Configuration Express

SSH

Beginner guide

TECHNICIEN SUPÉRIEUR GESTIONNAIRE
DE RESSOURCES INFORMATIQUES
ET RÉSEAUX

O RLY?

Marchal Ludovic

Table des matières

Openssh

Présentation

Historique

OpenSSH (OpenBSD Secure Shell) est un ensemble d'outils informatiques libres permettant des communications sécurisées sur un réseau informatique en utilisant le protocole SSH.

Créé comme alternative Open Source à la suite logicielle proposée par la société SSH Communications Security (en), OpenSSH est développé depuis 1999 par l'équipe d'OpenBSD, dirigée par son fondateur, Theo de Raadt, et diffusé sous licence BSD.

(source: Wikipédia)

Aspect technique

Composition

Openssh est composé de 2 éléments :

- le client :openssh-client

  qui permet de se connecter à une machine distante

- le serveur :openssh-server

  installé sur une machine distante

Bien sur les 2 éléments peuvent être installé sur une même machine, toutefois la partie serveur n'est pas installée par défaut sur un poste client.

Openssh provenant de OpenBSD, les versions pour Debian et dérivées (comme Ubuntu) se nomment openssh-server et openssh-client en version dite portable, ce qui apparaît dans le numéro de version avec p.

Fonctionnalités

Configuration

Par défaut les fichiers de configuration se situent ici :

- /etc/ssh/sshd_config – partie serveur.
- /etc/ssh/ssh_config – partie client.
- ~/.ssh/ - configuration de l'utilisateur.
- ~/.ssh/authorized_keys - Liste des clés public (RSA or DSA)autorisée
- /etc/nologin – si ce fichier existe, sshd interdit la connexion à tout utilisateur sauf root./etc/hosts.allow et /etc/hosts.deny : Liste de contrôle d'accès.
- SSH default port : TCP 22 (port par défaut)

/etc/ssh/sshd_config

Sshd est le nom du logiciel serveur, utilisant le port 22 pour communiquer, il dispose de nombreuses options.

Le port est défini ici ; ligne 4, il est possible de le changer.

Cela ne change en fait pas grand-chose. Il y a des bot qui scannent les réseaux, des scripts kiddies, ou n'importe qui d'autres..

```
# Package generated configuration file
# See the sshd_config(5) manpage for details
# What ports, IPs and protocols we listen for
Port 22                                    Port d'écoute du serve
```

Restreindre les interfaces d'écoutes, attention en cas de modification du réseau, faire un memento.

```
# Use these options to restrict which interfaces/protocols sshd will bind to
#ListenAddress ::
#ListenAddress 0.0.0.0
```

Version du protocole, choisir 2

```
Protocole 2
# HostKeys for protocol version 2
```

Location des clés

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

Séparation des privilèges.Évidemment on laisse à yes

```
#Privilege Separation is turned on for security
UsePrivilegeSeparation yes
```

Regeneration et taille de clé

Keyregenerationintervalle indique le temps que le daemon attend avant de régénérer la clé. On peut réduire, éventuellement.

ServerKeyBits indique la taille, elle peut être de 1024, 2048 ou 40962

```
# Lifetime and size of ephemeral version 1 server key
KeyRegenerationInterval 3600
ServerKeyBits 1024
```

Authentification

Temps d'attente avant déconnexion, si l'utilisateur ne parvient pas à se connecter.

Interdire la connexion à root.

```
# Authentication:
LoginGraceTime 120
PermitRootLogin no
StrictModes yes
```

Basé sur les clés.

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

Ne pas baser l 'authentification par un hôte, ce n'est pas un gage de sécurité.

```
# Don't read the user's ~/.rhosts and ~/.shosts files
IgnoreRhosts yes
# For this to work you will also need host keys in /etc/ssh_known_hosts
RhostsRSAAuthentication no
# similar for protocol version 2
HostbasedAuthentication no
# Uncomment if you don't trust ~/.ssh/known_hosts for RhostsRSAAuthentication
#IgnoreUserKnownHosts yes
```

Ne pas permettre un MdP vide.

```
# To enable empty passwords, change to yes (NOT RECOMMENDED)
PermitEmptyPasswords no
```

Sur un poste , activer le X11Forwarding permet de déporter l'affichage d'une application distante sur un poste en local.

