Module n°1
Services réseaux et Sécurité

Auteur : Laboratoire Unix
0.1 – 13 mai 2004
Nombre de pages : 22

[Frame2]

Table des matières

Ce document est la propriété de Supinfo et est soumis aux règles de droits d'auteurs

1. SSH

1.1. Introduction

Le terme SSH (Secure SHell) désigne un ensemble de programmes et de protocoles qui remplacent les commandes **remote** (rlogin, rsh, rcp, telnet…), en offrant, en plus la confidentialité des échanges et l'authentification des correspondants. SSH vous permet par exemple de vous connecter à un ordinateur pour disposer d'une ligne de commande de manière transparente et sécurisée.

SSH a été créé par Tatu Ylonen. Il existe actuellement une version OpenSSH créée par Aaron Campbell qui est libre de droit. OpeSSH supporte plusieurs algorithmes symétriques pour le chiffrement, notamment le **trible DES** et **BlowFish**. En ce qui concerne l'authentification, il supporte le RSA et le DSA.

SSH est constitué de plusieurs parties :

sshd : Serveur ssh
scp : Copie distante sécurisée
ssh-keygen : génération de clefs d'authentification, management et conversion
sftp : Transfert sécurisé de fichiers
slogin/ssh : Client ssh
ssh-add : Ajoute les identités DSA ou RSA à l'agent d'authentification
ssh-agent : Agent d'authentification
ssh-keyscan : Recueille les clefs publiques ssh

1.2. Le client SSH

Le client ssh opère selon un ordre défini :

1. les paramètres en ligne de commande
2. les informations spécifiques à l'utilisateur du fichier **~/.ssh/config**
3. la configuration globale du système dans le fichier **/usr/local/etc/ssh_config** ou **/etc/ssh/ssh_config** (cela dépendant de la distribution que vous utilisez)

1.2.1. Le fichier /etc/ssh/ssh_config

Ce fichier permet de configurer le client ssh. Nous allons voir les différentes options disponibles.

Remarque : Le caractère de commentaire est le # comme dans beaucoup de fichiers de configuration.

Ce document est la propriété de Supinfo et est soumis aux règles de droits d'auteurs

Host permet de spécifier que la configuration qui suit concerne un ou plusieurs hôtes précis.
Il est possible d'employer des caractères joker tels que * ou ?.

```
# Host *
```

Autorise ou non la redirection du serveur graphique

```
# ForwardX11 no
```

Autorise la méthode d'authentification par Rhosts. Peu sûr.

```
# RhostsAuthentication no
```

Autorise la méthode d'authentification par Rhosts sur du RSA. Ne fonctionne que dans la version première du protocole et nécessite un setuid root. De préférence à ne pas utiliser.

```
# RhostsRSAAuthentication no
```

Méthode d'authentification par RSA, ne fonctionne qu'avec la première version du protocole. Il faut que le fichier identity existe. De préférence à ne pas utiliser.

```
# RSAAuthentication yes
```

Autorise la connexion par mot de passe, comme on pouvait la trouver dans rlogin ou telnet.

```
# PasswordAuthentication yes
```

Si l'indicateur **BatchMode** est positionné à yes, la demande de passphrase ou de mot de passe sera annulée. Cette option est utilisée dans le cas où seul des scripts interviennent et où il n'y a pas d'utilisateur présent pour insérer le mot de passe.

```
# BatchMode no
```

Vérification de l'adresse IP de l'hôte qui se connecte dans le fichier known_hosts

```
# CheckHostIP yes
```

Cette option permet de gérer l'ajout et le changement des clefs hôtes dans known_hosts.
Si la clef a changé, la connexion vous sera refusée, vous indiquant le motif.

```
# StrictHostKeyChecking ask
```

Localisation des fichiers identity, id_rsa, id_dsa

```
# IdentityFile ~/.ssh/identity
# IdentityFile ~/.ssh/id_rsa
```


Ce document est la propriété de Supinfo et est soumis aux règles de droits d'auteurs

```
# IdentityFile ~/.ssh/id_dsa
```

Port de connexion à l'hôte distant.

```
# Port 22
```

Version des protocoles supportés et désirés. Ici, on préfèrera employer la deuxième.

```
# Protocol 2,1
```

Caractère d'échappement.

```
# EscapeChar ~
```

Voici les options disponibles pour le client SSH :

-l login : Identifiant de l'utilisateur (il est possible d'utiliser la syntaxe **user@host**).

-v -vv -vvv : Mode verbose, permet d'obtenir les messages de debugage plus ou moins complets (plus il y a de v, plus vous obtenez d'information, le nombre maximum étant 3).

