

# CHAPITRE 2

## SSH

Puisque qu'ici nous allons faire pas mal de chose nous allons passez en « root » permanent :

```
ALBAN@bebeserv:~$ sudo -i
bebeserv:~#
```

Installons le client et le serveur SSH. Seul le serveur devrait nous servir mais il est toujours très pratique d'avoir un client à porter de main.

```
bebeserv:~# apt-get update
[...]
bebeserv:~# apt-get install openssh-client openssh-server
[...]
```

Commencez par couper « sshd » tant qu'il n'est pas configuré correctement :

```
bebeserv:~# /etc/init.d/ssh stop
[...]
```

Maintenant allons configurer tout ceci :

```
bebeserv:~# cd /etc/ssh
bebeserv:/etc/ssh#
```

Tout d'abord le client ssh :

```
bebeserv:/etc/ssh# nano ssh_config
```

Mettez les options suivantes :

```
CheckHostIP yes
EnableSSHKeysign yes
GSSAPIAuthentication yes
GSSAPIDelegateCredentials no
```

```
Protocol 2
ServerAliveCountMax 5
ServerAliveInterval 20
SetupTimeOut 30
TCPKeepAlive no
VerifyHostKeyDNS yes
```

Enregistrez et quittez.

Maintenant passons au serveur :

```
bebeserv:/etc/ssh# nano sshd_config
```

Mettez les options suivantes :

```
X11Forwarding
MaxAuthTries 3
MaxStartups 1
AllowUsers ALBAN
LoginGraceTime 60
PermitRootLogin no
AuthorizedKeysFile %h/.ssh/authorized_keys
X11Forwarding no
TCPKeepAlive no
Banner /etc/ssh/banner
ClientAliveCountMax 5
ClientAliveInterval 20
```

Enregistrez et quittez.

Créons notre bannière :

```
bebeserv:/etc/ssh# nano banner
```

Entrez le texte de votre choix :

```
ATTENTION les connexions sur ce serveur sont surveillées et tracées.

Tout acte malveillant, usurpation d'identité, vol de données pourra
amener des poursuites judiciaires importantes !
```

Nous allons maintenant redémarrer notre serveur.

```
bebeserv:/etc/ssh# /etc/init.d/ssh start
```

```
[...]
```

Maintenant nous allons créer les clés RSA et DSA de notre utilisateur qui lui permettra d'assurer ses futures connexions.

```
bebeserv:/etc/ssh# ssh-keygen -t dsa -b 1024  
[...]
```

Des questions vous seront posées, faites « [Entrée] » partout sans répondre.

```
bebeserv:/etc/ssh# ssh-keygen -t rsa -b 1024  
[...]
```

Même démarche.

Maintenant que tout est beau et propre nous allons devoir autoriser SSH à sortir de notre réseau.

Éditez le fichier « /etc/network/iptables » et ajoutez les lignes suivantes :

```
iptables -t filter -A INPUT -p tcp --dport 22 -i eth0 -d $MYIP -j  
ACCEPT  
iptables -t filter -A INPUT -p tcp --dport 22 -o eth0 -s $MYIP -j  
ACCEPT
```

Maintenant SSH est ouvert à l'extérieur.

Vous pouvez vous logger depuis votre serveur vers un autre serveur en utilisant :

```
ssh user@server.com
```

Si vous souhaitez vous connecter à un serveur depuis le votre sans avoir à y rentrer la clé en permanence :

```
ssh-copy-id -i ~/.ssh/id_dsa.pub user@server.com
```

Rentrez le mot de passe pour la dernière fois.

Après « ssh user@server.com » se connectera automatiquement.

Notre configuration est faite de sorte que seul l'utilisateur « ALBAN » puisse utiliser SSH pour une connexion distante.

Ainsi avec Putty.exe sous un poste Windows par exemple vous pouvez vous connecter et

administrer votre serveur à distance.

Allons un peu plus loin et protégeons-nous des attaques SSH, nous allons installer le paquet « fail2ban » qui nous protégera de ceci et qui part la suite nous protégera aussi de ce même type d'attaque pour les protocoles HTTP, POP etc.