Pour un serveur on met la valeur à no

```
X11Forwarding yes
```

Le FTP

```
Subsystem sftp /usr/lib/openssh/sftp-server
```

Pam est le gestionnaire de MdP, étant à yes, toutes les politiques de MdP sont appliqué.

```
# Set this to 'yes' to enable PAM authentication, account processing,
# and session processing. If this is enabled, PAM authentication will
# be allowed through the ChallengeResponseAuthentication and
# PasswordAuthentication.  Depending on your PAM configuration,
# PAM authentication via ChallengeResponseAuthentication may bypass
# the setting of "PermitRootLogin without-password".
# If you just want the PAM account and session checks to run without
# PAM authentication, then enable this but set PasswordAuthentication
# and ChallengeResponseAuthentication to 'no'.
UsePAM yes
```

IMPORTANT

- Penser à recharger la configuration après toute modification du fichier avec la commande :
- # /etc/init.d/sshd restart

Installation

Lors de l'installation 2 paires de clé asymétrique sont crée: RSA et DSA, qui sont 2 algorithmes de chiffrement, les clés RAS seront utilisées.

Ces clés se situent dans le répertoire

- /etc/ssh/ssh_host_rsa_key Clé privée
- /etc/ssh/ssh_host_rsa_key.pub Clé public

Il y aura échange de clés lors de la connexion à la machine distante.

Générer une paire de clés

Avant de se connecter à une machine distante via ssh, il faut générer une paire de clé, sous *nix cela se fait très simplement avec la commande :

- ssh-keygen

L'utilisateur est accompagné le long de la création, et devra répondre à une série de questions.

Rien de très compliqué.

```
username@SRV-ubuntu:~$ ssh-keygen
Generating public/private rsa key pair.
Laisser le paramètre suivant tel quel.
Enter file in which to save the key (/home/username/.ssh/id_rsa):
Saisissez une passphrase
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Les retours suivants indiquent ou sont enregistrées les clés.
Your identification has been saved in /home/username/.ssh/id_rsa.
Your public key has been saved in /home/username/.ssh/id_rsa.pub.
Vous avez ici l'empreinte de la clé en regard de la paire identifiant@machine
The key fingerprint is:
46:86:ed:3f:4e:ea:e5:06:f3:52:73:90:8f:11:ba:81 username@SRV-ubuntu
The key's randomart image is:
+--[ RSA 2048]----+
|                 |
|       o  .      |
|      ..+. o     |
|      E+o +      |
|        So =     |
|       .+.+ o    |
|         ==o     |
|        .*o.     |
|       .ooo      |
+-----------------+
```

IMPORTANT

Par défaut les clés sont identifiées avec le couple user@machine

Nous verrons comment créer des clés pour d'autres utilisateurs dans le cadre d'une équipe technique.

Astuce

Il est également possible de changer la phrasepass qui protège la clé avec la commande :

```
ssh-keygen -p -f ~/.ssh/id_rsa
```

Il sera alors demandé l'ancienne phrasepass puis une nouvelle.

Équipe infogérance

Afin de permettre aux différents collaborateurs d'une équipe d'infogérance de se connecter au serveur, cette opération

Création d'une paire de clé pour l'utilisateur ludo avec la commande :

- Ssh-keygen -f ~/cle-equipe/id_rsa_ludo -C « ludo »

  l'option -f indique le fichier

  l'option -C indique l'ID

```
tsgeri@srvxy:~$ ssh-keygen -f ~/cle-equipe/id_rsa_ludo -C "ludo"
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/tsgeri/cle-equipe/id_rsa_ludo.
Your public key has been saved in /home/tsgeri/cle-equipe/id_rsa_ludo.pub.
The key fingerprint is:
21:b0:64:82:5f:be:63:16:7e:34:cf:19:49:b8:96:c0 ludo
The key's randomart image is:
+--[ RSA 2048]----+
| ...+  .         |
|.  +Eo. .        |
| . o...+..       |
|  . o *.o.       |
|   . = +So       |
|    * . +        |
|   o o           |
|                 |
|                 |
+-----------------+
```

Il suffira de remettre la clé privé à l'utilisateur, avec son porte clé USB au logo de l'entreprise, et de placer les clés public dans authoryzed_keys comme indiqué dans le fichier de configuration, ou indiquer un répertoire de son choix.

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

Cela fait également partie de mon projet de synthèse...

