c't magazin für computer technik

LUKS: Verschlüsselte Linux-Systeme über SSH entsperren

| 10.08.2021 15:00 Uhr Keywan Tonekaboni

Eine verschlüsselte Linux-Systempartition schützt die gespeicherten Daten. Das geht bequem übers Netzwerk. Bei Debian, Ubuntu & Co. hilft dabei ein Paket.

Das gesamte System zu verschlüsseln, ist grundsätzlich empfehlenswert. Und sei es nur, damit Unbefugte auf entsorgten Datenträgern keine persönlichen Dateien finden. Unter den meisten Linux-Distributionen lässt sich die Verschlüsselung mit LUKS komfortabel direkt bei der Installation einrichten, auch für die Systempartition.

Was auf dem PC oder Laptop eine Lappalie ist, rangiert auf dem Server zwischen nervig und anspruchsvoll: die Eingabe des LUKS-Passworts während des Bootvorgangs. Statt hinter den heimischen Server zu krabbeln, um Tastatur und Display anzuschließen, oder sich mit WebVNC-Konsolen eines Hosters herumzuschlagen, geht es auch ganz bequem per SSH, egal ob vom Linux-Rechner, dem Windows-PC oder vom Mac aus.

MEHR ZU LINUX

- **Linux: Snap-Pakete für eigene Anwendungen erstellen [1]**
- **Scanner unter Linux nutzen [2]**
- **Der optimale PC: Linux auf den aktuellen PC-Bauvorschlägen einsetzen [3]**
- **Linux-Treiber: Warum das manchmal so mühsam ist [4]**
- **Linux-Kernel mit Systemd-Boot gekonnt starten [5]**
- **Linux barrierefrei einrichten für motorische Behinderungen [6]**

- **Linux mit mehreren Monitoren betreiben [7]**
- **Fedora 33 im Test: Neue Vorgaben mit Btrfs, Systemd-Resolved und zRAM [8]**
- **Linux: Flatpak-Pakete selbst bauen [9]**

Das Prinzip ist schnell erklärt: Während des Bootvorgangs startet ein Linux-System aus dem Initramfs-Image, das Hilfsprogramme und Treiber enthält. Steckt darin ein minimaler SSH-Server, kann man sich mit diesem verbinden und in der SSH-Sitzung die verschlüsselte Systempartition entsperren. Nach erfolgreicher Passwort-Eingabe beendet der SSH-Server die Verbindung, und das System setzt den Start wie gewohnt fort.

Da die Nutzer- und Passwortdaten auf der verschlüsselten Systempartition beim Booten noch nicht zugänglich sind, authentifiziert der SSH-Server die Nutzer stattdessen anhand eines Public-Key-Verfahrens mit vorher im Initramfs hinterlegten SSH-Keys. Durch die verschlüsselte SSH-Verbindung wird das LUKS-Kennwort sicher übertragen.

Die folgende Anleitung erklärt die notwendigen Befehle Schritt für Schritt beispielhaft unter Linux. Mit Server ist der Computer gemeint, der entschlüsselt werden soll, der PC ist der persönliche Computer oder Laptop, von dem aus man sich zum Entsperren einloggt.

[10]

