Annexe – SSH
Secure SHell

Sommaire :

I - Introduction

SSH **S**ecure **SH**ell, propose un shell sécurisé pour les connexions à distance et se présente dans ce domaine comme le standard. On peut qualifier **ssh** de **super-telnet-crypté**.

Le service **SSH** est construit sur un modèle **Client-Serveur**. Avec **SSH** la totalité de la transaction entre un client et le serveur est **cryptée**. **SSH** utilise le protocole de transport **TCP** (**T**ransport **C**ontrol **P**rotocol). Le **serveur SSH** utilise le **port 22**.

Ce protocole est utilisé généralement avec un mécanisme d'**authentification par mot de passe**. Lors de la première connexion du client au serveur, le serveur propose d'envoyer **la clé publique de chiffrement** au client :

```
jcabianca@jcabianca-HP-PC:~$ ssh etudiant@192.168.43.45
The authenticity of host '192.168.43.45 (192.168.43.45)' can't be established.
ECDSA key fingerprint is SHA256:Cf2h/nVfzceNWJnxtFh2iDIMPYmHNHNaac0aTMnBiRk.
Are you sure you want to continue connecting (yes/no)?
```

On accepte en saisissant **yes**. La **clé de chiffrement** est maintenant sauvegardée sur le client. Le client génère alors une **clé secrète** (privée) et l'envoie au serveur, en cryptant l'échange avec la **clé publique** du serveur (chiffrement **asymétrique**). Le client et le serveur peuvent alors établir un canal sécurisé grâce à la clé secrète commune (chiffrement **symétrique**).

II - Installation et utilisation de SSH

II.1. Installation de SSH

Il faut installer les packages **openssh-server** et **openssh-client** en tapant la commande :
apt-get install ssh

On pourra stopper|démarrer|redémarrer ce service en tapant la commande :
service ssh stop|start|restart

II.2. Utilisation de SSH

Depuis un poste **Unix/Linux**, la connexion au serveur **ssh** se fait en tapant la commande :

ssh user@serveur

serveur : l'adresse IP ou le nom du serveur auquel on veut se connecter ;
user : un compte valide défini sur le serveur.

Depuis un poste **Windows**, on pourra utiliser l'utilitaire **Putty**. Il suffit de saisir l'adresse IP ou le nom du serveur ssh puis d'ouvrir la connexion et d'accepter la clé de chiffrement.

II.3. Configuration de SSH

Les informations de configuration de SSH qui s'appliquent à l'ensemble du système sont stockées dans le fichier /**etc/ssh/sshd_config**. Voici les directives principales contenues dans ce fichier :

Port 22	Port par défaut
ListenAddress 0.0.0.0	*A* l'écoute du reste du monde
KeyRegenerationInterval 3600	Période de régénération de la clef du serveur (1 heure)
PermitRootLogin without-password	Interdit la connexion avec le login root. Remplacer **without-password** par **Yes** pour autoriser la connexion. Attention !!
IgnoreRhosts yes	Interdit l'utilisation du fichier rhosts, méthode non sécurisée
StrictModes yes	Vérifie la sécurité du répertoire perso avant d'autoriser le login
FascistLogging no	N'enregistre pas toutes la transactions. Il est conseillé de respecter la vie privée de l'utilisateur
PrintMotd yes	Le Message Of The Day « message de bienvenue »
SyslogFacility DAEMON	Système d'enregistrement des logs
RhostsRSAAuthentication yes	Ajoute la sécurité du fichier rhosts à celle du système RSA
RSAAuthentication yes	Authentification RSA activée
PasswordAuthentication no	Authentification par mot de passe en cas d'authentification RSA échouée. A mettre à no dès que le système RSA est opérationnel.(pas avant ça ne marcherait pas par scp.)
PermitEmptyPasswords no	Interdit la connexion par mot de passe vide en cas d'authentification par mot de passe. A DESACTIVER !!
AllowHosts 10.*	Liste des hôtes/domaines autorisés. Vous pouvez préciser des IP ou des noms qualifiés. Attention notez les domaines génériques en utilisant le joker *. Par exemple : 10.* pour le domaine 10.0.0.0/8.
DenyHosts ALL	Hôtes interdits. ALL permet de verrouiller les accès par Allowhosts.
Umask 022	Les droits par défaut donnés au répertoire.

III - Utilisation de scp et sftp

III.1. Introduction

Avec **SSH**, des programmes de la même famille comme « **scp** » ou « **sftp** » remplacent les commandes **rcp** ou **ftp**.

L'utilisation de ces commandes est relativement simple :
- **scp** (**S**ecure **C**o**P**y) permet de faire de la copie de fichiers ;
- **sftp** (**S**ecure **FTP**) est utilisable en mode interactif ou en mode batch et ressemble plus au **FTP**.

III.2. Utilisation de scp

syntaxe générale :

scp [-r] source destination, où **source** et **destination** désigne respectivement l'ensemble des fichiers à copier et le répertoire d'accueil.

Si les fichiers sont locaux, on utilise la syntaxe habituelle. S'ils sont distants, la notation est celle de **ssh** : **user@serveur:fichiers** .

Exemples :

scp -r user@serveur:fichiers rep-local, pour copier du serveur distant les fichiers vers le répertoire rep- local d'accueil local .

scp -r fichiers-locaux user@serveur:rep, pour copier les fichiers locaux vers le répertoire rep situé sur le serveur distant .

Pour copier les fichiers qui sont dans le répertoire « /home/eleve/test » du serveur « myserver.mydomain » dans un répertoire local « essai » on utilise :

```
cd
mkdir essai
scp eleve@myserver.mydomain:/home/eleve/test/* ~/essai
```

Pour envoyer les fichiers du répertoire local « test », vers le répertoire « /home/eleve/tmp » de la machine « myserver.mydomain » on utilise :

```
scp ~/test/* eleve@myserver.mydomain:/home/eleve/tmp
```

III.3. Utilisation de sftp

sftp peut être utilisé pour du transfert de fichier en mode sécurisé.

```
sftp eleve@myserver.mydomain
sftp>
```

On obtient le prompt « **sftp>** ». Pour avoir la liste des commandes, utiliser « **help** » ou « **?** » :

```
sftp> help
Available commands:
bye                                Quit sftp
cd path                            Change remote directory to 'path'
get [-afPpRr] remote [local]       Download file
put [-afPpRr] local [remote]       Upload file
etc ….
```