Fehlerbehebung

Andreas Teuchert

2. Februar 2015

Übersicht – Häufige Probleme

- Root-Passwort vergessen
- kein `sudo`-Zugriff
- SSH-Login nicht möglich
- fehlerhafte `/etc/network/interfaces`
- Probleme mit udev persistent net
- GRUB-Probleme

Die aufgelisteten Probleme haben verschieden starke Auswirkungen. Manche (wie Probleme mit dem SSH-Login) lassen sich ohne Neustart des Rechners beheben, für andere ist das Booten des Rechners mit einem geeigneten Rettungssystem nötig (z. B. bei vergessenem Root-Passwort). Manche können auch das System unbootbar machen und setzen ein Live-System zur Problemlösung voraus (z. B. Probleme mit dem Boot-Loader GRUB).

Im folgenden werden sowohl allgemeine als auch für spezielle Probleme geeignete Lösungsmöglichkeiten besprochen.

Lösungsmöglichkeiten

- lokaler Login auf Konsole
- nur möglich, wenn (Root)-Login noch möglich
- sonst:
- Booten mit `init=/bin/bash`
- Verwenden eines Live-Systems zur Reparatur

Liegt nur ein Netzwerk-Problem vor, lässt sich dieses nach einem Login auf der lokalen Konsole lösen. Dies setzt voraus, dass das Root-Passwort bekannt ist oder `sudo` benutzt werden kann.

Ist dies nicht der Fall, kann das System entweder direkt in eine Root-Shell gestartet werden oder ein Live-System verwendet werden.

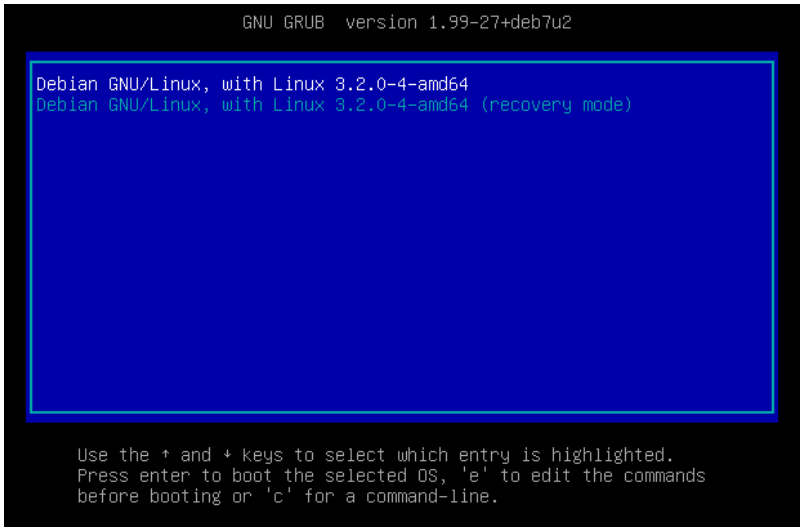
Booten mit init=/bin/bash

- beim Hochfahren des Rechners wird vom BIOS (bei neueren Rechnern UEFI-Firmware) der Boot-Loader geladen
- unter Linux üblicherweise GRUB2
- GRUB2 lädt den Kernel und das initramfs von der der Festplatte
- der Kernel bindet mit Hilfe der Programme im initramfs die Root-Partition ein und startet den Init-Prozess
- mit der Boot-Option `init=/bin/sh` startet der Kernel eine Root-Shell statt Init

Der Prozess vom Einschalten des Rechners bis zum Ende des Boot-Prozesses ist in verschiedene Abschnitte gegliedert, die die Möglichkeit bieten, den normalen Boot-Prozess zu unterbrechen.

Insbesondere werden dem Linux-Kernel beim Booten Parameter übergeben, die das Verhalten des Kernels beeinflussen. Mit dem Parameter `init` kann festgelegt werden, welches Programm als erstes nach dem Einbinden des Wurzeldateisystems gestartet wird. Dadurch kann der Kernel ohne Kenntnis des Root-Passworts zum Starten einer Root-Shell gebracht werden.

