Table des matières :

Avant-Propos

Compétences :

- A1.1.1 Analyse du cahier des charges d'un service à produire
- A1.2.4 Déterminer des tests nécessaires à la validation d'un service
- A3.3.1 Administrer sur site ou à distance des éléments d'un réseau, de serveurs
- A4.1.9 Rédaction d'une documentation technique

Le but de ce TP est l'authentification sous SSH.

Objectif :

Dans cette procédure, nous allons montrer comment installer et configurer une authentification sous SSH.

OS	Distribution
Debian	Linux

1. Le serveur SSH

1. Configuration

Par défaut, le serveur ssh est configuré pour une authentification par mot de passe. Pour changer cela, il faut modifier le fichier /etc/ssh/sshd_config avec l'éditeur nano et rajouter les lignes :

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

Redémarrer le service.

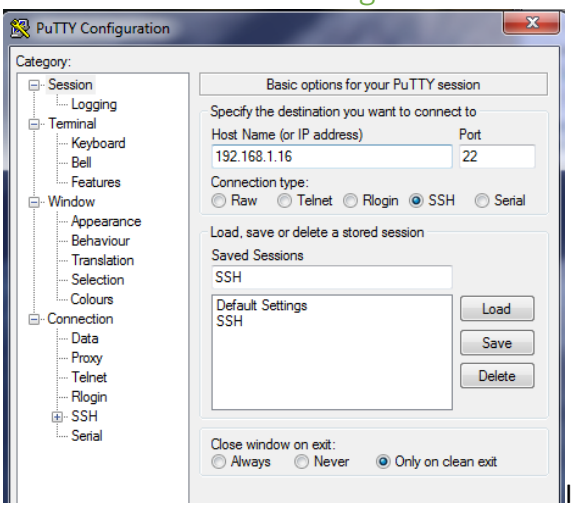
```
root@debian8:~# service ssh restart
```

2. Le client SSH

2. Installation

On utilisera le client ssh pour Windows appelé puTTY. Vous pouvez le trouver très facilement en téléchargement. Télécharger l'installer de préférence au binaire zippé afin de pouvoir bénéficier des utilitaires complémentaires Le logiciel puTTy s'installe ainsi que 3 autres logiciel : puTTYgen et Pageant (puTTY Authentification Agent)

3. Paramétrage de la connexion



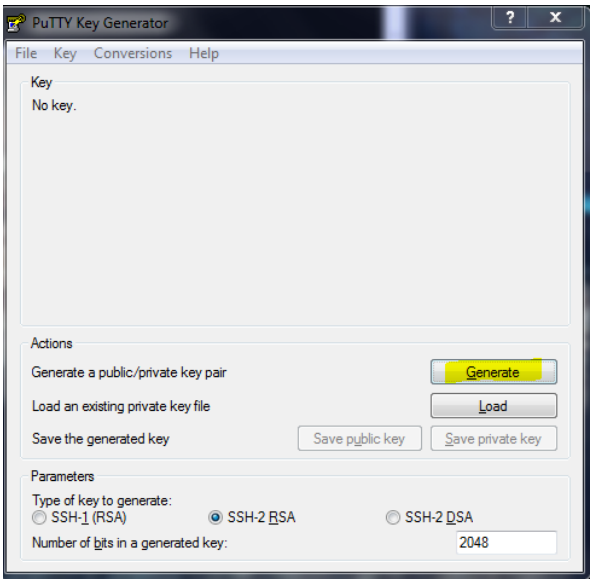
Il suffit d'ajouter une adresse ip de sa machine, on peut effectuer une sauvegarde de l'hôte. On ouvre la connexion, il faudra rentrer les identifiants comme une connexion normale