-1 ou -2 : version de ssh employé. Il est déconseillé d'employer la version 1 du protocole. Bien qu'aucun exploit public ne circule, les faiblesses cryptographiques du protocole ont été prouvées. De plus, la version 2 est reconnue par l'IETF.

-p port : Numéro du port distant.

Exemple :

```
ssh toto -l tata -p 42
```

-f : Permet de demander à ssh de se mettre en background après l'exécution de la commande. Cette option est recommandée pour le lancement d'une application graphique via ssh.

-X : Permet d'activer le forward X.

-L *port*:*host*:*hostport* : Demande à ssh de forwarder le port local ***port*** sur le port **hostport** de la machine distante ***host***.

-R *port*:*host*:*hostport* : Demande à ssh de forwarder le port distant ***port*** sur le port **hostport** de la machine locale ***host***.

Remarque : ***host*** peut être un nom ou une IP.

1.3.Création des clés

Ssh s'appuie sur des algorithmes à paire de clés, ce qui signifie que vous disposez d'une clé publique, disponible pour tout un chacun et une clé privée dont vous gardez précieusement l'entrée. Ce système

Ce document est la propriété de Supinfo et est soumis aux règles de droits d'auteurs

va nous permettre de nous identifier auprès des hôtes que nous désirons contacter. Il nous faut au préalable créer le trousseau. Pour cela il faut utiliser l'utilitaire **ssh-keygen**.

```
ssh-keygen -t rsa1 -f /etc/ssh/ssh_host_key
ssh-keygen -t rsa -f /etc/ssh/ssh host rsa key
ssh-keygen -t dsa -f /etc/ssh/ssh_host_dsa_key
```

La création des clefs est obligatoire lorsque OpenSSH est installé à partir sources. Dans la plupart des cas SSH est installé lors de l'installation du système, les clefs sont alors générées automatiquement.

1.4. Le serveur SSH

1.4.1. Le fichier /etc/sshd_config

Ce fichier permet de configurer le serveur ssh. Nous allons voir les différentes options disponibles.

La première chose à faire est de renseigner le serveur sur le port à utiliser et les protocoles SSH que nous souhaitons supporter :

```
Port 22
Protocol 2,1
```

Nous pourrions par exemple ne supporter qu'un seul protocole afin de simplifier la gestion des clients et des connexions. Un autre paramètre intéressant est :

```
ListenAddress <IP>
```

Celui-ci permet, dans le cas d'une machine possédant plusieurs interfaces réseau, de n'écouter que les connexions entrantes par l'interface possédant cette adresse IP.

Nous définissons ensuite les fichiers où se trouvent les différentes clefs de l'hôte :

```
HostKey /etc/ssh_host_key
HostKey /etc/ssh_host_rsa_key
HostKey /etc/ssh_host_dsa_key
```

Vous reconnaîtrez sans doute les fichiers que nous avons générés avec ssh-keygen dans l'étape précédente.

Le protocole SSH1 ne prévoit pas automatiquement de changement de clef de session en cours de connexion. Il est cependant possible de demander au serveur OpenSSH de le faire avec :

```
KeyRegenerationInterval 3600
ServerKeyBits 768
```

Nous spécifions un intervalle de temps en secondes pour la régénération et une taille pour les clés.

Viennent ensuite des paramètres concernant l'authentification avec, dans l'ordre, le temps accordé à la procédure de login et le refus ou l'autorisation au root de se connecter :

Ce document est la propriété de Supinfo et est soumis aux règles de droits d'auteurs

```
LoginGraceTime 600
PermitRootLogin no
```

Ici, le root n'a pas le droit de se connecter à notre serveur pour des raisons de sécurité (Cela évite les attaques de type brute force sur le compte root). Nous estimons que 600 secondes sont largement suffisantes pour qu'un utilisateur ait le temps de se loguer.

Nous entrons à présent dans le vif du sujet avec tout ce qui concerne les méthodes d'authentification des utilisateurs. Tout d'abord, nous devons choisir si, dans le cas du protocole SSH1, nous désirons une authentification RSA. Cette ligne ne concerne que le protocole SSH1 et permet de choisir entre une utilisation des clefs (rsa1) ou une authentification par mot de passe classique :

```
RSAAuthentication yes
```

Voici l'équivalent pour le protocole SSH2 avec authentification par clefs rsa ou dsa :

```
PubkeyAuthentication yes
```

Cette option qui va suivre ne devrait en aucun cas être mise à **yes**. Elle permet, en effet, une méthode d'authentification qui est loin d'être satisfaisante en termes de sécurité. Le principe est le même que pour le mécanisme **rhost** où, une fois les deux machines en présence sûres de leur identité, la connexion s'établit sans aucune vérification supplémentaire. OpenSSH, avec le protocole SSH1 ou SSH2, permet de faire la même chose de manière beaucoup plus sécurisée et surtout, avec une gestion des utilisateurs. Vous comprendrez alors que nous ne souhaitons pas faire usage de cette fonctionnalité :

```
RhostsAuthentication no
```

