1. Activation, par modification du fichier `fstab`, du support des quotas utilisateurs et/ou de groupes
2. *Démonter et Remonter la partition `/home`, afin de prendre en compte les changements réalisés dans `fstab`:*  
\$ `umount /home`  
\$ `mount /home`  
*Ou plus simplement :*  
\$ `mount /home -o remount`

Pour contrôler que les options ont bien été prises en compte, il faut simplement exécuter :

```
$ mount | grep /home
```

Normalement, les options `usrquota` et `grpquota` devraient apparaître dans le résultat.

3. L'étape suivante consiste à créer les fichiers qui vont gérer les informations de quota. Ces fichiers nommés `aquota.user` et `aquota.group` sont créés automatiquement par la commande `quotacheck`:  
\$ `quotacheck -aucvg`<sup>9</sup>
4. Il faut enfin informer le système d'exploitation que des quotas doivent être vérifiés sur le système de fichier considéré :  
\$ `quotaon /home`

Une fois activé, les quotas mémorisés dans les fichiers `aquota.user` et `aquota.group` sont d'application. Un utilisateur peut alors être contraint de respecter les limites qui sont imposées.

### 3.6.2 Edition des quotas

Une fois que le système de quota est actif, il est possible d'ajouter des quotas pour des utilisateurs ou, collectivement, pour un groupe donné. Nous allons désormais détailler ce point.

#### *Les limites*

Le quota disque limite l'espace disponible pour un utilisateur ou, collectivement, pour un groupe. Linux, dans sa gestion des quotas définit plusieurs paramètres. Ainsi, deux limites sont proposées : la *limite soft* et la *limite hard*. De plus, une période de temps, la *grace time period* est également précisée.

En fait, c'est assez simple : la *limite soft* est la limite en dessous de laquelle l'utilisateur doit, en moyenne, se trouver. La *limite hard* est la borne infranchissable définie pour l'utilisateur. La période

---

<sup>9</sup> L'option `-a` détermine que tous les systèmes de fichiers dont les quotas sont activés dans `fstab` vont être pris en compte, l'option `-u` traite les quotas utilisateurs, l'option `-g` traite les quotas groupes, l'option `-v` propose un résultat verbeux et, enfin, l'option `-c` provoque la création des fichiers nécessaires à la gestion des quotas.

*grace time period* définit le temps pendant lequel la *limite soft* peut être dépassée. Au-delà de ce temps, la *limite soft* se transforme en *limite hard* pour cet utilisateur.

Les *limites soft* et *hard* peuvent être définies en *nombre de blocs* (dans notre cas de 1 Ko - et donc, limite en volume -) et/ou en *nombre d'inodes* (et donc, en nombre de fichiers).

La période de temps *grace time period* par défaut est **7 jours**.

### Fixer un quota utilisateur

Pour fixer un quota utilisateur, il y a 2 possibilités : le mode interactif et la ligne de commande.

Le mode interactif utilise la commande `edquota`, qui lance l'éditeur par défaut<sup>10</sup> :

```
$ edquota -u lsw
Disk quotas for user lsw (uid 1000):
  Filesystem          blocks      soft      hard      inodes      soft      hard
    /dev/mapper/centos-home   3828        0        0     131        0        0
```

Le système nous affiche la consommation actuelle de cet utilisateur en nombre de blocs de 1 Ko (soit ici 3 828 Ko), les *limites soft* et *hard* en nombre de blocs actuellement définies (actuellement 0). Ensuite, nous avons le nombre d'*inodes* actuellement utilisés (et donc le nombre de fichiers de cet utilisateur) et les limites configurées (actuellement 0).

Pour modifier le quota de l'utilisateur, il faut simplement éditer les colonnes *soft* ou *hard* souhaitées et enregistrer les modifications.

Le mode en ligne de commande utilise la commande `setquota`. Cette commande est particulièrement intéressante car elle peut avantageusement être intégrée dans des scripts PERL pour fixer automatiquement les quotas des utilisateurs :

```
$ setquota -u lsw 450000 500000 0 0 /home
```

La commande demande de préciser l'utilisateur (par l'option `-u`), les *limites* en nombre de blocs *soft* 450000 Ko (ou 450 Mo environ) et *hard* 500 000 Ko (ou 500 Mo environ) et ensuite, les limites en nombre d'*inodes* (0 représente aucune limite). Enfin, il faut préciser le système de fichiers concerné (ici via le point de montage : `/home`).

### Fixer un quota groupe

Fixer un quota groupe collectif est presque identique à celui d'un utilisateur. Précisément, nous avons toujours parlé de quota groupe collectif : cela signifie que, collectivement, tous les utilisateurs membres du groupe renseigné dans le quota sont soumis, ensemble, aux limites précisées.

Ainsi, fixer un quota en nombre de blocs de 1000 Mo pour le groupe *users* signifie que tous les utilisateurs, collectivement, peuvent enregistrer des fichiers pour un volume total d'environ 1 Go. Il est admis qu'un utilisateur consomme 900 Mo et tous les autres, le reste : il s'agit bien d'un quota collectif.

<sup>10</sup> L'éditeur par défaut peut être modifié par la variable d'environnement `EDITOR`. Il est possible également de lancer la commande `edquota` en modifiant temporairement cette variable comme suit :

```
$ EDITOR=gedit edquota -u lsw
```

Pour définir un quota groupe, nous pouvons ici aussi, procéder en mode interactif ou via la ligne de commande. En mode interactif :

```
$ edquota -g users
```

Ou, par la ligne de commande :

```
$ setquota -g users 1000000 5000000 0 0 /home
```

### 3.7 Exercices

1. Installer les quotas sur la partition `/home`
2. Préciser que
  - a. Pour le compte créé pour votre voisin (voir leçons précédentes), la limite est de 250 Mo
  - b. Pour les membres du groupe *etudiant* (voir leçons précédentes), la limite collective est de 500 Mo
  - c. Pour l'utilisateur *bm1*, la limite est de 150 Mo
3. Vérifier que les quotas fonctionnent en copiant un large fichier dans le répertoire d'un utilisateur<sup>11</sup>.
4. Déterminer, à l'aide de la commande `du`<sup>12</sup>, l'espace disque occupé par chaque utilisateur.
5. Modifier le script de création des utilisateurs (voir leçons précédentes) pour inclure les quotas suivants :
  - a. Chaque étudiant de 1<sup>ère</sup> année aura un quota de 150 Mo
  - b. Chaque étudiant de 2<sup>ème</sup> année et 3<sup>ème</sup> année aura un quota de 200 Mo

Vérifier, avec `edquota`, que ceux-ci sont effectivement bien configurés.

---

<sup>11</sup> Pour que cette vérification puisse se faire, il faut que ce soit l'utilisateur en question qui effectue la copie.

<sup>12</sup> `du` est une commande UNIX permettant de connaître l'espace consommé (*disk usage*)

## Leçon 4 : Sauvegarde et planification

### 4.1 Introduction

Dans cette leçon, nous allons étudier les moyens de sauvegarder *en-ligne* l'information. Comme les serveurs Linux sont souvent présents sur Internet et connectés en permanence, il est très courant de sauvegarder les éléments essentiels (configuration, répertoire web, bases de données, ...) sur un autre serveur.

Il existe également des logiciels de sauvegarde plus classique. Ainsi, comme sur tous les systèmes, il existe des logiciels chargés de faire des copies de sauvegarde sur bande magnétique ou tout autre média utilisé pour le backup ou l'archivage. Cependant, si l'on souhaite réaliser un backup très rapide, la copie vers un serveur est l'un des moyens les plus simples.

### 4.2 Sauvegarde de fichiers

Pour sauvegarder des fichiers, on peut utiliser des *programmes d'archivage et de compression*. A l'instar des programmes permettant de créer des fichiers .ZIP, il existe, sous Linux, quelques formats courant pour la compression et la distribution de fichiers archivés.

#### 4.2.1 Le format standard ZIP

Le format ZIP, très ancien, est une véritable référence en moyen de distribution et de compression. Il n'est, par contre, pas très souvent utilisé dans le monde Unix. Cependant, les commandes standard `zip` et `unzip` sont présentes pour créer, manipuler ou extraire de tels fichiers.

##### Création d'une archive ZIP

```
$ zip -r /tmp/save-etc.zip /etc/*
```

*L'option `-r` assure que toute l'arborescence sera traitée, le fichier destination est d'abord mentionné et puis les fichiers à inclure dans l'archive. Dans cet exemple, une archive nommée `save-etc.zip` est créée dans le dossier `/tmp` et reprend l'ensemble des répertoires et sous-répertoires du dossier `/etc`. Les informations propres à Unix (propriétaire, permission, ...) sont perdues.*

##### Extraction d'une archive ZIP

```
$ unzip monfichier.zip
```

*Cette commande extrait le fichier `monfichier.zip` dans le répertoire courant.*

##### Lister les fichiers d'une archive ZIP

```
$ unzip -l monfichier.zip
```

*Cette commande permet de connaître les fichiers et dossiers qui se trouvent dans une archive ZIP.*

#### 4.2.2 Les formats tar.gz et tar.bz2

Parmi les formats de fichiers courant sous Linux, nous avons les formats `.tar.gz` ou `.tar.bz2`. La double extension mentionnée fait référence à 2 logiciels très particuliers.

Le premier, `tar`, permet de transformer un répertoire et ses fichiers en un seul fichier (une sorte de « regroupement »).

Les commandes `gzip` ou `bzip2` sont des programmes de compression : elles permettent de compresser un fichier (ici, en l'occurrence, le fichier `.tar`).

Sous **Linux**, il est possible de combiner ces commandes en une seule, alors que certains systèmes Unix nécessitent d'exécuter la commande `tar` puis `gzip` (ou `bzip2` au choix de l'utilisateur) successivement.

### Création d'une archive

```
$ tar cvzf /tmp/save-etc.tar.gz /etc  
$ tar cvjf /tmp/save-etc.tar.bz2 /etc
```

Dans le premier exemple, nous allons créer une archive `save-etc.tar.gz`. Cette archive contiendra le dossier `/etc` et tous les fichiers et dossiers contenus dans celui-ci. Le paramètre `c` permet de créer l'archive, le paramètre `v` provoque un affichage verbeux, le paramètre `z` indique d'exécuter `gzip` après la création du fichier `.tar` et enfin, le paramètre `f` indique que la destination est un fichier.

Le second exemple diffère simplement par le paramètre `j` et le nom de l'archive. Ici, c'est la commande `bzip2` qui est exécutée suite à la création du fichier `.tar`.

### Extraction d'une archive

```
$ tar xvzf /tmp/save-etc.tar.gz  
$ tar xvjf /tmp/save-etc.tar.bz2
```

Dans le premier exemple, nous allons extraire l'archive `save-etc.tar.gz` qui se trouve dans le dossier `/tmp`. L'archive sera extraite dans le répertoire courant. Le format de compression est `gzip`. Enfin, l'option `x` permet d'informer `tar` de réaliser l'extraction.

Dans le second exemple, nous allons extraire l'archive `save-etc.tar.bz2` qui se trouve dans le dossier `/tmp`. L'archive sera extraite dans le répertoire courant. Le format de compression est `bzip2`.

### Tester / lister les fichiers d'une archive

```
$ tar tvzf /tmp/save-etc.tar.gz  
$ tar tvjf /tmp/save-etc.tar.bz2
```

Ces deux commandes permettent de lister les fichiers et dossiers de l'archive `save-etc.tar.gz` ou `save-etc.tar.bz2`.

## 4.3 Sauvegarde d'une image disque

Une autre possibilité intéressante est de réaliser une sauvegarde d'un système complet en vue de sa duplication sur un grand nombre de machines. Cette possibilité est très utilisée dans les parcs d'ordinateur homogène (même configuration matérielle).

Parmi les logiciels connus<sup>13</sup>, nous avons Symantec Ghost Solution Suite, Acronis True Image, Paragon Hard Disk Manager. A coté de ces solutions commerciales, il y a, également, CloneZilla (<http://www.clonezilla.org>), comprenant une suite d'outils gratuits et open-sources permettant la duplication des machines.

---

<sup>13</sup> Cette liste ne reprend que les logiciels que j'ai été amené à utiliser. Il en existe certainement bien d'autres.

## 4.4 Sauvegarde incrémentale et différentielle

A coté des outils permettant des sauvegardes complètes du système, il y a aussi bon nombre de logiciels qui proposent des sauvegardes régulières et se basant sur les sauvegardes précédentes, déjà réalisées.

Ainsi, on parle de *sauvegarde complète* quand l'ensemble du système est sauvegardé. Une *sauvegarde incrémentale* est une sauvegarde reprenant les dernières modifications depuis la dernière sauvegarde incrémentale ou complète réalisée. Dans ce scénario, la restauration des données exige de rétablir la dernière sauvegarde complète et tous les incréments effectués.

On parle également de *sauvegarde différentielle* qui est une sauvegarde des différences par rapport à la dernière sauvegarde complète du système. Dans ce scénario, la restauration du système exige de rétablir la dernière sauvegarde complète et la dernière différentielle effectuée.

Dans le monde Unix, il y a également beaucoup de solution de sauvegarde permettant ce type de backup. A l'instar de Symantec BackupExec, qui est un des logiciels de backup les plus connus dans le monde Windows, des solutions analogues sont disponibles sous Linux :

- **MondoRescue** (<http://www.mondorescue.org/>) permet de faire une sauvegarde du système sur un périphérique externe, disque dur, disque réseau, ...
- **Bacula** (<http://www.bacula.org/>) permet de faire des sauvegardes complètes, incrémentales et différentielles. Les sauvegardes peuvent être centralisées sur un lecteur de bande, ... La solution Bacula est l'une des plus complètes dans le monde libre Linux.
- **Amanda** (<http://www.amanda.org/>) est un autre logiciel de sauvegarde dont les fonctionnalités sont proches également de Bacula. Ce logiciel a été développé par l'université du Maryland.

## 4.5 Sauvegarde en réseau

Autre moyen de sauvegarde très courant sur les serveurs, c'est la copie de données vers un autre serveur pour backup. Cette copie *en live, planifiée* permet une récupération des fichiers quasi-immédiate.

Les outils standard pour réaliser ces opérations sont :

- `scp` qui permet de faire une copie de fichiers vers une machine distante, en utilisant le service ssh installé par défaut sur la plupart des serveurs Linux.
- `rsync` qui permet de synchroniser deux dossiers (pour créer un *miroir* d'un dossier). Son atout étant que seules les modifications sont transférées.
- `lftp` - dont nous parlerons plus tard - permet de sauvegarder des fichiers sur un site FTP quelconque. Cette option est très intéressante lorsqu'on loue des serveurs *cloud* sur internet car il est courant que le fournisseur propose un espace FTP de backup.

### 4.5.1 Copie de fichiers à distance

```
$ scp monfichier.tar.gz p010544@dartagnan.cg.helmo.be:/tmp
```

*Copie le fichier monfichier.tar.gz sur la machine dartagnan, dans le répertoire /tmp, en utilisant le compte p010544.*

Dans ces commandes, on doit toujours mentionner *la source* puis *la destination*. La *source* ou la *destination* peuvent être locale ou distante. Ainsi, si l'on souhaite copier le fichier depuis `dartagnan`, nous pourrions écrire :

```
$ scp p010544@dartagnan.cg.helmo.be:/tmp/monfichier.tar.gz .
```

*Dans cet exemple, nous copions le fichier `monfichier.tar.gz` depuis le serveur `dartagnan` vers le répertoire courant (notez le « `.` » comme destination, désignant le dossier courant).*

Pour que cette copie fonctionne, il faut que le système distant exécute le service SSH. Nous étudierons celui-ci plus loin dans le cadre de ce cours.

#### 4.5.2 Création d'un miroir

La notion de dossier miroir est assez simple à comprendre : il s'agit de *synchroniser* deux dossiers de sorte que le dossier destination (appelé miroir du premier) contiennent au moins les fichiers du dossier source. L'intérêt étant que la synchronisation ne transfère que les fichiers modifiés ou ajoutés et n'existant pas sur le dossier destination.

Cette possibilité est donc intéressante si le répertoire source est important et que les mises à jour ne concerne que certains fichiers.

```
$ rsync -Cavz /home p010544@dartagnan.cg.helmo.be:~/backup -e ssh
```

*Cette commande crée un dossier miroir du dossier `/home` du serveur. Ainsi, l'objectif est que le dossier miroir distant contiennent tous les fichiers contenu dans `/home` pour des raisons de backup. La commande `rsync` permet de réaliser cette synchronisation. Comme pour `scp`, il faut mentionner le dossier source (qui sera copié) et le dossier destination (qui sera le miroir). Le dossier source ou destination peut être local ou distant. Dans cet exemple, le dossier local est `/home` et le dossier distant est `~/backup` (pour rappel « `~` » fait référence au dossier personnel de l'utilisateur, ici `p010544`, ce qui revient à dire qu'il s'agit du dossier distant `/home/p010544/backup`).*

## 4.6 Planification des tâches

Nous avons vu comment il était possible de créer des sauvegardes. Cependant, une sauvegarde ponctuelle ou réalisée par l'utilisateur, n'est pas très intéressante. Seule les sauvegardes planifiées, automatiques et programmées sont réellement intéressante car l'utilisateur n'intervenant pas dans le processus, il n'y a pas de danger que celle-ci soit oubliée.

Il y a deux types de planification possibles :

1. Les tâches *ponctuelles* qui sont lancées une seule fois à un moment déterminé. Ces tâches sont exécutées puis leur planification est supprimée.
2. Les tâches *répétitives* qui sont lancées régulièrement sur le système. La planification de celles-ci est mémorisée et le système maintient celles-ci.

Les commandes `at` et `cron` sont utilisées pour ces deux types. La commande `at` permet de planifier l'exécution d'une tâche ponctuelle alors que `cron` permet de planifier l'exécution de tâches répétitives.

Pour des raisons de sécurité, il est possible de limiter les utilisateurs autorisés à exécuter ces commandes. Par défaut, tous les utilisateurs peuvent programmer des tâches ponctuelles ou répétitives sur les systèmes CentOS 7. Il est possible de limiter les utilisateurs en ajoutant des éléments dans `/etc/cron.allow`, `/etc/cron.deny`, `/etc/at.allow` ou `/etc/at.deny`. Pour plus d'information, référez-vous aux pages de manuel de ces commandes.

**Attention !** Comme il s'agit de tâches planifiées, lancées automatiquement, *il ne peut pas y avoir d'interaction avec l'utilisateur*. Ces tâches doivent s'exécuter complètement automatiquement. L'environnement utilisé est réduit : il n'y a pas de `stdin` (ou entrée standard), ni de `stdout` (ou sortie standard). Ainsi, si l'application utilisée nécessite une entrée au clavier, il peut être intéressant de faire une redirection. Par défaut, toutes les sorties produites sont envoyées par mail à l'utilisateur ayant programmé la tâche.

#### 4.6.1 Les tâches ponctuelles

Une tâche ponctuelle se planifie grâce à la commande `at`. L'expression du moment de démarrage est assez large :

```
$ at 14:10
at> /usr/sbin/reboot
at> ^D
job 1 at Sun Sep 20 14:10:00 2015
```

Dans cette exemple, nous planifions l'exécution de la commande `reboot` aujourd'hui à 14h10 (soit le dimanche 20 septembre à 14h10). Par précaution, il vaut toujours mieux préciser le chemin complet de la commande ou du script qui doit être démarré. Une fois la tâche planifiée, on quitte et sauvegarde la planification en appuyant sur CTRL+D.

La commande `atq` permet de lister les tâches planifiées pour l'utilisateur courant et `atrm` permet de supprimer une tâche planifiée avec `at`.

Il est possible de préciser bien d'autres planifications.

Exemples	Explication
<code>at 14:3009202015</code>	Planification le 20 septembre 2015 à 14h30
<code>at 14:30 sep 20 2015</code>	Planification le 20 septembre 2015 à 14h30
<code>at now + 5 hours</code>	Planification dans 5 heures à partir de maintenant
<code>at now + 1 day</code>	Planification dans 1 jour à partir de maintenant
<code>at 11:00 next month</code>	Planification le même jour dans un mois, à 11h
<code>at 22:00 tomorrow</code>	Planification pour demain à 22h

#### 4.6.2 Les tâches répétitives

La planification de tâches répétitives se fait grâce à la commande `crontab`. Le format, un peu étrange, à utiliser permet d'être précis dans la planification.

Pour planifier une tâche répétitive, il faut utiliser :

```
$ crontab -e14
```

<sup>14</sup> Cette commande lance l'éditeur par défaut. Il est possible de changer d'éditeur en entrant:  
`$ EDITOR=gedit crontab -e`

La planification se fait en colonne, dans l'ordre suivant :

M        H        j        m        J        commande

Avec :

- M représentant la minute
- H représentant l'heure
- j représentant le jour
- m représentant le mois
- J représentant le jour de la semaine.
- commande représentant la commande à exécuter

Dans chaque colonne, on mentionne la donnée souhaitée ou « \* » pour dire « à chaque ».

Planification	Explication
0     0     20    09    1     cal	Exécute la commande cal tous les 20 septembre à minuit lorsque c'est un lundi (lundi = jour 1 ; dimanche = jour 7)
0     0     20    *     *     cal	Exécute la commande cal tous les 20 de chaque mois à minuit, peu importe le jour de la semaine
0     0     *     *     *     cal	Exécute la commande cal tous les jours à minuit quelque soit le jour, le mois ou le jour de la semaine.

On remarque que cette planification est finalement assez précise et simple à comprendre. Il est très courant de lancer des scripts par `crontab` afin de réaliser des tâches précises comme les backup automatiques.

## 4.7 Exercices

1. Réaliser une sauvegarde du répertoire `/var` dans un fichier `/tmp/var-back.bz2`. Ce fichier sera compressé par `bzip2`.
2. Réalisez le même exercice que le précédent, en utilisant la commande `zip` : créer l'archive `/tmp/var-back.zip` contenant l'ensemble du dossier `/var`.
3. Testez et décompressez l'une des archives précédentes dans votre dossier personnel
4. Programmez une tâche ponctuelle, pour le cours prochain, au milieu de celui-ci, et lancer la commande `poweroff`.
5. Programmez une tâche répétitive s'exécutant tous les début de cours et exécutant la commande `ntpdate` (synchronisation d'horloge avec un serveur distant) vers le serveur `time.belnet.be`
6. Programmez pour la fin du cours la synchronisation de vos répertoires personnels (`/home`) vers le serveur `dartagnan`. Créer dans votre dossier personnel sur `dartagnan` un dossier `back` qui deviendra le miroir du votre dossier `/home`. Est-ce que cela fonctionne ? Pourquoi ?

## Leçon 5 : Configuration réseau et démarrage / arrêt du système

### 5.1 Modifier la configuration réseau

#### 5.1.1 Les interfaces réseaux

Une *interface réseau* dans les systèmes Linux est une *carte réseau* physique ou virtuelle, qui est présente sur le système. Les interfaces réseaux sont nommées de manière unique afin de pouvoir configurer l'environnement réseau complètement.

Ainsi, sur certains systèmes Linux, les interfaces sont nommées `eth0`, `eth1`, ... (pour ethernet) ou encore `em0`, `em1`, .... Le nom des interfaces évolue beaucoup suivant la version du noyau Linux utilisé (afin de permettre une identification claire et non-ambigüe). Sur les systèmes CentOS 7, le nom des interfaces ethernet commencent par `eno` (`en` pour ethernet, `o` pour périphérique *on-board* et ensuite, un numéro identifiant provenant du matériel). Ainsi, l'interface réseau installée sur la machine se nomme `eno16777736`.

Les outils pour gérer les interfaces réseaux ont également beaucoup évolués. Sur tous les systèmes, on trouve les commandes classiques `ifconfig`, `netstat` ou `route`. Sur les systèmes plus récents, en plus de continuer à supporter les commandes *classiques*, la nouvelle commande `ip` est présente. Dans la suite du cours, les deux versions (anciennes et nouvelles) seront présentées.

Il est possible de visualiser les interfaces réseaux actives en utilisant les commandes `ip` suivantes :

```
[root@localhost Desktop]# ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eno16777736: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:0c:29:56:25:af brd ff:ff:ff:ff:ff:ff
    inet 192.168.190.101/24 brd 192.168.190.255 scope global dynamic eno16777736
        valid_lft 6555sec preferred_lft 6555sec
    inet6 fe80::20c:29ff:fe56:25af/64 scope link
        valid_lft forever preferred_lft forever
```

Cette commande affiche les interfaces (`lo` et `eno16777736` dans notre exemple), les informations sur celles-ci et les configurations réseaux associées.

Pour visualiser uniquement les informations au niveau de la couche liaison de données :

```
[root@localhost Desktop]# ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eno16777736: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT qlen 1000
    link/ether 00:0c:29:56:25:af brd ff:ff:ff:ff:ff:ff
```

Avec la commande `ifconfig`, il est possible de visualiser ces mêmes informations :

```
[root@localhost Desktop]# ifconfig

eno1677736: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.190.101 netmask 255.255.255.0 broadcast 192.168.190.255
        inet6 fe80::20c:29ff:fe56:25af prefixlen 64 scopeid 0x20<link>
            ether 00:0c:29:56:25:af txqueuelen 1000 (Ethernet)
            RX packets 125 bytes 14556 (14.2 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 83 bytes 10328 (10.0 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
            loop txqueuelen 0 (Local Loopback)
            RX packets 727 bytes 98958 (96.6 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 727 bytes 98958 (96.6 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Cette commande affiche la configuration des différentes interfaces réseaux actives. Avec l'option `-a`, les interfaces inactives ou non-configurées apparaissent également.

Grâce à ces commandes, il est également possible de modifier la configuration courante d'une interface (cela signifie qu'au prochain redémarrage, la configuration d'origine, sauvegardée, sera à nouveau installée).

Ainsi, avec la commande `ip`, il faut supprimer l'adresse précédente et en ajouter une nouvelle :

```
$ ip addr del 192.168.190.101/24 dev eno1677736
$ ip addr add 10.0.1.2/24 dev eno1677736
```

La commande `ifconfig` suivante modifie également la configuration courante de l'interface :

```
$ ifconfig eno1677736 10.0.1.2 netmask 255.255.255.0
```

Cette commande affecte l'adresse 10.0.1.2 à l'interface eno1677736 avec un masque /24.

Pour afficher la **table de routage** de la machine, nous avons les deux commandes suivantes :

```
[root@localhost Desktop]# ip route show
default via 192.168.190.2 dev eno1677736 proto static metric 100
192.168.190.0/24 dev eno1677736 proto kernel scope link src
192.168.190.101 metric 100

[root@localhost Desktop]# netstat -r -n
Kernel IP routing table
Destination     Gateway         Genmask        Flags   MSS Window irtt Iface
0.0.0.0          192.168.190.2  0.0.0.0        UG        0 0          0 eno1677736
192.168.190.0   0.0.0.0        255.255.255.0  U         0 0          0 eno1677736
```

Dans les 2 cas, nous avons la **route par défaut** (appelée `default` ou `0.0.0.0`) qui utilise le relai `192.168.190.2` (adresse IP de pfSense) pour atteindre internet. Ce relai est connecté sur l'interface `eno1677736`. La seconde ligne mentionne que la machine est **directement connectée** (sans relai donc) au réseau `192.168.190.0/24` (aussi associé à l'interface `eno1677736`).

## Modifier les paramètres réseaux configurés

Pour modifier définitivement la configuration réseau, nous pouvons utiliser différents moyens : *Webmin* peut être utilisé, tout comme un programme particulier, *NetworkManager Client* ou encore la modification manuelle des fichiers de configuration. Comme ces 3 méthodes permettent d'atteindre cet objectif, peu importe celle que vous décidez d'appliquer. Nous allons expliquer certaines d'entre-elles :

1. Via **Webmin**, il faut aller dans **Networking > Network Configuration**. Il faut ensuite cliquer sur **Network Interfaces** puis cliquer sur le nom de l'interface réseau à configurer (par exemple `eno16777736`). Il est alors possible d'entrer la configuration réseau souhaitée. Une fois les modifications terminées, il faut cliquer sur *Save and Apply* pour voir cette configuration devenir active.

**Attention !** Il faudra également configurer d'autres éléments réseaux importants comme la route par défaut (**Networking > Network Configuration** puis **Routing and Gateways**) et les serveurs DNS à interroger (**Networking > Network Configuration** puis **Hostname and DNS Client**).

2. En utilisant l'outil **Network Manager**. Cet outil permet de créer *des profils* contenant toutes la configuration réseau souhaitée. Ainsi, suivant l'emplacement, il est possible d'activer un ou l'autre profil. Il est clair que sur une configuration serveur, la plupart du temps, un seul profil sera utilisé (ce qui n'est pas le cas si nous avions installé CentOS sur un portable par exemple, ou nous pourrions avoir des profils différents pour le domicile ou le bureau). A nouveau, 2 outils sont présents : soit une version en ligne de commande, soit une version interactive :

- a. La version **en ligne de commande** est assez simple à utiliser. Nous pouvons commencer par lister les profils présents :

```
[root@localhost Desktop]# nmcli device status
DEVICE      TYPE      STATE      CONNECTION
eno16777736  ethernet  connected  eno16777736
lo          loopback  unmanaged  --
```

Nous observons ici que l'interface `eno16777736` est associée à une connexion (= un profil) du même nom. Il est tout à fait possible de lire, adapter ou supprimer ce profil.

Pour visualiser ce profil :

```
[root@localhost Desktop]# nmcli -p connection show eno16777736
```

Pour utiliser une adresse IP statique à la place du mode DHCP, nous utiliserons la commande suivante :

```
[root@localhost Desktop]# nmcli connection modify eno16777736
ipv4.method manual ipv4.address 10.0.1.2/24 ipv4.gateway
10.0.1.1 ipv4.dns 8.8.8.8 ipv4.dns-search localdomain
```

Cette commande modifie le profil `eno16777736` pour utiliser l'adresse IP statique `10.0.1.2` avec le masque `/24`. La passerelle par défaut est également configurée à `10.0.1.1` et le serveur DNS à interroger est `8.8.8.8`.

Pour activer les modifications, il est possible de redémarrer la machine ou simplement lancer les commandes :

```
$ nmcli device disconnect eno16777736  
$ nmcli device connect eno16777736
```

Pour revenir à une configuration utilisant DHCP, il faut entrer :

```
[root@localhost Desktop]# nmcli connection modify eno16777736  
ipv4.method auto
```

- b. La version **interactive** peut être lancée par la commande :

