Slackware: Partizioni Crittate

a cura di HelLViS69
hellvis69@netsons.org

Indice

1. Obiettivi

Questo howto si prefigge come obiettivo la crittazione delle partizioni /home e swap usando dm-crypt e chiave di cifratura su pendrive usb.

Si presume che il sistema abbia / e /home su due partizioni distinte. E' il sistema che sto usando io, testato su Slackware 10.2 e 11.0.

Usando Slackware 12.0 non si ha la necessità di installare i software di cui sotto in quanto già presenti nella distribuzione; bisogna però modificare gli script di avvio in quanto non prevedono l'uso delle chiavi su device esterni con pendrive usb come nel nostro caso.

Non mi prendo alcuna responsabilità su eventuali danni o perdite di dati

2. Background

Il *Device Mapper* è un framework generico per mappare delle periferiche a blocchi in altre. Questo sistema sta alla base di cryptoloop, per l'appunto, *LVM2* e il sistema di snapshot del filesystem.

Le applicazioni che usano questo sistema scambiano informazioni coi Device Mapper attraverso la libreria *libdevmapper.so* che solleva le chiamate di sistema (ioctl) all'i-node della periferica.

3. Preparazione dell'ambiente

Armatevi di pazienza, voglia di fare, qualche sigaretta e una birra :)

3.1. Preparazione del kernel

La prima cosa da fare per poter utilizzare Device Mapper è abilitarlo nel kernel. Consiglio di metterlo builtin (*) e non modulo (M) altrimenti ci toccherà preparare un initrd contenente il modulo.

Per abilitare il supporto, assicuratevi di avere i sorgenti del kernel ed entrate nella configurazione seguendo i seguenti passi:

```
# cd /usr/src/linux
# make menuconfig
```

Ora cercate la seguente sezione:

```
Device Drivers
  --> [*] Multi-device support (RAID and LVM)
     --> [*] Device Mapper
        --> [*] Crypt Target Support
```

Aspettate a ricompilare perché potreste aver bisogno di patchare il kernel a seconda della versione che state utilizzando.

3.2. Patchare il kernel

Se usate un kernel dalla versione **2.6.18** in poi, consiglio vivamente di applicare la patch sottostante, altrimenti all'avvio della macchina rischiate che il vostro pendrive non sia ancora stato rilevato dal kernel causando la non lettura della chiave di crittazione e quindi le vostre partizioni crittate **NON** verranno montate.

Applicare la patch è molto semplice: basta infatti modificare il delay nel rilevamento delle periferiche usb.

1. Editate il file **drivers/usb/storage/usb.c**
2. cercate la linea **static unsigned int delay_use = 5;** e modificate il valore a 0
3. salvate

3.3. Software

I seguenti software vanno installati in ordine, altrimenti incapperete sicuramente in qualche errore di dipendenza.

1. *libgpg-error*, che contiene i codici di errore per la gestione di funzioni usate da GnuPG
2. *libgcrypt*, libreria crittografica che fornisce funzioni per la crittografia: cifratura simmetrica (AES, DES, Blowfish, CAST5, Twofish, Arcfour), algoritmi di hashing (MD4, MD5, RIPE-MD160, SHA-1, TIGER-192), MAC (HMAC per tutti gli algoritmi di hash), algoritmi a chiave pubblica (RSA, ElGamal, DSA), funzioni a grandi interi, a numeri casuali e tutte le funzioni di supporto per essi
3. *luks*, *Linux Unified Key* Setup, tool per la creazione di partizioni crittate

4. Il lavoro sporco

Ma veniamo alla parte pratica di questa documentazione

4.1. Creare le partizioni

Per prima cosa dobbiamo montare il nostro pendrive. Ad esempio, se in /etc/fstab fosse mappato su /mnt/pendrive, il comando sarebbe

```
# mount /mnt/pendrive
```

Successivamente andremo a creare la chiave di crittazione a 4096 KByte (4MB), viva la sicurezza :)

```
# dd if=/dev/urandom of=/mnt/pendrive/keyfile bs=1024k count=4
```

dove keyfile sarà la nostra chiave, da usare per decrittare le partizioni all'avvio della macchina e successivamente recrittarle allo spegnimento. Qualcuno si chiederà perché fare 4 blocchi da 1024k invece che 1 da 4096. Semplice: in questo modo aumenta l'entropia della chiave ;)

Ora dobbiamo creare la prima partizione crittata, quella che diverrà la nostra home

```
# cryptsetup -v -c aes -d /mnt/pendrive/keyfile create
secret /dev/hdaX
```

dove:

`cryptsetup` - nome del programma
`-v` - verbose
`-c aes` - algoritmo di cifratura
`-d /dir` - chiave di crittazione
`create` - direttiva che dice di creare il nuovo device
`secret` - nome significativo del device
`/dev/hdaX` - device reale, dove X sta per la vostra partizione /home

Ora bisogna formattare la partizione crittata col filesystem che volete, io ho scelto ReiserFS. Questa operazione si effettua col comando:

```
# mkfs.reiserfs /dev/mapper/secret
```

Controlliamo che tutto sia andato a buon fine:

```
# ls -la /dev/mapper/
total 62
drwxr-xr-x  2 root root     120 2006-07-29 23:02 ./
drwxr-xr-x 20 root root   63624 2007-01-25 02:06 ../
crw-------  1 root root  10, 63 2006-07-29 23:02 control
brw-------  1 root root 254,  1 2006-03-06 23:33 secret
```

4.2. Crittare la /home

Ora come ora abbiamo solo creato una partizione crittata di nome *secret* che punta a */dev/hdaX*. Il vero lavoro di crittazione della /home avviene coi 2 semplici comandi che seguono:

montaggio della partizione crittata su un mount-point disponibile, esempio /mnt/hd

```
# mount /dev/mapper/secret /mnt/hd
```

e copia della home sulla partizione crittata

```
# cp -avx /home/* /mnt/hd
```

4.3. Crittare la swap

Vi chiederete perché. La swap può contenere dati sensibili e informazioni preziose.

