

# Rapport " Découverte de Linux " avec L<sup>A</sup>T<sub>E</sub>X

---

Clément MATTHEY - Robin PIERRET

# Contents

<b>1</b>	<b>Présentation</b>	<b>3</b>
1.1	Equipe . . . . .	3
1.2	Sujet . . . . .	3
1.3	Organisation . . . . .	3
<b>2</b>	<b>Mise en place du serveur Apache2</b>	<b>4</b>
2.1	Installation du serveur HTTP . . . . .	4
2.2	Installation du module php . . . . .	4
2.3	Installation du module SQL . . . . .	5
2.4	Installation de PHPmyadmin . . . . .	5
<b>3</b>	<b>Sécurisation du serveur via un firewall</b>	<b>6</b>
3.1	Installation d'Ipstables . . . . .	6
3.2	Configuration des règles . . . . .	6
3.3	Port Sentry . . . . .	7
3.4	Fail2Ban . . . . .	7
3.5	Rkhunter . . . . .	8
3.6	Configuration SSH et Apache . . . . .	8
<b>4</b>	<b>Sources</b>	<b>10</b>

# Chapter 1

## Présentation

### 1.1 Equipe

Notre équipe est formée de Robin PIERRET et Clément MATTHEY, nous sommes tous deux en deuxième année et notre intérêt commun pour l'informatique en générale et plus précisément dans notre cas la découverte d'un nouvel outil nous a incité à travailler ensemble.

### 1.2 Sujet

Nous nous sommes donc intéressé à la mise en place d'un serveur et sa sécurisation notamment grâce à l'usage d'un firewall.

Dans un premier temps nous vous expliquerons les différentes étapes de la mise en place d'un serveur apache puis nous parlerons de sa sécurisation.

Il nous a paru intéressant de nous pencher sur la sécurité où en tout cas au début de la mise en place de la sécurité du service. En effet, un site peut fréquemment subir des attaques et donc nous sommes d'avis qu'il est important de savoir protéger, même de façon sommaire notre travail. Pour cela, il est nécessaire de connaître les principes de base et c'est ce sur quoi nous avons voulu travailler pour ce rapport.

### 1.3 Organisation

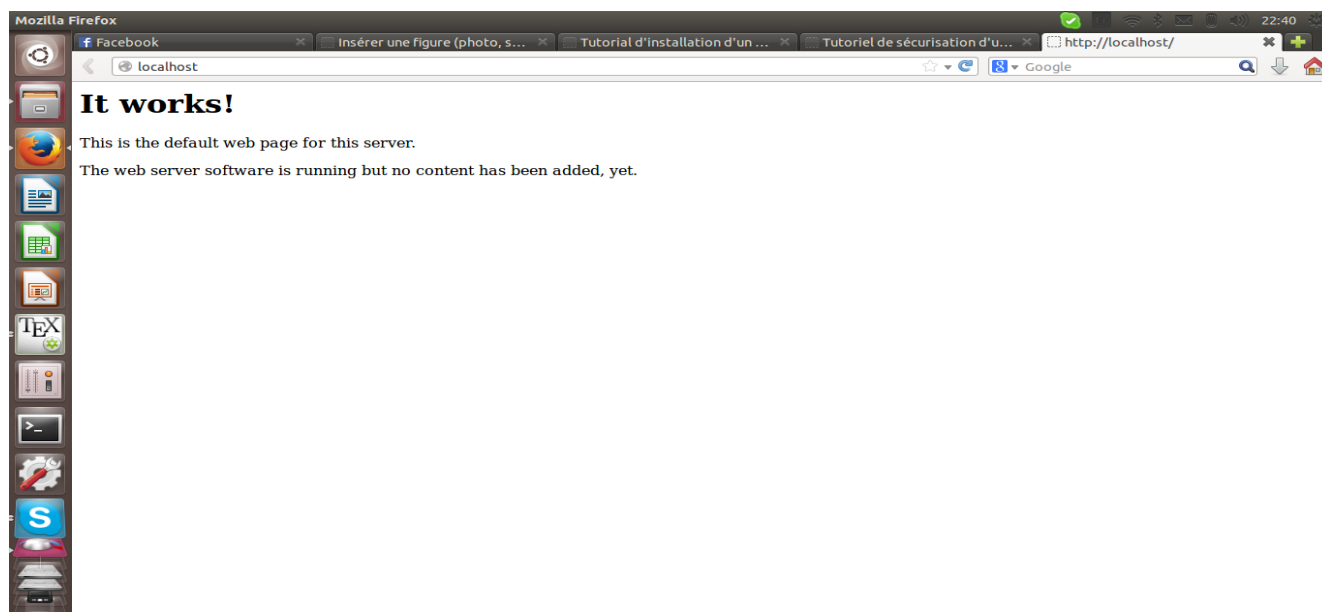
Au vu du délai accordé, nous avons choisi d'orienté notre travail sur deux axes principaux. Initialement nous pensions avoir plus de temps et donc la possibilité d'explorer d'autre facette concernant les serveurs, mais nous avons finalement choisis de nous recentrer sur des thèmes précis afin de pouvoir creuser plus profondément dans ce sujet.