```
$ nmtui
```

*Cette commande lance un outil de configuration interactif qui permet de modifier une connexion.*

Ainsi, l'option **Edit a connection** permet de modifier les profils présents (dans notre cas, eno16777736). Il est ensuite possible de modifier la configuration IPv4 (en basculant de *automatic* vers *manual*) et mentionner les paramètres réseaux à configurer.

3. En modifiant les **fichiers de configuration à la main** directement. Il faut commencer par le fichier `/etc/sysconfig/network-scripts/ifcfg-eno16777736` (pour une interface portant ce nom).

Ce fichier contient quelques lignes (les lignes en gras sont ajoutées ou modifiées, les lignes barrées sont supprimées) :

```
TYPE="Ethernet"  
BOOTPROTO=none  
NM_CONTROLLED="no"  
DEFROUTE="yes"  
IPV4_FAILURE_FATAL="no"  
IPV6INIT="yes"  
IPV6_AUTOCONF="yes"  
IPV6_DEFROUTE="yes"  
IPV6_FAILURE_FATAL="no"  
NAME="eno16777736"  
UUID="0b673394-de1e-492d-bf8e-5d65da08581c"  
DEVICE="eno16777736"  
ONBOOT="yes"  
PEERDNS=yes  
PEERROUTES=yes  
IPV6_PEERDNS=yes  
IPV6_PEERROUTES=yes  
IPADDR=10.0.1.2  
PREFIX=24  
GATEWAY=10.0.1.1  
DNS1=8.8.8.8  
DOMAIN=localdomain
```

Une fois la modification terminée, il faut activer celle-ci en utilisant la commande :

```
$ systemctl restart network
```

Cette commande redémarre toute la configuration réseau.

## Ajouter une 2<sup>ème</sup> adresse IP à une interface

Il est possible d'ajouter plusieurs adresses IP à la même interface. Ainsi les commandes suivantes (nouvelle et ancienne version) permettent de réaliser cet ajout de manière temporaire jusqu'au prochain redémarrage de la machine :

```
$ ip addr add 10.0.1.3/24 dev eno16777736  
$ ifconfig eno16777736:1 10.0.1.3 netmask 255.255.255.0
```

Ainsi, la machine peut maintenant recevoir des informations provenant de toutes les adresses IP configurée. Dans le 1<sup>er</sup> cas, nous ajoutons une nouvelle adresse à l'interface eno16777736. Dans le second cas, nous ajoutons une nouvelle interface, *virtuelle et liée à la première*, qui répond sur cette adresse IP.

La suppression de cette adresse se fait comme suit (suivant la méthode utilisée pour l'ajout) :

```
$ ip addr dele 10.0.1.3/24 dev eno16777736  
$ ifconfig eno16777736:1 down
```

Pour **rendre définitif**<sup>15</sup> une seconde adresse IP, il est possible de passer par *Webmin*, *NetworkManager* ou le *fichier de configuration* :

1. Via **Webmin**, il faut aller dans **Networking > Network Configuration**. Il faut ensuite cliquer sur **Network Interfaces** puis cliquer sur le nom de l'interface réseau à configurer (par exemple eno16777736).

Tout en dessous, nous avons *Virtual Interfaces*, il est alors possible de cliquer sur **Add virtual interface**. Dans l'écran suivant, il est possible de mentionner les paramètres pour cette interface virtuelle.

2. En utilisant **Network Manager**, nous pouvons réaliser l'opération :
  - a. En ligne de commande :  

```
$ nmcli connection modify eno16777736 +ipv4.addresses16  
10.0.1.3/24
```
  - b. Via l'outil interactif :  

```
$ nmcli
```

**Edit a connection** puis **choisir le profil concerné** (par exemple eno16777736), et enfin, via l'option *Add* sous l'adresse IP déjà configurée, il est possible d'en ajouter une seconde.
3. En **modifiant les fichiers de configuration directement** : /etc/sysconfig/network-scripts/ifcfg-eno16777736 (pour une interface portant ce nom). Pour ce faire, il faut **ajouter** les lignes suivantes :  
**IPADDR1=10.0.1.3**  
**PREFIX1=24**

<sup>15</sup> Il est nécessaire que l'adresse IP principale soit configurée statiquement (pas de DHCP)

<sup>16</sup> Pour supprimer cette configuration, il suffit de remplacer +ipv4.address par -ipv4.address

### 5.1.2 Désactiver / Redémarrer la configuration réseau

Nous avons, dans les exemples précédents, utilisés quelques commandes pour désactiver ou redémarrer la configuration réseau. Nous allons maintenant faire le point sur les différentes commandes permettant d'atteindre cet objectif.

En utilisant les outils *Network Manager* (pour autant que celui-ci soit utilisé), on peut désactiver le profil utilisé et le réactiver :

```
$ nmcli device disconnect eno16777736  
$ nmcli device connect eno16777736
```

En utilisant les outils *ip* ou *ifconfig*, on peut désactiver l'interface et la réactiver :

```
$ ip link set eno16777736 down  
$ ip link set eno16777736 up  
$ ifconfig eno16777736 down  
$ ifconfig eno16777736 up
```

Il est également possible de redémarrer la configuration complète du système linux comme suit :

```
$ systemctl restart network
```

### 5.2 Vérifier la configuration réseau

Comme nous venons de le remarquer, modifier la configuration est assez simple. Par contre, identifier le problème lorsqu'on est face à une configuration inadaptée n'est pas toujours aisé.

Comme nous pouvons le voir sur la figure 5.1, vérifier la configuration réseau peut se faire en suivant les étapes énoncées :

1. *Ping vers la passerelle par défaut.* L'objectif est de voir si une autre machine directement connectée à celle-ci répond aux requêtes ( ! Vérifiez bien qu'aucun firewall n'est actif !) Si la machine ne répond pas, il faut commencer par vérifier si la configuration réseau de base (adresse IP, masque, interface réseau, ...) est correcte.
2. *Ping vers 8.8.8.8.* L'objectif est de tester s'il est possible de sortir du réseau. Si aucune réponse n'est obtenue, il faut vérifier si la connexion internet est bien disponible, si la passerelle utilisée est la bonne.
3. *Ping vers www.google.com ou www.yahoo.fr.* L'objectif est de voir si un nom DNS peut être résolu. Si aucune réponse n'est renvoyée, il faut contrôler le serveur DNS utilisé. Est-il fonctionnel ?

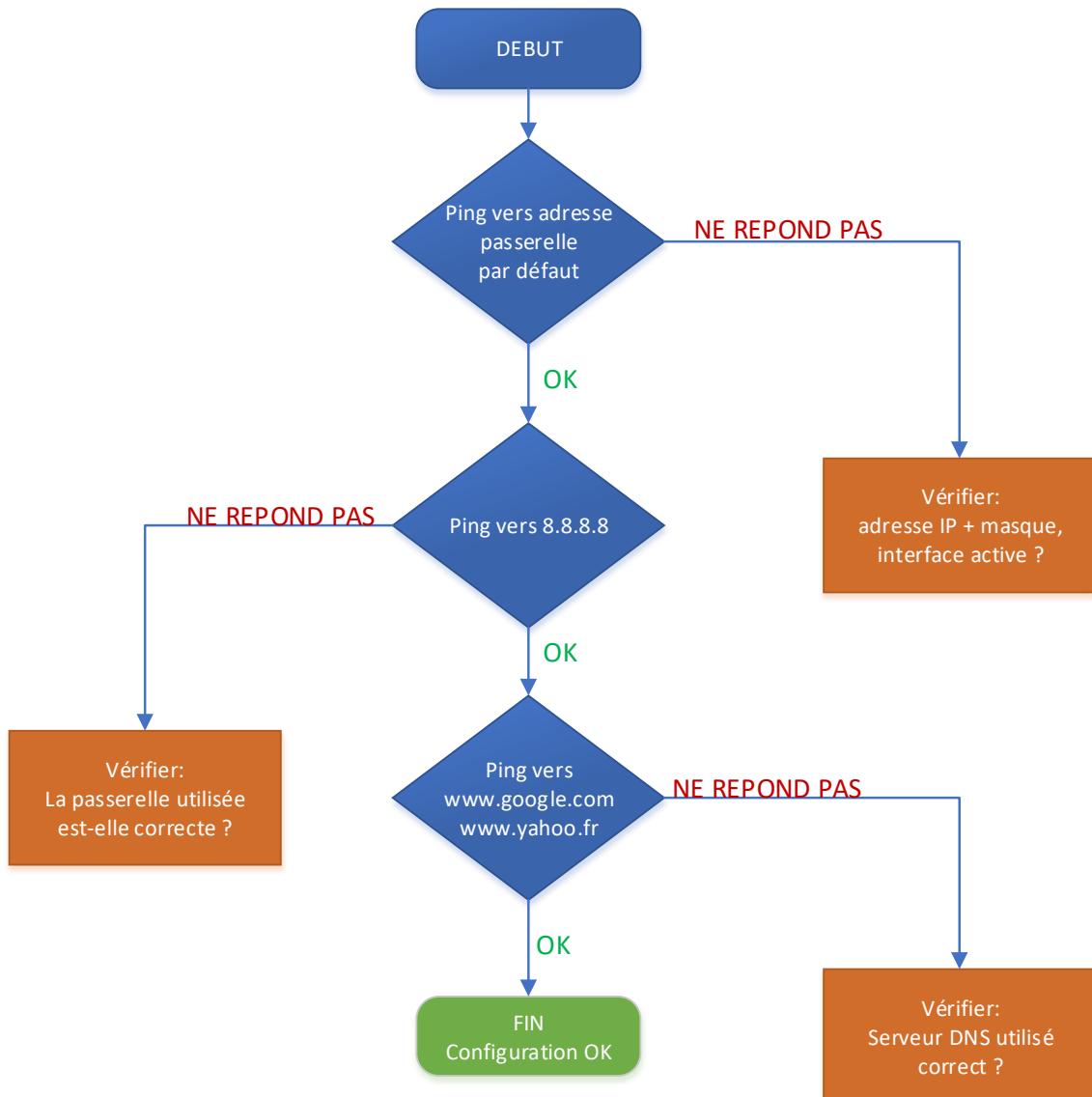


Figure 5.1 : Vérifier la configuration réseau

### 5.3 Quelques outils réseaux

Les outils réseaux abordés ici sont des outils courants, très souvent installés de base sur les systèmes. Ainsi, nous avons :

- **ping** qui permet de déterminer si les deux machines : locale (qui exécute la commande) et distante (qui est visée par la commande) peuvent s'échanger des messages ICMP echo-request/echo-reply. Cette commande est très souvent utilisée pour vérifier si les paramètres réseaux sont corrects.  
`$ ping 8.8.8.8`  
`$ ping www.google.com`

**Attention ! Les firewalls sont très souvent configurés pour bloquer ces requêtes. En effet, elles ont souvent été utilisées pour des attaques DDoS.**

- **traceroute** qui permet de connaître un chemin possible entre deux machines : locale (qui exécute la commande) et distante (qui est visée par la commande). Cette commande peut utiliser des paquets ICMP (option `-I`) ou UDP (option par défaut) pour tracer le chemin.  
\$ traceroute www.yahoo.fr  
\$ traceroute -I www.google.com

Comme pour la commande ping, traceroute est souvent bloqué par les firewall.

- **netstat** est un utilitaire ancien donnant des informations variées sur l'état réseau de la machine. Ainsi, en plus de donner la table de routage, il est possible de déterminer :  
\$ netstat -l -n  
*Liste les ports actuellement écoutés par un processus.*

\$ netstat -a

*La commande affiche les connexions en cours ou en attente.*

\$ netstat -i

*Cette commande affiche les statistiques réseaux.*

- **tcpdump** est l'utilitaire qui permet de capturer le trafic réseau en ligne de commande. Cet outil est particulièrement intéressant pour analyser l'origine d'un problème (le paquet est-il envoyé ? le paquet est-il reçu ?). De nombreux paramètres de filtrage peuvent être utilisés pour cibler le trafic souhaité.

\$ tcpdump -i eno16777736

*Cette commande capture tout le trafic qui traverse l'interface mentionnée.*

\$ tcpdump -i eno16777736 port 25

*Cette commande capture le trafic dont le port source ou destination est TCP/UDP 25.*

\$ tcpdump -i eno16777737 host 10.0.1.2

*Cette commande capture le trafic dont l'adresse IP source ou destination est 10.0.1.2*

Les pages de manuels donnent plus de détails sur les filtres qui peuvent être écrit.

- **wireshark** est la version graphique de l'outil **tcpdump**.  
\$ wireshark
- **arp** permet de gérer la table ARP de la machine. Pour rappel, la table ARP contient, de manière temporaire, les conversions entre les adresses MAC (ou adresses physiques) et les adresses IP. Si l'information se trouve dans la table, il n'est pas nécessaire de faire une résolution ARP<sup>17</sup>.

Il est également possible d'insérer *des entrées statiques* dans la table grâce à cet outil (ainsi, aucune requête ne sera faite pour cette adresse IP).

<sup>17</sup> Envoi en broadcast d'une trame ethernet posant la question « Quelle machine a l'adresse IP X.X.X.X ? »

```
$ arp -a
```

Affiche le contenu de la table ARP de la machine

- **nslookup** permet d'interroger, de manière interactive, un serveur DNS déterminé. Par défaut, c'est le serveur configuré sur la machine qui est interrogé.

```
$ nslookup  
> www.google.com
```

Demande la résolution du nom `www.google.com`

```
> server 8.8.8.8  
> www.swila.be
```

Change le serveur à interroger puis demande la résolution du nom `www.swila.be`

```
> 193.190.64.113
```

Demande le nom correspondant à l'adresse IP `193.190.64.113` (zone DNS inverse).

## 5.4 Démarrage du système

Le démarrage du système est géré par une multitude de logiciels prenant le relai les uns après les autres. Ainsi, tout débute avec *GRUB* qui est très souvent installé dans les premiers secteurs du disque dur. Celui-ci permet de sélectionner le système d'exploitation à lancer (il affiche un menu avec les différents systèmes et parfois, les différentes versions du noyau Linux). Ensuite, une fois que Linux commence à démarrer, c'est *systemd* qui prend le relai. Il s'agit du mécanisme démarrant l'ensemble des services souhaités par l'utilisateur ou l'administrateur. Ainsi, c'est grâce à *systemd* que le système démarre la *configuration réseau*, le *service SSH*, ou encore l'*interface graphique*.

L'ancien gestionnaire de démarrage, nommé *System V Init* est toujours présent pour les services ne supportant pas *systemd*.

Dans la suite, nous allons aborder chaque élément pour expliquer comment

### 5.4.1 GRUB2

GRUB<sup>18</sup> est le système de chargement lancé au démarrage de Linux. Il commence par afficher le système à démarrer (cela permet à l'utilisateur de démarrer une autre version du noyau linux par exemple) et puis passe la main au système sélectionné.

Il est possible d'utiliser GRUB pour démarrer d'autres systèmes d'exploitation comme Windows (ainsi les machines des labos fonctionnent ainsi). Il est possible d'adapter le fonctionnement de GRUB en modifiant les fichiers de configuration qui se trouvent dans `/etc/grub.d`. Une fois ces fichiers modifiés, il faut régénérer le fichier de configuration au moyen de la commande :

```
$ grub2-mkconfig -o /boot/grub2/grub.cfg
```

GRUB permet également, lors du démarrage, de saisir des options particulières. Ainsi, il est possible de démarrer le système Linux *en mode single* en ajoutant simplement l'option `single`. Le mode *single* est utilisé lorsque le système ne répond plus en mode normal ou lorsque l'administrateur a

---

<sup>18</sup> Acronyme de *Grand Unified Bootloader*

perdu le mot de passe *root*. En effet, le démarrage en mode *single* ouvre un terminal en *root* sans devoir fournir le moindre mot de passe.

Pour sécuriser l'installation, il est possible d'imposer un mot de passe *grub* avant de pouvoir spécifier toute option lors du démarrage. Pour ce faire, il suffit d'ajouter, dans le fichier `/etc/grub.d/40-custom`, les lignes suivantes :

```
set superusers="root"
password root passroot
```

Nous définissons le login *superuser* à *root* et ensuite, nous fixons le mot de passe du login *root* à *passroot*. Une fois le fichier modifié, il faut régénérer la configuration via la commande `grub2-mkconfig` montrée ci-dessus.

### 5.4.2 SystemD

*SystemD* acronyme de *system daemon* est l'ensemble logiciel permettant de déterminer quels services doivent démarrer lors du lancement du système. Il est composé de nombreuses commandes. Dans le cadre de notre cours, nous nous limiterons à quelques commandes standards.

#### Obtenir la liste des services

```
$ systemctl list-unit-files --type service
```

Cette commande liste la liste des services activés sur le système. Si l'on souhaite voir également la liste des services non-activé, il faut utiliser l'option `--all` en plus :

```
$ systemctl list-unit-files --type service --all
```

#### Démarrer / Arrêter un service

```
$ systemctl start sshd
```

Cette commande permet de démarrer le service (ou démon) SSH.

Les arguments possibles pour la commande `systemctl` sont nombreux, les plus courants sont `start` (démarrer), `stop` (arrête), `restart` (arrête et redémarre), `reload` (recharge la configuration) et `status` (affiche son état). Le dernier argument est toujours le service visé par la commande.

```
$ systemctl restart sshd
```

Cette commande permet de redémarrer le service (ou démon) SSH.

#### Activer / désactiver un service au démarrage

```
$ systemctl enable sshd
$ systemctl disable sshd
```

Les arguments `enable` (activer) et `disable` (désactiver) permettent de charger le service mentionné (ici, `sshd`) lors du démarrage de la machine.

### 5.4.3 System V Init

Comme indiqué dans l'introduction, *System V Init* est le mécanisme précédemment utilisé pour déterminer les services à démarrer lors du lancement du système. Pour des raisons de compatibilité<sup>19</sup>, ce mécanisme est toujours supporté pour les anciens services ou ceux ne supportant pas *SystemD*. Dans la suite, nous allons donc retrouver les commandes équivalentes à celles présentées dans la section précédente mais pour les services compatibles avec *System V Init*.

#### Obtenir la liste des services

```
$ chkconfig
```

Cette commande liste la liste des services (compatible *System V Init*) présents sur le système et leurs états. C'est ainsi que l'on peut voir que *Webmin* et *Usermin* sont 2 services utilisant toujours *System V Init*.

#### Démarrer / Arrêter un service

```
$ service webmin start
```

Cette commande permet de démarrer le service (ou démon) *Webmin*.

Le second argument pour la commande `service` est toujours le service visé par la commande. Le dernier argument mentionne l'action souhaitée, ainsi on trouve : `start` (démarrer), `stop` (arrête), `restart` (arrête et redémarre), `reload` (recharge la configuration) et `status` (affiche son état).

```
$ service webmin restart
```

Cette commande permet de redémarrer le service (ou démon) *Webmin*.

#### Activer / désactiver un service au démarrage

```
$ chkconfig --level 235 usermin off
```

```
$ chkconfig --level 235 usermin on
```

Cette commande permet de démarrer (`on`) ou arrêter (`off`) le service *usermin* lors du lancement du système et qu'il se trouve dans les niveaux<sup>20</sup> de démarrage 2 (mode texte multiutilisateurs, sans réseau), 3 (mode texte multiutilisateurs, avec réseau) et 5 (mode graphique, multiutilisateurs, avec réseau).

### 5.4.4 Démarrage en mode graphique ou mode texte

Il arrive très souvent sur des serveurs que celui-ci ne démarre qu'en mode texte. En effet, afin de ne pas encombrer la mémoire avec un environnement graphique inutile, on peut choisir de ne pas le démarrer automatiquement lors du lancement du système.

---

<sup>19</sup> CentOS 7 est la première version de la distribution CentOS supportant *SystemD*.

<sup>20</sup> Sans entrer dans les détails, nous considérerons toujours l'utilisation des niveaux 2, 3 et 5 dans les commandes relatives à *System V Init*.

```
$ systemctl isolate multi-user.target
```

Cette commande permet de « passer » dans le mode multi-utilisateur sans interface graphique.

```
$ systemctl isolate graphical.target
```

Cette commande permet de « passer » dans le mode multi-utilisateur avec interface graphique.

```
$ systemctl set-default multi-user.target
```

Cette commande change le mode par défaut de sorte qu'au prochain démarrage de la machine, celle-ci soit en mode multi-utilisateur sans interface graphique.

#### 5.4.5 Lancement d'une commande au démarrage

Dans bien des cas, il est intéressant de pouvoir lancer une commande lors du chargement du système. La solution idéale est, bien sûr, de créer les éléments nécessaires à l'intégration dans *SystemD*. Cependant, il est aussi possible de simplement ajouter la commande à lancer dans le fichier /etc/rc.d/rc.local.

Une fois la commande ajoutée, il faut rendre le fichier /etc/rc.d/rc.local exécutable comme suit :

```
$ chmod +x /etc/rc.d/rc.local
```

Cette opération doit seulement être faite une seule fois. Dès que le fichier rc.local dispose de la permission en exécution, il la garde.

### 5.5 Arrêt du système

Il est possible de provoquer l'arrêt ou le redémarrage du système en utilisant des commandes précises.

Pour arrêter le système et éteindre la machine :

```
$ systemctl poweroff  
$ poweroff  
$ halt -p  
$ shutdown -h +5 "Extinction dans 5 minutes"
```

Pour arrêter le système :

```
$ systemctl halt  
$ halt  
$ shutdown -H +5 "Arret dans 5 minutes"
```

Pour redémarrer le système :

```
$ systemctl reboot  
$ reboot  
$ shutdown -r +5 "Redemarrage dans 5 minutes"
```

## 5.6 Exercices

### 5.6.1 Configuration réseau

1. En utilisant **Network Manager** ou en modifiant les fichiers de configuration, on vous demande de :
  - a. Prendre note des informations réseaux reçues par DHCP
  - b. Configurer statiquement l'adresse IP sur votre machine
  - c. Redémarrer celle-ci
  - d. Vérifier sur votre configuration est fonctionnelle en essayant de surfer sur le web
2. En utilisant **Webmin**, on vous demande :
  - a. D'ajouter une interface réseau virtuelle
  - b. De spécifier l'adresse IP 10.0.1.x (ou x est votre numéro de machine)
  - c. De vérifier, au moyen de la ligne de commande, que cette interface existe

### 5.6.2 Démarrage du système

3. A chaque démarrage, assurez-vous que l'horloge est synchronisée avec le serveur de temps ntp1.oma.be (voir commande et page de manuel de `ntpdate`)
4. Assurez-vous que le service *mariadb* (i.e. évolution du service de base de données MySQL) est bien démarré automatiquement au lancement du système

## Leçon 6 : Configuration du routage

### 6.1 Introduction

Dans cette leçon, nous allons découvrir comment il est possible de **transformer sa machine Linux en routeur**. Cette caractéristique est très intéressante et souvent utilisée. Un système Linux est capable de se substituer à n'importe quel routeur : muni d'interface réseau rapide, il peut servir de véritable firewall pour une entreprise, de passerelle intelligent entre plusieurs réseaux ou encore de véritable routeur implémentant des algorithmes nombreux comme OSPF ou BGP.

Loin de toute cette complexité, nous allons découvrir dans cette leçon comment un serveur Linux peut servir de *passerelle* afin de sortir du réseau.

### 6.2 Rappel : un routeur

Un routeur finalement, **c'est quoi ?** Nous pourrions répondre de manière très complexe à cette question, disons simplement que, dans le cadre de ce cours, un routeur est une machine :

1. Disposant de plusieurs interfaces réseaux interconnectées à des sous-réseaux distincts
2. Configurée pour que le trafic réseau puisse passer d'une interface à l'autre

Ainsi, à la différence d'une machine « normale » (ne jouant pas le rôle de routeur), elle reçoit le trafic lorsqu'elle est destinataire de celui-ci mais jamais elle ne propage des paquets d'une interface réseau vers une autre.

Avant de continuer, il est nécessaire de rappeler l'importance de **la table de routage** présente sur chaque machine. En IPv4, cette table est obtenue comme suit :

```
$ ip route show
default via 192.168.190.2 dev eno16777736 proto static metric 100
10.0.1.0/24 dev eno16777736 proto kernel scope link src 10.0.1.2 metric 100
192.168.190.0/24 dev eno16777736 proto kernel scope link src 192.168.190.50
metric 100
```

En lisant ces informations, nous apprenons que :

- La route par défaut, nommée `default` (ou parfois `0.0.0.0/0`), utilise la passerelle `192.168.190.2` (`via 192.168.190.2`) sur l'interface réseau `eno16777736` (`dev eno16777736`). Donc la route par défaut utilise la passerelle pfSense pour sortir du réseau.
- Le réseau `10.0.1.0/24` est connecté directement (`scope link` – absence d'option `via`) à l'interface réseau `eno16777736` (`dev eno16777736`)
  - Donc un ping vers `10.0.1.5` sera transmis directement sur le réseau sans être adressé au routeur de sortie pfSense
- Le réseau `192.168.190.0/24` est connecté directement (`scope link` – absence d'option `via`) à l'interface réseau `eno16777736` (`dev eno16777736`)
  - Donc un ping vers `192.168.190.1` sera transmis également directement sur le réseau sans être adressé au routeur de sortie pfSense

A l'inverse, un ping vers 8.8.8.8 passera par la route par défaut configurée<sup>21</sup> et l'information sera alors adressée à pfSense pour sortir du réseau virtuel.

### 6.3 Configuration du routeur Linux

Supposons que nous désirons configurer le réseau comme suit :

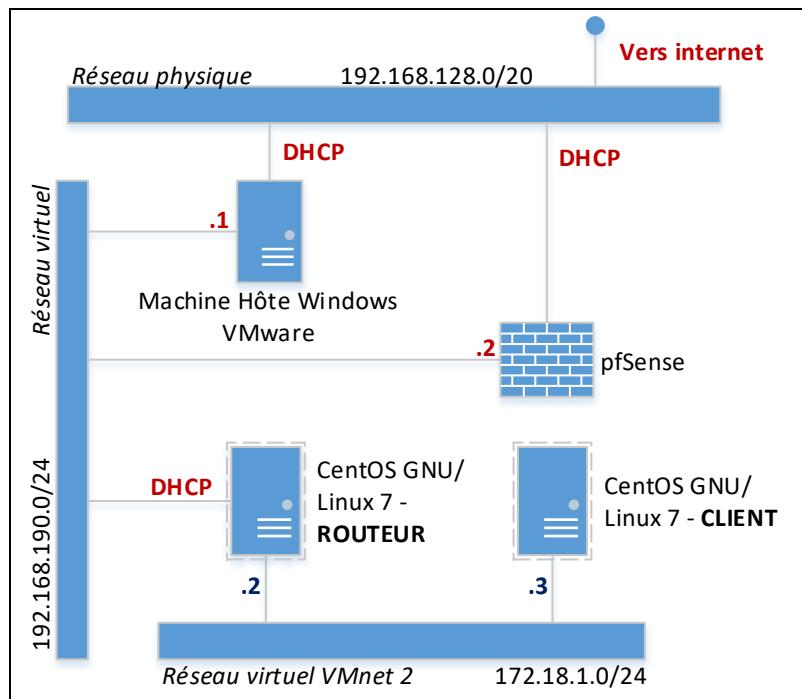


Figure 6.1 : Réseau virtuel souhaité

La figure 6.1 montre le réseau virtuel que nous souhaitons configurer. Ainsi, nous voyons que la machine *CentOS7 Routeur* est connectée à 2 réseaux : l'actuel réseau virtuel et un second (nommé VMnet 2). Nous avons également ajouté une seconde machine CentOS 7, nommée *CentOS 7 Client* qui est uniquement connectée au nouveau réseau virtuel. Pour atteindre internet, cette machine doit donc passer par la machine *CentOS7 Routeur*.

Nous supposerons donc dans la suite de cette leçon<sup>22</sup> que :

1. La machine *Routeur* dispose de 2 interfaces réseaux : `eno1677736` connectée au réseau virtuel 192.168.190.0/24 et `eno33554976` connectée au réseau virtuel 172.18.1.0/24.
2. La machine *Client* dispose d'1 interface réseau : `eno1677736` connectée au réseau virtuel 172.18.1.0/24.
3. Les adresses IP des machines *Routeur* et *Client* sont bien configurées (il faut se référer à la leçon sur la configuration du réseau pour y arriver – le routeur utilise l'adresse 172.18.1.2 et le *client* utilise l'adresse 172.18.1.3)
4. La machine *client* utilise la machine *routeur* comme route par défaut.

<sup>21</sup> Puisque cela ne correspond ni au réseau 192.168.190.0/24 ou 10.0.1.0/24.

<sup>22</sup> Nous reviendrons sur les modifications à apporter dans VMware pour atteindre cette connexion réseau

### 6.3.1 Configurer la machine en mode routeur

Pour configurer la machine *CentOS 7 Routeur* en mode « routeur », il faut ajouter un fichier dans le dossier `/etc/sysctl.d`:

```
$ vim /etc/sysctl.d/10-ipforward.conf
# Enabling IP Forwarding
net.ipv4.ip_forward=1
```

Cette configuration s'assure, **qu'au démarrage du système Linux**, celui-ci autorise les paquets à passer d'une interface à l'autre (`eno1677736 ⇔ eno33554976`).

Il est possible d'activer ce mode directement en entrant :

```
$ echo 1 > /proc/sys/net/ipv4/ip_forward
```

En utilisant cette commande, le mode « routeur » est activé temporairement jusqu'au redémarrage de la machine, à moins que la configuration précédente (ajout du fichier `10-ipforward.conf`) ait été faite également.

#### Problème

Nous remarquons qu'après l'activation du mode « routeur », la machine *Client* ne peut toujours pas atteindre Internet. Cependant, si nous analysons le trafic reçu par la machine *Serveur* nous voyons qu'il est bien reçu et propagé sur l'autre interface. La machine semble donc bien jouer son rôle de routeur.

#### Où se situe le problème alors ?

Le problème vient simplement du fait que le réseau `172.18.1.0/24` n'est pas connu plus loin. Puisque **les paquets sont propagés sans modification**, ils atteignent le réseau de l'école mais **comment la réponse peut-elle parvenir** ?

En effet, ni le réseau de l'école, ni pfSense ne connaît `172.18.1.0/24`. Dès lors, ils sont incapables de transmettre la réponse à notre machine *Client*.

Il y a 2 moyens de solutionner le problème :

- Configurer pfSense pour qu'il connaisse le réseau `172.18.1.0/24` en ajoutant *une route statique* pointant vers la machine *CentOS Routeur*. Ensuite, il faudrait autoriser ces paquets à aller sur internet dans la configuration du firewall.
- Configurer la machine *CentOS 7 Routeur* pour qu'elle translate les adresses (activation du NAT). Si nous adoptons cette solution, la machine *CentOS 7 Routeur* remplacera l'adresse IP source `172.18.1.2` par son adresse IP `192.168.190.x`, rendant ainsi invisible le réseau virtuel 2. C'est cette solution que nous allons configurer.

### 6.3.2 Activation du NAT sur la machine Routeur

Comme indiqué dans le paragraphe précédent, activer le NAT sur la machine *Routeur* est une solution au problème soulevé. Pour ce faire, il faut introduire une toute petite règle dans le firewall *iptables* de la machine *Routeur*.

Nous aurons, lors d'une prochaine leçon, l'opportunité d'étudier en profondeur le fonctionnement d'*iptables*. Nous allons ici simplement discuter de la règle à configurer :

```
$ iptables -t nat -A POSTROUTING -s 172.18.1.0/24 -j MASQUERADE
```

*Cette commande ajoute dans le système NAT une règle indiquant que le trafic dont l'IP source est comprise dans le sous-réseau 172.18.1.0/24 (-s 172.18.1.0/24) doit être translaté (-j MASQUERADE).*

Une fois la règle introduite, la machine *Client* devrait pouvoir accéder à Internet. En cas de problème il convient de vérifier la configuration réseau, conformément aux directives de la leçon précédente.

Afin de rendre cette configuration permanente, il faut sauvegarder la règle de firewall de sorte que celle-ci soit automatiquement ajoutée lors du lancement de la machine *Routeur*. Pour ce faire, il faut simplement entrer :

```
$ service iptables save
```

*Cette commande sauvegarde la configuration actuelle du firewall iptables de sorte que celle-ci soit rechargée automatiquement lors du démarrage de la machine.*

## 6.4 Quelques outils réseaux

Nous allons décrire ici quelques outils réseaux intéressants. Parmi ceux-ci, il y a *nmap*, un outil qui permet notamment de scanner les ports ouverts et *hping*, un outil qui permet de construire des paquets IP. Utiliser de tels outils sur des réseaux étrangers **PEUT ÊTRE CONSIDÉRÉ COMME UN ACTE HOSTILE**. Il faut toujours restreindre l'utilisation de ces outils sur des réseaux que vous gérez et configurez.

### 6.4.1 wireshark

Le programme *wireshark* permet de capturer et analyser le trafic réseau. Il est particulièrement utile pour trouver les problèmes qui surviennent. Dans notre configuration réseau actuelle, lancer *wireshark* sur la machine *CentOS 7 Routeur* permet de capturer tout le trafic venant de la machine *CentOS 7 Client* de manière discrète.

Afin de voir uniquement le trafic qui vous intéresse, vous pouvez définir des filtres. Reportez-vous à la documentation pour la construction de ceux-ci.

Pour le démarrer :

```
$ wireshark
```

### 6.4.2 NMap

Le programme *nmap* est une boîte à outils réseaux. Il peut être utilisé pour identifier le système distant, lister les ports ouverts, ... **Comme annoncé en introduction, cet outil doit être réservé à des réseaux que vous gérez.**

Quelques commandes intéressantes :

```
$ nmap -A -T4 127.0.0.1
```

Cette commande lance l'analyse en mode agressif (-A et -T4). Le résultat donne la liste des ports ouverts sur la machine visée (ici 127.0.0.1), une identification du système d'exploitation et de la version. L'option -A est à déconseiller si l'on veut rester discret. L'option -T détermine le comportement (T0 – Paranoid, T1 – Sneaky, T2 – polite, T3 – normal, T4 – aggressive, T5 – Insane).

