I. Objet

Les connexions SSH son très souvent utiliser par les administrateurs réseau. Les pirates informatiques conscient de ce moyen de connexion profite par tout les moyens d'accéder au machine avec de protocole de communication.

Pour évité cela je vais mettre en place un système dit key public/private, personnel à l'utilisateur. Les informations circulant sur le réseau entre la machine cliente et le serveur serons crypté, donc ne pourra être lu par un logiciel de capture de trame ou autre de type Wireshark.

Les connexions via un mot de passe seront refusées, uniquement l'authentification RSA le sera.

II. Domaine d'application

Cette procédure s'applique au service informatique de la société GSB.

III. Documents associés

AUCUN DOCUMENT ASSOCIE

IV. Définitions

- SSH avec authentification par clef

Avec SSH, l'authentification peut se faire sans l'utilisation de mot de passe ou de phrase secrète en utilisant la cryptographie asymétrique. La clé publique est distribuée sur les systèmes sur lesquels on souhaite se connecter. La clé privée, qu'on prendra le soin de protéger par un mot de passe, reste uniquement sur le poste à partir duquel on se connecte. L'utilisation d'un « agent ssh » permet de stocker le mot de passe de la clé privée pendant la durée de la session utilisateur.

Cette configuration profite aussi à SCP et à SFTP qui se connectent au même serveur SSH.

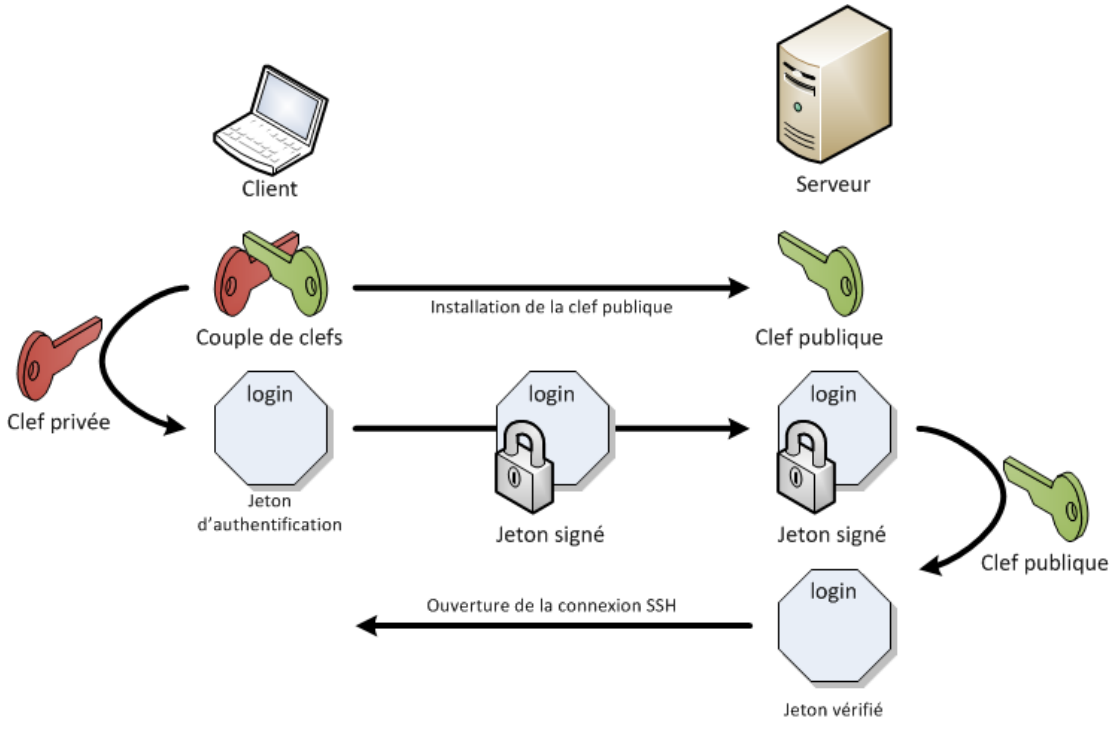
Source : Wikipédia

Introduction :
Le client SSH s'utilise le plus souvent avec un couple utilisateur/mot de passe sur la machine distante.