## Chapter 2

# Mise en place du serveur Apache2

### 2.1 Installation du serveur HTTP

Pour installer un serveur sur nos distributions, nous avons installé apache2. Pour cela nous avons lancé un terminal et tapé : "sudo apt-get install apache2". Nous avons ensuite testé celui-ci en allant dans notre navigateur et en tapant : "http://localhost" Nous avons remarqué que les paramètres de base suffisaient pour la suite de notre travail sur la sécurité. Ceux-ci ont été configuré par le biais de l'installation (dans le fichier apache2.conf)



### 2.2 Installation du module php

Concernant le php, nous avons choisis de travailler avec la dernière version de celui-ci disponible sur les repository, nous avons donc installé php5. Pour cela nous avons ouvert un terminal et avons tapé : "apt-get install libapache2-mod-php5 php5-mysql php5-gd php5-cli"

**libapache2-mod-php5:**Ce paquet fournit le module PHP 5 pour le serveur web Apache 2 (celui trouvé dans le paquet apache2-mpm-prefork). Ce paquet fonctionne uniquement avec le serveur web Apache 2 prefork MPM car il n'est pas **thread-safe**(il ne peut pas travailler en multi-thread)

**php5-mysql:**Ce paquet fournit les modules pour se connecter directement aux bases de données MySQL avec des scripts PHP. Il contient le module générique "mysql" utilisé pour se connecter à toute version de MySQL, le

module amélioré "mysqli" pour les versions 4.1 et supérieures de MySQL et le module "pdo\_mysql" pour utiliser les extensions PDO (PHP Data Object).

**php5-gd:**Ce paquet fournit un module pour manipuler des images directement depuis des scripts PHP. Il gère les formats PNG, JPEG, XPM ainsi que les polices Freetype et ttf.

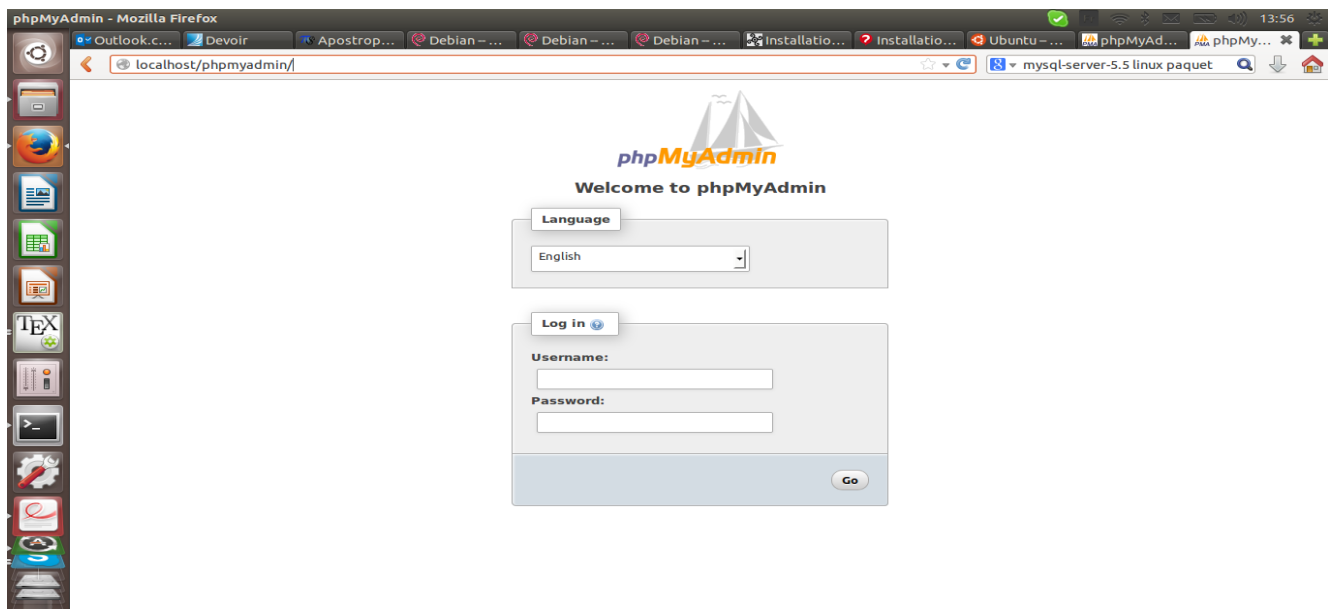
**php5-cli:**Ce paquet fournit l'interpréteur de commande /usr/bin/php5, utile pour tester des scripts PHP depuis la ligne de commande.

