TP SSH

Contents

1. Introduction

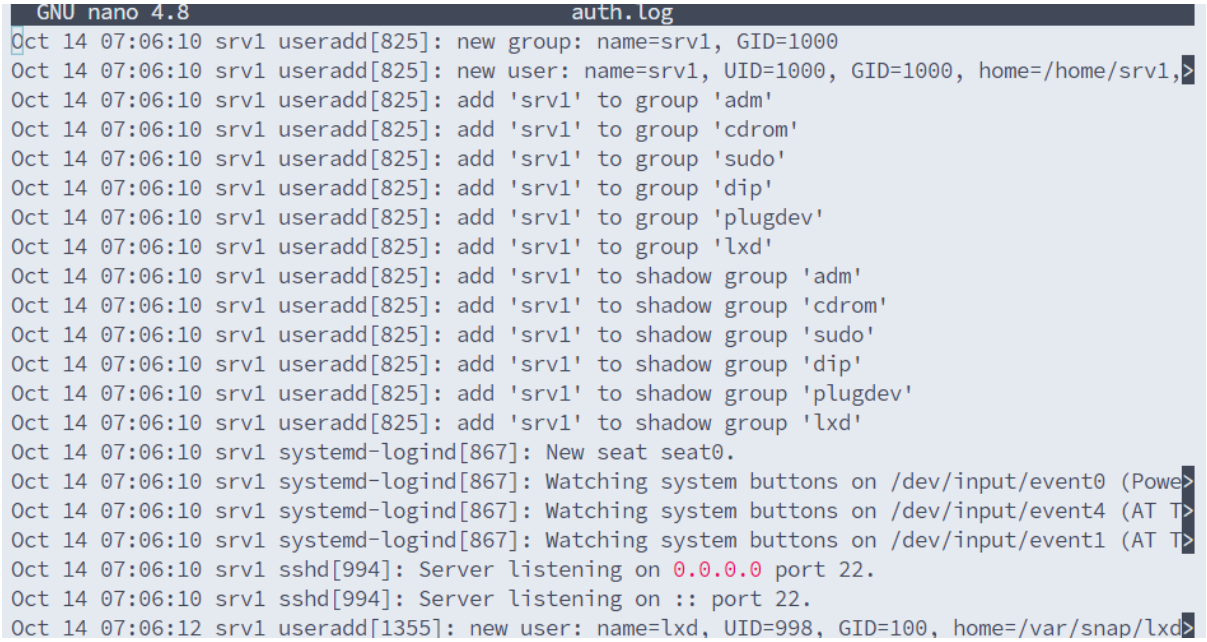
SSH : Secure Shell

Connexion sécurisée entre un client et un serveur.

Version libre : OpenSSH

SSH doit être sécurisé :

- Mise à jour
- Mots de passe complexes
- Surveiller régulièrement les connexions dans /var/log/auth.log



Test de mots de passe avec John :

Sudo apt install john

```
jack@raspberrypi:~ $ sudo apt install john
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets suivants ont été installés automatiquement et ne sont plus nécessaires :
  chromium-browser chromium-browser-l10n chromium-codecs-ffmpeg-extra libcamera0.1
  libraspberrypi0 libssl1.1 libwpe-1.0-1 libwpebackend-fdo-1.0-1
  linux-headers-6.1.0-rpi7-common-rpi linux-headers-6.1.0-rpi7-rpi-2712
  linux-headers-6.1.0-rpi7-rpi-v8 linux-headers-6.1.0-rpi8-common-rpi
  linux-headers-6.1.0-rpi8-rpi-2712 linux-headers-6.1.0-rpi8-rpi-v8
  linux-headers-6.6.31+rpt-common-rpi linux-headers-6.6.31+rpt-rpi-2712
  linux-headers-6.6.31+rpt-rpi-v8 linux-image-6.1.0-rpi7-rpi-2712
  linux-image-6.1.0-rpi7-rpi-v8 linux-image-6.1.0-rpi8-rpi-2712
  linux-image-6.1.0-rpi8-rpi-v8 linux-image-6.6.31+rpt-rpi-2712
  linux-image-6.6.31+rpt-rpi-v8 linux-kbuild-6.1 linux-kbuild-6.6.31+rpt
```