D'ailleurs, par la même occasion, nous ne prenons pas même le temps de lire les fichiers en cause :

```
IgnoreRhosts yes
```

Nous ne voulons pas même de la version utilisant un système rshost basé sur RSA :

```
RhostsRSAAuthentication no
```

Idem pour le protocole SSH2 :

```
HostbasedAuthentication no
```

Nous avons parlé de l'authentification RSA (SSH1) et Pubkey (SSH2), mais dans la phase de test, en cas de problème, nous souhaitons disposer d'une solution de rabattage utilisant le bon vieux système de mots de passe :

```
PasswordAuthentication yes
```

Cependant, nous ne sommes pas assez fous pour autoriser la connexion sur des comptes ne possédant pas de mot de passe :

```
PermitEmptyPasswords no
```


Ce document est la propriété de Supinfo et est soumis aux règles de droits d'auteurs

Il nous reste à configurer quelques éléments concernant les fonctionnalités à disposition après la phase d'authentification. Nous commençons par autoriser le tunneling de l'affichage X11 déporté et le display à mettre en place :

```
X11Forwarding yes
X11DisplayOffset 10
```

Toujours concernant la session alors qu'elle est déjà ouverte, l'option suivante permet, si elle est activée, d'envoyer régulièrement un message afin de tester la connexion. Ceci permet d'éviter que, si le client disparaît, la connexion reste ouverte indéfiniment :

```
KeepAlive yes
```

1.5. Lancement du serveur SSH

Il est possible de lancer le serveur SSH grâce a la commande **sshd** ou bien en utilisant le script de démarrage :

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

Les autres options (les plus courantes) étant :

- stop
- restart
- status
- reload

1.6. La copie sécurise

ssh fournit un outil de copie sécurisée en standard, sous le nom de **scp** pour **SecureCoPy**. Il remplace son ancêtre **rcp**. Son usage est très simple :

Pour transférer le fichier test1.txt situé dans le répertoire courant vers le home du compte toto de la machine distante ordi1.exemple.org sur laquelle tourne un serveur SSH :

```
# scp test1.txt toto@ordi1.exemple.org:
```

Pour récupérer le fichier test2.txt situé le home de l'utilisateur toto de la machine distante ordi2.exemple.org et l'écrire dans le répertoire courant :

```
# scp toto@ordi2.exemple.org:test2.txt .
```

Pour récupérer tous les fichiers ayant l'extension.txt situés dans le répertoire /usr/local de la machine ordi2.exemple.org et l'écrire dans le sous répertoire test-scp du répertoire courant :

```
# scp toto@ordi2.exemple.org:/usr/local/*.txt test-scp
```


Ce document est la propriété de Supinfo et est soumis aux règles de droits d'auteurs

Pour transférer l'intégralité du sous répertoire test-scp du répertoire courant vers le sous répertoire incoming du home de l'utilisateur toto de la machine ordi1.exemple.org :

```
# scp -r test-scp toto@ordi1.exemple.org:incoming
```

1.7. Xforwarding

Il est possible d'utiliser des applications graphiques via SSH. Pour cela il faut exporter le display et activer l'option **X11Forwarding.**

Exemple :

Vous êtes utilisateur de la machine locale machine1 qui a comme IP 192.168.1.1 et vous vous connectez en SSH à la machine distante machine2 qui a comme IP 192.168.1.2. Vous souhaitez lancer l'application **gtop** sur la machine machine2 mais que la fenêtre apparaisse sur votre serveur X local sur machine1. Voici comment procédez :

```
toto@machine1 # ssh user@machine2
Password :

user@machine2 # export DISPLAY=192.168.1.1:0
user@machine2 # gtop
```

Remarque : Votre serveur X local doit accepter les connexions venant de l'extérieur et doit être lancé sur le display 0 de la machine locale pour que le client X distant puisse joindre le serveur local. Les permissions du serveur X sont gérées grâce à la commande xhost (Pour accepter toutes connexions : # **xhost +**).

Il existe une autre méthode pour effectuer cela en une seule commande :

Exemple :

Forward d'une application X (gkrellm2 est un outil de monitoring graphique):

```
# ssh -X -f user@serveur DISPLAY=ip_machine_locale:0 gkrellm2
```


Ce document est la propriété de Supinfo et est soumis aux règles de droits d'auteurs

2. Service NFS

2.1.Présentation de NFS

Le service NFS permet à des ordinateurs d'architecture différente et qui exécutent un système d'exploitation différent de partager des systèmes de fichiers à travers un réseau, chaque système appliquant le modèle NFS aux sémantiques de son système de fichiers.
Développé par SUN Microsystems, son protocole utilise la méthode de communication RPC (Remote Procedure Call).
NFS est comparable au service de partage de fichier sous Windows (SAMBA)

2.2. Client NFS

Pour utiliser le client NFS il faut activer le support du système de fichiers NFS dans le noyau. Ainsi vous pourrez monter les partages NFS grave à la commande **mount** (expliquer dans le chapitre Système de fichiers).

