

Linux am Dienstag

LUKS Laufwerke

via

USB entschlüsseln

LUKS Laufwerke via USB entschlüsseln

Freigabe:

Dieser Foliensatz darf explizit von jedem zum eigenen Vortrag genutzt werden.

LUKS Laufwerke via USB entschlüsseln

Vorwort

Diese Präsentation setzt voraus,
daß es bereits eine Installation mit

Festplattenvollverschlüsselung
(**F**ull**D**isk**E**ncryption)

durch den Einsatz von LUKS gibt.

LUKS Laufwerke via USB entschlüsseln

Dabei werden normalerweise bei der Installation des Betriebssystems eine oder zwei Partitionen mit LUKS Verschlüsselung angelegt, wenn man im Partitionierer „**meine Daten verschlüsseln**“ anhakt.

Das „Passwort“ wird auch als „Passphrase“ bezeichnet.

LUKS Laufwerke via USB entschlüsseln

Wir brauchen

Befehle: `fdisk,dd,cryptsetup,dracut`

einen USB Stick

Möglichst haltbar, benutzt **und nicht mehr benötigt.**

LUKS Laufwerke via USB entschlüsseln

Schritt 1:

Den USB Stick vorbereiten

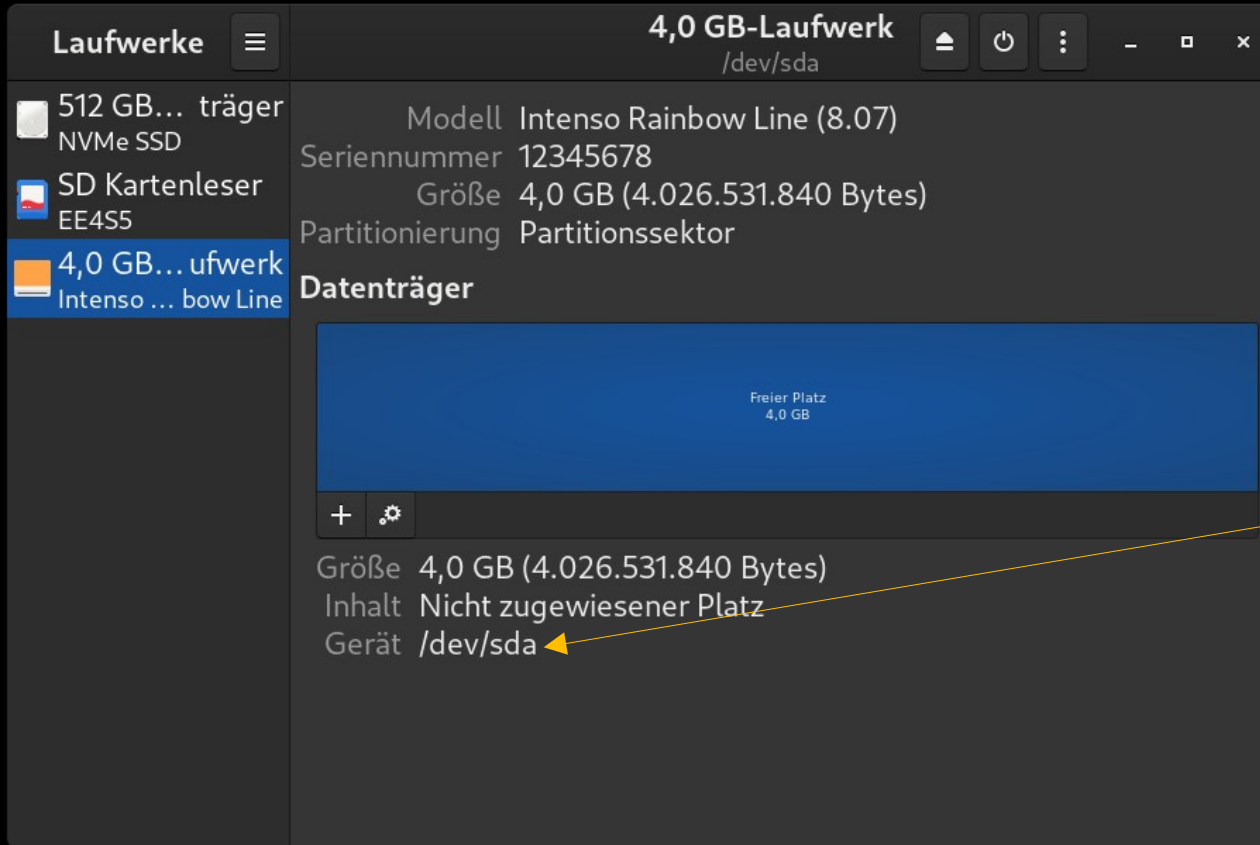
LUKS Laufwerke via USB entschlüsseln

Mit

„`cat /proc/partitions`“
oder dem Laufwerktool

finden wir den Devicenamen heraus

LUKS Laufwerke via USB entschlüsseln



In diesem Fall

/dev/sda

LUKS Laufwerke via USB entschlüsseln

Hinweis: /dev/**sda** ist ok

wenn man zur Bootzeit nie andere
Speichermedien hinzufügen wird,

z.B. auf einem Tablet oder Laptop!

