

CHROOT-Umgebung: schroot

Nachfolgend wird die Installation einer minimalen Linux-Umgebung mittels **debootstrap**, sowie die Nutzung dieser mittels **schroot** (erweiterter CHROOT), beschrieben.

debootstrap

Der Befehl **debootstrap** (zu finden im gleichnamigen Paket) ermöglicht das Entpacken einer minimalen Linuxumgebung in ein vorgegebenes Verzeichnis. Das Programm kann aus dem Standard-Repository installiert werden.

Der Befehl zum Erstellen der Linuxumgebung sieht wie folgt aus:

```
~# debootstrap --arch <Architektur> <Version> <Zielverzeichnis>
<Quellrepository>
```

Am Beispiel:

```
~# debootstrap --arch amd64 trixie /mnt/debian
http://ftp.de.debian.org/debian
I: Retrieving Release
I: Retrieving Release.gpg
I: Checking Release signature
...
I: Configuring ifupdown...
I: Configuring libc-bin...
I: Base system installed successfully.
~#
```

schroot

Das Programm **schroot** kann ebenfalls aus dem Standard-Repository installiert werden und ist eine Erweiterung des Programms **chroot**.

Das Konfigurationsverzeichnis ist unter „/etc/schroot/“ zu finden und besitzt folgenden Aufbau:

- „buildd“, „default“, „desktop“, „minimal“ und „sbuild“: Hier liegen jeweils die Dateien, die an die CHROOT-Umgebung übergeben werden (Dateien und eingehängte Verzeichnisse)
 - die jeweiligen Verzeichnisse stehen für jeweilige Zwecke
 - „fstab“: Die Datei sollte um die Verzeichnisse erweitert werden, die in der CHROOT-Umgebung aus dem Host zur Verfügung stehen sollen
- „chroot.d“: Hier können die Konfigurationsdateien für die jeweiligen CHROOT-Umgebungen abgelegt werden
 - normalerweise stehen diese Informationen in der Datei „schroot.conf“
- „setup.d“: Skripte, die Aufbau einer CHROOT-Umgebung abgearbeitet werden

Konfigurationsbeispiel

Die Konfiguration einer CHROOT-Umgebung soll an einem Beispiel gezeigt werden. Es wird eine Umgebung für DEBIAN Stretch 64 Bit erstellt.

Im ersten Schritt wird ein bestehendes Konfigurationsprofil (zum Beispiel das Verzeichnis „default“) mit allen Dateien kopiert zum Verzeichnis „standard“:

```
~# cp -rv /etc/schroot/default /etc/schroot/standard
'/etc/schroot/default' -> '/etc/schroot/standard'
'/etc/schroot/default/copyfiles' -> '/etc/schroot/standard/copyfiles'
'/etc/schroot/default/fstab' -> '/etc/schroot/standard/fstab'
'/etc/schroot/default/nssdatabases' -> '/etc/schroot/standard/nssdatabases'
```

Jetzt kann die Datei „/etc/schroot/standard/fstab“ erweitert werden:

/proc	/proc	none	rw,bind	0	0
/sys	/sys	none	rw,bind	0	0
/dev	/dev	none	rw,bind	0	0
/dev/pts	/dev/pts	none	rw,bind	0	0
/home	/home	none	rw,bind	0	0
/tmp	/tmp	none	rw,bind	0	0
/data	/data	none	rw,bind	0	0

Hinweis: Das Verzeichnis „/data“ wurde hinzugefügt.

Es wird eine Konfigurationsdatei benötigt, in welcher alle Angaben zur CHROOT-Umgebung eingetragen werden. Nachfolgend wird dafür die Datei „/etc/schroot/chroot.d/debian-trixie-amd64-directory.conf“ verwendet:

```
[debian-trixie-amd64-directory]
description=DEBIAN Trixie AMD64 Directory
type=directory
directory=/data/system/schroot/directories/debian-trixie-amd64-directory
users=root,service
profile=standard
```

Zur Erklärung:

- „description“: Eine Beschreibung der CHROOT-Umgebung
- „type“: (nachfolgend nur die wichtigen Typen)
 - „directory“: Die CHROOT-Umgebung liegt in dem angegebenen Verzeichnis unter „directory“ (Veränderungen in der Umgebung bleiben erhalten)
- „directory“: Hier muss das Verzeichnis angegeben werden, in welches gewechselt werden soll
 - unter „type“ muss „directory“ stehen
- „users“: Liste mit Benutzern, die auf die CHROOT-Umgebung zugreifen dürfen
- „profile“: Angabe des Konfigurationsprofiles, welches verwendet werden soll

Beispielkonfiguration mit Angabe eines Archives in der Datei „/etc/schroot/chroot.d/debian-trixie-amd64-archive.conf“:

```
[debian-trixie-amd64-archive]
description=DEBIAN Trixie AMD64 Archive
type=file
file=/data/system/schroot/archives/debian-trixie-amd64-archive.tgz
users=root,service
profile=standard
```