```
$ nmap 192.168.190.2
```

Celle commande analyse les ports ouverts (du côté LAN) du firewall pfSense.

```
$ nmap -sP 172.18.1.0/24
```

Cette commande liste toutes les adresses IP actives en utilisant des requêtes ICMP. Attention, beaucoup de firewall bloquent ces requêtes.

```
$ nmap -PS80 172.18.1.0/24
```

Cette commande liste toutes les adresses IP actives en utilisant des demandes de connexion sur le port 80. Nous obtiendrons ainsi la liste des machines exécutant un serveur web.

Il existe également une version graphique de nmap :

```
$ nmapfe
```

#### 6.4.3 Telnet

Telnet est un outil simple : il permet d'ouvrir une connexion TCP sur n'importe quel port et puis interagir avec le serveur facilement. Par exemple :

```
$ telnet mail.helmo.be 110
+OK Dovecot ready.
```

Cette commande permet d'ouvrir une connexion TCP vers le serveur POP3 (port 110) de HELMo. Une fois l'invite affiché (+OK Dovecot ready), il est possible d'introduire des commandes POP3 pour obtenir des réponses (USER/PASS/LIST/RETR/DELETE/QUIT).

Cette méthode permet souvent de connaître le type de serveur installé (ici Dovecot) et parfois la version déployée. Ces informations sont nécessaires pour déterminer si des vulnérabilités sont présentes dans les logiciels installés.

```
$ telnet localhost 22
SSH-2.0-OpenSSH_6.6.1
```

Ici nous remarquons que le service SSH utilise la version OpenSSH 6.6.1. On peut faire de même avec le service FTP :

```
$ telnet ftp.belnet.be 21
220 ProFTPD 1.3.4a Server (Belnet FTP Server) [193.190.67.98]
```

Il faut donc être prudent lorsqu'on configure un service réseau : il faut s'assurer qu'il ne diffuse pas d'information d'identification le concernant.

#### 6.4.4 HPing

HPing est une autre boîte à outils réseau. Grâce à hping, il est possible de construire des paquets TCP de toute pièce.

*hping* permet notamment de lancer *une commande ping* en utilisant TCP/IP plutôt que ICMP (qui est souvent bloqué par les firewalls).

```
$ hping -S 192.168.190.2 -p 80
HPING 192.168.190.2 (eno16777736 192.168.190.2): S set, 40 headers + 0 data bytes
len=46 ip=192.168.190.2 ttl=64 DF id=31862 sport=80 flags=SA seq=0 win=65228
rtt=0.4 ms
```

Cette commande envoie des paquets TCP sur le port 80 (-p 80). Si la réponse est flags=SA, c'est que le port en question est ouvert. Dans le cas contraire, la réponse flags=RA est rapportée.

Il est également possible d'utiliser *hping* pour scanner les ports ouverts d'une machine (-p ++80).

Il est même possible d'utiliser *hping* pour échanger des fichiers entre deux machines de manière très discrète, par exemple en utilisant des paquets ICMP :

Emetteur	Destinataire
\$ hping 172.18.1.2 --icmp -d 100 --sign monfichier --file /etc/passwd	\$ hping --listen 172.18.1.2 -I eno33554976 --sign monfichier --icmp

Il est également possible d'effectuer un *traceroute* en utilisant *hping*. L'intérêt étant de pouvoir utiliser **n'importe quel paquet TCP** (au lieu des traditionnels paquets UDP ou ICMP).

Par exemple, avec la commande :

```
$ hping -z -t 1 -S www.swila.be -p 80
```

Cette commande permet de lancer des connexions sur le port TCP 80 avec des TTLs croissants en commençant à 1 (-t 1) vers la destination www.swila.be (option -S). Chaque fois que l'utilisateur appuie sur CTRL+Z, hping incrémente le TTL (option -z) et tente de résoudre l'adresse IP obtenue.

L'outil *hping* a encore bien d'autres possibilités comme lancer des attaques DoS vers un serveur (via l'option --flood). Je vous renvoie aux pages de manuels pour plus de détails.

#### 6.5 Exercices

On vous demande :

1. Eteindre votre machine *Linux CentOS 7*.
  - Modifier ses paramètres (*Edit virtual machine settings*) pour lui ajouter une nouvelle interface réseau (onglet *hardware*, Cliquer sur *Add* puis *Network Adapter*)
  - Choisir *Network Connection Custom => VMnet2* puis *Finish*
  - Onglet *Options* > Changer le *virtual machine name* (à droite) en *CentOS7-Routeur*
  - Démarrer la machine modifiée. La nouvelle interface réseau devrait pouvoir être configurée.

2. Ajouter une nouvelle machine CentOS 7 :
  - Décompresser le fichier contenant la machine CentOS **dans un nouveau dossier** (attention de ne pas écraser votre machine actuelle)
  - Ajouter cette machine dans VMware
  - Modifier ses paramètres (Edit virtual machine settings)
    - i. Pour Network Adapter, choisir Network Connection *Custom > VMnet2*
    - ii. Onglet Options > Changer le *Virtual machine name* en *CentOS7-Client*
  - Démarrer la nouvelle machine virtuelle – répondez *I copied it* à la question posée par VMware.
3. Configurer votre machine *CentOS 7 – Routeur* en mode « routeur ». L'adresse IP à configurer sur la 2<sup>ème</sup> interface réseau est 192.168.131.2
4. Configurer votre machine *CentOS 7 – Client*. L'adresse IP à configurer est 192.168.131.15. Cette machine doit utiliser la machine *routeur* comme passerelle. N'oubliez pas de mentionner un serveur DNS.

Essayez de surfer à partir de la seconde machine. Utilisez *wireshark* pour capturer le trafic qui transite par la machine *routeur*.
5. A l'aide de *nmap*, lister tous les ports ouverts sur la machine *CentOS – Routeur* et *CentOS – Client*.
6. A l'aide de *hping*, simulez un *traceroute* sur le port 80 vers [www.yahoo.fr](http://www.yahoo.fr)
7. A l'aide de *nmap*, Vérifiez les ports ouverts sur la machine de votre voisin
8. Utiliser *hping* pour échanger un fichier texte entre vos 2 machines.

## Leçon 7 : Le service SAMBA

### 7.1 Introduction

Le service *samba* est un service intéressant dans les environnements mixtes mêlant à la fois des postes Windows et Linux. En effet, *samba* permet à un poste Linux de **partager des ressources avec les ordinateurs Windows.**

Ainsi, grâce à *samba* il est possible de créer des partages disques et imprimantes qui peuvent être connectés comme ressources réseaux depuis les postes Windows. Cette particularité est parfois très intéressante car on peut facilement partager des informations et profiter des avantages des deux systèmes en même temps.

Pour y arriver, il faut configurer le service *samba* dont le fichier de configuration se trouve dans `/etc/samba/smb.conf`.

Les versions récentes de *samba* permettent d'intégrer une machine linux comme **membre d'un domaine Active Directory Microsoft**. Cependant, dans le cadre de cette leçon, nous étudierons surtout la possibilité de mettre à disposition des ressources disques et imprimantes sans faire une intégration à Active Directory.

### 7.2 La configuration de samba

Comme annoncé dans l'introduction, la configuration du service *samba* est concentrée dans le fichier texte `/etc/samba/smb.conf`. Ce fichier **est composé de plusieurs sections**. Une section est identifiée par un nom entre crochets. Ainsi, les sections suivantes sont présentes :

Section	Utilisation
<code>[global]</code>	Cette section contient la configuration <i>globale</i> du serveur (c'est-à-dire indépendamment de la ressource partagée)
<code>[homes]</code>	Cette section décrit le partage des dossiers personnels des utilisateurs. Par défaut, lorsqu'un utilisateur se connecte au serveur via <i>samba</i> , il a accès automatiquement à son dossier personnel <code>/home/login</code>
<code>[printers]</code>	Cette section décrit le partage des imprimantes installées sur le serveur. Par défaut, toutes les imprimantes configurées sur le serveur sont partagées. Dans cette leçon, nous n'étudierons pas le partage des imprimantes.

#### 7.2.1 La sections globales

Nous allons passer en revue les principales options de ce fichier en commençant par la section `[global]`.

Option	Utilisation
<code>workgroup</code>	Cette option permet de définir le groupe de travail dans lequel ce serveur appartient. Cette option facilite la navigation du réseau depuis les postes Windows puisque, par défaut, les postes d'un même groupe de travail se voient mutuellement.
<code>server string</code>	Cette option permet d'afficher une description affichée par les postes Windows
<code>netbios name</code>	Cette option permet de spécifier le nom sous lequel le serveur apparaît. Par défaut, c'est le nom de la machine qui est utilisé. Il est possible d'en spécifier un autre ici.

<b>interfaces hosts allow</b>	<p>L'option <code>interfaces</code> permet de déterminer sur quelles interfaces le serveur samba écoutera. Par exemple, mentionner uniquement <code>lo</code> signifierait que le serveur écouterait uniquement sur l'interface <code>localhost</code> et ne serait pas accessible à distance.</p> <p>L'option <code>hosts allow</code> permet de déterminer quelles IP sont autorisées à se connecter sur le serveur <i>samba</i>.</p> <p>Par défaut, toutes les interfaces sont actives et aucune restriction d'adresse n'est présente.</p>
<b>log file</b>	Renseigne le chemin vers les fichiers journaux. Ces fichiers sont utiles en cas de problème.
<b>security</b>	<p>Cette option est une des plus importantes du serveur <i>samba</i> puisqu'elle indique comment le serveur va réaliser l'authentification. Les paramètres possibles sont, aujourd'hui, <code>user</code>, <code>domain</code> ou <code>ads</code>.</p> <p>Par défaut, l'option <code>user</code> est configurée et indique que l'utilisateur doit être connu (et donc authentifié) <b>avant</b> de pouvoir accéder aux ressources partagées.</p> <p>Les options <code>domain</code> ou <code>ads</code> sont à utiliser quand on fonctionne en mode domaine Windows. La distinction est que <code>domain</code> est un mode de fonctionnement compatible avec les anciennes versions de Windows (Windows NT 4) alors que <code>ads</code> permet au serveur <i>samba</i> d'intégrer un domaine Active Directory (&gt;= Windows 2000 Server).</p>
<b>passdb backend</b>	Cette option mentionne comment les mots de passe sont sauvegardés sur le serveur. L'option <code>tdbsam</code> configurée par défaut est adaptée à nos besoins.
<b>load printers</b>	Cette option indique si la gestion des imprimantes doit être activée ou non.

Il y a encore bien d'autres options présentes dans le fichier `smb.conf`. Pour plus d'information, je vous renvoie à la page de manuel de `smb.conf`.

### 7.2.2 Les partages

Configurer un partage sur un serveur *samba* est simple, il suffit de créer une section. Ainsi, les sections `[homes]` et `[printers]` sont des déclarations de partage. Nous pourrions, si nous le souhaitons créer un nouveau partage simplement en créant une section entre crochets. Par exemple `[info]` créerait une nouvelle section et un nouveau partage nommé `info`.

Les options que nous allons décrire maintenant s'appliquent à tous les partages.

Option	Description
<code>comment</code>	Décrit le partage concerné, optionnel
<code>browsable = yes no</code>	Indique si le partage doit être affiché dans l'explorateur réseau de Windows.
<code>writeable = yes no</code> <code>writable = yes no</code> <code>read only = yes no</code>	Indique si le partage est en lecture seule ou en lecture / écriture
<code>read list</code>	L'option <code>read only</code> est l'inverse de <code>writeable</code> .
	Indique quels utilisateurs / groupes ont accès en lecture seule au partage. Il est possible de mentionner un groupe d'utilisateurs en

	le préfixant par @ ou +.
<b>write list</b>	Indique quels utilisateurs / groupes ont un accès en lecture / écriture au partage. Il est possible de mentionner un groupe d'utilisateurs en le préfixant par @ ou +.
<b>create mask</b>	Indique avec quelle permission UNIX <u>les fichiers</u> sont créés. En effet, quand un fichier est copié depuis un poste Windows, <i>samba</i> ne peut pas déterminer la permission UNIX qui lui sera attachée. Cette option permet de la préciser.  Par défaut : <code>create mask = 0744</code>
<b>directory mask</b>	Indique avec quelle permission UNIX <u>les dossiers</u> sont créés. En effet, quand un dossier est créé ou copié depuis un poste Windows, <i>samba</i> ne peut pas déterminer quelle permission UNIX il doit lui associer.  Par défaut : <code>directory mask = 0755</code>
<b>inherit acls = yes no</b>	Cette option assure que les ACLs sont propagées sur les partages <i>samba</i> comme sur les fichiers et dossiers Linux. Ainsi, si des ACLs par défaut sont définies, les fichiers et dossiers copiés ou créés depuis les postes Windows hériteront de celles-ci. C'est une façon simple de gérer les droits souhaités.  Par défaut : <code>inherit acls = no</code>
<b>printable = yes no</b>	Cette option indique si le partage en cours de configuration est un partage imprimante ou non. S'il s'agit d'un partage imprimante, il est obligatoire de préciser cette option à yes.
<b>path</b>	Cette option <b>obligatoire</b> mentionne le chemin vers le dossier qui est partagé (dans le cas d'un partage disque) ou vers la file d'attente d'impression (partage imprimante).

Il y a, ici aussi, bien d'autres options présentes pour les partages de ressource dans le fichier `smb.conf`. Pour plus d'information, je vous renvoie à la page de manuel de `smb.conf`.

### 7.3 Vérification des permissions

A l'instar des partages sous Windows, pour que l'accès à un partage soit autorisé sous Linux, il est nécessaire que :

1. Les **autorisations du partage**, matérialisés par les `read list` et `write list`, mentionnent les utilisateurs qui doivent bénéficier d'un accès
2. Les **permissions UNIX** (traditionnelles ou ACLs si l'option `inherit acls` est active) permettent l'accès au dossier pour l'utilisateur en question.

C'est après **cette double vérification seulement** que l'accès sera autorisé. Je vous encourage à **rester simple** dans la gestion des permissions. Ainsi, privilégiez toujours les permissions UNIX quand c'est possible et rester large dans les autorisations.

## 7.4 La gestion des utilisateurs samba

Aussi étrange que cela puisse paraître, les utilisateurs *samba* ne sont pas nécessairement tous les utilisateurs du système. Ainsi, il y a une base de données des utilisateurs samba distincte des utilisateurs du système.

Cependant, il est **ABSOLUMENT NECESSAIRE** qu'un utilisateur *samba* existe également sur le serveur Linux (alors que l'inverse n'est pas obligatoire : on peut avoir des utilisateurs Linux qui ne sont pas renseignés dans *samba*).

### 7.4.1 Ajout/suppression d'un utilisateur samba

La commande `smbpasswd` permet de gérer les utilisateurs *samba*. Pour rappel, un compte UNIX doit exister pour chaque utilisateur *samba*.

```
$ smbpasswd -a swila
```

*Cette commande permet d'ajouter l'utilisateur swila à la base de données des utilisateurs samba.*

*Cette commande est interactive car le mot de passe pour cet utilisateur est demandé lors de l'ajout.*

```
$ smbpasswd -x swila
```

*Cette commande supprime l'utilisateur swila de la base de données des utilisateurs samba.*

D'autres options sont possibles comme `-d` qui permet de désactiver un utilisateur donné ou `-e` qui permet de réactiver l'utilisateur désactivé.

```
$ pdbedit -L
```

*Cette commande permet de lister tous les utilisateurs présents dans la base de données des utilisateurs samba.*

## 7.5 Démarrer le service samba

Pour démarrer et/ou activer le service *samba*, il faut utiliser la commande `systemctl` :

```
$ systemctl start smb  
$ systemctl start nmb
```

Pour activer *samba* au démarrage, il faut remplacer `start` par `enable` pour les 2 services en question.

Pour accéder au partage Linux depuis Windows, il faut se référer à la leçon sur les partages dans le cours d'administration Windows et utiliser la commande Windows `net use`.

## 7.6 Accéder à un partage Windows depuis Linux

Il est également possible, depuis la machine Linux, de profiter des ressources misent à disposition par un serveur Windows. Ainsi, dans ce cas de figure, le poste Linux est client et le poste Windows est serveur.

Bien sûr, il est possible d'utiliser les commandes suivantes entre 2 machines Linux, pour autant que l'une d'elles ait activé le partage *samba* et joue le rôle de serveur.

### 7.6.1 Lister les partages d'un serveur

A l'instar de la commande Windows `net view`, il est possible, à partir de Linux, de lister les partages d'une machine Microsoft comme suit :

```
$ smbclient -L \\NomOuIPduServer -U nomUtilisateurWindows
```

## 7.6.2 Monter / Démonter une ressource disque

A l'instar de ce que propose net use pour assigner une lettre à un partage disque distant, il est possible, depuis Linux, de monter un système de fichier Windows distant. Pour ce faire, il faut utiliser la commande `mount`. Exemple :

```
$ mount -t cifs //192.168.190.1/Backup /media -o username=lsw,vers=3.0
```

Cette commande monte le partage `\|192.168.190.1\Backup` dans le dossier `/media` de la machine Linux en utilisant le compte `lsw` et la version 3.0 du protocole CIFS<sup>23</sup>.

```
$ mount -t cifs //192.168.128.3/Users /media -o username=p010544, domain=CG, vers=3.0
```

Cette commande monte le partage `\|192.168.128.3\Users` dans le dossier `/media` de la machine Linux en utilisant le compte `p010544` sur le domaine `CG`<sup>24</sup> et la version 3.0 du protocole CIFS

```
$ umount /media
```

Cette commande démonte le partage. La ressource est donc libérée et le dossier `/média` n'est plus un lien vers le partage distant.

## 7.7 Exercices

On vous demande de :

1. Monter le partage `public` du serveur DATA (192.168.128.3) sur votre machine Linux
2. Configurer le service `samba` pour :
  - a. Créer un partage `documents` (placé dans `/home/documents`) accessible en lecture/écriture aux utilisateurs suivants : `swila, bm1, bm2, bm3`
  - b. Permettre aux utilisateurs d'accéder à leur dossier personnel
  - c. Partager le dossier `/home/group_biomed`<sup>25</sup> de sorte que seuls les membres du groupe `biomed3` puissent y accéder en lecture uniquement.
  - d. Créer un partage public (placé dans `/home/public`) accessible en lecture/écriture à tous les utilisateurs. Chacun doit pouvoir supprimer les fichiers / dossiers déposés par d'autres. Pour ce faire, vous réaliserez 2 versions de cette configuration : la première faisant appel aux autorisations de partage et la seconde en utilisant les droits UNIX).
3. Créer un script PERL permettant d'ajouter les utilisateurs présents dans le fichier CSV dans la base de données des utilisateurs de `samba`

Vérifiez votre configuration en utilisant le poste Windows pour accéder à votre serveur `samba` (via l'adresse IP `\|192.168.190.x`).

<sup>23</sup> La mention de la version de CIFS semble nécessaire lorsqu'on s'adresse à un serveur Windows récent (Windows 8 ou Server 2012) pour éviter une erreur de type *Input/Output error*.

<sup>24</sup> Cette option est obligatoire s'il s'agit d'un domaine Active Directory (voir partie Windows)

<sup>25</sup> Ce dossier a été créé dans une leçon précédente

## Leçon 8 : Le serveur Web Apache

### 8.1 Introduction

Le serveur web *apache* est probablement un des serveurs web les plus déployés sur Internet. Il est présent sur toutes les plateformes (Windows / Unix / Mac OS X) et est un grand classique. Ainsi, les packages LAMP, XAMP ou WAMP utilisent abondamment Apache.

La popularité de PHP a également beaucoup aidé le déploiement d'*apache*. En effet, c'est une des plateformes les plus stables pour installer des sites web PHP. Il faut cependant être honnête, d'autre serveur web, plus léger qu'*apache* ont vu le jour ces dernières années. C'est ainsi que *nginx*, *lighttpd* et bien d'autres viennent compléter le podium.

Enfin, n'oublions pas qu'actuellement, le déploiement d'application .NET nécessite l'utilisation du serveur Web IIS concurrent, proposé par Microsoft.

Dans cette leçon nous allons découvrir la configuration de base d'un serveur web Apache, la possibilité de sécurisation d'un site au moyen d'une authentification HTTP de base et finalement, les éléments à mettre en place pour activer SSL.

### 8.2 Configuration de Apache

La configuration du service *apache* est localisée dans le fichier `/etc/httpd/conf/httpd.conf` et dans le dossier `/etc/httpd/conf.d/`.

#### 8.2.1 Fichier httpd.conf

Dans le fichier `/etc/httpd/conf/httpd.conf`, nous allons trouver les options *globales* du serveur *apache*.

Option	Explication
<code>ServerRoot</code>	Mentionne le chemin vers la configuration du serveur <i>apache</i>
<code>Listen</code>	Mentionne le port d'écoute utilisé. Le fichier de configuration propose le port 80. Il est possible de spécifier une adresse IP en plus du port pour <i>limiter</i> l'écoute du serveur uniquement sur cette IP.
<code>Include conf.modules.d/*.conf</code>	Cette directive va lire tous les fichiers portant l'extension <code>.conf</code> et les importer dans la configuration. Le dossier <code>conf.modules.d</code> contient principalement des fichiers comprenant les directives <code>LoadModule</code> .  Ces directives permettent d'activer des extensions au niveau du serveur <i>apache</i> . Ainsi, si le langage PHP est installé, une directive <code>LoadModule</code> est nécessaire pour charger la librairie en question. Cette directive se trouve dans le fichier <code>/etc/httpd/conf.modules.d/10-php.conf</code>
<code>User apache</code> <code>Group apache</code>	Cette option détermine l'utilisateur et le groupe avec lequel le service <i>apache</i> est démarré. Cela implique que le service <i>apache</i> devra accéder aux fichiers HTML et dossiers web avec cet utilisateur. Il convient donc de fixer les droits correctement.

<b>ServerAdmin</b>	Mentionne l'adresse mail de l'administrateur du serveur. Cette adresse peut être présente sur les pages d'erreur.
<b>ServerName</b>	Mentionne le nom DNS du serveur <i>apache</i> . Si aucun service DNS n'est relié à ce serveur web, il est nécessaire de mentionner l'adresse IP du serveur.
<b>DocumentRoot</b>	Indique l'emplacement principal des pages web. Ainsi, les pages web sont placées dans le dossier <code>/var/www/html</code> .
<b>&lt;Directory ...&gt;</b> ... <b>&lt;/Directory&gt;</b>	<p>L'option <code>Directory</code> permet de préciser des options, autorisations ou restrictions d'accès au dossier mentionné.</p> <p>On trouve, très souvent, la directive <code>Options</code> qui permet, notamment de <i>lister le contenu du dossier</i> (via <code>Indexes</code>) ou encore de <i>suivre les liens symboliques</i> (via <code>FollowSymLinks</code>).</p> <p>La directive <code>AllowOverride</code> permet d'indiquer si les fichiers cachés <code>.htaccess</code> qui peuvent être présent dans les dossiers web doivent être pris en compte ou pas. Les fichiers <code>.htaccess</code> permettent de modifier la configuration locale du serveur et d'activer une authentification. Pour activer la prise en charge complète, il faut mentionner <code>AllowOverride All</code>.</p> <p>La directive <code>Require</code> qui permet de protéger l'accès à un dossier. Nous privilégierons l'utilisation des fichiers <code>.htaccess</code> pour arriver à cette configuration.</p>
<b>AddDefaultCharset</b>	Permet de mentionner le codage de caractère par défaut utilisé par le serveur Web.

Il existe encore de nombreuses options qu'il est possible de définir dans ce fichier. Je vous renvoie à la page de manuel pour une présentation plus complète.

### 8.2.2 Le dossier conf.d

Le dossier `/etc/httpd/conf.d/` permet de configurer les sites web souhaités. En effet, tous les fichiers présents se terminant par `.conf` sont pris en charge par *apache* comme élément de configuration.

Une des options importantes est `VirtualHost`. Grâce à cette option, il est possible d'héberger plusieurs sites web sur le même serveur *apache*.

Ainsi, pour créer le site web par défaut du serveur, nous pourrions créer un fichier `/etc/httpd/conf.d/default-site.conf` (le nom doit juste se terminer par `.conf`). A titre d'exemple, voici la configuration pour celui de HELMo :

```
<VirtualHost 192.168.3.206:80 [2001:6a8:2cc0:8000::206]:80>
    ServerName      project.helmo.be
    ServerAlias     webmail.helmo.be
    DocumentRoot   /var/www/html/default
    <Directory "/var/www/html/default">
        AllowOverride All
        Options FollowSymLinks
        Require all granted
```

```
</Directory>

ErrorLog logs/error_log
CustomLog logs/access_log combined
</VirtualHost>
```

Dans cet exemple, nous déclarons *un hôte virtuel* attaché à l'adresse IPv4 192.168.3.206 et l'adresse IPv6 2001:6a8:2cc0:8000::206 sur le port TCP 80.

Le nom DNS associé au site web (via `ServerName`) est `project.helmo.be`. C'est de cette manière qu'il est possible de définir plusieurs site web sur le même serveur, en utilisant des noms DNS différents et donc, des `ServerName` distincts. La directive `ServerAlias` permet de mentionner d'autres noms pour le même site comme `webmail.helmo.be` dans cet exemple.

La directive `DocumentRoot` mentionne l'emplacement des fichiers HTML et/ou PHP de ce site.

Comme déjà abordé dans les options globale d'Apache, la directive `Directory` permet d'adapter, modifier, restreindre l'accès au dossier précisé (ici `/var/www/html/default`). A l'intérieur de cette directive, nous avons l'option `AllowOverride All` qui permet de prendre en compte les fichiers `.htaccess` qui se trouveraient avec les pages HTML (dans `/var/www/html/default` donc), la mention `FollowSymLinks` qui permet de suivre les liens symboliques s'ils sont présents et finalement, l'option `Require` qui permet à tout le monde d'accéder à ce site.

Enfin, les directives `ErrorLog` et `CustomLog` contrôlent les informations placées dans les fichiers journaux *d'apache*. Ils sont situés dans `/var/log/httpd`. Le premier indique les erreurs rencontrées (ce fichier est une mine d'or lorsqu'on développe son application web et que celle-ci ne fonctionne pas correctement) tandis que le second reprend la liste des accès au site web (qui est intéressant pour les statistiques).

### 8.2.3 Activation de SSL

La manière la plus simple d'activer SSL sur un site web *apache* est de modifier le fichier `/etc/httpd/conf.d/ssl.conf`. Avant de commencer, il est nécessaire de disposer du certificat SSL associé au nom DNS du site dans le dossier `/etc/pki/tls/certs` (ainsi que les éventuels certificats intermédiaires) et de la clé privée correspondante dans `/etc/pki/tls/private`.

Si l'on souhaite activer SSL pour le site web par défaut, on peut se contenter de modifier le `VirtualHost _default_:443` présent dans le fichier de configuration. On retrouve, sans surprise, les directives de configuration que nous venons de voir pour `VirtualHost`. Ainsi, on peut utiliser `DocumentRoot`, `ServerName`, `Directory`, `ErrorLog` et `CustomLog` comme ci-dessus.

A ces options, nous avons des directives propres à SSL.

Option	Utilisation
<code>SSLEngine</code>	Active le support SSL pour le site
<code>SSLProtocol</code>	Détermine les protocoles supportés. Il convient d'adapter la valeur par défaut afin de désactiver des protocoles complètement obsolètes : <code>SSLProtocol all -SSLv2 -SSLv3</code>

<b>SSLCipherSuite<sup>26</sup></b>	Détermine les algorithmes de chiffrement qui peuvent être utilisés. Ici aussi, il vaut mieux modifier l'option par défaut : SSLCipherSuite "EECDH+ECDSA+AESGCM EECDH+aRSA+AESGCM EECDH+ECDSA+SHA384 EECDH+ECDSA+SHA256 EECDH+aRSA+SHA384 EECDH+aRSA+SHA256 EECDH+AESGCM EECDH EDH+AESGCM EDH+aRSA HIGH !MEDIUM !LOW !aNULL !eNULL !LOW !RC4 !MD5 !EXP !PSK !SRP !DSS"
<b>SSLHonorCipherOrder</b>	Cette option détermine s'il faut respecter l'ordre dans lequel les algorithmes sont précisés. Il vaut mieux activer cette option : SSLHonorCipherOrder on
<b>SSLCertificateFile</b>	Mentionne le chemin vers le certificat du serveur. Le nom inclut dans le certificat doit correspondre au nom DNS du serveur pour éviter une erreur de connexion depuis le navigateur. SSLCertificateFile /etc/pki/tls/certs/swilabus.be.crt
<b>SSLCertificateKeyFile</b>	Mentionne le chemin vers la clé privée associée à ce certificat. SSLCertificateKeyFile /etc/pki/tls/private/swilabus.be.key
<b>SSLCertificateChainFile</b>	Mentionne éventuellement le chemin vers les certificats intermédiaires fournis par l'autorité de certification. SSLCertificateChainFile /etc/pki/tls/certs/alphassl.crt

### 8.3 Sécurisation d'un site web avec une authentification simple

Il est facile d'activer une authentification simple avec *apache* sur un site ou une partie d'un site web existant. Pour ce faire, il faut :

1. Activer la prise en charge des fichiers .htaccess sur le site en question
2. Créer un fichier .htaccess à la racine du dossier à protéger avec un contenu déterminé
3. Créer le fichier des utilisateurs spécifiant les logins et mots de passe autorisés.

L'activation de la prise en charge des fichiers .htaccess a déjà été abordée précédemment avec l'option AllowOverride. Ainsi, si nous désirons protéger toutes les pages se trouvant dans le dossier /admin, par exemple, il faut créer un fichier .htaccess dans le dossier admin du site web. Ce fichier devra contenir les éléments suivants :

```
AuthType Basic
AuthName "Message affiché par le navigateur"
AuthBasicProvider file
AuthUserFile "/var/www/admin.passwd"
Require valid-user
```

Il faut encore créer le fichier admin.passwd mentionné dans le fichier .htaccess. Ce fichier peut être créé à l'aide de la commande htpasswd fournie par Apache :

```
$ htpasswd -B -c /var/www/admin.passwd admin
```

Cette commande crée un nouveau fichier (-c) de mot de passe nommé admin.passwd en chiffrant le mot de passe de manière sûr (-B) et ajoute le login admin. Le mot de passe sera demandé pour compléter l'ajout de l'utilisateur.

