Encrypted Root Filesystem HOWTO

Christophe Devine

Revision History

Revision v1.2	2004-08-01	Revised by: cd
Updated the packages version.		
Revision v1.1	2003-12-01	Revised by: cd
Added support for GRUB.		
Revision v1.0	2003-09-24	Revised by: cd
Initial release, reviewed by LDP.		
Revision v0.9	2003-09-11	Revised by: cd
Updated and converted to DocBook XML.		

Questo documento spiega come rendere i vostri dati personali sicuri cifrando il vostro linux root filesystem con algoritmi di crittografia forte.

Preparazione del sistema

Settare il layout delle partizioni.

Il vostro hard disk (hda) dovrebbe contenere almeno tre partizioni:

- hda1: questa piccola (circa 8 mb) partizione non crittata ti chiedera' la password per montare la partizione con il filesystem di root criptato.
- hda2: questa partizione conterrà il tuo file system di root criptato; stai attento che sia abbastanza capiente.
- hda3: questa partizione conterra' il tuo sistema Gnu\Linux.

A questo punto, sia hda1 e hda2 sono inutilizzati. In hda3 si trova installata la tua distribuzione linux; /usr e /boot non devono essere separate da questa partizione.

Qui trovi un esempio di come potrebbe assomigliare il tuo schema di partizionamento:

```
# 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
```

Installare Linux-2.4.27

Ci sono due progetti principali che aggiungono un supporto di crittografia forte al kernel: CryptoAPI e loop-AES. Questo how-to e' basato su loop-aes, poiché ha un'esecuzione estremamente veloce ed altamente ottimizzata di Rijndael in linguaggio assembler e quindi fornisce le prestazioni massime su una CPU IA-32 (x86).

Prima di tutto, scarichiamo e scompattiamo il pacchetto per loop-aes:

```
wget http://loop-aes.sourceforge.net/loop-AES/loop-AES-v2.1b.tar.bz2
tar -xvjf loop-AES-v2.1b.tar.bz2
```

Dopo devi scaricare i sorgenti del kernel ed applicarvi la patch:

```
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
rm include/linux/loop.h drivers/block/loop.c
patch -Np1 -i ../loop-AES-v2.1b/kernel-2.4.26.diff
```

Setta la mappa per la tastiera:

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

Dopo, configura il tuo kernel; poni attenzione che queste opzioni sia attivate:

```
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

(important note: do not enable /dev file system
support)
```

Compila il kernel ed installalo:

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

Se usi grub come bootloader, aggiorna /boot/grub/menu.lst e /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 ro root=/dev/hda3
EOF
```

Altrimenti, aggiorna /etc/lilo.conf e fai partire lilo :

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

A questo punto riavvia il tuo sistema.

Installare Linux-2.6.7

Procedi come descritto nella sezione precedente, usando invece la patch *kernel-2.6.6.diff* di loop-aes. Nota che il supporto per il moduli richiede che tu abbia il pacchetto module-init-tools installato.

Installare le util-linux-2.12.

Il programma losetup, che e' parte del pacchetto util-linux, deve essere patchato e ricompilato per aggiungere il supporto per la crittografia forte.

Scarica, scompatta e applica la patch alle util-linux :

```
wget http://ftp.kernel.org/pub/linux/utils/util-
linux/util-linux-2.12a.tar.bz2
tar -xvjf util-linux-2.12a.tar.bz2
cd util-linux-2.12a
patch -Np1 -i ../loop-AES-v2.1b/util-linux-2.12a.diff
```

Per usare password piu' corte di 20 caratteri, digita :

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

Se per te la sicurezza e' importante, non abilitare password piu' corte di 20 caratteri. La sicurezza non e' gratis, si deve pagare in termini di lunghezza della password.

Compila losetup e installalo come utente root:

```
./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/
```

Creazione del filesystem root crittato

Riempi la partizione che ospitera' il filesystem root criptato con dati casuali:

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

Iniziallizza il loopback device criptato:

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

Per prevenire attacchi ottimizzati a dizionario, e' raccomandato di aggiungere l'opzione -S xxxxxx, dove xxxxxx e' il tuo seme scelto in modo casuale (pe esempio, potreste scegliere "gPk4lA0v"). Inoltre, per evitare problemi in fase di boot con la mappa della tastiera, non usare caratteri non-ascii (accenti, etc..) per la tua password. Il sito Diceware ti offre un semplice modo per creare forti, ancora facili da ricordare, passphrases.

Adesso crea il filesystem ext3:

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

Controlla che hai inserito correttemente la password:

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

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

Puoi confrontare i dati crittati con quelli non crittati:

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

E' il momento di installare il tuo filesystem linux criptato. Se usi una distribuzione Gnu/Linux (come sono Debian, Slackware, Gentoo, Mandrake, RedHat/Fedora, SuSe, etc..) fallo con i seguenti comandi :

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

Se usi il libro di Linux From Scrath , procedi come descritto sul manuale, con le seguenti differenze:

- Capitolo 6 - Installare util-linux:

  Applicare la patch loop-AES dopo aver scompattato i sorgenti.

- Capitolo 8 - Rendere il sistema LFS avviabile:

  Si faccia riferimento al prossimo paragrafo.

Configurare il dispositivo di avvio

Creazione del ramdisk

Per iniziare, esegui chroot dentro la partizione criptata e crea il mount point per il dispositivo di boot:

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

Dopo, crea il ramdisk iniziale (initrd), che sara' necessario in seguito:

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

Se state usando le grsecurity, potete ottenere un messaggio di errore di “Permission denied”; in questo caso dovrete eseguire il comando mount fuori dalla chroot.

Crea la gerarchia del filesystem e copia i file richiesti su di essi :

```
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/
```

E' tutto ok se vedrai il messaggio "/lib/libncurses.so.5: No such file or directory", o "/lib/libtermcap.so.2: No such file or directory"; la bash richiede solo uno di queste due librerie. Puoi controllare manualmente quale delle due e' necessaria con:

```
ldd /bin/bash
```

Compile the sleep program, which will prevent the password prompt being flooded by kernel messages (such as usb devices being registered).

Compila il programma sleep, che impedira' che il prompt della richiesta della password venga floodato dai messaggi del kernel (come quando un dispositivo usb viene registrato)

```
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
```

Crea lo script di init (senza dimenticare di sostituire “xxxxxxx” con il seme scelto):

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

/bin/sleep 3
/sbin/losetup -e aes256 -S xxxxxxxx /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 xxxxxxxx /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
```

Smonta il dispositivo di loopback e comprimi l'initrd:

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

Avvio da un CD-ROM

Si consiglia fortemente di avviare il tuo sistema con un dispositivo a sola lettura, come un cdrom avviabile.

Scarica e scompatta syslinux:

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

Configura isolinux:

```
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
```

Crea e masterizza l'immagine bootabile su cd-rom :

```
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}
```

Avviare il sistema da una partizione

La partizione di avvio puo' essere utile se il tuo CD di avvio va perso. Ricordati che hda1 e' un dispositivo scrivibile e quindi insicuro; usalo solo in caso di emergenza!

Crea e monta il filesystem ext2:

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

Copia il kernel e il ramdisk iniziale:

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

Se usi 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
```

Se usi 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
```

Passi finali

Modifica /etc/fstab in modo che contenga:

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

Rimuovi /etc/mtab ed esci dalla chroot. Per concludere, lancia "umount - la d/mnt/efs" e riavvia. Se tutto andasse bene, puoi criptare sia hda3 e hda4; diciamo che hda3 conterra' il tuo dispositivo di swap e hda4 conterra' la /home. Dovresti inizializzare ambedue le partizioni prima:

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

Dopodiche' crea uno script nella directory di avvio del sistema e aggiorna fstab :

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

if [ "`/usr/bin/md5sum /dev/hda1`" != \
     "5671cebdb3bed87c3b3c345f0101d016  /dev/hda1" ]
then
    echo -n "WARNING! hda1 integrity verification FAILED - press
enter."
    read
fi

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

echo "2nd password chosen above" | \
    /sbin/losetup -p 0 -e aes256 -S xxxxxxxx /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
```

A proposito di questo HOWTO

L' Encrypted Root Filesystem HOWTO fu scritto per la prima volta nel novembre del 2002 per il progetto Linux From Scratch . Vorrei ringraziare molte persone che mi hanno aiutato a migliorare questo howto (in ordine cronologico inverso): 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 and Zibeli Aton.

Questo HOWTO e' stato tradotto in svariate lingue:

- French
- Italian
- Hungarian

Perpiacere manda i tuoi commenti a Christophe Devine.

Note per la traduzione italiana

Questo howto e' stato tradotto in italiano da Gaetano Zappulla <gaetano(at)linux.it>.
Potete trovare la sua versione piu' aggiornata all'indirizzo www.linux.it/~gaetano/erfs.

Revision 1.0 terminata di tradurre il 29 Novembre 2003.

Revision 1.1 terminata di tradurre il 3 Dicembre 2003.

Revision 1.2 terminata di tradurre il 13 Agosto 2004.