Serveur - HOWTO

par Cédric LEULLIER <http://cedric.leullier.free.fr/> $Id: serveur-howto.sgml,v 1.2 2008/03/14 $

Tout pour configurer mon serveur sous Linux.

Contents

1 Introduction

1.1 Ce document

A propos de ce document...

1.2 Versions

Vous pourrez trouver la dernière version de ce document à l'adresse <http://cedric.leullier.free.fr/linux/howto/Mon_Serveur/Mon_Serveur-HOWTO.html> .

Pour une version PDF, voir : <http://cedric.leullier.free.fr/linux/howto/Mon_Serveur/Mon_Serveur-HOWTO.pdf> .

Pour le fichier source au format SGML, téléchargez : <http://cedric.leullier.free.fr/linux/howto/Mon_Serveur/Mon_Serveur-HOWTO.sgml> .

Les versions HTML et PDF ont été respectivement générées à partir du fichier source par les commandes :

```
linuxdoc --backend="html" --charset="latin" Mon_Serveur-HOWTO.sgml
linuxdoc --backend="latex" --output="pdf" --charset="latin" Mon_Serveur-HOWTO.sgml
```

1.3 Ma configuration Linux

Le serveur est basé sur un système d'exploitation de type LINUX. C'est une distribution GNU/Linux DEBIAN Etch (stable) avec un kernel 2.6.18.

2 Configuration du réseau

2.1 Configuration de l'adresse IP

Editez le fichier `/etc/network/interfaces` comme suit, en choisissant si vous voulez une attribution automatique de l'adresse IP par le protocole DHCP ou si vous voulez fixer vous-même cette adresse :

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface with DHCP
#auto eth0
#iface eth0 inet dhcp

# The primary network interface with Static IP
auto eth0
iface eth0 inet static
address 192.168.0.3
netmask 255.255.255.0
network 192.168.0.0
broadcast 192.168.0.255
gateway 192.168.0.1
```

2.2 Configuration du nom d'hôte

Editez le fichier `/etc/hostname` et indiquez le nom de votre serveur :

```
nuts.fr
```

2.3 Configuration du résolveur de noms

Editez le fichier `/etc/resolv.conf` et indiquez la ou les adresse(s) IP du ou des serveur(s) de noms (DNS) :

```
search
nameserver 192.168.0.1
```

2.4 Démarrage du service réseau

En tant que superutilisateur (utilisateur *root*), entrez les commandes suivantes dans une console :

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

pour que le système mette à jour la nouvelle configuration du réseau et active ce dernier.

2.5 Vérifications

2.5.1 Vérification de l'interface réseau

Pour vérifier que l'interface réseau est fonctionnelle, entrez les commandes suivantes dans une console :

```
# ifconfig
eth0      Lien encap:Ethernet  HWaddr 00:01:02:A7:D3:81
          inet adr:192.168.0.3  Bcast:192.168.0.255  Masque:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2151 errors:0 dropped:0 overruns:1 frame:0
          TX packets:1827 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          RX bytes:317173 (309.7 KiB)  TX bytes:577207 (563.6 KiB)
          Interruption:5 Adresse de base:0xa400

lo        Lien encap:Boucle locale
          inet adr:127.0.0.1  Masque:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:78 errors:0 dropped:0 overruns:0 frame:0
          TX packets:78 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:0
          RX bytes:5164 (5.0 KiB)  TX bytes:5164 (5.0 KiB)
```

Vérifiez que le paragraphe *eth0* est bien présent et contient bien les informations que vous avez déclarées.

2.5.2 Vérificaiton du routage des paquets internet

Pour vérifier que le routage des paquets réseau se fait correctement, faire :

```
# route
Table de routage IP du noyau
Destination     Passerelle      Genmask         Indic Metric Ref    Use Iface
192.168.0.0     *               255.255.255.0   U     0      0        0 eth0
default         192.168.0.1     0.0.0.0         UG    0      0        0 eth0
```

La première ligne indique que tout ce qui est à destination du réseau local (192.168.0.0) passe directement (*) par l'interface éthernet *eth0*.

La deuxième ligne indique que tout ce qui est destiné à un autre réseau, dont qui est destiné à passer par internet, passe par la passerelle définie plus haut (192.168.0.1) en utilisant l'interface éthernet *eth0* du serveur. En effet, le serveur pourrait très bien avoir plusieurs cartes réseaux (eth1, eth2, etc...).

2.5.3 Vérification de la résolution des noms de domaines

Pour vérifier que la résolution des noms de domaines se fait correctement, faire :

```
# host www.google.fr
www.google.fr is an alias for www.google.com.
www.google.com is an alias for www.l.google.com.
www.l.google.com has address 209.85.129.147
www.l.google.com has address 209.85.129.99
www.l.google.com has address 209.85.129.104
```

Si rien ne s'affiche, c'est qu'il y a problème.

3 Petits outils divers

3.1 ntpdate

A venir...

3.2 ddclient

A venir...

4 Configuration de la console d'administration

4.1 Introduction

Pour administrer le serveur, nous pouvons soit brancher un écran et un clavier sur ce dernier et l'administrer directement, soit utiliser un accès distant.

Pour pouvoir utiliser un accès distant, nous avons besoin d'un canal sécurisé (crypté). Nous utilisons pour cela *OpenSSH*, "SSH" signifiant "Secure SHell".

4.2 Configuration d'OpenSSH

4.2.1 Installation d'OpenSSH

Nous avons besoin du paquet DEBIAN suivant : *openssh*. Pour cela, en tant que superutilisateur (utilisateur *root*), entrez les commandes suivantes dans une console :

