Serveur Linux : FTP

Mise en place d'un service FTP sous Linux

Bouron Dimitri

20/04/2014

Ce document sert de démonstration concise pour l'installation, la configuration, la sécurisation, d'un serveur FTP sous Linux utilisant proftpd.

Table des matières

I. Installation et configuration de base

Une fois l'installation de Ubuntu 12.04 faite, il faut se connecter avec l'identifiant de connexion crée lors de l'installation de l'OS. Dans notre exemple nous avons créé l'utilisateur « mangetsu:P@ssword ». Une fois que nous avons réussi à nous connecter nous allons devoir activer l'utilisateur root pour les configurations à venir en utilisant la commande **$ sudo passwd root**. Le mot de passe que nous donnerons à root sera : P@ssword (faites ce que vous voulez). Puis on se connectera avec les identifiants de ce nouvel utilisateur avec la commande **$ su root**.

À partir de maintenant, nous considérons que toutes les commandes qui suivront sont faites à partir du compte root, même si l'utilisateur simple peut faire la modification pour des raisons de facilités.

A. Apt-get update

On met rapidement la base à jour avec la commande **# apt-get update**.

B. Hostname

L'hostname correspond au nom d'hôte de la machine, que l'on renseigne une première fois lors de l'installation de l'OS. Mais si l'on souhaite créer une nouvelle machine virtuelle à partir de celle fournie alors il nous faut modifier le nom d'hôte pour l'adapter à notre situation. La commande qui permet de visualiser le nom d'hôte est **# hostname**. Pour modifier le nom d'hôte de la machine, on utilise la commande **# nano /etc/hostname**, puis nous modifierons la première ligne en la remplaçant par le nom d'hôte souhaité (attention, il faut éviter de mettre un ou plusieurs underscore, il se peut que cela ne fonctionne pas correctement). Une fois la ligne modifiée et le fichier hostname sauvegardé, nous allons devoir faire relire le fichier hostname par le serveur, le plus bourrin serait d'utiliser **# reboot** pour redémarrer le serveur, mais une solution est de forcer la relecture du fichier hostname en utilisant la commande suivante : **# /etc/init.d/hostname restart**. Pour vérifier si la modification a bien été lu après le redémarrage du réutiliser la commande **# hostname**.

Attention, on peut remarquer que la modification est bien prise en compte actuellement en utilisant la commande pour afficher l'hostname mais la valeur affichée dans « **root@serveur-FTP-base:~#** » est toujours la même, elle n'a pas été modifié malgré le changement prit en compte. Toutefois, il suffit de taper la commande **# exit** pour revenir à l'écran d'identification et nous pouvons constater que la modification apparaît bien désormais.

C. Interfaces

Nous utilisons un serveur, FTP pour être exacte, ce qui implique de lui attribuer une adresse IP fixe, nous allons donc attribuer l'adresse fixe directement sur notre serveur FTP. Pour cela nous allons utiliser la commande **# nano /etc/network/interfaces** et nous modifierons le fichier en remplaçant la ligne « iface eth0 inet dhcp » comme suit :

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
        address 192.168.1.150
        netmask 255.255.255.0
        network 192.168.1.0
        broadcast 192.168.1.255
        gateway 192.168.1.1
        dns-nameservers 192.168.1.1
```

Les adresses IP sont modifier selon la situation naturellement.

Renseigner toutes les valeurs peut rapidement devenir fastidieux. Du coup on peut utiliser cette méthode tout en conservant l'utilisation du DHCP du réseau (encore faut-il en avoir un). Afin d'éviter de faire des erreurs il est préférable d'utiliser le DHCP pour renseigner les autres valeurs On renseigne donc les valeurs souhaitées et on fait appel au DHCP pour renseigner le reste automatiquement et donc s'adapter au mieux à la situation. Voici la configuration à faire :

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
        address 192.168.1.150
        netmask 255.255.255.0

iface eth0 inet dhcp
```

