LINUX: Créer une prison chroot minimale avec Dropbear et BusyBox sur CentOS

Table of Contents

Cet article décrit la construction d'un environnement minimal ne contenant qu'un shell BusyBox et accessible via le serveur SSH de Dropbear. Cette configuration est probablement inutile pour une utilisation quotidienne, mais peut être utilisée comme base de test avec un minimum de bibliothèques disponibles. L'utilisation de cette configuration donne la possibilité d'expérimenter avec un environnement minimal sans reconstruire de petites images Linux.

Afin de créer cet environnement on va créer un système Linux accessible via SSH et contenant le moins possible de bibliothèques en utilisant:

- **Dropbear**, une alternative au serveur OpenSSH
- **BusyBox** pour disposer d'un shell. BusyBox est un seul exécutable qui contient des alternatives pour les utilitaires Linux courants.
- **Une prison chroot** est un environnement isolé fonctionnant au-dessus d'un autre, il exécute des processus sous une racine modifiée (un sous-répertoire de la racine d'origine).

Dropbear s'exécutera sous la prison chroot. Le système de fichiers racine de dropbear est en réalité `/chroot/jail`. L'accès au serveur SSH dans la prison peut être effectué à l'aide de la même adresse IP et du port sur lequel Dropbear est à l'écoute. Lorsque cet environnement est en place, on peuts facilement ajouter ou supprimer des bibliothèques pour vérifier les dépendances ou la compatibilité. Ces bibliothèques ne doivent pas nécessairement correspondre à celles de l'hôte.

Préparation chroot

La première chose à faire est de créer la structure du système de fichiers racine dont on a besoin pour un système minimal sous le répertoire choisi comme notre prison racine. Dans cet exemple: `/chroot/busybox`.

```
mkdir -pv /chroot/busybox

  mkdir: created directory ‘/chroot’
  mkdir: created directory ‘/chroot/busybox’
cd /chroot/busybox/
```

```
mkdir -pv dev/pts proc etc lib usr/lib var/run var/log

  mkdir: created directory ‘dev’
  mkdir: created directory ‘dev/pts’
  mkdir: created directory ‘proc’
  mkdir: created directory ‘etc’
  mkdir: created directory ‘lib’
  mkdir: created directory ‘usr’
  mkdir: created directory ‘usr/lib’
  mkdir: created directory ‘var’
  mkdir: created directory ‘var/run’
  mkdir: created directory ‘var/log’

ln -s lib/ lib64
```

Outre la structure de répertoires, nous avons également besoin de périphériques et de liens vers les répertoires `/dev/pts` et `/proc` d'origine:

```
mknod dev/urandom c 1 9
chmod 0666 dev/urandom
mknod dev/ptmx c 5 2
chmod 0666 dev/ptmx
mknod dev/tty c 5 0
chmod 0666 dev/tty
mount -o bind /dev/pts dev/pts/
mount -o bind /proc proc/
```

Puisqu'on veut se connecter au chroot, on a aussi besoin d’autres fichiers:

```
cp /etc/localtime etc/
cp /etc/nsswitch.conf etc/
cp /etc/resolv.conf etc/
cp /etc/host.conf etc/
cp /etc/hosts etc/
cp /etc/shells etc/
touch var/log/lastlog
touch var/run/utmp
touch var/log/wtmp
```

Compilation de Dropbear

Maintenant que la base du chroot est prête, on peut compiler la dernière version de Dropbear.

Tout d’abord, installer tous les composants nécessaires à la compilation de Dropbear, on a besoin d’un compilateur, de make et de quelques dépendances.

```
yum install wget gcc make zlib-devel bzip2 -y
```

```
...
Complete!
```

Télécharger la source de Dropbear et l'extraire. On peut trouver le code source de la dernière version sur: `https://matt.ucc.asn.au/dropbear/releases/`.

```
wget https://matt.ucc.asn.au/dropbear/releases/dropbear-2015.71.tar.bz2
  --2015-12-09 16:49:38--
https://matt.ucc.asn.au/dropbear/releases/dropbear-2015.71.tar.bz2
  Resolving matt.ucc.asn.au (matt.ucc.asn.au)... 130.95.13.18,
2405:3c00:5200:100::18
  Connecting to matt.ucc.asn.au (matt.ucc.asn.au)|130.95.13.18|:443...
connected.
  HTTP request sent, awaiting response... 200 OK
  Length: 1616280 (1.5M) [application/x-bzip2]
  Saving to: ‘dropbear-2015.71.tar.bz2’

  100%[=====================================================>] 1,616,280
596KB/s   in 2.7s

  2015-12-09 16:49:42 (596 KB/s) - ‘dropbear-2015.71.tar.bz2’ saved
[1616280/1616280]

tar -xvjf dropbear-2015.71.tar.bz2
```

Exécuter `./configure` pour générer un fichier `options.h` et un fichier `make` pour la compilation. Indiquer l’emplacement de la prison chroot dans le préfixe.

