

Table des matières

I. Objectifs du projet.....	2
1. Introduction.....	3
2. Choix matériel.....	4
3. Choix logiciel.....	5
4. Récapitulatif.....	6
II. Installation du serveur.....	7
1. Serveur web.....	8
2. Serveur SQL.....	10
3. Serveur FTP.....	11
4. Serveur SSH.....	14
III. Installation du CMS.....	15
1. Définition et choix.....	16
2. Installation.....	17
III. Sécurisation.....	21
1. Introduction.....	22
IV. Améliorations de sécurité.....	25
1. Sécurité des données.....	26
2. Sécurité du Réseau.....	29
1. Port et login SSH.....	30
2. Fail2ban.....	31

I. Objectifs du projet

1. Introduction

Une association désire mettre en place un site internet de type « web 2.0 » c'est à dire qui leur propose une interface d'administration pour écrire des articles, ajouter des pages et photographies. Les visiteurs auront également le droit d'écrire des commentaires ou envoyer des mails à l'aide de formulaires.

Ils disposent d'un accès internet et la machine utilisée sera placée en zone DMZ, donc virtuellement directement reliée à internet (pas de parefeu, pas de NAT). Un contrat de maintenance a été passé avec l'association, le technicien a donc besoin d'un accès SSH au cas où il devrait dépanner à distance. Un accès FTP sera également mis en place pour pouvoir modifier simplement les fichiers du site.

2. Choix matériel

Un serveur web sous Linux fonctionnant en mode texte demande très peu de ressources matérielles, c'est pourquoi nous avons opté pour une machine de type nettop. Voici les composants que nous avons acheté :

- Carte mère **Intel D945GSEJT** (Processeur Intel Atom) - 90€
- 1 Barette de mémoire 512MB PC6400 - 20€
- 1 Disque dur 2,5 pouces 160Go – 35€
- 1 bloc d'alimentation 12V – 20€
- Boitier Cooler Master C100 - 40€

La consommation de l'ensemble est estimée à 20W, contre 150 à 200 pour une machine de bureau classique. Les économies d'énergie sont donc très importantes, surtout pour une machine destinée à fonctionner 24h/24.

3. Choix logiciel

La distribution Linux retenue est **Debian GNU/Linux**, en raison de sa stabilité et sécurité réputée. Elle sera installée en mode texte, fera tourner un serveur de type LAMP (Linux+Apache+MySQL+PHP) et sera administrable à distance.

Voici le détail des logiciels retenus :

- **Serveur web** : Lighttpd, pour sa légèreté et simplicité par rapport à Apache2
- **Base de données** : Mysql-server
- **Administration Mysql** : PhpMyAdmin
- **Accès distant** : Openssh-server
- **Serveur FTP** : vsftpd pour sa réputation de sécurité
- **Administration parefeu** : Iptables
- **CMS** : Drupal
- **Sauvegardes** : script cron + partage NFS

4. Récapitulatif

- ▶ Installation Debian GNU/Linux dans sa version **stable**.
- ▶ Installer et configurer le serveur web
- ▶ Configurer le serveur web pour utiliser php et cgi.
- ▶ Installer et configurer le serveur de bases de données
- ▶ Installer l'interface d'administration de bases de données
- ▶ Installer et configurer le serveur FTP
- ▶ Installer le CMS
- ▶ Valider le fonctionnement du serveur web
- ▶ Mettre en place l'accès SSH
- ▶ Sécuriser le serveur en paramétrant le parefeu
- ▶ Mettre en place le système de sauvegarde automatisée
- ▶ Proposer des améliorations de sécurité pour la machine.

II. Installation du serveur

1. Serveur web

Notre serveur est relié à internet, nous pouvons donc nous servir d'apt-get (ou aptitude) pour installer des logiciels. Note : les passages tirés du terminal sont surlignés en gris. Le symbole « # » au début d'une commande signifie que nous sommes en root, alors que le « \$ » désigne le fait que nous sommes en utilisateur classique.

■ Installation de lighttpd :

```
# apt-get install lighttpd
```

Par défaut, lighttpd lit les pages web situées dans /var/www. Cela est problématique car cela nous oblige à travailler en root. Nous allons donc demander à Lighttpd de changer de répertoire de travail. Nous allons utiliser /home/jerouf/public_html

```
$ mkdir public_html  
# nano /etc/lighttpd/lighttpd.conf
```

La ligne :

```
server.document-root = "/var/www"
```

A été remplacée par :

```
server.document-root = "/home/jerouf/public_html"
```

■ Installation du support PHP :

```
# apt-get install php5-cgi php5-gd  
# lighty-enable-mod fastcgi  
# /etc/init.d/lighttpd restart
```

php5-cgi est un paquet qui fournit le support de PHP5 ainsi que CGI pour communiquer avec le serveur web. Le paquet php5-gd est une librairie graphique permettant de générer des miniatures d'images, ce qui est très utilisé dans les CMS et gestionnaires de blogs.

■ Validation du fonctionnement :