```
$ htpasswd -B /var/www/admin.passwd godswila
```

---

<sup>26</sup> Cette liste est extraite de nombreuses sources comme celle-ci ( ! elle est souvent mise à jour !):  
[https://raymii.org/s/tutorials/Strong\\_SSL\\_Security\\_On\\_nginx.html](https://raymii.org/s/tutorials/Strong_SSL_Security_On_nginx.html)

Cette commande **ajoute** au fichier mentionné `admin.passwd` le login `godswila` en utilisant un chiffrement sûr (-B). Le mot de passe sera demandé pour compléter l'ajout de l'utilisateur.

A l'inverse du service `samba` qui nécessitait qu'un compte UNIX du même nom existe, avec `htpasswd`, il est possible d'ajouter des logins n'existant pas sur le système.

**Attention !** Les mots de passe circulent en clair entre le navigateur et le serveur. Sans SSL, cette option ne permet pas réellement de sécuriser un site puisqu'un espion peut facilement intercepter le mot de passe entré.

## 8.4 Site web personnel des utilisateurs

Une autre options intéressantes d'`apache` est de permettre, facilement, à des utilisateurs du système de disposer d'un site web personnel. Cette option est déployée à HELMo pour permettre aux étudiants d'avoir leurs sites web accessibles via `http://192.168.128.13/~login`.

Cette directive s'appelle `UserDir` et est configurable dans le fichier `/etc/httpd/conf.d/userdir.conf`. Ainsi à HELMo, la directive suivante est activée :  
`UserDir public_html`

Cette directive mentionne que tous les utilisateurs peuvent créer un site web personnel. Ce site doit être hébergé dans le dossier `public_html` placé dans leur espace personnel (et donc dans le dossier suivant `/home/login/public_html`).

**Attention !** Comme nous l'avons vu, le service `apache` est lancé par un utilisateur particulier, l'utilisateur `apache`. Il est **absolument nécessaire que l'utilisateur apache puisse atteindre le dossier public\_html et lire tous les fichiers** qui s'y trouvent. Dans le cas contraire, une erreur 403 pourrait être retournée par le serveur.

## 8.5 Exercices

On vous demande de configurer `apache` :

1. Et placer une page web de votre création sur le site web par défaut
2. Et autoriser les utilisateurs à déployer leurs sites web personnels. Les fichiers seront placés dans un dossier `web` présent dans leur répertoire personnel.
3. Et créer un dossier `admin` dans le site web par défaut. Ajoutez-y une page web particulière. Protégez le site web au moyen d'une authentification simple et ajouter l'utilisateur `letmesee` et le mot de passe `yesICanREAD`. Utilisez Wireshark pour capturer le mot de passe.  
**!! Vérifier votre configuration précisément !!**
4. Créer un script PERL permettant d'ajouter les utilisateurs dans le fichier `htpasswd` protégeant votre site web `admin`.
5. Configurer un site SSL `intranet.swilabus.be` en récupérant le certificat mis à votre disposition sur la page accompagnant la leçon. Vérifiez que la connexion sécurisée s'établit correctement en utilisant un navigateur depuis votre machine Client pour y accéder.

## Leçon 9 : Le service DNS

« Celui qui contrôle le DNS, contrôle le monde »

### 9.1 Introduction

Le service DNS est un des services les plus critiques et les plus utilisés sur Internet. En effet, c'est grâce à ce service qu'il est possible de *traduire un nom en une adresse IP*. Sans ce service, nous serions obligés de mémoriser des centaines d'adresses IP, ce qui, avouons-le, ne serait guère pratique (ni en IPv4 et encore moins en IPv6).

Ce service est également impliqué dans la livraison du courrier électronique en répondant à la question suivante : « *A qui adresser les mails à destination du nom de domaine @helmo.be ?* ». La réponse à cette question est donnée dans un enregistrement particulier du DNS, indiquant le serveur à contacter pour les mails.

Comme nous le voyons dans cette introduction, le DNS est important au bon fonctionnement d'Internet.

### 9.2 Rappels sur le fonctionnement du DNS

Comme annoncé dans l'introduction, le service DNS est responsable de *la traduction* d'un nom DNS en adresse IP (et inversement, d'une adresse IP vers un nom DNS). Il faut aussi rappeler qu'il s'agit d'un système hiérarchique dans lequel un domaine est responsable des sous-domaines qu'il contient. Ainsi, `helmo.be` est responsable des sous-domaines `student.helmo.be`, `cg.helmo.be` ou `co.helmo.be`.

Un serveur DNS peut *déléguer* la gestion d'un sous-domaine à un autre serveur. Ainsi, le serveur DNS `.be` a délégué la gestion de `helmo.be` au serveur `193.190.64.113`. Pour des raisons de sécurité et de performance, il est très courant de disposer de *serveurs DNS secondaires* complètement et automatiquement synchronisé avec le serveur DNS principal.

#### 9.2.1 Un peu de vocabulaire

Un **domaine** est un nom DNS complètement qualifié, qui peut être *local* (interne à une entreprise) ou *officiel* (acheté auprès d'un registrar). Ainsi, par convention, les *domaines locaux* portent le suffixe `.local` ou `.localdomain`.

Un **serveur DNS** est une machine exécutant le service `bind`<sup>27</sup> qui permet de répondre aux requêtes DNS.

Une **zone DNS** reprend *la liste de tous les enregistrements* d'un domaine. Elle peut être *directe* (transformation d'un nom en adresse IP) ou *inverse* (transformation d'une adresse IP en nom). Ainsi, la zone DNS directe attachée au domaine `helmo.be` contient des enregistrements pour les noms `www.helmo.be`, `elearning.helmo.be` ou encore `webmail.helmo.be`.

<sup>27</sup> Par exemple – il existe d'autres serveurs DNS mais `bind` est celui distribué en standard dans CentOS 7

Un **enregistrement** est une entrée dans une zone DNS associée à un domaine. On trouve des enregistrements de plusieurs types :

Type d'enregistrement	Explication
A	Permet de renseigner l'adresse IPv4 associée à un nom
AAAA	Permet de renseigner l'adresse IPv6 associée à un nom
CNAME	Permet de faire pointer un nom vers un enregistrement de type A, AAAA ou PTR. Il n'est pas autorisé de faire pointer un CNAME vers un autre CNAME.
MX	Permet de renseigner les noms des serveurs mails à contacter pour le domaine ou le sous-domaine
NS	Permet de renseigner les noms des serveurs DNS en charge du domaine ou du sous-domaine mentionné
PTR	Permet de renseigner le nom associé à l'adresse IP considérée. Uniquement dans les zones DNS inverses.

## 9.2.2 DNS à HELMo

La haute école HELMo dispose de plusieurs serveurs DNS. Ainsi, il y a :

```
$ host -t NS helmo.be
helmo.be name server ns.helmo.be.
helmo.be name server ns2.helmo.be.
helmo.be name server ns6.gandi.net.
```

*La commande host (que nous détaillerons plus loin) permet d'interroger le service DNS. Ainsi, nous lui demandons ici les enregistrements de type NS pour le domaine helmo.be. Nous obtenons 3 serveurs responsables de la zone helmo.be : le serveur principal ns.helmo.be et les 2 serveurs secondaires ns2.helmo.be et ns6.gandi.net.*

Nous pouvons également savoir les serveurs qui gèrent le mail :

```
$ host -t MX helmo.be
helmo.be mail is handled by 10 smtp.helmo.be.
helmo.be mail is handled by 20 smtp2.helmo.be.
```

*Nous apprenons ici qu'il y a 2 serveurs gérant le courrier à HELMo : le serveur principal smtp.helmo.be et le serveur secondaire smtp2.helmo.be. Les valeurs 10 et 20 représentent la priorité avec laquelle le serveur doit être contacté : plus ce nombre est faible, plus ce serveur est prioritaire. Ainsi, c'est d'abord le serveur smtp.helmo.be qui doit être contacté. Si celui-ci ne peut répondre, le serveur de backup smtp2.helmo.be peut alors être contacté.*

## 9.3 DNS interne et DNS externe

Il arrive très souvent que la réponse du serveur DNS doive être **adaptée en fonction de l'origine de la requête**. Ainsi, si l'on se trouve dans le réseau interne, il est souvent nécessaire que le serveur DNS retourne l'adresse IP interne correspondant à la requête. Par contre, si la requête provient de l'extérieur du réseau, il faut alors que la réponse mentionne l'adresse IP publique du serveur.

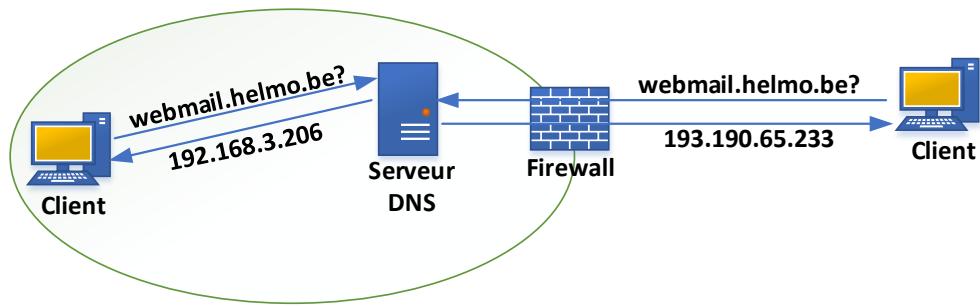


Figure 9.1 : réponse différente en fonction de l'origine

Ainsi, comme le montre la figure 9.1, la réponse pour la requête `webmail.helmo.be` sera différente selon que le client se trouve à l'intérieur du réseau de HELMo (et donc l'adresse IP interne `192.168.3.206` est retournée) ou à l'extérieur de celui-ci (et, dans ce cas, c'est l'adresse IP publique `193.190.65.233` qui est retournée).

## 9.4 Configuration du service DNS

Si nécessaire, les packages DNS peuvent être installés :

```
$ yum install bind-chroot
```

La version *chroot* est une installation sécurisée du service DNS. C'est la raison pour laquelle c'est cette version qui est installée et utilisée.

La configuration est concentrée en plusieurs endroits :

- Le fichier `/var/named/chroot/etc/named.conf` – Ce fichier doit lister les différents domaines configurés et gérés par le serveur DNS. Dans ce fichier, chaque domaine est lié au fichier décrivant les enregistrements de zone DNS. Par exemple, celui de HELMo :
  - Liste tous les domaines que le serveur DNS gère. On y trouve ainsi les domaines `helmo.be`, `helmo.eu`, `student.helmo.be`, `cg.helmo.be` ou encore `co.helmo.be`.
  - Pour chaque domaine listé, un fichier de zone, placé dans le dossier `/var/named/chroot/var/named` est mentionné.
- Le dossier `/var/named/chroot/var/named` – Ce dossier doit contenir les fichiers décrivant les zones DNS. Ainsi, il convient de créer un fichier par nom DNS souhaité (au moins) et d'y mentionner, à l'intérieur, les enregistrements de zone. Ainsi, à HELMo,
  - Le fichier décrivant les enregistrements de la zone `helmo.be` se trouve dans `/var/named/chroot/var/named/db.helmo.be`. A l'intérieur, on trouve les enregistrements `www`, `elearning`, `webmail`, ... et les adresses IP correspondantes.

Dans la suite, nous allons détailler pas à pas la configuration d'un nouveau serveur DNS.

### 9.4.1 Etape 1 : récupérer le modèle de configuration

Dans cette étape, nous allons récupérer le modèle de fichier de configuration. Une fois ce modèle récupéré, nous pourrons modifier celui-ci.

```
$ cd /var/named/chroot/etc
$ cp /usr/share/doc/bind-9.9.4/sample/etc/* .
```

Cette dernière commande copie les fichiers `named.conf` et `named.rfc1912.conf` dans le dossier `/var/named/chroot/etc`.

#### 9.4.2 Etape 2 : configuration du fichier `named.conf`

Nous allons, dans cette étape, parcourir le fichier `named.conf` et pointer les éléments de configuration importants.

##### Section options

Dans la section `options`, il est possible de définir les *options globales* applicables sur le serveur DNS. Le paramètre `directory` désigne le répertoire de travail. Les options `dump-file`, `statistics-file` et `memstatistics-file` sont bien configurées par défaut.

Voici quelques options importantes qu'il peut être nécessaire d'ajouter/d'adapter :

Option	Explication
<code>listen-on</code>	Détermine sur quelles adresses IPv4 le serveur DNS écoute. Par défaut, il s'agit uniquement <code>127.0.0.1</code> , ce qui posera un problème. Il est possible de changer cette option en <code>any</code> . Par défaut : <code>listen-on port 53 { 127.0.0.1; };</code>
<code>listen-on-v6</code>	Détermine sur quelles adresses IPv6 le serveur DNS écoute. Par défaut, il s'agit uniquement <code>::1</code> , ce qui posera un problème. Il est possible de changer cette option en <code>any</code> . Par défaut : <code>listen-on-v6 port 53 { ::1; };</code>
<code>allow-query</code>	Détermine à partir d'où les requêtes sont autorisées. <code>allow-query</code> concerne les domaines gérés par le serveur DNS. Par défaut, seul <code>localhost</code> est accepté comme origine. Il est préférable de placer cette option à <code>any</code> pour autoriser le serveur à répondre sur les domaines qu'il gère. Par défaut : <code>allow-query { localhost; } ;</code>
<code>allow-query-cache</code>	Détermine à partir d'où les requêtes sont acceptées. <code>allow-query-cache</code> concerne les autres requêtes (celles qui ne concernent pas les domaines gérés par le serveur). Ainsi, le serveur doit-il répondre à « Qui est <code>www.google.com</code> ? » à ce client ? Par défaut, seul <code>localhost</code> est accepté comme origine. Cette option devra également être modifiée pour répondre aux requêtes extérieures. Par défaut : <code>allow-query-cache { localhost; } ;</code>
<code>version</code>	Cette option détermine la chaîne qui sera retournée si un client demande la version du serveur <code>bind</code> . Il est intéressant de définir une valeur pour des raisons de sécurité. Ex : <code>version "1234";</code>
<code>recursion</code>	Cette option détermine si le serveur interroge les autres serveurs pour obtenir une réponse (est-il configuré pour donner une réponse au client ou bien se limite-t-il à retourner la liste des serveurs mondiaux s'il ne gère pas le nom DNS demandé). Par défaut : <code>recursion yes;</code>
<code>dnssec-enable</code> <code>dnssec-validation</code> <code>dnssec-lookaside</code>	Ces options activent les extensions de sécurité dnssec. Si DNSSEC n'est pas utilisé, il est possible de désactiver celui-ci. Par défaut : <code>dnssec-enable yes;</code>

	<code>dnssec-validation yes ; dnssec-lookaside auto;</code>
<b>forwarders</b>	<p>Cette option liste les serveurs DNS vers lesquels ce serveur peut faire suivre les requêtes pour les domaines qu'il ne gère pas. Cette option est utilisée pour alléger le serveur DNS en transmettant les requêtes à un autre serveur qui est alors chargé de faire la résolution. Cette option est également nécessaire si l'administrateur réseau a limité les serveurs DNS autorisés.</p> <p>Par exemple :</p> <pre>forwarders { 192.168.128.2; };</pre>

### Les vues

Le fichier de configuration comporte plusieurs sections `view`. Ces sections **permettent d'adapter la réponse du serveur en fonction de l'origine de la requête**. Ainsi, si la requête provient du réseau interne, il faut s'assurer que la réponse apportée sera l'adresse IP interne alors que si la requête provient de l'extérieur, l'adresse IP publique doit être retournée (voir figure 9.1).

La structure de la description d'une vue est :

```
view "nom_de_la_vue"
{
    match-clients { ... } ;
    recursion ... ;
    ...
}
```

Les vues sont directement liées à l'option `match-clients` qui impose que la vue choisie est sélectionnée si l'origine de la requête est listée dans `match-clients`. Il est possible de lister des adresses IP, des prefixes IP sous la forme ADRESSE/MASQUE, ou encore d'utiliser les mots clés `localhost` (cette machine seulement), `localnets` (tous les réseaux auxquels cette machine est directement connectée) ou `any` (n'importe quelle origine).

Exemple :

```
match-clients { 192.168.5.0/24; 192.168.3.5; 2001:6a8:2cc0:8020::/64; };
match-clients { localnets; };
match-clients { any; };
```

Le modèle de configuration propose les 3 vues suivantes dans cet ordre (en partant du plus spécifique au plus général) :

Nom de la vue	match-clients	Explication
<b>localhost_resolver</b>	localhost	Cette vue est utilisée par le serveur lorsque la requête émane de lui-même. Il est important d'y lister toutes les zones connues du serveur et de mentionner l'option <code>recursion yes</code> ;
<b>internal</b>	localnets	Cette vue est utilisée par le serveur lorsque la requête émane d'un réseau auquel il est directement connecté (réseau interne donc). Il est important d'y lister toutes les zones connues du serveur et d'y ajouter l'option <code>recursion yes</code> ;
<b>external</b>	any	Cette vue est utilisée par le serveur lorsque la requête

		provient d'ailleurs (ni du serveur lui-même, ni d'un réseau directement relié à ce serveur). Il est important de lister uniquement les zones gérées par ce serveur et de préciser l'option <code>recursion no;</code> afin que le serveur ne réponde pas aux requêtes pour lequel il ne fait pas autorité (ie. Les domaines qu'il ne gère pas).
--	--	---

### La description d'une zone directe

A l'intérieur de chaque vue, nous trouvons la liste des domaines gérés par ce serveur. Il est possible de lister ces domaines dans le fichier `named.conf` ou, via l'option `include`, de les spécifier dans un fichier annexe.

Comme le nombre de zone ne sera pas trop important ici, nous choisirons de les lister dans le fichier `named.conf`.

La description d'une zone prend toujours la forme suivante :

```
zone "nom.dns.de.la.zone" {  
    type master|hint;  
    file "fichierDeZone.db";  
    allow-update { none; };  
    allow-transfer { none; };  
};
```

On commence par donner la zone DNS en question. Ainsi pour les serveurs de HELMo, nous avons une entrée zone "helmo.be". Ensuite, on doit mentionner le type de zone : `hint` (uniquement valide pour la zone « . » qui mentionne les serveurs DNS mondiaux) ou `master` (indique que le serveur gère la zone DNS en question). Donc pour une zone directe (sur le serveur principal, nous aborderons le serveur de backup plus loin), ce sera toujours `master`.

Ensuite, avec `file`, il faut donner le fichier contenant les enregistrements DNS de cette zone. Dans le fichier mentionné, nous trouverons tous les noms des machines et les adresses IP associées à ces noms. Il s'agit du fichier de zone correspondant dont nous détaillerons le contenu à l'étape suivante.

L'option `allow-update` est utilisée dans la configuration des zones DNS dynamiques (par exemple lorsqu'un serveur DHCP ajoute dynamiquement les entrées quand un nouveau client se connecte). Nous n'étudierons pas cette possibilité, raison pour laquelle les mises à jour dynamiques sont refusées.

Enfin, l'option `allow-transfer` détermine vers quels serveurs DNS le transfert de zone est autorisé. Il s'agit d'une option à prendre en compte lorsqu'on dispose d'un serveur principal et d'un ou plusieurs serveurs secondaires (ou backups) qui doivent se synchroniser avec le serveur principal.

Par exemple, à HELMo, la description de la zone `helmo.be` dans la vue `internal` prend la forme :

```
zone "helmo.be" in {  
    type master;  
    file "db.helmo.be.internal28";
```

---

<sup>28</sup> Afin de faire la distinction entre les réponses internes (avec les adresses IP privée) et externes (avec les adresses IP publiques), des fichiers de zones différents sont utilisés. Ainsi, dans la vue interne, il y a un fichier

```

        allow-update { none; };
        allow-transfer { 192.168.3.30; 192.168.3.19; 192.168.3.208;
192.168.0.2; 10.0.0.2; 192.168.0.10; 192.168.224.200; 192.168.20.2;
192.168.20.3; 192.168.3.206; 192.168.27.2; 192.168.128.2; };
    };

```

Comme nous pouvons le voir dans le fichier named.conf, celui-ci liste des zones en exemple comme *my.internal.zone*, *my.slave.zone*, *my.ddns.internal.zone* et *my.external.zone*. **Toutes ces zones doivent être supprimées ou adaptées en fonction des besoins.**

### **La description d'une zone inverse**

Décrire une zone inverse est semblable à la définition d'une zone directe. Pour rappel, la zone inverse est utilisée lorsqu'il faut *traduire* une adresse IP en nom. A l'exception du nom de la zone, qui est très particulier, on y trouve les mêmes informations que pour la zone directe. Ainsi, le nom de la zone est formé, le plus souvent, par les 3 derniers octets de l'adresse IP, inversé et suffixé par *in-addr.arpa*.

Nom de la zone	Utilisée pour traduire les IP
<b>128.168.192.in-addr.arpa</b>	192.168.128.x → nom DNS
<b>1.0.10.in-addr.arpa</b>	10.0.1.x → nom DNS

Exemple de définition d'une zone inverse :

```

zone "21.168.192.in-addr.arpa" in {
    type master;
    file "db.192.168.21";
    allow-update { none; };
    allow-transfer { 192.168.3.30; 192.168.3.19; 192.168.3.208;
192.168.0.2; 10.0.0.2; 192.168.0.10; 192.168.224.200; 192.168.20.2;
192.168.20.3; 192.168.3.206; 192.168.27.2; 192.168.128.2; };
};

```

Comme nous pouvons le voir, celle-ci est très proche de la définition d'une zone directe.

### **La description d'une zone de backup (ou slave, secondaire)**

La description d'une zone de backup (également appelée *slave* ou *secondaire*) ne demande pas beaucoup de configuration. On définit le nom de la zone (identique à celle définie sur le serveur maître), on place le type à *slave*, on mentionne le fichier décrivant cette zone et on mentionne, avec l'option *masters*, le ou les serveurs DNS maîtres.

Sur les serveurs DNS maîtres, il **faut autoriser le transfert de zone**. Pour ce faire, il faut que l'IP du serveur de backup apparaisse dans la section *allow-transfer* du serveur maître.

Par exemple, sur le serveur DNS secondaire de HELMo (*ns2.helmo.be*), nous avons la déclaration suivante pour le transfert de la zone *helmo.be* :

---

de zone *db.helmo.be.internal* (listant des IP privées) alors que la vue externe référence le fichier *db.helmo.be.external* (listant des IP publiques).

```
zone "helmo.be" in {
    type slave;
    file "slaves/db.helmo.be.internal";
    allow-transfer { none; };
    masters { 192.168.3.209; };
};
```

#### 9.4.3 Etape 3 : Les enregistrements d'une zone DNS

Nous avons jusqu'à présent indiqué au serveur DNS le nom des domaines qu'il devait gérer (par exemple `mondomaine.be` ou `monautredomaine.org`). Pour chacun d'eux, nous avons mentionné un fichier de zone. Nous allons maintenant détailler les enregistrements DNS qu'il est possible d'indiquer dans ces fichiers de zone pour permettre au serveur de répondre à des requêtes comme « qui est www.mondomaine.be ? » ou « webmail.monautredomaine.org ? ».

Les fichiers décrivant les enregistrements des zones doivent être placés dans le dossier `/var/named/chroot/var/named/`.

#### Récupération des modèles de fichiers de zone

Nous allons commencer par récupérer les modèles des fichiers de zone :

```
$ cd /var/named/chroot/var/named/
$ cp -a /usr/share/doc/bind-9.9.4/sample/var/named/* .
$ chown -R named.named *
$ rm my.* slaves/my.*
```

Nous avons récupéré les dossiers `data` et `slaves` ainsi que les fichiers `named.ca` (reprenant la liste des serveurs DNS mondiaux, renseigné par la zone « . »), `named.empty`, `named.localhost` et `named.loopback`. Grâce à la commande `chown`, nous changeons le propriétaire de tous les fichiers et dossiers par l'utilisateur `named` et le groupe `named`. Enfin, la commande `rm` supprime les zones parasites et vides données à titre d'exemple.

#### Ajout du fichier de zone directe

Le fichier qui doit contenir les enregistrements DNS doit porter le nom référencé par l'option `file` de la zone correspondante. Dans l'exemple de HELMo donné ci-avant, le nom du fichier doit être `db.helmo.be.internal`.

Ce fichier texte doit contenir des éléments particuliers. Ainsi l'en-tête commence ainsi :

```
$TTL 86400
@ IN SOA monserver.mondomaine.be. admin.mondomaine.be. (
                    2015080801 ; Serial
                    28800      ; Refresh
                    14400      ; Retry
                    3600000   ; Expire
                    3600 )     ; Name Error
```

L'entête comment par \$TTL qui indique, en seconde, la durée pendant laquelle les informations peuvent être mise en cache. Ainsi, une valeur de 0 mentionnerait que ces données ne peuvent pas être mise en cache.

La seconde ligne contient les éléments suivants, **tous séparés par des tabulations** :

Elément	Explication
@	Référence automatiquement le nom de la zone actuelle. Nous aurions pu indiquer mondomaine.be.
IN SOA	Indique quel serveur fait autorité sur la zone. Le nom du serveur, complètement qualifié (et donc se terminant par un .) est mentionné ensuite.
monserver.mondomaine.be.	Nom du serveur principal faisant autorité pour la zone définie (ici mondomaine.be.). Le nom du serveur doit être <i>complètement qualifié</i> .
admin.mondomaine.be.	Adresse mail du gestionnaire de la zone (pour laquelle le symbole « @ » a été changé par « . »)
2015080801	Numéro de série attaché à la zone DNS. A chaque changement, il faut mettre à jour (incrémenter par exemple) celui-ci. Ce numéro de série permet aux serveurs secondaires de se synchroniser correctement. Traditionnellement, le numéro de série est écrit sous la forme AAAAMJJXX (avec AAAAMJJ la date de création du fichier de zone et XX la version actuelle).
28800	Indique le temps, en secondes, après lequel les serveurs backups (ou secondaires) se resynchroniseront (ici 8 heures).
14400	Indique le temps, en seconde, entre les tentatives de connexion aux serveurs maîtres lorsque celui-ci ne répond plus (ici 4 heures).
3600000	Indique le temps, en secondes, après lequel les serveurs secondaires considéreront que l'information est périmée quand ils n'arrivent plus à contacter les serveurs principaux (ici 1000 heures, soit 41 jours).
3600	Indique le temps, en secondes, pendant lequel un réponse de type NXDOMAIN peut-être conservée par un serveur (ici 1 heure). En cas de défaillance du DNS ou de nom inexistant, les serveurs qui tentent de résoudre un nom sur mondomaine.be généreront une erreur de type NXDOMAIN. Cette réponse peut être mise en cache par le serveur pendant 1 heure maximum (après, il doit à nouveau tenter de résoudre le nom, si celui-ci est demandé à nouveau).

Une grande difficulté dans la gestion du DNS sur Internet est la présence des *caches* sur tous les serveurs. En effet, pour éviter d'interroger plusieurs fois le DNS pour les mêmes noms, ceux-ci sont maintenus en caches par les serveurs DNS. Cela améliore les performances et diminue le trafic DNS global. Cependant, le désavantage est qu'en cas de modification de la zone DNS, la propagation de cette modification **peut prendre jusqu'à 24h**.

Après cette entête, nous trouverons les enregistrements de type NS, MX, A, AAAA et CNAME. A titre d'exemple illustratif, voici un extrait du fichier de zone db.helmo.be.internal :

```
$TTL      86400
@       IN      SOA      ns.helmo.be.    hostmaster.helmo.be.  (
                           2008011117      ; Serial
                           28800        ; Refresh
                           14400        ; Retry
```

```

3600000      ; Expire
3600 )      ; Name error

; CRITICAL INFORMATION
; ** Name server
        IN      NS      ns.helmo.be.
        IN      NS      ns2.helmo.be.
        IN      NS      ns6.gandi.net.