`./configure --prefix=/chroot/busybox checking for gcc... gcc ... configure: Now edit options.h to choose features.`

Comme le suggère le résultat de `./configure`, éditer options.h pour désactiver le transfert X. Cela exigerait que l'on ai Xauth et quelques dépendances. Puisque on veut avoir un nombre minimal de bibliothèques dépendantes,on peut éviter cela.

`sed -i '/define\ XAUTH_COMMAND/d' options.h sed -i '/define\ ENABLE_X11FWD/d' options.h`

Consulter toutes les définitions du fichier pour personnaliser un peu plus.

Après les modifications, on peut compiler Dropbear et l’installer dans la prison chroot:

```
make
...
make install
  install -d /chroot/busybox/sbin
  install dropbear /chroot/busybox/sbin
```

```
  install -d /chroot/busybox/share/man/man8
  install -m 644 ./dropbear.8 /chroot/busybox/share/man/man8/dropbear.8
  install -d /chroot/busybox/bin
  install dbclient /chroot/busybox/bin
  install -d /chroot/busybox/share/man/man1
  if test -e dbclient.1; then install -m 644 dbclient.1
/chroot/busybox/share/man/man1/dbclient.1; fi
  install -d /chroot/busybox/bin
  install dropbearkey /chroot/busybox/bin
  install -d /chroot/busybox/share/man/man1
  if test -e dropbearkey.1; then install -m 644 dropbearkey.1
/chroot/busybox/share/man/man1/dropbearkey.1; fi
  install -d /chroot/busybox/bin
  install dropbearconvert /chroot/busybox/bin
  install -d /chroot/busybox/share/man/man1
  if test -e dropbearconvert.1; then install -m 644 dropbearconvert.1
/chroot/busybox/share/man/man1/dropbearconvert.1; fi
```

Tout ce qui reste à faire pour la partie Dropbear, c'est de générer des clés qui seront utilisées par Dropbear. Dropbear a son propre format de clé, il est donc préférable d'utiliser ces utilitaires:

```
mkdir -pv /chroot/busybox/etc/dropbear

  mkdir: created directory ‘/chroot/busybox/etc/dropbear’

/chroot/busybox/bin/dropbearkey -t dss -f
/chroot/busybox/etc/dropbear/dropbear.dss

  Generating key, this may take a while...
  Public key portion is:
  ssh-dss
AAAAB3NzaC1kc3MAAACBAMaVlM+SXpO6y7SSkPZcW87XvAw5oGXq1TAk+guDWEOezUN0Q8HNfYtm
E8z7WwiXPOabJEaJgpOCfJWFxjpcc2OtZ+lDYesNJ8tfEFx6Pi3ObFp6UH0X9DCbZNes6iwxjuJR
kRdB95rId3t4Cu/jKaqqJAY/U2U1TDXTVMwohQi/AAAAFQCqJbXOrkLadJ9+CF4ct96VJIDjewAA
AIEAxIS6PfvqhPGGiaHbbSuu/60cUukdn6+SgJy11G1fUw70FssLndBSdMLHFQWMfl1CtYACC+f6
sX4MoqzAKJoAKDwwt5ro+Q7afVj2XgeCghyosL9IoBx7HWzDXu+9ESCnGh9GQ1XtCoHS0wj9NFSj
VT2KwUuE9LDh5qIRHnmql/IAAACAKr2ZePSW7ovjIx7By4EjhdJaOScnDxRMXWfmmURBDvSf7RWt
+1icBr2zkBKSgyA8u4/v9Qhv9TijW98Y+/G84vcFW6L9qfkI1ZLo1wSDFpVo/bg4YjIqsnwY2W8A
xVxNgJAj3TovPMk74GMcVwTllA8KVNugEvglwmNU2D9wapA= root@cen7
  Fingerprint: md5 b0:e1:25:d7:c1:2f:18:24:55:94:2b:d2:88:6f:98:76

/chroot/busybox/bin/dropbearkey -t rsa -s 4096 -f
/chroot/busybox/etc/dropbear/dropbear.rsa
  Generating key, this may take a while...
  Public key portion is:
  ssh-rsa
AAAB3NzaC1yc2EAAAADAQABAAACAQDTCBH2uVvrzmQHoW93dX52DnqT8jIkb0XI3Y6mdSN8ziepa
Ge9vQLYZs6rN4TfYqJTvpUFziS3Bj2FwKFKamrfvTw1uomTB/7YcfRtXnZLfuY3YXO+yKFcQfGzt
e7ulzcFKXZsnZhM1tDIH42+50z/qOdW7ZpnCtaMGD6ue3E6j0m/eqCQjkpRrEMNxTt5YKDIhtmq/
zZOCGdTuUTvxzkqQwFWz2hwS5gruCHrcZtkRv9RzRnjspbbcQ0fDKns4jbjOhbsJtH06bW4FtSCd
```

```
y67ySX88HzIwiAqGzJF0R8HtYM/yFgdemdOsgn+BEfgzcQjkdAdkLL1tmmgyTQZfndwTy1Q4zMHc
wv3HCZJecGjQe67xJyWTNf5uH17Dlc5FYkyjf8MhPACWqbxyyRCSf3XOm0Y4JMyYtSGvR2iWnolN
nrvdjYH+zhEWs8gfA2U74c5pvk6GrIprXnp9Ys/P0ItDXWA2kserN2JBWbuUyRWUPhJJAcDHSeV3
SpCPNwwBmWPBkkduzqJCEkjJklTlPgBYEQxMcxrx6TkicsasFJEMzeD++s/tc5dMaedsMH5fwiEM
QXT7n/9NTB3tEyY8PezoNai47HUjytzbiFsaPf5H8jFZ9xeEMP2Ce96+4A4elAzL4zZbEqqbyntC
wkC9vqb+dIClU9+WF+2haQ9+NJjCQ== root@cen7
  Fingerprint: md5 ee:33:a3:1d:7b:69:15:f4:38:25:1b:35:b3:aa:93:82
```