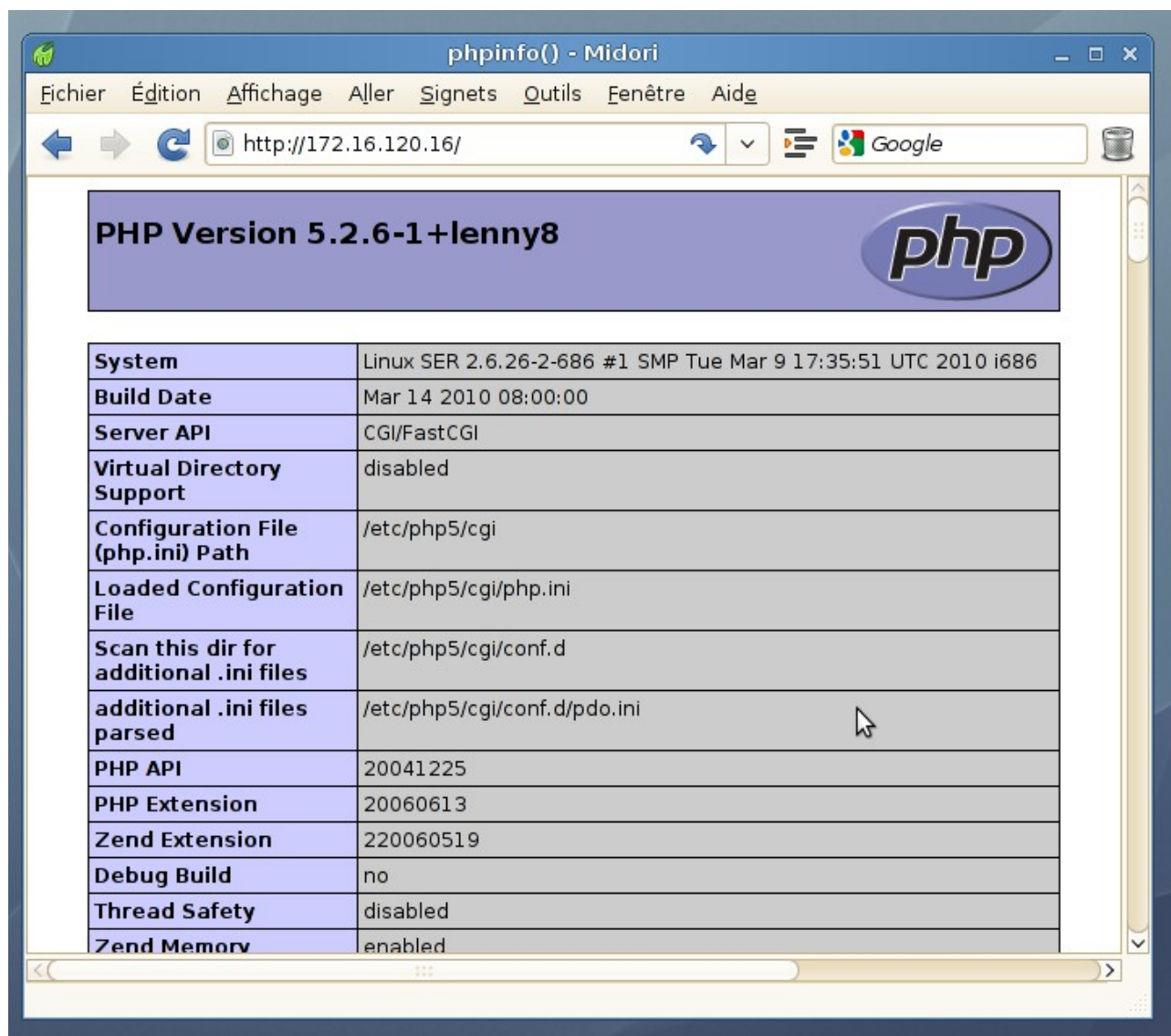
Pour valider que notre serveur web fonctionne bien, et qu'il supporte PHP, nous allons utiliser un phpinfo bien classique :

```
$ cd ~/public_html  
$ nano index.php
```

Nous avons nommé notre fichier « index.php » car il sera lu directement par le serveur web sans devoir compléter l'url. Dans ce fichier nous écrivons le code :

```
<?php phpinfo() ; ?>
```

Puis nous nous dirigeons vers une machine cliente située sur le même réseau que le serveur, et dans le navigateur web nous entrons l'adresse IP de ce dernier. Nous arrivons sur une page (**Figure1**) détaillant la liste des fonctionnalités du serveur web, cela veut dire que notre phpinfo() est correctement interprété donc notre configuration est bonne.



PHP Version 5.2.6-1+lenny8

System	Linux SER 2.6.26-2-686 #1 SMP Tue Mar 9 17:35:51 UTC 2010 i686
Build Date	Mar 14 2010 08:00:00
Server API	CGI/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/cgi
Loaded Configuration File	/etc/php5/cgi/php.ini
Scan this dir for additional .ini files	/etc/php5/cgi/conf.d
additional .ini files parsed	/etc/php5/cgi/conf.d/pdo.ini
PHP API	20041225
PHP Extension	20060613
Zend Extension	220060519
Debug Build	no
Thread Safety	disabled
Zend Memory	enabled

Figure1 : interprétation du phpinfo

2. Serveur SQL

■ Installation de Mysql-server :

```
# apt-get install mysql-server
```

■ Installation de PhpMyAdmin :

```
# apt-get install phpmyadmin
```

■ Test de la base de données :

A l'aide de la machine cliente, située sur le même réseau, nous avons entré l'adresse IP du serveur suivi de /phpmyadmin. Cela nous a mené sur l'interface de PhpMyAdmin (**Figure2**). Le fonctionnement est donc validé.

