Bachelier en informatique et Systèmes

Informatique industrielle

3ème année

HELHa
Haute École Louvain
en Hainaut

Catégorie technique

Laboratoire de réseaux

SSH

2015 – 2016

Lievens Benjamin

Table des matières

SSH

1 Serveur SSH

Daemon du serveur DHCP : sshd

Fichier de configuration : /etc/ssh/sshd_config

Lancement/arrêt/redémarrage : service sshd start/stop/restart

Logs: /var/log/message /var/log/secure

2 Fichier et dossier important

Clés publiques des serveurs consultés: /home/jean/.ssh/known_hosts

La clé publique du serveur: /etc/ssh/ssh_host_(rsa ou dsa)_key.pub

La clé privée du serveur: /etc/ssh/ssh_host_(rsa ou dsa)_key

La clé publique du client: ~/.ssh/id_rsa

La clé privée du client: ~/.ssh/id_rsa.pub

Fichier contenant la liste clés autorisée: ~/.ssh/authorized_keys

3 Installer serveur SSH

Dans CentOS c'est déjà préinstaller☺.

4 Lancer SSH en mode debug

/usr/sbin/sshd **-d**

5 Générer une paire de clé SSH

ssh-keygen -t rsa **-b 1024**

6 Se connecter à une autre machine/exécuter une commande/copier un fichier

ssh user2@10.0.0.2

ssh user2@10.0.0.2 'cat /etc/passwd'

scp /root/install.log user2@10.0.0.2:/tmp

7 Générer/Vérifier une empreinte

sha1sum f1.txt > f1.txt.sha1	ou	md5sum f1.txt > f1.txt.md5
sha1sum –c f1.txt.md5	ou	md5sum -c f1.txt.md5

8 Vérifier la clé publique

L'administrateur du serveur génère un 'fingerprint' (chaîne générée lors de la génération de la clé publique du serveur.

La commande 'sshkeygen –lf /etc/ssh/ssh_host_**rsa**_key.pub' permet d'afficher cette empreinte.

Ex. 2048 fe:91:17:91:c1:bd:ab:ae:5e:05:8b:70:40:1b:e8:c2 (**RSA**)

• Le client demande le 'fingerprint' à l'administrateur.

• Le client compare le 'fingerprint' présenté lors du téléchargement à celui reçu de l'administrateur.

9 Régénérer une paire de clé SSH pour des machines clonés

rm –f /etc/ssh_host*

service sshd restart

10 Visualisez les portes d'écoutes d'une connexion

netstat –tn

11 Fichier de configuration SSH

```
Port 22
```
Port d'écoute de sshd

```
Protocol 2
```
Protocoles à supporter (ssh2)

```
ListenAddress 0.0.0.0
```
Adresse Ip de l'interface d'écoute (ici toutes les interfaces)

```
HostKey /etc/ssh/ssh_host_key
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key
```
Toutes les clés privées du serveur

```
KeyRegenerationInterval 3600
ServerKeyBits 1024
```
Regénération d'une clé de session de 1024 bits après 3600 sec. de connexion

```
LoginGraceTime 120
```
Temps accordé à la procédure de login

RSAAuthentication yes
Authentification par paire de clés SSH1 acceptée.

PubkeyAuthentication yes
Authentification par paire de clés SSH2 rsa ou dsa acceptée.

```
HostbasedAuthentication no
IgnoreRhosts yes
RhostsRSAAuthentication no
```
Pour empêcher les authentifications de type remote (car non-sécurisées).

PasswordAuthentication yes
Authentification par mot de passe (rabattage possible en cas d'échec à l'authentification par clés).

```
PermitEmptyPasswords no
```
Mais pas pour les comptes sans mdp.

```
KeepAlive yes
```
Pour éviter que la connexion reste ouverte si le client disparaît. le serveur coupe la connexion s'il ne reçoit plus du client un message « Je suis en vie» envoyé régulièrement par celui-ci.

```
DenyUsers
AllowUsers test admin
```
Autoriser les deux utilisateurs (test et admin) et aucun autre à se connecter.