Copy de la clé

Une clé public s'envoie, de la machine hôte vers le server, très facilement avec la commande :

- ssh-copy-id <IP-server>

```
>~ ssh-copy-id 172.25.3.110
/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
username@172.25.3.110's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh '172.25.3.110'"
and check to make sure that only the key(s) you wanted were added.
```

Attention

Il s'agit de la clé correspondant au couple username@MachineLocale, il faut donc que cet utilisateur existe sur le système distant.

Connexion

Simplement avec la commande :

- ssh <adresse IP du serveur>

```
>~ ssh 172.25.3.110
Enter passphrase for key '/home/username/.ssh/id_rsa':

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.
You have new mail.
Last login: Thu Feb  4 12:29:07 2016
username@srv-share:~$
```

Changer d'utilisateur

Je me connecte depuis SRV-ubuntu avec le compte tsgeri existant sur la machine 172.25.3.106.

Ceci étant la première connexion au serveur, celui-ci nous envoie son empreinte qui sera ajouté dans ~/.ssh/know_hosts.

- user@machine~$ ssh <autre-utilisateur>@<IP machine distante>

```
username@SRV-ubuntu:~$ ssh tsgeri@172.25.3.106
The authenticity of host '172.25.3.106 (172.25.3.106)' can't be established.
ECDSA key fingerprint is 37:a3:ab:02:83:eb:36:79:93:76:76:21:82:da:e1:82.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.25.3.106' (ECDSA) to the list of known hosts.
tsgeri@172.25.3.106's password:
Welcome to Ubuntu 14.04.3 LTS (GNU/Linux 3.16.0-30-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

  System information as of Mon Dec  7 13:27:24 CET 2015

  System load:  0.15              Processes:           403
  Usage of /:   4.5% of 28.18GB   Users logged in:     1
  Memory usage: 8%                IP address for eth0: 172.25.3.106
  Swap usage:   0%

  Graph this data and manage this system at:
    https://landscape.canonical.com/

7 packages can be updated.
7 updates are security updates.

Last login: Mon Dec  7 13:27:24 2015 from 172.25.3.21
```

Changer de port

Si le port d 'écoute a été modifier il faut l'indiquer lors de la connexion au serveur.

Forcer l'authentification par mot de passe

```
$ ssh -p 1022 username@hostname
```

En indiquant que l'on ne souhaite pas utiliser la clé.

```
$ ssh -o "PubkeyAuthentication no" username@hostname
```

Fichier .pem

Il existe encore d'autre mode d'authentification, les fichiers .pem, il suffit de l'indiquer avec l'option -i

```
$ ssh -i /path/to/identity.pem username@hostname
```

Il peut etre nécessaire de changer les droits sur le fichier.

```
chmod 0600 identity.pem
ou
chmod u+rw identity.pem && chmod go-rwx identity.pem
```

On pourra à ce moment là aliasé la connection. En éditant le fichier vu plus haut, ~/.ssh/config

```
How aws
    HostName some.address.ec2.aws.com
    User awsuser
    IdentityFile  ~/.ssh/aws_identity.pem
    IdentitiesOnly yes
```

ssh-agent

Il est possible d'automatiser la connexion sans taper le MdP en utilisant le module ssh-agent en ajoutant la clé avec la commande :

```
ssh-add
Enter passphrase for /home/username/.ssh/id_rsa:
Identity added: /home/username/.ssh/id_rsa (/home/username/.ssh/id_rsa)
```

Clés autorisées

Les clés autorisées sont enregistrées dans le répertoire de l'utilisateur distant:

~/.ssh/authorized_keys

On peut donner l'accés au compte root en plaçant les clés ici :

/root/.ssh/authoryzed_keys

On peut pour de multiple raison autoriser l'accés et la lecture de certains fichiers, à des utilisateurs n'ayant pas l'accés root.

Ci-dessous la paire de clé de Gognol.

La clé privé , est celle qu'il faut remettre à Gognol, avec le porte clé USB, comme ça c'est niquel.

Il existe des produits sur le marché.