Toutefois, il est possible d'utiliser une clef RSA (authentification par clef publique) pour s'authentifier sur la machine distante. L'utilisation de cette méthode d'authentification couplée avec une *passphrase* permet de mettre en place un mécanisme d'authentification forte, et donc d'augmenter le niveau de sécurité.

Authentification RSA

Le principe de l'authentification RSA se base sur la signature de jetons d'authentification lors de la connexion. Le client dispose d'une paire de clefs d'authentification : une clef privée qu'il est le seul à détenir et une clef publique que les serveurs utiliseront pour vérifier l'identité du client. Au moment de l'authentification, le client signe un jeton avec sa clef privée et le donne au serveur sur lequel il a besoin de s'authentifier. Le serveur doit avoir accès à la clef publique du client pour décrypter le jeton d'authentification (principe de signature numérique). Le serveur, avec la clef publique, peut donc vérifier la signature du jeton d'authentification, en ayant l'assurance que seul le client a pu générer la signature (le client est le seul à détenir la clef privée).



I. Génération d'une paire de clefs sur Linux

Sur ma machine serveur je génère une paire de clef pour l'utilisateur « root »

```
root@srv-web1-GSB:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):"NOM DE LA CLE"
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/
Your public key has been saved in /root/.ssh/"NOM DE LA CLE".pub.
The key fingerprint is:
44:1b:94:be:04:4a:34:ca:ad:af:16:43:a2:28:b4:8c root@srv-web1-GSB
The key's randomart image is:
+--[ RSA 2048]----+
|   .o  .+.       |
| . o.....o       |
|  o... oo        |
|.....  .o        |
|*oo    .S.       |
|Eoo.    .        |
|.  o.            |
|  ..             |
| ..              |
+-----------------+
```

Par défaut la clef privée est sauvegardée dans le fichier id_rsa dans un répertoire caché nommé .ssh de la homedir de l'utilisateur. La clef publique correspondante est sauvegardée dans le fichier id_rsa.pub dans le même répertoire. Lorsque l'on génère la clef privée, il est possible de protéger l'accès à cette clef par une phrase (appelée passphrase). Cela renforce la sécurité mais oblige à saisir cette phrase à chaque fois que l'on utilise la clef privée, ce qui perd tout intérêt quand l'authentification se fait dans un script.

II. Ajout de la clef publique sur le serveur

Pour permettre l'authentification par clef RSA sur le serveur, il est nécessaire d'ajouter la clef publique de l'utilisateur dans le fichier de clefs autorisées sur le serveur. Pour cela il faut connaitre le mot de passe de l'utilisateur sur le serveur et utiliser la commande suivante :