```
# aptitude install openssh
```

et suivez les instructions.

4.2.2 Configuration d'OpenSSH

Modifiez le fichier de configuration `/etc/ssh/sshd_config` comme suit :

```
# What ports, IPs and protocols we listen for
Port 2222

# Use these options to restrict which interfaces/protocols sshd will bind to
#ListenAddress ::
#ListenAddress 0.0.0.0
Protocol 2

# HostKeys for protocol version 2
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key

#Privilege Separation is turned on for security
UsePrivilegeSeparation yes

# Lifetime and size of ephemeral version 1 server key
```

```
KeyRegenerationInterval 3600
ServerKeyBits 768

# Logging
SyslogFacility AUTH
LogLevel INFO

# Authentication:
LoginGraceTime 600
PermitRootLogin yes
StrictModes yes


RSAAuthentication no
PubkeyAuthentication yes
#AuthorizedKeysFile     %h/.ssh/authorized_keys

# Don't read the user's ~/.rhosts and ~/.shosts files
IgnoreRhosts yes
# For this to work you will also need host keys in /etc/ssh_known_hosts
RhostsRSAAuthentication no
# similar for protocol version 2
HostbasedAuthentication no
# Uncomment if you don't trust ~/.ssh/known_hosts for RhostsRSAAuthentication
#IgnoreUserKnownHosts yes

# To enable empty passwords, change to yes (NOT RECOMMENDED)
PermitEmptyPasswords no

# Change to no to disable s/key passwords
#ChallengeResponseAuthentication yes

# Change to yes to enable tunnelled clear text passwords
PasswordAuthentication no


# To change Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#AFSTokenPassing no
#KerberosTicketCleanup no

# Kerberos TGT Passing does only work with the AFS kaserver
#KerberosTgtPassing yes

X11Forwarding no
X11DisplayOffset 10
PrintMotd no
PrintLastLog yes
KeepAlive yes
#UseLogin no

#MaxStartups 10:30:60
#Banner /etc/issue.net
```

```
Subsystem       sftp    /usr/lib/sftp-server

UsePAM yes
```

4.2.3 Lancement d'OpenSSH

Pour lancer le daemon `sshd`, entrez en tant que superutilisateur (utilisateur *root*) la commande suivante :

```
# /etc/init.d/ssh start
```

4.3 Création et échange des clefs de cryptage

A développer....

4.4 Accéder à la console d'administration depuis un poste sous Linux

4.4.1 Installation du logiciel OpenSSH

Voir plus haut.

4.4.2 Se connecter au serveur

Dans une console (shell), entrez la commande suivante :

```
# ssh -p 2222 login@nuts.fr
```

où "login" est à remplacer par le votre. Entrez ensuite votre mot-de-passe.

Vous vous retrouvez maintenant dans le mode console du serveur.

4.5 Accéder à la console d'administration depuis un poste sous Windows XP

4.5.1 Installation du logiciel PuTTY sous Windows XP

Récupérez le logiciel à l'adresse suivante : `http://www.chiark.greenend.org.uk/~sgtatham/putty/` et installez le.

4.5.2 Se connecter au serveur

Lancez le programme `PuTTY`.

Entrez les paramètres suivants :

```
Host Name : serveur
Port : 2222
Protocol : SSH
```

et cliquez sur le bouton "Open".

Vous vous retrouvez maintenant dans le mode console du serveur où il vous faut vous authentifier.

4.6 Continuons...

A partir de maintenant, tout ce qui suit peut se faire à distance depuis un autre ordinateur en utilisant la console d'administration SSH.

5 Serveur FTP

5.1 Introduction

A venir...

5.2 Configuration de vsFTPd

5.2.1 Installation de vsFTPd

Nous avons besoin du paquet DEBIAN suivant : *vsftpd*. Pour cela, en tant que superutilisateur (utilisateur *root*), entrez les commandes suivantes dans une console :

```
# aptitude install vsftpd
```

et suivez les instructions.

5.2.2 Configuration de vsFTPd

Dans ce fichier de configuration, nous déterminons les paramètres généraux du serveur *vsFtpd*, les options de connexion et l'accès des utilisateurs.

Dans la section *Options du Daemon*, nous définissons les ports utilisés par *vsFTPd* (2121 pour le port de contrôle et 2020 pour le port de données).

Dans la section *Options de connexion*, nous autorisons l'utilisateur du mode passif (mode PASV) pour que ce soit le serveur qui impose le numéro de port de données (ici 2020). Et nous interdissons l'utilisation du mode actif (mode PORT) pour empêcher au client de décider du numéro de port de données à utiliser. Ceci permet de limiter l'ouverture des ports au niveau du firewall aux ports 2020 et 2121 pour l'utilisation du FTP. Nous devons aussi préciser notre adresse IP pour l'emploi du mode passif.

Dans la section *Gestion des utilisateurs anonymes*, nous interdissons toute connexion non authentifiée, par conséquence l'emploi du mode anonyme est interdit.

Dans la section *Gestion des utilisateurs locaux*, nous autorisons l'accès à tous les utilisateurs ayant un compte sur le serveur. Ces derniers sont alors automatiquement dirigés vers le répertoire `/var/ftp`. Il faut donc le créer et lui attribuer les bons droits d'accès. Nous créons pour cela un utilisateur et un groupe *ftp* :