Rappel

Pour monter le partage NFS /toto du serveur tati sur /mnt/temp :

```
# mount -t nfs tati:/toto /mnt/temp
```

Pour visualiser les partages NFS du serveur tata :

```
# showmount -e tata
```

2.3. Le serveur NFS

2.3.1. Les deamons

Pour activer un serveur NFS, il faut exécuter plusieurs démons au démarrage du système. Quand on installe NFS serveur, les scripts de démarrage des démons sont associes au niveau 3 de fonctionnement du processus init.

Daemon	Description
portmap	Le deamon **portmap** permet d'adresser un service RPC
rpc.mountd	Le deamon **rpc.mountd** réalise le montage demandé par un client
rpc.nfsd	Le deamon **nfsd** exécute les requetes NFS

2.3.2.

2.3.3. Le fichier /etc/exports

Ce document est la propriété de Supinfo et est soumis aux règles de droits d'auteurs

Le fichier **/etc/exports** indique les arborescences du serveur qui peuvent être monter à distance et par quel client. Son format est le suivant :

```
Répertoire client(droits/options)
```

Exemple :

```
/ftp/films toto(ro)          bui(rw)
/cours         (ro)
```

L'arborescence /ftp/films est accessible par les clients toto (lecture seule) et bui (lecture/écriture).
L'arborescence /ftp/films est accessible par tout le monde en lecture seule.

2.3.4. Lancer le serveur

Tout comme pour OpenSSH il existe un script permettant de contrôler le serveur :

```
/etc/init.d/nfs start
```

Les autres options (les plus courantes) étant :

- stop
- restart
- status
- reload

Ce script lance (ou arrête) tout les daemons nécessaire au serveur NFS.

2.4. Tester le service NFS

Plusieurs commandes permettent de tester le service NFS :

Commandes	Description
rpcinfo	Affiche les services RPC offert par le serveur
showmount	Affiche les clients NFS d'un serveur ou les ressources exportées d'un serveur.
nfsstat	Affiche des statistiques concernant les RPC et NFS

Ce document est la propriété de Supinfo et est soumis aux règles de droits d'auteurs

3. Serveur FTP

3.1.Introduction

Pure-ftpd est un serveur ftp libre, sécurisé, de qualité et répondant aux standard du protocole ftp. Il tourne sur de nombreux os telle que Linux, FreeBSD, NetBSD, OpenBSD, ISOS, EkkoBSD, BSDi, Solaris, Darwin, Tru64, Irix, AIX et HPUX.

Il possède de nombreux atouts tels que le “chroot” des repertoire home, le choix des ports pour le téléchargement passif, le support du fxp, le réglage de la bande passante, le support des ratios, le support de ldap / mysql / postgreSQL pour l'hautentification, et bien d'autres choses encore.

Les options du serveur (utilisateurs virtuel, support ldap, etc ...) ont besoin d'être spécifiés lors de la compilation. Les paquets binaires ont généralement toutes les options activées par défaut.

3.2.Installation

Les sources et les binaires sont disponibles à l'adresse suivante:

```
ftp://ftp.pureftpd.org/pub/pure-ftpd/releases
```

L'installation se fait via les commandes:

```
tar jxvf pure-ftpd-x.x.xxx.tar.bz2
cd pure-ftpd
./configure
make
make install (en root)
```

De nombreuses options sont disponibles pour le "**./configure**", vous en aurez la liste en tapant "**./configure --help**". Voir le chapitre 5.1 pour plus d'informations.

3.3.Utilisateurs virtuels

3.3.1. Introduction aux utilisateurs virtuels

Le système d'utilisateurs virtuels permet une gestion simple et efficace des utilisateurs de votre serveur ftp. De plus il vous permet d'instaurer des quotas, des limitations de bande passante ou de limiter la quantité de donnés pouvant être transféré, propre a chaque utilisateur.

Il est cependant fortement conseillé de placer vos utilisateurs virtuels sous l'uid et le gid d'un groupe et d'un utilisateur de votre système.

```
groupadd ftpgroup
useradd -g ftpgroup -d /dev/null -s /bin/false ftpuser
```

Ici l'utilisateur 'ftpuser' à comme groupe 'ftpgroup', comme répertoire personnel *"/dev/null"* et ne pourra pas se loguer sur votre système (le shell spécifié n'est pas valide). Cet utilisateur système ne possède donc strictement aucun pouvoir.