Se contenter de ça ne nous suffira pas, il nous faut faire comme pour l'hostname forcer la reconfiguration avec la commande **# /etc/init.d/networking restart** et utiliser la commande **# ifconfig** pour vérifier que le changement de valeur est pris en compte puis on peut toujours vérifier l'accès Internet avec un ping. Voir ci-dessous :

```
root@serveur-FTP-base:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:95:c7:70
          inet adr:192.168.1.150  Bcast:192.168.1.255  Masque:255.255.255.0
          adr inet6: fe80::a00:27ff:fe95:c770/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Packets reçus:1246 erreurs:0 :0 overruns:0 frame:0
          TX packets:35 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          Octets reçus:84962 (84.9 KB) Octets transmis:3830 (3.8 KB)

lo        Link encap:Boucle locale
          inet adr:127.0.0.1  Masque:255.0.0.0
          adr inet6: ::1/128 Scope:Hôte
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          Packets reçus:56 erreurs:0 :0 overruns:0 frame:0
          TX packets:56 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:0
          Octets reçus:4304 (4.3 KB) Octets transmis:4304 (4.3 KB)
```

D. Mise en place de openssh-server

1) Installation du paquet

Maintenant que notre serveur de base est prêt. Nous allons mettre en place les différents paquets dont nous avons besoin pour mettre en place le service FTP de base et avant ça encore activer l'accès ssh. Installons notre paquet openssh-server avec la commande **# apt-get install openssh-server**.

```
root@serveur-FTP-base:~# apt-get install openssh-server_
```

On peut désormais vérifier le fonctionnement du ssh avec la commande **# service ssh status** et en essayant de se connecter avec putty au serveur.

```
root@serveur-FTP-base:~# service ssh status
ssh start/running, process 1292
```

Si l'on voit start/running, on peut essayer putty. Sinon il faut essayer de le lancer avec la commande **# /etc/init.d/ssh start**.

2) Configuration de openssh-server

Nous allons maintenant configurer openssh-server correctement, ou plutôt sécuriser l'accès ssh un minimum pour l'instant en modifiant le port d'écoute, ainsi que la restriction de l'utilisation de root pour se connecter en ssh. Pour cela on utilise la commande **# nano /etc/ssh/sshd_config**, puis on va remplacer le numéro de port (22) à la ligne 5 par le port souhaité mais en faisant attention à utiliser un port correcte et non utilisé. Il faut donc un port inférieur à 65535 et éviter les ports utilisés (pour en savoir plus sur les ports utilisés le plus souvent par les autres services et donc les éviter il faut suivre ce lien). Ce n'est pas tout, on va modifier la valeur du PermitRootLogin à la ligne 27 en remplaçant la valeur actuelle (yes) par la valeur no. Voir ci-dessous :

```
# What ports, IPs and protocols we listen for
Port 12543_
```

```
# Authentication:
LoginGraceTime 120
PermitRootLogin no
StrictModes yes
```

Une fois fait et sauvegardé, on utilisera la commande **# /etc/init.d/ssh restart** pour forcer la prise en compte des modifications.

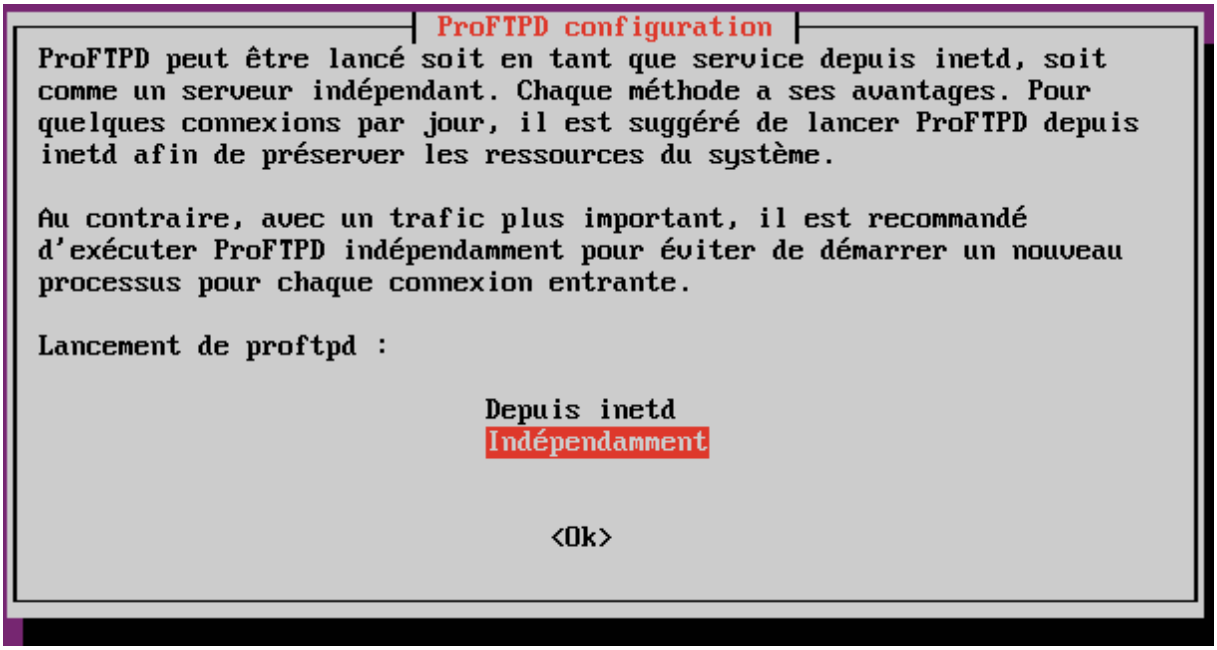
Maintenant pour vérifier que cela fonctionne, il suffit simplement d'utiliser putty et d'essayer de se connecter sur le port 22, normalement aucune réponse possible. Ensuite on se connecte au bon port (12543 pour l'exemple), cette fois on va essayer de se connecter en tant que root, si on rentre les bons identifiants le serveur répondra d'un « access denied ». Donc si on est là tout va bien, on va encore se connecter au serveur mais avec un autre utilisateur (pour l'exemple, on crée un utilisateur administrateur / P@ssword avec lequel on se connectera en ssh et sftp à l'avenir, celui-ci est ajouté au groupe root avec la commande **# adduser administrateur root**).

