Kein Klartext, nirgends

encfs als flexibles Werkzeug für verschlüsselte Dateisysteme

> Der RHZ-Schwerpunkt zu Werkzeugen für sichere EDV ist *die* Ausrede, mal etwas echten Nerd-Kram unterzubringen – encfs, ein pfiffiges kleines Tool, das auf Linux-Systemen (bzw. allen, die fuse können) kleinräumige Verschlüsselung erlaubt. Wenn ihr vor der Shell keine Angst habt, lest weiter. Ansonsten dürft ihr für dieses Mal mit gutem Gewissen weiterblättern.

Zwar machen mittlerweile die meisten Betriebssysteme die Einrichtung systemweiter Verschlüsselung der Platte auch technisch weniger versierten Menschen zugänglich – aber was ist, wenn den Bullen die entschlüsselte Platte in die Hände fällt, vielleicht, weil der Rechner gerade lief? Und wie ist das mit USB-Sticks oder Netzwerkspeicher? Oder Speicher auf der Kiste an der Uni, die ihr nicht in der Hand habt?

Für solche Zwecke gibt es filebasierte Verschlüsselung. Sie ist aus vielen Gründen (etwa Informationslecks durch Metadaten, unverschlüsselte Zwischendateien, einfachere Angriffe auf Datenintegrität) kein vollständiger Ersatz für die datentägerweite Verschlüsselung. Umgekehrt ist sie relativ einfach aufzusetzen, und die Möglichkeit, jeweils nach Aufgabe nur den Bereich der Platte zu entschlüsseln, der gerade gebraucht wird und dann wieder verschlüsseln zu lassen, kann selbst in Szenarien, bei denen die Angreifer_innen sich ein entschlüsseltes System ertricksen können, Schlimmeres verhindern. Extrabonus: Die Container filebasierter Verschlüsselung – im Wesentlichen schlichte Unterverzeichnisse – sind beim Backup ungefähr ebenso effizient wie unverschlüselte Verzeichnisse.

Deshalb wollen wir hier etwas Werbung für encfs machen (aber auch hinweisen auf bestehende Kritikpunkte aus einem Audit von 2014[1]; Fazit davon in etwa: speichert keine Anschlagserklärungen mit encfs auf dropbox).

Encfs basiert auf einer Technik namens Filesystem in Userspace (fuse), was heißt, dass ihr nicht root sein müsst, um damit verschlüsselte Bereiche auf der Platte anzulegen; auf vielen Systemen (etwa Debian und Ableitungen) müsst ihr allerdings Mitglied der Gruppe fuse sein, um es nutzen zu können (dazu müsst ihr einmalig etwas wie `sudo adduser ‘id -nu‘ fuse` laufen lassen).

Ihr braucht dann einen Mountpoint, also ein Verzeichnis, unter dem euer verschlüsselter Kram erscheinen soll; hierzu für alle verschlüsselten Container jeweils das gleiche Verzeichnis zu verwenden (sagen wir, `~/vorsicht`), ist keine ganz schlechte Strategie, denn dann könnt ihr Programme wie z.B. vim relativ einfach so konfigurieren, dass sie für Dateien dort keine Spuren im Home erzeugen, und ihr könnt das Klartext-Verzeichnis von Backup und Indizierung ausschließen.

Nach einmaligem Anlegen des Mountpoints (ggf. `mkdir ~/vorsicht`) ist das Erzeugen *und* Einhängen eines verschlüsselten Containers ein Einzeiler:

```
encfs ~/.ufallkram ~/vorsicht
```

Beim Erzeugen des Containers wird euch encfs verschiedene Fragen stellen; die Voreinstellungen ist in der Regel ok, wenn ihr Genaueres wissen wollt, erklärt die Manpage recht ausführlich, was die einzelnen Antworten bedeuten.