Server und öffentlicher Schlüssel

Server-Installation

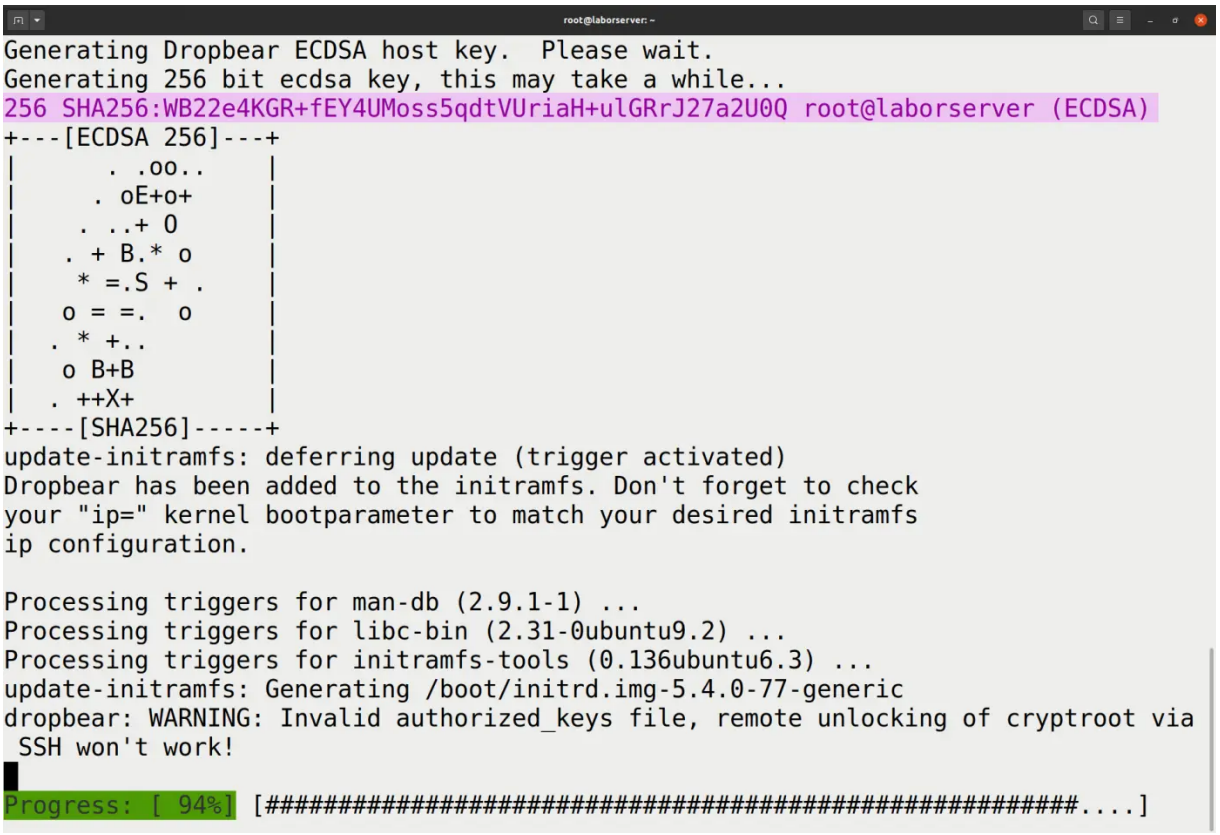
Debian, Ubuntu und davon abgeleitete Linux-Systeme bringen in ihren Repositories das Paket "dropbear-initramfs" mit, das die notwendigen Bestandteile enthält. Anderen Distributionen fehlt so ein vorkonfiguriertes Paket leider.

Installieren Sie zunächst auf dem Server das Paket dropbear-initramfs:

```
sudo apt install dropbear-initramfs
```

Während der Installation erzeugt Dropbear die benötigten Schlüssel (Host-Keys) und gibt für jeden Schlüssel je

einen Fingerprint und eine wirre ASCII-Grafik aus. Mit den ausgegebenen Fingerprints kann man beim Aufbau der SSH-Verbindung den Host-Key des Servers verifizieren. Die ASCII-Grafik ist als visuelle Hilfe gedacht, aber wird für diese Anleitung nicht weiter benötigt. Die Installation endet mit einer Warnung, dass es wegen einer fehlerhaften authorized_keys-Datei nicht möglich sei, die Systempartition aus der Ferne zu entsperren. Ignorieren Sie die Warnung vorerst.



Während der Installation zeigt Dropbear die Fingerprints der SSH-Host-Keys an, mit denen man später die Verbindung verifizieren kann.