Une dernière chose nécessaire pour pouvoir exécuter Dropbear dans le chroot est de vérifier les bibliothèques dont Dropbear a besoin:

```
ldd /chroot/busybox/sbin/dropbear
        linux-vdso.so.1 =>  (0x00007ffd8e70b000)
        libutil.so.1 => /lib64/libutil.so.1 (0x00007f2546bbd000)
        libz.so.1 => /lib64/libz.so.1 (0x00007f25469a7000)
        libcrypt.so.1 => /lib64/libcrypt.so.1 (0x00007f254676f000)
        libc.so.6 => /lib64/libc.so.6 (0x00007f25463ae000)
        libfreebl3.so => /lib64/libfreebl3.so (0x00007f25461ab000)
        /lib64/ld-linux-x86-64.so.2 (0x00007f2546dc6000)
        libdl.so.2 => /lib64/libdl.so.2 (0x00007f2545fa6000)
```

On peut simplement copier ces bibliothèques et certaines bibliothèques supplémentaires requises pour la connexion (et les mots de passe masqués) dans la lib (ou lib64) de la prison chroot:

```
cp /lib64/libutil.so.1 /chroot/busybox/lib/
cp /lib64/libz.so.1 /chroot/busybox/lib/
cp /lib64/libcrypt.so.1 /chroot/busybox/lib/
cp /lib64/libc.so.6 /chroot/busybox/lib/
cp /lib64/libfreebl3.so /chroot/busybox/lib/
cp /lib64/libdl.so.2 /chroot/busybox/lib/
cp /lib64/ld-linux-x86-64.so.2 /chroot/busybox/lib/
cp /lib64/libnss_dns.so.2 /chroot/busybox/lib/
cp /lib64/libnss_files.so.2 /chroot/busybox/lib/
cp /lib64/libnspr4.so /chroot/busybox/lib/
cp /lib64/libfreeblpriv3.chk /chroot/busybox/lib/
cp /lib64/libfreeblpriv3.so /chroot/busybox/lib/
cp /usr/lib64/libz.so.1 /chroot/busybox/usr/lib/
```

Installation de BusyBox

La prochaine partie de l'installation consiste à «installer» BusyBox. L'installation se fait entre guillemets, car installer BusyBox n'est rien de plus que télécharger un fichier binaire à partir d'Internet. On peut trouver la dernière version de BusyBox ici: https://busybox.net/downloads/binaries/.

Télécharger BusyBox, le renommer et lui accorder suffisamment d'autorisations:

```
cd /chroot/busybox/bin/
wget https://busybox.net/downloads/binaries/busybox-x86_64
  --2015-12-09 17:10:52--
https://busybox.net/downloads/binaries/busybox-x86_64
  Resolving busybox.net (busybox.net)... 140.211.167.224
  Connecting to busybox.net (busybox.net)|140.211.167.224|:443... connected.
  HTTP request sent, awaiting response... 200 OK
  Length: 1009384 (986K) [text/plain]
  Saving to: ‘busybox-x86_64’

  100%[=====================================================>] 1,009,384
508KB/s   in 1.9s

  2015-12-09 17:10:56 (508 KB/s) - ‘busybox-x86_64’ saved [1009384/1009384]

  FINISHED --2015-12-09 17:10:56--
  Total wall clock time: 4.4s
  Downloaded: 1 files, 986K in 1.9s (508 KB/s)

mv busybox-x86_64 busybox
chmod 0755 busybox
```

C’est essentiellement tout ce qu’il faut pour «installer» BusyBox.