```
# adduser ftp
# mkdir /var/ftp
# chown ftp.ftp /var/ftp
# chmod ug+rwx /var/ftp
# chmod o-rwx /var/ftp
# chmod g+s /var/ftp
```

Nous autorisons l'accés en lecteur, écriture, exécution pour l'utilisateur et le groupe et interdisons tout accès pour les autres utilisateurs. Le SetGid bit est positionné pour que tout nouveau fichier ou répertoire créé soit du groupe *ftp* et en hérite des droits d'accès.

Modifiez le fichier de configuration `/etc/vsftpd.conf` comme suit :

```
# Configuration de vsftpd pour ftp.nuts.fr
# Fichier  /etc/vsftpd.conf
#
#
#
#####################
# Options du Daemon #
#####################
#
# Utilisation en mode "standalone"
listen=YES
# Support de l'IPv6
listen_ipv6=NO
# Port pour le canal de controle
listen_port=2121
# Port pour le canal de donnees
ftp_data_port=2020
# Nombre maximum de clients connectes simultaneement
max_clients=4
# Nombre de connections par clients
max_per_ip=1
# Executer vsftpd en tant qu'utilisateur "ftpsecure"
nopriv_user=ftpsecure
# Options non definies
#background=NO
#check_shell=YES
#connect_from_port_20=NO
#one_process_model=NO
#run_as_launching_user=NO
#tcp_wrappers=NO
#listen_address=
#listen_address6=
#
#
#
##########################
# Options de connections #
##########################
#
# Utilisation du mode PASV
pasv_enable=YES
pasv_max_port=2020
pasv_min_port=2020
pasv_address=217.217.36.25
#pasv_promiscuous=NO
#accept_timeout=60
#
# Utilisation du mode PORT
port_enable=NO
ftp_data_port=2020
#connect_from_port_20=NO
#port_promiscuous=NO
#connect_timeout=60
```

```
#
# Autres options
#data_connect_timeout=300
#
#
#
############################
# Cryptage des connections #
############################
#
# This option specifies the location of the RSA certificate to use for SSL
# encrypted connections.
ssl_enable=YES
ssl_tlsv1=YES
ssl_sslv2=NO
ssl_sslv3=NO
ssl_ciphers=DES-CBC3-SHA
dsa_cert_file=/etc/ssl/certs/vsftpd.pem
rsa_cert_file=/etc/ssl/certs/vsftpd.pem
force_local_data_ssl=YES
force_local_logins_ssl=YES
#
#
#
####################
# Gestion des logs #
####################
#
# Utilisation de 2 fichiers de logs ?
dual_log_enable=YES
# Fichier de logs de vsftpd
vsftpd_log_file=/var/log/vsftpd.log
# Utiliser syslog plutot que le fichier de logs de vsftpd ?
syslog_enable=NO
# Utilisation de xferlog
xferlog_enable=YES
xferlog_file=/var/log/xferlog
xferlog_std_format=YES
# Logguer toutes les requetes et reponses ?
log_ftp_protocol=NO
# Options non definies
#no_log_lock
#
#
#
#####################################
# Gestion des utilisateurs anonymes #
#####################################
#
# Autoriser le FTP anonyme ?
anonymous_enable=NO
# Autoriser les connections anonymes non securisees ? (ssl_enable doit etre actif)
allow_anon_ssl=NO
# Ne pas demander de mots de passe pour les utilisateurs anonymes ?
no_anon_password=YES
```

```
# Utilisation d'une liste d'adresses e-mail interdites ?
deny_email_enable=NO
#banned_email_file=/etc/vsftpd.banned_emails
# Utilisation d'une liste d'adresses e-mail de confiance ?
secure_email_list_enable=NO
#email_password_file=/etc/vsftpd.email_passwords
# Se connecter dans le repertoire suivant
anon_root=/var/ftp
# en tant qu'utilisateur "ftp"
ftp_username=ftp
# Autoriser la creation de nouveaux repertoires ? (write_enable doit etre actif)
anon_mkdir_write_enable=NO
# Autoriser la suppression et le renomage des fichiers ?
anon_other_write_enable=NO
# Autoriser l'upload ? (write_enable doit etre actif) - Faut-il changer l'utilisateur du fichier ? Si oui,
anon_upload_enable=NO
chown_uploads=YES
chown_username=ftp
# Seuls les fichiers en lecture pour tout le monde peuvent etre telecharges
anon_world_readable_only=YES
#
#
#
###################################
# Gestion des utilisateurs locaux #
###################################
#
# Autoriser les utilisateurs locaux ?
local_enable=YES
# Repertoire des utilisateurs locaux
local_root=/var/ftp
# Mask des utilisateurs locaux
local_umask=007
# Taux de transfert maximum (0=illimite)
local_max_rate=0
# Chrooter les utilisateurs locaux ?
chroot_local_user=YES
chroot_list_enable=NO
#chroot_list_file=/etc/vsftpd.chroot_list
#passwd_chroot_enable=NO
#
#
#
###############
# Options FTP #
###############
# Baniere de login
ftpd_banner=Welcome to the ftp.nuts.fr FTP service.
#banner_file=
# Activer les messages de repertoires
dirmessage_enable=YES
message_file=.message
# Autoriser de lister les repertoires
dirlist_enable=YES
# Autoriser la recursivite pour le listage des repertoires
```