E. Mise en place de proftpd

1) Installation de proftpd

Nous allons installer le paquet proftpd avec la commande **# apt-get install proftpd**.

On configure proftpd lors de l'installation comme suit :



À ce stade, on vérifie que proftpd est bien démarré avec la commande **# service proftpd status** :

```
root@serveur-FTP-base:~# service proftpd status
ProFTPD is started in standalone mode, currently running.
```

Sinon, on le démarre avec **# service proftpd start**.

2) Configuration de proftpd

Nous allons maintenant configurer protftp, dans un premier temps nous allons modifier le fichier /etc/proftpd/proftpd.conf avec la commande **# nano /etc/proftpd/proftpd.conf**.

On va modifier le nom du serveur (ServerName) qui par défaut est « debian », donnez-lui le nom souhaite (pour l'exemple : serveur-FTP-base).

```
# If set on you can experience a longer connection delay in many cases.
IdentLookups                    off

ServerName                      "serveur-FTP-base"
```

Il faut modifier la directive PassivePorts, car en théorie un serveur FTP devrait pouvoir être accessible en local mais aussi depuis un autre réseau. Il faudrait pour cela configurer la redirection de port mais le FTP peut ouvrir une multitude de ports pour son fonctionnement (le port 21 par défaut, ainsi que 5 autres ports utiles). Pour ne pas avoir à créer 10 000 redirections de ports (beaucoup trop long), on limite la plage de ports utilisable par le FTP à seulement 5 ports (5 ports minimum pour le bon fonctionnement) pour n'avoir que 5 redirections de port en plus du port 21.

On utilisera les ports 10001 à 10005 pour l'exemple.

```
# Port 21 is the standard FTP port.
Port                            21

# In some cases you have to specify passive ports range to by-pass
# firewall limitations. Ephemeral ports can be used for that, but
# feel free to use a more narrow range.
PassivePorts                    10001 10005_
```

Il est possible de modifier le port 21 pour plus de sécurité, nous n'en tiendrons pas compte dans l'exemple mais il serait judicieux de le faire en situation réelle.

Nous permettons donc la redirection de port, mais dans le cas présent cela ne fonctionnera pas. Il faut modifier la directive MasqueradeAddress pour renseigner l'adresse IP externe (publique dans la plupart des cas) afin que le client puisse établir la connexion data channel.

```
# If your host was NATted, this option is useful in order to
# allow passive tranfers to work. You have to use your public
# address and opening the passive ports used on your firewall as well.
# MasqueradeAddress               1.2.3.4
```

Pour l'exemple, on ne modifie pas la valeur prévoyant une utilisation locale uniquement.

3) Utilisateurs FTP

Nous allons mettre en place un système d'authentification à FTP qui se basera sur MySQL pour conserver en mémoire les utilisateurs.