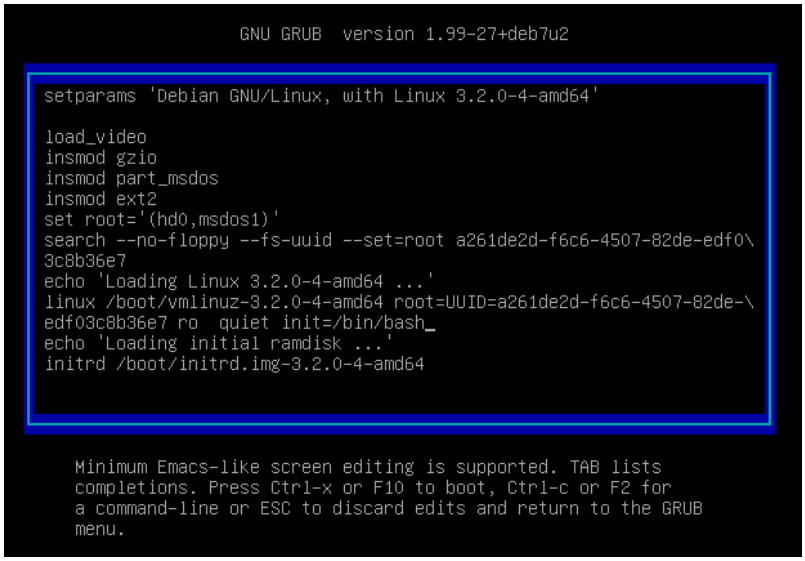
Booten mit init=/bin/bash



- `e` drücken um die Boot-Optionen zu bearbeiten

Der Screenshot zeigt den Auswahlbildschirm des GRUB-Boot-Loaders. Üblicherweise wird nach einem kurzen Timeout automatisch der Standard-Eintrag gestartet. Durch Drücken der Taste `e` kann der Boot-Vorgang unterbrochen werden und ein Editor geöffnet werden, mit dem sich die Boot-Parameter verändern lassen.

Booten mit init=/bin/bash



- an die `linux`-Zeile `init=/bin/bash` anhängen; F10 drücken

Interessant ist hier vor allem die Zeile, die mit `linux` beginnt. Diese legt fest, welche Datei als Kernel geladen werden soll und welche Optionen dem Kernel übergeben werden sollen. Durch Anhängen der Option `init=/bin/bash` wird direkt in eine Root-Shell gebootet. Beim Editieren der Optionen ist zu beachten, dass die amerikanische Tastaturbelegung eingestellt ist.

Mit der Taste F10 kann das System dann mit den veränderten Optionen gestartet werden. Die Optionen sind nur für den aktuellen Boot-Vorgang wirksam, sie werden nicht dauerhaft gespeichert.

Booten mit init=/bin/bash

- nach dem Booten wird der Prompt `root@(none):/#` angezeigt
- Achtung: Amerikanische Tastaturbelegung
- Dateisystem ist read-only gemountet
- mit `mount -oremount,rw /` schreibbar machen
- nun können beliebige Änderungen vorgenommen werden
- evtl. weitere Dateisysteme einbinden
- z. B. mit `passwd` neues Root-Passwort setzen
- danach mit `mount -oremount,ro /` wieder read-only machen
- vorsichtshalber Dateisystem mit `sync` synchronisieren
- Reboot mit `reboot -f`

Nach kurzer Zeit ist der Boot-Vorgang beendet und der Prompt wird angezeigt. Es können nun beliebige Befehle als Root ausgeführt werden. Dabei ist auf die amerikanische Tastaturbelegung zu achten. Außerdem ist das Dateisystem standardmäßig nicht schreibbar. Dies kann mit dem Befehl `mount` geändert werden.

Eventuell ist es notwendig, weitere Dateisysteme einzubinden, um daran Änderungen vorzunehmen.

Nach abgeschlossener Arbeit sollte das Dateisystem wieder read-only gemacht werden und sicherheitshalber mit dem Befehl `sync` synchronisiert werden. Dadurch wird Datenverlust beim Neustarten vermieden.

Da kein Init-Prozess läuft, kann das System nicht auf normale Weise neu gestartet werden. Daher wird dafür der Befehl `reboot -f` verwendet.

Lab 9.1: Root-Passwort ändern

- System mit `init=/bin/bash` starten
- Root-Passwort ändern

Verwenden eines Live-Systems zur Reparatur

- Booten des Rechners mit einem Live-System von CD/DVD oder USB-Stick
- z. B. Knoppix oder GRML
- Dateisysteme müssen von Hand eingebunden werden
- evtl. aufwändig bei RAID, LVM und verschlüsselten Dateisystemen
- Wechseln in das zu reparierende System mit `chroot`

Sollte das System von sich aus nicht mehr bootbar sein, kann nur noch ein Live-System zur Reparatur verwendet werden. Dieses wird üblicherweise von CD/DVD oder USB-Stick geladen. Netzboot ist aber auch möglich. Bekannte Live-Systeme sind z. B. Knoppix und GRML. Beide basieren auf Debian.

Der Nachteil von Live-Systemen gegenüber Booten mit `init=/bin/bash` ist, dass alle Dateisysteme von Hand eingebunden werden müssen. Dies ist bei RAID, LVM und verschlüsselten Dateisystemen mit etwas Aufwand verbunden.