Voir aussi les directives AllowGroups/DenyGroups

```
PermitRootLogin yes
```
Le root peut-il se connecter ?

```
PermitRootLogin without-password
```
Le root ne peut se connecter que par paire de clés. Cela évite les tentatives d'attaque ssh par force brute sur le compte root.

12 Connexion par mot de passe

12.1 Vérifiez qu'aucun coupe-feu ne tourne sur la machine

iptables –L

service iptables stop

chkconfig --level 35 iptables off

12.2 Vérifiez sur la machine qu'un trousseau de clés existe

#ls –l /etc/ssh

Régénérer une paire de clé SSH pour des machines clonées:

rm –f /etc/ssh_host*

service sshd restart

12.3 Démarrer et configurer le service

Vérifiez que la directive PasswordAuthentication est à yes:

cat /etc/ssh/sshd_config | grep Password

Redémarrer le service et vérifier qu'il est bien relancé:

service sshd restart

ps ax | grep sshd

12.4 Se connecter d'une machine à l'autre

ssh user@10.0.0.1

12.5 Visualiser les ports d'écoutés

netstat –tn

12.6 Utilisez ssh et scp

root@MV1# ssh user2@10.0.0.2 'cat /etc/passwd'

root@MV1# scp /root/install.log user2@10.0.0.2:/tmp

13 Connexion par clés

13.1 Vérifier les clauses PubkeyAuthentication

grep PubkeyAuthentication /etc/ssh/sshd_config

13.2 Générer un trousseau de clé et copier sa clé publique

user1@MV1$ ssh-keygen –t rsa

user1@MV1$ cd ~/.ssh

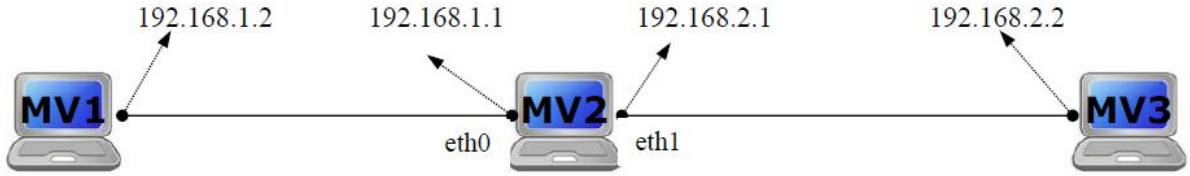
user1@MV1 .ssh$ ssh-copy-id –i id_rsa.pub user3@10.0.0.3

13.3 Attention aux permissions

Permissions

/home/user3	→ 700
/home/user3/.ssh	→ 700
/home/user3/.ssh/authorized_keys	→ 600

14 Port forwarding



14.1 - vérifiez qu'aucun coupe-feu ne tourne sur aucune machine

Sur chaque station:

iptables –L

service iptables stop

chkconfig --level 35 iptables off

14.2 -N'oubliez pas d'activer l'ip forwarding sur MV2

mcedit /etc/sysctl.conf

Ip_forward=1

#sysctl -p

14.3 Stopper les services sshd sur MV1 et MV2 (s'ils tournent)

MV1# service sshd stop

MV2# service sshd stop

14.4 Configurez un service sshd sur MV3 et relancez le service

(bien vérifiez que la directive 'PasswordAuthentication' est à yes)