Zur Erklärung (nur die Ergänzung):

- „type“: (nachfolgend nur die wichtigen Typen)
 - „file“: Die CHROOT-Umgebund wird aus dem angegebenen Archiv unter „file“ („tar.gz“, „tgz“ oder „tar.bz2“) entpackt und nach Beenden wieder entfernt
- „file“: Hier muss das Archiv angegeben werden, in welches entpackt und genutzt werden soll
 - unter „type“ muss „directory“ stehen
- „users“: Liste mit Benutzern, die auf die CHROOT-Umgebung zugreifen dürfen
- „profile“: Angabe des Konfigurationsprofiles, welches verwendet werden soll

Syntax

Anzeige aller vorhandenen CHROOT-Umgebungen:

```
~# schroot -l
chroot:debian-trixie-amd64-archive
chroot:debian-trixie-amd64-directory
source:debian-trixie-amd64-archive
```

Zur Erklärung:

- „chroot:...-archive“: Das Archiv wird entpackt und nach dem Verlassen wieder entfernt
- „source:...-archive“: Das Archiv wird entpackt und nach dem Verlassen in ein neues Archiv mit gleichem Namen gepackt

Soll das Archiv verwendet werden, hilft folgender Befehl:

```
~# schroot -c chroot:debian-trixie-amd64-directory
CHR00T@root@client~:~#
```

Hinweis: Die CHROOT-Umgebung ist hier an der Kennung „CHR00T@“ im Prompt zu erkennen.

Der Aufruf einer CHROOT-Umgebung die aus einem Archiv erstellt wird, läuft dabie nicht anders ab:

```
schroot -c chroot:debian-trixie-amd64-archive
CHR00T@root@client~:~#
```

Hinweis: Der Unterschied macht sich beim Abmelden bemerkbar, weil dieser Vorgang hier viel länger dauert.

Anpassungen

Damit mit der „nackten“ CHROOT-Umgebung ein besseres Arbeiten möglich ist, sollten noch ein paar

Anpassungen durchgeführt werden.

Paketquellen

Die Datei „/etc/apt/sources.list.d/debian.sources“ kann wie folgt angepasst (oder erstellt) werden:

```
Types: deb deb-src
URIs: http://ftp.gwdg.de/debian/
Suites: trixie
Components: main contrib non-free non-free-firmware
Signed-By: /usr/share/keyrings/debian-archive-keyring.gpg
```

```
Types: deb deb-src
URIs: http://security.debian.org/debian-security/
Suites: trixie-security
Components: main contrib non-free non-free-firmware
Signed-By: /usr/share/keyrings/debian-archive-keyring.gpg
```

```
Types: deb deb-src
URIs: http://ftp.gwdg.de/debian/
Suites: trixie-updates
Components: main contrib non-free non-free-firmware
Signed-By: /usr/share/keyrings/debian-archive-keyring.gpg
```

.bashrc

Die Datei „/root/.bashrc“ kann den nachfolgenden Inhalt haben:

```
alias ll='ls $LS_OPTIONS -al'
PS1='[schroot]:\[033[01;35m\]\u@\h\[033[00m\]:\[033[01;35m\]\w\[033[00m\]\$ '
```

.vimrc

Auch die Datei „/root/.vimrc“ sollte angepasst werden:

```
set mouse-=a
syntax enable
set tabstop=4 shiftwidth=4
```

Pakete

Innerhalb der CHROOT-Umgebung sollten dann noch einige Pakete installiert werden, welche das Arbeiten angenehmer machen:

```
~# apt update && apt -y install vim mc gpg wget
```

Wichtig : An dieser Stelle sollte von der CHROOT-Umgebung eine Sicherheitskopie erstellt werden.

X-Programme

Die CHROOT-Umgebung bietet die Möglichkeit an, X-Programme zu starten. Bisher ist mir das aber nur mit einem kleinen Trick gelungen. Dafür wurde ein zusätzliches Skript „/usr/local/bin/schrott“ erstellt, was folgenden Inhalt hat:

```
#!/bin/bash
if [ "${1}" = "-c" ]; then
    xhost + local:
    /usr/bin/schroot "${@}";
else
    /usr/bin/schroot "${@}";
fi;
```

Zusätzlich muss in der CHROOT-Umgebung in der Datei „/etc/profile“ folgende Zeile ergänzt werden:

```
export DISPLAY=:0
```

Wird das Wrapper-Skript „/usr/local/bin/schrott“ jetzt anstatt des Befehls „/usr/bin/schroot“ verwendet, können innerhalb der CHROOT-Umgebung X-Programme ausgeführt werden.

— *Steffen Bornemann* 07.11.2018

[DEBIAN](#), [debootstrap](#), [schroot](#), [CHROOT](#), [X](#)

From:
<https://looper.de/wiki/> - **Linux4Ever**



Permanent link:
<https://looper.de/wiki/doku.php?id=software:schroot>

Last update: **2025/12/11 15:00**