# OpenSSH

Ce tutoriel développe la mise en place d'un serveur SSH2 et d'un client.

## Installation des packages

(sur chaque machines)

| Mandrake | debian |
|---|---|
| openssh-3.6.1p2-12.rpm<br>openssh-server-3.6.1p2-12.rpm<br>openssh-clients-3.6.1p2-12.rpm | ssh_3.8.1p1-8_i386.deb |

## Configuration du serveur /etc/ssh/sshd_config

```
#    $OpenBSD: sshd_config,v 1.59 2002/09/25 11:17:16 markus Exp $
# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.
# This sshd was compiled with
PATH=/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin
# The strategy used for options in the default sshd_config shipped
with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options change a
default value.

Port 22

# on utilise que SSH2
# si vous utilisez SSH1 et 2 vous mettrez Protocol 2,1
Protocol 2

# adresse IP de l'interface d'où vont arriver les requêtes
ListenAddress 192.168.0.1

# HostKey for protocol version 1
# HostKey /etc/ssh/ssh_host_key
# HostKeys for protocol version 2
# Attention le fichier doit avoir pour droit 600
# HostKey /etc/ssh/ssh_host_rsa_key
# on autorise que les clefs DSA
HostKey /etc/ssh/ssh_host_dsa_key

# Ne sert à rien pour SSH2
# Lifetime and size of ephemeral version 1 server key
#KeyRegenerationInterval 3600
#ServerKeyBits 768
```

```
# Logging
#obsoletes QuietMode and FascistLogging
SyslogFacility AUTH
LogLevel INFO

# Authentication:

# le serveur se déconnecte au bout de 120s si l'utilisateur n'a pas
réussi à se loguer
LoginGraceTime 120

# si vous devez réaliser des sauvegardes par rsync, il faut vous
# connecter en root pour conserver les owners. Sinon mettre no
PermitRootLogin yes

# Le serveur vérifie les droits et le propriétaire du home directory
avant d'accepter la connexion
StrictModes yes

# en cas d'authentification RSA Protocol 1
RSAAuthentication yes

# Si protocole SSH2 mettre à yes
PubkeyAuthentication yes

# fichier contenant les clés publiques
AuthorizedKeysFile  .ssh/authorized_keys

# Concerne SSH1
# rhosts authentication should not be used
# RhostsAuthentication no

# 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/ssh_known_hosts
RhostsRSAAuthentication no

# similar for protocol version 2
HostbasedAuthentication no

# Change to yes if you don't trust ~/.ssh/known_hosts for
# RhostsRSAAuthentication and HostbasedAuthentication
IgnoreUserKnownHosts yes

# Autorise la connexion par mot de passe
# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes

# refuse les mots de passe vide
PermitEmptyPasswords no
```

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

# forwarde X
X11Forwarding yes

# le DISPLAY sera fixé à serveur-ssh:10.0
X11DisplayOffset 10

X11UseLocalhost yes

# On peut afficher à la connexion un message contenu dans /etc/motd
PrintMotd no

# pour afficher la dernière date et heure de login
PrintLastLog yes

# lors d'une connexion, vérification régulière pour voir si le serveur
# n'est pas down pour pouvoir avertir l'utilisateur à temps
KeepAlive yes

# Laisser à no sinon le X forwarding ne marche pas
UseLogin no

# pour que certaines opérations soient réaliser en tant que non root
UsePrivilegeSeparation yes

PermitUserEnvironment no

# compression des données
Compression yes

MaxStartups 10

# no default banner path
#Banner /some/path

# Pour avoir le ftp sécurisé
Subsystem sftp /usr/lib/ssh/sftp-server

# pour spécifier les utilisateurs et groupes d'utilisateurs habilités
à se connecter ou non
#DenyUsers hacker pirate
#AllowUsers replicateur
#DenyGroups users
#AllowGroups cdwriter adm
```

Configuration du client /etc/ssh/ssh_config

```
#    $OpenBSD: ssh_config,v 1.16 2002/07/03 14:21:05 markus Exp $
# This is the ssh client system-wide configuration file.  See
# ssh_config(5) for more information.  This file provides defaults for
# users, and the values can be changed in per-user configuration files
# or on the command line.
# Configuration data is parsed as follows:
#  1. command line options
#  2. user-specific file
#  3. system-wide file

# On peut spécifier des paramètres différents suivant les serveurs.
# dans ce cas tous les serveurs auront la meme configuration
 Host *

# ne forwarde pas l'agent d'authentification (ssh-agent)
   ForwardAgent no

# forwarde X
   ForwardX11 yes

# concerne SSH1
   RhostsAuthentication no
   RhostsRSAAuthentication no

#   RSAAuthentication yes

# Autorise la connexion par mot de passe
   PasswordAuthentication yes

#   HostbasedAuthentication no

# si vous ne voulez ni rentrer la passphrase ni le mot de passe, on
met à yes
# utilisé pour lancer une connexion ssh en mode batch (dans un script)
   BatchMode no

# Vérification de l'adresse IP du serveur par rapport a known_host
   CheckHostIP yes

# Avant d'ajouter la clé publique d'un serveur
# dans known_host ssh va demander à l'utilisateur
   StrictHostKeyChecking ask