Clé privée

Exemple utilisateur Gognol Premier

```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,F0C8167E9FD9DCF3BACE3BAF122EDFBC

4b+bBvX1wGfIK1FIkwd4/q7FiMEu12dOj02UxdWTplLkEvH94fIQ8QJPQEOmEcFq
suO8NsX9XLs5GzD9gm9CmH+uO0W+VTa6ECAYU7tUmqbJd8IVw+3W5QuqfVmxTOCD
IjA5rq7nJSPG+U8/+MB54byQQ48j6Ehy1QT2gYHRsGhiPj98G1CFfDLf9gUbyjZR
QfogZdKvTxCdHlApKLi5q9FeIOLlLxVpcYPrme51PZV92nqgi0M7rH+sF34BxN+X
tOkRBc1A1owHi4hWKEDjxWiWb0TAG5o1m2PbHIoipkvOrx8bw4JvPQUNuWoWEzE1
vQ959xT58X9gvaPnPTheZAIyD4bpxtO/Hnfey2DqHL62Uf5kUkOpsOX7fitjdR5Z
TbFl6AkQ/LoF60nqhT/xwy6jroeu73RPbK34kBbK+SLZtxa7V8VPw1YRcxrt5Cwl
612Uhx4e+9rPT9en3E+JkF/CnTD6mTQQ9AbTjH8j6rxBFJR4FHyOjJNENp/Ee09k
YJht2rNLRaNVOQZyRXqgW40uFA+2wPE5YSX1+9o7tWh2VB7M26Qo/JI7C69HxIuV
rmfZ7lVafEiq9hn9/TVhFUJDF5A0RlTiCbUIYd4LD3zXm9jz1g2I0bt8lJv9Q5dV
afeEAIlSAONLNAlTR36qodO3gPWkz6d3f1aVbeiq0AZ2u1/fF2MKTbVJJU8YAIrE
8x6kqAYPOA4PzsS5Y0scWOre7wbzyObHED4u1DVTXSFFGCtdUkT84hp32UXYZobZ
oCijmrNC3/bWuxMNGbrE/avI5jPLWPV55cqlAXGqQHjr+A9q1BMMfuDA/uJrCpK3
WYFuJ1RpOHoJ5X5HI6804IcV1Raf93fj34hKvbXtBtQkKuoR8XvTOfrfvBJSNKr1
/SJFqZGYnB80ksCkfK3j43uGH+dJx2fOTb5THGeLV0OPV86Zu4UV2Jypv6C6dL9C
/nIALk9+EcNkVhRItvvUWlESNJcl3UWY/po6SC7jto/1PXCZlH/fdUQK54trCTP/
UOxABUrz1YMOsrNWYrtPKuzuCIQVL20Y8MznsG2XHGm96L4mZpWUkrS31As/odeQ
ZcjBwTdPWTKgjCaXJYuGAixZkvYtu2Sx7IptKsVAvXDChzcnJbBq2/2A7Cy0DYLv
0c+WEyhhuPXhFEIpMjxh8XpINs/f+l/xotkq4h/KE2ycTHeQyFWMRUNQtN4dB0zf
z1mIuuEmRC0aCvEXK7nI66IhZEEFqTYiGG63T34oqDo0/WtPTSaQQfNCGfEvo2C3
ViY9LNaNXUvkVC75RmK0x0MN0PbBPM7/cu/xrYk3MCUIpWSG/nHZ2xj8dQAalNHx
R1cDbxQyKpgsCTV8pm6xkM1NM/+AtEezmCGyiromx+rH6qlHAc4iFlJDF9GVaQxy
qdgLamWgqIJOQskOG4cT7tlUAXTPbOFsURZdsOt1z9pVyuc4iHIprdaQbpbpRqDr
pltpr/fnv0MJ4vHZ8RDr7lvkUfzueviKQWcV7bh95yX3QUDYfJ5ntniGWxdlGt14
/CJ9RWCT34t4/bwjwDbyADdXWrgR1aT3/OAntGX6wh5GpuekOEV/mvsF60FuESOu
-----END RSA PRIVATE KEY-----
```

Clé public

Exemple utilisateur Gognol Premier

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQC5JQIOsES2VH1y3PVfgcZVnsbSbaHHeL22NdsuyUda3lszoM+bYpKEEQ8NuNfu/
eXRx01HQNYMoEydGKp3mA9t9xhAG/bHkYKwoKzSzoYncnA05wPjWfmEW63Hqwwu9ur5Kex367Wbt8BY6mCr+6VTdSLbF6
N6CZxiF3rsRuba9p8rWYEtYTj6ruUMkD9ZIp/fi6slJZmyYXTYzHwMDbmHEwPel/mVuD4I9efESt9lH/XJFdHCrCXCJkj
zyOFa3D9IOyaXoAFdASxiri32ueOYN0eITGP1jXFPzgePUOpx6XPhg39IXxgDvC1PtTXlvW7rJM3UKw2YDlXqnrNvXrp9
Gognol Premier
```

Copy de fichier

Ssh permet la copie de fichiers de manière sécurisée en remplacement rcp (rcp=remotecopy) avec la commande :

- scp user@machine:<chemin distant> <chemin local>