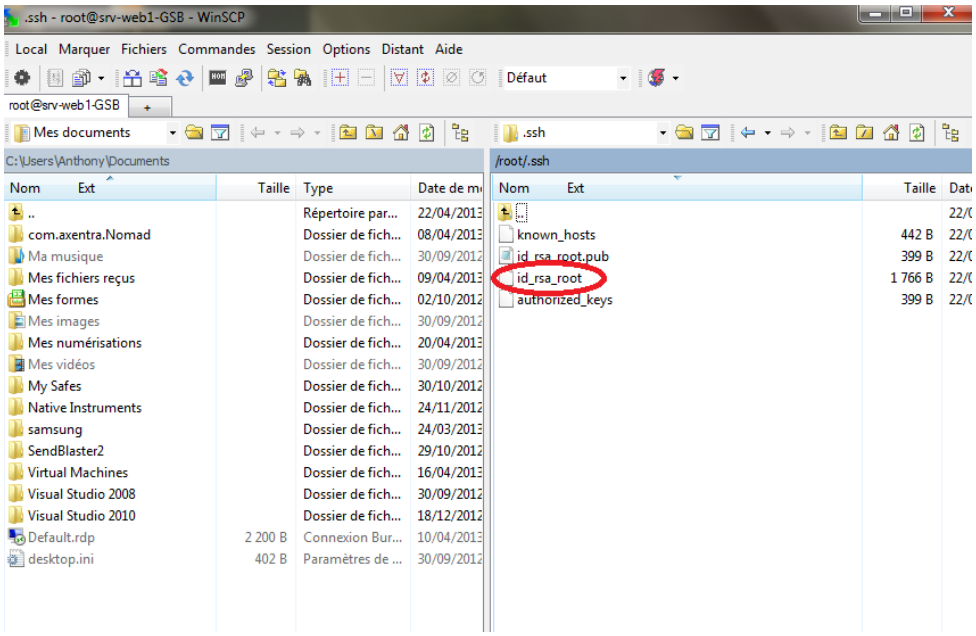
```
ssh-copy-id -i /root/.ssh/"NOM DE LE CLE".pub root@srv-web1-GSB
```

```
Password:
Now try logging into the machine, with "ssh 'root@srv-web1-GSB'", and check in:

 .ssh/authorized_keys

to make sure we haven't added extra keys that you weren't expecting.
```

La clef publique (.pub) à été automatiquement ajouter au fichier « authorized_keys » du serveur.

III. Copie de la clef priver RSA sur la machine client

Après avoir installé le logiciel « WinSCP » sur la machine Windows.
Connecter vous sur la machine serveur, aller dans le répertoire ou et stocker la clef priver (/root/.ssh/ « Nom de la clef »). Pour terminer copier la sur votre machine Windows.



IV. Modification du fichier de configuration sshd

Le service SSH doit également être configuré pour accepter l'authentification par clefs RSA. Le fichier de configuration du serveur SSH doit contenir les paramètres suivants :

```
nano /etc/ssh/sshd.cong
```

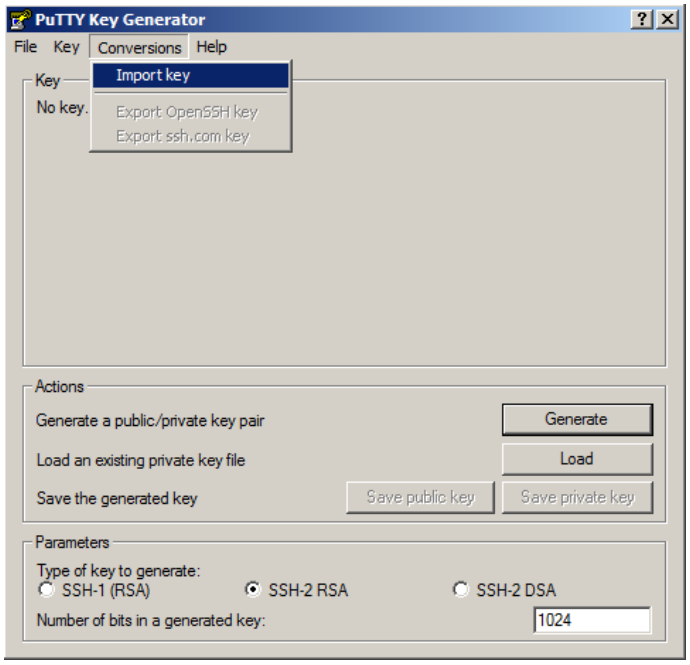
```
RSAAuthentication yes
PubkeyAuthentication yes
AuthorizedKeysFile      $h/.ssh/authorized_keys
PasswordAuthentification no
```

