Verschlüsselte Dateisysteme unter Linux

Michael Gebetsroither

http://einsteinmg.dyndns.org
gebi@sbox.tugraz.at

Einteilung

- Theorie
  - Kurze Einführung
  - Verschiedene Möglichkeiten der Verschlüsselung
  - Unsicherheitsfaktoren und wie man sie vermeiden kann
- Workshop
  - Verschlüsselter swap in einer Minute
  - Arbeitsumgebung
  - cryptsetup
  - Linux Unified Key Setup (LUKS)

Kurze Einführung

Warum eigentlich verschlüsseln (Wünsche) ?

- Niemand soll meine Daten lesen können!
- Wenn jemand in mein System einbricht sollen meine Daten sicher sein.
- Es soll nicht möglich sein nachzuweisen, dass ich bestimmte Daten auf meiner Festplatte gespeichert habe.
- Niemand soll gezielt verschlüsselte Daten manipulieren können.
- Niemand soll nachweisen können, dass ich in letzter Zeit bestimmte Daten geschrieben habe.

Kurze Einführung

Warum eigentlich verschlüsseln (Praxis) ?

(• Niemand soll meine Daten lesen können!)

- ~~Wenn jemand in mein System einbricht sollen meine Daten sicher sein.~~
- ~~Es soll nicht möglich sein nachzuweisen, dass ich bestimmte Daten auf meiner Festplatte gespeichert habe.~~
- ~~Niemand soll gezielt verschlüsselte Daten manipulieren können.~~
- ~~Niemand soll nachweisen können, dass ich in letzter Zeit bestimmte Daten geschrieben habe.~~

Verschiedene Möglichkeiten der Verschlüsselung

- Kernelseitig
  - Cryptoloop
  - Loop-aes
  - dm-crypt (Devicemapper crypto target) - cryptsetup
  - LUKS (crypsetyp-luks)
- Filesystem in Userspace (FUSE)
  - Encrypted Filesystem (EncFS)
  - PhoneBook (Filesystem in Userspace)

Unsicherheitsfaktoren

- Wichtige Informationen landen im Swap:
  - Swap verschlüsseln
- Daten können vom Angreifer auf der Festplatte (gezielt) verändert werden:
  - public-IV CBC mode vermeiden
- Brute-force Angriffe erschweren:
  - Sichere Passphrases
  - Sicherer Passwortspeicher (LUKS)

ideales Setup unter Linux

- Kernel >2.6.10
  - um ESSIV CBC zu verwenden
- LUKS (cryptsetup-luks)
  - Um brute-force Angriffe zu erschweren
  - Möglichkeit mehrere Passphrasen

Verschlüsselter swap in einer Minute

- Voraussetzungen
  - Debian
  - Kernel mit device-mapper und dm-crypt
  - cryptsetup
- Konfigurationsdateien
  - /etc/default/cryptdisks
  - /etc/crypttab
  - /etc/fstab

Verschlüsselter swap in einer Minute

swapoff -a

/etc/default/cryptdisks:

# Run cryptdisks at startup ?
CRYPTDISKS_ENABLE=Yes

Verschlüsselter swap in einer Minute

/etc/crypttab:

```
# <target device> <source device> <key file> <options>
cswap                   /dev/hda1      /dev/random    swap
```

/etc/fstab:

```
/dev/mapper/cswap  none    swap  sw,pri=1        0  0
```

```
# /etc/init.d/cryptdisks reload
```

```
# swapon -a
```

Arbeitsumgebung

Per ssh auf den Arbeitsrechner einloggen:

user = grml
password= test

Eigenes Verzeichniss erzeugen:

mkdir playground/<irgendwas>
cd playground/<irgendwas>

Cryptsetup

- Voraussetzungen
  - Kernel > 2.6.4 (besser >2.6.10)
  - Kerneloptionen:
    - device-mapper
    - dm-crypt
    - einige cipher (AES empfehlenswert)
  - cryptsetup
  - Freie Partition oder in Datei per losetup

Cryptsetup - Vorbereitung

- Erzeugen der Containerdatei/Mountpoint

  ```
  # dd if=/dev/zero of=./image bs=1M count=100
  # mkdir mp
  ```

- Erzeugen des loop-devices

  ```
  # sudo losetup -v -v $(sudo losetup.orig -f) `pwd`/image
  ```

  Merken des benutzten loop-devices ;-)

- Wiederfinden des loop-devices falls vergessen

  ```
  # sudo losetup -a |grep `pwd`
  ```

Cryptsetup – real task

- Erzeugen des Mappings

  # sudo cryptsetup -c aes-cbc-essiv:sha256 -s 256 create \
  <NAME> <LOOP>

  NAME == Name des privaten Ordners

- Anzeigen der Configuration

  # sudo cryptsetup status <NAME>

Gratulation, nun ist die Partition verschlüsselt!