```
scp /home/user/downloads/InstallDB.sql myuser@192.168.8.102:/home/myuser/InstallDB.sql

myuser@192.168.8.102's password:
InstallDB.sql       100% 6885     6.7KB/s   00:00
```

Ici une capture lors de la copy du fichier d'installation de la base de données.

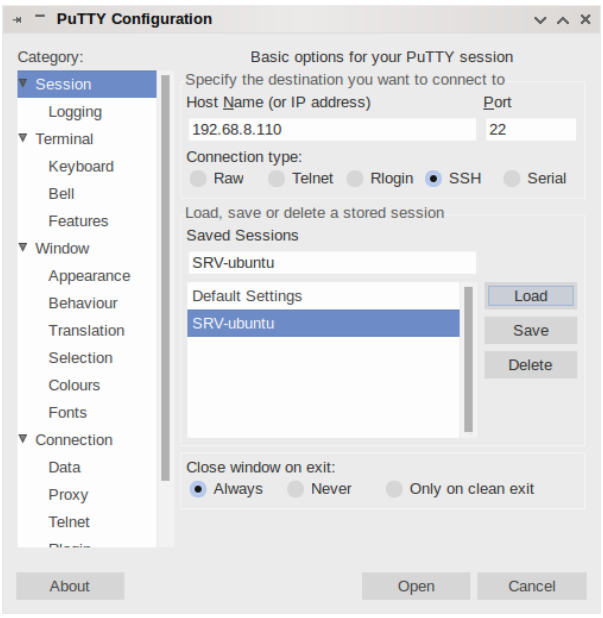
Tunnelisation d'applications

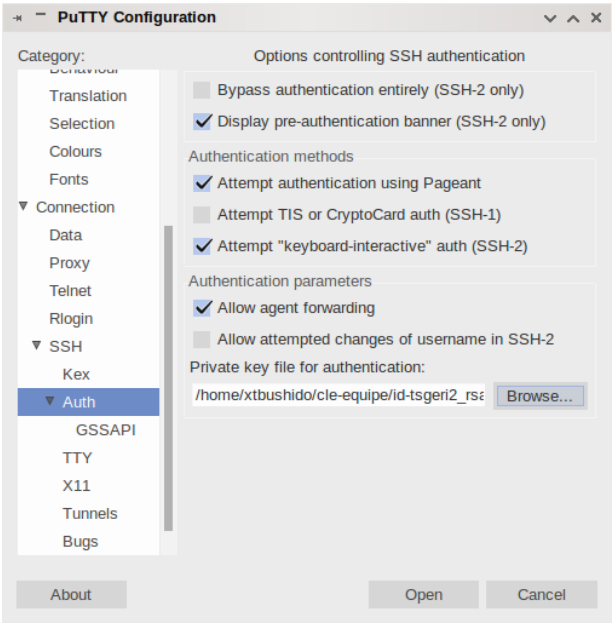
Se connecter a un serveur ftp avec sftp

user@machine~$ sftp user@machine

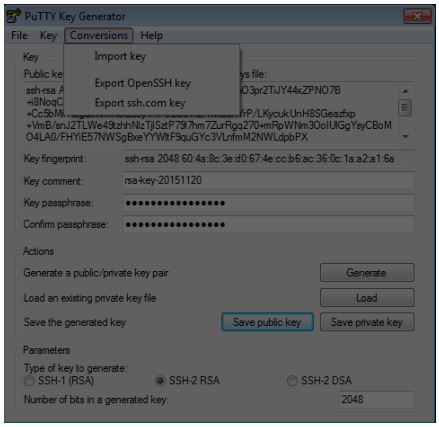
Outil tiers : Putty

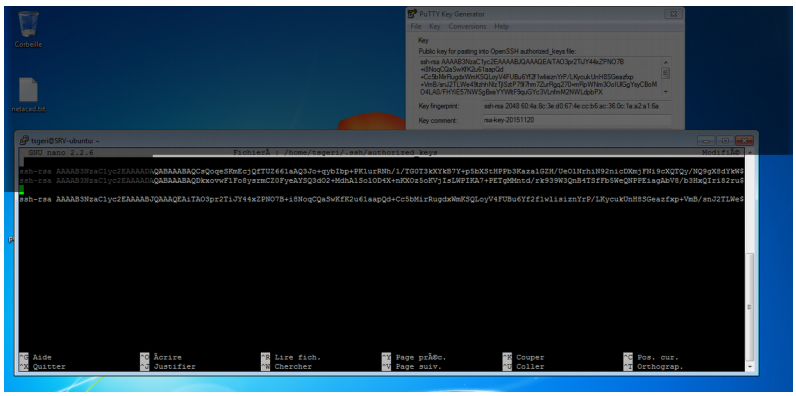
Il existe une interface (à clics) graphique qui se nomme putty disponible sous *nix, disposant de nombreuse fonctionnalités, et options.





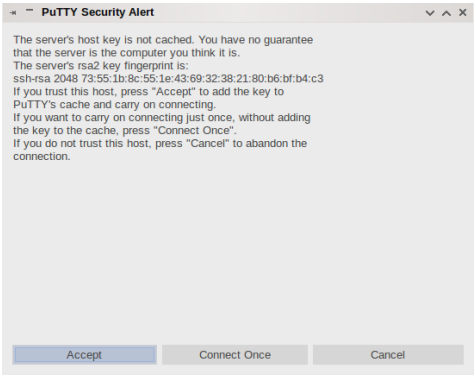
Ainsi que puttygen, en tant que générateur de clés. Et le copier-coller à la souris...

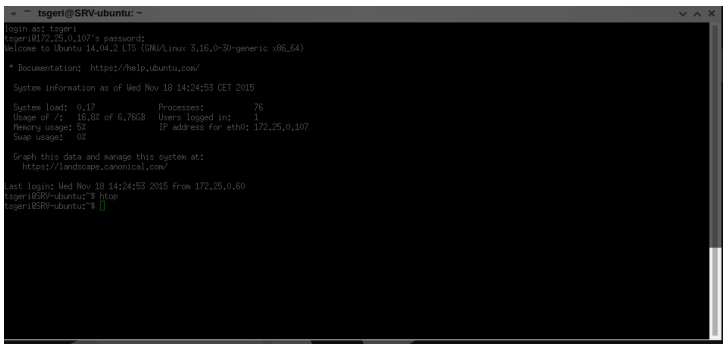




Pour automatiser la connexion, il faut retourner dans putty, et lui indiquer l'emplacement de la clé privé, attention putty ne voit les dossiers cachés (capture 2)

Putty permet tout de même de se connecter avec un autre compte que celui de l'utilisateur local





Bonus1

Un fichier de config épuré :

