

Komplettsicherung des Systems

Zuverlässiger (z.B. offene Dateien) und möglicherweise einfacher (z.B. udev) ist es aus einem Rettungssystem heraus zu sichern. Die folgende Anleitung zeigt dagegen wie man aus einem laufenden System heraus sichert. Hürden wie eine nach /dev gemountete Ramdisk für udev werden dabei gemeistert. Dienste wie Mailserver oder Datenbanken (offene Dateien) sollten vorher angehalten werden.

Abschätzen, wie viel Platz benötigt wird:

```
df -Thlx tmpfs -x devtmpfs
```

→ Summe der Daten auf persistenten, lokalen Dateisystemen

Vorbereitung Zielsystem

evtl. Platz schaffen wie in [Plattenplatz](#), [Partitionierung](#) und [lvm](#) beschrieben

```
mkdir -p /mnt/backup/dateien
```

Ubuntu

Zielverzeichnis /mnt/backup muss dem Ubuntu-Nutzer gehören:

```
chown nutzerXX /mnt/backup
```

Debian (ab 8)

Entweder

- [sudo](#) einrichten und wie unter Ubuntu beschrieben vorgehen.
- einen [ssh](#)-Schlüssel beim Benutzer root hinterlegen

oder

- root das Anmelden mit Passwort erlauben ¹⁾

Sicherung

Diese Schritte werden auf dem zu sichernden System ausgeführt.

Sicherung der UEFI-Einstellungen

Bei UEFI-Systemen ist es manchmal hilfreich die EFI Variablen zu sichern:

```
efibootmgr -v > sicherung.efivars
```

Sicherung der Partitionierung

Mehr zur Vorgehensweise siehe [partitionierung](#). Wenn die Wiederherstellung nicht automatisiert erfolgen muss, reicht auch die Ausgabe von parted

```
fdisk -l -o device,size,type,uuid /dev/sda > sicherung.fdisk
```

oder

```
parted /dev/sda print > sicherung.parted
```

Ubuntu:

```
scp sicherung.parted nutzer@server:/mnt/backup
```

andere Distributionen (bei denen root ein Passwort hat):

```
scp sicherung.parted root@server:/mnt/backup
```

2)

Sicherung LVM Informationen

```
vgcfgbackup -f sicherung.vg  
pvdisplay > sicherung.pvdisplay  
vgdisplay > sicherung.vgdisplay  
lvdisplay > sicherung.lvdisplay
```

Sicherung anderer Storage Informationen

- [Software RAID](#)
- drbd
- [iSCSI](#)
- ...

Sicherung der Dateisystem-Einstellungen

Von welchen Dateisystemen müssen Einstellungen gesichert werden?

```
lsblk -f
```

oder

```
df -Thlx tmpfs -x devtmpfs
```

→ alle persistenten, lokalen Dateisysteme müssen gesichert werden

ext2 / ext3 / ext4

```
tune2fs -l /dev/sdaX > sicherung.ext.sdaX
```

3)

xfs

```
xfs_admin -l -u /dev/sdaX > sicherung.xfs.sdaX  
xfs_info /dev/sdaX >> sicherung.xfs.sdaX
```

4)

btrfs

btrfs lässt sich nicht einfach mit rsync oder tar sichern. Die beste Alternative ist, statt dessen ein [Image](#) des Dateisystems mit fsarchiver oder partclone zu erstellen. ⁵⁾

vfat

```
blkid /dev/sdaX > sicherung.vfat.sdaX
```

6)

swap

```
blkid /dev/sdaX > sicherung.swap.sdaX
```

7)

Einstellungs-Dateien auf Zielsystem kopieren

Ubuntu:

```
scp sicherung.* nutzer@server:/mnt/backup
```

andere Distributionen (bei denen root ein Passwort hat):

```
scp sicherung.* root@server:/mnt/backup
```

Sicherung der Dateien

Einhängen der Dateisysteme unter /mnt/system

In der Ausgabe von `df -hT -x tmpfs -x devtmpfs` die Mountpoints aller persistenten, lokalen Dateisysteme (mit Ausnahme von loop-Devices) identifizieren und unterhalb von `/mnt/system` bind-mounten:

```
mkdir /mnt/system
mount --bind / /mnt/system
mount --bind /boot /mnt/system/boot
...
```

8)

mit rsync über ssh Dateien kopieren

Auf Quell- und Zielsystem muss rsync installiert sein. Alternativ kann man auch [tar über ssh](#) verwenden.

