Guide pratique de chiffrement des systèmes de fichiers racines

Adaptation française du *Encrypted Root Filesystem HOWTO*

Christophe Devine

Adaptation française: Guillaume Lehmann

Relecture de la version française: Claude Thomassin

Préparation de la publication de la v.f.: Jean-Philippe Guérard

Version : 1.1.fr.1.2

2004-10-05

Historique des versions

Version v1.1.fr.1.2	2004-10-05	JPG
Correction orthographique mineure. Passage en DocBook 4.3.		
Version v1.1.fr.1.1	2004-01-11	GL,CT,JPG
Correction orthographique mineure.		
Version v1.1.fr.1.0	2004-01-11	GL,CT,JPG
Adaptation française		
Version v1.1	2003-12-01	CD
Ajout des informations relatives à GRUB (*Added support for GRUB*)		
Version v1.0	2003-09-24	CD
Publication initiale, après validation par le projet de documentation Linux [LDP] (*Initial release, reviewed by LDP*)		
Version v0.9	2003-09-11	CD
Mis à jour et converti en XML DocBook (*Updated and converted to DocBook XML*)		

Ce document explique comment sécuriser vos données personnelles en chiffrant le système de fichiers racine de Linux avec un chiffrement fort.

Table des matières

1. Préparer le système

1.1. Configuration de la disposition de la partition

Votre disque dur (hda) devrait contenir au moins trois partitions :

`hda1` cette petite partition (~ 4 Mo) non-chiffrée demandera un mot de passe afin de monter le système de fichiers racine chiffré.

`hda2` cette partition chiffrée contiendra votre système de fichiers racine ; assurez-vous qu'elle soit suffisamment grande.

`hda3` cette partition contient le vrai système GNU/Linux.

Lorsque vous en serez arrivé là, ni `hda1` ni `hda2` ne seront plus utilisés. `hda3` est l'endroit où votre distribution Linux sera réellement installée ; `/usr` et `/boot` *doivent tous les deux* se trouver sur cette partition.

1.2. Installer Linux-2.4.23

Deux projets principaux travaillent sur l'ajout de capacités de chiffrement fort au noyau : CryptoAPI et loop-AES. Ce guide pratique se base sur loop-AES, car il offre une mise en œuvre de Rijndael en langage assembleur, rapide et très optimisée, et donc des performances maximales aux possesseurs de processeurs IA-32 (x86).

Tout d'abord, téléchargez et décompressez le paquet loop-AES :

```
wget http://loop-aes.sourceforge.net/loop-AES/loop-AES-v2.0d.tar.bz2
tar -xvjf loop-AES-v2.0d.tar.bz2
```

Vous devez aussi télécharger et appliquer un correctif aux sources du noyau :

```
wget http://ftp.kernel.org/pub/linux/kernel/v2.4/linux-2.4.23.tar.bz2
tar -xvjf linux-2.4.23.tar.bz2
cd linux-2.4.23
patch -Np1 -i ../loop-AES-v2.0d/kernel-2.4.23.diff
```

Configurez la carte du clavier :

```
dumpkeys | loadkeys -m - > drivers/char/defkeymap.c
```

Ensuite, configurez votre noyau ; assurez-vous que les options suivantes sont activées :

```
make menuconfig

    Block devices  --->

        <*> Loopback device support
        [*]   AES encrypted loop device support (NEW)

        <*> RAM disk support
        (4096)   Default RAM disk size (NEW)
        [*]   Initial RAM disk (initrd) support

    File systems  --->

        <*> Ext3 journalling file system support
        <*> Second extended fs support

(note importante : n'activez pas le support du système de fichiers /dev)
```

Compilez le noyau et installez-le :

```
make dep bzImage
make modules modules_install
cp arch/i386/boot/bzImage /boot/vmlinuz-2.4.23
```