; ** Main mail server
helmo.be.      IN      MX      10      smtp.helmo.be.
                IN      MX      20      smtp2.helmo.be.
student.helmo.be.    IN      MX      10      smtp.helmo.be.
                    IN      MX      20      smtp2.helmo.be.

; COMPUTER
helmo.be.      IN      A       192.168.3.203
ns1            IN      A       192.168.3.206
                IN      AAAA    2001:6a8:2cc0:8000::206
webmail         IN      CNAME   ns1.helmo.be.
www             IN      A       192.168.3.203

```

Dans l'extrait présenté, juste après l'entête, nous avons l'enregistrement `NS` avec la liste de tous les serveurs DNS (principal et secondaires) gérant cette zone. Cet un élément important car ces informations sont utilisées pour notifier aux serveurs les modification de zones.

Nous avons ensuite, avec `MX`, la liste des serveurs mails utilisés et leurs priorités.

Enfin, nous avons les différentes entrées créées dans la zone (ici `ns1.helmo.be`, `webmail.helmo.be` et `www.helmo.be`). Il faut noter qu'un nom qui n'est pas complètement qualifié (et qui ne se termine pas par un « . ») se verra automatiquement suffixer par le nom de la zone.

Ainsi `ns1` désigne `ns1.helmo.be`. Nous aurions pu écrire, également `ns1.helmo.be.` qui est complètement qualifié.

Enfin, rappelons que `A` permet d'associer le nom et l'adresse IPv4 alors que `AAAA` permet d'associer le nom et l'adresse IPv6 et que `CNAME` représente un raccourci.

### *Ajout d'un fichier de zone inverse*

Pour les zones DNS inverses, la configuration est semblable à celle que nous venons de voir. Ainsi, nous retrouvons une entête similaire à celle de la zone directe et puis les enregistrements `NS` et enfin nous avons, pour les adresses IP, le nom vers lequel elle pointe (enregistrements `PTR`).

Voici un exemple de fichier de zone inverse pour `192.168.21.x` :

```

$TTL 86400
@     IN      SOA     ns.helmo.be.      hostmaster.helmo.be.  (
                                2013072703      ; Serial
                                28800          ; Refresh
                                14400          ; Retry
                                3600000        ; Expire

```

```
3600 ) ; Name error

; CRITICAL INFORMATION
; ** Name server
    IN      NS      ns.helmo.be.
    IN      NS      ns2.helmo.be.

; PTR RECORD
1      IN      PTR      sw-hp1910-campusguillemins.net.helmo.be.
2      IN      PTR      sw-hp2920-campusourthe.net.helmo.be.
252     IN      PTR      cp4600-stmartin.net.helmo.be.
253     IN      PTR      sw-hp2920-campusguillemins.net.helmo.be.
```

Il faut ainsi comprendre que l'adresse IP 192.168.21.1 pointe vers le nom sw-hp1910-campusguillemins.net.helmo.be. Il en va de même pour les autres entrées.

#### 9.4.4 Démarrer le service DNS

Pour démarrer le service DNS, il faut utiliser la commande `systemctl` comme suit :

```
$ systemctl start named-chroot
```

Pour activer le service au démarrage de la machine :

```
$ systemctl enable named-chroot
```

#### 9.4.5 Configurer les postes clients

Configurer les postes clients signifie que ces derniers utiliseront le service DNS installé sur cette machine. Il faut donc modifier les paramètres DNS utilisés. Cela peut se faire de plusieurs manières :

- Si un serveur DHCP est utilisé pour configurer automatiquement les postes clients, il est nécessaire de modifier la configuration de ce service pour qu'il renseigne l'adresse IP du nouveau serveur DNS.
- Si la configuration IP est statique sur les postes clients, il **faut se référer à la leçon décrivant la configuration réseau** pour adapter les serveurs DNS utilisés.

### 9.5 Les outils DNS

Il existe quelques outils permettant d'interroger, de manière directe, les serveurs DNS souhaités. Ces outils sont particulièrement intéressants pour déterminer si le serveur DNS répond correctement aux requêtes.

#### 9.5.1 nslookup

L'outil nslookup est standard sur toutes les plateformes (Windows, Linux et MacOS X). Il permet d'interroger, de manière interactive, un serveur DNS.

```
$ nslookup
> www.helmo.be
Server: 192.168.3.209
```

```
Address: 192.168.3.209#53

Name: www.helmo.be
Address: 192.168.3.203
> 192.168.21.253
Server: 192.168.3.209
Address: 192.168.3.209#53

253.21.168.192.in-addr.arpa name = sw-hp2920-campusguillemins.net.helmo.be.
> set type=MX
> helmo.be
Server: 192.168.3.209
Address: 192.168.3.209#53

helmo.be mail exchanger = 20 smtp2.helmo.be.
helmo.be mail exchanger = 10 smtp.helmo.be.
```

Comme nous pouvons le voir, il est possible d'obtenir des réponses pour des requêtes sur la zone directe, sur la zone inverse et même spécifier le type d'enregistrement souhaité.

### 9.5.2 host

La commande `host` permet le même type d'interrogation que `nslookup`. Elle n'est, cependant, pas interactive puisque l'utilisateur doit décrire sa requête par la ligne de commande.

```
$ host www.helmo.be
www.helmo.be has address 192.168.3.203

$ host -t NS helmo.be
helmo.be name server ns2.helmo.be.
helmo.be name server ns6.gandi.net.
helmo.be name server ns.helmo.be.

$ host 192.168.21.252
252.21.168.192.in-addr.arpa domain name pointer cp4600-stmartin.net.helmo.be.
```

### 9.5.3 dig

Le dernier outil intéressant pour interroger le service DNS est `dig`. A la différence des outils précédents, `dig` permet également de *tracer* une requête DNS, c'est à dire de visualiser tous les serveurs interrogés et les réponses apportées. Cet outil est particulièrement apprécié dans l'analyse des problèmes DNS.

`dig` permet d'effectuer beaucoup de tâches, je vous renvoie vers la page de manuel pour plus d'explication sur cet outil. Nous nous contenterons, ici, d'explorer la possibilité de *tracer* une requête.

```
$ dig +trace www.swila.be

; <>> DiG 9.7.3-RedHat-9.7.3-1.el5 <>> +trace www.swila.be
;; global options: +cmd
.          449507      IN      NS      f.root-servers.net.
.          449507      IN      NS      k.root-servers.net.
.          449507      IN      NS      h.root-servers.net.
.          449507      IN      NS      l.root-servers.net.
```

```
449507      IN      NS      c.root-servers.net.  
449507      IN      NS      d.root-servers.net.  
449507      IN      NS      b.root-servers.net.  
449507      IN      NS      m.root-servers.net.  
449507      IN      NS      j.root-servers.net.  
449507      IN      NS      e.root-servers.net.  
449507      IN      NS      a.root-servers.net.  
449507      IN      NS      g.root-servers.net.  
449507      IN      NS      i.root-servers.net.  
;; Received 272 bytes from 192.168.3.209#53(192.168.3.209) in 40 ms  
  
be.          172800    IN      NS      a.ns.dns.be.  
be.          172800    IN      NS      b.ns.dns.be.  
be.          172800    IN      NS      c.ns.dns.be.  
be.          172800    IN      NS      d.ns.dns.be.  
be.          172800    IN      NS      x.ns.dns.be.  
be.          172800    IN      NS      y.ns.dns.be.  
;; Received 397 bytes from 2001:7fd::1#53(k.root-servers.net) in 11 ms  
  
swila.be.    86400   IN      NS      ns1.sllabs.net.  
swila.be.    86400   IN      NS      ns6.gandi.net.  
swila.be.    86400   IN      NS      ns2.sllabs.net.  
;; Received 100 bytes from 2001:dcd:7::8#53(y.ns.dns.be) in 8 ms  
  
www.swila.be.        172800    IN      A      178.32.45.186  
;; Received 46 bytes from 2001:41d0:2:25a8:ff10::10#53(ns1.sllabs.net) in  
14 ms
```

La 1<sup>ère</sup> étape consiste à trouver les serveurs mondiaux (la zone « . »). Un de ces serveurs est interrogé (`k.root-servers.net`) et nous retourne la liste des serveurs gérant la zone « `.be` ». Un de ces serveurs est interrogé (`y.ns.dns.be`) et nous renvoie la liste des serveurs gérant la zone « `swila.be` ». A nouveau, un de ces serveurs est interrogé (`ns1.sllabs.net`) et nous renvoie la réponse demandée 178.32.45.186.

© Louis SWINNEN 2020, tous droits réservés

## 9.6 Exercices

On vous demande de :

- 1) Configurer un serveur DNS pour la zone <nomdefamille>.cg.helmo.be.
    - a) Dans la « vue locale », préciser les enregistrements suivants :
      - o Entrée NS : ns.<nomdefamille>.cg.helmo.be
      - o Les adresses suivantes :
        - ns.<nomdefamille>.cg.helmo.be → 192.168.190.x (IP de votre serveur)
        - gate.<nomdefamille>.cg.helmo.be → IP de pfSense
        - pfsense.<nomdefamille>.cg.helmo.be → CNAME vers gate
    - b) Dans la « vue externe », préciser les enregistrements suivants :
      - o Entrée NS : gate.<nomdefamille>.cg.helmo.be
      - o Les adresses suivantes :
        - ns.<nomdefamille>.cg.helmo.be → CNAME gate

- gate.<nomdefamille>.cg.helmo.be → IP WAN de pfSense
  - pfsense.<nomdefamille>.cg.helmo.be → CNAME vers gate
- c) Configurer votre serveur pour qu'il interroge le service DNS que vous venez de configurer.  
Activer ce service DNS au démarrage de la machine.
  - d) Modifier la configuration réseau de votre serveur pfSense pour ouvrir le port 53 et le rediriger vers votre serveur Linux. Votre serveur DNS est ainsi accessible à distance.
  - e) Tester votre configuration à partir du poste Windows. Vous pouvez utiliser l'outil nslookup pour valider votre configuration.
- 2) Configurer votre serveur DNS pour qu'il soit serveur secondaire de votre voisin. Pour ce faire, vous utiliserez l'adresse IP WAN.
  - 3) Ajouter une zone DNS inverse pour les IPs 192.168.190.x. Ajoutez-y la machine Windows, le firewall pfSense et votre serveur Linux.

« *Celui qui contrôle le DNS, contrôle le monde* »

Comme nous l'avons vu, l'administrateur du serveur DNS peut configurer celui-ci comme il veut. Rien n'empêche d'indiquer dans votre serveur que vous gérez la zone `facebook.com` ou `google.com`<sup>29</sup>. Ces manipulations vous permettraient ainsi de *capturer* le trafic à destination de ces sites sur votre réseau, d'afficher un avertissement, ...

Une extension du DNS appelée DNSSEC permet de s'assurer que les informations obtenues auprès du serveur sont licites. Ainsi, pour `www.swila.be`, DNSSEC est activé et permet de vérifier **depuis les serveurs mondiaux jusqu'au domaine swila.be** que l'entrée `www` pointe bien vers l'adresse IP licite.

Nous remarquons cependant que DNSSEC se déploie timidement aujourd'hui. Cette leçon étant déjà assez longue et compliquée à appréhender sans la présentation de cette extension, j'ai choisi de ne pas l'aborder. Cependant, je ne peux que vous encourager à déployer ce mécanisme si vous devez mettre en place un serveur DNS sur Internet.

<sup>29</sup> A ce titre, Google a pris des mesures de sécurité importantes dans son navigateur Chrome pour éviter ce genre de manipulations. En effet, la clé publique des certificats utilisés par Google est connue du navigateur et celui-ci peut donc déterminer s'il est sur le site licite de Google ou non. De plus, le déploiement progressif de la technologie *certificate transparency* permet également de contrer ces manipulations pour les sites protégés par des certificats.

## Leçon 10 : Administration à distance et installation d'applications

### 10.1 Introduction à l'administration à distance

Administrer un serveur Linux à distance peut se faire de plusieurs façons. La plus simple est l'accès en mode terminal (ie. *ligne de commande*) au serveur, en utilisant SSH. Ce service, très utile pour démarrer des tâches à distance est installé par défaut dans le plupart des distributions. Bien sûr, à côté de cette possibilité de connexion *en mode texte*, il existe diverses méthodes permettant également un accès *en mode graphique* à des serveurs Linux, même depuis un poste client Windows.

Ainsi, parmi les solutions très courantes, il y a l'antique XDMCP véritable précurseur dans le domaine, qui permettait à des terminaux très léger de se connecter à un serveur puissant. Une autre solution, répandue également dans le monde Microsoft, est l'utilisation de VNC, cependant la sécurité assez basique de ce protocole le rend inutilisable depuis internet sans la configuration de VPN ou autre mécanisme de sécurité avancé. Enfin, il y a le service NX et son évolution X2Go qui permet de se connecter en mode graphique à un serveur Linux en se basant sur le service SSH.

### 10.2 Configuration et utilisation de SSH

Le service SSH est un des services les plus déployés sur les machines exécutant Linux. Il est même activé par défaut dans la plupart des distributions, ce qui n'est pas nécessairement une bonne idée<sup>30</sup>. Cependant, grâce à ce service, il est possible de :

1. Se connecter, de manière interactive et sécurisée, à un serveur distant. Sous Windows, l'utilisation de Putty permet de réaliser cette tâche.
2. Echanger des fichiers en utilisant le protocole SCP (*secure copy*) dont nous parlerons plus tard. Sous Windows, des logiciels comme WinSCP ou FileZilla supportent ce protocole.
3. Démarrer une commande à distance, même à partir d'un langage de programmation. Ainsi, il est possible de lancer un script, déployé sur un serveur, pour réaliser des tâches particulières<sup>31</sup>. Citons, par exemple, **TamirSSH** dans l'environnement .NET ou encore **JSch** dans l'environnement Java qui permettent de réaliser cela.

La configuration de SSH se trouve dans le dossier `/etc/ssh` et est scindée en 2 parties distinctes :

1. La configuration du **client SSH** qui se trouve dans `/etc/ssh/ssh_config`. Celle-ci est utilisée lorsqu'on se connecte, depuis la machine linux courante vers un serveur extérieur.
2. La configuration du **service SSH** qui se trouve dans `/etc/ssh/sshd_config`. Celle-ci est utilisée lorsqu'une connexion est établie depuis une autre machine vers ce serveur, en utilisant SSH.

Nous allons dans la suite aborder la configuration *serveur* c'est-à-dire, celle du **service SSH**.

Ainsi, les options suivantes, qui peuvent être placées dans le fichier `sshd_config`, peuvent être intéressantes :

Option	Explication
<code>Port 22</code>	C'est le port d'écoute du service SSH. Le port 22 est, par

<sup>30</sup> En effet, quelle est la réelle utilité d'avoir un service SSH exécuté sur un PC de bureau ?

<sup>31</sup> A HELMo, cette fonctionnalité est utilisée pour la création automatique des boîtes mails utilisateurs lors de l'inscription d'un nouvel étudiant. Un programme .NET appelle un script PERL via SSH en utilisant TamirSSH.

	défaut, dédié à ce service. Il est parfois intéressant (ou nécessaire) de modifier le port par défaut.
<b>ListenAddress 0.0.0.0</b> <b>ListenAddress ::</b>	Ces options permettent de déterminer sur quelle adresse IP le service SSH est actif. Par défaut, il est actif sur l'ensemble des adresses IP (IPv4 et IPv6) de la machine.  Si la machine dispose de plusieurs interfaces réseaux, on peut ainsi limiter l'accès au service SSH à une seule adresse.
<b>Protocol 2</b>	Il existe 2 version du protocole SSH. La version 1 est complètement dépassée et non-sûre. Il est donc recommandé de ne jamais activer cette version.
<b>PermitRootLogin yes</b>	Cette option permet à l'administrateur (i.e. l'utilisateur root) de se connecter à distance. Une bonne pratique est de désactiver cette possibilité.  Ainsi pour se connecter au serveur, il faut connaître un couple nom d'utilisateur / mot de passe. Il est ensuite possible de <i>devenir</i> administrateur grâce à la commande su.
<b>PermitEmptyPasswords no</b>	Cette option désactive tous les comptes qui ont un mot de passe vide, ce qui est une très bonne idée. Cette configuration est faite par défaut.
<b>X11Forwarding yes</b>	Permet d'activer le lancement d'application graphique à distance. Cette option est intéressante pour administrer un serveur, et lancer une application dont l'affichage se fait localement mais l'exécution à distance.  Pour utiliser cette option, il faut disposer d'un serveur X11. Ce serveur est présent dans les systèmes Unix principalement.
<b>AllowTcpForwarding yes</b>	Cette option autorise la création de tunnel SSH. Pour introduire ce sujet (sur lequel nous reviendrons par la suite), un tunnel SSH permet de se connecter au serveur « <i>comme si</i> » une connexion VPN existait. Cela permet de traverser le firewall pour accéder à des ressources internes.

### 10.2.1 Démarrer / Arrêter le service SSH

Pour démarrer le service SSH, il faut passer par la commande `systemctl` :

```
$ systemctl restart sshd
```

Pour désactiver le service SSH (sur des postes clients) :

```
$ systemctl disable sshd
```

### 10.2.2 Authentification dans SSH

Le service SSH, comme bien d'autres, utilise les modules PAM (*Pluggable Authentication Modules*) pour authentifier les utilisateurs. PAM est intéressant car il permet de généraliser l'utilisation d'autres mécanismes d'authentification que le traditionnel login / mot de passe. Ainsi, il est possible d'utiliser *Kerberos* (serveur de distribution de clés), *S/Key* (utilisation de mots de passe à usage unique), ou même *l'authentification à 2 facteurs* (Google Authenticator PAM module, Yubico PAM...).

A côté de PAM, le service SSH supporte également l'authentification par clé RSA ou DSA. Cette méthode d'authentification est simple :

1. On génère sur le poste client une paire de clé publique/privée.
2. On ajoute la clé publique dans les clés autorisées pour le compte visé
3. L'authentification n'utilise plus le mot de passe mais la clé privée correspondant à la clé publique déposée.

### **Etape 1 : générer les clés**

La génération de la paire de clés publiques / privées se fait par l'utilisation de la commande `ssh-keygen`. Lors de l'utilisation de la commande, il faut mentionner le type de clé souhaité.

Type de clé	Explication
<b>rsa</b>	Utilisation d'une paire de clés RSA. Ce sont les clés les plus courantes utilisées. Dans un avenir proche, ces clés polynomiales seront remplacées par des clés basées sur des courbes elliptiques
<b>dsa</b>	Utilisation d'une paire de clés DSA. Ces clés sont moins courantes et sont basées sur des logarithmes discrets. Ce chiffrement cédera sa place aux clés basées sur des courbes elliptiques dans un avenir proche.
<b>ecdsa</b>	Standard sur les clés basées par courbes elliptiques. Leur déploiement est encore très restreint. Ainsi, le programme Putty ne supporte pas encore ces clés.
<b>ed25519</b>	Second standard basé sur les clés par courbes elliptiques. Le déploiement est toujours très restreint. Il n'y a pas de grande différence entre ecdsa et ed25519, ni en terme de sécurité, ni en terme de performance.

Actuellement, l'option **rsa** semble est celle qu'il faut privilégier :

```
$ ssh-keygen -t rsa
```

Cette commande génère une clé **rsa** de 2048 bits. Il est possible de générer une clé plus longue, pour davantage de sécurité en ajoutant à la fin de la commande l'option suivante : `-b 4096`.

**Attention !** Lors de la génération de la clé, le système demande *un code de verrouillage* protégeant la clé. Ainsi, celle-ci ne peut être lue sans entrer le *code de verrouillage* correspondant. L'utilisation d'un code de verrouillage non-vide est recommandée.

Une fois la clé installée, le système ne demande plus le mot de passe pour se connecter mais le code de verrouillage de la clé.

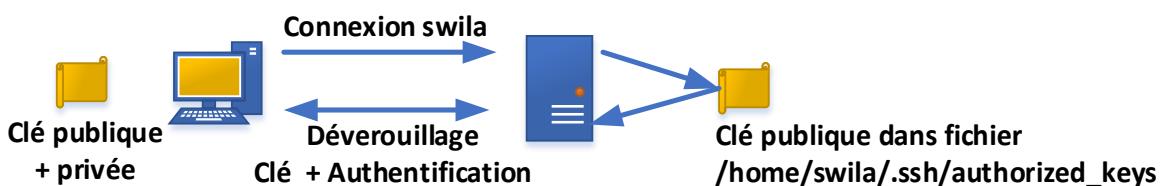


Figure 10.1 : Authentification SSH par clé publique / privée

La figure 10.1 résume le fonctionnement. Le client doit disposer de la clé publique et privée, le serveur distant dispose uniquement de la clé publique. Lors de la connexion, l'authentification utilise

la clé publique / privée au lieu du mot de passe : l'utilisateur est invité à entrer le code de verrouillage de la clé privée pour permettre à son ordinateur client de lire celle-ci.

### **Etape 2 : Ajout de la clé pour le compte visé**

**ATTENTION !** Comme nous l'avons annoncé, il faut ajouter la clé publique créée dans la liste des clés autorisées pour un compte donné. Nous avons également vu qu'il était recommandé de mentionner un *code de verrouillage* pour la clé en question. Il est par contre **COMPLETEMENT DECONSEILLE DE DEFINIR UNE CLE SANS CODE DE VERROUILLAGE VERS L'UTILISATEUR ROOT.**

Et d'une manière générale, l'authentification avec le compte *root* est déconseillée à distance. Il vaut toujours mieux se connecter avec un utilisateur et puis, en utilisant la commande `su`, devenir *root*.

Pour ajouter la clé créée sur la machine distante, il faut pouvoir s'y connecter (avec son login / mot de passe par exemple). La solution la plus simple consiste à utiliser le script fourni par SSH pour réaliser cette opération, à savoir `ssh-copy-id`<sup>32</sup>. Par exemple, à partir de la machine *CentOS7-Client* :

```
[swila@localhost ~]$ ssh-copy-id -i .ssh/id_rsa.pub lsw@172.18.1.2
```

Cette commande ajoute la clé contenue dans le fichier `id_rsa.pub` au compte `lsw` de la machine dont l'IP est `172.18.1.2`. Pour réaliser la copie, il faut entrer le mot de passe correspondant au compte `lsw`. Les authentifications ultérieures utiliseront la clé.

#### **POUR DES RAISONS DE SECURITE, LA CLE PRIVEE DOIT RESTER SECRETE.**

```
[swila@localhost ~]$ ssh lsw@172.18.1.2
```

Donc si nous revenons à la figure 10.1, il est possible d'expliquer le mécanisme d'authentification complètement désormais : lorsqu'une connexion est initiée (par exemple vers le compte `lsw` comme ci-dessus), le service SSH va vérifier si une (ou plusieurs) clé publique est présente dans le fichier `/home/lsw/.ssh/authorized_keys`. Si c'est le cas, et si le client dispose de la clé privée correspondante, alors l'authentification sur base de la clé publique / privée est initiée. Le système demande *le code de verrouillage* (si celui-ci existe) et authentifie l'utilisateur.

### **10.2.3 Code de verrouillage et agent**

Comme mentionné plus haut, un code de verrouillage devrait toujours être présent pour protéger les clés privées. **Il y a cependant une exception à cette règle : les tâches planifiées**<sup>33</sup>. En effet, quand une tâche est planifiée et doit réaliser une opération à distance, le service SSH est un bon moyen d'y arriver. Cependant, devoir entrer un mot de passe ou un code de verrouillage est un frein au déploiement du service. Il est dans ce cas possible de ne pas utiliser de code de verrouillage. Cependant, la remarque précédente s'applique : pas de connexion vers le compte *root* sans code de verrouillage.

Ainsi, en cas d'utilisation de `rsync`, `scp` ou toute autre commande se basant sur SSH, utiliser une authentification par clé sans code est possible.

---

<sup>32</sup> Il est possible de réaliser l'opération à la main : il faut copier-coller le contenu du fichier `id_rsa.pub` dans le fichier `/home/<login>/.ssh/authorized_keys` sur le serveur distant.

<sup>33</sup> Par exemple, une sauvegarde effectuée automatiquement durant la nuit.

### Comment faire pour ne pas devoir entrer son code de verrouillage tout le temps ?

Une question importante est de déterminer s'il est possible **de ne pas entrer tout le temps son code de verrouillage**. Il y a une façon simple, utiliser l'agent SSH.

L'agent SSH est un programme hautement sécurisé qui garde en mémoire une copie de la clé privée déverrouillée. Ainsi, lors de l'authentification, aucun code de verrouillage n'est nécessaire puisque la clé privée est déjà lue et disponible. Cette méthode facilite la vie de l'administrateur qui peut, ainsi, charger les clés en mémoire lors du lancement de sa machine et travailler sans avoir à entrer un seul mot de passe par la suite.

Le système *CentOS 7* lance automatiquement l'agent SSH. Ainsi, lors de la 1<sup>ère</sup> connexion utilisant la clé privée, le code de déverrouillage est demandé et la clé est gardée en mémoire.

Les connexions ultérieures ne nécessite plus de devoir entrer le code. La connexion est immédiate.

Si vous devez lancer l'agent à la main :

```
$ ssh-agent
```

Pour ajouter une clé en mémoire :

```
$ ssh-add ~/.ssh/id_rsa
```

### 10.2.4 Lancement d'application graphique à distance

Comme nous l'avons mentionné en introduction, il est possible de lancer l'exécution d'un programme à distance **mais avec un affichage local**. Pour ce faire, il faut que le serveur SSH distant ait activé l'option `X11Forwarding`, que le poste client dispose d'un serveur graphique X11 (ce qui est le cas sous la plupart des systèmes UNIX<sup>34</sup>) et que le client démarre sa session avec une option particulière :

```
$ ssh -X p010544@dartagnan.cg.helmo.be
$ xeyes
```

L'application `xeyes` est lancée depuis le serveur *dartagnan* mais est affichée localement.

### 10.2.5 Etablissement d'un tunnel SSH

En utilisant SSH, il est possible d'établir un tunnel vers un serveur distant et accéder à des applications qui ne sont pas disponibles sur Internet. Pour ce faire, il faut que le serveur soit configuré avec l'option `AllowTcpForwarding` activée et que le client dispose d'un compte SSH sur ce serveur.

Il est ainsi possible d'accéder à un site web depuis l'extérieur :

```
$ ssh -L 8000:dartagnan.cg.helmo.be:80 p010544@dartagnan.cg.helmo.be
```

Cette commande lance un accès SSH sur le serveur *dartagnan* mais établit, en plus, un tunnel SSH. Cela signifie que lorsqu'on accède à `http://localhost:8000`, la requête est passée dans le tunnel SSH pour être transmise à *dartagnan.cg.helmo.be* sur le port 80 (port web).

<sup>34</sup> Cette remarque n'est pas vraie pour MacOS X. Pour ce dernier, il est nécessaire de démarrer un serveur X particulier comme XQuartz

Cette commande permet donc de voir les sites web publiés par les étudiants et hébergés sur *dartagnan* sans devoir démarrer la connexion VPN. Le firewall de HELMo voit uniquement passer du trafic SSH sans se douter qu'il y a du trafic web à l'intérieur.

Le tunnel reste actif tant que la connexion SSH est lancée. Dès que celle-ci est déconnectée, le tunnel SSH est fermé en même temps.

### 10.3 Le service X2Go

Le service X2Go utilise SSH pour établir une connexion en mode graphique sur le serveur Linux. Le programme client est disponible sur les principales plateformes.

Toutes les informations d'installation et de configuration sont disponibles sur le site web du projet : <http://www.x2go.org>. Le serveur existe sous la forme de *package* pour les distributions RedHat /CentOS 7.

### 10.4 Installation d'applications

Les applications peuvent prendre deux formes distinctes : des ***applications précompilées et prévues pour la distribution utilisée***. Ainsi, les versions CentOS 7 utilisent des *packages RPM* (présents aussi dans les distributions Suse, CentOS, RedHat, Scientific Linux ou encore Fedora). Si nous souhaitons installer un package pour CentOS 7, il faut qu'il soit étiqueté el7 (*enterprise linux 7*) ou CentOS 7 afin d'être compatible. Avec d'autres distributions comme Ubuntu, ce sont des *packages DEB* (initialement prévu pour Debian, mais adopté par toutes les distributions qui sont basées sur celle-ci comme Ubuntu et toutes ses variantes).

La seconde forme pour l'installation d'application est ***de télécharger les sources du programme (en C, C++, ...), de compiler celui-ci sur le système et de l'installer***. Cette manière peut sembler difficile d'un premier abord mais elle a l'avantage d'être indépendante de la distribution Linux présente.

#### 10.4.1 Installation d'une application compilée

Avant de commencer, il convient de bien comprendre comment le système fonctionne : des dépôts contenant les applications compilées sont présents sur Internet. Ces dépôts sont utilisés pour installer de nouvelles applications ou encore maintenir le système à jour. Ainsi, les installations utilisent la commande `yum` :

```
$ yum upgrade
```

*Cette commande vérifie si des packages installés doivent être mis à jour. Si c'est le cas, les nouvelles versions sont téléchargées et installées.*

On peut demander à `yum` d'installer un programme particulier :

```
$ yum install xterm
```

*Cette commande installe le package `xterm` (contenant un terminal graphique standard). Il faut noter que toutes les dépendances sont automatiquement installées.*

```
$ yum search php
```

*Cette commande interroge les dépôts pour obtenir les packages référençant le mot `php` (soit dans le nom, soit dans la description).*

