

OpenSSH

Présentation :

OpenSSH est un ensemble d'outils informatiques libres permettant des communications sécurisées sur un réseau informatique en utilisant le protocole SSH (port 22 par défaut). Il peut être utilisé comme remplaçant direct de rlogin, rsh, rcp, et telnet. De plus, OpenSSH peut sécuriser n'importe quelle connexion TCP/IP via un tunnel. OpenSSH chiffre tout le trafic de façon à déjouer les écoutes réseau, les prises de contrôle de connexion, et aux attaques au niveau du réseau.

L'objectif de ce TP est de créer des clefs publiques et privées depuis un serveur et une autre machine afin de pouvoir se connecter en utilisant une clef privée.

Pré requis :

- Ordinateur sur Linux (ici, on utilisera la Debian 7.9).
- Avoir une connexion internet.
- On utilisera l'utilisateur root et pm
- L'adresse IP de la machine est 192.168.1.32/24 et s'appelle debian8

Sommaire :

- I. Installation des paquets sur la machine Linux.
- II. Configuration principale
- III. Connexion avec Putty par authentification de clef
- IV. Transfert de fichiers

I. Installation des paquets sur la machine Linux.

On met à jour les paquets :

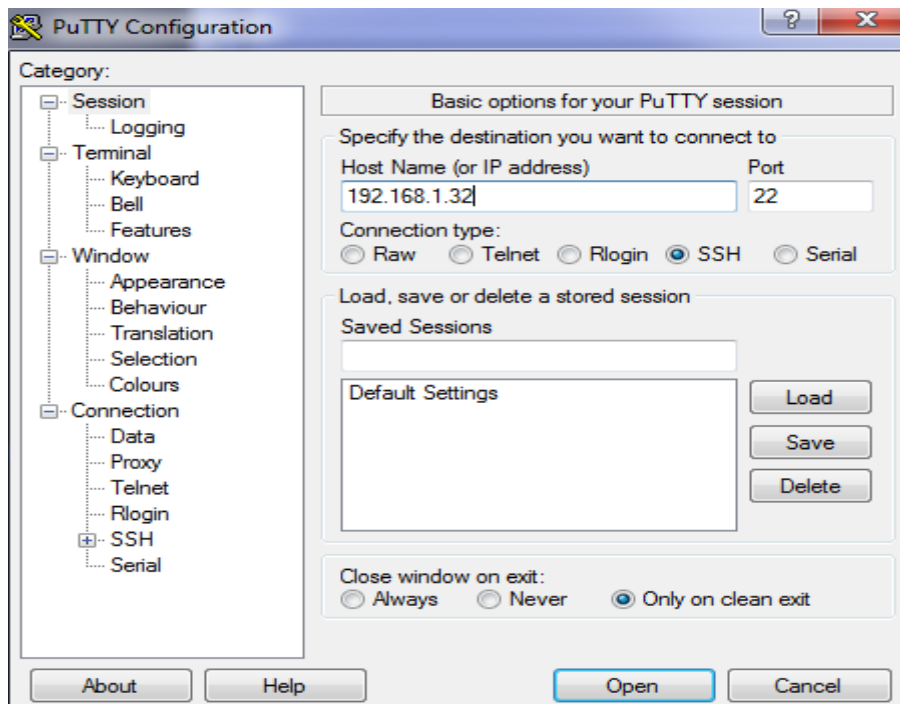
```
root@debian8:/# apt-get update
```

Et on installe openssh (déjà installé normalement) :

```
apt-get install openssh
```

II. Génération des clefs privée et public.

On va utiliser l'utilisateur pm dans la suite du TP. On se connecte en SSH avec un client :



Puis on se log avec pm. On va générer la clef on tape la commande ssh-keygen:

```
pm@debian8:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/pm/.ssh/id_rsa):
Created directory '/home/pm/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/pm/.ssh/id_rsa.
Your public key has been saved in /home/pm/.ssh/id_rsa.pub.
The key fingerprint is:
fe:ab:53:fb:6a:97:4c:48:9d:b9:b0:06:00:1f:8c:35 pm@debian8
The key's randomart image is:
+---[RSA 2048]-----+
|
|  . = E
| ..oo
|  .. . o
|   . o +
|    So + .
|   . = o
|   .o + .
|  ..o +
|   .++=.
|-----+-----+
```

Puis on vérifie et l'on voit bien les deux clefs :

```
pm@debian8:~$ cd .ssh/
pm@debian8:~/.ssh$ ls
id_rsa id_rsa.pub
```