# fichier pour SSH1 seulement
#   IdentityFile ~/.ssh/identity
# clés DSA pour SSH2
#   IdentityFile ~/.ssh/id_rsa
   IdentityFile ~/.ssh/id_dsa

   Port 22

   Protocol 2
```

```
# on utilise SSH2 donc à yes
   PubkeyAuthentication yes

# type de chiffrement pour SSH1
#   Cipher 3des
# type de chiffrement pour SSH2 dans l'ordre de préférence
   Ciphers aes128-cbc,3des-cbc,blowfish-cbc,cast128-
cbc,arcfour,aes192-cbc,aes256-cbc

   EscapeChar ~

# compression des données
Compression yes
# précise le niveau de compression, valable pour Protocol 1 seulement
(de 1 faible à 9 fort)
#CompressionLevel 2
```

Utilisation de SSH

Vous avez 2 possibilités pour vous connecter :
1) Par mot de passe sans utilisation de clef (dsa ou rsa)
2) Par utilisation de clef sans saisir de mot de passe (sauf si ajout d'une passphrase)

Connexion par mot de passe :

```
[user@pc user]$ ssh unlogin@unserveur
(à la 1er connexion vers ce serveur, votre fichier .ssh/known_hosts ne
contient pas l'identité de ce serveur)
The authenticity of host 'unserveur (192.168.0.3)' can't be
established.
DSA key fingerprint is
51:a7:82:69:19:21:0c:7a:d0:f2:34:1a:5a:8b:07:a7.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'unserveur,192.168.0.3' (DSA) to the list
of known hosts.
unlogin@unserveur's password:( password de unlogin)
[unlogin@pc3 unserveur]$
```

Connexion par paires de clefs :

Il faut générer ses paires de clefs, deux algorithmes sont possible : DSA et RSA

```
[user@pc user]$ ssh-keygen -d
Generating public/private dsa key pair.
Enter file in which to save the key (/home/user/.ssh/id_dsa): (bon par
défaut)
Enter passphrase (empty for no passphrase):
(une passphrase vous est demandé(facultatif), si vous voulez encore
plus de sécurité saisissez en une)
...

[user@pc user]$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
...
```

Il faut transmettre votre id_dsa.pub ou id_rsa.pub à la personne vers qui vous voulez vous connecter, pour qu'elle créer un fichier ~/.ssh/authorized_keys contenant votre clef publique (sur une seule ligne !)
Faire de meme sur votre serveur avec la sienne.
L'échange peut se faire par disquette, mail, ... ou plus adapté encore par SSH :

Il y a deux solutions :
1) Copie manuel de votre clef publique vers la personne
2) Copie automatique de votre clef publique vers la personne

Copie manuel avec scp :

```
[user@pc user]$ scp .ssh/id_dsa.pub unlogin@unserveur:/home/
unlogin/.ssh/dsapub-user
Warning: Permanently added 'unserveur,192.168.0.3' (DSA) to the list
of known hosts.
unlogin@unserveur's password:( password de unlogin)
id_dsa.pub  100% |*********************************| 610  00:00
[user@pc user]$ (votre clef a était copié chez unlogin, à lui de
copier son contenu dans ~/.ssh/authorized_keys)
```

Copie automatique avec ssh-copy-id :

```
[user@pc user]$ ssh-copy-id -i .ssh/id_dsa.pub unlogin@unserveur
The authenticity of host 'unserveur (192.168.0.3)' can't be
established.
DSA key fingerprint is
51:a7:82:69:19:23:0c:7a:d0:f2:34:1a:5a:8b:07:a7.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'unserveur,192.168.0.3' (DSA) to the list
of known hosts.
unlogin@unserveur's password:
Now try logging into the machine, with "ssh 'unlogin@unserveur'", and
check in:
  .ssh/authorized_keys
```

```
to make sure we haven't added extra keys that you weren't expecting.
[user@pc user]$
```
Votre clef a été insérée sans rien faire !

Pour vous connecter dans les deux cas vous n'aurez plus qu'a taper :
```
[user@pc user]$ ssh unlogin@unserveur
[unlogin@pc3 unserveur]$
```

## Xforwarding

Le Xforwarding permet d'exporter des applications distantes.
Vous lancez un logiciel installé sur une machine distante depuis votre machine en utilisant les ressources de la machine distante.

```
[user@pc user]$ ssh unlogin@unserveur gmplayer
```

## Applications SSH

ssh-keygen => générer ses clefs SSH
ssh => connexion vers un serveur
    scp => copie distante par SSH
    sftp => FTP par SSH
ssh-copy-id => insérer votre clef publique dans l'authorized_keys d'une personne.
ssh-agent => lancé au début d'une session, il retient la passphrase
    ssh-add => ajoute les identités à l'agent

Il existe des interfaces pour utiliser SSH :
Sous Linux il y a Secpanel (interface pour scp)
Sous Window$ il y a : WinSCP (interface pour scp + sftp), Secure iXplorer et PuTTY (console)

Sources :
http://www.funix.org/
http://lea-linux.org/reseau/
http://www.mandrakesecure.net/en/docs.php#papers

Document mis à jour : 14/09/04