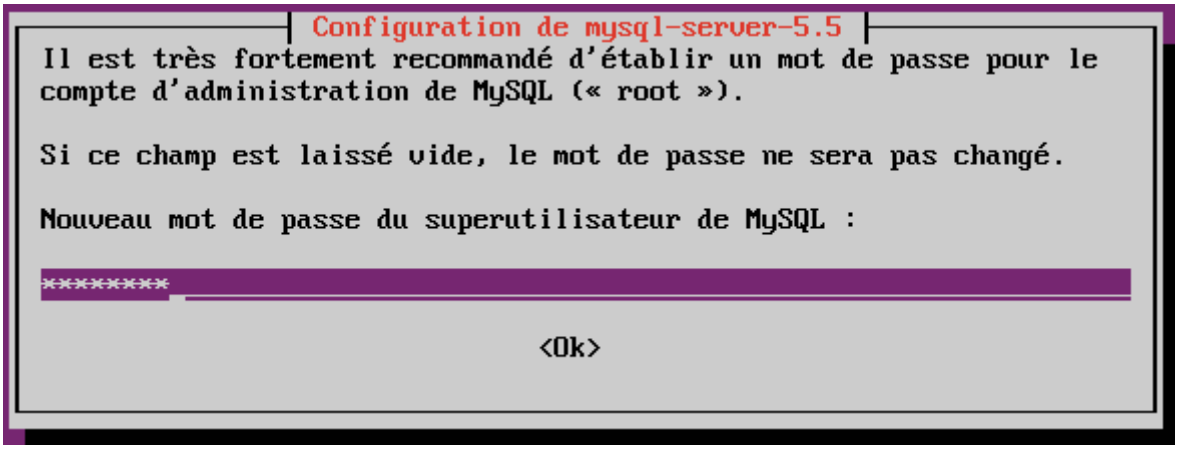
L'intérêt ? Cela permet de ne pas créer des utilisateurs systèmes pour que ceux-ci aient un accès FTP. Dans le cas d'un utilisateur système, il peut se connecter en ssh et sftp. On utilisait donc le chroot avec rssh pour éviter qu'un utilisateur puisse utiliser le ssh comme il le souhaitait. Chrooter est certes une méthode qui fonctionne, elle n'est pas suffisamment sûre en terme de sécurité.

Cette fois, on définit nos utilisateurs dans une base de données SQL qui aura pour effet de n'accorder l'accès qu'en FTP et en aucun cas un accès système. C'est là qu'il y a un intérêt à définir dès le début un nouvel utilisateur système ayant des droits d'administration sans pour autant que l'on ait à utiliser root. On va pouvoir utiliser ensuite notre administrateur pour se déplacer d'environnements en environnements sans restriction contrairement aux utilisateurs définis par le SQL qui auront pour racine (/) leur Home Directory et n'auront accès à rien d'autre.

On doit donc installer apache2 et MySQL, on utilise les commandes :

 # apt-get install apache2 et **# apt-get install mysql-server mysql-client**.

On définit le mot de passe root pour MySQL (P@ssword pour l'exemple).



On va ensuite installer le module qui permet à proftpd de communiquer avec MySQL avec la commande **# apt-get install proftpd-mod-mysql**.

Il faut ensuite activer les modules SQL dans le fichier /etc/proftpd/modules.conf avec la commande :

 # nano /etc/proftpd/modules.conf

```
# Install one of proftpd-mod-mysql, proftpd-mod-pgsql or any other
# SQL backend engine to use this module and the required backend.
# This module must be mandatory loaded before anyone of
# the existent SQL backeds.
LoadModule mod_sql.c
```

```
# Install proftpd-mod-mysql and decomment the previous
# mod_sql.c module to use this.
LoadModule mod_sql_mysql.c
```

On indique à proftpd comment gérer les utilisateurs avec MySQL :

 # nano /etc/proftpd/proftpd.conf

Cherchez la zone suivante :

```
# Set the user and group that the server normally runs at.
User                            proftpd
Group                           nogroup
```

Ensuite modifiez-la comme suit :

```
ServerIdent     Off
IdentLookups    Off

# Set the user and group that the server normally runs at.
User            ftp
Group           www-data

CreateHome      on 711
SQLAuthTypes    Crypt
SQLBackend      mysql
SQLAuthenticate users
SQLConnectInfo  proftpd@localhost userftp passftp
SQLUserInfo     utilisateur nom mot_de_passe uid gid repertoire shell
SQLDefaultUID   107
SQLDefaultGID   33
SQLMinID        100
AllowOverWrite  on
AllowForeignAddress     on
ListOptions     "-a"
DefaultRoot     ~
SQLLogFile      /var/log/proftpd/mysql.log
ExtendedLog     /var/log/proftpd/auth.log AUTH auth
```