John /etc/shadow

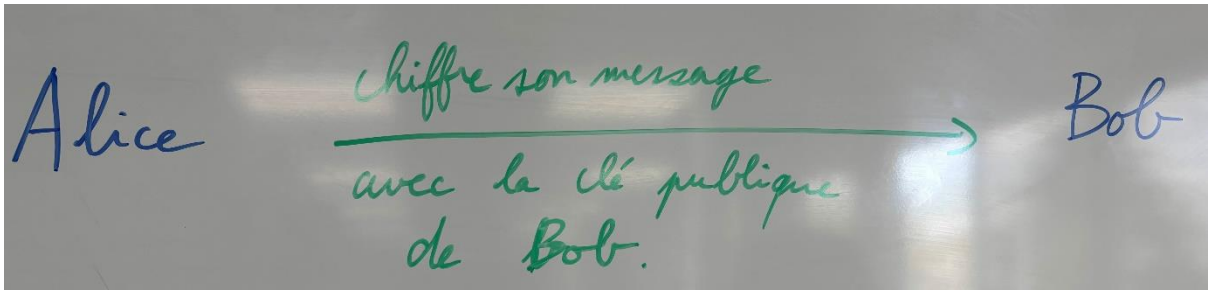
```
jack@raspberrypi:~ $ sudo john /etc/shadow
No password hashes loaded (see FAQ)
jack@raspberrypi:~ $
```

2. Le système de clés

- Cryptographie asymétrique :

  Chaque personne dispose d'un couple de clés : publique et privée.

  La connaissance de la clé publique ne permet pas d'en déduire la clé privée.



- Cryptographie synétique :

  Bob et Alice ont tous les deux la même clé secrète.
  Cette méthode est beaucoup moins gourmande en ressource mais le problème est l'échange de la clé. Dans SSH, les 2 méthodes sont utilisées : d'abord la cryptographie asymétrique pour échanger la clé secrète puis la cryptographie symétrique pour le reste de la conversation.

  Un couple de la clés RSA, généré a l'installation du serveur, est stocké dans le dossier /etc/ssh

  - Clé privée : ssh_host_rsa_key 600
  - Clé publique : ssh_host_rsa_key.pub 644

Etapes :

1. Le serveur envoie sa clé publique au client.
2. Client génère une clé secrète et l'envoie au serveur en la cryptant avec la clé publiques (asymétrique). Le serveur déchiffre cette clé avec sa clé privée. Privée (ce qui prouve qu'il est le bon serveur).
3. Le serveur crypte un message standard avec sa clé secrète et l'envoie au client. Si le client déchiffre le message standard avec secrète, il a la preuve que c'est le bon serveur
4. Le canal est établi entre le client et le serveur (symétrique).
5. Echange du login et mot de passe utilisateur

3. Installation et configuration

Sudo apt install openssh-server

Fichier de configuration: /etc/ssh/ssh_config

Port : 22

PermitRootLogin : Permissions sont déconseillée

X11 Forwarding : Transmission graphique

Démarrage du serveur : systemctl start sshd

4. Connexion

Commande de connexion : ssh user@machine

Ou sont stockée la clé publique du serveur : ~/.ssh/known_hosts

```
srv1@srv1:~$ exit
logout
Connection to 192.168.0.149 closed.
jack@raspberrypi:~ $ cd .ssh
jack@raspberrypi:~/.ssh $ ls
known_hosts  known_hosts.old
jack@raspberrypi:~/.ssh $ sudo cat known_hosts
|1|NayaBfrNjn5zvYlQn3LJ/3D/oA8=|rzp39kfYG8KNudV0TPJD1wWsI9I= ssh-ed25519 AAAAC3NzaC1lZDI1NTE5
AAAAIPW8nN+U8BdE9TURAWzhBfcpSN/vpjsdZDQsd07Zze3w
|1|umiZZxCVInNX04At57VqjInozLw=|V2QnO2bw+ZatdoaovyYsVw6sj4c= ssh-ed25519 AAAAC3NzaC1lZDI1NTE5
AAAAIMaA+l+JI6B7gbPNsh0LMGfsJkkZee8XpA2FJxF3udh4
|1|0D4imMrFThxC3Qjdj3zHLXE5tUo=|hGMH4z3abQ5I197DbEL9c8zIT6g= ssh-rsa AAAAB3NzaC1yc2EAAAADAQAB
AAABgQCRKFqH69Njp/fXQOhXQNHCp1JmWDx5Z0pkZys6IGTj22L0m0YdunJrhhRPbGCJl/+hIlJzC3Wj1xj/ZSYJm39/5
H4t7c8yU+nLN1T0QBbh1/Ru4TWpOprUTKtJ4dIFeS6EFz49Yho7KwyEUjCnp7QQ0VRryel2Cl562CXrYf2oM1FN8Bg0Z6
TjH4rn6nf6MzByZOi5VwkXwV8ss6hBvqPw7hK8BC68LaY5+h53wICsQAJk4XY7EjgmiFaERguAgU/i4p29/2c0ZqttGPZ
7ICXjZ3YVx0jWWfd4rtu8Ybh7btsADjD1zpYLnvq7Xs5iPds1hmmOMGNtl6SU45LOipsIsZFBKK45BEEUUqZ1YfSbFv6V
p86YCP3dDRcdD+zZ75g0mimHMNfU9/QU8R3dZ+AmGTtU7IxJRycxQU9ufkmn0wCUsRw68VyHNy8Zh2JCdWe52rNQXxEzG
QdHGS6evlpaSZYQib4GrQIJ4KREl+4m0N0WZYj6xXohL1pF02es3nk=
|1|v79S5zhd1u7ebCw3YrDjkNPWy8A=|MRLj9AkTsrJ+rnc/SHln4GvMk1A= ecdsa-sha2-nistp256 AAAAE2VjZHNh
LXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBM33nfcGkteCZg7qlvWEshjx1kdfjPc2O10v7cDs5KxU8FWo4Te/F
5EIKiaPh8KzWpOVUPvKAT1WaTwhNJVgXew=
```

