Utilisation de clés SSH

L'utilisation de clés SSH permet de s'affranchir de la saisie de mot de passe pour se connecter aux systèmes distants. Cela peut éviter des risques de sécurité comme taper son mot de passe au mauvais moment (et le faire apparaître en clair sur son terminal), ou au mauvais endroit, etc.

On va voir également comment protéger sa clé SSH privée par une phrase de sécurité (*passphrase*). Cela ajoute une pretite contrainte (la saisie de la *passphrase*), mais couplé avec l'agent SSH, cela apporte un niveau de sécurité supplémentaire.

1 - Principe

Le principe repose sur l'utilisation d'une paire de clés asymétriques. La clé publique peut être installée sur les machines distantes. La clé privée est la base de cette sécurité et doit être conservée ***en lieu sûr*** sur son poste de travail.

1.1 - Création d'une paire de clé sous Linux

Par défaut, c'est une clé RSA 2048bits qui est générée. C'est aujourd'hui le minimum à utiliser. Dans l'exemple ci-dessous, on ne renseigne pas de passphrase.

```
herve@ubuntu:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/herve/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/herve/.ssh/id_rsa.
Your public key has been saved in /home/herve/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:OsosLK1jGdPJkkbieW/Q3rZOb6KbW0erFQa/y1UfNEc herve@ubuntu
The key's randomart image is:
+---[RSA 2048]----+
|                E|
|               . |
|       .       o.|
|..      o     . o|
|+ = o   S=   . . |
| O * . .o + . . .|
|.o* + +o = .   . |
|oo+o =+=*.o      |
|o+ .==*+++       |
+----[SHA256]-----+
herve@ubuntu:~$ ls -al .ssh/
total 16
drwx------  2 herve rv 4096 févr.  1 09:59 .
drwxr-xr-x 22 herve rv 4096 sept. 23 12:42 ..
```

```
-rw-------  1 herve rv 1671 févr.  1 09:59 id_rsa
-rw-r--r--  1 herve rv  396 févr.  1 09:59 id_rsa.pub
```

Il reste à copier la clé publique sur les systèmes distants.

```
herve@ubuntu:~$ ssh-copy-id -i ~/.ssh/id_rsa pinvidic-h@remote
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed:
"/home/herve/.ssh/id_rsa.pub"
The authenticity of host 'remote (X.X.X.X)' can't be established.
ECDSA key fingerprint is SHA256:FltlHYWTOguqFRXDau57pTT+rXkCaZZjDkyvg1Prm4Q.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to
filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are
prompted now it is to install the new keys
pinvidic-h@remote's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'pinvidic-h@remote'"
and check to make sure that only the key(s) you wanted were added.
```

La connexion se fait sans demande de mot de passe:

```
herve@ubuntu:~$ ssh pinvidic-h@remote
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-112-generic x86_64)

pinvidic-h@remote:~$
```

1.2 - Création d'une paire de clé sous Windows

Nous allons prendre exemple sur la suite d'outils SSH Putty (client SSH putty ou kitty portable, puttygen pagent).

Les clés SSH peuvent être générées à l'aide de puttygen. Elles seront dans un format utilisable par Putty ou Kitty Portable. L'interface propose de mettre une *passphrase*. On peut choisir la longueur de la clé (par exmple 2048, 3072 ou 4096).

Pour protéger la clé privée, on définit une *passphrase*. On sauvegarde la clé publique et la clé privée (en lieu sûr).

On peut installer la clé publique dans les clés SSH autorisées sur les systèmes distants. Pour cela, on transfère la clé public sur le serveur (linux) et on convertit la clé Putty au format SSH-RSA :

```
ssh-keygen -i -f putty1.ppk.pub > putty1.rsa.pub
```

```
cat putty1.rsa.pub >> ~/.ssh/authorized_keys
```

2 - Renforcer les accès aux systèmes Linux

Configurer le système “sans” compte root :

1. passer à une base de comptes LDAP (en ldaps)
2. autoriser seulement un groupe LDAP à se connecter
3. configurer SUDO pour ce groupe
4. Renforcer la configuration SSH

Modifier /etc/ssh/sshd_config :

```
# Désactiver l'accès SSH par mot de passe
ChallengeResponseAuthentication no
PasswordAuthentication no
UsePAM no

# interdire les connexions distantes à root :
PermitRootLogin no

# ou n'autoriser le login root uniquement par clé depuis un reseau :
PermitRootLogin prohibit-password
AllowUsers root[at]172.20.13.*

# restreindre les accès au groupe 'admins'
AllowGroups admins
```

2.2 - Passphrase et agent SSH

Si l'utilisation de clés SSH permet d'augmenter la sécurité en désactivant sur les systèmes distants les connexions avec mot de passe, il persiste le risque de se faire pirater sa clé privée.