Pour les directives SQLDefaultUID et SQLDefaultGID, les valeurs dans l'exemple sont : UID et GID, il faut en réalité saisir des valeurs numériques que l'on va obtenir de la manière suivante pour ensuite remplacer UID par l'UID et GID par le GID :

Pour le GID :

 # cat /etc/group | grep www-data:

Pour l'UID :

 # cat /etc/passwd | grep ftp:

```
root@serveur-FTP-base:~# cat /etc/group | grep www-data:
www-data:x:33:
root@serveur-FTP-base:~# cat /etc/passwd | grep ftp:
ftp:x:107:65534::/srv/ftp:/bin/false
```

On identifie nos deux valeurs qui sont : GID=33 et UID=107 que l'on va pouvoir renseigner dans /etc/proftpd/proftpd.conf :

```
SQLDefaultUID   107
SQLDefaultGID   33
```

On crée la base de données proftpd et la table utilisateur :

Remarque : Faite attention aux quote et anti-quote (‘ et `), qui sont utilisées, anti-quote ( ` : Alt Gr + 7) pour les variables et quote (‘ : 4) pour les valeurs.

 # mysql –u root –p

mysql> CREATE DATABASE proftpd;

mysql> USE proftpd;

```
mysql> CREATE DATABASE proftpd;
Query OK, 1 row affected (0.00 sec)

mysql> USE proftpd;
Database changed
```

mysql> CREATE TABLE `utilisateur` (

`nom` VARCHAR(50) NOT NULL DEFAULT ' ',

`mot_de_passe` VARCHAR(30) NOT NULL DEFAULT ' ',

`uid` INT(11) DEFAULT '101',

`gid` INT(11) DEFAULT '65534',

`repertoire` VARCHAR(255) DEFAULT NULL,

`shell` VARCHAR(255) DEFAULT '/bin/sh',

UNIQUE KEY `nom` (`nom`),

KEY `i_nom` (`nom`),

) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```
mysql> CREATE TABLE `utilisateur` (
    -> `nom` VARCHAR(50) NOT NULL DEFAULT ' ',
    -> `mot_de_passe` VARCHAR(30) NOT NULL DEFAULT ' ',
    -> `uid` INT(11) DEFAULT '101',
    -> `gid` INT(11) DEFAULT '65534',
    -> `repertoire` VARCHAR(255) DEFAULT NULL,
    -> `shell` VARCHAR(255) DEFAULT '/bin/sh',
    -> UNIQUE KEY `nom` (`nom`),
    -> KEY `i_nom` (`nom`)
    -> ) ENGINE=MyISAM DEFAULT CHARSET=latin1;
Query OK, 0 rows affected (0.02 sec)
```

mysql> CREATE USER userftp@localhost IDENTIFIED BY 'passftp';

On vient de créer l'utilisateur, on lui donne le droit de se connecter en local uniquement avec la valeur @localhost, ce qui permet d'empêcher un internaute de se connecter à la base avec cet utilisateur.

```
mysql> CREATE USER userftp@localhost IDENTIFIED BY 'passftp';
Query OK, 0 rows affected (0.00 sec)
```

On lui accorde les privilèges nécessaires sur la base proftpd :

mysql> GRANT SELECT,INSERT,UPDATE,DELETE ON proftpd .* TO userftp@localhost;

```
mysql> GRANT SELECT,INSERT,UPDATE,DELETE ON proftpd .* TO userftp@localhost;
Query OK, 0 rows affected (0.00 sec)
```

Voyons maintenant pour créer un nouvel utilisateur (adaptez les valeurs selon l'utilisateur à créer) :

mysql> INSERT INTO utilisateur VALUES ('nom_user', encrypt('password_user'), 'UID', 'GID', '/var/www/dossier_user', '/bin/sh');

