

## Compte rendu TP SSH

### 1) Se connecter en SSH sur la VM

La syntaxe d'une connexion ssh est : `ssh -p numero_port_source login@ip_source`

Pour se connecter en ssh à la machine virtuelle, on utilise donc la commande :

```
ssh -p 2222 lpasr@127.0.0.1
```

On utilise l'option `-p 2222` pour se connecter au port 2222 de l'hôte 127.0.0.1

- 2222 est le port qui est redirigé vers le port 22 de la machine virtuelle serveur SSH

- On utilise l'adresse IP 127.0.0.1 car VirtualBox est sur la même machine que le shell qui lance la connexion SSH.

### 2) Vider le fichier knownhost

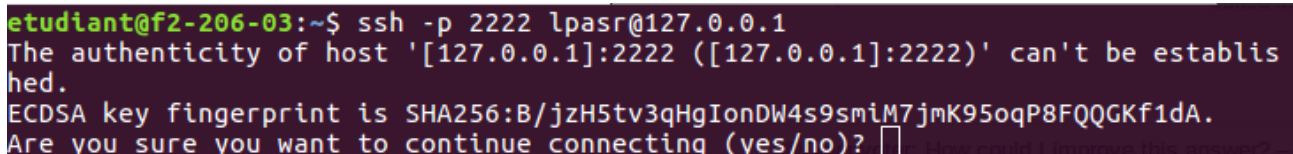
On vide l'entrée du fichier `known_hosts` du CLIENT via la commande :

```
ssh-keygen -R [127.0.0.1]:2222
```

L'option `-R` de `ssh-keygen` permet de supprimer l'emprunte d'un host dans le fichier.

Comme on utilise pas le port 22 pour se connecter, il faut préciser de supprimer l'emprunte en indiquant le bon numéro de port.

### 3) Se reconnecter : qu'est-ce qui change ?



```
etudiant@f2-206-03:~$ ssh -p 2222 lpasr@127.0.0.1
The authenticity of host '[127.0.0.1]:2222 ([127.0.0.1]:2222)' can't be established.
ECDSA key fingerprint is SHA256:B/jzH5tv3qHgIonDW4s9smIM7jmK95oqP8FQQGKf1dA.
Are you sure you want to continue connecting (yes/no)?
```

Cette fois, un message nous indique que l'authenticité de l'hôte auquel on souhaite se connecter n'a pas pu être établie. En effet, on a supprimé l'identité du serveur dans le fichier `known_hosts`.

Le fichier `known_hosts` contient la liste des hôtes connus avec leur empreinte. Si on supprime une l'entrée d'un host, on ne connaît plus l'empreinte de l'hôte auquel on souhaite se connecter et il est donc impossible de savoir si quelqu'un n'essaie pas d'usurper son identité.

Si on répond « yes », un message nous indique : « Warning: Permanently added '[127.0.0.1]:2222' (ECDSA) to the list of known hosts. »

L'empreinte de l'hôte a donc été enregistrée dans le fichier `known_hosts` du client.

#### 4) Générer une nouvelle paire de clés SSH

Pour générer une nouvelle paire de clés ssh on utilise la commande `ssh-keygen` depuis le dossier `.ssh` :

```

etudiant@f2-206-03:~/.$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/etudiant/.$ssh/id_rsa): cle_vm_ubuntu
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in cle_vm_ubuntu.
Your public key has been saved in cle_vm_ubuntu.pub.
The key fingerprint is:
SHA256:uPKTRylB9LWfKjLRr7KReIDJZ9bJio/ka2A4/sOpSZc etudiant@f2-206-03
The key's randomart image is:
+----[RSA 2048]-----+
|          .. .          |
|          .. . .        |
| . . . . .              |
|+O O O.+ . .           |
|+.+ = *.S. O           |
|.. =.+.+O. .           |
| o=E= B+. O            |
|.O+B ++++O             |
| O+.O O=.              |
+----[SHA256]-----+

```

## 5) Copier la clé publique sur le serveur

On copie la clé ssh dans le dossier `.ssh` du serveur.