L'utilisation de *passphrase* (ou phrase secrète) permet de protéger sa clé privée. Couplé avec le système d'agent SSH, cela apportera la sécurité optimale du système de clés SSH.

L'idée est donc d'ajouter une protection à sa clé privée qui demandera à taper la *passphrase* pour l'utiliser. L'agent SSH permettra d'enregistrer en mémoire la *passphare* et la fournira au programme SSH lors des futures utilisations de la clé. Cela permet de taper une seule fois la *passphrase* pendant sa session.

Ajouter/modifier une passphrase à sa clé privée

Note: cela aurait pû être fait lors de la création de la clé SSH.

Illustration en ligne de commande:

```
herve@ubuntu:~$ ssh-keygen -p -f ~/.ssh/id_rsa
Enter new passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved with the new passphrase.
```

A la première utilisation de sa clé privée (ou pour chaque utilisation si l'agent SSH n'est pas lancé), le programme SSH demandera la passphrase :

```
herve@ubuntu:~$ ssh pinvidic-h@remote
Enter passphrase for key '/home/herve/.ssh/id_rsa':
```

Si l'agent SSH n'est pas lancé, on peut l'activer et ajouter sa passphrase :

```
herve@ubuntu:~$ ssh-agent
SSH_AUTH_SOCK=/tmp/ssh-sDvOD3tGJZNK/agent.15834; export SSH_AUTH_SOCK;
SSH_AGENT_PID=15835; export SSH_AGENT_PID;
echo Agent pid 15835;

herve@ubuntu:~$ ssh-add
Enter passphrase for /home/herve/.ssh/id_rsa:
Identity added: /home/herve/.ssh/id_rsa (/home/herve/.ssh/id_rsa)
```

A partir de maintenant, les connexion utilisant la clé privée ne demanderont plus la passphrase :

```
herve@ubuntu:~$ ssh pinvidic-h@remote
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-112-generic x86_64)

pinvidic-h@remote:~$
```

Illustration avec l'agent SSH gnome:

L'agent SSH GNOME demande la passphrase et la stocke temporairement pour les utilisations dans la session. Il n'est pas recommandé de cocher *Déverrouiller automatiquement cette clé quand je suis connecté*.

Pour effacer les passphrases chargées dans l'agent SSH , faire :

```
ssh-add -D
```

Illustration avec putty agent:

On lance le programme pageant.exe

On charge la clé générée par puttygen et on renseigne la *passphrase*

On se connecte avec putty ou kitty sur le serveur distant sans mot de passe ou passphrase grâce à l'agent :

2.3 - Limiter les accès SSH par IP avec des clés

Il peut être judicieux de limiter la provenance des connexions SSH utilisant des clés à certaines IP. Cela peut être le cas pour l'utilisation de script d'automatisation depuis une machine centrale à partir de laquelle les configuration sont poussées via SSH vers les machines distantes. Ou bien pour un serveur de sauvegarde qui utilise SSH pour communiquer avec ces clients (backuppc).

Le fichier ~/.ssh/authorized_keys peut contenir des lignes de la forme : <option> <format de clé> <clé publique SSH> <commentaire>

On peut avoir par exemple :

```
ssh-rsa AAAAB3Nza...== pinvidic-h@pcbureau 20171225
from="backuppc.univ-nantes.prive" ssh-rsa AAAAB3Nza...==
backuppc@backuppc.univ-nantes.prive
from="172.20.1.*" ssh-rsa AAAAB3Nza...== user1
```

Attention : il ne faut pas introduire de risque d'intrusion avec les “jockers” (*, ?, etc) dans les noms de machines ou IP: donner l'accès à backuppc*.univ-nantes.prive n'est pas la même chose que backuppc*.u12.univ-nantes.prive. Dans ce cas précis, la DSIN gère le DNS de u12.univ-nantes.prive et l'enregistrement des machines avec ce FQDN est sous son contrôle. Avec backuppc*.univ-nantes.prive, on ouvre des accès également pour des machines comme backuppc.<sous-domaine>.univ-nantes.prive.

Dans le *pattern* from=“...”, les noms de machines ne peuvent pas être des alias DNS.

Quelques articles à lire

- https://delicious-insights.com/fr/articles/comprendre-et-maitriser-les-cles-ssh/
- http://www.linux-france.org/prj/edu/archinet/systeme/ch13.html
- https://the.earth.li/~sgtatham/putty/0.70/htmldoc/Chapter8.html#pubkey
- https://openclassrooms.com/courses/reprenez-le-controle-a-l-aide-de-linux/la-connexion-securisee-a-distance-avec-ssh
- https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.2.0/com.ibm.zos.v2r2.foto100/authkeyf.htm