Notieren Sie sich den Fingerprint des ECDSA- oder RSA-Schlüssels. Die Unterschiede zwischen den Schlüsselverfahren haben wir in dem Artikel "**Sichere Kommunikation: Symmetrische und asymmetrische Verschlüsselung [11]** " erläutert. Der Fingerprint wird jeweils über der ASCII-Grafik angezeigt. Er beginnt mit der Schlüssellänge und dem Hash-Verfahren, gefolgt von einem Buchstabensalat (zum Beispiel "2048 SHA256:sqPNOb/CgTSCdK6...") und endet mit den Daten des Servers und dem verwendeten Algorithmus ("root@laborserver (RSA)").

Als Nächstes müssen Sie den öffentlichen Teil Ihres SSH-Schlüsselpaares in der Dropbear-Konfiguration hinterlegen. Diesen Public-Key finden Sie auf dem PC in Ihrem Benutzerverzeichnis im Unterordner .ssh, also unter Linux sowie macOS ~/.ssh oder %HOMEPATH%\.ssh\ bei Windows. Der Public-Key hat einen Dateinamen wie id_rsa.pub, id_ecdsa.pub oder ähnlich.

Wenn Sie noch kein persönliches SSH-Schlüsselpaar haben, dann generieren Sie dieses auf der Kommandozeile mit `ssh-keygen`. Dropbear unterstützt nur Schlüssel vom Typ DSA, RSA und ECDSA. DSA gilt zwar noch als sicher, ist aber veraltet. Üblicherweise erzeugt `ssh-keygen` ein RSA-Schlüsselpaar. Bevorzugen Sie lieber einen schlanken ECDSA-Schlüssel, geben Sie dies mit der Option -t an:

```
ssh-keygen -t ecdsa
```

Zunächst fragt der Befehl den gewünschten Speicherort ab, wobei Sie die Vorgabe /home/USER/.ssh/id_ecdsa unverändert mit Enter bestätigen können. Daraufhin fragt `ssh-keygen` nach einem Kennwort (Passphrase), mit dem der Schlüssel vor unerlaubter Verwendung geschützt wird. Diese Passphrase sollten Sie festlegen. Mehr zu SSH und `ssh-keygen` steht in dem Artikel "**Mit SSH sicher auf andere Rechner zugreifen [12]** ".

Öffentlichen Schlüssel hinterlegen

Haben Sie bei der Schlüsselgenerierung als Speicherort .ssh/id_ecdsa angegeben, finden Sie den öffentlichen Schlüssel in der Datei .ssh/id_ecdsa.pub. Kopieren Sie nun den Inhalt dieser Datei. Wechseln Sie wieder auf den Server und führen Sie die folgenden Schritte per sudo oder als root mit Superuser-Rechten aus. Öffnen Sie mit einem Editor Ihrer Wahl die Datei /etc/dropbear-initramfs/authorized_keys beziehungsweise legen Sie diese neu an. Fügen Sie dort dann den kopierten Schlüssel ein, aber in einer einzigen Zeile und ohne Umbrüche. Für jeden weiteren Schlüssel legen Sie eine neue Zeile an. Die authorized_keys-Datei sieht dann grob so aus:

```
ssh-rsa AAAAB3NzaC1… cttest@laborratte
ecdsa-sha2-nistp256 AAAA… ct@labormaus
```

Die Nutzerkennung am Ende dient nur menschlichen Betrachtern zur Orientierung und wird bei Aufbau der SSH-Verbindung nicht abgeglichen oder anderweitig überprüft.

Als Nächstes legen Sie fest, dass nach einer erfolgreichen Verbindung mit dem Dropbear-SSH-Server dieser direkt den Befehl zum Entsperren der Systempartition aufruft (`cryptroot-unlock`). Öffnen Sie dazu die Datei /etc/dropbear-initramfs/config und passen Sie die Zeile mit `DROPBEAR_OPTIONS` entsprechend an:

```
DROPBEAR_OPTIONS="-c cryptroot-unlock"
```

Wer sich gut mit SSH auskennt, könnte auf die Idee kommen, den Aufruf von `cryptroot-unlock` über die authorized_keys zu erzwingen. Das klappt zunächst auch, aber wenn man sich beim LUKS-Passwort vertippt, ist man über SSH ausgesperrt. Mit der empfohlenen Methode erfragt das Skript erneut das LUKS-Passwort.