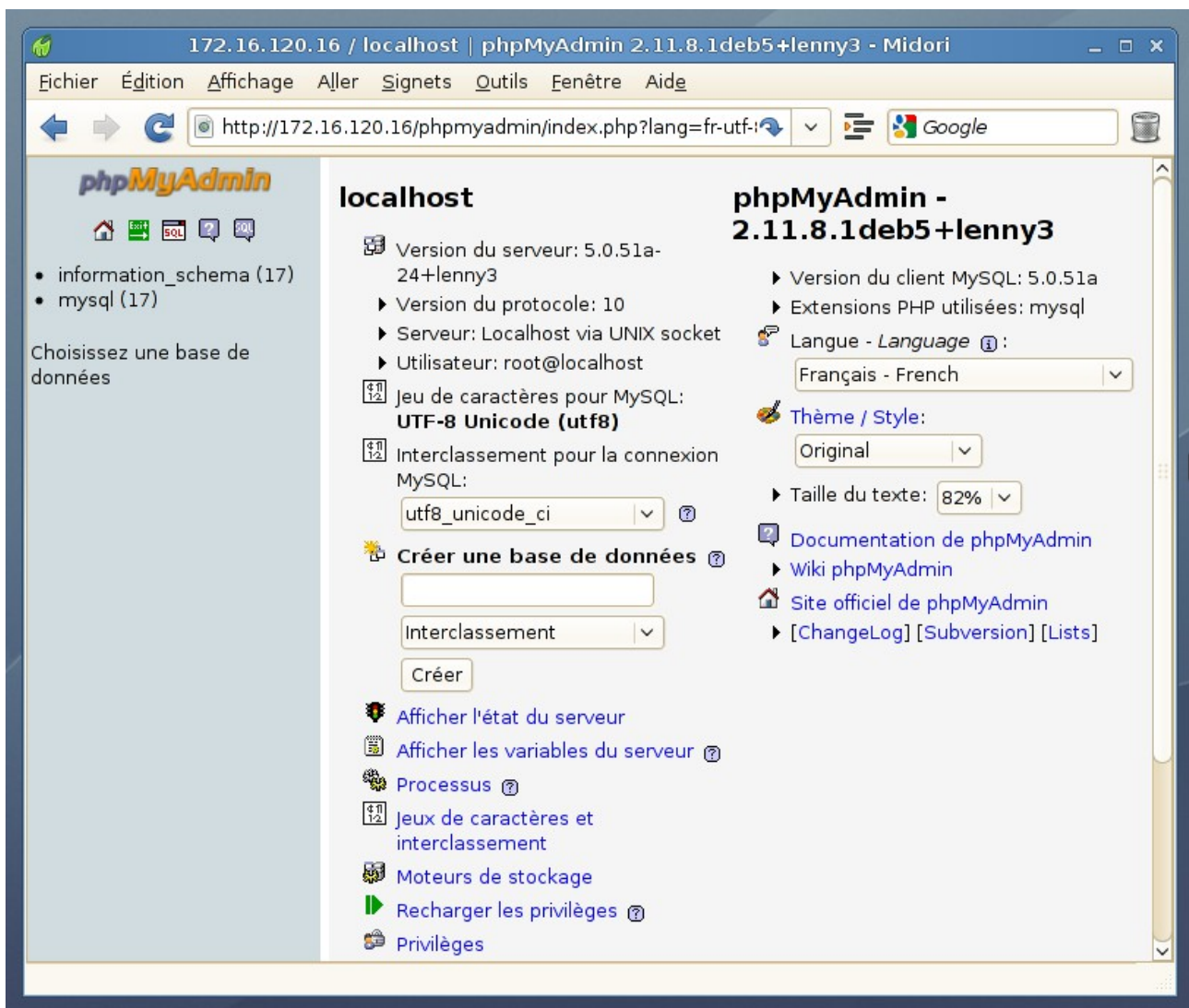


Figure 2 : Interface phpmyadmin

3. Serveur FTP

■ Installation de VSFTPd

```
# apt-get install vsftpd
```

Par défaut, seul les anonymes sont autorisés à se connecter, avec des droits limités. Nous voulons au moins un utilisateur qui ait des droits limités mais qui puisse écrire. VSFTPd permet 3 gestion des utilisateurs :

- Utilisateurs locaux de Linux
- Utilisateurs virtuels de VSFTPd
- Utilisateurs virtuels gérés par Mysql

Comme nous avons peu d'utilisateurs, et que l'accès sera réservé à un cercle privé, nous allons choisir la première solution.

■ Configuration de VSFTPd

```
# nano /etc/vsftpd.conf
```

► Interdiction des connexions anonymes :

Ligne :

```
anonymous_enable=YES
```

Modifiée en :

```
anonymous_enable=NO
```

► Autoriser les utilisateurs locaux :

Ligne (commentée):

```
#local_enable=YES
```

Modifiée en :

```
local_enable=YES
```

► Autoriser les commandes d'écriture :

Ligne (commentée):

```
#write_enable=YES
```

Modifiée en:

```
write_enable=YES
```

► chroot :

Ligne (commentée) :

```
#chroot_local_user=YES
```

Modifiée en :

```
chroot_local_user=YES
```

Rajout :

```
local_root=/home/jerouf/public_html
```

Cela va emprisonner les utilisateurs dans le dossier /home/jerouf/public_html, ils ne pourront pas remonter plus haut dans cette arborescence.

► Limiter les comptes utilisateurs :

Avec la configuration que nous venons de faire, tous les utilisateurs du système sont autorisés à se connecter en FTP, ce qui n'est pas très sûr. Pour remédier à cela il est possible de faire une « whitelist » dans laquelle nous autoriserons uniquement ceux que nous voulons. Les autres auront la réponse du serveur, mais dès que leur login sera identifié, ils seront rejetés sans même se voir demander le mot de passe.

Rajout :

```
userlist_enable=YES
userlist_deny=NO
userlist_file=/etc/vsftpd/vsftpd_users
```

« userlist_enable » permet d'activer cette blacklist. « userlist_deny » permet d'indiquer que ce n'est pas une blacklist (donc, par opposition, c'est une whitelist). Et enfin « userlist_file » indique quel fichier sera notre whitelist.

Nous devons donc créer le répertoire /etc/vsftpd puis mettre notre fichier « vsftpd_users » dedans.

```
# mkdir /etc/vsftpd
# nano /etc/vsftpd/vsftpd_users
```

Ajout :

```
jerouf
```

► Activer le monitoring

Rajout (dans /etc/vsftpd.conf) :

```
setproctitle_enable=YES
```

Ensuite lorsque le serveur sera fonctionnel, il suffira d'entrer :

```
watch -n 1 'ps ax | grep vsftpd | grep -v grep'
```

Pour savoir qui est connecté.

► Mode passif

Lors d'un fonctionnement en mode passif, le port 21 FTP sert juste à initier les connexions. Pour envoyer des commandes ou transférer des fichiers

Rajouter :

```
pasv_enable=YES
pasv_promiscuous=YES
pasv_min_port=40000
pasv_max_port=40100
port_promiscuous=YES
```

Ici nous avons spécifié que le mode passif utilisera les ports 40000 à 40100 avec divers contrôles de sécurité.

■ Validation

Sur la machine cliente (reliée au réseau) nous lançons un client FTP, par exemple Filezilla. Pour l'adresse du serveur, nous entrons son adresse IP. Le nom d'utilisateur est « jerouf » et son mot de passe est celui qu'il utilise sur leur serveur.

Filezilla se connecte au serveur, et affiche le contenu du dossier ~/public_html (**Figure3**). Notre configuration est donc validée.