cat /etc/ssh/sshd_config | grep Password

service sshd restart

ps ax | grep sshd

14.5 Vérifiez sur MV3 qu'un trousseau de clés existe

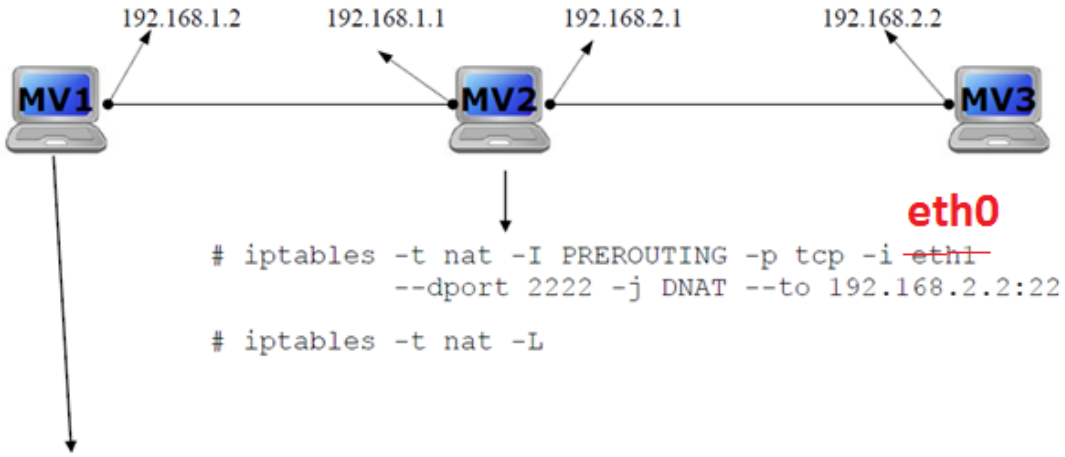
MV3# ls –l /etc/ssh

14.6 Créez user3 sur MV3

root@MV3# adduser user3

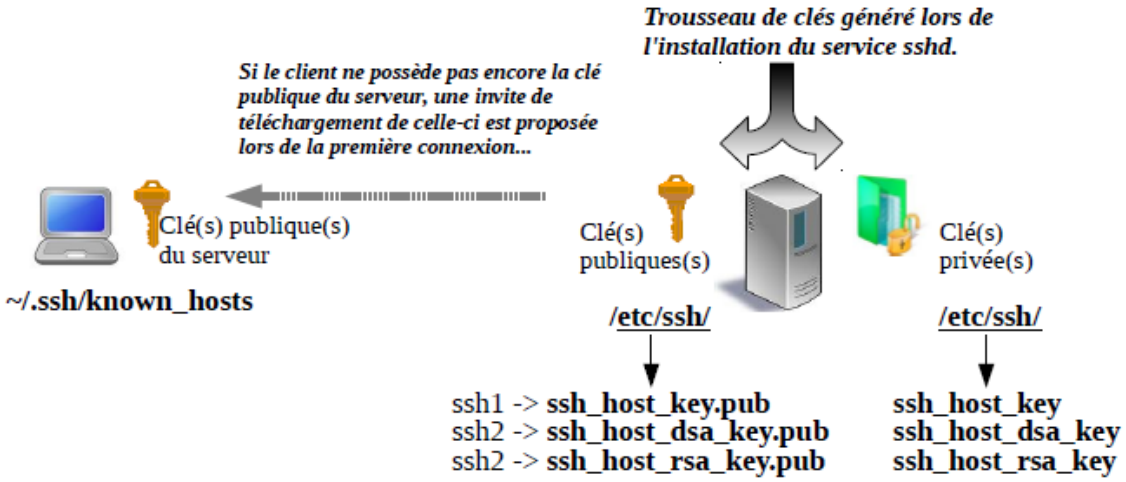
root@MV3# passwd user3

14.7 Configurez et testez un forwarding de port de MV1 vers MV3 en passant par MV2.



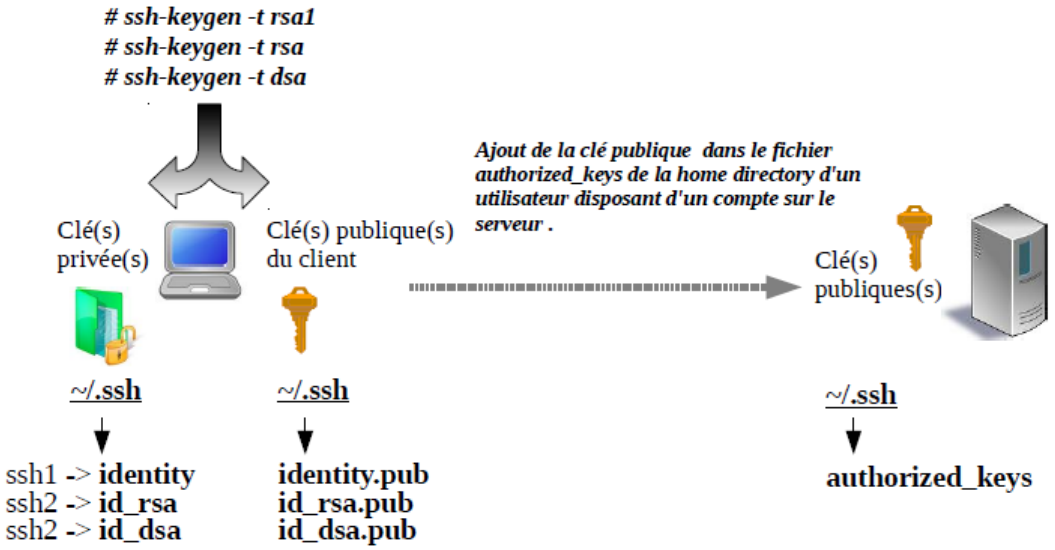

```
# slogin -p 2222 user3@192.168.1.1
Password: …

ou

# ssh -p 2222 -l userb 192.168.1.1
Password: …
```