Il faut donc au préalable créer manuellement le répertoire `.ssh` dans le `homedir` du serveur :

```
mkdir .ssh
```

On utilise ensuite la commande `scp -P 222 cle_vm_ubuntu.pub`

```
lpasr@127.0.0.1 :~/.ssh
```

```
etudiant@f2-206-03: ~/.ssh$ scp -P 2222 cle_vm_ubuntu.pub lpasr@127.0.0.1:~/.ssh
lpasr@127.0.0.1's password:
cle_vm_ubuntu.pub                                100% 400      0.4KB/s   00:00
etudiant@f2-206-03: ~/.ssh$
```

Côté serveur, on renomme la clé `cle_vm_ubuntu` en `authorized_keys`.

On procède de cette manière car le fichier n'existe pas encore. S'il avait existé, il aurait fallu ajouter la clé à la fin avec un cat

```
cle_vm_ubuntu >> authorized_keys par exemple.
```

## 6) Configurer le serveur pour interdire les connexions par mot de passe

On commence par se connecter en ssh au serveur puis on édite le fichier /etc/ssh/sshdconfig

```
etudiant@f2-206-03:~$ ssh -p 2222 lpasr@127.0.0.1
Welcome to Ubuntu 12.04.3 LTS (GNU/Linux 3.2.0-53-generic-pae i686)

 * Documentation:  https://help.ubuntu.com/

System information as of Fri Oct  7 14:23:11 CEST 2016

System load:  0.0               Processes:            70
Usage of /:   19.7% of 7.12GB    Users logged in:     1
Memory usage: 30%              IP address for eth0: 10.0.2.15
Swap usage:   0%

Graph this data and manage this system at https://landscape.canonical.com/

Last login: Fri Oct  7 14:15:13 2016 from 10.0.2.2
lpasr@ubuntu:~$ sudo su
[sudo] password for lpasr:
root@ubuntu:/home/lpasr# vi /etc/ssh/sshd_config
```

Pour interdire les connexions par mot de passe à tous les utilisateurs, on met PasswordAuthentication à no dans le fichier de configuration global de sshd (ssh serveur).

```
# Change to no to disable tunnelled clear text passwords
PasswordAuthentication no
```

Puis on relance le serveur ssh pour prendre en compte les modifications :

```
root@ubuntu:/home/lpasr# service ssh restart
ssh stop/waiting
ssh start/running, process 4879
```

Si on avait voulu un réglage spécifique pour un utilisateur en particulier, il aurait fallut créer un fichier config dans le dossier .ssh de son homedir. La syntaxe de ce fichier est la même que le fichier de configuration globale.

## 7) Se connecter au serveur avec la clé

On utilise la commande ssh add cle\_vm\_ubuntu dans le dossier .ssh du client pour charger la clé privée.

Si on tente de se connecter en ssh, le mot de passe n'est pas demandé et on est automatiquement connecté au serveur.

Pour vérifier que l'authentification par mot de passe a bien été désactivée, on supprime la clé publique du serveur. Si on tente de se connecter, on a un message d'erreur :

```
etudiant@f2-206-03:~$ ssh -p 2222 lpasr@127.0.0.1
Permission denied (publickey).
etudiant@f2-206-03:~$
```

La connexion par mot de passe a donc bien été désactivée.