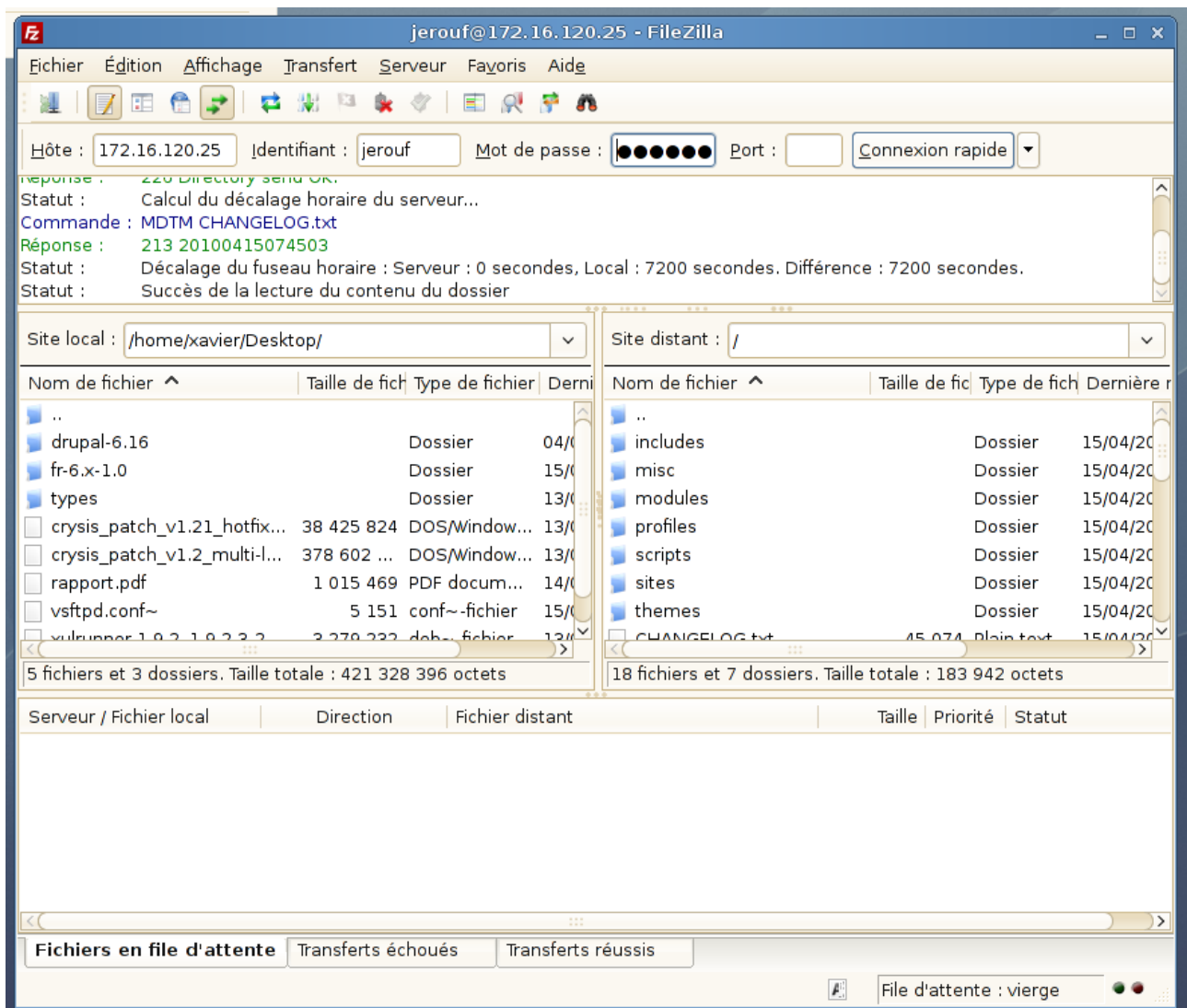


Figure3 : Filezilla se connecte au serveur FTP

4. Serveur SSH

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

Test : dans la console de la machine cliente, nous avons entré :

```
$ ssh jerouf@$IPSERVER
```

Et nous avons pu nous connecter (**Figure 3b**).

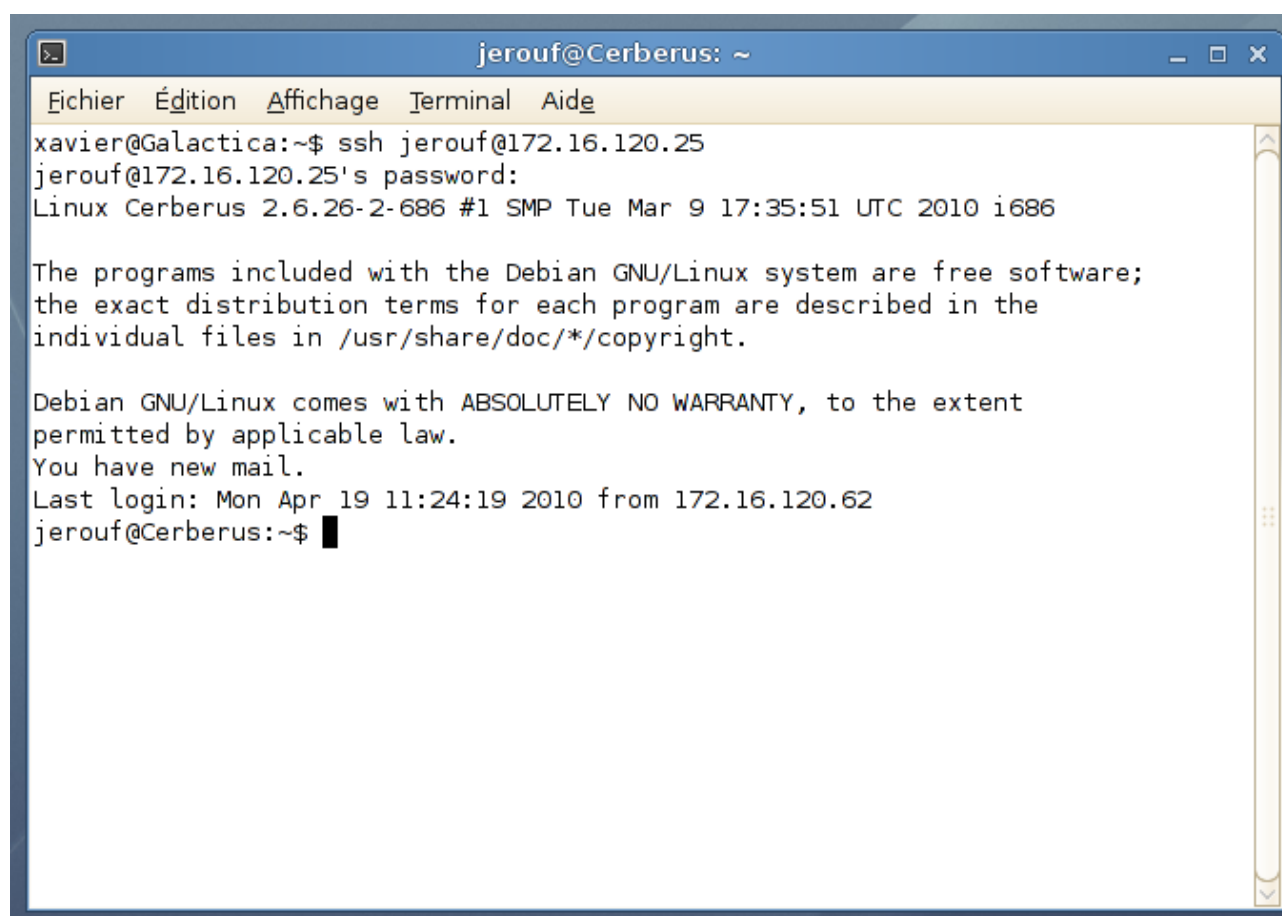


Figure 3b : Connexion en SSH

III. Installation du CMS

1. Définition et choix

Il y a quelques années, le seul moyen de construire et mettre à jour un site web était d'éditer soit-même les fichiers HTML. Toute mise à jour demandait de travailler dans le code source et de télécharger-uploader sans arrêt des pages.