Authentification par clé au lieu de login/password :

- Générer un couple de clés DSA :

  Ssh-keygen -t dsa

  ➔ ~/.ssh/id_dsa 600
  ➔ ~/.ssh/id_dsa.pub 644

```
jack@raspberrypi:~/.ssh $ ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/home/jack/.ssh/id_dsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/jack/.ssh/id_dsa
Your public key has been saved in /home/jack/.ssh/id_dsa.pub
The key fingerprint is:
SHA256:QmWW86hVR+JRi0SGzvEWGlw7epeCBf9VU2O16r1GEzk jack@raspberrypi
The key's randomart image is:
+---[DSA 1024]----+
|       .==Boo  +*|
|       +B*o=....+|
|      .o XBo. ...|
|     .  *++o o.E |
|      .oS.o +.  o|
|      .. . o. .o |
|            ....|
|              ..|
|              .. |
+----[SHA256]-----+
```

- Autoriser vos clés : copier votre clé publique dans ~/.ssh/authorized_keys

```
jack@raspberrypi:~/.ssh $ cp id_dsa authorize
jack@raspberrypi:~/.ssh $ ls
authorize  id_dsa  id_dsa.pub  known_hosts  known_hosts.old
jack@raspberrypi:~/.ssh $ cat authorize
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABswAAAAdzc2gtZH
NzAAAAgQD0uVdneluJY3wNXaLiL1S5Qq+nJvvN/m1srvRhEB52aOh5dCYxoh9K2L9AZMPO
y3AXqR9tzkfJTwwIWl6C3L8pK3ia/uRPWD6qkyeMLQqoZHEOYHmDrDGfA24JYv22lBHh7z
GQ76M2favTWn150P91a9R4nqPkQedBKlqJdlCZYwAAABUAuE4QVd8uhsYVvHDMsq0zHyE6
kWkAAACBAOiaGQGMlJrsxUuL2BtltQu7LsBbMxWYMs10Cs4fKYbthDvOc+uzXYWUe9JdRu
08UeF24fz/T0I9kikwdjHiV2c/cI/+yIE0ZkM6pFXSnWU0qs5YojZ2JmWK+CXSr3MhS+Vz
```

5. Transfert de fichiers

Commande scp

Pour transférer test 1.txt

Scp test1.txt toto@ip:

```
srv1@srv1:~$ ls
test1.txt
srv1@srv1:~$ scp test1.txt jack@192.168.0.201
srv1@srv1:~$
```

Pour télécharger test2.test

Scp toto@ip:test2.txt

```
srv1@srv1:~$ scp jack@192.168.0.201:test2.txt .
jack@192.168.0.201's password:
test2.txt                                     100%    0     0.0KB/s   00:00
srv1@srv1:~$ ls
jack@192.168.0.201  test1.txt  test2.txt
srv1@srv1:~$
```

6. Se connecter sans mot de passe

On utilise le couple de clés publique privé mais on ne veut pas taper la passphrase a chaque fois. On utilise le ssh-agent pour la garder en mémoire.

Eval $(ssh-agent)

Ssh-add

```
jack@raspberrypi:~ $ eval $(ssh-agent)
Agent pid 3692
jack@raspberrypi:~ $ ssh-add
Identity added: /home/jack/.ssh/id_dsa (jack@raspberrypi)
jack@raspberrypi:~ $
```

7. Tunnel SSH

Pour chiffrer n'importe quelle communication TCP entre 2 machines.

Exemple pour une connexion HTTP :

- Creation du tunnel :
  Ssh -L 2024 : ip_server :80

```
jack@raspberrypi:~ $ sudo ssh -L 2024:192.168.0.149:80 srv1@192.168.0.149
srv1@192.168.0.149's password:
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.4.0-196-generic x86_64)
```