```
# This is ssh server systemwide configuration file.

        Port 22
        ListenAddress 192.168.1.1
        HostKey /etc/ssh/ssh_host_key
        ServerKeyBits 1024
        LoginGraceTime 600
        KeyRegenerationInterval 3600
        PermitRootLogin no
        IgnoreRhosts yes
        IgnoreUserKnownHosts yes
        StrictModes yes
        X11Forwarding no
        PrintMotd yes
        SyslogFacility AUTH
        LogLevel INFO
        RhostsAuthentication no
        RhostsRSAAuthentication no
        RSAAuthentication yes
        PasswordAuthentication yes
        PermitEmptyPasswords no
        AllowUsers admin
```

Bonus2

Il est possible d'aliaser les machines, il faut pour cela editer, ici 2 exemples et 2 méthodes différentes :

```
~/.ssh/config
$ vim ~/.ssh/config

Host somealias                                  # On indique l'alias de son choix
    HostName example.com                        # On désigne le site
    Port 2222                                   # On indique le port si nécessaire
    User someuser                               #
    IdentityFile  ~/.ssh/id_example             # On indique l'emplacement de la clé
    IdentitiesOnly yes

Host anotheralias                               # On peut donc en créer plusieurs
    HostName 192.168.33.10                      # Aves l'adresse IP
    User anotheruser                            # Un autre utilisateur
    PubkeyAuthentication no                     # Sans utilisation de la clé
```

Références :

Magazine	Linux Pratique	Éditions Diamond	Fleur Broussea u	2013	P. 18	N° 76
Magazine	Les Guides de Linux Magazine	Editions Diamond	Linux Magazine		P. 82	N° 72
Magazine	Linux Pratique	Editions Diamond	Sebastie n Maccagno ni-Munch	2013	P. 20	76

Sites

Openssh.com

Cybercity

http://www.isc.cnrs.fr/informatique/ssh/putty/sshISC-Putty.html

http://www.eila.univ-paris-diderot.fr/sysadmin/windows/tunnels-putty