Grâce aux possibilités offertes par des langages comme PHP, les CMS – Content Management System – les modifications sont devenues beaucoup plus simples. L'utilisateur accède à une interface de gestion du site, sur laquelle il peut ajouter/modifier des articles, créer des catégories, générer des miniatures d'images (**Figure4**) ...

Parmi les CMS les plus célèbres on peut citer Dotclear, Wordpress, Drupal, spip... les deux premiers sont plutôt optimisés pour gérer des blogs. Nous avons donc choisi Drupal qui est très complet pour gérer un véritable site web. Il nécessite PHP et Mysql.

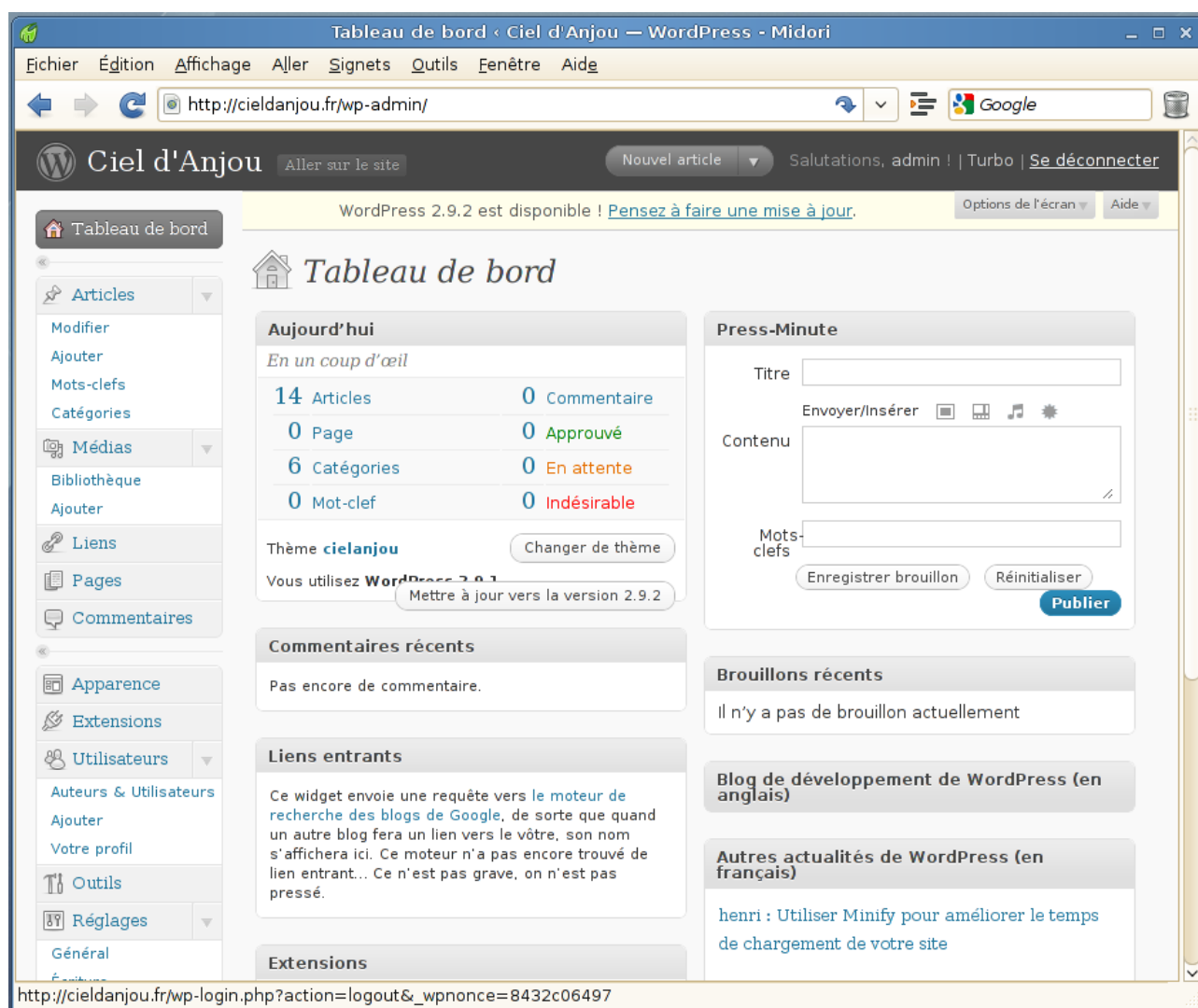


Figure 4 : Exemple de CMS - Wordpress

2. Installation

Nous allons télécharger [Drupal](#) sur le site officiel ainsi que le [pack de traduction FR](#), le tout depuis la machine cliente. A l'aide de Filezilla, nous nous connectons en FTP sur notre serveur et nous envoyons le contenu de l'archive Drupal. Ensuite, nous envoyons le contenu du pack de traduction FR, toujours à la racine.

■ Configuration

Drupal nécessite une base de données, nous allons donc en créer une. Mais pour le moment, sur notre serveur Mysql il n'y a que l'utilisateur root. Comme il est mal avisé de travailler en root, nous allons tout d'abord créer un utilisateur limité auquel nous affecterons une base de données.

A l'aide de PhpMyAdmin, nous nous connectons en root dans le serveur Mysql. Puis nous créons un utilisateur nommé « drupal » auquel nous donnons des droits très limités (Figure5).

Ensuite, toujours connectés en root, nous créons une nouvelle base de donnée nommée « drupal ». Puis nous attribuons des droits sur cette base pour l'utilisateur nouvellement créé (Figure5b).