```
$ yum install ./monfichier.rpm
```

Cette commande installe un package RPM dont le fichier a été téléchargé et stocké sur le disque dur. Une vérification des dépendances sera effectuée durant l'installation.

### Les dépôts

Les dépôts utilisables peuvent être nombreux. Par exemple, *Adobe* propose un dépôt pour l'installation du plugin *flash* dans les systèmes EL7 (CentOS et Redhat). Les dépôts peuvent être nombreux sur un serveur, en fonction des programmes souhaités. Il est, cependant, toujours préférable de privilégier les dépôts officiels de la distribution pour l'installation d'un programme. Si celui-ci n'est pas disponible sur le dépôt, on peut alors ajouter un dépôt alternatif proposant le logiciel en question.

Les dépôts installés ont un fichier `.repo` installé dans le dossier `/etc/yum.repos.d`. Sur notre machine virtuelle, les dépôts suivants sont installés :

- CentOS
- EPEL
- Webmin

L'ajout d'un autre dépôt ajoutera un fichier dans ce dossier.

### Le programme YumEx

La commande `yum` est une commande en mode texte. Il existe une version graphique de ce logiciel nommé `yumex` (*Yum Extender*). Celui-ci peut être installé depuis les dépôts :

```
$ yum install yumex  
$ yumex
```

### 10.4.2 Installation depuis les sources

L'autre méthode pour installer une application est de télécharger son code source et de compiler celle-ci directement sur la machine qui est destinée à l'exécuter. Le processus est assez standardisé. Cependant, il est nécessaire de bien s'assurer que toutes les librairies nécessaires à la compilation du programme sont bien présentes (il n'y a pas de gestion *des dépendances quand on compile depuis les sources*).

La première étape est **de télécharger l'application depuis internet**. Le plus souvent, l'application est fournie dans une archive compressée en `.tar.gz` ou `.tar.bz2`. Il faut décompresser cette archive :

```
$ tar xvzf monfichier.tar.gz (par exemple)
```

L'étape suivante consiste à **compiler l'application** :

```
$ cd dossier_application  
$ ./configure  
$ make
```

Une fois l'application compilée, **il faut installer celle-ci sur le système (en root)**:

```
$ make install
```

Une fois l'application installée, elle est le plus souvent présente dans le dossier `/usr/local`.

## 10.5 Exercices

On vous demande de :

1. Changer le mot de passe du compte *root* par votre mot de passe HELMo
2. Ouvrir le port 22 sur votre firewall pfSense
3. Planifier une sauvegarde automatique en utilisant *rsync* de votre dossier */home* vers la machine de votre voisin, en utilisant le compte qu'il vous avait créé (voir leçon précédente)
4. Configurer un tunnel SSH pour accéder au serveur web de votre voisin. Pour ce faire, rediriger le port local 8080 vers sa machine, sur le port 80. Accéder à son site web en utilisant l'adresse `http://127.0.0.1:8080`.
5. En vous basant sur les informations du site <http://www.if-not-true-then-false.com/2010/install-adobe-flash-player-10-on-fedora-centos-red-hat-rhel/>, ajouter le dépôt *Adobe* et installer *flash player* sur votre machine.
6. En vous basant sur les informations présentes ici : <http://wiki.x2go.org/doku.php>, installer le serveur X2Go sur votre machine *CeNoS7 Serveur* et un client sur l'autre. Tenter ensuite une connexion.
  - a. Astuce : lors de la configuration du poste client, remplacer KDE ou GNOME par XFCE.

## Leçon 11 : Le service FTP

### 11.1 Introduction

Le service FTP est un service complémentaire au service Web puisqu'il permet aux utilisateurs de charger et déposer des fichiers sur espace configuré (par exemple des fichiers *html* ou *php*). Il est parfois proposé comme moyen de backup par les entreprises vendant de l'espace web. Il existe de nombreux programmes serveurs FTP sous Linux. Ainsi les plus connus sont : *proftpd*, *vsftpd*, *wu-ftpd*. Sous Windows, d'autres logiciels peuvent remplir ce rôle comme *Microsoft FTP* (intégrant *IIS*) ou *FileZilla FTP Server*.

Il faut cependant être très prudent avec le service FTP afin :

- D'autoriser uniquement les utilisateurs souhaités à se connecter au service
- Limiter les dossiers et fichiers visibles

La distribution *CentOS 7* livre en standard le logiciel *vsftpd* que nous allons découvrir dans cette leçon.

### 11.2 FTP et la sécurité

Le service FTP est un service qu'il faut tenir à l'œil car il peut être utilisé pour accéder ou découvrir des informations sensibles. Ainsi, il est possible de *scripter* des tentatives de connexion au serveur et découvrir des comptes (login et mot de passe) valides. Par conséquent, des mesures additionnelles sont souvent mises en place pour sécuriser celui-ci :

- Autoriser uniquement certains comptes à accéder au service FTP (et systématiquement bannir des logins connus : bin, daemon, root, admin, ...).
- Forcer les utilisateurs à utiliser un mot de passe **fort** (composé d'au moins 10 caractères comprenant majuscules, minuscules, chiffres et caractères spéciaux) et **non issu du dictionnaire** (beaucoup d'utilisateurs choisissent des mots courants dans leur mot de passe).
- Protéger la phase d'authentification par SSL afin d'imposer un échange chiffré du login / mot de passe
- Utiliser des techniques de sécurité avancées pour limiter les tentatives de connexion

### 11.3 FTP Actif et FTP Passif

Le service FTP supporte 2 modes de fonctionnement : le mode **actif** et le mode **passif**. Historiquement, le mode **actif** est celui qui était présent dès le départ, cependant, avec le déploiement du NAT et des firewalls, le mode **passif** s'est vite répandu. Aujourd'hui, c'est le mode de fonctionnement par défaut. Nous allons détailler ces deux modes de fonctionnement.

### 10.3.1 Le mode actif



En mode actif, le client se connecte au serveur FTP sur le port 21 (couramment utilisé à cet usage). Ce canal de communication est utilisé pour transmettre les commandes FTP du client vers le serveur (transmission des requêtes). Lorsque des données doivent être transmises depuis le serveur (liste des fichiers / dossiers, contenu d'un fichier, ...), celui-ci **ouvre une nouvelle connexion vers le client**.

Or le problème se situe ici surtout depuis la généralisation du NAT et des firewalls du côté client. En effet, le firewall (par exemple celui contenu dans la box VDSL) devrait se souvenir de qu'une connexion a été demandée par le client et autoriser la connexion – sur n'importe quel port – depuis ce serveur.

Autant dire que ce mode de fonctionnement n'est pas du tout utilisable aujourd'hui. Il reste cependant parfois présent à l'intérieur des entreprises ou des réseaux internes.

### 11.3.2 Le mode passif



Le mode passif fonctionne un peu différemment. Comme nous pouvons le voir, le client lorsqu'il doit envoyer une commande au serveur, il ouvre un canal de connexion sur le port 21 (couramment utilisé à cet usage). Lorsque le client souhaite transmettre ou recevoir des données depuis le serveur, il ouvre un second canal vers le serveur.

Ainsi, il n'y a plus de connexion du serveur vers le client, ce qui simplifie beaucoup la gestion de la sécurité chez le client. Cependant, nous avons reporté le problème du côté serveur : en effet, le canal d'échange de donnée doit arriver jusqu'au serveur FTP sur n'importe quel port, lui aussi protégé par un firewall.

Afin de sortir de ce problème, la solution proposée est que le serveur FTP peut suggérer au client le port que celui-ci peut utiliser pour ouvrir le canal de transmission des données. Grâce à cet élément de configuration, l'administrateur du firewall côté serveur doit simplement : *ouvrir le port 21 pour le canal de commande et ouvrir une série de ports (range) pour le canal de transmission* (par exemple : les ports TCP entre 15000 et 15500).

#### *Ouverture de ports dans pfSense*

Pour ouvrir des ports dans pfSense, il faut se connecter à l'interface web de configuration du firewall et puis aller dans le menu **Firewall > NAT**.

Nous allons devoir ajouter 2 règles<sup>35</sup> pour permettre la connexion sur le serveur FTP depuis le réseau de l'école :

3. Interface : WAN ; Protocol : TCP ; Destination : WAN address ; Destination port range from : (other) 15000 to (other) 15500 ; Redirect target IP : <ip de votre serveur Linux> ; Redirect target port : 15000 ; Cliquer sur SAVE
4. Interface : WAN ; Protocol : TCP ; Destination : WAN address ; Destination port range from : FTP to FTP ; Redirect target IP : <ip de votre serveur Linux> ; Redirect target port : FTP ; Cliquer sur SAVE

Cliquer sur **Apply Changes**

Le service FTP devrait, quand il sera configuré et activé, être accessible depuis l'extérieur du réseau virtuel.

## 11.4 Configuration du service FTP

La configuration du service *vsftpd* est concentrée dans le fichier */etc/vsftpd/vsftpd.conf*. Ce fichier texte permet de préciser les options souhaitées. Nous allons, dans la suite, pointer quelques options importantes<sup>36</sup> :

- **anonymous\_enable=YES | NO**  
Permet d'autoriser ou interdire la connexion en mode anonyme sur le serveur. Le mode anonyme est le mode de connexion par défaut lorsqu'on entre une URL *ftp://ftp.monsite.be* dans son navigateur (connexion avec l'utilisateur *anonymous* ou *ftp* sans vérification du mot de passe). Cet utilisateur ne doit pas avoir un compte sur le système. Dans CentOS 7, la connexion avec ce compte donne accès au dossier */var/ftp*.
- **local\_enable=YES | NO**  
Permet d'autoriser ou interdire la connexion des utilisateurs existants sur le système (i.e. disposant d'un compte). Si l'authentification réussit, l'utilisateur est automatiquement connecté dans son dossier personnel dans */home/login*.
- **write\_enable=YES | NO**  
Permet d'autoriser ou interdire l'écriture (dépot d'un fichier, création d'un dossier, modification des permissions, ...). Attention, les permissions UNIX (droits, ACLs) doivent également permettre l'opération pour que celle-ci soit autorisée.
- **local\_umask=022    umask par défaut :    local\_umask=077**  
Détermine le masque appliqué sur les permissions. Cette option est utilisée lors du dépôt d'un fichier ou la création d'un dossier afin de savoir quelle permission UNIX définir sur le fichier / dossier déposé. Ainsi, les permissions 777 – 022, soit 755 seront appliquées.

<sup>35</sup> Les ports proposés sont donnés à titre d'exemple (21 et ceux compris entre 15000 et 15500)

<sup>36</sup> En **gras** : la valeur par défaut si l'option est absente du fichier de configuration. En **rouge**, la valeur présente dans le fichier *vsftpd.conf* à l'installation.

- `anon_upload_enable=YES | NO`  
`anon_mkdir_write_enable=YES | NO`  
 Détermine si un utilisateur non authentifié (en mode anonyme) peut déposer des fichiers ou créer des dossiers. **Ces deux options devraient toujours être désactivées !**
- `dirmessage_enable=YES | NO`  
 Permet d'activer ou non la prise en charge des fichiers .message présents dans les dossiers. Le contenu du fichier texte est alors envoyé au client FTP qui peut décider de l'afficher.
- `connect_from_port_20=YES | NO`  
 Impose que la commande FTP PORT émane du port 20 du serveur. Ce comportement est complètement obsolète mais certains clients primitifs peuvent l'exiger.
- `ftpd_banner=GodSwila FTP Serveur version 2025`  
 Permet de changer l'annonce par défaut du serveur. Cette option est utile pour cacher le type de serveur et la version qui est installée.
- `chroot_local_user=YES | NO`  
`chroot_list_enable=YES | NO`  
`chroot_list_file=/etc/vsftpd/chroot_list`  
`allow_writeable_chroot=YES | NO`  
 L'option `chroot_local_user` permet d'emprisonner l'utilisateur qui se connecte dans son dossier personnel. En conséquence, il ne peut pas en sortir et naviguer sur tous les dossiers accessibles du système. Cette option devrait toujours être activée. Afin d'augmenter la flexibilité dans la configuration du serveur FTP, il est possible de déterminer modifier quelque peu ce comportement :

<code>chroot_local_user</code>	<code>chroot_list_enable</code>	Explication
YES	NO	Tous les utilisateurs sont emprisonnés dans leurs dossiers personnels
YES	YES	Tous les utilisateurs sont emprisonnés dans leurs dossiers personnels excepté ceux listés dans le fichier pointé par <code>chroot_list_file</code>
NO	YES	Seuls les utilisateurs listés dans le fichier pointé par <code>chroot_list_file</code> sont emprisonnés dans leurs dossiers personnels.

Lorsqu'on active l'option `allow_writeable_chroot`, `vsftpd` ne vérifie pas si l'utilisateur à le droit d'écrire dans son dossier personnel. A activer en cas d'erreur du type « *refusing to run with writable root inside chroot()* ».

- `listen=YES | NO`  
 Permet de déterminer si le serveur FTP fonctionne en mode *standalone*. Le mode *standalone* devrait toujours être utilisé (soit via `listen=YES` ou `listen_ipv6=YES`). Cette option démarre l'écoute uniquement sur les adresses IPv4 de la machine. **Elle ne peut être utilisée en même temps que `listen_ipv6`.**

- `listen_ipv6=YES | NO`  
Permet de déterminer si le serveur FTP fonctionne en mode *standalone*. Le mode *standalone* devrait toujours être utilisé (soit via `listen_ipv6=YES` ou `listen =YES`). Cette option démarre l'écoute sur les adresses IPv4 et IPv6 de la machine (pour autant que l'adresse d'écoute ne soit pas modifiée). **Elle ne peut être utilisée en même temps que listen.**
- `pam_service_name=vsftpd`  
Cette option détermine le nom sous lequel les modules PAM (pluggable authentication modules) reconnaissent le serveur FTP. Grâce à ceux-ci, il est possible de diversifier les méthodes d'authentification (comme utilisé des mots de passe à usage unique, ...).
- `userlist_enable=YES | NO`  
`userlist_deny=YES | NO`  
`userlist_file=/etc/vsftpd/user_list`  
Permet de contrôler les utilisateurs autorisés à se connecter au serveur. Ainsi, nous avons les combinaisons possibles pour ces options :

<code>userlist_enable</code>	<code>userlist_deny</code>	Explication
YES	YES	Tous les utilisateurs connus sont autorisés à se connecter sur le serveur excepté ceux listés dans le fichier pointé par <code>userlist_file</code> .
YES	NO	Seuls les utilisateurs connus et listés dans le fichier pointé par <code>userlist_list_file</code> peuvent se connecter sur le serveur.
NO	-	Tous les utilisateurs connus peuvent se connecter sur le serveur FTP.

- `tcp_wrappers=YES | NO`  
Permet de vérifier que les connexions sont vérifiées en tenant compte des fichiers `/etc/hosts.allow` et `/etc/hosts.deny`. Il est possible d'autoriser par ces fichiers les connexions par adresse IP, préfixe IP, ... Cette méthode de sécurisation est souvent trop limitée par rapport aux possibilités des firewalls.
- `pasv_min_port=borne1`  
`pasv_max_port=borne2`  
`pasv_promiscuous=YES | NO`  
Permet de déterminer, pour le mode passif, les ports qui sont suggérés par le serveur FTP au client pour le transfert des données. Il est nécessaire d'*ouvrir tous les ports compris entre ces 2 bornes* au niveau du firewall. L'option `pasv_promiscuous` doit être activée pour permettre la connexion derrière un système NAT
- `ssl_enable=YES | NO`  
`rsa_cert_file=/path/to/certificate.pem`  
`rsa_private_key_file=/path/to/private_key.pem`  
`ssl_ciphers=HIGH`  
Ces options permettent d'activer le support SSL dans le serveur FTP. Pour ce faire, il faut placer, dans un seul fichier, le certificat du serveur et tous les certificats intermédiaires – par exemple dans `/etc/pki/tls/certs/vsftpd-allcerts.pem` – et faire pointer `rsa_cert_file` vers ce fichier. Il faut également placer la clé privée dans un fichier – par exemple `/etc/pki/tls/private/vsftpd-privatekey.pem` – et faire pointer

`rsa_private_key_file` vers ce fichier. Enfin, il faut également déterminer les algorithmes de chiffrement autorisés, `HIGH` désigne les algorithmes de sécurité élevés.

Une fois la configuration de `vsftpd` terminée, le service peut être démarré au moyen de la commande :

```
$ systemctl start vsftpd
```

En cas d'erreur, il est possible d'obtenir des informations complémentaires via l'argument `status` :

```
$ systemctl status vsftpd
```

Il est parfois plus simple de lancer le service `vsftpd` à la main et voir si un message d'erreur survient :

```
$ vsftpd /etc/vsftpd/vsftpd.conf
```

Quand il n'y a plus d'erreur, on peut alors tuer le processus `vsftpd` lancé à la main et redémarrer le service `vsftpd` normalement :

```
$ killall vsftpd  
$ systemctl start vsftpd
```

## 11.5 Le client FTP

Suivant les systèmes d'exploitation clients utilisés, il existe bon nombre de programme client. Ainsi *FileZilla* est un client FTP connus (et disponible sous Windows, Linux et Mac OS X). Il y a également *FireFTP* qui est une extension pour Mozilla Firefox qui est également un client FTP intéressant et courant.

En ligne de commande, des clients existent également : le programme `ftp` est probablement le client le plus connu (une version très limitée existe aussi sous Windows). Nous découvrirons également `lftp` qui permet de scripter les échanges FTP facilement.

Si nécessaire, l'installation de ces outils se fait au moyen de `yum` :

```
$ yum install ftp lftp
```

### 11.5.1 Le programme FTP en ligne de commande

```
[root@localhost pam.d]# ftp  
ftp> open ftp.belnet.be  
Trying 193.190.67.98...  
Connected to ftp.belnet.be (193.190.67.98).  
220-Welcome to the Belnet public FTP server ftp.belnet.be !
```

```
This server is located in Brussels, Belgium and operated by Belnet, ...
```

```
220 ProFTPD 1.3.4a Server (Belnet FTP Server) [193.190.67.98]  
Name (ftp.belnet.be:root): ftp  
331 Anonymous login ok, send your complete email address as your password  
Password: test@test.com  
230 Anonymous access granted, restrictions apply  
Remote system type is UNIX.  
Using binary mode to transfer files.
```

```
ftp> ls
227 Entering Passive Mode (193,190,67,98,141,42).
150 Opening ASCII mode data connection for file list
lrw-r--r-- 1 ftp      ftp          28 Nov 14 2012 debian ->
mirror/ftp.debian.org/debian
lrw-r--r-- 1 ftp      ftp          31 Nov 14 2012 debian-cd ->
mirror/ftp.debian.org/debian-cd
drwxr-xr-x 89 ftp     ftp          4096 Feb 10 08:29 mirror
lrw-r--r-- 1 ftp      ftp          7 Nov 10 2012 mirrors -> mirror/
lrw-r--r-- 1 ftp      ftp          6 Nov 14 2012 pub -> mirror
226 Transfer complete
ftp> ?
ftp> quit
221 Goodbye.
```

Les commandes FTP autorisées sont nombreuses. Pour les connaître, il faut entrer ? à l'invite FTP.

Parmi les commandes courantes, citons :

- `passive` qui permet d'activer / désactiver le mode passif
- `get` qui permet de télécharger un fichier du serveur et de le placer dans le dossier courant
- `put` qui permet de déposer un fichier sur le serveur à partir du dossier courant
- `prompt` qui permet d'activer / désactiver le mode interactif. Par défaut, ce mode est activé et demande une confirmation lors de l'utilisation des commandes *multiples mget et mput*.
- `mget` qui permet de recevoir des fichiers dans le dossier courant. Si le mode interactif est activé, une confirmation sera demandée pour chaque fichier.
- `mput` qui permet de déposer des fichiers sur le serveur à partir du dossier courant. Si le mode interactif est activé, une confirmation sera demandée pour chaque fichier.
- `chmod` qui permet de modifier les permissions d'un fichier ou d'un dossier
- `cd` qui permet de changer de répertoire sur le serveur
- `lcd` qui permet de changer le dossier courant (sur le poste client). `lcd` sans paramètre affiche le dossier local dans lequel on se trouve.
- `pwd` qui affiche le répertoire distant dans lequel on se trouve
- `exit` qui permet de fermer la connexion FTP et quitter le client.

### 11.5.2 Le programme lftp

Le programme *lftp* dispose d'une option intéressante : il est possible de scripter les commandes FTP à envoyer au serveur. Ainsi, le programme peut être utilisé pour planifier et automatiser une sauvegarde par FTP sur un serveur distant.

Ainsi, on peut, par exemple, transférer un fichier automatiquement avec la commande :

```
$ lftp monlogin:monpass@NomOuIPserveur -e "cd dossier; put
/home/swila/monfichier; exit;"
```

Cette commande se connectera au serveur `NomOuIPserveur` (pour lequel on peut mentionner son nom complet ou son adresse IP) et exécutera, successivement, les commandes `cd`, `put` et `exit` avec les paramètres indiqués.

Il est évidemment possible d'intégrer cette commande dans un script PERL.

## 11.6 Exercices

On vous demande de :

1. Créer un utilisateur *backup*. Définir son mot de passe.
2. Configurer votre serveur FTP afin de :
  - a. Ne pas autoriser l'accès anonyme
  - b. Permettre à l'utilisateur *swila* et *backup* de se connecter au serveur FTP
  - c. Permettre à *votre voisin* (pour lequel un compte a été créé lors d'une leçon précédente) de se connecter au serveur FTP
  - d. Emprisonnez tous les utilisateurs, excepté l'utilisateur *backup*, dans leurs dossiers personnels
  - e. Activer le mode passif avec les ports compris entre 45000 et 45500
  - f. Vérifier votre configuration en vous connectant à votre serveur FTP en local
3. Configurer pfSense pour ouvrir les ports nécessaires au fonctionnement FTP. Vérifier votre configuration en utilisant le poste Windows pour vous connecter (et en utilisant l'adresse IP externe en 192.168.[128-143].x)
4. Créer un script PERL :
  - a. qui compresse, sous la forme d'une archive ZIP, le contenu du dossier /etc
  - b. qui transfert le fichier ZIP créé sur la machine de votre voisin, en utilisant le compte qu'il vous a créé

Planifiez l'exécution du script PERL tous les semaines, durant les séances de laboratoire (à 10h ou à 15h par exemple).

## Leçon 12 : Le service mail

### 12.1 Introduction

Le service mail existe depuis les débuts d'internet. C'est probablement un des premiers services qui a vu le jour. Bien qu'il fût, par moment et encore régulièrement, décrié, il est toujours indispensable aujourd'hui.

Il est indispensable aux entreprises, premier moyen de communication, il souffre également de son plus grand démon : le spam.

Bien que nous allons expliquer le fonctionnement et la configuration du service mail ci-après, il faut bien être conscient qu'expliquer toutes les techniques possibles et déployées pour combattre le spam pourrait faire l'objet d'un cours séparé complet. J'ai donc choisi une présentation plus simple et plus claire, en listant à la fin de celle-ci les techniques indispensables qu'il faut déployer pour protéger son serveur mail.

### 12.2 Le mail interne vs externe

Beaucoup d'entreprises séparent le mail interne du mail externe. Le mail interne est davantage utilisé pour le fonctionnement de l'entreprise, il doit être entouré d'outils comme les agendas, les carnets d'adresse partagés, ... alors que le mail externe est plutôt utilisé pour la communication avec les clients et les fournisseurs.

Ces deux objectifs séparés ont très souvent conduit à séparer les serveurs mails utilisés. Ainsi, pour le mail interne, on trouve davantage des *serveurs collaboratifs* comme *Microsoft Exchange Server* (dans le monde Microsoft), *Zimbra* (dans le monde Unix), ou bien d'autres plateformes (OBM, SOgo, Zarafa, ...) intégrant bon nombre d'outils à côté du courrier électronique. A l'inverse, le mail externe est presque exclusivement dévolu à des serveurs Unix et des logiciels comme *sendmail*, *postfix*, *qmail*, *exim*, ...

Certains programmes comme *qmail* sont utilisés par de grands noms comme Yahoo pour le service Yahoo Mail. D'autres comme *sendmail* sont présents sur tous les systèmes UNIX. C'est ce programme serveur que nous allons découvrir dans cette leçon car il est un des plus anciens et sa présence le rend, probablement, incontournable.

### 12.3 Rappel sur le fonctionnement du mail

Pour rappel, le service mail utilise le protocole SMTP. Ce protocole est associé au port TCP 25 et il sert à l'envoi d'un mail. La réception d'un mail est réalisée par le protocole POP3 (port TCP 110) ou IMAP (port TCP 143).

Pour des raisons de sécurité, seul le serveur configuré pour votre réseau peut être utilisé pour envoyer votre mail<sup>37</sup>. Ainsi, si vous êtes client chez *Proximus*, vous devez utiliser leurs serveurs pour envoyer vos mails.

Cet élément est très important pour éviter les serveurs mails configurés en *open-relay*, c'est-à-dire des serveurs qui relaient des mails de n'importe qui. Ils peuvent alors servir massivement à transmettre du SPAM.

## 12.4 Configuration de sendmail

La configuration de *sendmail* est localisée à deux endroits distincts :

1. Le fichier /etc/aliases qui reprend *les alias* pour les mails
2. Le dossier /etc/mail/ qui reprend la configuration du serveur mail

Nous allons aborder ces deux éléments.

### 12.4.1 Les alias

Un *alias* est un raccourci pour une boîte mail. On peut configurer des *alias* pour des usages divers : changer le nom d'une boîte mail, ajouter nom supplémentaire à la boîte, créer des adresses de groupes qui transmettent le mail à plusieurs destinataires, ...

Ces *alias* sont concentrés dans le fichier texte /etc/aliases. Chaque ligne définit un *alias* et est structurée simplement : on commence par indiquer *le nom de l'alias (son identifiant, ce que l'utilisateur devra entrer)*, on continue avec « : » et à droite, on indique le (ou les) destinataires qui recevront ce courrier.

Ainsi, nous pourrions définir les *alias* suivants :

```
maitre.swinnen: p010544  
prof-b3: p010544, d.bayers@helmo.be, v.reip@helmo.be, c.mathy@helmo.be
```

La modification du fichier /etc/aliases doit être suivie du lancement de la commande :