Nach Einbinden der benötigten Dateisysteme (mindestens `/`) kann mit dem Programm `chroot` in das zu reparierende System gewechselt werden. Es steht dann eine Shell zur Verfügung, in der beliebige Befehle ausgeführt werden können.

Zugriff auf das zu reparierende System

```
# mount /dev/sda1 /mnt
# mount --bind /proc /mnt/proc
# mount --bind /sys /mnt/sys
# mount --bind /dev /mnt/dev
# chroot /mnt /bin/bash
# ... Reparatur, mit Strg-D System verlassen
# umount /mnt/proc
# umount /mnt/sys
# umount /mnt/dev
# umount /mnt
```

- `/proc`, `/sys`, `/dev` nicht immer benötigt
- zur Installation bestimmter Pakete oder Reparatur des Boot-Loaders

Die Folie zeigt die Befehle, die benötigt werden, um das Linux-System auf `/dev/sda1` einzubinden und mit `chroot` hineinzuwechseln.

Zusätzlich wurden hier die Dateisysteme `/proc`, `/sys` und `/dev` im Chroot verfügbar gemacht. Dies ist für manche Aufgaben notwendig. So benötigt man z. B. Zugriff auf `/dev` um den GRUB-Boot-Loader zu installieren.

Dem Befehl `chroot` wurde hier neben dem Mountpoint des Dateisystems auch die im System zu startende Shell (`/bin/bash`) als Parameter übergeben. Dies ist notwendig, wenn die im Live-System verwendete Shell im zu reparierenden System nicht verfügbar ist. GRML verwendet beispielswiese die `zsh`, welche standardmäßig unter Debian nicht installiert ist.

Nach erledigter Arbeit kann das Chroot mit Strg-D wieder verlassen werden. Danach sollten vor dem Neustarten die Dateisysteme wieder ausgehangen werden.

RAID, LVM und verschlüsselte Dateisysteme

- GRML enthält Werkzeuge zum Einbinden von RAID, LVM und verschlüsselten Dateisystemen
- RAID und LVM werden automatisch erkannt
- ggf. RAID schreibbar machen: `mdadm -w /dev/mdXXX`
- LVM aktivieren: `/etc/init.d/lvm2 start`
- verschlüsselte Dateisysteme mit `cryptsetup` aktivieren
- siehe auch Storage-Teil des Kurses
- vor dem Herunterfahren sicherheitshalber Dateisystem aushängen
- LVM deaktiveren: `vgchange -an <vg-name>`
- RAID deaktivieren: `mdadm --stop /dev/mdXXX`

Sollen Dateisysteme auf RAID-Verbünden oder LVM-Volumes eingebunden werden, müssen diese vorher aktiviert werden. GRML bringt die dazu notwendigen Werkzeuge bereits mit. RAID und LVM werden auch automatisch erkannt. Es sind vor der Benutzung daher nur noch wenige Schritte auszuführen.

Eventuell müssen erkannte RAID-Verbünde zuerst schreibbar gemacht werden. Dazu kann `mdadm` verwendet werden. Die Datei `/proc/mdstat` gibt Auskunft zum Status erkannter RAID-Verbünde.

LVM kann einfach durch Starten des entsprechenden Init-Skripts aktiviert werden.

Verschlüsselte Dateisysteme müssen manuell mit dem Befehl `cryptsetup` aktiviert werden. Dieser Befehl wurde im Storage-Teil dieses Kurses besprochen.

Nach dem Aktivieren können die Dateisysteme wie gewohnt eingebunden werden.

Vor dem Neustarten des Rechners sollten die Dateisysteme wieder ausgehangen werden und RAID, LVM und verschlüsselte Dateisysteme deaktivieren werden. Andernfalls kann es zu Datenverlust kommen.

sudo-Probleme

- defekte `/etc/sudoers` kann `sudo` unbrauchbar machen
- vorbeugen: `/etc/sudoers` nur mit `visudo` editieren
- führt rudimentäre Überprüfungen durch
- wenn kein Benutzer in `/etc/sudoers` eingetragen ist oder in der Gruppe sudo/admin ist, auch kein `sudo` möglich
- `/etc/sudoers` reparieren oder Benutzer in sudo/admin-Gruppe hinzufügen

Wird `sudo` verwendet um Root-Rechte zu erlangen, können Fehler in der Datei `/etc/sudoers` zu erheblichen Problemen führen. Um das Risiko dafür von vornherein zu verringen, sollte die Datei nur mit dem Programm `visudo` bearbeitet werden. Dadurch können Fehler an der Datei schon vor dem Speichern gefunden werden.

