

Inhalt

Inhaltsverzeichnis

1. Vorabinformation
2. Warnung
3. Allgemeines
4. Laufwerke verschlüsseln mit Loop-AES
 1. Vorwort
 2. Loop-AES
 3. Einrichten einer verschlüsselten Partition (als root!)
 4. Einrichten eines verschlüsselten Containers (als root!)
 5. Verschlüsseln des Home-Verzeichnisses
 6. Passwort ändern oder entfernen
 7. Image auf DVD backen
 8. Sonstiges

1. Vorabinformation

CryptoLoopDevice soll durch dm-crypt abgelöst werden. dm-crypt funktioniert in der Praxis schon sehr gut. Wer also sich in die Materie einarbeiten möchte, sollte gleich zu dm-crypt greifen.


2. Warnung

Das Thema Verschlüsselung ist wirklich ziemlich umfangreich und man kann eine Menge Zeit damit verbringen - um es kurz zu machen: Das Loop-AES Design ist wirklich ziemlich schlecht, es gab schon einige Sicherheits-Probleme und auch der Nachfolger dm-crypt ist wirklich nicht viel besser. Wer sich ernsthaft für gute Implementierung von Verschlüsselung interessiert, sollte einen Blick auf GELI von FreeBSD werfen! Spart Euch die Zeit!

3. Allgemeines

Man kann mit Hilfe eines Loopback-Device auch Verschlüsselung realisieren: alle Zugriffe auf das entsprechende Device werden dann verschlüsselt.

Der Datenverkehr wird direkt mit dem eingegebenen Schlüssel verschlüsselt, deshalb sollte er nicht zu kurz sein. losetup und mount müssen dafür gepatcht sein.

 losetup kann nicht feststellen, ob der Schlüssel richtig ist, erst der mount-Befehl erkennt das gültige Dateisystem. Wenn der mount unbedingt auf einer Dateisystemprüfung besteht, ist Vorsicht angesagt.

Die Methode über das CryptoLoopDevice ist recht stabil.

⚠ Unter SuSE kann das Anlegen eines verschlüsselten Containers oder einer verschlüsselten Partition bequem mit YaST über den Bereich Partitionieren erledigt werden. In der Version 9.0 sollten zuvor jedoch über das Online-Update unbedingt die Patches eingespielt werden, da das Modul auf dem ausgelieferten Datenträger einen Bug enthält.

⚠ Unter Mandriva kann das Anlegen einer verschlüsselten Partition bequem mit dem Kontrollzentrum unter Einhängpunkte/Partitionen, Aktivieren des Expertenmodus und der Option "encrypted" erfolgen.

⚠ Im Debian Kernel in der Version 2.6 ist aes schon dabei. Man ersetzt in der folgenden Anleitung AES128 einfach durch aes. Selbst kompilieren ist nicht notwendig, aber es ist darauf zu achten, dass die Kernelmodule cryptoloop und aes geladen sind.

4. Laufwerke verschlüsseln mit Loop-AES

4.1. Vorwort

Verschlüsselung ist in vielen Fällen sinnvoll, auch dann, wenn es eigentlich nichts zu verheimlichen gibt. Wer möchte schon Pinnummern, Passwörter, Emails, Adressen von Freunden und Bekannten, Firmendaten, Steuererklärung, Finanzverwaltung oder was sonst noch auf dem Rechner liegt, anderen, unbekanntenen Personen zugänglich machen? Ein sauber eingerichtetes Linux-System ist für Angriffe aus dem Internet nur wenig anfällig und bei einem nicht dauerhaft verbundenen Rechner auch eher unwahrscheinlich. Dies nützt aber insbesondere Notebookbesitzern wenig, wenn es gestohlen oder versehentlich liegengelassen wurde. Und so wünscht sich der eine oder die andere vielleicht doch ein bisschen mehr Schutz für die Daten auf dem eigenen Rechner.

4.2. Loop-AES

Unter Linux gibt es verschiedene Möglichkeiten Laufwerke zu verschlüsseln, einigen wird das kommerzielle BESTCRYPT der Firma Jetico oder der internationale Kernelpatch, der den Kernel mit ein paar Krypto-Algorithmen ausstattet, bekannt sein. Seit kurzem ist ein dritter, wichtiger Vertreter hinzugekommen - Loop-AES von Jari Ruusu.