```
mysql> INSERT INTO utilisateur VALUES ('client1', encrypt('P@ssword'), '107', '3
3', '/var/www/client1', '/bin/sh');
Query OK, 1 row affected (0.00 sec)
```

On peut quitter l'interface de commandes SQL avec la commande **mysql> exit**

On va créer manuellement le répertoire de l'utilisateur et modifier les propriétaires du répertoire avec les commandes suivantes :

 # mkdir /var/www/client1

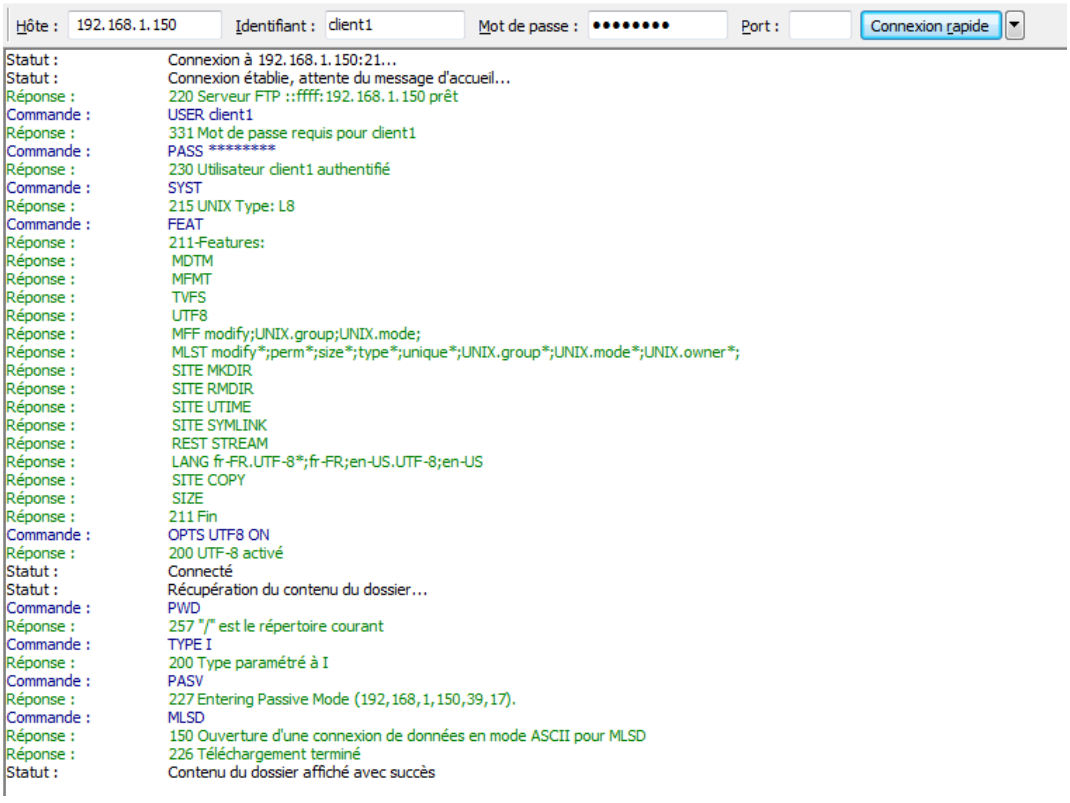
 # chown ftp:www-data /var/www/client1

```
root@serveur-FTP-base:~# mkdir /var/www/client1
root@serveur-FTP-base:~# chown ftp:www-data /var/www/client1
```

On redémarre le service proftpd pour prendre en compte les changements :

 # /etc/init.d/proftpd restart

On peut désormais tester le fonctionnement en utilisant FileZilla et en se connectant avec l'utilisateur créé.



F. Sécurisation par FTPES

Actuellement, le service FTP fait circuler toutes les informations en clair. Donc il est parfaitement possible d'analyser les trames et de voir les identifiants en clair dont le mot de passe !

Nous allons remédier à ça avec FTPES, on va chiffrer les échanges.

Par précaution, on fait une copie de proftpd.conf en cas d'erreur :

 # cp /etc/proftpd/proftpd.conf /etc/proftpd/proftpd.conf.old

On édite le fichier pour dé-commenter la ligne 170 et ajouter la directive TLSOptions NoSessionReuseRequired :

 # nano /etc/proftpd/proftpd.conf

```
#
# This is used for FTPS connections
#
Include /etc/proftpd/tls.conf
TLSOptions NoSessionReuseRequired_
```

On va créer une clé et un certificat ssl pour le chiffrement :

 # mkdir /etc/proftpd/ssl

 # cd /etc/proftpd/ssl

 # openssl genrsa –out proftp.key 1024

 # openssl req –new –x509 –days 3650 –key proftp.key –out proftp.crt