Nous allons forcer l'authentification par clef RSA, en désactivant l'option « PasswordAuthentification no ».
Si vous avez besoin d'ajouter ces lignes de configuration ou de les dé-commenter, n'oubliez pas de redémarrer le service sshd !

```
service ssh restart
```

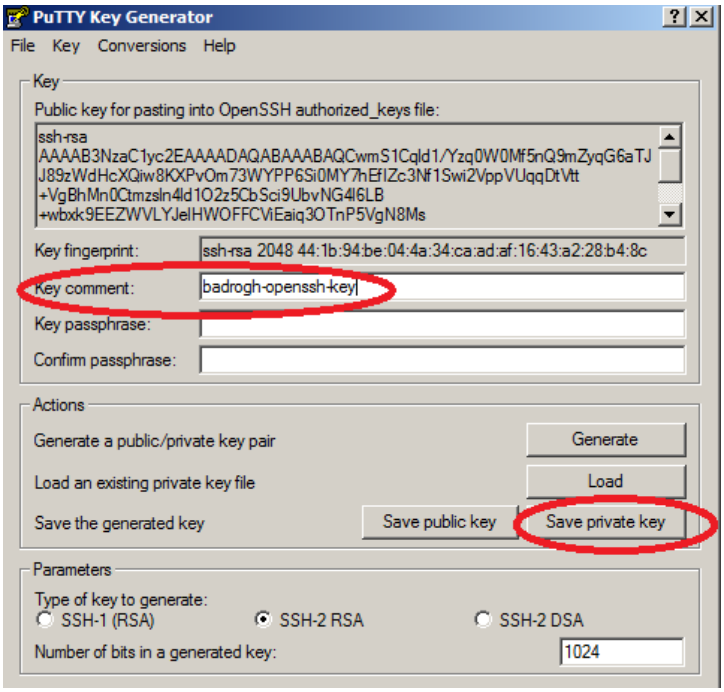
V. Utilisation de clefs RSA avec PuTTy sous Windows

Ensuite nous allons convertir cette clef dans un format utilisable par PuTTy avec l'outil « puttygen.exe », aurait pu être également utilisé pour générer une paire de clefs RSA, dans ce cas nous aurions dû sauvegarder la clef publique générée et ajouter cette clef dans le fichier « autorized_keys » du serveur. Ici nous allons convertir notre clef existante :



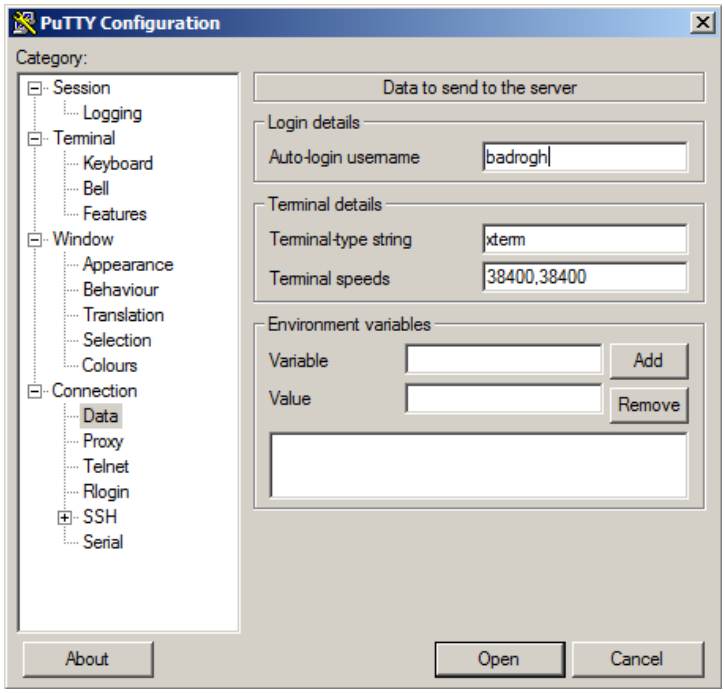
On peut nommer la clef ainsi convertie et la sauvegarder au format ppk utilisé par PuTTy :

Une fois la clef sauvegardée (bouton « Save private key »), on a maintenant une clef utilisable pour nos sessions SSH.
Enregistrer votre key privé dans un emplacement identifiable, pour ne pas le supprimer malencontreusement.