Wenn auf Ihrem Server auch ein regulärer SSH-Server läuft, sollten Sie auf Ihrem PC die SSH-Konfiguration anpassen, damit der SSH-Client dort den Dropbear-Server zum Entsperren und den regulären SSH-Server auseinanderhalten kann. Beide haben unterschiedliche Host-Keys und das würde stets für Warnungen sorgen.

Für beide SSH-Server den gleichen Host-Key zu verwenden, ist eine schlechte Idee, da der Host-Key des Dropbear-Servers unverschlüsselt im Initramfs-Archiv liegt. Außerdem speichern Dropbear und OpenSSH den Host-Key in unterschiedlichen Formaten. Sortieren können Sie dies, indem Sie den Dropbear-Server auf einem anderen Port lauschen lassen. Dafür ergänzen Sie bei `DROPBEAR_OPTIONS` die Angabe - `p 2222` , wobei 2222 für den von Ihnen frei gewählten Port steht.

IP-Adresse und Fingerprint

IP-Adresse festlegen

Mit Dropbear an Bord versucht das Initramfs-System, eine IP-Adresse per DHCP zu beziehen. Eine feste IP-Adresse können Sie dem Startprozess über eine Kernel-Option verpassen. Diese IP-Adresse ist unabhängig von der Netzwerkkonfiguration des Servers selbst. Fügen Sie für eine statische IP-Adresse in der Datei /etc/initramfs-tools/initramfs.conf die Variable `IP` hinzu, deren Inhalt nach dem Schema `IP=IP-Adresse:NFS-Server:Gateway:Netzmaske:Hostname:Device:` aufgebaut ist. Nicht benötigte Felder wie das des NFS-

Servers lassen Sie leer. Für die statische IP-Adresse 192.168.122.90 sieht die minimal notwendige Konfiguration so aus:

```
IP=192.168.122.90::192.168.122.1:255.255.255.0:
```

Wichtig sind die zwei Doppelpunkte zwischen IP- und Gateway-Adresse, da das Feld für den NFS-Server leer bleibt. Übrigens bleibt die IP-Adresse dem Server dauerhaft zugewiesen, auch nachdem das System komplett gebootet ist und seine eigene Netzwerkkonfiguration angewendet hat. Verwenden Sie daher in beiden Setups dieselbe IP-Adresse, um Probleme zu vermeiden.

Wenn Sie die SSH-Keys hinterlegt und die Konfiguration angepasst haben, müssen Sie noch das Initramfs-Image neu erzeugen:

```
update-initramfs -u
```

Diesen Aufruf müssen Sie immer dann wiederholen, wenn Sie etwas an der Dropbear-Konfiguration ändern, die SSH-Keys austauschen oder neue hinzufügen. Hat alles geklappt, sollte auch keine Warnung mehr über eine ungültige authorized_keys-Datei kommen, wie noch nach der Installation. Der Server ist nun bereit, per SSH die Systempartition zu entsperren.

```
[    3.106090] raid6: using algorithm avx512x4 gen() 19161 MB/s
[    3.106442] raid6: .... xor() 10225 MB/s, rmw enabled
[    3.107927] raid6: using avx512x2 recovery algorithm
[    3.115503] xor: automatically using best checksumming function   avx
[    3.125749] async_tx: api initialized (async)
done.
Begin: Running /scripts/init-premount ... done.
Begin: Mounting root file system ... Begin: Running /scripts/local-top ...   Volume
  Cannot process volume group ubuntu-vg
Internet Systems Consortium DHCP Client 4.4.1
Copyright 2004-2018 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/enp1s0/52:54:00:41:e1:2b
Sending on   LPF/enp1s0/52:54:00:41:e1:2b
Sending on   Socket/fallback
DHCPDISCOVER on enp1s0 to 255.255.255.255 port 67 interval 3 (xid=0xc69a367f)
DHCPOFFER of 192.168.122.90 from 192.168.122.1
DHCPREQUEST for 192.168.122.90 on enp1s0 to 255.255.255.255 port 67 (xid=0x7f369ac6)
DHCPACK of 192.168.122.90 from 192.168.122.1 (xid=0xc69a367f)
Please unlock disk dm_crypt-0: bound to 192.168.122.90 -- renewal in 1377 seconds.
Begin: Starting dropbear ...
```

Mit Dropbear-Initramfs wird recht früh im Bootprozess eine IP-Adresse per DHCP geholt. Das System akzeptiert das LUKS-Kennwort anschließend per SSH oder über die lokale Konsole.