Ce document est la propriété de Supinfo et est soumis aux règles de droits d’auteurs

3.3.2. Création d'un utilisateur virtuel

Le fichier qui contient les utilisateurs possède un utilisateur par ligne et prend cette syntaxe la:

```
<account>:<password>:<uid>:<gid>:<gecos>:<home directory>:<upload
bandwidth>:<download bandwidth>:<upload ratio>:<download ratio>:<max
number of connections>:<files quota>:<size quota>:<authorized local
Ips>:<refused local Ips>:<authorized client IPs>:<refused client
Ips>:<time restrictions>
```

Seuls les champs <account>, <password>, <uid>, <gid> et <home directory> sont obligatoires.

Pour créer un utilisateur nous utilisons la commande **'pure-pw'**:

```
pure-pw useradd <login> [-f <passwd file>] -u <uid> [-g <gid>]
-D/-d <home directory> [-c <gecos>]
[-t <download bandwidth>] [-T <upload bandwidth>]
[-n <max number of files>] [-N <max Mbytes>]
[-q <upload ratio>] [-Q <download ratio>]
[-r <allow client host>[/<mask>][,<allow client host>[/<mask>]]...]
[-R <deny client host>[/<mask>][,<deny client host>[/<mask>]]...]
[-i <allow local host>[/<mask>][,<allow client host>[/<mask>]]...]
[-I <deny local host>[/<mask>][,<deny local host>[/<mask>]]...]
[-y <max number of concurrent sessions>]
[-z <hhmm>-<hhmm>] [-m]
```

Exemple:

```
pure-pw useradd user1 -u ftpuser -d /home/ftpusers/user1
```

L'option '-d' lui permet d'être "**chrooté"** dans son répertoire maison, au contraire de '-D' qui lui aurait donné accès à tous le système.

Vous pouvez à tout moment vérifiez les informations concernant un utilisateur avec la commande “**pure-pw show**”

```
pure-pw show    <login> [-f <passwd file>]
```

Exemple:

```
pure-pw show toto

Login              : toto
Password           : $1$LX/3.F60$bYdYwsQOYIaWq.Ko.hfI3.
UID                : 500 (ftpuser)
GID                : 101 (ftpgroup)
Directory          : /home/ftpusers/toto/./
Full name          :
Download bandwidth : 0 Kb (unlimited)
Upload   bandwidth : 0 Kb (unlimited)
Max files          : 1000 (enabled)
Max size           : 10 Mb (enabled)
Ratio              : 0:0 (unlimited:unlimited)
Allowed local  IPs :
Denied  local  IPs :
Allowed client IPs : 192.168.0.0/16
Denied  client IPs : 192.168.1.1,blah.verybadhost.com
Time restrictions  : 0900-1800 (enabled)
Max sim sessions   : 0 (unlimited)
```

Le './' a la fin de “directory” signifie que l'utilisateur est "**chrooté**".

Ce document est la propriété de Supinfo et est soumis aux règles de droits d’auteurs

3.3.3. Modification d'un utilisateur virtuel

Une fois votre utilisateur créé, vous pouvez le modifier avec la commande “**pure-pw usermod**”. Cette commande fonctionne comme pour l'ajout d'un utilisateur.

Ceci ajoute une limite de 1000 fichiers et de 10 mo à l'utilisateur user1:

```
pure-pw usermod user1 -n 1000 -N 10
```

Pour supprimer une option, il suffit de ne pas spécifier de valeur à l'option (avec des quottes):

```
pure-pw usermod user1 -n '' -N ''
```

La modification du mot de passe se fait via la commande "**pure-pw passwd**" :

```
pure-pw passwd <login> [-f <passwd file>] [-m]
```

3.3.4. Suppression d'un utilisateur virtuel

La commande "**pure-pw userdel**" nous permet de supprimer un utilisateur:

```
pure-pw userdel <login> [-f <passwd file>] [-m]
```

Le répertoire de l'utilisateur ne sera pas effacé, cette opération est laissée à votre charge.

3.3.5. Application des changements

Toutes les commandes citées jusqu'a présent modifient le fichier “**/etc/pureftpd.passwd**”. Cependant le serveur ftp utilise un autre fichier, qui lui est en binaire, pour des raisons de rapidité. Toutes vos actions ne sont donc pas effectives lors de vos commandes.

Afin de mettre à jour ce nouveau fichier, il faut utiliser la commande suivante:

```
pure-pw mkdb
```

Cependant vous pouvez lancez automatiquement cette mise à jour lorsque vous utilisez la commande '**pure-pw**' en lui donnant l'option '**-m**'. Les deux fichiers seront alors modifiés.