LUKS Laufwerke via USB entschlüsseln

Hinweis:

Statt `/dev/sda/` kann man z.b. eindeutige Geräte wie `/dev/disk/by-path/...` benutzen, in dem sogar der Port des USB Sticks in den Gerätenamen einfließt.

Man darf ihn dann aber auch nirgendwo anders reinstecken ;)

LUKS Laufwerke via USB entschlüsseln

Als ROOT Benutzer geben wir ein:

```
fdisk /dev/sda
```

LUKS Laufwerke via USB entschlüsseln

```
root@fedora:~# fdisk /dev/sda
```

```
Welcome to fdisk (util-linux 2.40.1).
```

```
Changes will remain in memory only, until you decide to  
write them.
```

```
Be careful before using the write command.
```

```
Command (m for help):
```

LUKS Laufwerke via USB entschlüsseln

Nun löschen wir mit dem „**d**“ Befehl alle alten Partitionen, legen mit „**n**“ eine neue an und schreiben diese Änderungen mit „**w**“ auf den Stick.

Eine neue Partition anzulegen ist keine Pflicht!

LUKS Laufwerke via USB entschlüsseln

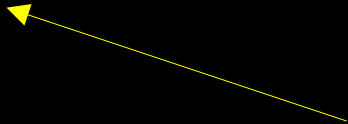
```
root@fedora:~# fdisk /dev/sda
```

```
Welcome to fdisk (util-linux 2.40.1).
```

```
Changes will remain in memory only, until you decide to  
write them.
```

```
Be careful before using the write command.
```

```
Command (m for help): d
```



So oft eingeben, bis keine alten
Partitionen mehr da sind

LUKS Laufwerke via USB entschlüsseln

Dann

LUKS Laufwerke via USB entschlüsseln

Command (m for help): **n**

Partition type

 p primary (0 primary, 0 extended, 4 free)

 e extended (container for logical partitions)

Select (default p): **p**

Partition number (1-4, default 1): **1**

First sector (2048-7864319, default 2048):

Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-7864319, default 7864319): **{RETURN TASTE}**

Created a new partition 1 of type 'Linux' and of size 3,7 GiB.

Command (m for help): **w**

LUKS Laufwerke via USB entschlüsseln

Hinweis:

Wer seinen Key **nicht vorn** drauf schreiben und dabei ggf. den Partitionstable zerstören möchte, der kann dies gern am Ende des USB Sticks machen, in dem die letzte Partition zuerst verkleinert wird.

Dann müßtet Ihr Euch aber auch das BlockOffset für diese Stelle **selbst ausrechnen**, denn das kann diese Anleitung unmöglich vorhersehen und daher nicht leisten, weil wir natürlich nicht wissen, wie groß Eurer Stick oder Eurer Key ist.

„Hinten“ speichern hat den Vorteil den Stick ganz sicher weiterhin benutzen zu können.

LUKS Laufwerke via USB entschlüsseln

Nun brauchen wir jede Menge **Zufall** auf dem USB Stick:

```
dd if=/dev/urandom of=/dev/sda bs=512 seek=1 count=2046
```

LUKS Laufwerke via USB entschlüsseln

bs=512 meint, dass wir eine Blockgröße von 512Byte haben

und

seek=1 sagt dem Tool, dass es den Block 0 überspringen soll

und

count=2046 sagt, dass 2046 Blöcke geschrieben werden sollen.

LUKS Laufwerke via USB entschlüsseln

Nun müssen wir uns diese Zufallsdaten in eine Datei sichern, damit wir diese

- a) auf mehreren Sticks schreiben können
- b) und LUKS den Key geben können

LUKS Laufwerke via USB entschlüsseln

```
dd if=/dev/sda bs=512 skip=1 count=16 of=/root/luks.key
```

LUKS Laufwerke via USB entschlüsseln

Wieso „16“ Blöcke und nicht „2046“ ?

LUKS Laufwerke via USB entschlüsseln

Weil: $16 \times 512 = 8192$ Bytes sind.

Bei den eingesetzten Algorithmen gilt diese Länge schon als „Overkill“, weil es ... 256^{8192} ...

„Überlauf: Das Ergebnis kann nicht berechnet werden“

.. Möglichkeiten wären.