```
ls_recurse_enable=YES
# Afficher les identifiants des utilisateurs et des groupes en textuel
text_userdb_names=YES
# Autoriser le telechargement
download_enable=YES
# Autoriser l'ecriture
write_enable=YES
# Changer les droits des fichiers en upload
file_open_mode=0666
# Autoriser les transferts en mode ASCII
ascii_upload_enable=YES
ascii_download_enable=YES
# Options non definies
#async_abor_enable=YES
#force_dot_files=NO
#hide_ids=NO
#tilde_user_enable=NO
#use_localtime=NO
#cmds_allowed=
#deny_file=
#hide_file=
#
#
#
# Debian customization
#
# Some of vsftpd's settings don't fit the Debian filesystem layout by
# default.  These settings are more Debian-friendly.
#
# This option should be the name of a directory which is empty.  Also, the
# directory should not be writable by the ftp user. This directory is used
# as a secure chroot() jail at times vsftpd does not require filesystem
# access.
secure_chroot_dir=/var/run/vsftpd
#
# This string is the name of the PAM service vsftpd will use.
pam_service_name=vsftpd
#
```

Pour que le système de logs fonctionne bien, il faut créer les fichiers (initialement vides) de logs :

```
# touch /var/log/vsftpd.log
# touch /var/log/xferlog
```

5.2.3 Adresse IP dynamique

Comme nous utilisons une adresse IP dynamique, nous avons besoin de modifier la ligne :

```
pasv_address=217.217.36.25
```

de la section *Option de connexion* de notre fichier de configuration.

Pour cela, nous avons besoin d'un script de mise à jour pour notre fichier de configuration.

Voici ce petit script : `vsftpd_ip`

```
#!/bin/sh
#vsftpd.conf IP update

vsftpd_conf=/etc/vsftpd.conf
vsftpd_log=/var/log/vsftpd.log

#change to your domain name in next line
my_ip=‘host mon_site.dyndns.org | cut -f4 -d" "‘
vsftpd_ip=‘grep pasv_address $vsftpd_conf | cut -f2 -d=‘

    if [ "$my_ip" != "$vsftpd_ip" ] ; then
       sed -e "s/$vsftpd_ip/$my_ip/g" $vsftpd_conf
       ( echo ",s/$vsftpd_ip/$my_ip/g" && echo w ) | ed - $vsftpd_conf
       echo ‘date‘ "$vsftpd_conf updated with $my_ip IP address" >> $vsftpd_log
       /etc/init.d/vsftpd reload
    fi
```

Pour automatiser la mise à jour de l'adresse IP au démarrage de *vsFTPd*, nous rajoutons notre script dans les sections "start" et "restart" du fichier `/etc/init.d/vsftpd`

```
#!/bin/sh
# /etc/init.d/vsftpd
#
# Written by Sander Smeenk <ssmeenk@debian.org>

set -e

# Exit if vsftpd.conf doesn't have listen=yes or listen_ipv6=yes
# (mandatory for standalone operation)
if [ -f /etc/vsftpd.conf ] && ! egrep -iq "^ *listen(_ipv6)? *= *yes" /etc/vsftpd.conf; then
    exit 0
fi

DAEMON=/usr/sbin/vsftpd
NAME=vsftpd

test -x $DAEMON || exit 0

case "$1" in
  start)
    echo -n "Starting FTP server: $NAME"
    /root/vsftpd_ip
    start-stop-daemon --start --background -m --pidfile /var/run/vsftpd/vsftpd.pid --exec $DAEMON
    echo "."
    ;;
  stop)
    echo -n "Stopping FTP server: $NAME"
    start-stop-daemon --stop --pidfile /var/run/vsftpd/vsftpd.pid --oknodo --exec $DAEMON
    echo "."
      ;;
  restart)
    echo -n "Restarting FTP server: $NAME"
    start-stop-daemon --stop --pidfile /var/run/vsftpd/vsftpd.pid --oknodo --exec $DAEMON
    /root/vsftpd_ip
    start-stop-daemon --start --background -m --pidfile /var/run/vsftpd/vsftpd.pid --exec $DAEMON
```

```
    echo "."
    ;;
  reload|force-reload)
    echo "Reloading $NAME configuration files"
    start-stop-daemon --stop --pidfile /var/run/vsftpd/vsftpd.pid --signal 1 --exec $DAEMON
    echo "."
    ;;
  *)
    echo "Usage: /etc/init.d/$NAME {start|stop|restart|reload}"
    exit 1
    ;;
esac

exit 0
```

5.2.4 Génération du certificat de sécurité

Création des clefs de cryptage RSA :

```
# openssl genrsa -out server.key 1024
```

Création du certificat autosigné :

```
# openssl req -new -x509 -days 365 -key server.key -out server.crt
```

Création du fichier `vsftpd.pem` :

```
# cat server.key server.crt > /etc/ssl/certs/vsftpd.pem
```

5.2.5 Gestion des utilisateurs

Comme nous n'autorisons l'accès qu'aux *utilisateurs locaux*, seuls les utilisateurs ayant un compte sur le serveur peuvent y accéder.

Pour ajouter un utilisateur, il suffit de lui créer un compte et de l'ajouter au groupe *ftp*. Ainsi pour ajouter l'utilisateur *bob*, faire en tant que root :