Le redémarrage du serveur pureftpd n'est pas nécessaire pour que les changements soient effectifs.

3.4.Le démon pure-ftpd

Le deamon pureftpd se lance avec toutes ses options en paramètre (Exemple "**/usr/local/sbin/pure-ftpd -lpuredb:/etc/pureftpd.pdb -H -B**")**.** Comme cela n'est guère pratique ni très parlant, la plupart des distributions vous proposent un fichier de configuration (**/etc/pure-ftpd.conf** par exemple). Nous ne parlerons ici que de la configuration telle que pure-ftpd la propose.

3.4.1. Démarrage et arrêt du serveur

Le démarrage se fait via le programme "**/usr/sbin/pure-ftpd**" ou "**/usr/local/sbin/pure-ftpd**". Si vous utilisez un binaire vous aurez accès à un script "**/etc/init.d/pure-ftpd**" à partir duquel vous pourrez démarrez ou arrêter votre serveur, tout en utilisant le fichier de configuration proposé par votre

Ce document est la propriété de Supinfo et est soumis aux règles de droits d’auteurs

distribution. Si vous avez installé à partir des sources, c'est à vous de créer un script de démarrage qui contiendra tous vos paramètres de configuration.

Création d'un script de démarrage simple:

```
root@chibbi:/home# vi /etc/init.d/pure-ftpd
```

Dans le fichier /etc/init.d/pure-ftpd:

```
#! /usr/bin/sh
/usr/local/sbin/pure-ftpd [mes options]
```

On créer ensuite un lien symbolique pour que le serveur soit lancé au démarrage:

```
ln -s /etc/init.d/pure-ftpd /etc/rc2.d/S95pure-ftpd
```

Vous pouvez également créer le lien symbolique en utilisant les outils de votre distribution (rc-update ou update-rc).

3.4.2. Autres programmes

L'utilitaire **pure-ftpwho** vous permet de consulter la liste des personnes présente sur votre serveur ftp.

```
root@chibbi:/home# pure-ftpwho

+------+---------+-------+------+---------------------------------+
| PID  |  Login  |For/Spd| What |        File/IP                  |
+------+---------+-------+------+---------------------------------+
|13357 | user1   | 00:00 | IDLE |                                 |
|  ''  |    ''   |   ''  |  ''  | ->              172.16.100.xx   |
+------+---------+-------+------+---------------------------------+
```

Il existe également d'autre programme à votre disposition, que je ne détaillerais pas ici. Ils commencent généralement par '**pure-**'.

3.5.Configuration avancée

3.5.1. Compilation

Certaines options doivent être précisés lors de la compilation (au stade du '**./configure**'). Les packages sont généralement compilés avec les options "--with-everything --with-paranoidmsg --without-capabilities --with-virtualchroot".

Options de compilations

Options	Description
`--with-altlog`	Permet de loguer les informations dans des formats spécifiques afin d'être utilisé par des analyseurs de logs (webalizer, Analog, etc …)
`--with-brokenrealpath`	Permet de garder une compatibilité avec certains systèmes solaris.
`--with-cookie`	Affiche un '**fortune**' ou une bannière customisée quand un utilisateur se logue.

Ce document est la propriété de Supinfo et est soumis aux règles de droits d'auteurs

--with-diraliases	Support des alias pour le parcours des répertoires.
--with-everything	Compilation avec toutes ces options : altlog, cookies, throttling, ratios, ftpwho, upload script, virtual users (puredb), quotas, virtual hosts, directory aliases et external authentication.
--with-extauth	Support pour les authentifications par des modules externes.
--with-ftpwho	Support de la commande 'pure-ftpwho'.
--with-language=french	Support de la langue française.
--with-largefile	Support des fichiers de plus de 2 go.
--with-ldap	Support de LDAP.
--with-minimal	Compilation avec un minimum d'options (déconseillé).
--with-mysql	Support de MySQL.
--with-nonroot	Permet de lancer le serveur sans être root. Vous n'aurez cependant pas accès à toutes les possibilités de pure-ftpd (déconseillé).
--with-pam	Support de PAM (pluggable authentification modules).
--with-paranoidmsg	Permet de supprimer les messages "user-friendly".
--with-peruserlimits	Permet de limiter le nombre de sessions qu'un utilisateur peut ouvrir.
--with-pgsql	Support de PostgreSQL.
--with-probe-random-dev	Permet d'utiliser "/dev/arandom", "/dev/urandom" ou "/dev/random" pour la génération des nombres aléatoires.
--with-puredb	Support des utilisateurs virtuels.
--with-quotas	Support des quotas.
--with-ratios	Support des ratios.
--with-sysquotas	Support des quotas du système et non des quotas propre a pure-ftpd.
--with-throttling	Support de la gestion de la bande passante.
--with-uploadscript	Support du programme 'pure-uploadscript'.
--with-virtualchroot	Permet de suivre les liens symboliques. Une utilisation courante est le pointage vers un dossier d'uploads commun à tous les utilisateurs.
--with-virtualhosts	Support des "virtual hosts". Les "virtual hosts" permettent notamment d'avoir plusieurs espaces sur votre ftp selon l'adresse IP spécifiée.
--with-welcomemsg	Permet d'utiliser un fichier 'welcome.msg' à la place de '.banner'.