Si vous utilisez grub comme chargeur de démarrage, mettez à jour le fichier `/boot/grub/menu.lst` ou `/boot/grub/grub.conf` :

```
cat > /boot/grub/menu.lst << EOF
default 0
timeout 10
color green/black light-green/black
title Linux
    root (hd0,2)
    kernel /boot/vmlinuz-2.4.23 ro root=/dev/hda3 vga=4
EOF
```

Sinon, mettez à jour `/etc/lilo.conf` et exécutez la commande **`lilo`** :

```
cat > /etc/lilo.conf << EOF
lba32
boot=/dev/hda
prompt
timeout=100
image=/boot/vmlinuz-2.4.23
    label=Linux
    read-only
    root=/dev/hda3
    vga=4
EOF
lilo
```

Vous devez maintenant redémarrer votre système.

1.3. Installation de util-linux-2.12

Il faut maintenant appliquer un correctif et recompiler le programme **losetup** afin de lui donner la capacité d'utiliser un chiffrement fort. Ce programme est une composante du paquet *util-linux*. Téléchargez, décompressez *util-linux*, puis appliquez-lui le correctif :

```
wget http://ftp.cwi.nl/aeb/util-linux/util-linux-2.12.tar.gz
tar -xvzf util-linux-2.12.tar.gz
cd util-linux-2.12
patch -Np1 -i ../loop-AES-v2.0d/util-linux-2.12.diff
```

Afin de pouvoir utiliser des mots de passes de moins de 20 caractères, entrez :

```
CFLAGS="-O2 -DLOOP_PASSWORD_MIN_LENGTH=8"; export CFLAGS
```

Si la sécurité est importante pour vous, je vous en prie, n'autorisez pas les mots de passes de moins de 20 caractères. La sécurité a un prix, et l'une des contraintes d'une bonne sécurité est la nécessité d'utiliser des mots de passes longs.

Compilez **losetup** et installez-le en tant que super-utilisateur (compte root) :

```
./configure && make lib mount
```

```
cp -f mount/losetup /sbin
rm -f /usr/share/man/man8/losetup.8.gz
cp -f mount/losetup.8 /usr/share/man/man8
```

2. Création du système de fichiers racine chiffré

Remplissez la partition cible avec des données aléatoires :

```
shred -n 1 -v /dev/hda2
```

Configurez le périphérique de bouclage[1] chiffré :

```
losetup -e aes256 -S xxxxxxxxxx /dev/loop0 /dev/hda2
Password:
```

Pour se prémunir contre les attaques par dictionnaire, il est recommandé d'ajouter l'option `-S` *`xxxxxxxxxx`*, où « *`xxxxxxxxxx`* » est votre graine choisie aléatoirement. En outre, afin d'éviter les problème de paramétrage clavier lors du démarrage, n'utilisez que des caractères ASCII (ie pas de caractères accentués, et cætera) dans votre mot de passe.

Maintenant, créez le système de fichiers ext3 :

```
mke2fs -j /dev/loop0
```

Vérifiez que vous avez correctement entré le mot de passe :

```
losetup -d /dev/loop0
losetup -e aes256 -S xxxxxxxxxx /dev/loop0 /dev/hda2
Password:
```

```
mkdir /mnt/efs
mount /dev/loop0 /mnt/efs
```

Vous pouvez comparer les données chiffrées et les données non-chiffrées :

```
xxd /dev/hda2  | less
xxd /dev/loop0 | less
```

C'est le moment d'installer le système Linux chiffré. Si vous utilisez une distribution GNU/Linux (tel que Debian, Slackware, Gentoo, Mandrake, RedHat/Fedora, SuSE, et cætera), exécutez la commande suivante :

```
cp -avx / /mnt/efs
```

Si vous utilisez le livre Linux From Scratch, procédez comme il est indiqué dans le manuel, avec les modifications suivantes :

[1] *Loopback device.*

- Chapitre 6 - Installation de util-linux :

  Appliquez le correctif loop-AES après décompression des sources.

- Chapitre 8 - Rendre amorçable le système LFS :

  Référez-vous à la section suivante.