Es baut wie die beiden anderen auf einem Kernelmodul auf, die verschlüsselten Daten befinden sich wahlweise in einer großen Datei (Container) oder einer Partition. Loop AES steht unter der GPL und verwendet den bekannten AES-Algorithmus in auswählbaren Stufen von 128, 192 und 256 Bit. Der erstellte Container bzw. die verschlüsselte Partition kann als Backup auch auf eine CD-R gebrannt werden, aus einer Partition (hier: /dev/hda7) erhält man beispielsweise mit folgendem Befehl eine brennbare Datei:

```
cat /dev/hda7 > /tmp/backup_von_hda7
```

Diese Datei kann auch auf einen anderen Linux-Rechner aufgespielt und verwendet werden, sofern ebenfalls Loop-AES installiert ist. Loop-AES ist sehr schnell, der Datendurchsatz der Festplatte wird quasi nicht gebremst - lediglich die CPU-Last beim Lesen oder Schreiben steigt etwas an. Der AES-Algorithmus ist seit kurzem in der 128 Bit-Stufe als Assemblercode für Pentium-PCs enthalten und belastet das System nur minimal.

Loop-AES ist zwar nicht so einfach zu installieren wie Scramdisk oder PGPDisk unter Windows, im späteren Einsatz ist es jedoch sehr unkompliziert und stabil. Ich selbst verwende Loop-AES seit rund einem halben Jahr und hatte bisher keine Probleme. Anders als beim internationalen Kernelpatch muss nicht eine spezielle Kernelversion verwendet werden, Loop-AES sollte mit allen Linuxversionen ab 2.0 zusammenarbeiten.

Für die Installation sind folgende Programmpakete erforderlich:

- Kernelquellcode (2.0, 2.2 oder 2.4): <ftp://ftp.kernel.org/pub/kernel/>
- Quellcode von util-linux: <ftp://ftp.kernel.org/pub/linux/utils/util-linux/>
- Loop-AES <http://sourceforge.net/projects/loop-aes/>
- Entwickler-Umgebung (gcc, binutils, glibc, ...)

Meistens muß man nun den Kernel mit `CONFIG_BLK_DEV_LOOP=n` vorher neu backen, da die meisten ja das loopdevice schon vorher im Kernel auf Y oder M hatten. Die Readme empfiehlt sowieso vorher den Kernel neu zu bauen, um sicherzustellen dass laufender Kernel und Kernelsource stimmig sind. (Wenn man allerdings loop-aes direkt in den kernel bauen willen (patch aus der loop-aes Distribution), dann sollte man `CONFIG_BLK_DEV_LOOP` schon eingeschaltet lassen.)

Zuerst sollte das Archiv von Loop-AES heruntergeladen und entpackt werden. Ausserdem sollten zumindest die Header-Dateien der aktuellen Kernelkonfiguration installiert sein, meist findet sich das entsprechende Paket auf den Distributions-CDs. Danach lädt man die aktuelle bzw. die von Loop-AES vorgeschlagene Version von util-linux herunter und entpackt das Archiv, es enthält die Quellcodes von Programmen wie z.B. mount, die für die Verschlüsselung modifiziert werden müssen. Mit `cd util-linux*` wird in das Verzeichnis des Programmpakets gewechselt. Mit

```
cat /pfad/zu/Loop-AES/util-linux-2.xx.diff | patch -p1
```

werden die entsprechenden Programme gepatcht, mit `make` übersetzt und können danach installiert werden. Dies geschieht mit folgenden Befehlen (als root!):

```
cd util-linux-2.11n
./configure
make ADD_RAW=no
cd mount
install -m 4755 -o root mount umount /bin
install -m 755 losetup /sbin
```

Im Verzeichnis mount befinden sich noch die leicht veränderten Manpages von `fstab`, `losetup`, `umount` und `mount`. Sie sollten in die entsprechenden Pfade kopiert werden und die alten Versionen überschreiben. Leider sind die Pfadangaben zu den Manpages nicht bei allen Linux-Distributionen gleich, bei Mandriva befinden sie sich in `/usr/share/man`.

Nach der Installation der Programme `mount`, `umount` und `losetup` wird wieder zurück ins Verzeichnis von Loop-AES gewechselt und das Kernelmodul mit `make` (als root!) installiert. Der Befehl `make tests` zeigt, ob soweit alles funktioniert. Wenn ja, kann die verschlüsselte Partition oder der Container erstellt werden. Die beiden Vorgänge sind sich sehr ähnlich, ich behandle sie zugunsten der Übersichtlichkeit dennoch getrennt.