III. Connexion avec Putty par authentification de clef

On va modifier le fichier /etc/ssh/sshd_config :

```
GNU nano 2.2.6      Fichier : /etc/ssh/sshd_config

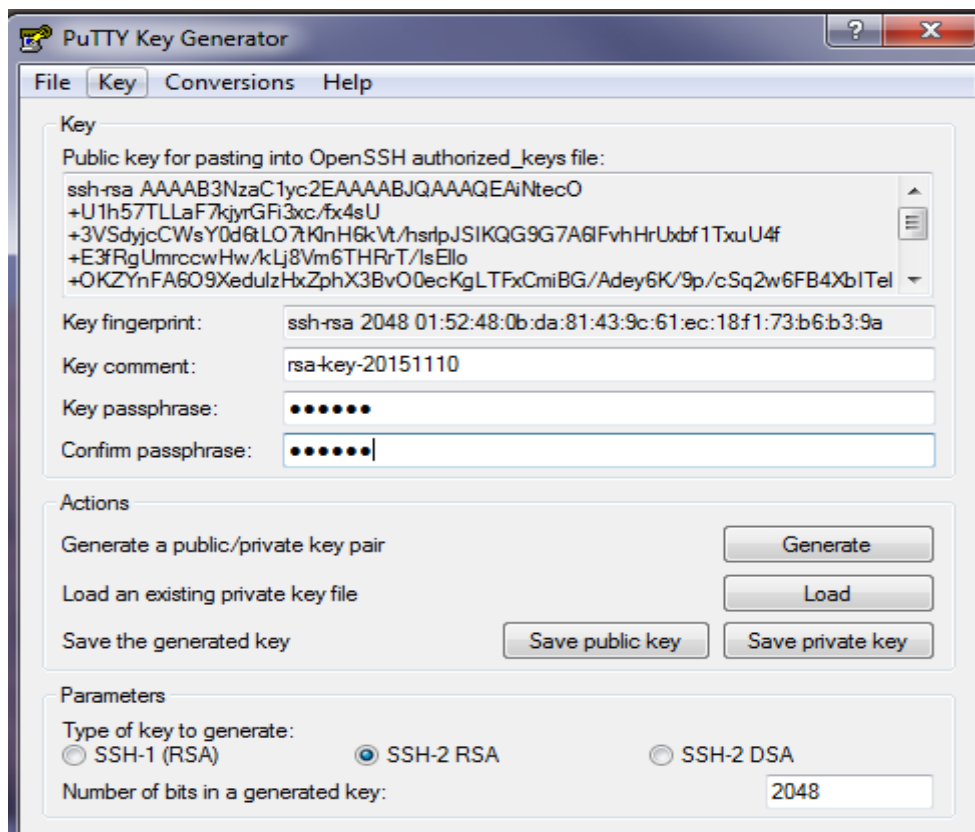
SyslogFacility AUTH
LogLevel INFO

# Authentication:
LoginGraceTime 120
PermitRootLogin yes
StrictModes yes

RSAAuthentication yes
PubkeyAuthentication yes
AuthorizedKeysFile      %h/.ssh/authorized_keys

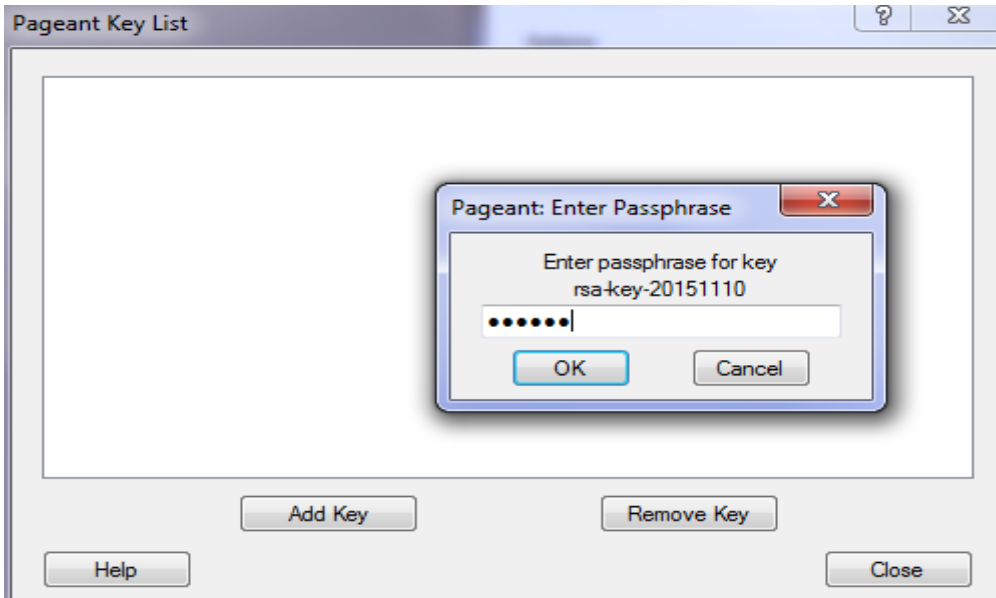
# Don't read the user's ~/.rhosts and ~/.shosts files
IgnoreRhosts yes
# For this to work you will also need host keys in /etc/ssh_known_hosts
RhostsRSAAuthentication no
```