Debian (ab 8)

Wie oben unter „[Vorbereitung Zielsystem](#)“ beschrieben kann man entweder so vorgehen

- wie bei anderen Distributionen, bei denen root ein Passwort hat

oder

- wie bei Ubuntu

Ubuntu

Unter **Ubuntu** gibt es defaultmäßig kein root Passwort. Wenn ein **Ubuntu** System das **ZIEL** ist, muss man zunächst dem User, mit dem man sich auf dem Ziel einloggen will, einen Eintrag machen, der rsync via sudo ohne Passwort erlaubt:

```
sudo visudo -f /etc/sudoers.d/backup
```

und diese Zeile ergänzen:

[/etc/sudoers.d/backup](#)

```
%sudo ALL=(ALL) NOPASSWD: ALL
```

9)

Der Benutzer muss in der Gruppe *sudo* sein

Und dann lautet der Befehl zum Backup:

```
rsync -aSH --xattrs --acls --numeric-ids --del --rsync-path="sudo rsync" /mnt/system/ user@zielsystem:/mnt/backup/dateien
```

SuSE / BTRFS

Btrfs snapshot subvolumes können nicht sinnvoll mit *rsync* oder *tar* gesichert und wieder hergestellt werden, da die Deduplizierung von Btrfs dabei nicht mehr genutzt wird und die Dateien mehrfach im Backup landen würden.

andere Distributionen (bei denen root ein Passwort hat)

```
rsync -aSH --acls --xattrs --numeric-ids --del /mnt/system/ root@server:/mnt/backup/dateien
```

10) 11) 12)

erweiterte Attribute und ACLs mit tar und rsync

RedHat's Versionen von [tar](#) und [rsync](#) können mit ACLs und erweiterten Dateiattributen umgehen, bei Debian ab Version 8. Dazu die Optionen `-xattrs (tar)` oder `-xattrs -acls (rsync)` angeben.

Plausibilitätsprüfung

Die wichtigsten Fragen sind:

- sind alle Dateien angekommen? → Dateien auf Original und Backup zählen
- sind die Berechtigungen korrekt? → Eigentümer/Berechtigungen auf Original und Backup vergleichen, dabei die [Unterschiede in den Benutzer IDs](#) auf Original- und Backupsystem beachten
- Sind die Dateien vollständig angekommen?

Sind alle Dateien angekommen?

- einfach mal mit `ls` stichprobenartig nachschauen
- mit Hilfe von `find` und `wc` Dateien zählen

Sind die Berechtigungen korrekt?

- einfach mal mit `ls -l` stichprobenartig nachschauen

Sind die Dateien vollständig angekommen?

Einfache Lösung:

- den selben `rsync`-Aufruf noch mal ergänzt durch die Schalter `-nv`
- Größenvergleich mit `du -sh` → Achtung: die Größen sind nie exakt gleich, die Größenordnung pro Dateisystem sollte passen.

Optional, für Freude der Unix-Kommandozeile:

Auf dem Zielsystem Checksummen für Backup-Dateien erzeugen:

```
cd /mnt/backup/dateien
find . -xdev -type f -print0 | sort -z | xargs -0 cksum > /run/cksum.backup
find . -xdev -print0 | sort -z | xargs -0 getfacl -P -n >
/run/getfacl.backup
```

Auf dem Original-System Checksummen für Dateien erzeugen: (hier müssen neben `.` und `./boot` möglicherweise weitere Dateisysteme angegeben werden)

```
cd /mnt/system
find . ./boot -xdev -type f -print0 | sort -z | xargs -0 cksum >
/run/cksum.orig
find . ./boot -xdev -print0 | sort -z | xargs -0 getfacl -P -n >
/run/getfacl.orig
```