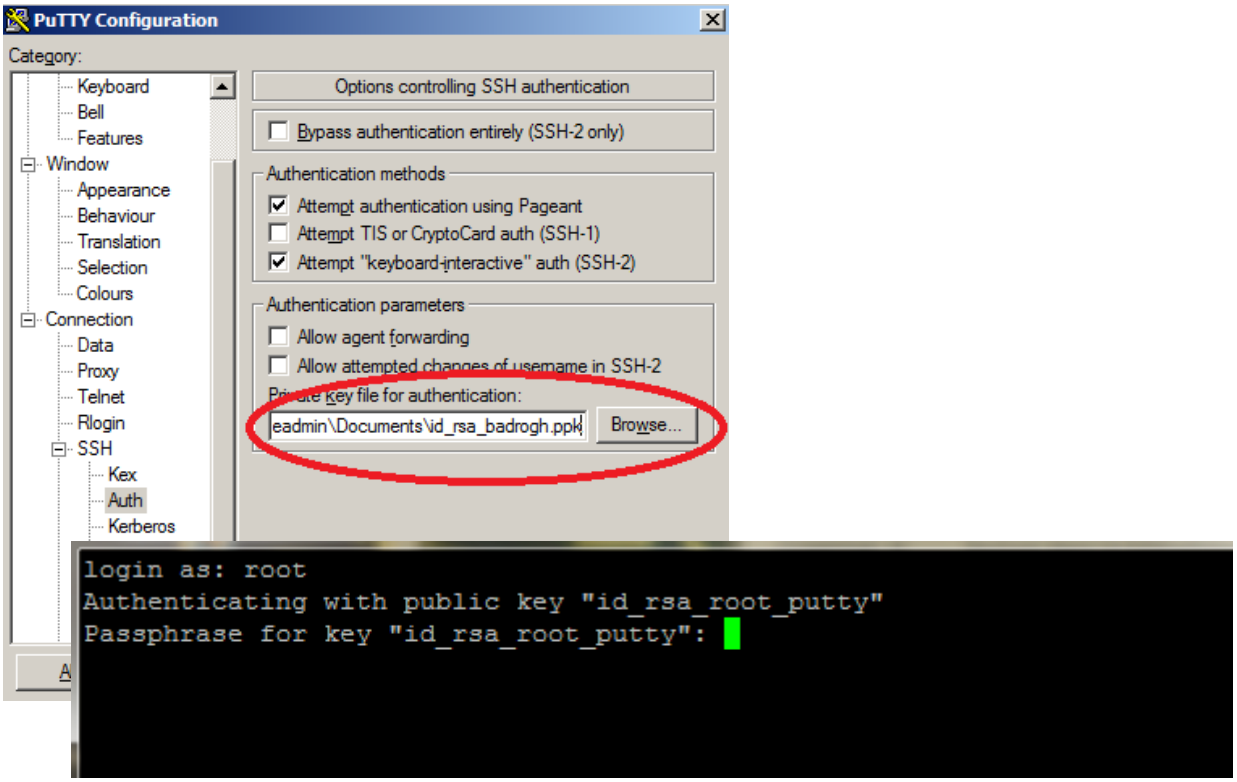


VI. Configuration de la connexion SSH avec PuTTy

Commençons par ouvrir PuTTy puis précisons le nom de login à utiliser pour la connexion SSH dans l'onglet « Connexion » > « Data »:



Puis ajoutons la clef privée à utiliser pour la connexion SSH dans l'onglet « Connection » > « SSH » > « Auth » :



www.anthony-cochet.fr