3. Installation du périphérique d'amorçage

3.1. Création du ramdisque

Pour commencer, redéfinissez la racine du système (via **chroot**) à l'intérieur de la partition chiffrée et créez le point de montage du périphérique d'amorçage :

```
chroot /mnt/efs
mkdir /loader
```

Ensuite, créez le ramdisque initial (initrd), lequel sera nécessaire par la suite :

```
cd
dd if=/dev/zero of=initrd bs=1k count=4096
mke2fs -F initrd
mkdir ramdisk
mount -o loop initrd ramdisk
```

Créez la hiérarchie du système de fichiers et copiez-y les fichiers requis :

```
mkdir ramdisk/{bin,dev,lib,mnt,sbin}
cp /bin/{bash,mount,umount} ramdisk/bin/
ln -s bash ramdisk/bin/sh
mknod -m 600 ramdisk/dev/console c 5 1
mknod -m 600 ramdisk/dev/hda2    b 3 2
mknod -m 600 ramdisk/dev/loop0   b 7 0
cp /lib/{ld-linux.so.2,libc.so.6,libdl.so.2} ramdisk/lib/
cp /lib/{libncurses.so.5,libtermcap.so.2}    ramdisk/lib/
cp /sbin/{losetup,pivot_root} ramdisk/sbin/
```

Ne soyez pas surpris si vous voyez un message du genre « /lib/libncurses.so.5: No such file or directory », ou « /lib/libtermcap.so.2: No such file or directory », c'est normal : bash n'a besoin que de l'une de ces deux bibliothèques. Vous pouvez vérifier laquelle est actuellement nécessaire avec la commande suivante :

```
ldd /bin/bash
```

Créez le script d'initialisation `init` (n'oubliez pas de remplacer « *xxxxxxxxxx* » par la graine que vous avez choisi) :

```
cat > ramdisk/sbin/init << "EOF"
#!/bin/sh

/sbin/losetup -e aes256 -S xxxxxxxxxx /dev/loop0 /dev/hda2
/bin/mount -r -n -t ext2 /dev/loop0 /mnt

while [ $? -ne 0 ]
```

```
do
    /sbin/losetup -d /dev/loop0
    /sbin/losetup -e aes256 -S xxxxxxxxxx /dev/loop0 /dev/hda2
    /bin/mount -r -n -t ext2 /dev/loop0 /mnt
done

cd /mnt
/sbin/pivot_root . loader
exec /usr/sbin/chroot . /sbin/init
EOF

chmod 755 ramdisk/sbin/init
```

Démontez le périphérique de bouclage (*loopback device*) et compressez le fichier initrd :

```
umount -d ramdisk
rmdir ramdisk
gzip initrd
mv initrd.gz /boot/
```

3.2. Démarrage depuis un cd-rom

Je vous conseille vivement de démarrer votre système depuis un média en lecture seule, tel qu'un cd-rom amorçable.

Téléchargez et décompressez syslinux :

```
wget ftp://ftp.kernel.org/pub/linux/utils/boot/syslinux/syslinux-2.07.tar.gz
tar -xvzf syslinux-2.07.tar.gz
```

Configurez isolinux :

```
mkdir bootcd
cp /boot/vmlinuz-2.4.23 bootcd/vmlinuz
cp /boot/initrd.gz syslinux-2.07/isolinux.bin bootcd/
echo "DEFAULT vmlinuz initrd=initrd.gz ro root=/dev/ram0 vga=4" \
    > bootcd/isolinux.cfg
```

Créez et gravez l'image iso sur un cd-rom amorçable :

```
mkisofs -o bootcd.iso -b isolinux.bin -c boot.cat \
        -no-emul-boot -boot-load-size 4 -boot-info-table \
        -J -hide-rr-moved -R bootcd/

cdrecord -dev 0,0,0 -speed 4 -v bootcd.iso

rm -rf bootcd{,.iso}
```