3. Les clés

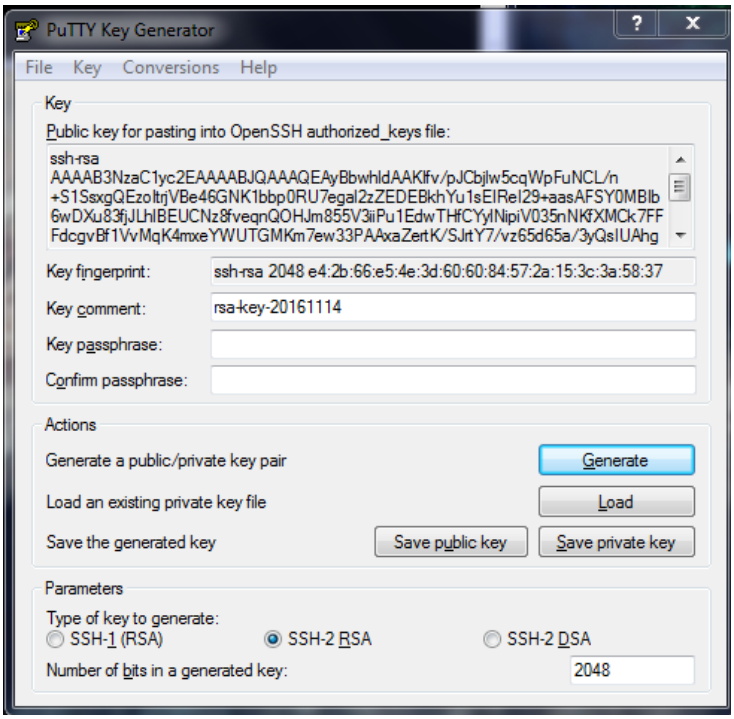
Pour que la connexion s'établisse entre une machine A et une machine B, les deux parties doivent s'échanger leur clé publique. Dès qu'ils sont possession de cette clé, ils l'utilisent pour crypter les données à envoyer et ne peuvent commencer à réellement dialoguer qu'à partir de ce moment. Nous avons vu que, lors du premier contact, le client avait reçu la clé publique du serveur.

4. Génération de la clé

Pour crée cette clé, il faut lancer puTTYgen et faire Generate.



Voici le résultat que vous devez obtenir, lorsque la barre de progression est complète.



On remplit les champs vides par un mot de passe qui permet de protéger l'utilisation de la clé privée.

5. 3.3 Sauvegardé la clé

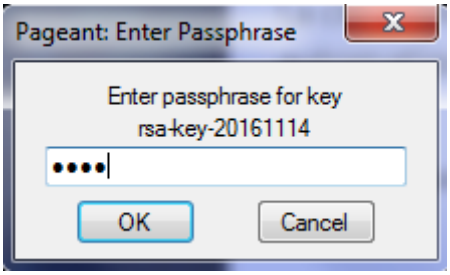
On sauvegarder maintenant de deux façons suivante :

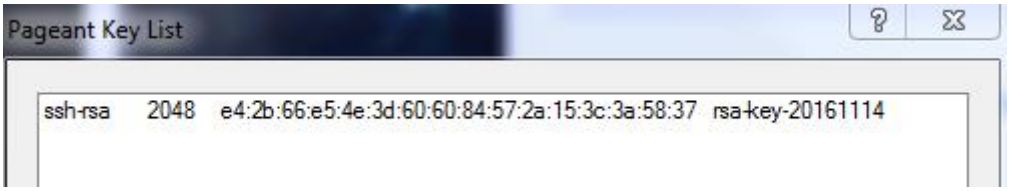
4. L'agent de déverrouillage

La clé privée que nous venons de sauvegarder va nous servir pour nous authentifier automatiquement, pour cela nous avons besoin d'un utilitaire supplémentaire (Pageant)

 en bas de l'écran ds la barre des taches

Pour charger la clé, cliquer sur le bouton « Add key » et sélectionner la clé privée en question (extesion .ppk) Le mot de passe sera demander :