Fingerprint

Starten Sie den Server neu und prüfen auf dem PC mit `ping IP-ADRESSE`, wann der Server wieder erreichbar ist. Sobald `ping` ein Lebenszeichen liefert, bauen Sie mit dem SSH-Client die Verbindung auf:

```
ssh -p 2222 root@192.168.122.90
```

Passen Sie die IP-Adresse entsprechend Ihrer Konfiguration an und lassen Sie die Option `-p 2222` weg, wenn

Sie keinen alternativen Port festgelegt haben. Der Dropbear-Server erwartet eine Verbindung für den User.

Bei der ersten Verbindung fordert der SSH-Client auf, den Fingerprint zu bestätigen. Sie sollten die angezeigte Prüfsumme mit dem bei der Installation von Dropbear ausgegebenen Fingerprint vergleichen. Bestätigen Sie, indem Sie im Terminal "yes" eintippen. Sie können stattdessen den am Anfang kopierten Fingerprint einfügen, dann vergleicht der SSH-Client die Fingerprints für Sie miteinander. Dabei fügen Sie nur den Mittelteil ("SHA256:AbC...dEf") ohne Schlüssellänge und Nutzerkennung ein. Passt der Fingerprint nicht, dann haben Sie vielleicht nur den Fingerprint des falschen Schlüssels verglichen, zum Beispiel jenen des RSA- statt des ECDSA-Keys. Akzeptierten Sie den angezeigten Fingerprint, hinterlegt der SSH-Client diesen in der Datei ~/.ssh /known_host.

Kommt die SSH-Verbindung zustande, sollten Sie die Aufforderung "Please unlock disk dm_crypt-0:" sehen. Geben Sie das LUKS-Kennwort ein. Sobald Cryptsetup die Systempartition erfolgreich entschlüsselt hat, beendet Dropbear die SSH-Verbindung. Wenn das System erfolgreich gestartet ist, können Sie sich über den OpenSSH-Server wie gewohnt anmelden. Wer nicht jedes Mal den sperrigen SSH-Befehl eingeben will, nutzt stattdessen ein Alias für den Zielhost. Wie das geht, erklärt der Kasten "SSH Abkürzungen". Das spart Tipparbeit und schont die grauen Zellen.