3.3. Démarrer depuis une partition

La partition d'amorçage est un périphérique de démarrage de remplacement : vous en aurez besoin si vous perdez votre CD amorçable. *Rappellez-vous que `hda1` est un média sur lequel il est possible d'écrire et que cela n'est pas sécurisé ; utilisez-le seulement en cas d'urgence !*

Créer et montez le système de fichiers ext2 :

```
dd if=/dev/zero of=/dev/hda1 bs=8192
```

```
mke2fs /dev/hda1
mount /dev/hda1 /loader
```

Copiez le noyau et le ramdisque initial :

```
cp /boot/vmlinuz-2.4.23 /loader/vmlinuz
cp /boot/initrd.gz /loader/
```

Si vous utilisez **grub** :

```
mkdir /loader/boot
cp -av /boot/grub /loader/boot/
cat > /loader/boot/grub/menu.lst << EOF
default 0
timeout 10
color green/black light-green/black
title Linux
    root (hd0,0)
    kernel /vmlinuz ro root=/dev/ram0 vga=4
    initrd /initrd.gz
EOF
grub-install --root-directory=/loader /dev/hda
umount /loader
```

Si vous utilisez **lilo** :

```
mkdir /loader/{boot,dev,etc}
cp /boot/boot.b /loader/boot/
mknod -m 600 /loader/dev/hda  b 3 0
mknod -m 600 /loader/dev/hda1 b 3 1
mknod -m 600 /loader/dev/ram0 b 1 0
cat > /loader/etc/lilo.conf << EOF
lba32
boot=/dev/hda
prompt
timeout=100
image=/vmlinuz
    label=Linux
    initrd=/initrd.gz
    read-only
    root=/dev/ram0
    vga=4
EOF
lilo -r /loader
umount /loader
```

4. Dernières étapes

Modifiez /etc/fstab pour qu'il contienne :

```
/dev/loop0      /      ext3    defaults             0 1
```

Supprimez /etc/mtab et annulez la redéfinition de la racine du système (sortez de **chroot**). Enfin, exécutez **`umount -d /mnt/efs`** et redémarrez. `hda3` n'est désormais plus nécessaire, donc vous pouvez créer un système de fichiers chiffré sur cette partition et l'utiliser comme sauvegarde.

Maintenant, si votre machine n'a pas beaucoup de mémoire, vous aurez besoin d'un peu d'espace d'échange. Supposons que `hda4` contiendra votre partition d'échange chiffrée ; vous devez tout d'abord créer le périphérique d'échange :

```
shred -n 1 -v /dev/hda4
losetup -e aes256 /dev/loop1 /dev/hda4
mkswap /dev/loop1
```

Ensuite, créez un script (`S00swap`) dans le répertoire de démarrage du système (`/etc/rcS.d/` sous Debian) :

```
#!/bin/sh

echo "mot de passe choisi précédemment" | \
    losetup -p 0 -e aes256 /dev/loop1 /dev/hda4
swapon /dev/loop1
```

5. À propos de ce guide pratique

Ce document a tout d'abord été écrit en novembre 2002 pour le projet Linux From Scratch [http://www.fr.linuxfromscratch.org/]. Je voudrai remercier les nombreuses personnes qui m'ont aidé depuis lors à améliorer ce guide pratique (dans l'ordre chronologique inverse) : Julien Perrot, Grant Stephenson, Cary W. Gilmer, James Howells, Pedro Baez, Josh Purinton, Jari Ruusu et Zibeli Aton.

Merci d'envoyer vos commentaires (en anglais) à Christophe Devine [http://www.cr0.net:8040/about/].

La dernière version française [http://www.traduc.org/docs/howto/lecture/Encrypted-Root-Filesystem-HOWTO.html] de ce document est disponible sur le site du projet traduc.org [http://www.traduc.org].

N'hésitez pas à faire parvenir tout commentaire relatif à la version française de ce document à `<commentaires CHEZ traduc POINT org>` en précisant le nom et la version du document auquel vous faites référence.