Le principe de « fail2ban » est très simple, il scan les fichiers de log « /var/log/auth.log », « /var/log/apache/access.log » etc ... à la recherche de tentative d'authentification qui sont en échec. Au bout de 3 tentatives échouées par exemple il ajoutera une règle dans « iptables » afin de bannir pendant un certain temps l'adresse IP qui tente de se connecter sans y arriver. Ainsi les attaques brute force sont compromises.

Vous l'aurez compris « fail2ban » joue avec « iptables » nous allons donc revenir sur les fichiers du chapitre 1 pour arranger tout ça.

Première chose, installons « fail2ban » :

```
bebeserv:/etc/ssh# apt-get install fail2ban
[...]
```

Une fois fais, « fail2ban » est déjà en action, voyez les modifications apporter sur notre « iptables » :

```
bebeserv:/etc/ssh# iptables -L -v -n
```

Afin de que notre script « iptables-conf » et celui de « fail2ban » ne s'embête pas au démarrage, sachant que les deux vont de paire nous allons supprimer le démarrage de « fail2ban » tout en gardant son script de démarrage de la manière suivante :

```
bebeserv:/etc/ssh# /etc/init.d/fail2ban stop
bebeserv:/etc/ssh# update-rc.d -f fail2ban remove
```

Maintenant nous allons améliorer « iptables-conf » pour y inclure « fail2ban »

```
bebeserv:/etc/ssh# nano /etc/init.d/iptables-conf
```

Modifiez les fonctions « start » et « stop » de la manière suivante :

```
iptables_start() {
    if [ -f /etc/network/iptables ]; then
        . /etc/network/iptables
    fi
}
```

```
        /etc/init.d/fail2ban start
    }

iptables_stop() {
    /etc/init.d/fail2ban stop
    iptables -t filter -F INPUT
    iptables -t filter -P INPUT ACCEPT
    iptables -t filter -F OUTPUT
    iptables -t filter -P OUTPUT ACCEPT
}
}
```

Enregistrez et quittez.

Maintenant allons paramétrer « fail2ban » :

Rendez vous dans « /etc/fail2ban/ »

```
bebeserv:/etc/ssh# cd /etc/fail2ban
bebeserv:/etc/fail2ban# nano fail2ban.conf
```

Personnellement ici je change la dernière ligne du fichier de configuration :

```
socket = /var/run/fail2ban.sock
```

Je trouve ceci plus propre que de le mettre dans « /tmp », autant l'enregistrer avec ses collègues.

Maintenant réglons la manière dont « fail2ban » travail :

```
bebeserv:/etc/fail2ban# nano jail.conf
```

« bantime = 600 » et « maxretry = 3 » me semble être des bonnes valeurs (pour 3 mauvaises tentatives le host est banni 10 minutes)

Je ne rajoute pas mon propres réseau car je préfère ne pas faire confiance à mon réseau (question de sécurité, un cheval de troie sur un de mes PCs pourrait faire des dégâts)

Plus bas vous voyez des sections dont une pour SSH activée. Nous laissons tel quel pour l'instant, nous activerons les autres services au fur et à mesure ou nous les monterons.

Quittez et enregistrez si vous avez fait des modifications.

Redémarrez le tout :

```
bebeserv:/etc/fail2ban# /etc/init.d/iptables-conf restart
```

Voilà c'est terminer pour SSH.

Une dernière chose avant de clore le sujet.

Vous avez sûrement remarquer qu'un nouveau fichier de log sera présent dans « /var/log » : « fail2ban.log ».

Il est de bon ton d'aller vérifier si le « logrotate » fera tourner ce fichier de log avant qu'il ne devienne trop gros et génère quelque défaillance du système. (il faut toujours avoir ce reflex).

Allons donc dans « /etc/logrotate.d/ »

```
bebeserv:/etc/fail2ban# cd /etc/logrotate.d/  
bebeserv:/etc/logrotate.d# ls
```

Il y a bien un fichier visiblement dédié à « fail2ban »

Affichons le :

```
bebeserv:/etc/fail2ban# cat fail2ban
```

De ce que l'on peut en lire, ce fichier sera tourné toutes les semaines avec 4 fichiers conservés et compressés. Pour moi c'est bon =)

```
bebeserv:/etc/fail2ban# exit  
logout  
ALBAN@bebeserv:~$
```