SSH-Abkürzungen

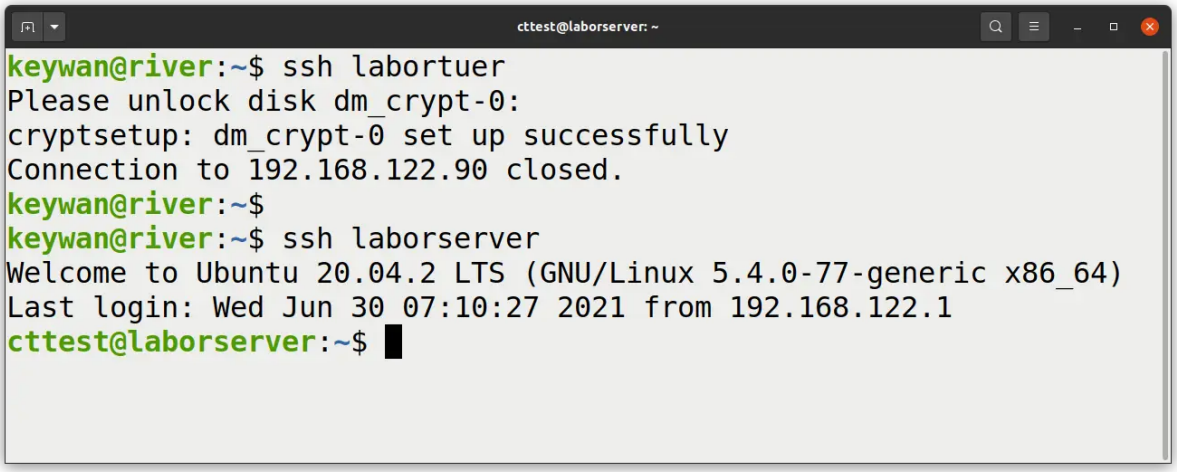
Statt bei jeder Verbindung Angaben wie User, Port und IP-Adresse lästig einzutippen, kann man für einzelne Hosts oder ganze Rechnergruppen Vorgaben für den SSH-Client festlegen. Diese Profile definieren Sie in der Datei ~/.ssh/config. Die Einstellungen schreibt man hinter den Namen der jeweiligen Option. Welche es gibt und wie die heißen, zeigt man `ssh_config` an. Das Schlüsselwort `Host` markiert einen neuen Eintrag und verpasst dem im Folgenden beschriebenen System einen frei wählbaren Namen. Alle weiteren Zeilen bis zur nächsten Host-Zeile gehören zusammen. Das Beispiel definiert den Alias "labortuer" und legt IP-Adresse, Port und Username fest:

```
Host labortuer
HostName 192.168.122.90
Port 2222
User root
```

In diesem Beispiel ist für `HostName` die IP-Adresse angegeben, die der Server erhält. Stattdessen kann man auch einen Domainnamen wie "homeserver.local" oder "mycloud.example.com" angeben. Wenn Sie nun `ssh labortuer` aufrufen, baut der SSH-Client eine Verbindung zu "root@192.168.122.90" über Port 2222 auf.

Wenn Sie wollen, können Sie die ~/.ssh/config um einen weiteren Eintrag für die SSH-Verbindung zum eigentlichen System ergänzen.

```
Host laborserver
HostName 192.168.122.90
User cttest
```



Mit der ersten SSH-Verbindung entsperrt man die verschlüsselte System-Partition. Mit der zweiten meldet man sich am System an.

Mit diesen beiden Einträgen können Sie sich mit `ssh labortuer` zum Entsperren des Passworts verbinden und mit `ssh laborserver` im vollständig gestarteten System anmelden. (**ktn [13]**)

URL dieses Artikels:

https://www.heise.de/-6157460

Links in diesem Artikel:

[1] https://www.heise.de/ratgeber/Linux-Snap-Pakete-fuer-eigene-Anwendungen-erstellen-6060442.html

[2] https://www.heise.de/ratgeber/Scanner-unter-Linux-nutzen-6052985.html

[3] https://www.heise.de/ratgeber/Der-optimale-PC-Linux-auf-den-aktuellen-PC-Bauvorschlaegen-einsetzen-5022919.html

[4] https://www.heise.de/hintergrund/Treiber-unter-Linux-Warum-das-manchmal-so-muehsam-ist-5023706.html

[5] https://www.heise.de/ratgeber/Linux-Kernel-mit-Systemd-Boot-gekonnt-starten-5024202.html

[6] https://www.heise.de/ratgeber/Linux-barrierefrei-einrichten-fuer-motorische-Behinderungen-4638557.html

[7] https://www.heise.de/ratgeber/Linux-mit-mehreren-Monitoren-betreiben-4698735.html

[8] https://www.heise.de/tests/Fedora-33-im-Test-Neue-Vorgaben-mit-Btrfs-Systemd-Resolved-und-zRAM-4983667.html

[9] https://www.heise.de/ratgeber/Linux-Flatpak-Pakete-selbst-bauen-4991018.html

[10] https://www.heise.de/ct/

[11] https://www.heise.de/hintergrund/Sichere-Kommunikation-Symmetrische-und-asymmetrische-Verschluesselung-5077465.html

[12] https://www.heise.de/ratgeber/Mit-SSH-sicher-auf-andere-Rechner-zugreifen-4160536.html

[13] mailto:ktn@heise.de