Ce document est la propriété de Supinfo et est soumis aux règles de droits d'auteurs

--with-boring --with-privsep	Affiche les messages à aspect professionnel. Permet D'utiliser deux processus par client. Cela réduit les performances mais augmente la sécurité.
--without-ascii	Désactive le support des transferts en mode ASCII. (Déconseillé).
--without-banner	Désactive l'affichage de la bannière d'accueil.
--without-capabilities --without-globbing	Ignore la librairie "Libcap". Désactive les expressions régulières (tel que 'ls *.rpm') (déconseillé).
--without-humor	??
--without-inetd	Désactive le support du mode de démarrage inetd.
--without-iplogging	Désactive le logue des IP.
--without-nonalnum	Désactive le support des caractères non alphanumériques. (Déconseillé).
--without-sendfile	Permet de garder une compatibilité avec les systèmes de fichier réseau (comme NFS).
--without-shadow	Ignore les "shadow password". (Déconseillé).
--without-standalone --without-usernames	Désactive le mode de démarrage standalone. Affiche les uid ou gid plutôt que les noms des utilisateurs.

3.5.2. Options du serveur

Voici les différentes options avec lesquels le serveur peut être lancé. Certaines options nécessitent que leur équivalent ait été activé à la compilation.

Liste des options de configuration du serveur

Options	Description
-4	Le serveur refusera le format ipv6.
-a <gid>	Crée un groupe qui ne sera pas "chrooté".
-A	"Chroot" tout le monde sauf root.
-b -B	Assure une compatibilité avec certains clients ftp. Le serveur démarre en mode démon.
-c <nombre de clients>	Nombre maximum de client. 50 par défaut.

Ce document est la propriété de Supinfo et est soumis aux règles de droits d'auteurs

-C <nombre de connexion maximum par IP>	Nombre maximum de client par IP. Augmente la sécurité.
-d	Augmente le nombre d'informations loguer. Déconseillé.
-D	Affiche les fichiers caché sans que le client ne spécifie l'option '-a'.
-e	Le serveur n'acceptera que les connexions anonymes.
-E	Le serveur n'acceptera pas les connexions anonymes.
-f <facility>	Système utilisé pour les logs. 'ftp' par défaut.
-F <fichier>	Affiche un 'fortune' lors de la connexion.
-g <chemin du fichier pid>	Change l'emplacement du fichier contenant le pid du serveur.
-G	Refuse le renomage des fichiers.
-h	Affiche l'aide ☺.
-H -i	Le serveur ne cherche pas à transformer les IPs en noms dans les logs. Désactive l'upload pour les utilisateurs anonymes.
-I <minute>	Temps maximum avant la déconnexion d'un utilisateur inactif.
-J	Créer le répertoire maison de l'utilisateur automatiquement si cela n'est pas déjà fait.
-k <pourcentage>	Arrête les uploads si le pourcentage d'espace libre spécifié est dépassé.
-K	Autorise les utilisateurs a résumer ou uploader des fichiers, mais pas de les effacer ou renommer.
-l <auth> ou <auth>:<fichier de configuration>	Ajoute une nouvelle règle d'utilisation.
-L <nombre:nombre>	Permet de spécifier le nombre de fichier afficher par la commande 'ls' ainsi que le nombre de répertoire que parcours un 'ls' récursif. 2000 et 5 par défaut.
-m <charge>	Désactive les téléchargements anonymes si le serveur dépasse la charge spécifiée.
-M	Permet aux utilisateurs anonymes de créer des répertoires.
-N	Force le mode ACTIF.
-o	Permet d'utiliser le script 'pure-uploadscript'.
-O <format>:<fichier de log>	Permet d'écrire les logs dans des formats spécifiques. CLF (Apache-like), Stats et W3C.