Siccome siamo paranoici, crittiamo anche quello, col comando che segue:

```
# cryptsetup -d /dev/random create swapfs /dev/hdaX
```

dove:

`cryptsetup` - nome del programma
`-d /dir` - chiave di crittazione generata casualmente
`create` - direttiva che dice di creare il nuovo device
`swapfs` - nome significativo del device
`/dev/hdaX` - device reale, dove X sta per la vostra partizione di swap

Ora bisogna formattare la partizione crittata come swap, quindi procederemo col seguente comando:

```
# mkswap /dev/mapper/swapfs
```

Non sognatevi di attivare ora la swap!

Se lo fate, il sistema quasi sicuramente si incarterà e non swapperà più fino al prossimo riavvio!

Controlliamo che tutto sia andato a buon fine:

```
# ls -la /dev/mapper/
total 62
drwxr-xr-x  2 root root     120 2006-07-29 23:02 ./
drwxr-xr-x 20 root root   63624 2007-01-25 02:06 ../
crw-------  1 root root  10, 63 2006-07-29 23:02 control
brw-------  1 root root 254,  1 2006-03-06 23:33 secret
brw-------  1 root root 254,  0 2006-03-06 23:11 swapfs
```

Notate la differenza del device tra la home e la swap

5. Automatizzazioni

5.1 Automount

Il prossimo passo è mappare le nostre nuove partizioni in /etc/fstab. Editate il file, eliminate le vecchie righe per il mount della /home e della swap, e sostituitele con queste:

```
/dev/mapper/swapfs   none     swap        sw,pri=1
/dev/mapper/secret   /home    reiserfs    defaults      0   0
```

5.2. Script di avvio di Slackware

Ora bisogna dire alla nostra Slackware di montare la swap crittata, invece di attivare quella creata durante l'installazione del sistema, e di montare la nostra home crittata. Per fare ciò, bisogna editare il file **/etc/rc.d/rc.S**, lo script di startup appunto.

1. cercate le linee:
   ```
   # Enable swapping:
   /sbin/swapon -a
   ```
2. e modificatele come segue:
   ```
   # Enable swapping (with crypted swap):
   /usr/sbin/cryptsetup -d /dev/random create swapfs /dev/hdaX
   /sbin/mkswap /dev/mapper/swapfs
   /sbin/swapon -a
   ```

5.2.1 Slackware 10.2

1. cercate le righe:
   ```
   # mount non-root file systems in fstab (but not NFS or SMB
   # because TCP/IP is not yet configured, and not proc because
   # that has already been mounted):
   /sbin/mount -a -v -t nonfs,nosmbfs,noproc
   ```
2. e modificatele come segue:
   ```
   # mount non-root file systems in fstab (but not NFS or SMB
   # because TCP/IP is not yet configured, and not proc because
   # that has already been mounted):
   /sbin/mount -v /mnt/pendrive
   /usr/sbin/cryptsetup -v -c aes -d /mnt/pendrive/keyfile
   create secret /dev/hdaX
   /sbin/mount -a -v -t nonfs,nosmbfs,noproc
   /sbin/umount -v /mnt/pendrive
   ```

5.2.2 Slackware 11.0

1. cercate le righe:
```
# Mount non-root file systems in fstab, but not NFS or SMB
# because TCP/IP is not yet configured, and not proc or sysfs
# because those have already been mounted.  Also check that
# devpts is not already mounted before attempting to mount
# it.  With 2.4.x kernels devpts is mounted from an fstab
# entry while with a 2.6.x or newer kernel udev mounts it.
# We also need to wait a little bit to let USB and other
# hotplugged devices settle (sorry to slow down the boot):
echo "Mounting non-root local filesystems:"
sleep 3
if /bin/grep -wq devpts /proc/mounts ; then
  /sbin/mount -a -v -t
nonfs,nosmbfs,nocifs,noproc,nosysfs,nodevpts
else
  /sbin/mount -a -v -t nonfs,nosmbfs,nocifs,noproc,nosysfs
fi
```
2. e modificate come segue:
```
# Mount non-root file systems in fstab, but not NFS or SMB
# because TCP/IP is not yet configured, and not proc or sysfs
# because those have already been mounted.  Also check that
# devpts is not already mounted before attempting to mount
# it.  With 2.4.x kernels devpts is mounted from an fstab
# entry while with a 2.6.x or newer kernel udev mounts it.
# We also need to wait a little bit to let USB and other
# hotplugged devices settle (sorry to slow down the boot):
echo "Mounting non-root local filesystems:"
/sbin/mount /mnt/pendrive -v
/usr/sbin/cryptsetup -c aes -d /mnt/pendrive/key create
secret /dev/hdaX -v
#sleep 3
if /bin/grep -wq devpts /proc/mounts ; then
/sbin/mount -a -v -t
nonfs,nosmbfs,nocifs,noproc,nosysfs,nodevpts
else
/sbin/mount -a -v -t nonfs,nosmbfs,nocifs,noproc,nosysfs
fi
/sbin/umount /mnt/pendrive -v
```

Questo procedimento automatizzerà il mount delle partizioni crittate all'avvio. Notate che se non avete il pendrive usb pluggato al pc durante il boot, il sistema parte, ma non trovando la chiave di cifratura le home non verranno montate lasciando l'utente nel limbo :)

6. Ending credits

L'articolo è stato scritto da HelLViS69

hellvis69@netsons.org
http://hellvis69.netsons.org
LinuxVar – Linux User Group Varese