Cryptsetup

- Formatieren der verschlüsselten Partition:

  # sudo mkfs.xfs /dev/mapper/<NAME> mp

- Mounten der verschlüsselten Partition:

  # sudo mount /dev/mapper/<NAME> mp

- Testen, Testen, Testen:

  # sudo cp, mv, touch, dd, ...

Cryptsetup

- ACHTUNG!!

  cryptsetup erstellt das Mapping auch, wenn die Einstellungen falsch sind:

  - falsche key-length
  - falscher cipher
  - falsche Passphrase

Wenn mit einem falschen Mapping auf die Partition geschrieben wird sind die Daten verloren!!

=> LUKS (Linux Unified Key Setup)

LUKS

- Referenzimplementation cryptsetup-luks

  Home: http://luks.endorphin.org/
  Debianpaket: http://einsteinmg.dyndns.org/debian

- Vorteile
  - “Sicher” gegenüber brute-force Angriffen
  - Speichert Einstellungen
  - Kein Vernichten der Daten bei falscher Passphrase
  - Mehrere Passphrases pro Device möglich
  - Kein Neuverschlüsseln bei Passphrasewechsel

LUKS

- Nachteile
  - zurzeit noch mangelnde Unterstützung der Distributionen
  - ... ?
- Plattformen
  - Debian
  - Gentoo
  - Redhat

LUKS

- Vorbereiten für LUKS

  # sudo umount mp

  # sudo cryptsetup remove <NAME>

- Erstellen des LUKS Headers

  # sudo cryptsetup -c aes-cbc-essiv:sha256 -s 256 \
  luksFormat <LOOP>

- Erstellen des Mappings

  # sudo cryptsetup luksOpen <LOOP> <NAME>

LUKS

- Formatieren der verschlüsselten Partition:

  # sudo mkfs.xfs /dev/mapper/<NAME> mp

- Mounten der verschlüsselten Partition:

  # sudo mount /dev/mapper/<NAME> mp

- Testen, Testen, Testen:

  # sudo cp, mv, touch, dd, ...

LUKS

- Anzeigen des LUKS Headers

  # sudo cryptsetup luksDump <LOOP>

- Hinzufügen einer zusätzlichen Passphrase

  # sudo cryptsetup luksAddKey <LOOP>

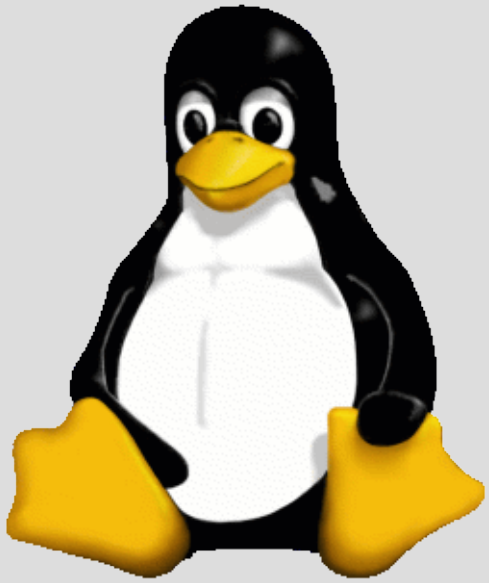
- Löschen der alten Passphrase

  # sudo cryptsetup luksDelKey <LOOP> 0

=> Erfolg, wir haben nun die Passphrase der Partition ohne erneute Verschlüsselung geändert!

FRAGEN

?



Links

- Cryptsetup - http://www.saout.de/misc/dm-crypt/
- LUKS - http://luks.endorphin.org/
- FUSE - http://fuse.sourceforge.net/
- PhoneBook - http://www.freenet.org.nz/phonebook/
- EncFS - http://arg0.net/users/vgough/encfs.html
- loop-aes - http://loop-aes.sourceforge.net/
- Sicherheit - http://clemens.endorphin.org/LinuxHDEncSettings
- Cryptsetup-luks - http://einsteinmg.dyndns.org/debian