LUKS Laufwerke via USB entschlüsseln

Bei einem normalen Passwort **dieser Länge** aus Groß- und Kleinbuchstaben, sowie Zahlen, wären es „nur“ 62^{8192} mögliche Kombinationen die man als Angreifer austesten müßte.

Jetzt sind Passwörter aber mit 20 Zeichen, also 62^{20} Möglichkeiten, schon „**zu lang**“ für einige Menschen um sich das zu merken.

Immerhin kann man das ausrechnen: $7,044234255 \times 10^{35}$

LUKS Laufwerke via USB entschlüsseln

Fazit:

Schmeißt Passwörter weg,
nutzt Keys zum Entschlüsseln.

LUKS Laufwerke via USB entschlüsseln

Jetzt haben wir die **luks.key** Datei, fügen wir Sie als eigenen Keyslot zu unserem LUKS hinzu.

LUKS Laufwerke via USB entschlüsseln

Was ist den eigentlich „unser Luks“?

LUKS Laufwerke via USB entschlüsseln

Damit sind die **Partitionen** gemeint,
die mit **LUKS** verschlüsselt sind.

LUKS Laufwerke via USB entschlüsseln

In **diesem** Beispiel ist das genau **eine** Partition:

`/dev/nvme0n1p3`

LUKS Laufwerke via USB entschlüsseln

Einfacher geht es nicht:

```
cryptsetup luksAddKey /dev/nvme0n1p3 /root/luks.key
```

Das Format ist immer: `cryptsetup luksAddKey <DEVICE> <KEYFILE>`

LUKS Laufwerke via USB entschlüsseln

Ihr müsst das **LUKS-Passwort** eingeben, um diesen Key dem LUKS hinzufügen,

nicht Eurer ROOT- oder Benutzerpasswort!

LUKS Laufwerke via USB entschlüsseln

Schauen wir mal nach, ob es geklappt hat:

```
cryptsetup luksDump /dev/nvme0n1p3
```


LUKS Laufwerke via USB entschlüsseln

Keyslots:

0: luks2

Key: 512 bits

Priority: normal

Cipher: aes-xts-plain64

Cipher key: 512 bits

PBKDF: argon2id

...

1: luks2

Key: 512 bits

Priority: normal

Cipher: aes-xts-plain64

Cipher key: 512 bits

PBKDF: argon2id

...

LUKS Laufwerke via USB entschlüsseln

Vor unserer Aktion, war da nur ein Slot ;)

LUKS Laufwerke via USB entschlüsseln

Wenn Ihr mehrere Schlüssel benutzen wollt, weil mehrere Personen den PC entschlüsseln sollen, dann hängt es stark von Eurer Organisationsform ab, was für Euch das Beste ist.

- a) den Key auf alle USB Sticks schreiben
- b) mehrere verschiedene Schlüssel erzeugen

LUKS Laufwerke via USB entschlüsseln

Szenario B: Die Firma

Für eine Firma macht es Sinn, dass berechtigte Mitarbeiter einen eigenen Schlüssel bekommen, weil man dann einfach diesen einen Schlüssel aus dem LUKS entfernt, wenn diese das Unternehmen verlassen.

LUKS Laufwerke via USB entschlüsseln

Szenario B: Die Firma

Dazu ist es ganz wichtig, die Keys gut zu beschriften, damit Ihr den Schlüssel entweder über die Keydatei oder über den richtigen Slot(**Nummer notieren!**) entfernt.

LUKS Laufwerke via USB entschlüsseln

über den Slot entfernen:

```
cryptsetup luksKillSlot /dev/nvme0n1p3 1
```

oder über die Keydatei:

```
cryptsetup luksRemoveKey /dev/nvme0n1p3 luks.key
```

LUKS Laufwerke via USB entschlüsseln

Das hat den riesigen **Vorteil**,

dass man die anderen Schlüssel nicht neu vergeben muß, was das Management stark vereinfacht.

LUKS Laufwerke via USB entschlüsseln

Szenario A: Die Kopie

```
dd if=/root/luks.key of=/dev/sda bs=512 seek=1 count=16
```

Schon weil USB Sticks mal kaputt gehen, sollte man eine Kopie anlegen.

LUKS Laufwerke via USB entschlüsseln

Schritt 2:

Die `/etc/crypttab` ändern

LUKS Laufwerke via USB entschlüsseln

Jetzt kommen wir **zum Eingemachten**, weil wir müssen dem **InitramFilesystem** auch noch sagen, dass es überhaupt nach diesem **USB Key** suchen soll um den zu benutzen.