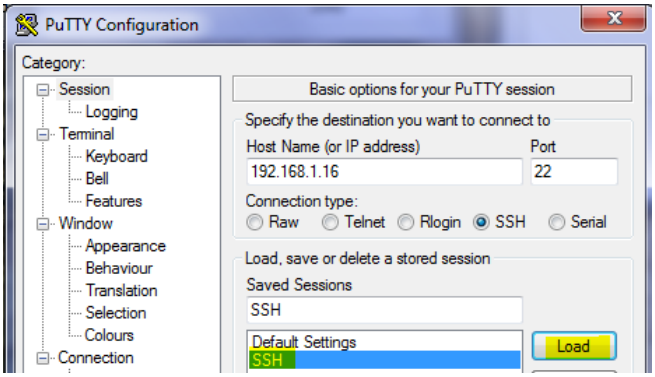


La clé apparait dans la liste.

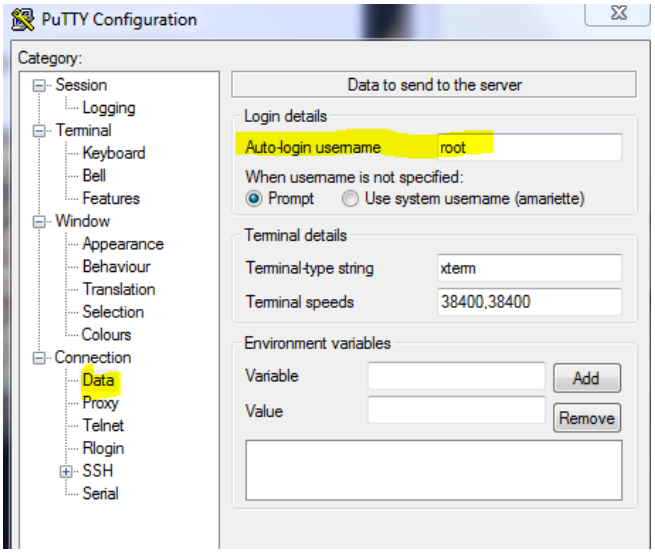
5. Automatiser la connexion

Paramétrage de la connexion :

On démarre putty, on charge la session qu'on a sauvegardée précédemment.

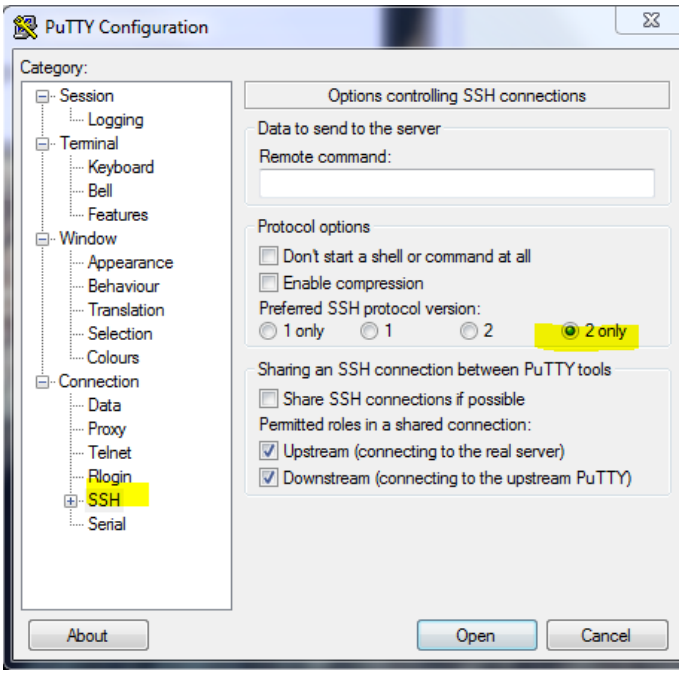


2. Dans le menu, aller sur DATA puis dans Auto-login usename, frapper le nom du compte que vous avez rentré plus haut par la commande useradd pour vous connecter à cet ordinateur.