## 2.3 Installation du module SQL

Pour installer le module MySQL, il suffit de taper dans un terminal : "sudo apt-get install mysql-server-5.5" avec 5.5 la version du SGBD que nous voulons installer. Il est tout à fait possible de travailler avec une version antérieure de ce gestionnaire, mais il reste tout de même plus sûr de travailler avec une version récente afin de profiter de toutes les fonctionnalités et de toutes les mises à jour de sécurité. Ce module nous permet donc d'avoir une application de gestion de base de données en local ce qui peut être utile pour notre serveur.

## 2.4 Installation de PHPmyadmin

PHPmyadmin est un outil permettant de gérer de façon aisée les bases de données. Pour l'installer il faut entrer dans un terminal et taper : "sudo apt-get install phpmyadmin" Une fois fait, il est possible d'accéder à ce logiciel en tapant dans un navigateur : "http://localhost/phpmyadmin/"



## Chapter 3

# Sécurisation du serveur via un firewall

### 3.1 Installation d'Iptables

Le logiciel Iptables va nous permettre de gérer et de configurer les règles du pare-feu déjà présent dans le noyau. C'est grâce à ce logiciel que nous allons créer les règles de notre Firewall, ce qui est indispensable pour commencer à sécuriser le serveur. Il va être en charge d'observer et d'analyser le trafic sur le serveur.

Les règles que l'ont écrit s'appliquent sur les adresses IP, sur les différents accès (ports) et sur les différents protocoles (udp, tcp, etc). On peut ainsi assurer le bon fonctionnement de chacun des ports et surtout contrôler qui rentre et qui sort. Il faut également penser à ne pas se bloquer l'accès à soi-même.

### 3.2 Configuration des règles

Pour configurer chacune des règles que nous avons décidé de d'appliquer à notre serveur, nous avons créer un script (qui est ci-joint (Script Firewall) dans le .zip). Ce script a été demander par la ligne de commande suivante : `nano /etc/init.d/firewall`. C'est à l'intérieur de ce script que nous créons un pare-feu simple qui va contrôler les entrées et les sorties en autorisant seulement certains ports d'entrées. Il est également possible de bannir certaines adresses IP pour éviter des attaques récidivistes(Notons cependant que ces sécurités sont basiques).

Exemple de règle : `sudo iptables -t filter -A INPUT -p tcp --dport 40 -j ACCEPT`