4.3. Einrichten einer verschlüsselten Partition (als root!)

```
losetup -e AES128 /dev/loop0 /dev/hda7
mkfs -t ext2 /dev/loop0
losetup -d /dev/loop0
mkdir /mnt/rypted
```

Mit `losetup` wird die zu verschlüsselnde Partition ausgewählt, hier `/dev/hda7`. Es wird nach einem Passwort von mindestens 20 Buchstaben gefragt, dieses darf nicht verlorengehen. Eine Partitionstabelle der Festplatte gibt der Befehl `fdisk` aus. Der Befehl `mkfs -t ext2` erstellt ein ext2-Dateisystem, es können jedoch auch andere Dateisysteme wie ReiserFS erstellt werden.

Eventuell sollte man vorher die gesamte Partition mit Pseudozufallszahlen ueberschreiben, damit nicht sofort klar ist, welche Daten noch alten Muell enthalten, und was verschuesselte echte Daten sind:

```
head -c 30 /dev/urandom | losetup -p0 -e AES128 /dev/loop0 /dev/hda7
# mit zufaelligem Schluessel das loop device aufsetzen
dd if=/dev/zero of=/dev/loop0 # und das loop device mit Nullen
fuellen
losetup -d /dev/loop0 # und wieder weg
```

Durch das Fuellen des loop Devices mit Nullen, wird das darunterliegende Block Device mit einem mehr oder weniger guter Strom an Zufallszahlen beschrieben. Man kann natuerlich auch `/dev/random` drauf tun, aber das dauert ewig.

Als nächstes muss die Datei `/etc/fstab` abgeändert werden, die Zeile

```
/dev/hda7 /mnt/rypted ext2 defaults,loop=/dev/loop0,encryption=AES128
0 0
```

wird hinzugefügt. Möchte man nicht bei jedem Rechnerstart automatisch nach dem Passwort gefragt werden, empfiehlt es sich die Option `noauto` noch unterzubringen.

Das verschlüsselte Laufwerk kann - wie andere Laufwerke auch - mit dem einfachen Befehl `mount /mnt/rypted` eingebunden werden. Mit `umount /mnt/rypted` und `losetup -d /dev/loop0` wird es wieder ausgehängt.

4.4. Einrichten eines verschlüsselten Containers (als root!)

Diese Option empfiehlt sich dann, wenn gerade keine geeignete Partition frei ist und die Partitionstabelle nicht geändert werden soll.

```
dd if=/dev/zero of=/container bs=1024 count=5120
# optional (Erklaerung siehe oben, bei den Partitionen):
head -c 30 /dev/random | losetup -p0 -e AES128 /dev/loop0 /container
dd if=/dev/zero of=/dev/loop0
losetup -d /dev/loop0
losetup -e AES128 /dev/loop0 /container
mkfs -t ext2 /dev/loop0
losetup -d /dev/loop0
mkdir /mnt/rypted
```

Der Befehl `dd` lässt eine Datei (unseren Container) mit 5 MB Größe entstehen

(5120x1024Byte). *losetup* weist dem Container das Loop-Device zu und fragt nach dem Passwort, *mkfs* erstellt ein ext2-Dateisystem und *mkdir* erstellt das Verzeichnis, in das später die zu verschlüsselnden Daten kommen.

Wie bei dem Beispiel mit der verschlüsselten Partition muss wieder die Datei */etc/fstab* angepasst werden, die Zeile


```
/container /mnt/rypted ext2
defaults,loop=/dev/loop1,encryption=AES128 0 0
```

wird hinzugefügt.

Das Laufwerk kann mit `umount /mnt/rypted` und `losetup -d /dev/loop0` wieder ausgehängt werden.

4.5. Verschlüsseln des Home-Verzeichnisses

Anstelle des Verzeichnisses */mnt/rypted* kann auch ein Home-Verzeichnis eines beliebigen Benutzers verschlüsselt werden. Hier ein Beispiel: Zuerst meldet man sich unter seiner eigenen Kennung ab und als root an. Die alten Daten aus dem entsprechenden Home-Verzeichnis werden nach */tmp/benutzername* verschoben (Achtung: auch versteckte Dateien mitverschieben!), danach wird das verschlüsselte Laufwerk in das entsprechende Home-Verzeichnis (z.B. */home/jan*) eingehängt und die Daten zurückverschoben (auf Zugriffsrechte achten). Hat alles geklappt, sollten die alten Daten unter */tmp/benutzername* auf sichere Weise gelöscht werden, dazu empfiehlt sich die Wipe-Option im Konqueror, BCWipe oder dies umfangreichen Secure Deletion Tools von THC. Das Home-Verzeichnis des Benutzers jan ist nun verschlüsselt, allerdings sollte in der Datei */etc/fstab* der Verzeichnispfad auf */home/jan* umgeändert werden, damit beim nächsten Rechnerstart alles funktioniert. Zur automatischen Passwortabfrage beim Rechnerstart darf die Option `noauto` nicht angegeben sein. Dies stellt sicher, sofern das Passwort richtig eingegeben wurde, dass das verschlüsselte Home-Verzeichnis bereits vor der Anmeldung des Benutzers eingehängt wird und vermeidet damit spätere Konflikte. Vielleicht schreibt auch jemand ein kleines Anmeldungsskript, damit das Verschlüsseln des Home-Verzeichnisses auch bei Mehrbenutzersystemen reibungslos verläuft.