`/run/*.orig` auf das Zielsystem kopieren

```
scp /run/*.orig root@server:
```

Checksummen vergleichen

```
diff ~/cksum.orig /run/cksum.backup
```

Berechtigungen vergleichen

```
diff ~/getfacl.orig /run/getfacl.backup
```

Sicherung der ACL-Dateirechte

nur nötig, wenn tar oder rsync das nicht kann

```
cd /mnt/system
getfacl --skip-base -P -n -R . > sicherung.acl
scp sicherung.acl root@server:/mnt/backup
```

Sicherung der erweiterten Dateiattribute (z.B. SELINUX, capabilities)

nur nötig, wenn tar oder rsync das nicht kann

```
cd /mnt/system
getfattr -Rh -m . -d . > sicherung.attr
scp sicherung.attr root@server:/mnt/backup
```

Löschen des Systems

13)

schnell

```
wipefs -af /dev/sda
```

Oder bei SSDs:

```
blkdiscard --force /dev/sda
```

14)

sicher

```
dd if=/dev/zero of=/dev/sda
```

... und nach ein paar Sekunden ist das System zumindest unbrauchbar. Oder bei SSDs:

```
blkdiscard -s /dev/sda
```

paranoid

```
dd if=/dev/urandom of=/dev/sda
```

... und laaaange warten

Gefahr der Beschädigung der Hardware

Bis Kernel 4.5 ist es möglich, dass durch das Löschen der Dateien unter `/sys/firmware/efi/efivars` die UEFI-Firmware beschädigt wird und der Rechner nicht mehr booten kann.

Daher vorsichtig sein mit

```
rm -rf /...
```

Mehr dazu bei [heise](#) und im entsprechenden [Kernel-Patch](#)

Wiederherstellung des Systems

Rettungssystem (z.B. sysrcd, knoppix) booten

Schritte im Rettungs-System

Wiederherstellung der Partitionierung

Partitionierung anhand der Informationen aus der gesicherten Datei `sicherung.parted` und/oder gemäß `etc/fstab` aus dem Backup wie in [Partitionierung](#) beschrieben anlegen

Bitte Typ der Partitionstabelle (gpt oder msdos) beachten.

Wiederherstellung LVM

In Knoppix muss dazu zuerst `lvm2` gestartet werden:

```
service lvm2 start
```

mit `vgcfgrestore`

IDs der Physical Volumes herausfinden:

```
less ~/sicherung.vg
```

Physical Volumes anlegen:

```
pvcreate -ff --zero y --restorefile ~/sicherung.vg --uuid xxxxxx-xxxx-xxxx-xxxx-xxxx-xxxx-xxxx-xxxx /dev/sdaX
```



```
...
```

Name der Volume Group herausfinden:

```
lss ~/sicherung.vg
```

Volume Group und Logical Volumes wiederherstellen:

```
vgcfgrestore -f ~/sicherung.vg centos_notebook17
```

von Hand

LVM anhand der Informationen aus der gesicherten Datei `sicherung.pvdisplay`, `sicherung.vgdisplay`, `sicherung.lvdiskdisplay` und/oder gemäß `etc/fstab` aus dem Backup wie in [lvm](#) beschrieben anlegen.

Dateisysteme anlegen

Dateisysteme mit `mkfs.*` gemäß `etc/fstab` aus dem Backup anlegen. Dabei wenn nötig die Dateisystem-Einstellungen aus der Sicherung berücksichtigen (UUID, ...) ¹⁵⁾

ext4

```
mkfs.ext4 -U xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx /dev/sdaX
```

16)

xf

```
mkfs.xfs -m uuid=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx /dev/sdaX
```

17)

fat

```
mkfs.vfat -i xxxxxxxx /dev/sdaX
```

18)

Swap anlegen

19)

```
mkswap -U xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx /dev/sdaX
```

Mounten der Zielpartitionen

In `/mnt/backup/dateien/etc/fstab` aufgeführte Mountpoints mit `mkdir -p` anlegen und Dateisysteme mit `mount` einhängen: ^{20) 21)}

```
mkdir -p /tmp/system
mount /dev/sdaX /tmp/system
```

```
mkdir -p /tmp/system/boot
mount /dev/sdaY /tmp/system/boot
```

...

