Titkosított root fájlrendszer HOGYAN

Christophe Devine

Ez a dokumentum leírja, hogyan helyezzük biztonságba adatainkat a Linux root fájlrendszer erős titkosítási algoritmust használó titkosításával.

Tartalomjegyzék

1. A rendszer előkészítése

1.1. A partíciók kialakítása

A lemezünk (hda) legalább három partíciót tartalmazzon:

- hda1: ez a kicsi (~4 Mb), nem titkosított partíció fogja kérni a jelszavunkat a titkosított root fájlrendszer felcsatolásához.
- hda2: ez a partíció tartalmazza a titkosított root fájlrendszert; legyen megfelelően nagy.
- hda3: ez a partíció tartalmazza az aktuális GNU/Linux rendszert.

Ekkor még a hda1 és a hda2 partíciók nincsenek használatban. A hda3 partíció tartalmazza az aktuálisan telepített Linux terjesztést; az /usr és a /boot *nem* lehet ettől a partíciótól elkülönítve.

1.2. A Linux-2.4.23 rendszermag telepítése

Két nagyobb projekt létezik, ami erős titkosítási támogatást ad a rendszermaghoz: a CryptoAPI és a loop-AES. Ez a HOGYAN a loop-AES projekten alapul, mivel ez egy assembly nyelven írt, nagyon gyors és optimalizált implementáció, így maximális teljesítményt nyújt az IA-32 (x86) alapú processzorok számára. Készítője Rijndael.

Mindenek előtt töltsük le, majd csomagoljuk ki a loop-AES csomagot:

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

Ezután töltsük le a rendszermag forrását, majd alkalmazzuk a foltot:

```
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-v2.0b/kernel-2.4.23.diff
```

Állítsuk be a billentyűzet kiosztását:

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

A következő lépésben beállítjuk a rendszermagot; a következő lehetőségek mindenképp legyenek beállítva:

```
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
```

(fontos: ne legyen beállítva a /dev file system support lehetőség)

Fordítsuk le és telepítsük a rendszermagot:

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

Ha a grub rendszerbetöltőt használjuk, szerkesszük a /boot/grub/menu.lst illetve /boot/grub/grub.conf fájlt:

```
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
```

Ha viszont lilo-t használunk, akkor szerkesszük az /etc/lilo.conf fájlt, és futtassuk a lilo-t:

```
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
```

Indítsuk újra a rendszert.

1.3. Az util-linux-2.12 telepítése

A losetup programon - amely az util-linux csomag része - alkalmaznunk kell a foltot, és újra kell fordítanunk az erős titkosítás támogatásához. Töltsük le és csomagoljuk ki az util-linux csomagot, majd alkalmazzuk a foltot:

```
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-v2.0b/util-linux-2.12.diff
```

20 karakternél rövidebb jelszó használata esetén írjuk be:

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

Ha fontos kérdés a biztonság, ne használjuk ezt a lehetőséget. A biztonságnak megvan az ára, jelen esetben ez a hosszú jelszó használata.

Fordítsuk le a losetup-ot, majd root felhasználóként telepítsük azt:

```
./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. A titkosított root fájlrendszer létrehozása

A célpartíció feltöltése véletlenszerű adattal:

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

A titkosított loopback eszköz beállítása:

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

A szótárral optimalizált támadások kivédése érdekében ajánlott a -S xxxxxxxxxx opció használata, ahol "xxxxxxxxxx" a véletlenszerűen kiválasztott álvéletlen sorozat kiindulóérték (random seed). Ezen kívül a rendszerindításkor esetleg fellépő billentyűzetkiosztási hibák megelőzése érdekében ne használjunk nem-ASCII karaktereket (ékezeteket stb.) a jelszóban.

Hozzuk létre az ext3 fájlrendszert:

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

Ellenőrizzük, hogy helyesen írtuk be a jelszót:

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

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

Összehasonlíthatjuk a titkosított és az eredeti adatokat:

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

Itt az ideje, hogy telepítsük a titkosított Linux fájlrendszert. Ha egy GNU/Linux terjesztést használunk (például Debian-t, Slackware-t, Gentoo-t, Mandrake-et, RedHat/Fedora-t, SuSE-t stb.), futtassuk a következő parancsot:

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

Ha a Linux From Scratch könyvet használjuk, az alábbi eltéréseket kivéve a dokumentum szerint folytathatjuk a munkát:

- 6. fejezet - Az util-linux telepítése:

  Alkalmazzuk a loop-AES foltot a forrás kicsomagolása után.

- 8. fejezet - Az LFS rendszer indíthatóvá tétele:

  Szorítkozzunk a következő fejezetre.

3. A boot eszköz beállítása

3.1. A virtuális fájlrendszer (ramdisk) létrehozása

Első lépésként chroot-oljunk a titkosított partícióra, és hozzuk létre a boot eszközhöz a felcsatolási pontot:

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

Ezután hozzuk létre a virtuális rendszerindító fájlrendszert (initial ramdisk, initrd), amelyre később szükségünk lesz:

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

Hozzuk létre a fájlrendszer könyvtárszerkezetét, és másoljuk be a szükséges fájlokat:

```
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/
```

Ha a következő vagy hasonló hibaüzenetet kapjuk, az nem jelent problémát: "/lib/libncurses.so.5: No such file or directory", vagy "/lib/libtermcap.so.2: No such file or directory"; a bash-nek csak ezen programkönyvtárak egyike szükséges. Megtudhatjuk azt, hogy esetünkben melyik szükséges:

```
ldd /bin/bash
```

Hozzuk létre a rendszerindító (init) szkriptet (ne felejtsük a "xxxxxxxxxx" helyére beírni a kiválasztott álvéletlen sorozat kiindulóértéket (random seed)):

```
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
```

Csatoljuk le a loopback eszközt, és tömörítsük be a virtuális rendszerindító fájlrendszert:

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

3.2. Rendszerindítás CD-ROM-ról

Erősen ajánlott a rendszert egy írásvédett eszközről indítani, például egy indítható CD-ROM-ról.

Töltsük le, majd csomagoljuk ki a syslinux csomagot:

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

Állítsuk be az isolinux-ot:

```
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
```

Hozzuk létre az indítható cd-képet (cd image), és írjuk ki egy írható cd-re:

```
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. Rendszerindítás fizikai partícióról

A boot partíció egy alternatív rendszerindító eszköz: szükség lehet rá, ha az indítható CD elvész. *Vegyük figyelembe, hogy a hda1 egy írható eszköz, ezért nem biztonságos; csak szükség esetén használjuk!*

Hozzuk létre és csatoljuk fel az ext2 fájlrendszert:

```
dd if=/dev/zero of=/dev/hda1 bs=8192
mke2fs /dev/hda1
mount /dev/hda1 /loader
```

Másoljuk át a rendszermagot és a virtuális rendszerindító fájlrendszert:

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

Ha grub-ot használunk:

```
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
```

Ha lilo-t használunk:

```
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. Utolsó lépések

Módosítsuk az /etc/fstab fájlt úgy, hogy tartalmazza a következő sort:

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

Töröljük az /etc/mtab fájlt, és lépjünk ki a chroot-ból. Legvégül futtassuk a "umount -d /mnt/efs" parancsot, majd indítsuk újra a rendszert. A hda3-ra már nincs szükség, létrehozhatunk egy titkosított fájlrendszert ezen a partíción, és használhatjuk biztonsági mentésként.

Ha kevés a fizikai memóriánk, szükség lesz swap területre. Tételezzük fel, hogy a hda4 fogja tartalmazni a titkosított swap partíciót; először létre kell hozni a swap eszközt:

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

Majd hozzunk létre egy indítószkriptet (S00swap) a rendszer indítókönyvtárában (/etc/rcS.d/ Debian esetén):

```
#!/bin/sh

echo "az előzőleg kiválasztott jelszó" | \
    losetup -p 0 -e aes256 /dev/loop1 /dev/hda4
swapon /dev/loop1
```

5. A HOGYANról

A Titkosított root fájlrendszer HOGYAN 2002 novemberében készült el a Linux From Scratch (http://www.linuxfromscratch.org/lfs/news.html) projekt részére. Köszönet mindazoknak, akik azóta segítettek a HOGYAN tökéletesítésében (fordított időrendben): Julien Perrot, Grant Stephenson, Cary W. Gilmer, James Howells, Pedro Baez, Josh Purinton, Jari Ruusu és Zibeli Aton.

A hozzászólásokat Christophe Devine (http://www.cr0.net:8040/about/) várja.

5.1. Magyar fordítás

A magyar fordítást Vadon Péter (mailto:vape[kukac]maffia[pont]hu) készítette (2004.06.15). A lektorálást Daczi László (mailto:dacas[kukac]freemail[pont]hu) végezte el (2004.06.24). A dokumentum legfrissebb változata megtalálható a Magyar Linux Dokumentációs Projekt (http://tldp.fsf.hu/) honlapján.