Configuration de l'utilisateur

Maintenant que Dropbear et BusyBox sont installés dans la prison, il faut configurer un utilisateur qui aura accès à Dropbear. La méthode la plus simple consiste à créer un utilisateur normal avec `useradd`, puis à générer les fichiers nécessaires en fonction des lignes créées par `useradd`. Une autre option consiste à créer et à écrire les fichiers manuellement.

```
useradd test
passwd test
  Changing password for user test.
  New password:
  Retype new password:
  passwd: all authentication tokens updated successfully.

grep ^test /etc/passwd > /chroot/busybox/etc/passwd
grep ^test /etc/group > /chroot/busybox/etc/group
grep ^test /etc/shadow > /chroot/busybox/etc/shadow
cat /chroot/busybox/etc/passwd
  test:x:1001:1001::/home/test:/bin/bash
```

Comme on peut le voir dans la dernière ligne de la sortie ci-dessus. Le fichier `/etc/passwd` de la prison chroot contient un utilisateur, **test**, avec un répertoire personnel `/home/test` et shell `/bin/bash`. Le répertoire personnel n’existe pas dans la prison, pas plus que `/bin/bash`. Alors il faut créer le répertoire personnel:

`mkdir -p /chroot/busybox/home/test chown test:test /chroot/busybox/home/test/`

Il faut remplacer le shell de l'utilisateur par **BusyBox**. Afin d'appeler BusyBox en tant que shell et non pas en tant que commande unique, il est possible de créer une lien symbolique de `/bin /sh` à `/bin/busybox`. Puis on remplace `/bin/bash` par `/bin/sh` pour utiliser ce lien symbolique:

```
pwd

 /chroot/busybox/bin

ln -s busybox sh
ls -l sh

  lrwxrwxrwx. 1 root root 7 Dec  9 20:29 sh -> busybox

sed -i 's/bash/sh/' ../etc/passwd
```

Comme on peut le voir dans la sortie ci-dessus, il est important de créer le lien à partir du chemin correct afin de définir le chemin relatif comme lien pour sh dans `/bin` de la prison chroot.

Démarrer la prison chroot

On peut de démarrer la prison et vérifier si on peut se connecter au port 44 avec SSH.

```
chroot /chroot/busybox/ /sbin/dropbear -d /etc/dropbear/dropbear.dss -r
/etc/dropbear/dropbear.rsa -E -w -g -p 0.0.0.0:44
  [13394] Dec 09 20:36:24 Failed loading /etc/dropbear/dropbear_rsa_host_key
  [13394] Dec 09 20:36:24 Failed loading /etc/dropbear/dropbear_dss_host_key
  [13394] Dec 09 20:36:24 Failed loading
/etc/dropbear/dropbear_ecdsa_host_key
  [13395] Dec 09 20:36:24 Running in background
```

Dans la sortie ci-dessus, on peut voir que Dropbear a démarré, on a explicitement demandé à envoyer des messages à stdout avec l'option -E et à l'écoute sur le port 44.

La prochaine étape est un test sur le port 44 pour le test utilisateur.

```
ssh -p 44 localhost -l test
  [13397] Dec 09 20:36:31 Child connection from 127.0.0.1:48570
  The authenticity of host '[localhost]:44 ([127.0.0.1]:44)' can't be
established.
  RSA key fingerprint is ee:33:a3:1d:7b:69:15:f4:38:25:1b:35:b3:aa:93:82.
  Are you sure you want to continue connecting (yes/no)? yes
  Warning: Permanently added '[localhost]:44' (RSA) to the list of known
hosts.
  test@localhost's password:
  [13397] Dec 09 20:36:35 Password auth succeeded for 'test' from
```

```
127.0.0.1:48570
  ~ $ help
  Built-in commands:
  ------------------
        . : [ [[ alias bg break cd chdir command continue echo eval exec
        exit export false fg getopts hash help history jobs kill let
        local printf pwd read readonly return set shift source test times
        trap true type ulimit umask unalias unset wait

  ~ $ exit
  Connection to localhost closed.
  [13397] Dec 09 20:36:47 Exit (test): Disconnect received
```

On peut voir que la connexion réussit et qu'on se retrouve dans un shell BusyBox.

pour démarrer la prison chroot après un redémarrage, il faut remonter les supports de liaison

```
mount -o bind /dev/pts dev/pts/
mount -o bind /proc proc/
chroot /chroot/busybox/ /sbin/dropbear -d /etc/dropbear/dropbear.dss -r
/etc/dropbear/dropbear.rsa -E -w -g -p 0.0.0.0:44
...
  [13422] Dec 09 20:40:41 Running in background
```

Et spour qu'il démarre automatiquement au démarrage, ajouter simplement les trois commandes ci-dessus à `/etc/rc.d/rc.local`