```
$ newaliases
```

Cette commande permet la prise en compte des *alias* présents.

Comme nous pouvons le voir, il est possible pour un *alias*, de mélanger *des comptes locaux* (comme *p010544*) et des adresses mails. Ainsi l'envoi d'un mail à destination de *prof-b3@mondomaine.be* (pour autant qu'une entrée *MX* dans le DNS mentionne que les mails de ce domaine sont gérés par le serveur) sera transmis à 4 personnes différentes.

L'envoi d'un mail à *maitre.swinnen@mondomaine.be* placera le mail dans la boîte mail de l'utilisateur local *p010544*.

---

<sup>37</sup> Cette remarque est *partiellement vraie* seulement puisque grâce aux extensions d'authentification et de cryptage, il est possible d'envoyer des mails depuis des réseaux "étrangers". Ainsi, le serveur mail de HELMo autorise l'envoi de mail depuis n'importe où si le client s'authentifie au préalable. Par contre, sans authentification, seuls les ordinateurs connectés au réseau de HELMo peuvent utiliser le serveur pour envoyer leur mail.

### 12.4.2 Configuration du serveur

La configuration du serveur mail est concentrée dans le dossier `/etc/mail`. Nous allons, dans la suite, détailler les fichiers de configuration et les options qu'il faut configurer.

Notons au passage que les boîtes aux lettres des utilisateurs sont, par défaut, stockées dans le dossier `/var/spool/mail/<login>`

#### Etape 1 : le fichier access

Le premier fichier sur lequel nous allons nous concentrer est le fichier `/etc/mail/access`. Ce fichier détermine quelles machines peuvent utiliser ce serveur comme serveur d'envoi. Ce fichier peut également être utilisé pour interdire certaines adresses mails connues de spameurs.

Par défaut, il contient les entrées suivantes :

Connect:localhost.localdomain	RELAY
Connect:localhost	RELAY
Connect:127.0.0.1	RELAY

Ces entrées mentionnent que seul l'ordinateur local (`localhost` ou l'IP `127.0.0.1`) peut utiliser le serveur pour relayer des mails. On peut mentionner également des sous-réseaux comme suit :

Connect: 192.168.128	RELAY
----------------------	-------

Ainsi, nous venons d'autoriser toutes les machines dont l'adresse IP se trouve dans ce sous-réseau (`192.168.128.x`) à utiliser ce serveur pour transmettre des mails vers l'extérieur.

Nous pouvons également mentionner des adresses mails que nous souhaitons proscrire :

1.swinnen@helmo.be	REJECT
--------------------	--------

Ca peut-être une mesure simple pour combattre les mails publicitaires.

#### Etape 2 : Le fichier local-host-names

Le fichier `/etc/mail/local-host-names` est un autre fichier très important dans la configuration du service mail. En effet, grâce à ce fichier, le serveur mail peut déterminer **si le mail est arrivé à destination**.

**Comment le serveur peut-il déterminer si le mail est arrivé à destination ?** En fait, il doit analyser **la partie droite de l'adresse mail** (derrière le symbole « @ »). Il extrait donc le nom de domaine qui est mentionné et va vérifier si :

- Il s'agit du nom de la machine (le *hostname*) ;
- Le nom apparaît dans le fichier `local-host-names` ;

Si c'est le cas, le mail est considéré comme arrivé à destination par le serveur et il va alors déterminer dans quelle boîte mail il doit placer ce courrier.

Ainsi pour le serveur mail de HELMo, nous trouvons les informations suivantes dans ce fichier :

helmo.be  
helmo.eu  
student.helmo.be  
mx.helmo.be  
smtp.helmo.be  
mail.helmo.be  
relay.helmo.be

salto.helmo.be  
hemes.be  
isell.be

Ainsi, lorsqu'un mail arrive avec, comme nom DNS derrière le symbole « @ » l'un de ces domaines, le serveur mail sait qu'il doit délivrer ce courrier localement. Notons au passage que l'on mentionne dans ce fichier aussi bien des noms de domaine (comme helmo.be, helmo.eu, student.helmo.be par exemple) que des noms de machines (comme smtp.helmo.be ou relay.helmo.be qui sont tous deux des noms différents pour le serveur mail).

### ***Etape 3 : Le fichier relay-domains***

Le fichier `/etc/mail/relay-domains` est nécessaire **sur le serveur mail secondaire (ou backup)**. Comme annoncé précédemment, le service mail étant assez critique, il est courant de proposer plusieurs serveurs pour traiter le courrier d'un domaine donné. Cependant, comme les boîtes mails sont stockées sur un seul serveur<sup>38</sup>, nous devons configurer le serveur *secondaire* comme pouvant relayer certains mails (et donc *mettre en file d'attente*) vers le serveur principal.

Cette méthode est intéressante. Ainsi, lorsque le serveur principal est déconnecté, le serveur mail *secondaire* prend le relai et stocke, temporairement, les mails qui arrivent. Dès que le serveur principal est à nouveau en ligne, le serveur secondaire lui transmet les mails qu'il a reçus. Ces mails peuvent alors être placés dans les boîtes aux lettres des utilisateurs concernés.

Et donc l'objet du fichier `/etc/mail/relay-domains` est de lister les domaines qui peuvent être acceptés par le serveur mail *secondaire*. Ainsi, le serveur mail *secondaire* de HELMo, `smtp2.helmo.be`, mentionne les informations suivantes dans ce fichier :

helmo.be  
helmo.eu  
isell.be  
hemes.be

Lorsqu'on mentionne `helmo.be`, on autorise le relai pour tous les domaines et sous-domaines de HELMo (comme `student.helmo.be` ou encore `salto.helmo.be`).

### ***Etape 4 : Le fichier virtusertable***

Le fichier `/etc/mail/virtusertable` est également un fichier intéressant. En effet, sans ce fichier, c'est en fonction du **login de l'utilisateur** que les mails sont délivrés. Or, il n'est pas souhaitable de publier les logins des utilisateurs à l'extérieur. De plus, pour avoir une grande latitude, ce fichier est très intéressant.

Grâce à ce fichier, nous pouvons faire correspondre une adresse mail à un compte utilisateur (ou une autre adresse mail). **Attention !** Il n'est pas possible, dans ce fichier, de faire correspondre à une adresse mail plusieurs destinataires.

Exemple de contenu pour le fichier `/etc/mail/virtusertable`:

---

<sup>38</sup> Ceci n'est pas nécessairement vrai. Il est très courant de stocker les boîtes mails sur un stockage réseau en haute disponibilité (comme un SAN) et celui-ci est alors accessible sur plusieurs serveurs. C'est d'autant plus vrai si l'entreprise doit gérer beaucoup de mails et donc met en place plusieurs serveurs mails fonctionnant en parallèles. Cependant, dans notre leçon, nous ne considérerons que le cas d'un serveur *principal* avec les boîtes aux lettres des utilisateurs et un serveur secondaire pouvant relayer les mails vers le serveur principal.

l.swinnen@helmo.be	p010544
l.swinnen@helmo.eu	p010544
godswila@helmo.be	l.swinnen@helmo.be

### Attention ! Le séparateur est bien une (ou plusieurs) tabulation(s).

Comme nous pouvons le voir, les entrées sont **des adresses mails complètes**. Il est donc possible d'inclure des domaines comme helmo.be et helmo.eu par exemple. Il est également possible de mentionner une adresse mail (qui ne doit pas être nécessairement dans les domaines gérés par ce serveur).

### *Etape 5 : Le fichier sendmail.mc*

La dernière étape consiste à modifier ou adapter la configuration du serveur mail via le fichier /etc/mail/sendmail.mc. Ce fichier est un peu particulier car il contient des *commandes de configuration*.

Ces *commandes de configuration* seront ensuite utilisées pour générer le fichier /etc/mail/sendmail.cf qui décrit exactement le comportement attendu du service *sendmail*. Cependant, la compréhension et l'analyse du fichier sendmail.cf est particulièrement éprouvante. C'est pourquoi les *commandes de configuration* sont bien plus simples lorsqu'il s'agit de configurer *sendmail*.

Dans le fichier sendmail.mc, les **lignes commençant** par `dnl` sont des lignes de commentaires.

Voici un certain nombre d'options importantes :

Option	Explication
<code>define(`confSMTP_LOGIN_MSG', `\$j Sendmail; \$b')dnl</code>	Cette option détermine les informations affichées par <i>sendmail</i> lors de la connexion. Afin de ne pas annoncer la version, il est intéressant de modifier cette option.
<code>define(`SMART_HOST', 'smtp.your.provider')dnl</code>	Cette option permet de transmettre les mails sortant (ceux qui ne sont pas arrivés à destination) à un autre serveur mail qui se chargera de l'expédition.  Cette option est nécessaire parfois à cause des mesures de sécurité de certains ISP (comme le blocage des ports 25 sur les réseaux DSL).
<code>define(`confAUTH_OPTIONS', `A p')dnl  TRUST_AUTH_MECH(`EXTERNAL DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl  define(`confAUTH_MECHANISMS', 'EXTERNAL GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl</code>	L'option confAUTH_OPTIONS contrôle si l'authentification est activée sur le serveur SMTP. Par défaut, aucune authentification n'est nécessaire.  Les options TRUST_AUTH_MECH et confAUTH_MECHANISMS sont nécessaires pour indiquer au serveur mail comment cette authentification peut avoir lieu.
<code>define(`confCACERT_PATH', '/etc/pki/tls/certs')dnl  define(`confCACERT', '/etc/pki/tls/certs/ca-</code>	Ces options actives SSL au niveau du serveur mail. L'activation de SSL est nécessaire si l'authentification utilise le login et le mot de passe de l'utilisateur. Ainsi ces informations sont

<code>bundle.crt')dn1</code>	transmises de manière chiffrée au serveur.
<code>define(`confSERVER_CERT', `/etc/pki/tls/certs/sendmail.pem')dn1</code>	Il faut indiquer l'emplacement des certificats (le chemin ou le fichier contenant les autorités racines, et le chemin vers le certificat serveur) et la clé privée.
<code>define(`confSERVER_KEY', `/etc/pki/tls/certs/sendmail.pem')dn1</code>	Cette option est activée et la mention de l'adresse <code>Addr=127.0.0.1</code> empêche le serveur mail d'être disponible sur le réseau. En effet, par défaut, seul <code>localhost</code> peut atteindre celui-ci.
<code>DAEMON_OPTIONS(`Port=smtp, Addr=127.0.0.1, Name=MTA')dn1</code>	Pour que le serveur puisse être utilisé, il est nécessaire de retirer la mention de l'adresse.

### Etape 6 : Activation de la configuration

Comme nous l'avant découvert à l'étape précédente, le fichier `sendmail.mc` contient *des commandes de configuration*. Une fois ces commandes correctement configurées, il faut activer la configuration en construisant le fichier `sendmail.cf`

Pour ce faire, il suffit de lancer, à l'invite de commande :

```
$ make -C /etc/mail
```

Cette instruction `make` va construire le fichier `sendmail.cf` conforme aux *commandes de configuration*.

Pour redémarrer le service mail, il faut passer par la commande `systemctl` :

```
$ systemctl restart sendmail
```

**ATTENTION ! CETTE CONFIGURATION DE BASE N'INCLUT AUCUNE PROTECTION CONTRE LES SPAM.  
IL EST IMPORTANT, DANS UN ENVIRONNEMENT DE PRODUCTION, DE PRENDRE DES DISPOSITIONS  
POUR ERADIQUER LES COURRIERS INDESIRABLES.**

### 12.5 Les techniques anti-spam

Les techniques anti-spam sont nombreuses et très différentes. Ainsi, bon nombre d'entreprises ont fait de ces techniques des solutions vendues et, parfois, très chères. Ainsi, *IronPort* est une solution complète pour éradiquer le spam. Sans débourser un euro, il est possible d'adoindre des mécanismes anti-spam à son serveur mail `sendmail`. Nous allons dans ce paragraphe introduire brièvement certaines techniques :

Outils	Description
<b>MailScanner</b>	MailScanner est un ensemble logiciel complet utilisé pour gérer les mails et les spams en se basant sur <code>sendmail</code> . Il permet de combiner bon nombre de techniques connues comme des antivirus, Spam Assassin (détecteur probabiliste), des listes RBL (liste de serveurs connus pour transmettre des spams), des outils comme <i>Pyzor</i> ou <i>DCC</i> (qui utilisent des réseaux <i>peer-to-peer</i> pour s'échanger des empreintes de SPAM).  Les techniques ici sont donc nombreuses et pas toujours évidentes à appréhender. Cependant, la stabilité et l'intégration de cet outil est

	vraiment un élément important.
	A titre d'exemple, le serveur mail de HELMo utilise MailScanner.
<b>greylist-milter</b>	<p>Ajout à <i>sendmail</i> pour implémenter une <i>liste grise</i>. Alors que les <i>listes blanches</i> (toujours autorisé) et les <i>listes noires</i> (tout est rejeté) sont bien connues, l'instauration de la <i>liste grise</i> est un peu particulière.</p> <p>En fait, la norme SMTP prévoit que la boîte aux lettres d'un utilisateur <i>peut être inaccessible</i> momentanément. Dans ce cas, le serveur mail est tenu de renvoyer le mail plus tard, jusqu'à ce qu'il soit accepté. Or, les outils anti-spam n'ont bien souvent pas la capacité de retransmettre le spam si le serveur répond qu'il faut <i>revenir plus tard</i>. De plus, le <i>délai</i> peut être déterminé en fonction du serveur qui transmet la demande (on peut, pour les pays réputés <i>plus spammeurs</i>, allonger le délai d'attente).</p> <p>C'est donc une technique assez efficace qui, ajoutée aux autres, permettent de diminuer significativement les spams.</p>
<b>Liste RBL</b>	<p>Il est possible d'indiquer des listes spécifiques, à interroger, lorsqu'un mail est reçu d'un serveur. L'intérêt est de déterminer si le serveur mail est fiable et s'il est autorisé à envoyer un mail.</p> <p>Par exemple, toutes les connexions utilisant des adresses IP dynamiques sont automatiquement répertoriées comme n'étant pas autorisée à envoyer un mail. L'utilisation de ces listes permet ainsi de diminuer les spams également.</p> <p>Les listes intéressantes sont :</p> <ul style="list-style-type: none"> <li>• dul.dnsbl.sorbs.net</li> <li>• sbl.spamhaus.org</li> </ul>
<b>DCC / Pyzor</b>	DCC et Pyzor sont deux outils plutôt surprenant car ils échangent, en <i>peer-to-peer</i> , des empreintes de mail spam. Ainsi, un mail transmis massivement peut ainsi se faire repérer grâce aux échanges réalisés par ces outils.

## 12.6 POP3 et IMAP

Comme mentionné en introduction, SMTP est utilisé pour l'envoi du mail alors que pour réceptionner les mails, il faut utiliser POP3 ou IMAP. Le choix entre les deux services est souvent une question de politique d'entreprise. IMAP stocke les mails sur les serveurs de l'entreprise alors que POP3 télécharge les mails sur les postes clients.

A l'instar du protocole SMTP, il existe de nombreux programmes permettant de gérer la réception des mails par POP3 ou IMAP. Citons, parmi les programmes courant *courrier*, *cyrus*, ou encore *dovecot*. C'est ce dernier que nous allons aborder dans cette leçon.

Le programme *dovecot* implémente aussi bien les protocoles POP3 qu'IMAP. On peut même décider d'activer les deux protocoles en même temps (ce qui ne pose aucun problème puisque des ports TCP différents sont utilisés pour chaque service). Nous allons, dans la suite de ce paragraphe, analyser les options de configuration de *dovecot*.

La configuration de *dovecot* est concentrée en 2 endroits : le fichier `/etc/dovecot/dovecot.conf` et le dossier `/etc/dovecot/conf.d`.

### 12.6.1 Le fichier `dovecot.conf`

Dans le fichier `/etc/dovecot/dovecot.conf`, il y a des éléments de configuration qu'il est intéressant de pointer :

Option	Explication
<b>protocols</b>	Permet d'activer les protocoles souhaités. Les options valides sont <code>imap</code> , <code>pop3</code> et <code>lmtp</code> .  Par défaut, les trois protocoles sont actifs.
<b>listen</b>	Permet de déterminer sur quelle adresse IP le serveur écoute. Par défaut, il écoute sur toutes les adresses IP configurées sur la machine.
<b>login_greeting</b>	Permet de changer l'information retournée par le serveur lors de la connexion. Cela permet de brouiller un peu les pistes en n'annonçant pas que <i>dovecot</i> est utilisé. <code>login_greeting = oui allo ?</code>

### 12.6.2 Le dossier `conf.d`

Dans le dossier `/etc/dovecot/conf.d`, tous les fichiers `.conf` sont pris en compte par le serveur comme des ajouts de configuration. Nous n'allons pas passer en revue tous les fichiers mais juste certaines options contenues dans certains d'entre-eux.

Fichier	Option	Explication
<b>10-auth.conf</b>	<code>disable_plaintext_auth</code>	Désactive l'authentification en clair à moins que SSL soit utilisé. Cette option doit être placée à <code>no</code> si l'on souhaite autoriser l'authentification sans certificat.
<b>10-auth.conf</b>	<code>auth_username_format</code>	Cette option permet d'indiquer si le nom d'utilisateur doit être transformé avant de tenter une authentification. Ainsi, mettre l'option <code>%Lu</code> assure que le login sera transformé en minuscule au préalable.
<b>10-mail.conf</b>	<code>mail_location</code>	Indique l'emplacement de la boîte mail de l'utilisateur.  <code>mail_location = mbox:~/mail:INBOX=/var/spool/mail/%u</code>  Cette valeur mentionne que la boîte aux lettres se trouve dans le dossier personnel de l'utilisateur et dans le dossier: <code>/var/spool/mail/login</code>
<b>10-ssl.conf</b>	toutes	Ce fichier contient la configuration pour l'activation de SSL sur IMAP et POP3. Ces options sont intéressantes pour sécuriser l'authentification.

		Cette sécurité est activée par défaut. Cependant, sans certificat valide, la configuration des postes clients risque de poser quelques soucis.
--	--	--

Voici un récapitulatif des ports utilisés suivants les protocoles activés :

Protocole	SSL / Certificat requis ?	Port TCP
<b>SMTP (envoi de courrier) sendmail</b>	Sans SSL	25
<b>SMTPS (envoi de courrier) sendmail</b>	Avec SSL+Certificat	465
<b>POP3 (téléchargement mail) dovecot</b>	Sans SSL	110
<b>POP3S (téléchargement mail) dovecot</b>	Avec SSL+Certificat	995
<b>IMAP (consultation mail depuis le serveur) dovecot</b>	Sans SSL	143
<b>IMAP (consultation mail depuis le serveur) dovecot</b>	Avec SSL+Certificat	993

### 12.6.3 Démarrer le service dovecot

Le démarrage du service *dovecot* se fait grâce à la commande `systemctl`:

```
$ systemctl start dovecot
```

## 12.7 Installation d'un Webmail

Il existe de nombreux programmes *Webmail* open-source. Parmi les plus courants, citons *IMP/Horde*<sup>39</sup>, *squirrelmail*<sup>40</sup> ou encore *RoundCube*<sup>41</sup> (qui est installé notamment à HELMo). Il en existe bien d'autres, notamment tous les logiciels collaboratifs cités en introduction de cette leçon.

Bon nombre de ces sites sont développés en PHP + MySQL. Pour les utiliser, il faut donc activer le serveur web Apache, s'assurer que PHP est installé et fonctionnel et, finalement, configurer une base de données pour que ces sites puissent l'utiliser.

Dans cette leçon, nous ne verrons pas l'installation étape par étape de ces programmes Webmail. En effet, l'installation est simple et est complètement décrite sur le site du fournisseur. C'est ainsi que l'installation et le programme RoundCube peuvent être téléchargé depuis le site web <http://www.roundcube.net>.

<sup>39</sup> <http://www.horde.org/apps/imp/>

<sup>40</sup> <http://www.squirrelmail.org/>

<sup>41</sup> <http://www.roundcube.net>

## 12.8 Exercices

On vous demande de :

1. Configurer votre serveur mail (sans SSL, ni authentification, ni techniques antispam) pour le domaine configuré (voir leçon DNS)
2. Installer un serveur POP3/IMAP sur votre machine et configurer celui-ci
3. Configurer Thunderbird installé sur le poste Windows pour interroger votre serveur mail
4. Capturer le trafic sur votre serveur mail pour trouver le mot de passe échangé

Testez votre configuration en envoyant des mails à vos comptes locaux (en effet, la configuration ne permettra pas d'envoyer des mails vers Internet).

5. Installer Roundcube Webmail sur votre serveur pour consulter, par le web, les mails envoyés et reçus.

## Leçon 13 : Le firewall

### 13.1 Introduction

Le firewall sous Linux est un élément réseau important. Nous l'avons déjà un peu utilisé lorsqu'on a dû activer le NAT lors d'une leçon précédente. Le firewall Linux, nommé `iptables`, est un outil puissant et complexe permettant de gérer et protéger le réseau convenablement.

Devant la complexité de ce dernier, la dernière version de CentOS GNU/Linux propose une gestion alternative, nommée `firewalld` (qui se base sur `iptables` pour fonctionner). Ce firewall plus simple est prévu pour protéger un serveur sur internet : tout le trafic sortant est autorisé, le trafic entrant est filtré et aucun trafic « de passage » n'est prévu.

Devant cette limitation (nous ne pourrions utiliser `firewalld` sur notre machine routeur), nous allons dans cette leçon détailler le firewall traditionnel `iptables`.

Pour éviter tout conflit, il est déconseillé d'utiliser conjointement `firewalld` et `iptables`. Sur un poste CentOS 7, cela revient à désactiver `firewalld` (**déjà fait dans les machines virtuelles**) :

```
$ systemctl stop firewalld  
$ systemctl disable firewalld  
$ yum install iptables-services  
$ systemctl enable iptables  
$ systemctl enable ip6tables
```

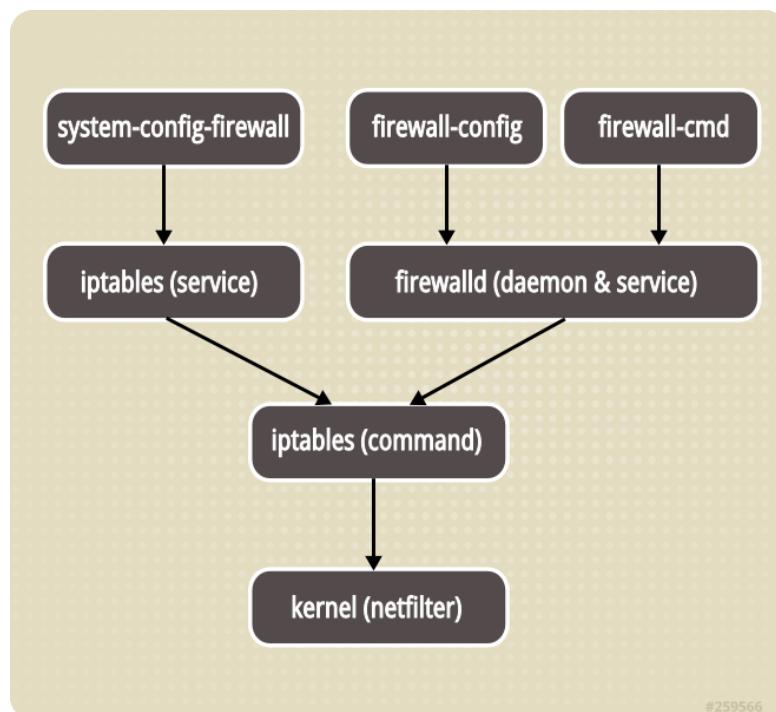


Figure 13.1 : Différences<sup>42</sup> `iptables` et `firewalld`

<sup>42</sup> Source : [https://access.redhat.com/documentation/en-US/Red\\_Hat\\_Enterprise\\_Linux/7/html/Security\\_Guide/sec-Using\\_Firewalls.html](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Security_Guide/sec-Using_Firewalls.html)

## 13.2 Aperçu d'IPTables

Sur la figure 13.2, nous pouvons voir schématiquement, le fonctionnement d'iptables.

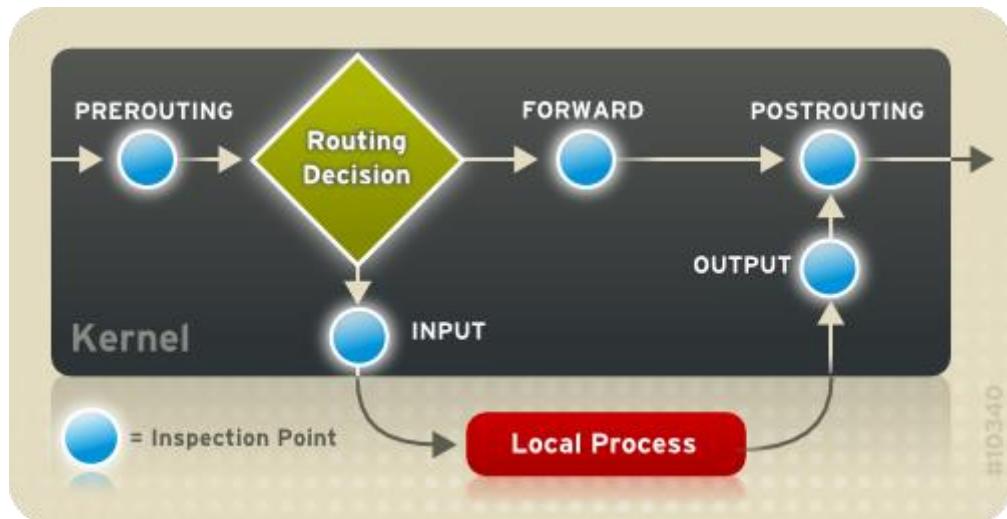


Figure 13.2 : fonctionnement schématique<sup>43</sup> d'IPTables

Lorsqu'un paquet arrive, il passe à travers plusieurs « points d'inspection » que nous appellerons également *chaîne de trafic*.

Tout d'abord, les règles **prerouting** sont appliquées au paquet qui arrive. Ces règles permettent, notamment, de modifier l'adresse IP de destination du paquet. C'est utile lorsqu'il faut « traduire » une adresse IP publique vers une adresse IP privée.

Une fois la phase de **prerouting** passée, le paquet arrive au système de routage. Si le paquet est destiné à la machine locale, il est transmis au point d'inspection (ou chaîne) **input**. Si le paquet est destiné à une autre machine du réseau (dans le cas où la machine est configurée en mode routeur), le paquet est transmis au point d'inspection (ou chaîne) **forward**.

Les règles **input** sont destinées à protéger la machine courante des accès depuis l'extérieur. Ainsi, tout paquet ayant pour destination la machine courante (qu'il vienne du réseau local ou de l'extérieur) est transmis à ce point de décision (ou chaîne). Il est courant dans ces règles d'autoriser uniquement les ports, protocoles et adresses IP autorisées à se connecter à la machine. Une fois autorisé, le trafic est transmis au processus local (programme serveur écoutant sur le port).

Les règles **forward** sont destinées à filtrer les paquets que la machine transmet. Ainsi, ce point de décision (ou chaîne) n'est utilisé que lorsque la machine est configurée en mode routeur. Il est courant dans ces règles d'autoriser les applications disponibles sur le réseau protégé par le routeur. Les règles peuvent aussi bien concerner le trafic émis que le trafic reçu par les machines du réseau.

Sur notre machine routeur, le point d'inspection **input** est utilisé pour tout le trafic reçu, à destination de la machine routeur (dont l'adresse IP destination correspond à l'une des IPs du

<sup>43</sup> Source : [https://access.redhat.com/documentation/en-US/Red\\_Hat\\_Enterprise\\_Linux/6/html/Security\\_Guide/sect-Security\\_Guide-IPTables.html](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Security_Guide/sect-Security_Guide-IPTables.html)

serveur). Le point d'inspection **forward** est utilisé pour tout le trafic à destination ou en provenance de la machine client.

Enfin, le point d'inspection **output** est utilisé pour le trafic qui est émis par la machine courante. Il est courant de placer dans ce point d'inspection (ou chaîne) les règles filtrant le trafic en sortie (bloquer un site, comme facebook.com par exemple, peut facilement être mis en place ici).

Pour terminer, il reste le point d'inspection **postrouting** que nous avons déjà utilisé : les règles prévues ici sont utiles pour modifier le paquet juste avant son expédition. Par exemple, le mécanisme de NAT (ie. la translation d'adresse) se met en place ici.

### 13.3 Les éléments d'une règle iptables

#### 13.3.1 Le point d'inspection (ou chaîne de trafic)

Le premier élément à déterminer est le point d'inspection dans lequel la règle doit prendre place. Pour les règles de firewall, il y a 3 points d'inspection à considérer : **input**, **forward** et **output**.

Comme dit précédemment, si le trafic est à destination de la machine locale, c'est dans la chaîne **input** qui faut placer la règle. Si le trafic *traverse* la machine pour atteindre un autre réseau, c'est dans la chaîne **forward** que la règle doit être ajoutée. S'il s'agit du trafic envoyé par la machine locale, c'est dans la chaîne **output** que la règle doit être ajoutée.

#### 13.3.2 La portée

Chaque règle doit être précise pour ne concerner que le trafic souhaité. Ainsi, la portée de chaque règle peut être limitée en spécifiant un ou plusieurs éléments : l'interface réseau d'entrée et/ou de sortie, l'adresse IP source et/ou destination, le protocole réseau ou transport (ICMP, IPv6, TCP, UDP, etc.), le port source et/ou destination (dans le cas de TCP/UDP).

Ainsi, pour cibler le service SSH sur la machine locale, on pourrait ajouter une règle *mentionnant le protocole TCP, à destination dont l'IP est donnée, pour le port destination 22*.

L'ordre des règles dans un point d'inspection (ou d'une chaîne) est important : les règles sont analysées les unes à la suite des autres. Dès qu'une règle correspond au trafic observé, elle est utilisée pour déterminer l'action à entreprendre.

Il est donc important de commencer par les règles les plus précises et terminer par les règles plus générales.

#### 13.3.3 L'action

Une fois qu'une règle correspond au trafic observé, il faut déterminer quelle action il faut prendre : *accept* pour mentionner que le trafic est autorisé par le firewall, *reject* pour indiquer que le trafic est refusé (avec une réponse ICMP adéquate), *drop* pour indiquer que le trafic est refusé (sans réponse ICMP, préférable), *log* pour mentionner que le trafic est consigné dans les journaux systèmes.

Il est également possible de préciser un *point d'inspection* comme action de la règle. Cette option est particulièrement intéressante quand l'utilisateur crée ses propres points d'inspection, nous parlerons de cette possibilité plus tard.

### 13.3.4 Exemples

```
$ iptables -A INPUT -i eno16777736 -d 192.168.190.50 -p tcp --destination-port 22 -j ACCEPT
```

Cette règle est ajoutée au point d'inspection **input** : le trafic reçu par la machine courante. Elle précise que si le trafic arrive (option **-A INPUT**) par l'interface d'entrée **eno16777736** (option **-i**), à destination de l'adresse IP **192.168.190.50** (option **-d**), dont le protocole de transport est **tcp** (option **-p**), avec comme port destination le port SSH **22** (option **--destination-port**), alors le trafic est autorisé (option **-j ACCEPT**).