Pour chacune des règles, nous avons dû les demandées en nous plaçant en super utilisateur pour être certain d'avoir tous les droits nécessaires. Dans cette règles, nous filtrons (autorisons) les entrées pour le port 40, avec le protocole tcp. Il faut également observer que l'on se doit de d'autoriser des entrées et des sorties(INPUT/OUTPUT).

```
linux@vm-ubuntu-64: ~
GNU nano 2.2.6          Fichier : /etc/init.d/firewall          Modifié

# Autorise ICMP avec son protocole sans port défini
sudo iptables -t filter -A INPUT -p icmp -j ACCEPT
sudo iptables -t filter -A OUTPUT -p icmp -j ACCEPT

# Autorise le SSH avec son port (40 ici port habituel : 22, on le modifie pour compliquer les attaques) et son protocole (tcp)
sudo iptables -t filter -A INPUT -p tcp --dport 40 -j ACCEPT
sudo iptables -t filter -A OUTPUT -p tcp --dport 40 -j ACCEPT

# Autorise le DNS avec ses 2 protocoles (tcp et udp) et son port 53
sudo iptables -t filter -A INPUT -p tcp --dport 53 -j ACCEPT
sudo iptables -t filter -A INPUT -p udp --dport 53 -j ACCEPT
sudo iptables -t filter -A OUTPUT -p tcp --dport 53 -j ACCEPT
sudo iptables -t filter -A OUTPUT -p udp --dport 53 -j ACCEPT

# Autorise le HTTP avec son port (80) et son protocole (tcp)
sudo iptables -t filter -A INPUT -p tcp --dport 80 -j ACCEPT
sudo iptables -t filter -A OUTPUT -p tcp --dport 80 -j ACCEPT

# Autorise le FTP : port (20 ou 21) protocole (tcp)
sudo iptables -t filter -A INPUT -p tcp --dport 20:21 -j ACCEPT
sudo iptables -t filter -A OUTPUT -p tcp --dport 20:21 -j ACCEPT

# Autorise le Mail SMTP : port (25) protocole (tcp)
sudo iptables -t filter -A INPUT -p tcp --dport 25 -j ACCEPT
sudo iptables -t filter -A OUTPUT -p tcp --dport 25 -j ACCEPT

# Autorise le Mail POP3 : port (110) protocole (tcp)
sudo iptables -t filter -A INPUT -p tcp --dport 110 -j ACCEPT
sudo iptables -t filter -A OUTPUT -p tcp --dport 110 -j ACCEPT

^G Aide          ^O Écrire      ^R Lire fich.   ^V Page préc.   ^K Couper      ^C Pos. cur.
^X Quitter      ^J Justifier   ^W Chercher    ^V Page suiv.  ^U Coller      ^T Orthograp.
```

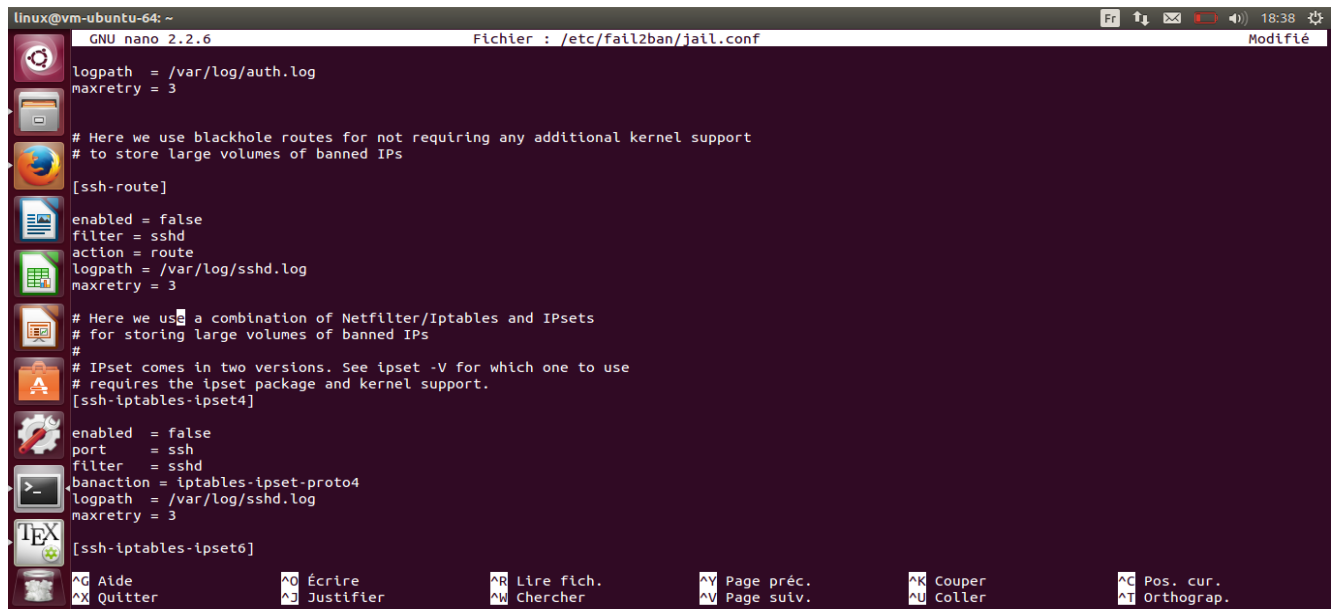
### 3.3 Portsentry

Le portsentry permet de gérer les scans de ports en temps réel, il a pour fonction de les bloquer. Le scan des ports permet de connaître les différents accès à notre serveur mais si la sécurisation du serveur est optimale, il n'y a aucun danger. Tout le monde peut lancer un scan des ports de serveur. Cependant Portsentry va bloquer ces scans ce qui va permettre de faire croire à l'assaillant que les ports ouverts sont fermés. Il faut lancer le logiciel sur chaque protocole utilisés par les différents ports.