Wiederherstellen der Dateien mit rsync über ssh

²²⁾

```
rsync -aSH --acls --xattrs --numeric-ids --del -e ssh
root@server:/mnt/backup/dateien/ /tmp/system
```

evtl. ACLs und erweiterte Dateisystemattribute berücksichtigen ²³⁾

Ubuntu

```
rsync -aSH --acls --xattrs --numeric-ids --del -e ssh --rsync-path="sudo
rsync" user@server:/mnt/backup/dateien/ /tmp/system
```

Wiederherstellen der ACL-Dateirechte

nur nötig, wenn `rsync` bzw. `tar` das nicht kann

```
mount -o remount,acl /tmp/system/
cd /tmp/system
setfacl --restore sicherung.acl
```

Wiederherstellen der SELinux Attribute

nur nötig, wenn `rsync` bzw. `tar` das nicht kann (nicht erfolgreich getestet ...)

```
mount -o remount,user_xattrs /tmp/system
cd /tmp/system
```

```
setfattr -h --restore sicherung.attr
```

oder

```
touch /tmp/system/.autorelabel
```

Bootfähig machen

chroot vorbereiten

```
mount --rbind /dev /tmp/system/dev  
mount --rbind /proc /tmp/system/proc  
mount --rbind /sys /tmp/system/sys
```

24)

25)

/run ist sinnvoll bei der Verwendung von lvm oder raid

```
mount --bind /run /tmp/system/run
```

UEFI

(nur bei UEFI-Systemen)

efivars schreibbar machen

```
mount -o rw,remount /tmp/system/sys/firmware/efi/efivars
```

oder (falls /tmp/system/sys/firmware/efi/efivars kein mountpoint ist)

```
mount -t efivarfs efivarfs /tmp/system/sys/firmware/efi/efivars
```

Schritte im chroot Zielsystem

```
chroot /tmp/system /bin/bash
```

grub-efi über efi

Bei einfachen Installationen ist grub-efi der zweite Bootloader hinter efi. In diesem Fall reicht

```
grub-install dummy
```

Voraussetzung dafür ist, dass die EFI-System-Partition (ESP) an die richtige Stelle gemountet ist (/boot/grub/efi)

26)

EFI Variablen prüfen

```
efibootmgr -v
```

- Partition unique GUID: muss zur ESP Partition passen
- The path of the EFI image to boot must use \ (backslash) instead of / (forward slash) as path separator.

BIOS: Bootloader in mbr schreiben

(nur bei BIOS-Systemen)

Schritte im chroot Zielsystem

```
chroot /tmp/system /bin/bash
```

grub2 wiederherstellen

Debian (ab 6.0), Ubuntu (ab 16.04)

```
grub-install /dev/sda  
update-grub2
```

openSUSE 12.2

```
grub2-install /dev/sda  
update-bootloader --refresh
```

CentOS 7

```
grub2-install /dev/sda  
grub2-mkconfig > /boot/grub2/grub.cfg
```

andere

```
grub2-install /dev/sda
```

```
grub-mkconfig > /boot/grub/grub.cfg
```

grub1 / grub legacy wiederherstellen

Grub in den MBR schreiben:

```
grub
> root (hd0,0)
> setup (hd0)
> quit
```

oder mit

```
grub-install --recheck --no-floppy hd0
```

Centos 8 Recovery mit UEFI

grub-install funktioniert nicht, also müssen die EFI-Einträge manuell gesetzt werden

```
efibootmbr -v
```

da wird wahrscheinlich nur noch der PXE-Eintrag sein

EFI SP merken

```
blkid
gdisk -l /dev/sda
```

Achtung: Je nach Umgebung, sind efivars gar nicht gemountet oder RO

```
mount -o remount,rw /sys/firmware/efi/efivars
```