Sollte es trotzdem zu einem Problem gekommen sein, ist festzustellen, ob es an einer fehlerhaften `/etc/sudoers` oder eine fehlerhaften Gruppenmitgliedschaft liegt. Üblicherweise sind die Benutzer, die über `sudo` Root-Rechte erlangen dürfen, nicht explizit in `/etc/sudoers` aufgeführt, sondern Mitglied einer Gruppe, die dort aufgeführt ist. Diese Gruppe heist üblicherweise sudo oder admin. Die Gruppenmitgliedschaft kann mit dem Befehl `adduser` korrigiert werden.

SSH-Login nicht möglich

- verschiedene mögliche Ursachen
- Einstellungen in `/etc/ssh/sshd_config`
- Account gesperrt (mit `passwd -S` überprüfen)
- SSH-Key wird nicht gefunden
- fehlerhafte `~/.ssh/authorized_keys`
- Welt- oder Gruppen-Schreibrechte auf `/`, `/home`, `~`, `~/.ssh`
- Meldungen in `/var/log/auth.log`

Ist das Einloggen per SSH nicht möglich, kann dies verschiedene Ursachen haben.

In `/etc/ssh/sshd_config` können Einstellungen das Anmelden verbieten oder einschränken (`PermitRootLogin`, `PasswordAuthentication`, `AllowGroups`, `DenyGroups`, `AllowUsers`, `DenyUsers`).

Außerdem ist das passwort-basierte Anmelden für gesperrte Accounts nicht möglich. Mit `passwd -S` kann überprüft werden, ob der Account gesperrt ist (`L` in der Ausgabe). Mit `passwd -u` kann der Account entsperrt werden.

Bei SSH-Login per public Key kann es zu Problemen kommen, wenn die Datei `~/.ssh/authorized_keys` fehlerhaft ist. Auch Schreibrechte auf die Datei oder übergeordnete Verzeichnisse für andere Benutzer als den Besitzer sorgen dafür, dass die eingetragenen Keys aus Sicherheitsgründen ignoriert werden.

Generell lässt sich die Ursache der Probleme anhand der Meldungen in `/var/log/auth.log` nachvollziehen.

Fehlerhafte /etc/network/interfaces

- kein Netzwerk nach dem Booten oder nicht alle Interfaces hochgefahren
- Interfaces manuell mit `ifup` hochfahren
- auf Fehlermeldungen achten
- `/etc/network/interfaces` korrigieren

Probleme mit udev persistent net

- udev ordnet Netzwerkkarten anhand der MAC-Adresse einen festen Namen zu
- Zuordnung in `/etc/udev/rules.d/70-persistent-net.rules`
- nicht bei virtuellen Netzwerkkarten (in VMs)
- nach Netzwerkkartentausch ändert sich der Name des Netzwerkinterfaces
- Eintrag in `/etc/udev/rules.d/70-persistent-net.rules` löschen oder MAC-Adresse anpassen

Um Netzwerkkarten unter Linux feste Namen zuzuweisen, wird der Name der MAC-Adresse zugeordnet. Auf diese Weise erhält die Netzwerkkarte bei jedem Start den gleichen Namen. Dies führt allerdings zu Problemen, wenn die Netzwerkkarte beispielsweise aufgrund eines Defekts ausgetauscht wurde. Die Ersatz-Netzwerkkarte erhält dann einen anderen Namen, wodurch die Einträge in `/etc/network/interfaces` nicht mehr funktionieren.

Dieses Problem kann behoben werden, indem die entsprechende Zeile in `/etc/udev/rules.d/70-persistent-net.rules` gelöscht oder angepasst wird. Als einfachste Lösung kann die Datei auch vollständig gelöscht werden. Sie wird dann beim nächsten Neustart neu generiert.

GRUB-Probleme

- nach Festplattentausch (v. a. bei RAID 1) ist GRUB auf der neuen Festplatte nicht im Boot-Sektor installiert
- System bootet evtl. nicht mehr
- GRUB mit `grub-install /dev/XXX` in den Boot-Sektor der Festplatte installieren
- bei Hotswap aus dem laufenden System heraus
- bei Live-Systemen Dateisystem mounten und `/dev`, `/proc` und ggf. `/boot` einbinden
- dann mit `chroot` ins System wechseln und dort `grub-install` aufrufen

Wurde die Festplatte, auf der der Boot-Loader installiert war, ausgetauscht, führt dies beim nächsten Neustart dazu, dass das System nicht mehr startet. Um dies zu vermeiden, kann nach einem Austausch bei laufendem System (Hotswap) der Boot-Loader mit `grub-install /dev/XXX` auf die neue Festplatte installiert werden.

Ist das System bereits neu gestartet worden und fährt nicht mehr hoch, wird ein Live-System benötigt. Dort muss `/`, `/dev`, `/proc` und falls vorhanden auch `/boot` gemoutet werden und `grub-install` dann über `chroot` ausgeführt werden.

Lab 9.2: GRUB neu installieren

- GRML booten
- Dateisysteme einbinden
- GRUB neu installieren