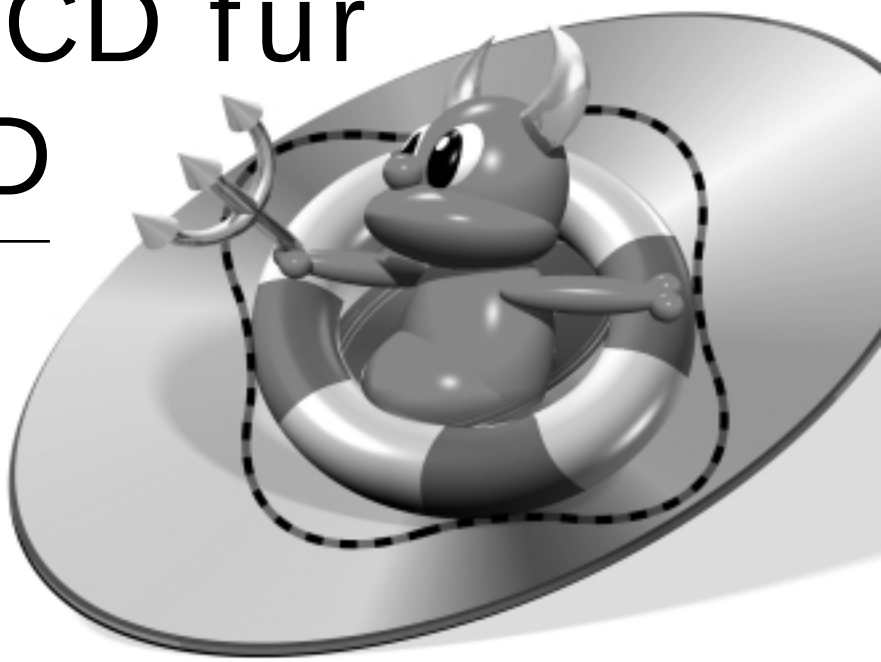
Rettungs-CD für SecureBSD



VON JÖRG BRAUN

In der letzten Ausgabe der freeX wurde die Installation von FreeBSD mit einem Schlüsselstick auf einer vollverschlüsselten Festplatte beschrieben. Was aber, wenn der Bootstick kaputtgeht?

Es gibt eine Reihe Möglichkeiten, ein Betriebssystem so abzusichern, daß im ausgeschalteten Zustand niemand an die Daten auf der Festplatte gelangt. Eine der sichersten ist die in der letzten Ausgabe der freeX (siehe [1]) detailliert beschriebene Variante einer FreeBSD-Installation, bei der das Festplattenpaßwort gegen einen USB-Stick entschlüsselt werden muß. Der eigentliche Sinn des Ganzen ist, daß selbst mit bekanntem Paßwort ohne den Stick kein Zugriff möglich ist und auch mit dem Stick erst noch das Paßwort geknackt werden muß. Ist das Paßwort ausreichend lang und gut gewählt, zum Beispiel nach einer Auswahl aus den Vorschlägen des Programms *apg*, kann eigentlich nichts schiefgehen: Die Festplatte des PCs ist ohne den Schlüssel nicht zu booten.

Der zerbrochene Schlüssel

Die eigentliche Schwachstelle des ganzen Systems ist der Schlüssel. Wenn er zerbricht, gelangt man mehr nicht an die Daten. Aus diesem Grund sollte man immer einen Notfall-Stick in Reserve haben. USB-Sticks haben jedoch die Eigenschaft, nicht sehr lange zu halten (auch wenn Hersteller wie Transcend 30 Jahre Garantie geben) und natürlich verabschieden sie sich, falls sie kaputtgehen, von einem Moment auf den anderen. Ein weiteres Problem bei USB-Sticks ist, daß zwar theoretisch jeder bootfähig gemacht werden kann, es dann aber noch lange nicht gesagt ist, daß der Ziel-Notebook auch wirklich von diesem speziellen Medium bootet. Das geht nur, wenn am Anfang keine längeren Timeouts auftreten.

Die ideale Lösung für den Notfall ist deshalb eine CD oder CD-R/W, die zwar natürlich auf keinen Fall überallhin mitgenommen werden sollte, aber an sicherer Stelle verwahrt, zum Beispiel im Schreibtisch in der Firma oder – soweit vorhanden – im Tresor, als Notfallmedium herangezogen werden kann. Ist der USB-Stick defekt, wird von dieser CD gebootet und dann aus dem abgesicherten Betriebssystem von einem Image schnell und bequem ein neuer Stick geschrieben. Das geht natürlich am einfachsten, wenn vorher vom ursprünglichen und funktionsfähigen USB-Stick ein Image gezogen wurde.