EFI-Eintrag machen (-p = ordinale Nummer der ESP)

```
efibootmgr -c -L "CentOS Linux" -l /EFI/centos/shimx64.efi -p 1
```

nochmal kontrollieren

```
blkid
efibootmgr -v
```

grub-config neu erzeugen

```
grub2-mkconfig -o /boot/efi/EFI/centos/grub.cfg
```

Fallen

- /etc/default/grub # resume devices und Parameter

reboot

chroot-Umgebung (z.B. mit St rg+d) verlassen und

```
reboot
```

Anpassungen bei geänderter Hardware oder geänderten Partitionen

Je nach dem wie sich die Hardware geändert hat, sind folgende Bereiche anzupassen:

Format Partitionstabelle

Wenn die neue Festplatte größer als 2TB ist, dann GPT als Format für die Partitionstabelle verwenden.

Wenn Linux von der Festplatte auch booten soll, dann braucht Grub vor der ersten Partition etwas Platz. Die erste Partition sollte daher - wie heute üblich - bei Block 2048 (= 1MiB \approx 1,05MB) starten. Alternativ kann man diesen unpartitionierten Bereich mit einer [BIOS Boot Partition](#) z.B. von Block 34 bis Block 2047 vor Veränderungen schützen.

[27\)](#) [28\)](#)

/boot auf separater Partition

Bei Debian mit UEFI anpassen: `/boot/efi/EFI/debian/grub.cfg`

Bootloader grub

`/boot/grub/menu.lst` :

Neue Partitionsnummern beachten, wichtig sind z.B.

- die Einstellungen zur Boot-Partition (`root`) in Grub-Terminologie
- die Lage des Kernels (`/boot/vmlinuz*` oder `/vmlinuz*`)
- Kernel-Parameter zum root-Dateisystem (`root= ...`)

Bootloader grub2

`/boot/grub/grub.cfg` : (sollte automatisch über `update-grub2` erstellt werden)

Neue Partitionsnummern beachten, wichtig sind z.B.

- die Einstellungen zur Boot-Partition (`root`) in Grub-Terminologie
- die Lage des Kernels (`/boot/vmlinuz*` oder `/vmlinuz*`)

- Kernel-Parameter zum root-Dateisystem (root= ...)

BIOS -> UEFI

Um ein (ehemaliges) BIOS System auf einer UEFI Hardware wieder herzustellen braucht man

- Partitionstabelle mit gpt (siehe oben)
- eine FAT32 Partition für /boot/efi, für den Linux Bootloader Grub allein reicht eine Größe von wenigen MByte. Wenn auch der Linux Kernel und/oder Windows auch drauf soll deutlich größer (500MB?).
- GRUB2:
 - Debian: die Pakete grub-efi, grub-efi-amd64-bin und efibootmgr installieren. Dann grub-install + update-grub ausführen.
 - Ubuntu: wie bei Debian, zusätzlich optional noch shim bzw. shim-signed
 - CentOS: Pakete grub2-efi, grub2-efi-x64-modules und efibootmgr installieren. Dann grub-install + update-grub
 - SuSE: grub2-x86_64-efi und efibootmgr installieren. Dann grub-install + update-bootloader ausführen.

/etc/fstab

/etc/fstab : Neue Partitionsnummern, Software-RAID und Logical Volumes beachten

/etc/mtab

/etc/mtab : Neue Partitionsnummern, Software-RAID und Logical Volumes beachten ²⁹⁾

Kernel-Module und initrd

Je nach Änderung muß eine neue [initrd](#) erzeugt werden und/oder die bei Booten geladenen Module müssen überarbeitet werden

dracut (CentOS 8)

in der chroot-Umgebung

```
dracut --hostonly --persistent-policy 'by-uuid' --force  
/boot/initramfs-4.18.0-80.11.2.el8_0.x86_64.img 4.18.0-80.11.2.el8_0.x86_64
```

dracut (CentOS 7 / openSuSE 42.1)

in der chroot-Umgebung

[/etc/dracut.conf.d/10-xfs.conf](#)

```
add_drivers+=" xfs "
```

```
dracut --force /boot/initrd-4.4.87-18.29-default 4.4.87-18.29-default
```