15 Clé du serveur



Rem. ssh-keygen -t rsa **-b 1024** → *pour générer une clé de 1024 bits ...*

16 Clé du client



ssh-copy-id *username@adresse* copie sa clé publique dans l'authorized_keys de l'autre user.

Exercices: Gestion d'un parc Linux

/root/admin/initvar.sh

```
#! /bin/bash
PUBKEYDIR="/root/.ssh"
SHDIR="/root/admin/sh"
LOGDIR="/root/admin/log"
IPFILE="/root/admin/ip.txt"
NAMELOG=`basename $0`
```

/root/admin/ip.txt

```
10.0.0.2
10.0.0.3
```

Exercice 1:

Ecrivez et testez un script (pushkey.sh) qui déploie la clé publique de root de la machine d'administration (MV1) vers toutes les stations du parc.

/root/admin/sh/pushkey.sh

```
#! /bin/bash

. /root/admin/initvar.sh

cd $SHDIR
date > $LOGDIR/$NAMELOG.log
date > $LOGDIR/$NAMELOG.errors.log

for IP in `cat $IPFILE`
do
  if ping -c 2 $IP >/dev/null 2>&1
  then
     ssh-copy-id -i $PUBKEYDIR/id_rsa.pub root@$IP
     echo "Copie vers $IP... OK" >> $LOGDIR/$NAMELOG.log
  else
     echo "$0: $IP ne repond pas" >> $LOGDIR/$NAMELOG.errors.log
  fi
done
exit 0
```

Exercice 2:

Ecrivez et testez un script (haltall.sh) qui éteint toutes les stations du parc encore « on-line ».

Programmez l'exécution de ce script à 21h00 tous les jours.
Pour ce faire le package crontab doit être installé...

```
# ps ax | grep crond
# yum install crontab (si nécessaire)

# crontab –e
00 21 * * * sh /root/admin/sh/haltall.sh
```

/root/admin/sh/haltall.sh

```
#! /bin/bash

. /root/admin/initvar.sh

cd $SHDIR
date > $LOGDIR/$NAMELOG.log
date > $LOGDIR/$NAMELOG.errors.log

for IP in `cat $IPFILE`
do
  if ping -c 2 $IP >/dev/null 2>&1
  then
     ssh root@$IP "shutdown -h now"
     echo "Arret de $IP... OK" >> $LOGDIR/$NAMELOG.log
  else
     echo "$0: $IP ne repond pas" >> $LOGDIR/$NAMELOG.errors.log
  fi
done
exit 0
```

Exercice 3:

Ecrivez et testez un script (chpwdroot.sh) qui change mot de passe de root sur toutes les stations du parc. Le nouveau de passe est passé en argument au script.

/root/admin/sh/chpwdroot.sh

```
#! /bin/bash

. /root/admin/initvar.sh

if [ $# -eq 1 ]
then
  cd $SHDIR
  date > $LOGDIR/$NAMELOG.log
  date > $LOGDIR/$NAMELOG.errors.log

  for IP in `cat $IPFILE`
  do
    if ping -c 2 $IP >/dev/null 2>&1
    then
       ssh root@$IP "echo $1 | passwd --stdin root > /dev/null
                                                    2>&1"
       echo "Changement du mdp de root sur $IP... OK"
                                        >> $LOGDIR/$NAMELOG.log
    else
       echo "$0: $IP ne repond pas"
                                   >> $LOGDIR/$NAMELOG.errors.log
    fi
  done
else
  echo "Erreur: Un et un seul argument"
  exit 1
fi
exit 0
```

Exercice 4:

Ecrivez et testez un script (`alladduser.sh`) qui ajoute un compte utilisateur à chaque station du parc.