Eine alternative Vorgehensweise, das Home auch erst im nachhinein einzuhängen, findet sich  hier.

Loop-AES bietet noch weitere Möglichkeiten, beispielsweise das Verschlüsseln der Root-Partition (also alles ab */*) oder das Verschlüsseln der SWAP-Partition (bei Notebooks wird der Hauptspeicher im Schlafmodus auf der Festplatte entladen - mitsamt den geschützten Passwörtern im Hauptspeicher). Nähere Beschreibungen befinden sich in der englischsprachigen README von Loop-AES.

4.6. Passwort ändern oder entfernen

Es gibt keine Möglichkeit das Passwort einer verschlüsselten Partition nachträglich zu ändern oder zu entfernen. Dazu muss man

1. eine Sicherheitskopie anlegen
 - z.B. `tar cvzf /tmp/rypted-backup.tgz /mnt/rypted` ()
Dabei können unverschlüsselte Spuren zurückbleiben!
 - oder wie oben gleich

```
cat /dev/hda7 > /tmp/rypted-backup-raw und später mit
loopback mounten
```

2. das Dateisystem wie oben neu anlegen .
3. die Sicherungskopie zurückspielen.

4.7. Image auf DVD backen

Näheres siehe: <http://plus-linux.de/wiki.cgi?Loop-Aes>

- `dvdrecord -dao -vv speed=4 dev=/dev/hdc -data image-cript.ext2 driveropts=burnproof`

4.8. Sonstiges

- Übernommen von http://www.pl-forum.de/t_system/loop-aes.html mit freundlicher Genehmigung von <jan_allgeier AT nexgo DOT de>
- Schaut euch auch mal `aespipe` vom gleichen Autor wie `loop-aes` an. Es stellt eine pipe dar, über die man AES verschlüsseln kann. Es benutzt die gleichen Algorithmen, wie `loop-aes`, braucht aber keinerlei kernel-unterstützung. Es lässt sich ganz einfach über die Quellen auf jedem System übersetzen und installieren (`./configure, make, make install`). Ein `"tar -cvzf - backupdir | aespipe -p3 > meinarchiv.tgz.aes 3< /etc/backuppas"` erzeugt so z.B. ein verschlüsseltes Backup, wobei das Passwort in `/etc/backuppas` abgelegt ist, damit man es in Skripten ausführen kann. Kurzum, es ist ein ideales Werkzeug, um ohne großen Aufwand Daten zu ver- und entschlüsseln. -- WinfriedMueller
- [HowTo von www.kerneli.org](http://www.kerneli.org)
- Der Befehl `"losetup -e AES128 /dev/loop0 /dev/hda7"` funktionierte bei mir (RomanKreisel) (Gentoo ~x86, Kernel 2.6.3-ck1) nicht. Etwas googlen ergab dann, dass man wohl `"-e aes-128"` nehmen muss. So ging's dann auch.
 - Vielleicht sollten wir eine Tabelle für die verschiedenen Distributionen machen
- Vorsicht: Abhängig von der Umgebung, von der aus das Passwort eingerichtet wird, kann sich die Schreibweise ändern, wenn beim Booten noch nicht der Zeichensatz geladen ist, mit dem das Passwort eingegeben wurde. Bei mir (WolfgangSchricker) wurden aus jedem 'z' ein 'y' und umgekehrt 😊
 - Da kann aber das arme Crypto-Device nichts für, wenn Deine Tastaturbelegung nicht stimmt. Ist es nicht eher so, dass auch bei den normalen Eingaben 'y' und 'z' vertauscht sind und Du es bei der Blindeingabe einfach nicht gemerkt hast?
 - Klar, deshalb der Hinweis auf die *Umgebung*. Ich war während der Einrichtung per `ssh` auf dem Rechner eingeloggt und habe das Kennwort festgelegt. Nach dem Reboot kam die Kennwort-Frage vor dem Laden des Zeichensatzes dieses Rechners. Deshalb der Hinweis - erspart eventuelles langes Nachdenken - wie bei mir 😊