## 8) Créer un tunnel SSH sur le service 80 puis tester en surfant depuis le poste client

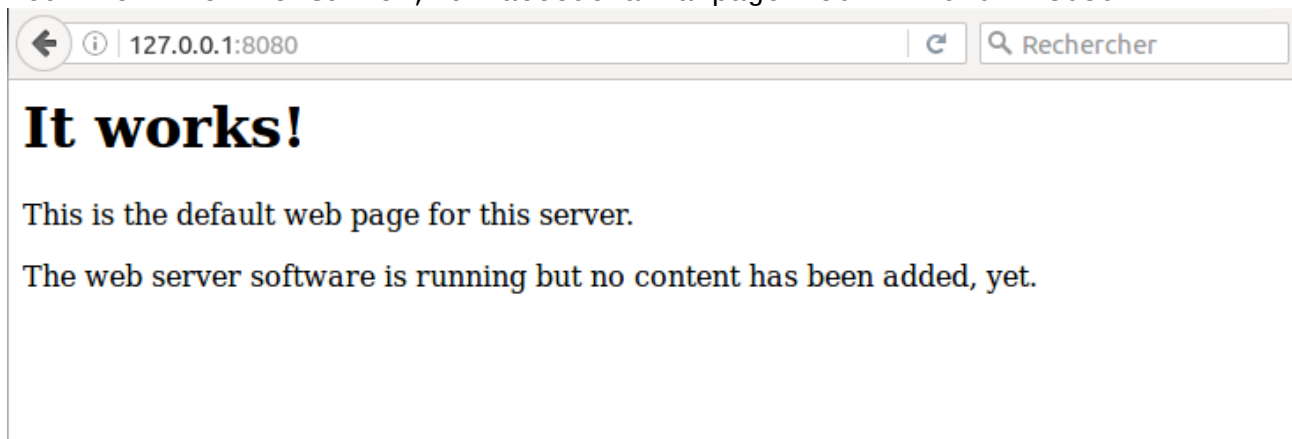
On met en place un tunnel ssh :

```
ssh -L 8080:localhost:80 -p 2222 lpasr@127.0.0.1
```

On fait la redirection du port 8080 vers le port 80 de [lpasr@127.0.0.1](mailto:lpasr@127.0.0.1).

```
ssh -L                                → créer un tunnel
8080:localhost:80 p 2222 lpasr@127.0.0.1 → rediriger le port 8080 vers
le port localhost du routeur, qui le redirigera vers le port 80 de
lpasr@127.0.0.1.
```

Pour vérifier le tunnel, on accède à la page web 127.0.0.1:8080 :



On a donc réussi à configurer notre tunnel.

## 9) Mettre en place fail2ban pour protéger le port 22

On commence par installer le paquet fail2ban via la commande :

```
sudo apt-get install fail2ban
```

(Pour que apt sache utiliser le proxy : ajouter  
Acquire::http::Proxy "<http://proxy2.iutsf.lan:3128>";  
au fichier /etc/apt/apt.conf)

Ensuite, on modifie le fichier de configuration jail.conf de fail2ban  
situé dans /etc/fail2ban.

On peut ajouter bantime = 60 pour ne pas se faire bloquer trop longtemps  
lors des tests.

On rétablit l'authentification par mots de passe sur le serveur ssh et on  
supprime la clé publique. On essaie ensuite de se connecter au serveur  
depuis le client en se trompant plusieurs fois de mot de passe.

Si la commande maxretry est sur 2, on peut lancer la commande ssh deux  
fois. Si on se trompe de mot de passe à tous les essais, on ne peut pas  
relancer de connection pendant 60 seconde (durée du bantime).

Extrait du fichier jail.conf (partie ssh) :

```
[ssh]
enabled = true
port    = ssh
filter  = sshd
logpath = /var/log/auth.log
maxretry = 2
bantime = 60
```

Pour observer ce que fait fail2ban, on peut lancer un tail -f sur son fichier de logs :

```
2016-10-07 16:29:57,063 fail2ban.jail : INFO Creating new jail 'ssh'
2016-10-07 16:29:57,064 fail2ban.jail : INFO Jail 'ssh' uses Gamin
2016-10-07 16:29:57,070 fail2ban.filter : INFO Added logfile = /var/log/auth.log
2016-10-07 16:29:57,070 fail2ban.filter : INFO Set maxRetry = 2
2016-10-07 16:29:57,071 fail2ban.filter : INFO Set findtime = 600
2016-10-07 16:29:57,071 fail2ban.actions: INFO Set banTime = 60
2016-10-07 16:29:57,086 fail2ban.jail : INFO Jail 'ssh' started
2016-10-07 16:29:59,096 fail2ban.actions: WARNING [ssh] Ban 10.0.2.2
2016-10-07 16:30:59,254 fail2ban.actions: WARNING [ssh] Unban 10.0.2.2
2016-10-07 16:31:10,288 fail2ban.actions: WARNING [ssh] Ban 10.0.2.2
2016-10-07 16:32:10,500 fail2ban.actions: WARNING [ssh] Unban 10.0.2.2
```

On voit bien que l'IP du client est bannie puis débannie au bout d'un certain temps.