Die CD braucht nicht unbedingt mit den neuesten Daten gebootet zu werden und sollte nur bei Versionssprüngen im Betriebssystem aktualisiert werden, damit die Binärdateien des Userlands auf der Festplatte noch mit dem Kernel zusammenarbeiten.

Eine bootfähige Rettungs-CD für das abgesicherte FreeBSD-System anzulegen ist nicht schwer. Benötigt werden für ein Minimalsystem nur die beiden Verzeichnisse */boot* und */etc* des USB-Schlüssels und selbst diese Daten können noch von aktuellem Ballast befreit werden.

Am einfachsten legt man als Vorbereitung für das Produzieren der Boot-CD unter */home* oder im Arbeitsverzeichnis des Administrators unter */root* (je nachdem, wo genug Platz ist) ein neues Masterverzeichnis (beispielsweise mit dem Namen *bootcd*) an. In dieses Verzeichnis kopiert man alle Daten aus den beiden angesprochenen Verzeichnissen. Hat man die Daten auf der Platte synchron zum Bootstick gehalten, kann man sie von dort kopieren, sonst muß man den USB-Stick einbinden und sich die Dateien von dort holen. Achtung bei der */etc/fstab*: Hier muß die Datei vom USB-Stick auf die CD gelangen, nicht die von der Festplatte! Kopiert man mit *cp*, dürfen die Parameter *-R* (rekursiv die Unterverzeichnisse mitnehmen) und *-p* (Dateirechte erhalten) nicht fehlen. Einfacher und bequemer überträgt man die Daten mit dem Midnight Commander. Natürlich benötigt man für die Arbeiten auch Administratorrechte, die man sich mit *su* global oder mit *sudo* bei jedem Aufruf holen sollte.

Angenommen, als Masterverzeichnis wurde */home/bootcd* gewählt, dann

sollten sich nach der Kopieraktion hier nur die beiden Unterverzeichnisse *boot* und *etc* befinden. Jetzt wird nach *boot/kernel* gewechselt. Hier werden alle Symboldateien mit *rm *symbol* gelöscht. Weiterhin dürfen aus dem lokalen Verzeichnis *boot* noch die Verzeichnisse und Dateien *firmware/*, *zfs/*, *zfsboot* und *zfsloader* entfernt werden. Wer ganz radikal Müll entsorgen will, sollte im laufenden Betrieb *kldstat* aufrufen und nachsehen, welche Module außer dem Kernel geladen sind. Alles andere kann dann ebenfalls entfernt werden, was enorm Platz spart. Löscht man im Kernelverzeichnis selbst, darf man das anschließende *kldxref*. (also auf das aktuelle Verzeichnis ..../*boot/kernel*) aber nicht vergessen.

Nur das Minimum

Das Ausmisten in *etc* spart nicht so viel Platz, es reicht aber völlig, wenn sich zum Schluß hier noch die Verzeichnisse *defaults/* und *rc.d/* sowie die Dateien *devd.conf*, *devfs.conf*, *fstab* (vom USB-Stick!), *rc*, *rc.conf* und *rc.subr* befinden.

Das Abspecken der CD scheint zwar auf den ersten Blick müßig, ist es aber nicht. Je kleiner der Datenträger ist, desto schneller wird er geschrieben und auch wieder beim Booten geladen. Gerade beim Systemstart von CD macht sich die Größenoptimierung durch das Verringern der Positionierungsmaßnahmen des Lesekopfs bei der Geschwindigkeit und auch akustisch deutlich bemerkbar.

Dieses Minimalsystem wird jetzt auf eine CD-R oder (besser) CD-R/W geschrieben. Letzteres ist deshalb günstiger, weil nach einem Systemupdate wieder auf den gleichen Datenträger aktualisiert werden kann und nachher kein jetzt überflüssiger Rohling mit der Verschlüsselungsdatei irgendwo – und sei es im Mülleimer (wohin er nicht gehört, CD-Rs sind Sondermüll!) – herumliegt. Wer mit normalen CD-Rs arbeitet, sollte aus Sicherheitsgründen den überflüssigen Datenträger nicht einfach wegwerfen, sondern physikalisch zerbrechen (oder wenigstens mit einem spitzen Gegenstand die Beschichtung stark zerkratzen) und dann erst entsorgen!

Die Minimal-Distribution wird jetzt auf den CD-Rohling gebrannt. Sind die Daten in */home/bootcd*, sollte man in */home* eine Datei anlegen, die alle Befehle vom Anlegen des Images bis zu seinem Schreiben enthält. Sie benötigt kein x-Attribut, sondern wird bei Bedarf *sh* als Parameter übergehen, zum Beispiel als

```
# cd home
# sh makecd
```

Die Datei legt über *mkisofs* aus den *cdrtools* ein ISO-Image an und brennt es auf das CD-Laufwerk in */dev/acd0*. Da nicht gewährleistet ist, daß das ATAPICD-Device im Kernel fest enthalten ist, wird das im Skript überprüft und bei Bedarf das Kernelmodul noch geladen. Zugegriffen auf das CD-Laufwerk wird mit dem FreeBSD-eigenen Programm *burncd*, das sehr viel bequemer zu bedienen ist als das komplexere (und mächtigere) *cdrecord*.

