Serveur SSH

I Présentation

Il existe deux protocoles connus pour se connecter à distance :

- **Telnet** : Telnet est le protocole le plus basique pour se connecter à distance à un ordinateur. Il n'est presque plus utilisé, car il ne crypte (protège) pas les données, alors si un pirate « écoute » votre connexion au serveur, il pourra avoir le mot de passe du serveur, par exemple ! Telnet sert beaucoup pour tester des services (HTTP, POP, SMTP, etc.) et savoir s'ils fonctionnent.
- **SSH** : Secure Shell (SSH) est à la fois un programme informatique et un protocole de communication sécurisé. Le protocole de connexion impose un échange de clés de chiffrement en début de connexion. Par la suite, tous les segments TCP sont authentifiés et chiffrés. Il devient donc impossible d'utiliser un sniffer pour voir ce que fait l'utilisateur.

Bref, SSH est recommandé, car il crypte les données et ceci rend difficile le piratage.

II Installation du serveur ssh

Le paquet qui comporte le serveur ssh s'appelle : « openssh-server » :

```
root@debian:~# apt-get install openssh-server
```

Pour activer le serveur ssh, il faut saisir dans un terminal la commande suivante :

```
root@debian:~# systemctl start ssh
```

Pour l'arrêter :

```
root@debian:~# systemctl stop ssh
```

Pour le relancer :

```
root@debian:~# systemctl restart ssh
```

III Configuration du serveur ssh

Pour pouvoir configurer SSH, vous devez éditer le fichier de configuration « **/etc/ssh/sshd_config** » dont les sections suivantes :

- La ligne **Port** permet de configurer le port d'écoute du serveur. Pour une question de sécurité, vous pouvez changer cette option, car elle va nuire un peu aux pirates.

```
# What ports, IPs and protocols we listen for
Port 22
```

- La ligne **ListenAdress** permet de dire au serveur d'écouter seulement sur certaines adresses IP. Pour pouvoir faire cela, enlevez le # du deuxième ListenAdress et remplacez 0.0.0.0 par votre adresse IP.

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

- La section **Authentication** permet de configurer l'authentification.

```
# Authentication:
LoginGraceTime 120
PermitRootLogin yes
```

- L'option **LoginGraceTime** est le temps d'attente maximum après une connexion à SSH pour l'identification. Le temps est exprimé en secondes.
- **PermitRootLogin** permet de dire si SSH autorise ou non la connexion au compte root. Par mesure de sécurité, il vous est recommandé de mettre cette option à no.

✗ L'option DenyUsers peut être ajouté pour interdire la connexion d'utilisateur :

```
DenyUsers user1 user2 user3
```

Après modification de ce fichier, il faut redémarrer le service ssh :

```
root@debian:~#  systemctl restart ssh
```

IV Connexion à une machine distante en utilisant ssh

IV.1 Installation du client

La partie cliente est fournie par le paquet openssh-client, qui est en général installé par défaut. Si vous installez ssh, le serveur et le client sont installés.

Remarque : Si vous devez prendre le contrôle depuis un poste tournant sous Windows, vous pouvez installer PuTTY ou Tera Term.

IV.2 Utilisation du client

La commande pour se connecter à distance est :

```
ssh votre_adresse -l utilisateur
```

ou

```
ssh utilisateur@votre_adresse
```

Par exemple pour se connecter avec l'utilisateur « util » qui doit bien sûr exister sur le poste ayant l'adresse IP « 192.168.0.200 ».

```
root@debian:~# ssh util@192.168.0.200
```

Normalement, par défaut l'utilisateur « root » n'a pas le droit de se connecter à une machine distante par ssh. On est obligé de se connecter à la machine distante avec un autre utilisateur. C'est une des raisons qui fait que lors de l'installation du système d'exploitation (Debian) en plus de l'utilisateur root, on vous demande de créer un utilisateur usuel.

V Copie de fichiers ou répertoire en utilisant la commande scp

Le transfert de fichiers par SSH est possible grâce à la commande scp (comme Ssh CoPy), qui s'utilise de la même manière que la commande cp :

✗ Copie d'un fichier d'une machine serveur1 vers une autre machine serveur2 :

```
scp Login1@Serveur1:Chemin1/NomFichier1 Login2@Serveur2:Chemin2/NomFichier2
```

✗ Copie d'un fichier depuis le répertoire courant vers un répertoire du serveur :

```
scp Fichier login@serveur:Chemin
```

✗ Copie d'un répertoire, avec éventuellement ses sous-répertoires, vers un répertoire du serveur :

```
scp -r Repertoire login@serveur:Chemin
```

✗ Copie d'un fichier du serveur vers le répertoire courant :

```
scp login@serveur:Chemin/Fichier .
```

✗ Copie d'un répertoire du serveur vers le répertoire courant :

```
scp -r login@serveur:Chemin/Repertoire .
```

Par exemple pour copier le fichier « image.png » de la machine locale vers le répertoire « /home/util/images » de la machine distante ayant pour adresse IP « 192.168.0.200 » :

```
root@debian:~# scp  /home/user/image.png  util@192.168.0.200:/home/util/images/
```

ou bien dans le sens inverse de la machine distante vers la machine locale :

```
root@debian:~# scp  util@192.168.0.200:/home/util/images/image.png  /home/user/
```