CentOS 6

in der chroot-Umgebung

```
mkinitrd --force /boot/initramfs-2.6.32-220.el6.x86_64.img  
2.6.32-220.el6.x86_64
```

udev

Gerätenamen in `/etc/udev/rules.d` anpassen

Alternative Sicherungswege

tar über ssh

Sicherung

```
tar cz --numeric-owner --sparse --xattrs --acls --directory /mnt/system . |  
ssh nutzer@server "cat > /mnt/backup/sicherung.tgz"
```

Wiederherstellen

```
ssh nutzer@server 'cat /mnt/backup/sicherung.tgz' | tar xz --numeric-owner -  
-sparse --xattrs --acls --xattrs-include='*' --directory /tmp/system
```

Dokumentation

- [Übersicht tar](#)
- `man tar`

cpio über ssh

Sicherung

30)

```
cd /mnt/system
find -xdev -depth -print0 | cpio -o0 --sparse --format=crc | bzip2 | ssh
nutzer@server 'cat > /mnt/backup/sicherung.cpio.bz2'
```

Wiederherstellen

```
cd /tmp/system
ssh nutzer@server 'cat /mnt/backup/sicherung.cpio.bz2' | bunzip2 | cpio -
dumin --sparse
```

rsync

Auf lokale Platte:

```
rsync -a --hard-links --sparse --acls --xattrs --numeric-ids --del /
/mnt/usbdisk/root/
```

Übers Netz via ssh:

```
rsync -a --hard-links --sparse --acls --xattrs --numeric-ids --del /
server:/mnt/backup/dateien
```

Übers Netz via rsyncd: ³¹⁾

```
rsync -a --hard-links --sparse --acls --xattrs --numeric-ids --del /
server::/backup/dateien/
```

Platzsparende Hardlink Backups mit rsnapshot

Installation mit

```
apt-get install rsnapshot
```

Konfiguration unter:

```
/etc/rsnapshot.conf
```

Dabei wichtig:

```
snapshot_root          - wohin soll gesichert werden
interval daily 7       - Wieviele Versionen vom täglichen
Backup sollen behalten werden?
```

```
interval weekly 4           -   Wieviele Versionen vom
wöchentlichen Backup sollen behalten werden?
interval monthly 6         -   Wieviele Versionen vom monatlichen
Backup sollen behalten werden?
backup /quelle zielrechner/ -   Zb localhost/ oder im Zusammenhang
mit SSH gebrauchen
```

Testen der Config

```
rsnapshot configtest
```

Danach kann man das Ganze erst einmal im Probedurchlauf testen ohne wirklich Dateien zu schreiben:

```
rsnapshot -t daily
```

Problem: udev

Seit Kernel 2.6 und der Einführung von [udev](#) haben Linux Distributionen das Verzeichnis `/dev` von einem temporären Dateisystem überdeckt. Linux benötigt diese überdeckten Dateien aber beim Booten. Neuere Distributionen (RHEL / CentOS 6, ...) tun das nicht mehr.

Lösungswege:

1. Beim Backup das Verzeichnis `/dev` explizit angeben. Die wichtigsten Gerätedateien sollten auch darin enthalten sein.
2. Auf das Originalsystem kann man mit `mkdir /tmp/system && mount -bind /tmp/system zugreifen`. Die Sicherung muss man dann mit `tar czpf /dev/st0 -numeric-owner -directory /tmp/system .` starten.

FRAGE: das müsste doch durch die bind-mounts nach `/mnt/system` gelöst sein, oder?!