Hier das Skript:

```
#!/bin/sh

# Kernelmodul bei Bedarf laden:
if [ ! -e /dev/acd0 ] ; then
  kldload atapicd
fi

# Löschen der CD-R/W, Timeout vermeiden:
burncd -f /dev/acd0 && sleep 10

# Anlegen des ISO-Images, der Port
# sysutils/cdrtools muß installiert sein
if [ ! -f bootcd.iso ] ; then
  mkisofs -r -no-emul-boot \
      -b boot/cdboot \
      -c boot/.catalog \
      -V "SecureBSD - bitte entfernen!"\
      -p "J. Braun <lektorat@CUL.DE>"  \
      -publisher "Redaktion freeX"     \
      -o /home/bootcd.iso bootcd
fi

# Schreiben und Fixieren des
# Datenträgers, nach den Arbeiten immer
# etwas warten, um Timeouts zu vermeiden
if [ -f bootcd.iso ] ; then
   burncd -f /dev/acd0 -s 8 \
           data /home/bootcd.iso && \
   sleep 10
   burncd -f /dev/acd0 fixate && \
   sleep 10
   rm bootcd.iso
fi
```

Image vom Bootstick

Ein Sicherungsimage des Bootsticks wird ganz einfach mit dem Aufruf *dd if=/dev/da0 of=stick.img bs=2048*) in das aktuelle Verzeichnis geschrieben. Durch das Vertauschen von *if* und *of* schreibt man die Datei wieder auf denselben oder einen anderen Stick zurück. Man sollte nicht vergessen, das Sicherungsimage nach jedem Aktualisieren des Betriebssystems neu auf die Festplatte zu schreiben.

Bei Bedarf kann das Skript noch mit Meldungen verschönert werden, notwendig ist das aber nicht, weil beide Tools nicht mit Ausgaben geizen. Heikel ist es nur, wenn ein I/O-Control-Fehler wie

```
burncd: ioctl(CDRIOCWRITESPEED): Device
busy
```

angezeigt wird.

Das liegt immer daran, daß der Brenner noch beschäftigt ist, wenn bereits der nächste Vorgang angestoßen werden soll. In diesem Fall muß man die Wartezeiten zwischen den Aufrufen verlängern.

Mini-CD

Der geschriebene Datenträger ist nur 43 MByte groß; hat man das Kernelverzeichnis selbst gereinigt, bleiben sogar – abhängig von der Hardware und der Konfektionierung des Kernels selbst – nur noch 11 bis 15 MByte an Nutzdaten übrig. Dies ist dann theoretisch gleichzeitig auch die minimale Größe des Schlüsselsticks selbst. Aus zwei guten Gründen sollte man aber so kleine USB-Sticks meiden. Der erste ist, daß alle Tests von älteren Sticks dieser Größe, egal ob USB 1.1 oder USB 2.0, scheiterten, weil der Stick nicht als bootfähig erkannt wurde. Das kann hardware-bedingt sein und am Testnotebook der Redaktion liegen. Dort war zuverlässiges Booten nur mit USB-Sticks der letzten beiden Jahre mit mindestens 2 GByte Größe möglich. Bei gekauften neueren und größeren USB-Sticks gab es nie Ärger,

X, der D-Bus und CDs

Das manuelle Einbinden von Datenträgern in FreeBSD kann umständlich und lästig sein, speziell, wenn man brav mit einem normalen User-Account unter Gnome oder auch KDE arbeitet. Es ist zwar möglich, das System so einzustellen, daß mit Usermounts gearbeitet werden kann, richtig bequem ist das aber auch nicht. Die modernen grafischen Oberflächen bieten da echten MacOS- und Linux-Komfort, dazu darf das CD-Laufwerk aber nicht in der */etc/fstab* eingetragen sein.

Wie jede Komfortlösung hat aber auch diese ihre Haken und Ösen. Handelt es sich nämlich bei dem eingebauten Laufwerk nicht um einen reinen CD/DVD-Leser, sondern um einen Brenner, poppt beim Löschen eines R/W-Datenträgers in der Shell die Meldung hoch, daß es sich um ein leeres R/W-Medium handle, das bereit wäre, Daten entgegenzunehmen. Leider bricht der Schreibversuch mit *burncd* also ab, wenn das ATAPI-Gerät als Leser nicht eingerichtet war und deshalb auch nicht in der */etc/fstab* steht. Um sinnvoll mit dem Brenner arbeiten zu können, müssen deshalb die Mount-Einträge für *cd0* (ATAPICAM) und *acd* (ATAPICD) beide in der fstab stehen.

