SSH

Table des matières

Introduction au ssh

1) Qu'est-ce que c'est ?

SSH = secure Shell
Protocole qui permet des communications chiffrées entre un serveur est un client.
Il remplace les anciens protocoles telnet, rlogin, rsh.
Version 1 en 1995, aujourd'hui obsolète.
Version 2 en 2006

2) Mise en garde sécurité

SSH permet à un utilisateur d'accéder à un système à distance avec son identifiant.
Est son mot de passe.
Trois contraintes majeures de sécurité :
-avoir un serveur SSH a jour
-les mot de passe de l'utilisateur doivent être suffisamment complexe
-surveiller régulièrement les connexions dans les logs.

3) cryptographie

SSH utilise la cryptographie asymétrique et symétrique.
Cryptographie asymétrique (RSA de préférence >DSA)

Chaque personne dispose d'un couple de clef :
Une clef publique est une clef priver. La première peut être librement publiée
Alors que la deuxième doit rester secrète. La connaissance de la clef publique
Ne permet pas de connaitre la clef privée

Si Alice veut envoyer un message privé à BOB elle doit le chiffre avec la clef publique
De bob. Seul bob pourra déchiffre le message avec sa clef privé

Cryptographie symétrique :
Si Alice et bob veulent s'envoyer un message confidentiel, ils doivent d'abord
Posséder tous les deux la même clef secrète la même clef secrète
Les algorithmes de cryptage symétrique sont beaucoup moins gourmands en ressource
Processeur que ceux du cryptage asymétrique.

Mais leur problème est l'échange de la clef secrète
Dans le protocole SSL utilise par SSH est les navigateur web, le cryptage
Asymétriques est utilisé au début pour échanger la clef secrète de manière sécuriser

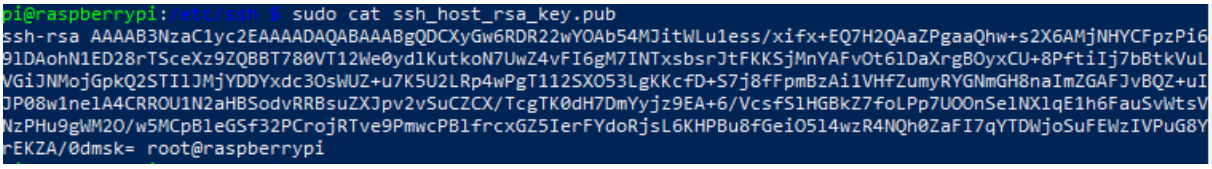
4) établissement d'une connexion SSH

Un serveur SSH possède un couple de clef attachée dans le dossier /etc/ssh

ssh_host_rsa_key contient la clef privée

Avec les permissions 600

ssh_host_rsa_key.pwb contient la clef publique avec les permissions 644.



Etapes :

1)le serveur envoie sa clef publique au client qui vérifie sil la déjà reçu

2)le client génère une clef secrète est l'envoie au serveur en la cryptant avec la clef publique du serveur

3)pour le prouver au client le serveur chiffre un message standard avec la clef secrète et l'envoie

au client qui le déchiffre avec sa clef secrète donc c'est le bon serveur

4)une fois la clef secrète échanger, le client et le serveur peuvent donc établirez

un canal sécuriser grâce à la clef secrète commune, on passe en cryptage symétriques

5)le client peu maintenant envoyer son id et mot de passe au serveur en toute sécurité

II) installation est configuration

1) on utilise la version openssh

-la partie client est normalement installer par défaut : paquet openssh-client

-pour installer le serveur paquet openssh-server

2)configuration

-client : /etc/ssh/ssh_config

```
pi@raspberrypi:/etc/ssh $ cat ssh_config

# This is the ssh client system-wide configuration file.  See
# ssh_config(5) for more information.  This file provides defaults for
# users, and the values can be changed in per-user configuration files
# or on the command line.

# Configuration data is parsed as follows:
#  1. command line options
#  2. user-specific file
#  3. system-wide file
# Any configuration value is only changed the first time it is set.
# Thus, host-specific definitions should be at the beginning of the
# configuration file, and defaults at the end.

# Site-wide defaults for some commonly used options.  For a comprehensive
# list of available options, their meanings and defaults, please see the
# ssh_config(5) man page.

Include /etc/ssh/ssh_config.d/*.conf

Host *
#   ForwardAgent no
#   ForwardX11 no
#   ForwardX11Trusted yes
#   PasswordAuthentication yes
#   HostbasedAuthentication no
#   GSSAPIAuthentication no
#   GSSAPIDelegateCredentials no
#   GSSAPIKeyExchange no
#   GSSAPITrustDNS no
#   BatchMode no
#   CheckHostIP yes
#   AddressFamily any
#   ConnectTimeout 0
#   StrictHostKeyChecking ask
#   IdentityFile ~/.ssh/id_rsa
#   IdentityFile ~/.ssh/id_dsa
#   IdentityFile ~/.ssh/id_ecdsa
#   IdentityFile ~/.ssh/id_ed25519
#   Port 22
#   Ciphers aes128-ctr,aes192-ctr,aes256-ctr,aes128-cbc,3des-cbc
#   MACs hmac-md5,hmac-sha1,umac-64@openssh.com
#   EscapeChar ~
#   Tunnel no
#   TunnelDevice any:any
#   PermitLocalCommand no
#   VisualHostKey no
#   ProxyCommand ssh -q -W %h:%p gateway.example.com
#   RekeyLimit 1G 1h
#   UserKnownHostsFile ~/.ssh/known_hosts.d/%k
    SendEnv LANG LC_*
    HashKnownHosts yes
    GSSAPIAuthentication yes
```