■ Problème de droits

Une page d'erreur s'affiche lorsque l'on tape l'url du serveur dans le navigateur. L'explication est simple : le serveur lighttpd n'a pas les droits pour lire dans le dossier où est drupal. Pour cela nous procédons en deux étapes :

► Ajout de l'utilisateur du serveur HTTP dans le groupe de l'utilisateur

Dans notre exemple l'utilisateur est « jerouf ». Le serveur web utilise « www-data ». Donc :

```
# adduser www-data jerouf
```

► Modifier les permissions du dossier public_html

Par défaut il n'y a que le propriétaire qui a des droits sur public_html. Nous allons en donner aussi au groupe, et ce de manière récursive sur tout le contenu :

```
$ chmod -R 770 public_html
```

Note : Dans notre cas, il n'y a que l'utilisateur « jerouf » sur la machine, et nous n'avons pas prévu d'en rajouter. Donc nous pouvons aussi travailler en 777.

■ Paramétrage

Par la suite, nous tapons l'adresse IP de notre serveur dans le navigateur et nous arrivons sur l'assistant de configuration de Drupal. Celui-ci nous demande les paramètres de notre serveur : base de données, utilisateur, mot de passe...

Drupal est maintenant installé (Figure 6).

Ajouter un utilisateur

Information pour la connexion

Nom d'utilisateur: drupal

Serveur: localhost

Mot de passe:

Entrer à nouveau:

Générer un mot de passe:

Base de données pour cet utilisateur

Aucune

Créer une base portant son nom et donner à cet utilisateur tous les privilèges sur cette base


Donner les privilèges passepartout ("%")

Privilèges globaux (Tout cocher / Tout décocher)

Veillez noter que les noms de privilèges sont exprimés en anglais

Données	Structure	Administration	Limites de ressources.
<input checked="" type="checkbox"/> SELECT <input checked="" type="checkbox"/> INSERT <input checked="" type="checkbox"/> UPDATE <input checked="" type="checkbox"/> DELETE <input checked="" type="checkbox"/> FILE	<input checked="" type="checkbox"/> CREATE <input checked="" type="checkbox"/> ALTER <input checked="" type="checkbox"/> INDEX <input checked="" type="checkbox"/> DROP <input checked="" type="checkbox"/> CREATE TEMPORARY TABLES <input checked="" type="checkbox"/> CREATE VIEW <input checked="" type="checkbox"/> SHOW VIEW <input checked="" type="checkbox"/> CREATE ROUTINE <input checked="" type="checkbox"/> ALTER ROUTINE <input checked="" type="checkbox"/> EXECUTE	<input type="checkbox"/> GRANT <input type="checkbox"/> SUPER <input type="checkbox"/> PROCESS <input type="checkbox"/> RELOAD <input type="checkbox"/> SHUTDOWN <input type="checkbox"/> SHOW DATABASES <input type="checkbox"/> LOCK TABLES <input type="checkbox"/> REFERENCES <input type="checkbox"/> REPLICATION CLIENT <input type="checkbox"/> REPLICATION SLAVE <input type="checkbox"/> CREATE USER	<p><i>Note: Une valeur de 0 (zero) enlève la limite.</i></p> <p>MAX QUERIES PER HOUR <input type="text" value="0"/></p> <p>MAX UPDATES PER HOUR <input type="text" value="0"/></p> <p>MAX CONNECTIONS PER HOUR <input type="text" value="0"/></p> <p>MAX USER_CONNECTIONS <input type="text" value="0"/></p>

Figure 5 : création d'un utilisateur Mysql aux droits limités

 Utilisateur '**drupal**'@'localhost' : Changer les privilèges

Privilèges globaux (Tout cocher / Tout décocher)


Veillez noter que les noms de privilèges sont exprimés en anglais

Données	Structure	Administration	Limites de ressources.
<input checked="" type="checkbox"/> SELECT <input checked="" type="checkbox"/> INSERT <input checked="" type="checkbox"/> UPDATE <input checked="" type="checkbox"/> DELETE <input checked="" type="checkbox"/> FILE	<input checked="" type="checkbox"/> CREATE <input checked="" type="checkbox"/> ALTER <input checked="" type="checkbox"/> INDEX <input checked="" type="checkbox"/> DROP <input checked="" type="checkbox"/> CREATE TEMPORARY TABLES <input checked="" type="checkbox"/> CREATE VIEW <input checked="" type="checkbox"/> SHOW VIEW <input checked="" type="checkbox"/> CREATE ROUTINE <input checked="" type="checkbox"/> ALTER ROUTINE <input checked="" type="checkbox"/> EXECUTE	<input type="checkbox"/> GRANT <input type="checkbox"/> SUPER <input type="checkbox"/> PROCESS <input type="checkbox"/> RELOAD <input type="checkbox"/> SHUTDOWN <input type="checkbox"/> SHOW DATABASES <input type="checkbox"/> LOCK TABLES <input type="checkbox"/> REFERENCES <input type="checkbox"/> REPLICATION CLIENT <input type="checkbox"/> REPLICATION SLAVE <input type="checkbox"/> CREATE USER	<p><i>Note: Une valeur de 0 (zero) enlève la limite.</i></p> <p>MAX QUERIES PER HOUR <input type="text" value="0"/></p> <p>MAX UPDATES PER HOUR <input type="text" value="0"/></p> <p>MAX CONNECTIONS PER HOUR <input type="text" value="0"/></p> <p>MAX USER_CONNECTIONS <input type="text" value="0"/></p>

Privilèges spécifiques à une base de données

Base de données | **Privilèges** | **"Grant"** | **Privilèges spécifiques à une table** | **Action**

aucune

Ajouter des privilèges sur cette base de données: Entrez une valeur: 

Modifier le mot de passe

aucun mot de passe

Mot de passe: Entrer à nouveau:

Hachage du mot de passe: MySQL 4.1+ compatible MySQL 4.0

Figure 5b : Donner des droits pour l'utilisateur « drupal » sur la page « drupal »