Möchte man keine Datenträger brennen oder ist kein Brenner eingebaut, kann man den von Linux und seinem D-Bus stammenden Komfortlösungen arbeiten. Besser und BSD-gemäßer als das ist aber das Einrichten des BSD Automouters, was unter anderem in [4] beschrieben ist, er hat keine solchen unter Umständen lästigen Nebenwirkungen.

Eine anderer Workaround ist, daß die hier beschriebenen Arbeiten auf der mit [Strg][Alt][F1] zu erreichenden Textkonsole durchgeführt werden und nicht in einem X-Fenster. Da Datenträger dann nur automatisch eingebunden werden, wenn jemand in einen der modernen X-Desktops eingeloggt ist, sollte man sich vorher auf dem GDM- oder KDM-Loginbildschirm zurückziehen.

aber aktuelle Werbesticks waren ebenfalls unbrauchbar. Der zweite und mindestens ebenso wichtige Grund für die Wahl eines größeren Bootmediums ist, daß bei den kleinen Medien kein Kernel-Update aus den Quellen durchgeführt werden kann, denn dabei werden alle Symboldateien kopiert. Dafür benötigt man mindestens 256 MByte Speicherplatz. Am USB-Stick zu sparen ist generell nicht gut. Die Notfall-CD ist auch nur für wirkliche Notfälle gedacht und sollte niemals im Laufwerk verbleiben, sie gehört in den Tresor!

Für den Notfall

Zum Schluß noch der Hinweis, daß sich diese Notfall-CD natürlich auch für die normale Festplatten-Installation ohne Verschlüsselung eignet, aber nur dann, wenn ausschließlich der Kernel zerschossen ist. Da sich keine Hilfsprogramme auf der CD befinden, kann ohne die Festplattenbootpartition nicht in den Single User Mode gebootet werden.

Wie beim USB-Stick benötigt die CD dann eine *fstab* mit der ausschließlichen Anweisung, das Betriebssystem von der Festplatte und eben nicht vom Datenträger mit dem Kernel zu starten. Sie lautet bei einer verschlüsselten Installation von FreeBSD in der ersten Partition der ersten IDE-Festplatte (die erste SATA-Festplatte trägt bei FreeBSD die Kennung *ad4*):

```
/dev/ad0s1.elia    /    ufs    rw    1    1
```

Bei einem unverschlüsselten System wird das ».elia« durch das übliche »a« ersetzt. Man braucht die CD aber eigentlich bei einer unverschlüsselten FreeBSD-Installation nicht, da man sich in diesem Fall mit der Live-Reparatur-CD, Livesystemen wie GhostBSD (siehe freeX 1/2011) oder auch mit einer regulären Installation auf einem USB-Stick helfen kann.

Die Notfall-CD ist eine weitere Stufe auf dem Weg zum sicheren Unix-PC. Das Anlegen geht so wie bisher beschrieben jedoch nur, wenn im Gerät auch ein ATAPI-CD-Brenner eingebaut ist. Wird der Brenner extern über USB angeschlossen, funktioniert im Prinzip alles genauso, außer daß beim Brennen anstelle von *burncd* das Programm *cdrecord* aus den CDRTools aufgerufen werden muß. Ihm wird das ATAPICAM-Device übergeben, das man vorher mit *cdrecord -checkdrive* ermittelt. Die Brenngeschwindigkeit muß mit *speed = < Zahl >* übergeben werden, das ISO-Image wird mit *-dao* als Disk at Once definiert und dahinter wird der Name der ISO-Datei angegeben. Die komplette Kommandozeile für das Brennen der CD lautet dann für das ermittelte Device 0

```
cdrecord -v speed=8 dev=0,0,0 blank=fast
cdrecord -v speed=8 dev=0,0,0 \
         -dao bootcd.iso
```

Die CD-R/W wird dabei vor dem Brennen erst einmal gelöscht. ◆

Literatur

[1] Jörg Braun und Jürgen Dankoweit: Festplatte mit Schlüssel. freeX 1/2011, S. 22-30.

[2] Jürgen Dankoweit: Sichtschutz für Dateien. freeX 1/2010, S. 14-21, C&L Verlag.

[3] Jörg Braun: Journaling mit UFS. freeX 6/2009, S. 92-94, C&L Verlag.

[4] Jürgen Dankoweit (Hrsg.): FreeBSD. C&L-Verlag 2009, ISBN 978-3936546-41-5.