

# Titkosított root fájlrendszer HOGYAN

Christophe Devine

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

## Tartalomjegyzék

<b>1. A rendszer előkészítése .....</b>	<b>2</b>
1.1. A partíciók kialakítása.....	2
1.2. A Linux-2.4.27 rendszermag telepítése.....	2
1.3. A Linux-2.6.8.1 rendszermag telepítése.....	4
1.4. Az util-linux-2.12b telepítése .....	4
<b>2. A titkosított root fájlrendszer létrehozása .....</b>	<b>5</b>
<b>3. A boot eszköz beállítása.....</b>	<b>6</b>
3.1. A virtuális fájlrendszer (ramdisk) létrehozása .....	6
3.2. Rendszerindítás CD-ROM-ról.....	8
3.3. Rendszerindítás merevlemez partícióról .....	8
<b>4. Utolsó lépések .....</b>	<b>9</b>
<b>5. A HOGYANról .....</b>	<b>11</b>
5.1. Magyar fordítás .....	11

# 1. A rendszer előkészítése

## 1.1. A partíciók kialakítása

A lemeziünk (hda) legalább három partíciót tartalmazzon:

- hda1: ez a kicsi, 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.

me egy példa, amely bemutatja milyen partícióknak kell lennie:

```
# fdisk -l /dev/hda
```

```
Disk /dev/hda: 255 heads, 63 sectors, 2432 cylinders  
Units = cylinders of 16065 * 512 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/hda1		1	1	8001	83	Linux
/dev/hda2		2	263	2104515	83	Linux
/dev/hda3		264	525	2104515	83	Linux
/dev/hda4		526	2047	12225465	83	Linux

## 1.2. A Linux-2.4.27 rendszermag telepítése

Két nagyobb projekt létezik, ami visszacsatolt titkosítási támogatást (loopback encryption support) ad a rendszermaghoz: cryptoloop és 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. Mindezek mellett van néhány aggasztó biztonsági kérdés (<http://groups.google.fr/groups?selm=1emrG-1Ck-25%40gated-at.bofh.it>) a cryptoloop-pal kapcsolatban.

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

```
wget http://loop-aes.sourceforge.net/loop-AES/loop-AES-v2.2b.tar.bz2  
tar -xvjf loop-AES-v2.2b.tar.bz2
```

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

```
wget http://ftp.kernel.org/pub/linux/kernel/v2.4/linux-2.4.27.tar.bz2  
tar -xvjf linux-2.4.27.tar.bz2  
cd linux-2.4.27
```

```
patch -Np1 -i ../loop-AES-v2.2b/kernel-2.4.27.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
```

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 ro root=/dev/hda3
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
    label=Linux
    read-only
    root=/dev/hda3
EOF
lilo
```

Indítsuk újra a rendszert.

### 1.3. A Linux-2.6.8.1 rendszermag telepítése

Az előző fejezetben leírtakhoz hasonlóan alkalmazzuk a loop-aes' *kernel-2.6.8.1.diff* foltot. Figyeljünk arra, hogy a modul támogatás szükséges, ha a module-init-tools csomag telepítve van.

### 1.4. Az util-linux-2.12b 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.kernel.org/pub/linux/utils/util-linux/util-linux-2.12b.tar.bz2
tar -xvjf util-linux-2.12b.tar.bz2
cd util-linux-2.12b
patch -Np1 -i ../loop-AES-v2.2b/util-linux-2.12c.diff
```

20 karakternél rövidebb jelszó használata esetén írjuk be:

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

A biztonság talán a legfontosabb kérdések egyike, ezért ne engedélyezzünk 20 karakternél rövidebb jelszavakat. 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
mv -f /sbin/losetup /sbin/losetup~
rm -f /usr/share/man/man8/losetup.8*
cd mount
gzip losetup.8
cp losetup /sbin
cp losetup.8.gz /usr/share/man/man8/
```

## 2. A titkosított root fájlrendszer létrehozása

A célpártíció feltöltése véletlenszerű adattal:

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

A titkosított loopback eszköz beállítása:

```
losetup -e aes256 -S xxxxxx /dev/loop0 /dev/hda2
```

A szótárral optimalizált támadások kivédése érdekében ajánlott a `-S xxxxxx` opció használata, ahol "xxxxxx" a véletlenszerűen kiválasztott álvéletlen sorozat kiindulóérték (random seed) (ez például lehet "gPk4lA"). Ezen kívül a rendszerindításkor esetleg fellépő billentyűzetkiosztási hibák megelőzése érdekében ne használjunk nem-ASCII karaktereket (ékezeteket stb.) a jelszóban. A Diceware (<http://www.diceware.com/>) webhely egy egyszerű módszert ajánl az erős és könnyen megjegyezhető jelszó létrehozására.

Hozzuk létre az ext3 fájlrendszert:

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

Ellenőrizzük, hogy helyesen írtuk be a jelszót:

```
losetup -d /dev/loop0  
losetup -e aes256 -S xxxxxx /dev/loop0 /dev/hda2
```

```
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:  
A következő fejezetben olvashatunk erről (A boot eszköz beállítása).

## 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
```

Ha a grsecurity-t használod, akkor "Permission denied" hibüzenetet kaphatsz; ebben az esetben a mount parancsot a chroot-on kívülről futtasd.

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

```
mkdir ramdisk/{bin,dev,lib,mnt,sbin}  
cp /bin/{bash,mount} 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-nak csak ezen programkönyvtárak egyike szükséges. Megtudhatjuk azt, hogy esetünkben melyik szükséges:

```
ldd /bin/bash
```

Fordítsuk le a sleep programot, amely majd megelőzi azt, hogy a jelszó bekérését "kinyomják" a képernyőről a rendszermag üzenetei (mint például az usb eszközök regisztrálása).