### 3.4 Fail2Ban

Fail2ban est un outil permettant de contrôler les actions effectuées sur les ports, par exemple, il va détecter si une personne tente de rentrer beaucoup de mot de passe pour entrer sur le serveur, Fail2ban va le bloquer et même l'empêcher de se reconnecter au serveur pendant un certain temps défini par le concepteur du système.

Les principales commandes que nous avons modifiées sont : bantime et maxretry. C'est deux commandes permettant de gérer le temps de bannissement des utilisateurs suspects et maxretry est le nombre de tentative autorisée pour chaque utilisateur.



```
linux@vm-ubuntu-64: ~
GNU nano 2.2.6                                Fichier : /etc/fail2ban/jail.conf                                Modifié

logpath = /var/log/auth.log
maxretry = 3

# Here we use blackhole routes for not requiring any additional kernel support
# to store large volumes of banned IPs
[ssh-route]
enabled = false
filter = sshd
action = route
logpath = /var/log/sshd.log
maxretry = 3

# Here we use a combination of Netfilter/Iptables and IPsets
# for storing large volumes of banned IPs
#
# IPset comes in two versions. See ipset -V for which one to use
# requires the ipset package and kernel support.
[ssh-iptables-ipset4]
enabled = false
port = ssh
filter = sshd
banaction = iptables-ipset-proto4
logpath = /var/log/sshd.log
maxretry = 3

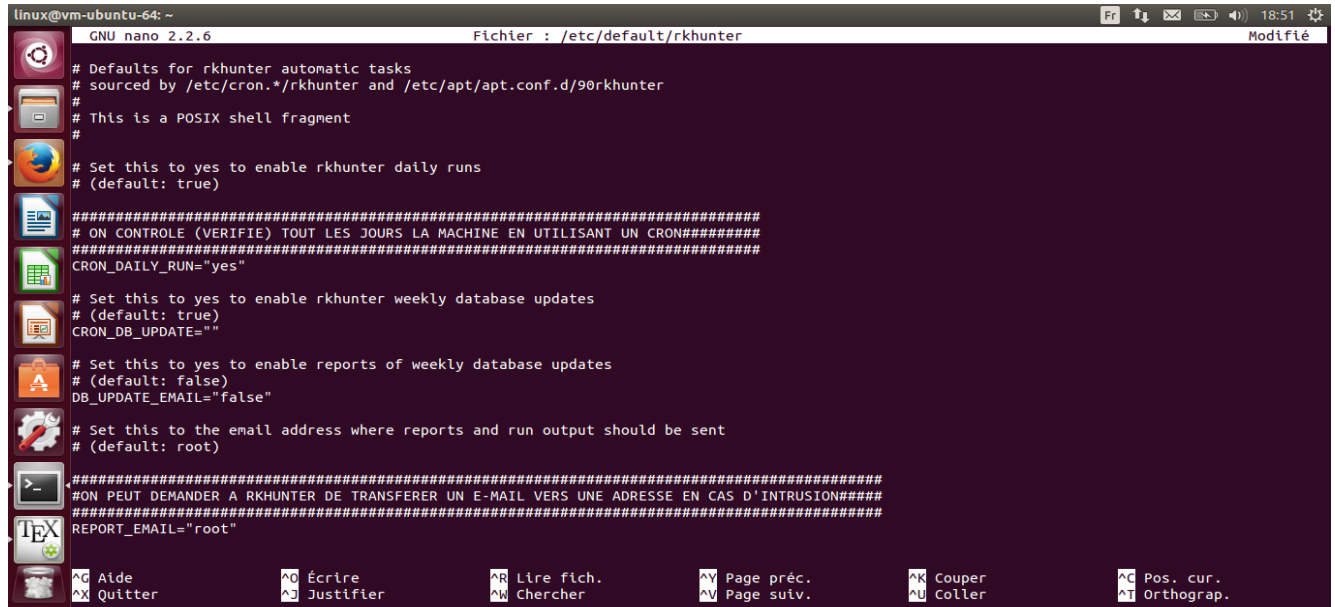
[ssh-iptables-ipset6]

^G Aide          ^O Écrire       ^R Lire fich.   ^Y Page préc.   ^K Couper
^X Quitter       ^J Justifier    ^M Chercher     ^V Page suiv.  ^U Coller
^_ Pos. cur.     ^T Orthograp.
```