```
# adduser bob
# adduser bob ftp
```

6 Serveur WEB

6.1 Introduction

6.1.1 Nos besoins

Pour tenir informés nos utilisateurs des modifications et des nouveautés apportées au serveur, nous allons créer une interface web accessible depuis un simple navigateur internet.

Pour cela, nous utilisons un éditeur de blogs très connu, *DotClear*. Ce dernier nécessite pour fonctionner l'utilisation d'un serveur web (nous choisissons *Apache*), d'un interpréteur de scripts PHP et d'une base de données *MySQL*.

Nous allons donc installer ces différents programmes sur notre serveur.

6.1.2 Un serveur LAMP

Nous avons finalement besoin d'un serveur *LAMP* = Linux + Apache + MySQL + Php.

C'est un terme consacré dans le monde des serveurs web !

6.1.3 Sécurité

Comme pour le serveur FTP défini au chapitre précédent, nous allons utiliser une transmission sécurisée par canal crypté en utilisant la technologie SSL. Cette dernière rend très difficile l'interception sur internet des communications entre notre serveur et les ordinateurs qui s'y connectent.

Cependant, la technologie SSL telle que nous l'utilisons (sans échange de clefs de confiance) ne permet pas de restreindre l'accès au serveur qu'à ces membres. Pour cela, nous utiliserons un système d'authentification par login et mot-de-passe.

Ces deux aspects de la sécurité de notre serveur seront mis en place au niveau du serveur web *Apache*.

6.2 Configuration d'Apache

6.2.1 Installation des paquets nécessaires

Nous avons besoin des paquets DEBIAN suivants : *apache2* et *apache2-common*. Pour cela, en tant que superutilisateur (utilisateur *root*), entrez les commandes suivantes dans une console :

```
# aptitude install apache2 apache2-common
```

et suivez les instructions.

Pour activer l'utilisation de l'encodage des caractères UTF8, il faut rajouter la ligne suivante dans le fichier de configuration du serveur Apache, `/etc/apache2/apache2.conf` :

```
addDefaultCharset UTF-8
```

Normalement, c'est la seule modification que nous apporterons au fichier `/etc/apache2/apache2.conf`.

Pour que le serveur Apache prenne en compte cette modification, il faut le "forcer" à relire ses fichiers de configuration :

```
# /etc/init.d/apache2 force-reload
```

6.2.2 Installation des modules complémentaires

Nous allons avoir besoin du support SSL pour crypter la connexion et du support PAM pour l'authentification des utilisateurs.

Le support SSL est maintenant intégré par défaut dans le paquet *apache2*. Nous devons juste rajouter le module *mod_auth_pam* comme suit :

```
# aptitude install libapache2-mod-auth-pam
```

et créer le lien suivant:

```
# ln -s /etc/pam.d/apache2 /etc/pam.d/httpd
```

Il nous faut aussi permettre à l'utilisateur d'Apache qui est en fait *www-data* de lire le fichier `/etc/shadow` en rajoutant l'utilisateur *www-data* au groupe *shadow* :

```
# adduser www-data shadow
```

Ensuite nous devons activer ces deux modules :

```
# a2enmod ssl
# a2enmod auth_pam
```

et "forcer" le serveur Apache à relire ses fichiers de configuration :

```
# /etc/init.d/apache2 force-reload
```

6.2.3 Création du certificat de sécurité

Apache2 possède sa propre commande pour créer un certificat auto-signé : *apache2-ssl-certificate*.

Si le script *apache2-ssl-certificate* n'est pas présent, voir plus loin pour une autre possibilité de créer un certificat de sécurité.

Si le script *apache2-ssl-certificate* est présent, faire :

```
# apache2-ssl-certificate

creating selfsigned certificate
replace it with one signed by a certification authority (CA)

enter your ServerName at the Common Name prompt

If you want your certificate to expire after x days call this programm
with -days x
Generating a 1024 bit RSA private key
................................++++++
..................++++++
writing new private key to '/etc/apache2/ssl/apache.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [GB]:FR
State or Province Name (full name) [Some-State]:France
Locality Name (eg, city) []:Plaisir
Organization Name (eg, company; recommended) []:www.nuts.fr
Organizational Unit Name (eg, section) []:Serveur web
Email Address []:webmaster@nuts.fr
```

Ce dernier génère en fonction des informations que vous stipulez un fichier de clefs : `/etc/apache2/ssl/apache.pem`.

Si le script *apache2-ssl-certificate* n'est pas présent, il y a une autre possibilité de créer un certificat de sécurité en utilisant le script *make-ssl-cert* du paquet DEBIAN *ssl-cert* :

```
# aptitude install ssl-cert
# make-ssl-cert /usr/share/ssl-cert/ssleay.cnf /etc/apache2/ssl/apache.pem
```

et suivez les instructions comme précédemment.

6.2.4 Configuration du site www.nuts.fr

Le fichier de configuration Editons un nouveau fichier `/etc/apache2/sites-available/nuts` comme suit :

```
# Le port 443 est le port pour l'accès sécurisé par HTTPS.
NameVirtualHost *:443
<VirtualHost *:443>
        ServerAdmin webmaster@localhost

        DocumentRoot /var/www/
        <Directory />
                Options FollowSymLinks
                AllowOverride None
        <\/Directory>
        <Directory /var/www/>
                Options Indexes FollowSymLinks MultiViews
                AllowOverride None
                Order allow,deny
                allow from all

                # This directive allows us to have apache2's default start page
                # in /apache2-default/, but still have / go to the right place
                RedirectMatch ^/$ /apache2-default/

                # PAM authentification module
                AuthType Basic
                AuthName "PAM"
                AuthPAM_Enabled On
                AuthPAM_FallThrough off
                AuthBasicAuthoritative off
                Require valid-user
                Require group ftp
        <\/Directory>

        ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
        <Directory "/usr/lib/cgi-bin">
                AllowOverride None
                Options ExecCGI -MultiViews +SymLinksIfOwnerMatch
                Order allow,deny
                Allow from all
        <\/Directory>

        ErrorLog /var/log/apache2/error.log

        # Possible values include: debug, info, notice, warn, error, crit,
        # alert, emerg.
        LogLevel warn
```

```
        CustomLog /var/log/apache2/access.log combined
        ServerSignature Off

        # Secure socket layer
        SSLEngine On
        SSLCertificateFile /etc/apache2/ssl/apache.pem

    Alias /ftp/ "/var/ftp/"
    <Directory "/var/ftp/">
        Options Indexes MultiViews FollowSymLinks
        AllowOverride None
        Order allow,deny
        allow from all
        # PAM authentification module
        AuthType Basic
        AuthName "PAM"
        AuthPAM_Enabled On
        AuthPAM_FallThrough off
        AuthBasicAuthoritative off
        Require valid-user
        Require group ftp
    <\/Directory>

<\/VirtualHost>
```

La ligne `ServerSignature Off` permet de masquer les informations relatives à la version du serveur, notamment sur la page d'erreur 404.

Sécurisation de la connexion Remarquons dans le fichier de configuration les lignes qui définissent l'utilisation d'un canal sécurisé par SSL et certificat de sécurité :

```
        # Secure socket layer
        SSLEngine On
        SSLCertificateFile /etc/apache2/ssl/apache.pem
```

On retrouve ici l'emplacement de notre fichier de clefs généré par *apache2-ssl-certificate* ou *make-ssl-cert*.

Il faut aussi modifier le fichier */etc/apache2/ports.conf* comme suit :

```
#Listen 80
Listen 443
```

pour interdire l'utilisation du canal non crypté et autoriser l'utilisation du *https*.

Authentification des utilisateurs Remarquons dans le fichier de configuration les lignes suivantes qui définissent l'authentification des utilisateurs par le système PAM - Pluggable Authentication Module :

```
        # PAM authentification module
        AuthType Basic
        AuthName "PAM"
        AuthPAM_Enabled On
        AuthPAM_FallThrough off
        AuthBasicAuthoritative off
        Require valid-user
        Require group ftp
```

Pour les versions d'Apache supérieures à 2.1, les lignes *AuthPAM_ FallThrough off* et *AuthBasicAuthoritative off* sont obligatoires pour faire fonctionner le module PAM et le module *authz_ user* doit être activé.

Accès au répertoire FTP Remarquons aussi le paragraphe concernant */var/ftp* qui nous permet d'accéder depuis l'interface web au répertoire FTP. Il faut pour cela, que l'utilisateur *www-data* ait accès en lecture au répertoire `/var/ftp/` en rajoutant l'utilisateur *www-data* au groupe *ftp* :

```
# adduser www-data ftp
```

6.2.5 Activer le site

Pour activer ce site sur le serveur Apache, nous devons entrer les commandes suivantes :

```
# a2ensite nuts
# /etc/init.d/apache2 force-reload
```

Pour désactiver le site par défaut, faire :

```
# a2dissite default
# /etc/init.d/apache2 force-reload
```

6.2.6 Test du bon fonctionnement du serveur Apache

Depuis un autre ordinateur, entrez l'adresse suivante dans votre navigateur favori : `https://www.nuts.fr/`. Vous devriez voir apparaître la page d'acceuil du serveur Apache.

Si ce n'est pas la cas, c'est qu'il y a un problème. Pour connaître l'origne de ce problème consultez les logs du serveur dans les deux fichiers : `/var/log/apache2/error.log` et `/var/log/apache2/access.log`.

6.3 Configuration du support PHP

6.3.1 Installation de l'interpréteur PHP

Nous avons besoin des paquets DEBIAN suivants : *php5* et *php5-mysql*. Pour cela, en tant que superutilisateur (utilisateur *root*), entrez les commandes suivantes dans une console :

```
# aptitude install php5 php5-mysql
```

et suivez les instructions.

6.3.2 Installation du module complémentaire pour Apache

Nous avons besoin du paquet DEBIAN suivant : *libapache2-mod-php5*. Pour cela, en tant que superutilisateur (utilisateur *root*), entrez les commandes suivantes dans une console :

```
# aptitude install libapache2-mod-php5
```

et suivez les instructions.

Ensuite nous devons activer ce module dans Apache :

```
# a2enmod php5
```

et "forcer" le serveur Apache à relire ses fichiers de configuration :

```
# /etc/init.d/apache2 force-reload
```

6.3.3 Configuration de PHP

Là, il y a normalement rien à faire, l'installation par défaut des paquets DEBIAN fait tout ce dont nous avons besoin à notre place.

6.3.4 Vérification de l'installation

Editions un nouveau fichier `/var/www/info.php` comme suit :

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

et accédons-y depuis un autre poste par notre navigateur web préféré à l'adresse `https://www.nuts.fr/info.php`.

Un tableau montrant les divers paramètres de la configuration de PHP devrait apparaître...

6.4 Configuration du support MySQL

6.4.1 Installation de MySQL

Nous avons besoin des paquets DEBIAN suivants : *mysql-server* et *mysql-client*. Pour cela, en tant que superutilisateur (utilisateur *root*), entrez les commandes suivantes dans une console :

```
# aptitude install mysql-server mysql-client
```

et suivez les instructions.

6.4.2 Configuration de MySQL

Ici encore, il y a normalement rien à faire, l'installation par défaut des paquets DEBIAN fait tout ce dont nous avons besoin à notre place.

6.4.3 Sécurisation de la base de données

Pour tester le fonctionnement de MySQL, lancez la commande :

```
# mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8 to server version: 4.0.24_Debian-10sarge2-log

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

Tapez `quit` pour sortir du gestionnaire.