```
root@serveur-FTP-base:~# mkdir /etc/proftpd/ssl
root@serveur-FTP-base:~# cd /etc/proftpd/ssl
root@serveur-FTP-base:/etc/proftpd/ssl# openssl genrsa -out proftp.key 1024
Generating RSA private key, 1024 bit long modulus
.....................++++++
.........................++++++
e is 65537 (0x10001)
root@serveur-FTP-base:/etc/proftpd/ssl# openssl req -new -x509 -days 3650 -key p
roftp.key -out proftp.crt
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) [AU]:FR
State or Province Name (full name) [Some-State]:France
Locality Name (eg, city) []:Le Mans
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Mangetsu Services
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:mangetsu.slyip.net
Email Address []:mangetsu.services@hotmail.fr
```

On va ensuite éditer /etc/proftpd/tls.conf (lignes 10, 11, 12, 27, 28, 45, 49) :

 # nano /etc/proftpd/tls.conf

On dé-commente les lignes 10 à 12 :

```
TLSEngine                               on
TLSLog                                  /var/log/proftpd/tls.log
TLSProtocol                             SSLv23
```

On dé-commente et modifie les valeurs des lignes 27 et 28, on indique comme valeur pour la ligne 27 l'emplacement de notre certificat (/etc/proftpd/ssl/proftp.crt) et pour la ligne 28 l'emplacement de notre clé (/etc/proftpd/ssl/proftp.key) :

```
#
TLSRSACertificateFile                   /etc/proftpd/ssl/proftp.crt
TLSRSACertificateKeyFile                /etc/proftpd/ssl/proftp.key
#
```

On dé-commente les lignes 45 et 49 :

```
# Authenticate clients that want to use FTP over TLS?
#
TLSVerifyClient                         off
#
# Are clients required to use FTP over TLS when talking to this server?
#
TLSRequired                             on
#
```

On redémarre le service proftpd avec la commande **# /etc/init.d/proftpd restart**

On peut tester la connexion à nouveau mais cette fois-ci il faudra saisir : ftpes://@IP_serveur.

On peut vérifier que les échanges sont bien cryptés grâce à un logiciel d'analyse de trames comme WireShark.

On peut voir la fenêtre nous informant que le certificat est inconnu (ou pas si votre certificat est reconnu), mais aussi la partie d'authentification du certificat dans les logs de connexion :

Commande : *AUTH TLS*

Statut : *Initialisation TLS…*

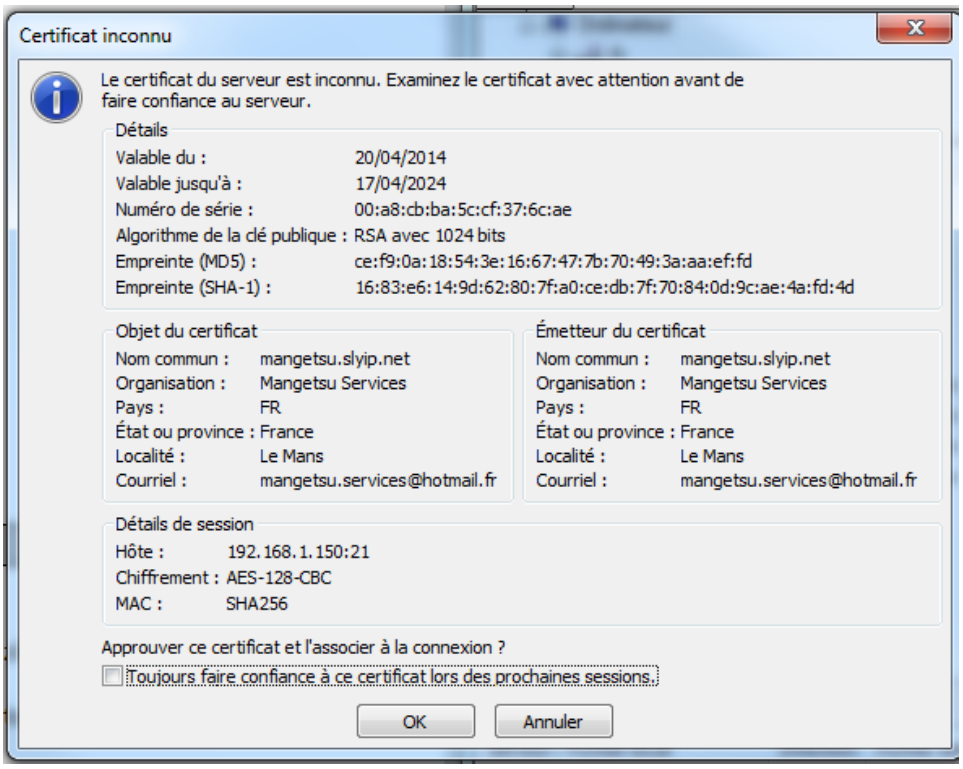
Statut : *Vérification du certificat…*

Statut : *Connexion TLS/SSL établie.*

Voir ci-dessous le certificat et les logs :

Hôte : xes://192.168.1.150 Identifiant : client1 Mot de passe : •••••••• Port : Connexion rapide