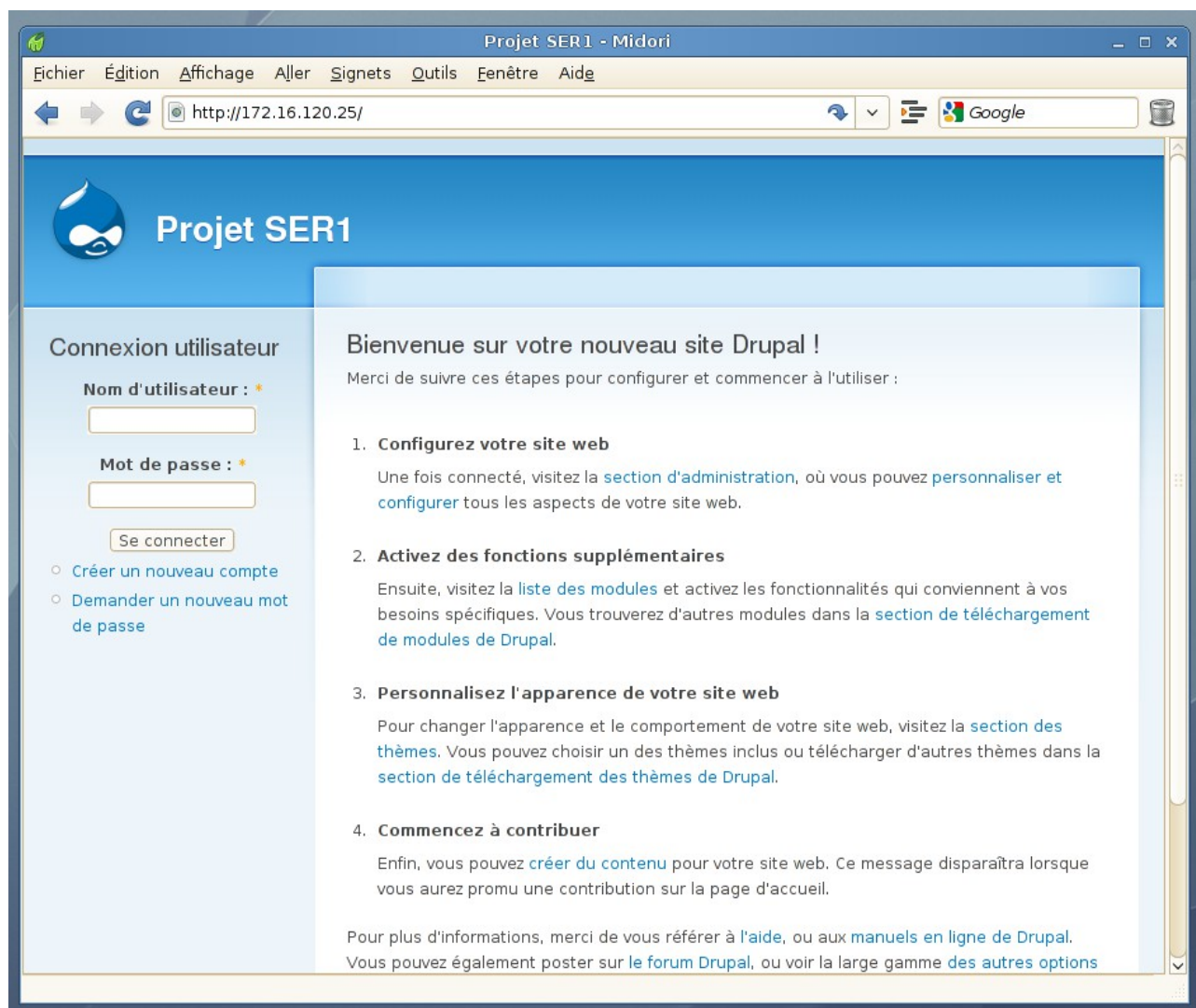


Figure 6 : Drupal fonctionne correctement

III. Sécurisation

1. Introduction

Notre serveur fonctionne, mais si on jette un coup d'œil à la configuration de iptables (Figure 7), on se rend compte qu'il accepte tout et ne filtre rien, ce qui est un comportement suicidaire dans un réseau public comme internet.

```
SER:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
SER:~#
```

Figure7 : Par défaut, iptables laisse tout passer

Nous devons donc définir le comportement idéal et sécurisé, qui fasse en sorte que notre serveur soit tout de même accessible.

■ Liste des services utilisés

Nom du service	Port utilisé	Protocole	Type
HTTP	80	TCP	Entrant
SSH	22	TCP	Entrant
FTP	21	TCP	Entrant
FTP transfert passif	40000-40100	TCP	Entrant
(Loopback)	N/A	N/A	N/A

La configuration de iptables étant volatile, nous allons créer un script s'exécutant au démarrage pour réitérer la configuration à chaque fois.

nano firewall

```
#!/bin/bash
#
#
## Configuration parefeu ##
#
# Variables #
#
ipt="/sbin/iptables"
lan="eth0"

# Module pour supporter le FTP passif #
modprobe ip_conntrack_ftp

# Vidage tables #
iptables -F
iptables -X
iptables -t nat -F
iptables -t nat -X
iptables -t mangle -F
iptables -t mangle -X

# Politique par default #
$Ipt -P INPUT DROP
$Ipt -P OUTPUT DROP
$Ipt -P FORWARD DROP

# Regle 0: Filtrage internet #
$Ipt -A OUTPUT -o $lan -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
$Ipt -A INPUT -i $lan -m state --state ESTABLISHED,RELATED -j ACCEPT

# Regle1: loopback #
$Ipt -A INPUT -i lo -j ACCEPT
$Ipt -A OUTPUT -o lo -j ACCEPT

# Regle2: SSH #
$Ipt -A INPUT -i $lan -p tcp --dport 22 -m state --state NEW,ESTABLISHED
-j ACCEPT

# Regle3: HTTP #
$Ipt -A INPUT -i $lan -p tcp --dport 80 -m state --state NEW,ESTABLISHED
-j ACCEPT

# Regle 4: FTP #
$Ipt -A INPUT -i $lan -p tcp --dport 21 -m state --state NEW,ESTABLISHED
-j ACCEPT
$Ipt -A INPUT -i $lan -p tcp --dport 40000:40100 -m state --state
NEW,ESTABLISHED -j ACCEPT
```

Rendre le script exécutable :

```
# chmod +x firewall
```

L'exécuter pour tester :

```
# ./firewall
```

Si tout est bon, nous allons automatiser le lancement du script à chaque démarrage:

Le copier dans le répertoire des services :

```
# cp firewall /etc/init.d/
```

Le rajouter au démarrage :

```
# update-rc.d firewall defaults
```


IV. Améliorations de sécurité

1. Sécurité des données

Pour assurer la sécurité des données, nous allons mettre en place un système de sauvegardes automatisé sur un autre ordinateur. Nous allons commencer par faire un partage NFS (dossier sur le réseau) et ensuite nous ferons un script CRON qui automatisera cela.

