Crypto-Container mit cryptsetup Luks

Diese Anleitung beschreibt die Erstellung und Nutzung einer verschlüsselten Datei unter Debian. Dabei kommt cryptsetup-luks zum Einsatz.

Variable	Bedeutung	Beispiel
$DATEI	Pfad für Abbild der verschlüsselten Partition	`/data/plastikpackung`
$MB	Grösse von $DATEI in Megabyte	100
$CRYPTODATEI	Name deiner Cryptodatei in `/dev/mapper`	gouda
$MOUNT	Verzeichnis, wo Partition eingehängt wird	`/mnt/kuehlschrank`

Container erstellen

Die Verschlüsselung baut auf dem sogenannten *Device Mapping* auf, welches ab Kernel 2.6.4 implementiert ist. Laut deren Entwickler ist es um einiges besser fuer Crypto-Zwecke geeignet als die Cryptoloop-Variante. Für die folgende Schritte benötigst du root-Rechte:

- Stell sicher, dass die folgenden Module geladen sind:

```
modprobe loop

modprobe dm_crypt
```

- Installiere nun die notwendige Pakete:

```
apt install cryptsetup hashalot
```

- Als erstes erstellst du eine neue Datei. Du kannst `/dev/urandom` benutzen, um zufällige Werte in diese Datei zu schreiben:

```
dd if=/dev/urandom of=$DATEI bs=1M count=$MB
```

- Anschliessend richtest du das loop-device ein:

```
losetup /dev/loop0 $DATEI
```

- Jezt wird aus dem loop-device eine Cryptodatei:

```
cryptsetup luksFormat /dev/loop0
```

- Danach kannst du die neue Partition mappen:

```
cryptsetup luksOpen /dev/loop0 $CRYPTODATEI
```

- Noch schnell die Datei mit einem Dateisystem formatieren:

```
mkfs.ext4 /dev/mapper/$CRYPTODATEI
```

- Mounten und fertig:

```
mount /dev/mapper/$CRYPTODATEI $MOUNT
```

Täglicher Gebrauch

Das folgende Skript vereinfacht den Umgang mit dem Container:

```
#!/bin/sh

set -eu


DATEI=$1
MOUNTPOINT="$2"
CRYPTODATEI=$(basename "$DATEI")

case "$1" in
  start)
        losetup /dev/loop0 "$DATEI"
        cryptsetup luksOpen /dev/loop0 "$CRYPTODATEI"
        mount -t ext3 -o defaults,user "/dev/mapper/$CRYPTODATEI"
"$MOUNTPOINT"
        ;;
  stop)
        umount "$MOUNTPOINT"
        cryptsetup luksClose "$CRYPTODATEI"
        ;;
  restart)
        "$0" stop
        "$0" start
        ;;
  *)
        echo "Usage: $(basename "$0") {start|stop|restart} DATEI MOUNTPOINT"
        ;;

 esac

exit 0
```

Luks ermöglicht es dir, mit verschiedenen Schlüsseln auf die Partition zuzugreifen. Mit folgendem Befehl fügst du einen weiteren hinzu:

```
cryptsetup luksAddKey /dev/$DEVICE
```

Dazu musst du das Passwort eines schon vorhandenen Schlüssels eingeben. Analog dazu kannst du mit `luksDelKey` ein Schlüssel wieder entfernen.

Paranoia

Mit `dmsetup info` lassen sich benutzte devicemappings anzeigen. Taucht hier ein Teil als ACTIVE auf, so hat jedermensch darauf Zugriff (auch wenn es nicht gemountet ist, könnte - wer die nötigen Rechte hat - das Teil ohne passendes Cryptokennwort mounten) - es ist in diesem Zustand quasi entschlüsselt!

Deswegen immer nach dem unmounten, das mapping entfernen (sonst gibt's (erfahrene Beamte vorausgesetzt) evtl. Stress bei der Hausdurchsuchung):

```
umount $MOUNT

cryptsetup luksClose $CRYPTODEVICE
```

Du kannst alle mappings gleichzeitig entfernen mit:

```
dmsetup remove_all
```

Denk dran: `cryptsetup luksClose` niemals nicht vergessen oder aber Stecker ziehen ;) !

From:
https://howto.wikis.systemausfall.org/ - **Das HowTo-Wiki**

Permanent link:
https://howto.wikis.systemausfall.org/privacy/crypto-container_mit_luks

Last update: **2022/11/20 15:47**