```
$ iptables -A INPUT -d 192.168.131.2 -p udp --destination-port 53 -j ACCEPT
```

Cette règle est également ajoutée au point d'inspection **input**. Elle précise que le trafic arrivant sur la machine (option **-A INPUT**), à destination de l'adresse IP **192.168.131.2**, en utilisant le port **UDP** (option **-p udp**) **DNS 53** (option **--destination-port 53**) est autorisé (option **-j ACCEPT**).

Dans cet exemple, j'ai choisi de ne pas mentionner l'interface d'entrée (via l'option **-i**). Cela ne pose aucun problème pour autant que la règle soit non-ambigüe. Si je n'avais pas mentionné l'adresse IP destination (option **-d**), la règle aurait autorisé le trafic à destination du DNS, quelle que soit l'interface réseau qui l'aurait reçu.

### 13.3.5 Un firewall simple pour un serveur

1	<code>#!/usr/bin/perl</code>
2	<code>use strict;</code>
3	<code>my \$IPTABLES="/usr/sbin/iptables";</code>
4	<code>my \$INTERNAL_NETWORK="10.0.0.0/24";</code>
5	<code>my \$DEVICE_NETWORK="eno16777736";</code>
6	<code>my \$PORTS_TCP="21,22,80,50000:50500";</code>
7	<code>my \$PORTS_UDP="53";</code>
8	<code>#### Reset all rules</code>
9	<code>`\$IPTABLES -F`;</code>
10	<code>`\$IPTABLES -X`;</code>
11	<code>`\$IPTABLES -t nat -F`;</code>
12	<code>####</code>
13	<code># INPUT</code>
14	<code>####</code>
15	<code>`\$IPTABLES -A INPUT -i lo -j ACCEPT`;</code>
16	<code>`\$IPTABLES -A INPUT -i \$DEVICE_NETWORK -s \$INTERNAL_NETWORK -p icmp -j ACCEPT`;</code>
17	<code>`\$IPTABLES -A INPUT -i \$DEVICE_NETWORK -p tcp -m state --state NEW -m multiport --dports \$PORTS_TCP -j ACCEPT`;</code>
18	<code>`\$IPTABLES -A INPUT -i \$DEVICE_NETWORK -p udp -m multiport --dports \$PORTS_UDP -j ACCEPT`;</code>
19	<code>`\$IPTABLES -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT`;</code>
20	<code>`\$IPTABLES -A INPUT -j LOG`;</code>
21	<code>`\$IPTABLES -P INPUT DROP`;</code>
22	<code>####</code>
23	<code># FORWARD</code>
24	<code>####</code>
25	<code>`\$IPTABLES -P FORWARD DROP`;</code>

```

#####
# OUTPUT
#####
19 `$IPTABLES -P OUTPUT ACCEPT`;

```

Script 13.3 : exemple firewall serveur

Le script 13.3 montre un *exemple de firewall* pour un serveur (qui ne joue pas le rôle de routeur). Nous allons rapidement parcourir les lignes principales pour en expliquer le fonctionnement.

Les lignes 4 à 7 permettent de spécifier la configuration souhaitée : \$INTERNAL\_NETWORK doit contenir le préfixe du réseau local, \$DEVICE\_NETWORK le nom de l'interface réseau, \$PORTS\_TCP et \$PORTS\_UDP, la liste des ports destination que l'on souhaite ouvrir. Dans notre exemple, on suppose que le réseau local est 10.0.0.0/24, l'interface réseau est eno1677736 et que les ports à ouvrir sont les ports TCP 21 (ftp), 22 (ssh), 80 (web), 50000-50500 (tous les ports entre 50000 et 50500). En UDP, le port 53 (dns) doit être ouvert sur la machine.

*On suppose donc dans cet exemple que la machine exécute un service Web, FTP, SSH et DNS. De plus, les ports entre 50000 et 50500 doivent également être ouverts.*

Les lignes 8 à 10 suppriment toutes les règles qui existeraient au niveau du firewall.

Les lignes 11 à 17 forment le cœur de ce firewall. Il faut bien comprendre que les règles sont analysées séquentiellement et que, lorsqu'une règle *correspond* et *est utilisée*, l'action mentionnée est exécutée et le traitement s'arrête pour ce trafic.

La ligne 11 autorise tout le trafic arrivant par l'interface *loopback*. Il est nécessaire de spécifier cette règle pour éviter un comportement erratique de la machine.

La ligne 12 permet d'autoriser le trafic ICMP (ping, traceroute, ...) si la source est le réseau local. Cela permet à la machine de répondre aux requêtes locales alors que les requêtes distantes sont refusées.

La ligne 13 autorise le trafic vers les ports TCP destinations mentionnés dans \$PORTS\_TCP. Le trafic est autorisé uniquement pour les demandes de connexion (-m state --state NEW).

La ligne 14 autorise le trafic vers les ports UDP destinations mentionnés dans \$PORTS\_UDP.

La ligne 15 **est très importante** : elle assure que le trafic qui arrive *qui serait une réponse<sup>44</sup> à une requête envoyée par la machine* est autorisé.

La ligne 16 enregistre dans le fichier journal tout trafic qui n'aurait pas encore été traité par une règle précédente. Le traitement continue ensuite.

La ligne 17 change le comportement par défaut du point d'inspection INPUT. Ainsi, le comportement par défaut est que, tout trafic non traité, est jeté – sans réponse à l'émetteur –.

---

<sup>44</sup> Si la machine envoie du trafic avec comme port source 1056 et comme port destination 80, la réponse arrivera avec ces ports inversés (port source 80 ; port destination 1056). La règle autorise toute réponse qui émane d'une requête faite par la machine. Sans cette règle, il faudrait écrire une règle spécifique autorisant toute réponse aux requêtes.

La ligne 18 change le comportement par défaut du point d'inspection FORWARD (utilisé pour le mode routeur) en rejetant tout trafic.

La ligne 19, s'appliquant au point d'inspection OUTPUT (trafic émis par la machine), assure que le comportement par défaut est d'autoriser le trafic. Ainsi, tout paquet émis par la machine est autorisé à sortir.

### 13.4 Firewall avec mode routeur

Installer un firewall sur une machine jouant le rôle de routeur nécessite une configuration plus précise. Il faut en effet garnir le point d'inspection *forward*. Cependant, en mode routeur, il faut prévoir des règles protégeant la machine routeur et des règles protégeant le réseau derrière celle-ci. Il faut également peut-être *rediriger* du trafic vers cette machine si certaines applications doivent être visibles depuis l'extérieur.

Reprendons notre réseau :

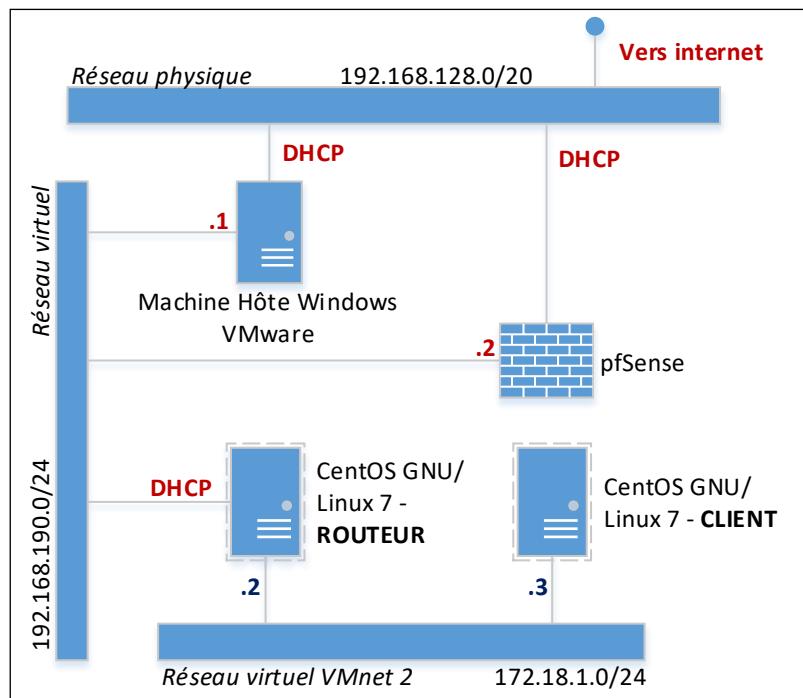


Figure 13.4 : exemple de réseau

Comme nous pouvons voir sur la figure 13.4, la machine routeur dispose de 2 interfaces réseaux et est connectée à la fois aux réseaux 192.168.190.0/24 et 172.18.1.0/24. De plus, la machine routeur réalise une *translation d'adresse* pour le réseau 172.18.1.0/24. Cela signifie que lorsqu'un paquet est transmis par la machine client 172.18.1.3, le firewall change l'adresse IP source par la sienne, 192.168.190.50 (par exemple). Cette translation nous épargne le fait de devoir configurer le réseau 172.18.1.0/24 sur les machines extérieures. En effet, puisque tout le trafic est masqué derrière l'adresse IP de la machine routeur, il faut juste que la machine routeur soit connue, ce qui est le cas.

Imaginons maintenant que nous souhaitons configurer un service FTP ou HTTP sur la machine client et rendre ce dernier accessible depuis l'extérieur, que devons-nous faire ?

Le problème principal vient du fait que le réseau 172.18.1.0/24 n'est pas connu hors de la machine routeur. Donc personne ne sait comment atteindre la machine 172.18.1.3 en dehors de la machine routeur.

Dès lors, comment rendre ce service, actif sur la machine client, accessible ?

Il faut configurer des règles particulières pour indiquer *que le trafic arrivant sur la machine routeur (sur un port déterminé) doit être redirigé vers la machine client*.

```
$ iptables -t nat -A PREROUTING -i eno16777736 -d 192.168.190.50 -p tcp --destination-port 5022 -j DNAT --to-destination 172.18.1.3:22
```

Dans cet exemple, nous redirigeons le trafic arrivant par l'interface eno16777736 (-i eno16777736), à la machine routeur (-d 192.168.190.50) sur port TCP 5022 (-p tcp --destination-port 5022) vers la machine client 172.18.1.3 sur le port 22 (-j DNAT --to-destination 172.18.1.3:22).

Ainsi, on peut se connecter au service SSH depuis l'extérieur en se connectant comme suit :

```
$ ssh 192.168.190.50 -p 5022
```

Comme nous le voyons, nous mentionnons l'adresse IP de la machine routeur et le port configuré, celui-ci est redirigé vers la machine client sur le port TCP 22 (ssh).

### 13.4.1 Créer ses propres points d'inspection

Dans des configurations complexes, comme celle d'une machine linux jouant le rôle de routeur, les règles à prévoir sont souvent nombreuses. Le risque principal réside dans un oubli malencontreux ouvrant la voie à un trafic non désiré.

En effet, dans le point d'inspection INPUT : nous avons le trafic arrivant sur la machine routeur, qu'il provienne de l'extérieur ou du réseau de la machine client. Dans le point d'inspection FORWARD, nous avons le trafic provenant de l'extérieur vers le réseau de la machine client et inversement alors que le point d'inspection OUTPUT reprend le trafic que la machine routeur envoie, aussi bien vers l'extérieur que vers le réseau de la machine client. Comme nous le voyons, les choses ne sont pas simples.

Pour simplifier la configuration de telle configuration, il est intéressant de créer ses propres points d'inspection. Ainsi, dans de tel cas, je conseille de créer des points d'inspection correspondant à chaque direction du trafic. Ainsi, nous pourrions créer les points d'inspection suivants :

- **trust2untrust** : du réseau de la machine client vers l'extérieur
- **fw2untrust** : de la machine routeur vers l'extérieur
- **fw2trust** : de la machine routeur vers le réseau de la machine client
- **untrust2trust** : de l'extérieur vers le réseau de la machine client
- **trust2fw** : du réseau de la machine client vers la machine routeur
- **untrust2fw** : de l'extérieur vers la machine routeur

A l'intérieur de chacun, il est possible de créer des règles, comme celles que nous avons déjà étudiées. Ainsi, nous pouvons autoriser le trafic vers l'extérieur en mentionnant les règles suivantes :

```
$ iptables -A trust2untrust -j ACCEPT
```

```
$ iptables -A fw2untrust -j ACCEPT
```

Nous pouvons ensuite permettre à la machine routeur d'atteindre la machine client :

```
$ iptables -A fw2trust -j ACCEPT
```

On peut, si on le souhaite, permettre au réseau de la machine cliente d'atteindre la machine routeur sans restriction :

```
$ iptables -A trust2fw -j ACCEPT
```

Il reste à configurer les règles destinées à protéger les machines avec le trafic provenant de l'extérieur :

```
$ iptables -A untrust2trust -m state --state RELATED,ESTABLISHED -j ACCEPT  
$ iptables -A untrust2trust -p tcp --destination-port 22 -j ACCEPT  
$ iptables -A untrust2trust -j LOG  
$ iptables -A untrust2trust -j DROP
```

*Dans ce premier exemple, nous autorisons uniquement le trafic vers le port 22 (ssh) et les réponses à des requêtes émises par l'une des machines du réseau client.*

```
$ iptables -A untrust2fw -m state --state RELATED,ESTABLISHED -j ACCEPT  
$ iptables -A untrust2fw -p tcp --destination-port 80 -j ACCEPT  
$ iptables -A untrust2fw -j LOG  
$ iptables -A untrust2fw -j DROP
```

*Dans ce second exemple, nous autorisons uniquement le trafic vers le port 80 (http) et les réponses à des requêtes émises par la machine routeur.*

Une fois les règles rédigées, il faut encore utiliser ces nouveaux points d'inspection. En effet, les noms choisis sont sans signification pour le système, il faut donc créer des actions utilisant ces nouveaux points d'inspection comme suit :

```
$ iptables -A INPUT -i lo -j ACCEPT  
$ iptables -A INPUT -i eno1677736 -j untrust2fw  
$ iptables -A INPUT -i eno33554976 -j trust2fw  
$ iptables -A FORWARD -i eno1677736 -o eno33554976 -j untrust2trust  
$ iptables -A FORWARD -i eno33554976 -o eno1677736 -j trust2untrust  
$ iptables -A OUTPUT -o lo -j ACCEPT  
$ iptables -A OUTPUT -o eno1677736 -j fw2untrust  
$ iptables -A OUTPUT -o eno33554976 -j fw2trust
```

Les règles mentionnées permettent d'utiliser les points d'inspection créés. Comme nous le voyons, la mention de l'interface d'entrée (option `-i`) et/ou de l'interface de sortie (option `-o`) permet de sélectionner le point d'inspection sans aucune ambiguïté.

### 13.4.2 Script de firewall pour machine routeur

```
1 UNTRUST_IP=192.168.190.50  
2 UNTRUST_IF=eno1677736  
3 TRUST_NET=172.18.1.0/24  
4 TRUST_IF=eno33554976  
5 SERVER1_IP=172.18.1.3  
6 SERVER1_TCP=21,22,53,80,443,30000:33000  
7 SERVER1_UDP=53
```

```

8  iptables -t nat -F
9  iptables -t nat -X
10 iptables -F
11 iptables -X

12 iptables -N untrust2fw
13 iptables -N fw2untrust
14 iptables -N fw2trust
15 iptables -N trust2fw
16 iptables -N untrust2trust
17 iptables -N trust2untrust

    # firewall => trust
18 iptables -A fw2trust -j ACCEPT

    # firewall => untrust
19 iptables -A fw2untrust -j ACCEPT

    # trust => untrust
20 iptables -A trust2untrust -j ACCEPT

    # untrust => trust
21 iptables -A untrust2trust -p tcp -d $SERVER1_IP -m multiport --dports
$SERVER1_TCP -m state --state NEW -j ACCEPT
22 iptables -A untrust2trust -p udp -d $SERVER1_IP -m multiport --dports
$SERVER1_UDP -j ACCEPT
23 iptables -A untrust2trust -m state --state ESTABLISHED,RELATED -j
ACCEPT
24 iptables -A untrust2trust -j LOG
25 iptables -A untrust2trust -j DROP

    # untrust => fw
26 iptables -A untrust2fw -m state --state ESTABLISHED,RELATED -j ACCEPT
27 iptables -A untrust2fw -j LOG
28 iptables -A untrust2fw -j DROP

    # trust => fw
29 iptables -A trust2fw -j ACCEPT

## NAT rules
30 iptables -t nat -A PREROUTING -i $UNTRUST_IF -d $UNTRUST_IP -p tcp -m
multiport --dports $SERVER1_TCP -j DNAT --to-destination $SERVER1_IP
31 iptables -t nat -A PREROUTING -i $UNTRUST_IF -d $UNTRUST_IP -p udp -m
multiport --dports $SERVER1_UDP -j DNAT --to-destination $SERVER1_IP

32 iptables -t nat -A POSTROUTING -s $TRUST_NET -j MASQUERADE

## Distribution rules
33 iptables -A INPUT -i lo -j ACCEPT
34 iptables -A INPUT -i $UNTRUST_IF -j untrust2fw
35 iptables -A INPUT -i $TRUST_IF -j trust2fw
36 iptables -A INPUT -j LOG
37 iptables -A INPUT -j DROP

38 iptables -A FORWARD -i $UNTRUST_IF -o $TRUST_IF -j untrust2trust
39 iptables -A FORWARD -i $TRUST_IF -o $UNTRUST_IF -j trust2untrust
40 iptables -A FORWARD -j LOG
41 iptables -A FORWARD -j DROP

42 iptables -A OUTPUT -o lo -j ACCEPT
43 iptables -A OUTPUT -o $UNTRUST_IF -j fw2untrust

```

```

44 iptables -A OUTPUT -o $TRUST_IF -j fw2trust
45 iptables -A OUTPUT -j LOG
46 iptables -A OUTPUT -j DROP

```

Script 13.5 : firewall pour machine routeur

Le script 13.5 présente une configuration supportant la machine routeur. Il est divisé en 4 sections importantes.

La **première** section (lignes 1 à 7) recense les adresses et interfaces réseaux à prendre en compte : aussi bien du coté *untrust* (interface connectée à internet) que du coté *trust* (interface reliée au réseau interne, celui de la machine client par exemple). Les variables ainsi définies sont utilisées comme paramètre des commandes de configuration du firewall.

La **seconde** section (lignes 8 à 17) est la phase d'initialisation avec suppression des règles existantes, des points d'inspection avant la recréation de ceux-ci (cela permet d'adapter le script et de le relancer).

La **troisième** section (lignes 18 à 32) configure les règles de firewall dans les points d'inspection créés. Ces règles utilisent les variables configurées à la première section. Les règles nécessaires au fonctionnement du NAT sont également créées (*ouverture* et *redirection* du trafic pour les connexions entrantes et *translation d'adresse* pour les connexions sortantes). Ainsi, les 6 points d'inspection sont configurés (lignes 18 à 29) :

- **fw2trust** : qui est le point d'inspection de *la machine routeur* vers *le réseau interne* est très souvent assez peu surveillé : on autorise ici tout le trafic sans restriction (-A ACCEPT)
- **fw2untrust** : qui est le point d'inspection de *la machine routeur* vers *le réseau internet* est également peu surveillé : la machine routeur peut envoyer et se connecter sans restriction à Internet (-A ACCEPT)
- **trust2untrust** : qui est le point d'inspection du *réseau interne* vers *le réseau internet* est, dans cet exemple, peu surveillé : le réseau interne peut se connecter n'importe où, avec n'importe quel port vers internet. Il est possible de garnir ce point d'inspection de règles plus restrictives si l'on souhaite empêcher l'accès à certains sites, ... au départ du réseau interne.
- **untrust2trust** : qui est le point d'inspection du *réseau internet* vers *le réseau interne* est un des points d'inspection les plus importants à configurer : les règles de protections pour le réseau interne se trouvent ici : on peut autoriser certains serveurs du réseau interne à être accéder depuis l'internet au moyen de règles précises mentionnées ici :
  - c'est le cas de la première règle de ce point qui autorise le trafic TCP vers certaines machines du réseau interne (vers \$SERVER1\_IP, sur les ports \$SERVER1\_TCP). La règle mentionne également qu'il doit s'agir d'une demande d'ouverture de connexion (-m state --state NEW).
  - La seconde règle autorise le trafic UDP vers la machine (\$SERVER1\_IP) et les ports (\$SERVER1\_UDP) mentionnés.
  - La troisième règle, quant à elle, autorise le trafic qui serait une réponse à une requête envoyée précédemment (-m state --state ESTABLISHED,RELATED). Cette règle doit, dans la plupart des configurations, être présente.
  - Les 2 dernières règles, classiques, enregistrent dans les journaux (-j LOG) le trafic inadéquat et jettent celui-ci de manière silencieuse – sans réponse (-j DROP).

- **untrust2fw** : qui est l'autre point d'inspection important de la configuration, matérialisant les règles à appliquer *du réseau internet vers la machine routeur*. Dans ce point, il s'agit de protéger la machine routeur correctement. Dans notre exemple, il n'y a pas beaucoup de règles proposées :
  - La première règle autorise le trafic qui serait une réponse à une requête envoyée précédemment (`-m state --state ESTABLISHED,RELATED`). Cette règle doit, dans la plupart des configurations, être présente.
  - Les 2 règles suivantes, classiques, enregistrent dans les journaux (`-j LOG`) le trafic inadéquat et jettent celui-ci de manière silencieuse – sans réponse (`-j DROP`).
- **trust2fw** : qui est le point d'inspection du *réseau interne* vers *la machine routeur* est très souvent assez peu surveillé : on autorise ici tout le trafic sans restriction (`-A ACCEPT`).

A la fin de la **troisième** section, les lignes 30 à 32 configurent l'ouverture des ports et le NAT. Ainsi :

- Les lignes 30 et 31 assurent que le trafic entrant (sur l'interface internet `$UNTRUST_IF`, avec l'IP du routeur comme destination `$UNTRUST_IP`), sur les ports mentionnés (en TCP `$SERVER1_TCP` ou en UDP `$SERVER1_UDP`) est bien redirigé vers la machine interne dont l'adresse est configurée (avec l'IP `$SERVER1_IP`).
- La ligne 32 est la ligne qui autorise la translation d'adresses en sortie (`-j MASQUERADE`) pour le réseau interne (`-s $TRUST_NET`).

Enfin, la **quatrième** section (lignes 33 à 46) installe les règles de distribution (envoi du trafic dans les points d'inspection créés et configurés).

Une fois le firewall configuré et le script exécuté, il convient **de tester celui-ci** pour être sûr que les règles mises en place sont conformes à ce qui est souhaité. Une fois le firewall bien adapté, il **faut sauvegarder** cette configuration pour **qu'elle s'active** automatiquement au démarrage de la machine. Pour ce faire, il faut lancer la commande :

```
$ service iptables save
```

## 13.5 Exercices

On vous demande de :

1. Configurer un firewall sur votre **machine routeur** de sorte à :
  - a. Permettre aux machines du réseau interne (192.168.131.x) d'accéder à internet (via le NAT)
  - b. Ouvrir les ports suivants sur la machine routeur (point d'inspection `untrust2fw`) :
    - i. Autoriser le port SSH (tcp/22) depuis l'extérieur
    - ii. Autoriser le port DNS (port udp/53 et tcp/53) depuis l'extérieur
    - iii. Autoriser, uniquement depuis les adresses 192.190.190.x, les requêtes ICMP.
  - c. Limiter le trafic sortant provenant du réseau interne (point d'inspection `trust2untrust`) au port web (tcp/80 et tcp/443) et dns (udp/53).
  - d. Ouvrir le port 4022 (TCP) et le rediriger vers la machine client. Configurer le port SSH de votre machine client sur ce port 4022. Votre machine client doit être accessible, en SSH, via ce port.
2. Configurer sur votre **machine client** un service FTP comme suit :
  - a. Le port d'écoute doit être le port TCP 4000 (à la place du port 21)
  - b. Le mode passif doit être configuré sur les ports 63000 à 63500
3. Configurer un firewall sur votre **machine client** de sorte à :
  - a. Permettre l'accès au port SSH 4022
  - b. Permettre l'accès au service FTP configuré au point 2 et aux ports configurés pour le mode passif.

Tester votre configuration depuis la machine routeur (pour le FTP) et depuis l'hôte Windows 7 (pour la configuration SSH).

## Leçon 14 : Le service DHCP

### 14.1 Introduction

Le service DHCP<sup>45</sup> est devenu, ces dernières années, un service indispensable présent sur la plupart des réseaux. En effet, le service DHCP permet **de distribuer la configuration réseaux aux postes clients**. Avec l'arrivée des périphériques nomades (portables, smartphones, tablettes, ...) et prochainement l'émergence de l'internet des objets, la possibilité de configurer automatiquement le réseau sur le poste client est indispensable.

Ainsi, peu importe où se trouve l'utilisateur, il ne doit pas se préoccuper des aspects techniques de la configuration réseau, hormis la phase d'authentification bien sûr. Tout paraît ainsi beaucoup plus simple.

Dans cette leçon, nous allons découvrir comment configurer le service DHCP pour distribuer cette configuration aux postes clients. **Attention !** Nous nous limiterons à la configuration IPv4 car, le déploiement de DHCPv6 est assez rare au profit du protocole NDP<sup>46</sup>.

### 14.2 Configuration du DHCP<sup>47</sup>

La configuration du service DHCP repose sur le fichier `/etc/dhcp/dhcpd.conf`. Ce fichier est, par défaut, vide. Pour commencer, nous allons récupérer le modèle présent dans la documentation :

```
$ cd /etc/dhcp
$ cp /usr/share/doc/dhcp-4.2.5/dhcpd.conf.example ./dhcpd.conf
```

Nous allons maintenant modifier le modèle de configuration récupéré de sorte que la configuration corresponde à nos besoins.

#### 14.2.1 Options globales

Option	Explication
<code>option domain-name</code>	Mentionne le nom de domaine DNS qui sera configuré au niveau des postes clients.
<code>option domain-name-servers</code>	Permet d'indiquer des serveurs DNS à transmettre aux clients. Les clients interrogeront ces serveurs DNS.
<code>default-lease-time</code>	Mentionne en seconde la durée du bail par défaut c'est-à-dire le temps pour lequel l'adresse IP est donnée. Quelques valeurs : 4 heures → 14400 secondes (beaucoup de mobiles) 8 heures → 28800 secondes (idéal pour les mobiles) 1 jour → 86400 secondes 8 jours → 691200 secondes (pour les PCs fixes)
<code>max-lease-time</code>	Mentionne, en seconde, la durée <i>maximale</i> du bail. C'est le temps maximum pour lequel l'adresse IP est donnée.
<code>ntp-servers</code>	Mentionne les serveurs de temps à transmettre aux postes client.
<code>ddns-update-style</code>	Active ou désactive les mises à jour DNS
<code>authoritative</code>	Mentionne si le serveur DHCP fait autorité sur le réseau
<code>log-facility</code>	Détermine comment les traces systèmes sont conservées.

<sup>45</sup> Dynamic Host Configuration Protocol

<sup>46</sup> Neighbor Discovery Protocol

<sup>47</sup> Le service DHCP utilise les ports UDP 67 et 68, important pour la configuration du firewall

Un certain nombre de ces options peuvent être utilisées *localement* dans la définition du sous-réseau, d'un groupe ou d'un réseau partagé.

**Attention ! Le modèle de configuration contient la définition de plusieurs sous-réseaux. IL CONVIENT DE SUPPRIMER TOUS LES ELEMENTS INUTILES.**

### 14.2.2 Groupement et étendue

Comme nous l'avons vu, toute option placée hors d'un bloc donné à une étendue globale. Pour éviter des conflits sur des serveurs DHCP importants, il faut utiliser les paramètres de groupement (proche d'une notion de *bloc* dans les langages de programmation) convenablement.

On trouve, dans la configuration DHCP, les trois groupements suivants : `group`, `shared-network` et `subnet`.

Le groupement `subnet` permet de mentionner toutes les options d'un sous-réseau donné. Son étendue se limite à la définition du `subnet`. C'est le groupement le plus élémentaire (ne contenant aucun autre groupement), il mentionne **l'adresse réseau** et **le masque** associé :

```
subnet 172.18.1.0 netmask 255.255.255.0 {  
    # Options du subnet  
    ...  
}
```

Le groupement `shared-network` définit les options du serveur DHCP pour un même réseau physique. Ce groupement peut contenir des options propres mais également plusieurs groupement `subnet` avec, pour chacun, ses paramètres spécifiques. Il mentionne **le nom de ce réseau** (choisi par l'administrateur) :

```
shared-network MON-RESEAU {  
    # Options communes à tous les subnets  
    ...  
    subnet 172.18.1.0 netmask 255.255.255.0 {  
        # Options du subnet  
        ...  
    }  
}
```

Le groupement `group` est plus général puisqu'il permet de définir les options locales à un groupe de déclaration. Nous ne l'utiliserons pas dans le cadre de cette leçon.

Ces groupements ont un intérêt lorsqu'un même serveur, connecté sur plusieurs réseaux, distribue des adresses pour plusieurs branches réseaux différentes.

On devrait trouver, dans la configuration DHCP, au moins un groupement `shared-network` contenant au moins un groupement `subnet` avec les options réseaux nécessaires.

### 14.2.3 Options dans un groupement

A l'intérieur d'un groupement, nous pourrons trouver des options déjà expliquées plus haut comme option `domain-name`, option `domain-name-servers`, `default-lease-time`, `max-lease-time` et `ntp-servers`.

### Options particulières du subnet

A l'intérieur d'un *subnet*, les options particulières suivantes peuvent (et doivent pour la plupart) être définies :

Option	Explication
<code>range</code>	Cette option mentionne la plage IP considérée. On y indique la première et la dernière adresse de la plage. Exemple : <code>range 172.18.1.50 172.18.1.100;</code>  Les postes clients se verront attribuer une adresse dans cette plage. Ces adresses doivent être comprises dans le sous-réseau indiqué.
<code>routers</code>	Cette option renseigne la passerelle par défaut, utilisée par les postes clients pour sortir du réseau et atteindre internet. L'adresse IP de la passerelle doit être comprise dans le sous-réseau indiqué. Exemple :  <code>routers 172.18.1.254;</code>
<code>deny unknown-clients</code>	Mentionne que seuls les clients ayant fait l'objet d'une réservation (voir ci-dessous) sont autorisés.
<code>option domain-name</code> <code>option domain-name-servers</code> <code>default-lease-time</code> <code>max-lease-time</code> <code>ntp-servers</code>	Voir explication ci-dessus.

#### 14.2.4 Configuration d'une réservation

Une réservation est une entrée spécifique dans le DHCP qui permet d'associer une *adresse physique* (ou adresse MAC) à *une adresse IP*. Ainsi, le serveur DHCP **donne toujours l'adresse IP configurée à ces clients**.

Cette option est particulièrement intéressante pour les administrateurs systèmes car cela permet de configurer des adresses IP déterminées (et donc connues) aux clients sans avoir à configurer les postes clients eux-mêmes.

Pour réaliser une réservation, il faut connaître **l'adresse physique du poste client**. Pour rappel, une adresse physique<sup>48</sup> (ou adresse MAC) fait 48 bits représenté sous la forme hexadécimale : 00:11:22:33:44:55.

Une fois cette adresse connue, il faut ajouter une entrée `host` dans la configuration DHCP. Cette configuration reprend les options suivantes :

- L'option `hardware ethernet` qui renseigne l'adresse physique du client ;
- L'option `fixed-address` qui renseigne l'adresse IP ou le nom DNS de la machine ;
- L'option `host-name`, facultative, qui suggère, au client, son nom.

```
host mon-PC {
    hardware ethernet 00:01:3E:5A:DF:AE;
    fixed-address 172.18.1.67;
}
```

<sup>48</sup> Cette adresse physique peut être obtenue par la commande `ip link address`

#### 14.2.5 L'importance du subnet

Les groupements *subnet* sont importants car le service DHCP les utilisent pour déduire sur quelles interfaces réseaux il doit être actif. Soyez donc bien vigilant à définir votre *subnet* correctement afin que le service DHCP n'envoie pas de configuration sur une partie du réseau que vous ne souhaitez pas.

N'oubliez pas, non plus, **qu'un seul service DHCP peut être actif sur un sous-réseau donné** à un instant donné. Si vous avez plusieurs services DHCP actifs, les postes clients risquent d'avoir des configurations pouvant poser problèmes.

Une fois la configuration terminée, le service *dhcpd* peut être démarré via `systemctl` :

```
$ systemctl start dhcpcd
```

### 14.3 Les bails

Comme nous l'avons vu dans la configuration de certaines options, les adresses IP sont données pour un temps déterminé. Le service DHCP maintient les adresses IP distribuées dans un fichier, appelé le *lease*. Ce fichier est important et est maintenu lors du redémarrage du service (pour que le DHCP « se souvienne » des adresses IP attribuées).

Le fichier contenant le *lease* se trouve dans `/var/lib/dhcpcd/dhcpcd.leases`.

### 14.4 Configuration des postes clients

Les postes clients doivent être en mode de configuration dynamique pour recevoir leurs configurations par DHCP. Cette configuration peut, sous Linux, être appliquée de trois façons différentes :

- Via **Webmin**
- Via **Network Manager** et les outils `nmcli` ou `nmtui`
- En **modifiant les fichiers de configuration** en fonction de l'interface considérée dans `/etc/sysconfig/network-scripts/ifcfg-...`

Pour plus de précision, reportez-vous à la leçon expliquant la configuration réseau.

### 14.5 Exercices

On vous demande de :

1. Configurer un service DHCP sur votre machine *CentOS7-Routeur*. Il faut configurer la plage d'adresses IP comme suit : entre 192.168.131.150 et 192.168.131.200. Fixer l'adresse 192.168.131.2 (normalement fait dans une leçon précédente) pour votre routeur. Préciser toutes les options nécessaires pour compléter la configuration DHCP du serveur. Adaptez vos firewalls au besoin.
2. Configurer votre machine *CentOS7-Client* en mode DHCP pour qu'elle reçoive sa configuration par le service DHCP.
3. Ajouter une réservation pour votre machine *CentOS7-Client* de sorte que l'adresse IP reçue soit toujours 192.168.131.15.