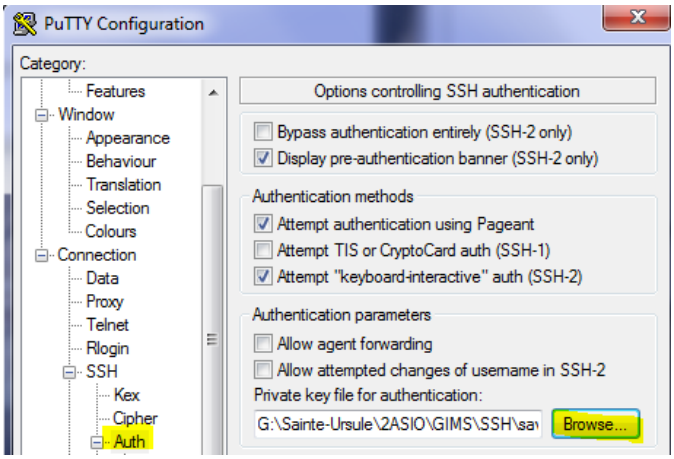


3. Aller ensuite dans le menu SSH.

Ici, il faut sélectionner 2 only dans les boutons radio preferred SSH protocol version.



4. Aller ensuite dans le menu SSH, sur le sous menu Auth. Dans le champ Private Key file for authentification, via le bouton (Browse), charger le fichier de clé privée que vous avez confectionné.



Pour sauvegarder me tout, il faut revenir au menu session et cliquer sur le bouton (save)

Au stade actuel, si nous essayons de nous connecter au serveur Linux, le login sera automatiquement rempli, mais pas le mot de passe. Il reste à informer le serveur de la clé publique que nous avons générée et pas encore utilisée. Cela évite de devoir taper le nom d'utilisateur :

Maintenant que nous avons nos clés et nos sessions, nous allons pouvoir configurer le serveur Linux pour automatiser l'authentification.

Sur votre poste :

Relancer l'utilitaire puTTyGen. Cliquer sur le bouton (load). Sélectionner le fichier correspondant à la clé privée que vous avez générée. Cliquer sur ouvrir. Rentrez le mot de passe la clé privée, ainsi que la clé publique.

On sélectionne le contenu du champ Public key for pasting into openssh authorized_keyss file. Et on le copies.

ssh-rsa
AAAAB3NzaC1yc2EAAAABJQAAAQEAyBbwhldAAKIfv/pJCbjlw5cqWpFuNCL/n+S1SsxgQEzoItrjVBe46G
NK1bbp0RU7egal2zZEDEBkhYu1sEIReI29+aasAFSY0MBIb6wDXu83fjJLhlBEUCNz8fveqnQOHJm855V3ii
Pu1EdwTHfCYylNipiV035nNKfXMCk7FFFdcgvBf1VvMqK4mxeYWUTGMKm7ew33PAAxaZertK/SJrtY7/v
z65d65a/3yQsIUAhgc9qp3NOYGvezpzrX+ANhi/mxYhzPFa4fncXGKefEmszZgpNZc2RBztUor0ljipbvLfAG
KUiTCRQe5Ql6jyi0ca39u0Ugj+MrNjVIGcFZEQ== rsa-key-20161114

On revient ensuite sur putty avec la connexion que nous avons sauvegardée sur le serveur linux.

On crée sur le serveur le répertoire .ssh

```
root@debian8:~# mkdir .ssh
```

On crée ensuite un fichier dans ce dossier sous le nom « authorized_keys », on copie le « champs public key for pasting into openssh authorized keys » que nous avons copié précédemment.

```
root@debian8:~/.ssh# nano authorized_keys
```

On modifie ensuite les droits du répertoire home en (lecture, écriture et exécution par le propriétaire seul.)

```
root@debian8:~# chmod u+rwx,g+---,o+--- /home/
```