Ce document est la propriété de Supinfo et est soumis aux règles de droits d'auteurs

-p <minport:maxport> -P <adresse IP>	Permet de spécifier les ports pour les connexions passives. Force l'IP passive du serveur.
-q <upload ratio>:<download ratio>	Active le ratio pour les utilisateurs anonymes.
-Q <upload ratio>:<download ratio>	Active les ratios pour tous (sauf root).
-r	Renomme les fichiers existant lors de l'upload.
-R	Désactive la commance 'CHMOD'
-s	Désactive la possibilité de télécharger les fichiers uploadés par des utilisateurs anonymes sans changement du groupe (généralement 'ftp') du fichier.
-S <address IP,port>	Spécifie le nom et le port du serveur. A mettre entre guillemets.
-t <bande passante (KB/s)>	Active la limitation de la bande passante pour les utilisateurs anonymes.
-T <bande passant (KB/s)> ou [<up bp>]:[<down bp>]	Active la limitation de la bande passante pour les utilisateurs authentifiés.
-u <uid>	Empêche les utilisateurs ayant un uid inférieur à celui spécifié de se loguer.
-U <masque>	Masque de création des fichiers. 133:022 par défaut.
-V <addresse IP>	Permet de spécifier une IP pour les utilisateurs authentifiés (ex 10.x.x.x).
-w	Support du protocole FXP pour les utilisateurs authentifiés.
-W	Support du protocole FXP pour les utilisateurs anonymes (déconseillé).
-x	Interdit l'écriture sur les fichiers cachés.
-X	Interdit la lecture sur les fichiers cachés.
-y <max par utilisateur>:<max par session anonyme>	Limite le nombre de sessions qu'un utilisateur peut avoir.
-Y <0 | 1 | 2 >	Désactive le ssl/tls, Accepte les deux standard, ou refuse les connexions n'utilisant pas le ssl/tls.
-z	Autorise les utilisateurs anonymes à lire les fichiers cachés.
-Z	Empêche les utilisateurs de ne plus pouvoir accéder à leurs fichiers suite a une mauvaise manipulation de 'chmod'.

Ce document est la propriété de Supinfo et est soumis aux règles de droits d'auteurs

3.5.3. Exemple

On crée un groupe et un utilisateur pour le ftp:

```
groupadd ftpgroup
useradd -g ftpgroup -d /dev/null -s /etc ftpuser
```

On crée les répertoires pour le ftp (se sont les permissions systèmes qui régissent le ftp, il ne faut pas oublier de les régler):

```
mkdir -p /home/ftp/incoming
chown -R root:ftpgroup /home/ftp/incoming
chmod -R 755 /home/ftp
chmod -R 1775 /home/ftp/incoming
```

On ajoute des utilisateurs:

```
pure-pw useradd user1 -m -u ftpuser -d /home/ftp
pure-pw useradd user2 -m -u ftpuser -d /home/ftp
```

On télécharge les sources et on installe avec les options voulues:

```
wget ftp://ftp.pureftpd.org/pub/pure-ftpd/releases/pure-ftpd-
1.0.16.tar.bz2
tar jxvf pure-ftpd-1.0.16.tar.bz2
cd pure-ftpd-1.0.16a
./configure --with-altlog --with-brokenrealpath --with-cookie --
with-diraliases --with-ftpwho --with-language=french --with-
peruserlimits --with-puredb --with-quotas --with-ratios --with-
throttling --with-virtualhosts --with-welcomemsg --without-inetd
make
make install (en root)
```

On crée notre script de démarrage:

```
root@chibbi:/home# vi /etc/init.d/pure-ftpd
```

```
#! /usr/bin/sh
/usr/local/sbin/pure-ftpd -b -B -c 5 -C 3 -E -H -K -U 111:022 -l
puredb:/etc/pureftpd.pdb
```

```
ln -s /etc/init.d/pure-ftpd /etc/rc2.d/S95pure-ftpd
```

Dans cet exemple, nous avons:

- Le serveur est lancé en mode démon et assure une compatibilité avec certains clients (-b -B)
- Le serveur peut avoir 5 connections simultanées dont 3 seulement de la même IP (-c 5 -C 3)
- Les utilisateurs anonymes sont refusé (-E)
- Les logs contiennent directement les IP (-H)
- Les utilisateurs peuvent uploader dans le dossier "/home/ftp/incoming", tout nouveau fichier uploadé aura comme permissions 666 (possibilité de résumer les fichiers). Mais ils ne pourront pas les effacer. (-U 111:022 -K)
- Le serveur utilisera les utilisateurs virtuels (-l puredb:/etc/pureftpd.pdb)

3.6. Lexique

Mots	Définitions

Ce document est la propriété de Supinfo et est soumis aux règles de droits d'auteurs

chroot	Permet d'enfermer un utilisateur dans un répertoire. Il ne pourra alors plus accéder aux répertoires parents mais aura accès à tous les répertoires enfant.
fortune	Utilitaire permettant de lire des phrases aléatoirement dans un fichier.

Ce document est la propriété de Supinfo et est soumis aux règles de droits d'auteurs