LUKS Laufwerke via USB entschlüsseln

Normal sieht Eure `/etc/crypttab` so aus:
(`<UUID>` ist ein Platzhalter)

```
luks-<UUID> UUID=<UUID> none discard
```

LUKS Laufwerke via USB entschlüsseln

Das meint:

Suche nach Devmapper luks-**<UUID>**
(ist redundant) mit der Partitions-ID **<UUID>**
benutze kein (**none**) Keyfile
und benutze für die SSDs einen Spezialmodus
(**discard**) für beschleunigte Routinen.

LUKS Laufwerke via USB entschlüsseln

Das müssen wir wie folgt ändern:

```
Luks-<UUID> UUID=<UUID> /dev/sda luks,tries=3,keyfile-size=8192,keyfile-offset=512,keyfile-  
timeout=20,nofail  
luks-<UUID> UUID=<UUID> none discard,force
```

LUKS Laufwerke via USB entschlüsseln

Hier ein Beispiel:

```
luks-1b4fec65-8c51-4149-8328-edba383243cd UUID=1b4fec65-8c51-4149-8328-edba383243cd /dev/sda luks,tries=3,keyfile-  
size=8192,keyfile-offset=512,keyfile-timeout=20,nofail  
luks-1b4fec65-8c51-4149-8328-edba383243cd UUID=1b4fec65-8c51-4149-8328-edba383243cd none discard,force
```

LUKS Laufwerke via USB entschlüsseln

Die Angaben meinen das hier:

`/dev/sda` ist unser USB Stick

`luks` kann mehr als nur LUKS, deswegen sagen wir es explizit

`tries=3` 3 Versuche ein Passwort zu bekommen (***Achtung: gilt GLOBAL!!!**)

`keyfile-size=8192` Wie lang ist der Key? (unsere 16 Blöcke)

`keyfile-offset=512` Wo finde ich denn den Key auf dem USB Stick? (Block 1)

`keyfile-timeout=20` Wie lange soll ich auf den Stick warten?

`nofail` wenn der Stick nicht da ist, ist das kein Beinbruch, mach einfach weiter!

LUKS Laufwerke via USB entschlüsseln

Hinweis:

Ja, ihr müßt zwei Zeilen für jede Partition angeben, weil sonst nur die erste beachtet wird.

Wegen eines Fehlers in älteren Systemd Versionen, wurde nur die erste Zeile ins initramfs kopiert, daher braucht es die undokumentierte „force“ Option, damit beide reingeschrieben werden.

In zukünftigen Systemd Versionen könnte das Logikproblem mit dem Failback auf Passwort behoben sein, dann reicht vermutlich eine Zeile aus.

LUKS Laufwerke via USB entschlüsseln

Und nun noch das **Initramfs** für den aktuellen Kernel neu bauen:

```
dracut --force
```

--force muß sein, **weil**

es schon ein File vom Kernelinstall gibt, das überschrieben werden muß.

LUKS Laufwerke via USB entschlüsseln

FERTIG!

LUKS Laufwerke via USB entschlüsseln

***Achtung: gilt global**

LUKS Laufwerke via USB entschlüsseln

Die **maximale** Anzahl an **Versuchen** schließt Keys von USB Sticks und die Anzahl der Versuche für die Passworteingabe ein.

Wenn man da **tries=1** benutzt, wird **NIE** nach dem Passwort gefragt, auch wenn man die Passwortanweisung mit einträgt, weil der eine Versuch vom USB Stick aufgebraucht wurde.

LUKS Laufwerke via USB entschlüsseln

Die Handhabung des USB Sticks

Man sollte den Key entweder als „normalen“ USBStick tarnen z.B. auf die Partition Daten einspielen und dann unter 20 gleichen verstecken, **oder** ihn sich nach dem Boot wieder um den Hals hängen bzw. in den Tresor legen.

LUKS Laufwerke via USB entschlüsseln

Der Satz: „Weiß doch keiner.“

zieht hier bestenfalls, solange Ihr Zuhause ausgeraubt werdet, **aber** bei einer **Hausdurchsuchung** oder Einbruch in die Firma, werdet Ihr nicht so viel Glück haben.

Deswegen: Achtet auf den Stick!

LUKS Laufwerke via USB entschlüsseln

Stand 13.10.2024:

Fedora Kernel Updates zerstören die Initramfs
Fähigkeit den USB Key zu benutzen.

Siehe: https://bugzilla.redhat.com/show_bug.cgi?id=2318294

LUKS Laufwerke via USB entschlüsseln

In wie weit das für andere Distros gilt,
ist unbekannt.

LUKS Laufwerke via USB entschlüsseln

Ein praktisches Beispiel gab es **LIVE** bei

Linux am Dienstag

LUKS Laufwerke via USB entschlüsseln

Version 1.01

- ein „seek“ durch „skip“ ersetzt
- Hinweise zu „sda“ und „crypttab“ eingefügt.