Vous constaterez que l'accès au gestionnaire de base de donnée en tant qu'administrateur n'est pas sécurisé, puisque vous n'avez pas eu à rentrer de mot-de-passe. Nous allons y remédier tout de suite. Entrez la commande suivante :

```
# mysqladmin password 'mot-de-passe en clair'
```

Maintenant l'accès au gestionnaire de la base de données est contrôlé par un mot-de-passe. Pour y accéder vous devrez désormais utiliser l'option *-p* comme ceci :

```
# mysql -u root -p
Enter password: ********
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8 to server version: 4.0.24_Debian-10sarge2-log

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

6.4.4 Interface d'administration phpMyAdmin

Pour faciliter l'administration de la base de données, nous allons utiliser un outil dédié qui a le bon gout d'avoir une interface graphique et d'être accessible depuis n'importe quel poste. Vous l'aurez compris, il s'agit de *phpMyAdmin*.

Nous avons besoin du paquet DEBIAN suivant : *phpmyadmin*. Pour cela, en tant que superutilisateur (utilisateur *root*), entrez les commandes suivantes dans une console :

```
# aptitude install phpmyadmin
```

et suivez les instructions.

Pour accéder à l'interface d'utilisation de *phpMyAdmin*, entrez l'URL suivante dans votre navigateur : `https://www.nuts.fr/phpmyadmin/` depuis n'importe quel poste ayant un accès au serveur `www.nuts.fr`.

6.4.5 Création de la base de données pour DotClear

Depuis l'interface d'administration *phpMyAdmin*, il ne nous reste plus qu'à créer une base de données et un utilisateur pour *DotClear*.

Pour créer un nouvelle base de donnée, rentrez son nom (par exemple : "dotclear") dans le champs "Créer une base de données" et cliquez sur le bouton "Créer". Voilà, c'est aussi simple que ça !

Pour ajouter un nouvel utilisateur, cliquez sur le lien "Privilèges" de la page principale. Ensuite, cliquez sur le lien : "Ajouter un utilisateur".

Donnez lui un nom et un mot-de-passe, spécifiez le serveur de connexion (ici *localhost*) et accordez lui tous les privilèges des colonnes "Données" et "Structure" et les privilèges "Super", "Process" et "Lock tables" de la colonne "Administration". Puis appuyez sur le bouton "Exécuter".

Votre base de données est maintenant prête pour une utilisation avec *DotClear*.

6.5 Configuration de DotClear

6.5.1 Installation de Dotclear

Pour installer *Dotclear*, il faut récupérer l'archive sur le site <http://www.dotclear.net/> et la décompresser dans le répertoire `/var/www/` :

```
# cd /var/www/
# wget http://www.dotclear.net/download/dotclear-1.2.5.tar.gz
# tar -zxvf dotclear-1.2.5.tar.gz
```

6.5.2 Configuration de DotClear

Dans un navigateur internet depuis un autre poste, entrez l'URL suivante : `https://www.nuts.fr/dotclear/install`.

Entrez les informations demandées et spécifiez les données pour la base MySQL, telles que vous les avez définies dans la section *Création de la base de données pour DotClear*.

Cliquez sur le bouton "Terminer l'installation". Voilà, c'est fini...

6.5.3 Voir le blog

Pour voir le blog, toujours depuis un autre poste, entrez l'URL suivante dans votre navigateur internet : `https://www.nuts.fr/dotclear/`.

6.5.4 Ecrire un nouveau billet

Pour écrire un nouveau billet, entrez l'URL suivante : `https://www.nuts.fr/dotclear/ecrire`.

L'interface est très claire et est en français, alors laissez-vous guidé !

6.5.5 Accéder directement à DotClear

Maintenant, paramétrons *Apache* pour qu'il nous redirige directement sur la page de notre blog depuis l'adresse `https://www.nuts.fr/`.

Pour cela il suffit de remplacer dans le fichier de configuration de notre site `/etc/apache2/sites-available/nuts` la ligne suivante :

```
RedirectMatch ^/$ /apache2-default/
```

par celle-ci :

```
RedirectMatch ^/$ /dotclear/
```

et de recharger la configuration du serveur *Apache* :

```
# /etc/init.d/apache2 force-reload
```

Et le tour est joué !

Voilà, notre serveur WEB est maintenant opérationel !

7 Serveur Peer-to-Peer

7.1 Introduction

Comme nous avons un serveur à notre disposition et que ce dernier fonctionnera 24h/24 et 7jours/7, autant s'en servir pour les téléchargements de type peer-to-peer qui prennent plusieurs heures voir plusieurs jours.

Nous allons donc installer un serveur *mldonkey* pour gérer les protocoles *edonkey, fasttrack, open-napster, bittorrents* et autres...

7.2 Configuration de MLDonkey

7.2.1 Installation de MLDonkey

Nous avons besoin du paquet DEBIAN suivant : *mldonkey-server*. Pour cela, en tant que superutilisateur (utilisateur *root*), entrez les commandes suivantes dans une console :