Le nom de l'utilisateur (ex. `toto`) sera passé en argument. Son mot de passe sera identique à son nom.

`/root/admin/sh/alladduser.sh`

```
#! /bin/bash

. /root/admin/initvar.sh

if [ $# -eq 1 ]
then
  echo $LOGDIR/$NAMELOG
  cd $SHDIR
  date > $LOGDIR/$NAMELOG.log
  date > $LOGDIR/$NAMELOG.errors.log

  for IP in `cat $IPFILE`
  do
    if ping -c 2 $IP >/dev/null 2>&1
    then
       ssh root@$IP "adduser $1 > /dev/null 2>&1"
       ssh root@$IP "echo $1 | passwd --stdin $1 > /dev/null
                                                    2>&1"
       echo "Ajout de l'utilisateur $1 sur $IP... OK"
                                        >> $LOGDIR/$NAMELOG.log
    else
       echo "$0: $IP ne repond pas"
                                  >> $LOGDIR/$NAMELOG.errors.log
    fi
  done
else
  echo "Erreur: Un et un seul argument"
  exit 1
fi
exit 0
```

Exercice 5:

Ecrivez et testez un script (`chgrub.sh`) qui permet de changer le 'timeout' du multi-boot de chaque machine à 5 secondes.

/root/admin/sh/chgrub.sh

```
#! /bin/bash

. /root/admin/initvar.sh

cd $SHDIR
date > $LOGDIR/$NAMELOG.log
date > $LOGDIR/$NAMELOG.errors.log

for IP in `cat $IPFILE`
do
  if ping -c 2 $IP >/dev/null 2>&1
  then
     ssh root@$IP "/bin/sed –i –e \"s/timeout=5/timeout=0/g\"
                                              /boot/grub/grub.conf"
     echo "Change timeout on $IP... OK" >> $LOGDIR/$NAMELOG.log
  else
     echo "$0: $IP ne repond pas" >> $LOGDIR/$NAMELOG.errors.log
  fi
done
exit 0
```

Exercice 6:

Ecrivez et testez un script (`chnetwork.sh`) qui change la configuration ip de chaque station du parc afin de les disposer sur le réseau d'adresse 192.168.0.0/24.

`/root/admin/sh/chnetwork.sh`

```
#! /bin/bash

. /root/admin/initvar.sh

  cd $SHDIR
  date > $LOGDIR/$NAMELOG.log
  date > $LOGDIR/$NAMELOG.errors.log

  cpt=2
  for IP in `cat $IPFILE`
  do
    if ping -c 2 $IP >/dev/null 2>&1
    then
       echo "DEVICE=eth0" > /tmp/ifcfg-eth0
       echo "BOOTPROTO=static" >> /tmp/ifcfg-eth0
       echo "IPADDR=192.168.0.$cpt" >> /tmp/ifcfg-eth0
       echo "NETMASK=255.255.255.0" >> /tmp/ifcfg-eth0
       echo "ONBOOT=yes" >> /tmp/ifcfg-eth0
       ssh root@$IP "cp /etc/sysconfig/network-scripts/
                                ifcfg-eth0 /root/ifcfg-eth0.bak"
       scp /tmp/ifcfg-eth0 root@$IP:/etc/sysconfig/network-
                             scripts/ifcfg-eth0 > /dev/null 2>&1
       echo "Changement de la configuration de $IP en
                    192.168.0.$cpt... OK" >> $LOGDIR/$NAMELOG.log
       cpt=`expr $cpt + 1`
    else
       echo "$0: $IP ne repond pas"
                                   >> $LOGDIR/$NAMELOG.errors.log
    fi
  done
rm -f /tmp/ifcfg-eth0
exit 0
```