-serveur : /etc/ssh/sshd_config

```
pi@raspberrypi:~ $ cat /etc/ssh/sshd_config
#       $OpenBSD: sshd_config,v 1.103 2018/04/09 20:41:22 tj Exp $

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/bin:/bin:/usr/sbin:/sbin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key
```

repérer :

-port 22 port d'écoute par default
-permitRootLogin yes/no permet de se connecter directement en root (à déconseiller)
-X11Forwarding yes : pour autoriser les transmissions graphiques

pour lancer le server : service sshd start

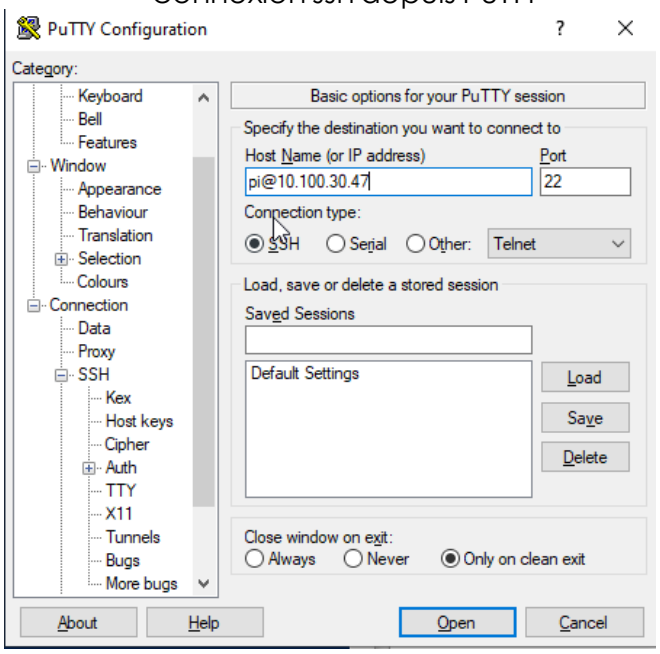
connexion du client : ssh login@machine

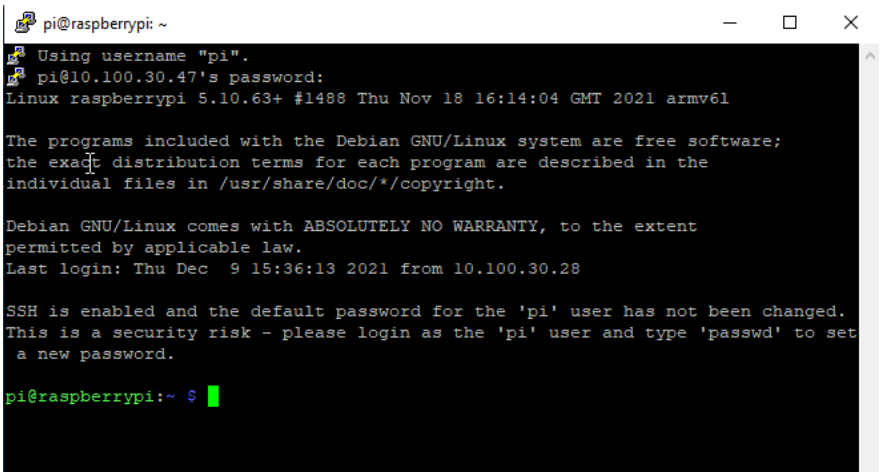
```
PS C:\Windows\system32> ssh pi@10.100.30.47
pi@10.100.30.47's password:
Linux raspberrypi 5.10.63+ #1488 Thu Nov 18 16:14:04 GMT 2021 armv6l

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: Thu Dec  9 14:21:24 2021

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set a new password.

pi@raspberrypi:~ $
```

III) le teste

Cette commande passe dans le Shell, ou PowerShell
ssh pi@10.100.30.47 # pour ouvrir le ssh en mode normal (Shell)
ssh -XC pi@10.100.30.47 # pour ouvrir le ssh en mode graphique + (Shell)

connexion ssh depuis PuTTY





ensuite « yes »
Et pour finir entres votre mot de passe

sudo apt install ufw pour ajouter un fire-wall ➔ il bloquera les connexions suivre tuto fire-wall (bientôt)

Tester le transport de fichier par SSH
cp [options] [source] [destination]
cp /etc/ssh/mathias.html /var/www/html/

copie le fichier mathias.html qui se trouve dans /etc/ssh/ dans le dossier /var/www/html/

mv [source] [destination]
mv /etc/ssh/mathias.html /var/www/html
déplace le fichier mathias.html qui se trouve dans /etc/ssh/ dans le dossier /var/www/html/

SCP

-transfère de fichier
-récupérer tout un dossier
-transférer tout un dossier

scp [autres options] [nom d'utilisateur source@IP]:/[dossier et nom de fichier] [nom d'utilisateur de destination@IP]:/[dossier de destination]

en réel ça donne ça
-scp C:\Users\Mathias\Desktop\01.png vraica@192.168.0.63:/home/vraica/

Tunnel SSH

Il sert à chiffre toutes les communications entre les deux machines

Exemple pour crée un tunnel ssh pour une connexion http du port 2021 au port 80
-ssh -L 2021 :IP_serveur :80 vraica@ip_serveur

Dans votre Navigateur : http://localhost:2021