► Partage NFS

Côté serveur NFS :

```
# apt-get install nfs-kernel-server
# nano /etc/default/nfs-kernel-server
```

(On laisse de côté la partie sécurisation et parefeu sur cette machine, car on suppose qu'elle est gérée par un autre administrateur. On ne s'occupe que du partage NFS)

Remplacer :

```
RPMOUNTOPTS=--manage-gids
```

Par :

```
RPMOUNTOPTS="--manage-gids --port 4000"
```

```
$ mkdir /home/xavier/nfs
```

Qui est propriétaire du dossier nouvellement créé ?

```
$ ls -l
```

Retourne :

```
drwxr-xr-x 2 xavier xavier 4096 avr 19 10:15 xavier
```

Donc il appartient à « xavier »

Quel est l'id de xavier ?

```
# id xavier
```

Donne :

```
uid=1000(xavier) gid=1000(xavier)
```

Cette information sera utile pour définir les permissions d'écriture sur le partage.

```
# nano /etc/exports
```

Ajouter :

```
/home/xavier/nfs 172.16.120.25(rw,anonuid=1000,anongid=1000)
```

Le dossier partagé est « /home/xavier/nfs » et lorsque quelqu'un écrira dedans (à distance, depuis NFS) les fichiers ajoutés appartiendront à xavier et au groupe xavier.

Côté client :

```
# mount -t nfs 172.16.120.11:/home/xavier/xavier /mnt/serveur
```

Le partage réseau est monté dans le dossier /mnt/serveur.

Mais ce partage est temporaire, si le serveur redémarre, il sera perdu. Pour automatiser le montage, il faut ajouter une entrée dans le fichier /etc/fstab :

```
# nano /etc/fstab
```

Et ajouter :

```
172.16.120.30:/home/xavier/nfs /mnt/serveur nfs defaults 0 0
```

► Automatiser la tâche

(Côté serveur) :

```
# apt-get install p7zip-full  
# crontab -e
```

Avec le contenu :

```
SHELL=/bin/sh  
  
# m h dom mon dow command  
  
* 2 * * * /root/sav
```

Puis :

```
# nano /root/sav
```

Le contenu du script est reporté sur la page suivante.

Puis :

```
chmod +x /root/sav
```

Fonctionnement : tous les jours, à 2h du matin, le script /root/sav est exécuté. Ce script compresse au format 7z le dossier home/jerouf/public_html puis fait ensuite une sauvegarde de la base de données « drupal ». Le tout est envoyé sur le répertoire /mnt/serveur (partage NFS).

```
#!/bin/bash
#
## Script de sauvegarde ##
#
## Sauvegarde des fichiers du site ##
#
cd /home/jerouf/public_html
7z a /mnt/serveur/"sav-`date +%d-%m-%Y`".7z *
#
## Sauvegarde de la base de données ##
#
Mysql_User="root"
Mysql_Paswd="overrated"
Mysql_host="localhost"
MYSQL="$ (which mysql) "
MYSQLDUMP="$ (which mysqldump) "
CHOWN="$ (which chown) "
CHMOD="$ (which chmod) "
GZIP="$ (which gzip) "

# Emplacement du dossier de backup local
DEST="/mnt/serveur"

#Rep ou on met le sql
DEST_mysql="$DEST/mysql"

#Date du jour
NOW="$ (date +%d-%m-%Y) "

# Databases a ne pas sauvegarder separer par des espaces
IGGY="drupal"

# On initialise les variables
FILE=""
DBS=""

#on cree le rep
[ ! -d $DEST_mysql ] && mkdir -p $DEST_mysql || :

# On liste les bases de donnees
DBS="$ ( $MYSQL -u $Mysql_User -h $Mysql_host -p$Mysql_Paswd -Bse 'show
databases' ) "

for db in $DBS
do
    skipdb=-1
    if [ "$IGGY" != "" ];
    then
        for i in $IGGY
        do
            [ "$db" == "$i" ] && skipdb=1 || :
        done
    fi

    if [ "$skipdb" == "-1" ] ; then
        FILE="$DEST_mysql/$db.$NOW.gz"
        # On boucle, et on dump toutes les bases et on les compresse
        $MYSQLDUMP -u $Mysql_User -h $Mysql_host -p$Mysql_Paswd $db | $GZIP -9 >
$FILE
    fi
done
```

2. Sécurité du Réseau

Pour minimiser les risques d'attaques, nous allons améliorer la sécurité des services par divers moyens :

- Changer le port par défaut de SSH
- Désactiver le login root dans SSH
- Utiliser fail2ban pour contrer les attaque brute-force

1. Port et login SSH

```
# nano /etc/ssh/sshd_config
```

Trouver la ligne :

```
Port 22
```

Et remplacer par exemple par :

```
Port 7666
```

Puis en bas on rajoute :

```
AllowUsers jerouf
```

Attention: après ces modifs, il ne faut pas oublier de faire les modifications dans le script de configuration de iptables. La ligne pour SSH doit ouvrir le port 7666 et non plus le 22.

Pour se connecter au serveur, on utilisera maintenant la commande :

```
$ ssh jerouf@172.16.120.25 -p 7666
```

2. Fail2ban

```
# apt-get install fail2ban
```

La configuration est située dans `/etc/fail2ban/jail.conf`
Par défaut le service SSH est protégé

Il y a également un paragraphe pour la protection de vsftpd. Il nous suffit de passer la valeur :

```
enabled = false
```

En :

```
enabled = true
```

Puisque nous avons changé le port de SSH, dans le paragraphe `[ssh]` il faut remplacer :

```
port = ssh
```

Par :

```
port = 7666
```

Puis relancer le service :

```
# /etc/init.d/fail2ban restart
```

Avec cette configuration, un utilisateur tentant de se connecter en SSH via le port 7666 sera banni si il échoue 6 fois, et ce pour 600 secondes. Même chose pour la connexion FTP sur le port 21.

Lorsqu'une IP est bannie, on peut le voir dans les règles iptables :

```
# iptables -L
```

Retourne :

```
[...]
Chain fail2ban-vsftpd (1 references)
target    prot opt source          destination
DROP     all  --  172.16.120.60    anywhere
RETURN   all  --  anywhere         anywhere
```