```
# aptitude install mldonkey-server
```

et suivez les instructions.

7.2.2 Post-installation de MLDonkey

Le programme est maintenant installé sur le serveur, mais il faut encore le configurer et préparer son premier lancement.

Nous commençons par créer un utilisateur *donkey* pour ne pas donner les droits d'accès *root* (du super-utilisateur) au programme *mldonkey*, qui est en fait *mlnet*. Nous créons donc notre utilisateur et notre groupe *donkey*, puis nous attribuons à son répertoire d'utilisateur les bons droits d'accès :

```
# adduser donkey
# chmod ug+rwx /home/donkey
# chmod o-rwx /home/donkey
# chmod g+s /home/donkey
```

Nous autorisons l'accés en lecteur, écriture, exécution pour l'utilisateur et le groupe et interdisons tout accès pour les autres utilisateurs. Le SetGid bit est positionné pour que tout nouveau fichier ou répertoire créé soit du groupe *donkey* et en hérite des droits d'accès.

Ensuite, nous devons préparer le répertoire de travail de *mldonkey*. Pour cela, nous passons sous l'utilisateur *donkey* et nous lançons "à la main" *mlnet* :

```
# su donkey
# cd /home/donkey
# mlnet
```

Ce dernier crée un répertoire `.mldonkey` dans `/home/donkey` et génère le fichier `downloads.ini`, qui est le fichier de configuration de *mldonkey* et qui est nécessaire à son fonctionnement.

Maintenant que cela est fait, il faut renseigner le fichier `/etc/default/mldonkey-server` pour donner au *daemon mldonkey* les informations dont il a besoin :

```
MLDONKEY_DIR=/home/donkey/.mldonkey/
MLDONKEY_USER=donkey
```

```
MLDONKEY_GROUP=donkey
MLDONKEY_UMASK=
MAX_ALIVE=
LAUNCH_AT_STARTUP=false
MLDONKEY_NICENESS=
```

Les informations laissées vides seront positionnées à leur valeur par défaut.

Comme la directive `LAUNCH_AT_STARTUP` est positionnée à `false`, *mldonkey* ne sera pas automatiquement lancé au démarrage du serveur. Il faudra donc utiliser la commande `/etc/init.d/mldonkey-server force-start` pour le lancer.

7.2.3 Configuration de MLDonkey

mldonkey est prêt à être lancé, mais il ne donnera pour le moment aucun résultat. Il faut encore le configurer pour pouvoir s'y connecter pour l'administrer à distance et pour permettre aux autres serveurs de le joindre.

Tout ce fait dans le fichier `~donkey/.mldonkey/downloads.ini`.

Nous supposons que l'adresse du serveur en local est `192.168.0.1`, que nous y accédons depuis le poste à l'adresse `192.168.0.2` et que son adresse sur internet est `217.217.36.25`.

Nous en profitons pour corriger une erreur au niveau du fichier qui sert à obtenir la liste des serveurs *edonkey*. Ce dernier que l'on récupère à l'adresse `http://www.gruk.org/server.met.gz` n'est en fait pas compressé et génère une erreur lors de sa lecture. Il suffit de récupérer le fichier non compressé `http://www.gruk.org/server.met` pour corriger le problème.

Éditions le fichier `~donkey/.mldonkey/downloads.ini` :

```
[...]

allowed_ips = ["127.0.0.1"; "192.168.0.2";]

[...]

client_ip = "217.217.36.25"

force_client_ip = true

user_agent = default

web_infos = [
  ("server.met", 0, "http://www.gruk.org/server.met");
  ("guarding.p2p", 96, "http://www.bluetack.co.uk/config/level1.gz");
  ("contact.dat", 168, "http://download.overnet.org/contact.dat");
  ("geoip.dat", 0, "http://www.maxmind.com/download/geoip/database/GeoIP.dat.gz");
  ("nodes.gzip", 0, "http://update.kceasy.com/update/fasttrack/nodes.gzip");]
```

et relançons le *daemon* :

```
# /etc/init.d/mldonkey-server stop
# /etc/init.d/mldonkey-server force-start
```

Il faut maintenant changer le mot-de-passe de l'administrateur de *mldonkey* :

```
# telnet localhost 4000
```

```
[...]
useradd admin password
[...]
q
```

7.3 Partageons nos fichiers

Pour partager nos fichiers, il faut les copiers dans le répertoire `~donkey/.mldonkey/shared/`.

7.4 Accéder à la console d'administration depuis un autre poste

Pour contrôler le serveur depuis un autre poste, il faut se connecter avec *telnet* :

```
# telnet 192.168.0.1 4000
[...]
auth admin password
[...]
```

et utiliser les options "?" et "??" pour connaître les commandes disponibles.

7.5 Gérer ses téléchargements depuis un autre poste

7.5.1 Principe

Pour gérer ses téléchargements depuis un autre poste, on peut utiliser un client graphique. En effet, *mldonkey* n'est que le coeur du système de téléchargement et le client graphique peut se connecter au *daemon* et lui envoyer des commandes.

7.5.2 Avec l'interface web

mldonkey fourni nativement une interface web pour piloter le daemon. On y accède en entrant l'adresse `http://192.168.0.1:4080/` dans votre navigateur internet.

7.5.3 Avec un autre logiciel

Personnellement, je préfère utiliser l'application *KMLDonkey* de mon environnement *KDE*. Il suffit de renseigner les champs suivant dans la boîte de dialogue "Configurer la connexion" :

```
Nom : NUTS
Adress : 192.168.0.1
GUI port : 4001
HTTP port : 4080
User name : admin
Password : *********
Core type : External core
```

et de se connecter au core *NUTS*.

8 IceCast

8.1 Avenir

Ah Ah !

```
A venir...
```