```
cat > sleep.c << "EOF"
#include <unistd.h>
#include <stdlib.h>

int main( int argc, char *argv[] )
{
    if( argc == 2 )
        sleep( atoi( argv[1] ) );
    return( 0 );
}
EOF

gcc -s sleep.c -o ramdisk/bin/sleep
rm sleep.c
```

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

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

/bin/sleep 3
/sbin/losetup -e aes256 -S xxxxxx /dev/loop0 /dev/hda2
/bin/mount -r -n -t ext3 /dev/loop0 /mnt

while [ $? -ne 0 ]
do
    /sbin/losetup -d /dev/loop0
    /sbin/losetup -e aes256 -S xxxxxx /dev/loop0 /dev/hda2
    /bin/mount -r -n -t ext3 /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
mkdir ramdisk
gzip initrd
mv initrd.gz /boot/
```

### 3.2. Rendszerindítás CD-ROM-ról

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

Töltsük le, majd csomagoljuk ki a syslinux csomagot:

```
wget http://ftp.kernel.org/pub/linux/utils/boot/syslinux/syslinux-2.10.tar.bz2
tar -xvjf syslinux-2.10.tar.bz2
```

Állítsuk be az isolinux-ot:

```
mkdir bootcd
cp /boot/{vmlinuz,initrd.gz} syslinux-2.10/isolinux.bin bootcd
echo "DEFAULT /vmlinuz initrd=initrd.gz ro root=/dev/ram0" \
> 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 merevlemez partícióról

A boot partíció jól jön, ha elveszítjük az indítható CD lemezt. *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,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
    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
EOF
lilo -r /loader
umount /loader
```

## 4. Utolsó lépések

Még mindig a chroot-ban, módosítsuk az /etc/fstab fájlt úgy, hogy tartalmazza a következő sort:

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

Töröljük az /etc/mtab fájlt, és lépünk ki a chroot-ból. Legvégül futtassuk a "umount -d /mnt/efs" parancsot, majd indítsuk újra a rendszert. Ha valami baj van, akkor még mindig el lehet indítani a nem titkosított partíción lévő rendszert, ha a LILO promptba begépeljük a "Linux root=/dev/hda3" parancsot.

Ha minden jól ment, akkor most újraparticionálhatjuk a hda3 partíciót, csakúgy, mint a hda4-et. A következő szkriptekben feltételezzük azt, hogy a hda3 partíción lesz a swap terület, a hda4-en pedig a /home, először ezt a két partíciót kell inicializálni (alapállapotba hozni - a lektor):

```
shred -n 1 -v /dev/hda3
shred -n 1 -v /dev/hda4
losetup -e aes256 -S xxxxxx /dev/loop1 /dev/hda3
losetup -e aes256 -S xxxxxx /dev/loop2 /dev/hda4
mkswap /dev/loop1
mke2fs -j /dev/loop2
```

Majd hozzunk létre egy indító szkriptet a rendszer indító könyvtárában, és frissítsük az fstab-ot:

```
cat > /etc/init.d/loop << "EOF"
#!/bin/sh

if [ "`/usr/bin/md5sum /dev/hda1`" != \
    "5671cebdb3bed87c3b3c345f0101d016  /dev/hda1" ]
then
    echo -n "FIGYELEM! A hda1 partíció sértetlenségének vizsgálata HIBÁT jelzett - üss enter-t."
    read
fi

echo "1st password chosen above" | \
    /sbin/losetup -p 0 -e aes256 -S xxxxxx /dev/loop1 /dev/hda3

echo "2nd password chosen above" | \
    /sbin/losetup -p 0 -e aes256 -S xxxxxx /dev/loop2 /dev/hda4

/sbin/swapon /dev/loop1

for i in `seq 0 63`
do
    echo -n -e "\33[10;10]\33[11;10]" > /dev/tty$i
done

EOF

chmod 700 /etc/init.d/loop
```

```
ln -s ../init.d/loop /etc/rcS.d/S00loop
vi /etc/fstab
...
/dev/loop2      /home          ext3           defaults      0             2
```

## 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 dokumentum tökéletesítésében (fordított időrendi sorrendben): Luc Vo Van, Jacobus Brink, Ernesto Pérez Estévez, Matthew Ploessel, Mike Lorek, Lars Bungum, Michael Shields, Julien Perrot, Grant Stephenson, Cary W. Gilmer, James Howells, Pedro Baez, Josh Purinton, Jari Ruusu és Zibeli Aton.

Ez a HOGYAN a következő nyelvekre van lefordítva:

- francia (<http://www.traduc.org/docs/HOWTO/lecture/Encrypted-Root-Filesystem-HOWTO.html>)
- olasz (<http://www.linux.it/~gaetano/erfs/>)
- magyar (<http://tldp.fsf.hu/HOWTO/Encrypted-Root-Filesystem-HOWTO-hu/>)

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](mailto:vape[kukac]maffia[pont]hu)) készítette (2004.06.15). A lektorálást Daczi László ([mailto:dacas\[kukac\]freemail\[pont\]hu](mailto:dacas[kukac]freemail[pont]hu)) végezte el (2004.06.24). Utoljára frissítve 2004.10.21.-én. A dokumentum legfrissebb változata megtalálható a Magyar Linux Dokumentációs Projekt (<http://tldp.fsf.hu/>) honlapján.