On va générer une clef avec Putty Ley Generator :

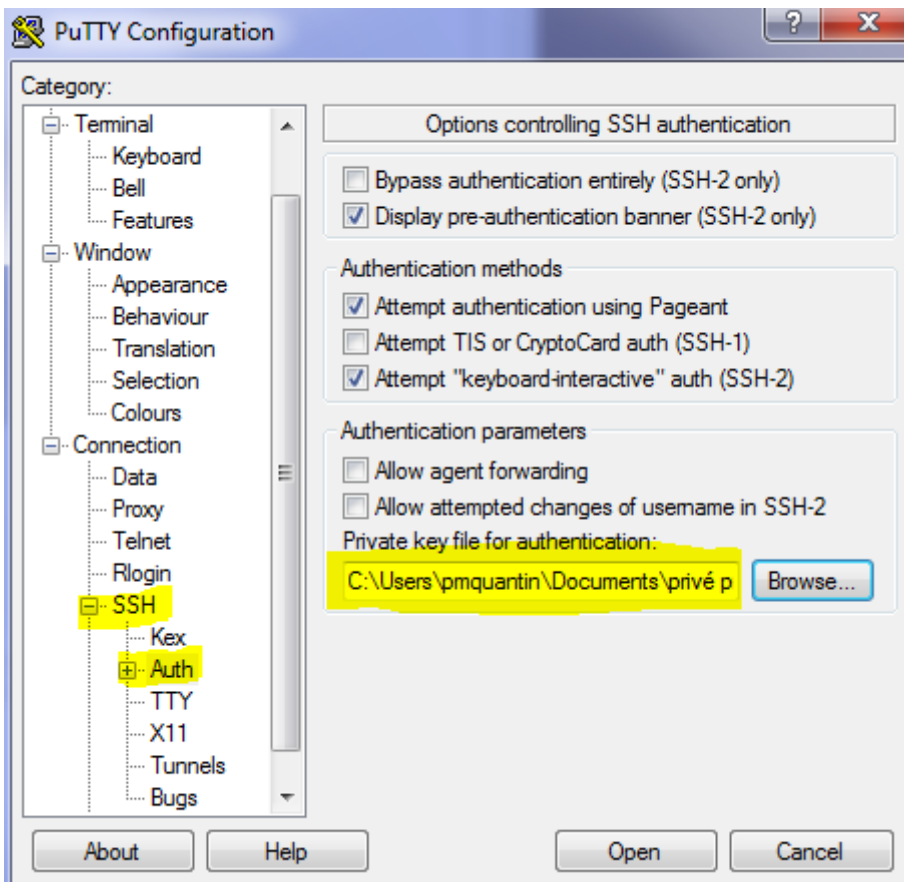


Puis, on sauvegarde la clef publique et privée. Ensuite, la public key, on la copie et par FTP, on la mets dans un fichier dans .ssh nommé authorized_keys.

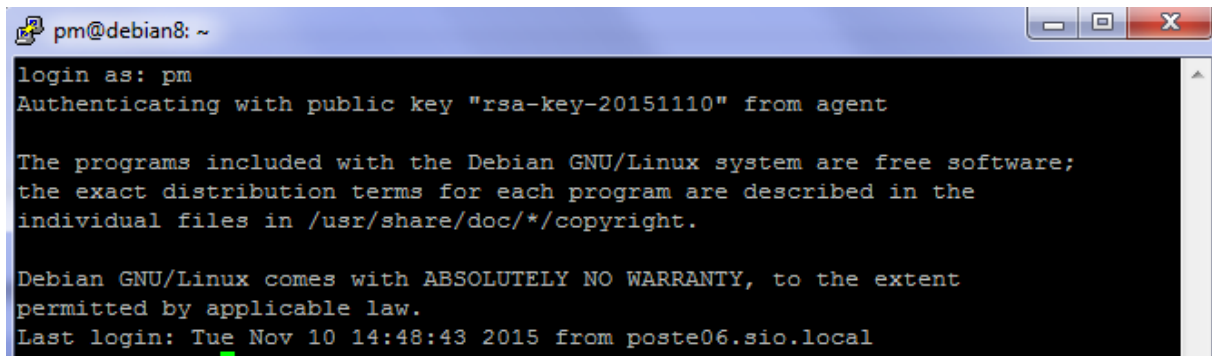
On va utiliser pageant maintenant pour ajouter la clef privée :



Puis, on revient sur putty :



Puis on se connecte avec le login pm :



```
pm@debian8: ~  
login as: pm  
Authenticating with public key "rsa-key-20151110" from agent  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Tue Nov 10 14:48:43 2015 from poste06.sio.local
```

IV. Transfert de fichiers.

De la machine locale à la machine distante :

Scp @nomdufichier @nomdelamachinedistante :

De la machine distante à la machine locale :

Scp @ nomdelamachinedistante : @nomdufichier .