Nach diesem Kommando könnt ihr im Unterverzeichnis `vorsicht` ganz normal basteln, und encfs sorgt dafür, dass in Wirklichkeit verschlüsselte Daten auf der Platte landen, und zwar im Beispiel im Unterverzeichnis `.ufallkram`. Das `~/` vor den Namen bedeutet im Groben, dass sich die Namen auf euer Home beziehen; encfs braucht an dieser Stelle aus technischen Gründen absolute Pfade.

Wenn ihr fertig seid mit der, sagen wir, U-Fallbearbeitung, sagt ihr:

```
fusermount -uz ~/vorsicht
```

Danach ist das Verzeichnis `vorsicht` leer, und es gibt nur noch die verschlüsselten Daten in `.ufallkram`. Wenn ihr das encfs-Kommando von oben wieder ausführt und das richtige Passwort eingebt, sind eure Daten aber wieder unter `vorsicht` zu finden.

Nun müssen diese Container nicht im Home liegen (und die Mountpoints auch nicht). Ihr könnt auch USB-Sticks nehmen zur Speicherung der verschlüsselten Daten oder, wenn es sein muss, Zeug wie dropbox. Besonders attraktiv ist die Kombination mit einem anderen fuse-Dateisystem, nämlich sshfs. Letzteres erlaubt, ein entferntes Dateisystem auf einer Maschine, die ihr per ssh erreichen könnt, als lokales Dateisystem einzubinden – und darauf könnt ihr dann wiederum ein encfs laufen lassen. Auf so einer Konstruktion will mensch wahrscheinlich keinen Videoschnitt machen, für das klassische Szenario von U-Fall-Bearbeitung durch mehrere Leute jedoch reicht es überall hin.

Um etwas dabei zu helfen, auch immer dran zu denken, Kram zu unmounten, den ihr nicht mehr braucht, empfiehlen sich Shellskripte. Bei uns hat sich etwas bewährt wie:

```
#!/bin/bash
encfs ~/.ufallkram ~/vorsicht || exit 1

cleanup() {
  fusermount -uz ~/vorsicht
}
trap cleanup EXIT

export BASH_POST_RC="PS1='ACHTUNG> '"
(cd ~/vorsicht; bash)
```

Was das tut: Es lässt zunächst unser mount-Kommando laufen und gibt gleich auf, wenn dabei etwas nicht klappt. Wenn es aber klappt, installiert es eine Aufräum-Funktion, die das unmounten übernimmt, sobald das Skript beendet wird. Die Buchstabensuppe mit export am Anfang sorgt im Folgenden für einen Prompt, der andeutet, dass ihr gerade kritisches Zeug gemountet habt. Schließlich startet das Skript eine Shell, in der die entschlüsselten Dateien ganz normal sichtbar sind. Verlässt mensch die Shell, läuft die Aufräumfunktion und es gibt keinen Klartext mehr.

Mit ein paar Handgriffen könnt ihr das Skript erweitern, um bei entschlüsselten Dateisystemen z.B. noch zusätzlich LEDs blinken zu lassen (das ginge, indem ihr beim reinlaufen `heartbeat` und in cleanup `none` in eine der Dateien `/sys/class/leds/*/trigger` schreibt; allerdings müsst ihr dann Rechte managen, denn das darf per default nur root) oder vielleicht den Bildschirmhintergrund zu ändern (xsetroot -solid red) oder was immer für euch funktioniert.

Ihr könnt jeweils ein Skript dieser Art für die verschiedenen Dinge haben, die ihr so mit dem Rechner macht; wer gerne aus Shells heraus arbeitet, hat so eine recht unaufdringliche Verschlüsselungsschnittstelle, die zumindest üblichen Behörden- und Naziangriffen widerstehen dürfte. Und speziell wer meint, gar nicht auf Dropbox verzichten zu können: Nie ohne encfs.

Datenschutzgruppe der Roten Hilfe Heidelberg

[1] https://defuse.ca/audits/encfs.htm