## 3.5 Rkhunter

Rkhunter est un autre logiciel que nous avons utilisé pour sécuriser notre serveur. C'est un nettoyeur de porte dérobée c'est-à-dire qu'il va détecter une intrusion déjà passé et avertir le concepteur de cette intrusion et bloquer tous les accès laissés par l'attaquant afin de garder un accès rapide à votre ordinateur. Attention toutes fois à la précision de Rkhunter qui n'est pas fiable à 100 pourcent (erreur de détection) mais qui assure toutes fois de détecter les intrusions. Rkhunter peut prévenir le concepteur par mail si on lui demande grâce à la commande REPORT\_EMAIL.



```
linux@vm-ubuntu-64: ~
GNU nano 2.2.6                                Fichier : /etc/default/rkhunter
# Defaults for rkhunter automatic tasks
# sourced by /etc/cron.*/rkhunter and /etc/apt/apt.conf.d/90rkhunter
# This is a POSIX shell fragment
#
# Set this to yes to enable rkhunter daily runs
# (default: true)
#####
# ON CONTROLE (VERIFIE) TOUT LES JOURS LA MACHINE EN UTILISANT UN CRON#####
#####
CRON_DAILY_RUN="yes"
# Set this to yes to enable rkhunter weekly database updates
# (default: true)
CRON_DB_UPDATE=""
# Set this to yes to enable reports of weekly database updates
# (default: false)
DB_UPDATE_EMAIL="false"
# Set this to the email address where reports and run output should be sent
# (default: root)
#####
#ON PEUT DEMANDER A RKHUNTER DE TRANSFERER UN E-MAIL VERS UNE ADRESSE EN CAS D'INTRUSION####
#####
REPORT_EMAIL="root"
^G Aide          ^O Écrire      ^R Lire fich.   ^V Page préc.   ^K Couper      ^C Pos. cur.
^X Quitter      ^J Justifier   ^M Chercher    ^V Page suiv.  ^U Coller      ^T Orthograp.
```

### 3.6 Configuration SSH et Apache

Enfin, pour optimiser la sécurité du serveur, nous avons décidé de modifier certains paramètres de base. Effectivement, on remarque que quand on tente d'attaquer un serveur, on teste en tout premier lieu, les ports habituels (par exemple le port ssh très souvent associé à 22). Pour empêcher une attaque, il faut que la sécurité semble compliquée et infranchissable mais pas forcément quelle le soit. Puis nous avons ajouté deux paramètres pour optimiser la discrétion du serveur (Serversignature et Servertokens)

```
linux@vm-ubuntu-64: ~  
GNU nano 2.2.6 Fichier : /etc/apache2/apache2.conf Modifié  
# * The binary is called apache2. Due to the use of environment variables, in  
# the default configuration, apache2 needs to be started/stopped with  
# /etc/init.d/apache2 or apache2ctl. Calling /usr/bin/apache2 directly will not  
# work with the default configuration.  
  
# Global configuration  
#  
#####  
### AFIN D'OPTIMISER LA DISCRETION DE NOTRE SERVEUR, NOUS UTILISONS LES DEUX COMMANDES SUIVANTES : ###  
#####  
ServerSignature off  
ServerTokens Prod  
  
#  
# ServerRoot: The top of the directory tree under which the server's  
# configuration, error, and log files are kept.  
  
# NOTE! If you intend to place this on an NFS (or otherwise network)  
# mounted filesystem then please read the Mutex documentation (available  
# at <URL:http://httpd.apache.org/docs/2.4/mod/core.html#mutex>);  
# you will save yourself a lot of trouble.  
#  
# Do NOT add a slash at the end of the directory path.  
#  
#ServerRoot "/etc/apache2"  
#  
  
Alde Écrire Lire fich. Page préc. Couper  
Quitter Justifier Chercher Page suiv. Collier  
Pos. cur.  
Orthograp.
```

# Chapter 4

## Sources

<https://packages.debian.org/fr>  
<http://httpd.apache.org/docs/2.0/fr/>  
<http://www.linux-france.org/prj/edu/archinet/systeme/ch16s02.html>  
<http://giminik.developpez.com/articles/apache/debian/>  
<http://www.formation-web.org/configuration-serveur-web-linux/>  
<http://fr.openclassrooms.com/informatique/cours/securiser-son-serveur-linux>  
<http://doc.ubuntu-fr.org/apache2>