```
Statut :      Connexion à 192.168.1.150:21...
Statut :      Connexion établie, attente du message d'accueil...
Réponse :     220 Serveur FTP ::ffff:192.168.1.150 prêt
Commande :    AUTH TLS
Réponse :     234 AUTH TLS exécuté avec succès
Statut :      Initialisation de TLS...
Statut :      Vérification du certificat...
Commande :    USER client1
Statut :      Connexion TLS/SSL établie.
Réponse :     331 Mot de passe requis pour client1
Commande :    PASS ********
Réponse :     230 Utilisateur client1 authentifié
Commande :    SYST
Réponse :     215 UNIX Type: L8
Commande :    FEAT
Réponse :     211-Features:
Réponse :      MDTM
Réponse :      MFMT
Réponse :      TVFS
Réponse :      AUTH TLS
Réponse :      UTF8
Réponse :      MFF modify;UNIX.group;UNIX.mode;
Réponse :      MLST modify*;perm*;size*;type*;unique*;UNIX.group*;UNIX.mode*;UNIX.owner*;
Réponse :      PBSZ
Réponse :      PROT
Réponse :      SITE MKDIR
Réponse :      SITE RMDIR
Réponse :      SITE UTIME
Réponse :      SITE SYMLINK
Réponse :      REST STREAM
Réponse :      LANG fr-FR.UTF-8*;fr-FR;en-US.UTF-8;en-US
Réponse :      SITE COPY
Réponse :      SIZE
Réponse :     211 Fin
Commande :    OPTS UTF8 ON
Réponse :     200 UTF-8 activé
Commande :    PBSZ 0
Réponse :     200 PBSZ 0 exécuté avec succès
Commande :    PROT P
Réponse :     200 Protection set to Private
Statut :      Connecté
Statut :      Récupération du contenu du dossier...
Commande :    PWD
Réponse :     257 "/" est le répertoire courant
Commande :    TYPE I
Réponse :     200 Type paramétré à I
Commande :    PASV
Réponse :     227 Entering Passive Mode (192,168,1,150,39,18).
Commande :    MLSD
Réponse :     150 Ouverture d'une connexion de données en mode ASCII pour MLSD
Réponse :     226 Téléchargement terminé
Statut :      Contenu du dossier affiché avec succès
```



G. Scripts de gestion

Il n'est pas pratique de devoir saisir à la main les commandes SQL pour chaque utilisateur à créer, il n'est pas non plus souhaitable d'utiliser un utilitaire phpMyAdmin.

Une solution simple et complète et de faire un script. Ce script va permet de créer un utilisateur dans la base pour la connexion FTP :

```
#!/bin/bash
#Parametres : 3 parametres
    #$1 : identifiant
    #$2 : mot de passe
    #$3 : repertoire

#Identifiants SQL
    usersql='userftp'
    passwordsql='passftp'
    basesql='proftpd'
    tablesql='utilisateur'
    uid='UID'
    gid='GID'


#Recuperation des parametres

    #Recuperation de l identifiant
    login=$1

    #Recuperation du mot de passe
    password=$2

    #Recuperation du repertoire
    homeDirectory=$3


#Execution de la requete SQL
mysql -u $usersql -p$passwordsql --database=$basesql -e "INSERT INTO '$tablesql' VALUES ('"'$login'"', encrypt('"'$password'"'), '"'$uid'"', '"'$gid'"', '"'/var/www/$homeDirectory'"', '"'/bin/sh'"');"
```

Le script étant difficile à voir sur l'image ci-dessus, vous pouvez le télécharger à l'adresse suivante :

http://www.mangetsu.slyip.net/cle/Download/scripts/newUserFTP.sh

Son utilisation est simple, on fait appel au script, on renseigne trois paramètres qui sont respectivement : l'identifiant, le mot de passe et le Home Directory :

 # sh newUserFTP.sh client2 P@ssword client2

Attention, par défaut le script définit que le Home Directory se trouvera dans /var/www. Donc pour le répertoire /var/www/client2 on ne saisit en paramètre que la valeur « client2 ».

N'oubliez pas que pour exécuter le script, il faut le rendre exécutable :

 # chmod u+x newUserFTP.sh