Antwort: richtig

Dokumentation

- <http://www.linux-magazin.de/Artikel/ausgabe/2002/04/rsync/rsync.html>

Software

- <http://www.dirvish.org>
- <http://www.mondorescue.org/>
- http://www.partimage.org/Main_Page

1)

[/etc/ssh/sshd_config](#)

```
PermitRootLogin yes
```

2)

alternativ für MSDOS-Partitionstabellen:

```
sfdisk -d /dev/sda > sicherung.sfdisk
scp sicherung.sfdisk root@server:/mnt/backup
```

bei GPT-Partitionstabellen eine inzwischen veraltete Warnung der Entwickler vorweg: *As of March 2014 (version 0.8.10), sgdisk should be considered beta software.*

3) 4) 6) 7) 15) 19)

/dev/sdaX ist hier nur ein Beispiel für ein Speichergerät (Partition, LVM, ...). Liegt das Dateisystem auf einem Logical Volume, dann heißt das Device /dev/mapper/xxx oder ähnlich

5)

Vielleicht ist auch <https://github.com/mwilck/btrfs-clone> eine brauchbare Alternative

8)

ab RedHat 6, openSuSE 13.1, Debian 8 und Ubuntu 16.04 könnte man auf diesen Schritt verzichten, und statt dessen bei rsync die Option -x bzw. --one-file-system nutzen und die entsprechenden Mountpoints einzeln angeben

9)

oder

[/etc/sudoers.d/backup](#)

```
%sudo ALL=(ALL) NOPASSWD: /usr/bin/rsync
```

10)

mehr zu [rsync](#), u.a. wie man hier rsync auch ohne root-Rechte benutzen kann

11)

mehr zu [ssh](#)

12)

wenn man ein Art Fortschrittsbalken haben will: progress installieren und progress -wm ausführen während rsync läuft.

13)

Und wie geht es weiter, wenn die Festplatte gelöscht ist? Wie kann ich dann noch rebooten?

```
echo 1 > /proc/sys/kernel/sysrq
echo b > /proc/sysrq-trigger
```

Siehe [sysctl](#)

14)

Das System lebt jetzt mit den Daten aus dem Cache etwas weiter. Cache löschen:

```
echo 1 > /proc/sys/vm/drop_caches
```

16) 17)

xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx steht hier für die UUID aus der Sicherung

18)

Eine UUID die von blkid im Format UUID="066B-5CE0" ausgegeben wurde muss mkfs.vfat mit 066B5CE0 (also ohne Minus) übergeben werden

20)

evtl. mount-optionen (z.B. acl) beachten

21)

/dev/sdaX ist hier nur ein Beispiel für ein Speichergerät (Partition, LVM, ...). Liegt das Dateisystem auf einem Logical Volume, dann heißt das root-Device /dev/mapper/xxx oder ähnlich

22)

mehr siehe [rsync](#)

23)

Notlösung: Berechtigungen (teilweise) wiederherstellen, wenn sie nicht richtig gesichert wurden

```
rpm -qa | xargs rpm --setperms
```

Todo: debian?

24)

nicht alle diese Befehle sind in allen Fällen notwendig! grub braucht hauptsächlich /dev, und das ist, wenn man zuvor beim Sichern mit mount `--bind` gearbeitet hat, schon minimal befüllt. Kann man also weglassen, aber wenn grub meckert, sollte man das versuchen! Ich habe es schon erlebt, dass die Datei /etc/mtab falsche Informationen enthält. Falls das so ist, kann man sie so ersetzen:

```
mv /tmp/system/etc/mtab /tmp/system/etc/mtab.bak
cp -a /proc/mounts /tmp/system/etc/mtab
```

Nach dem chroot nicht vergessen, die /etc/mtab wiederherzustellen:

```
mv /tmp/system/etc/mtab.bak /tmp/system/etc/mtab
```

25)

=== Alternativ ohne rbind ===

```
mount --bind /dev /tmp/system/dev
mount --bind /dev/pts /tmp/system/dev/pts
mount --bind /proc /tmp/system/proc
mount --bind /sys /tmp/system/sys
```

26)

Nachfolgend die Befehle, die notwendig sind, ein reines EFI-System wiederherzustellen oder wenn alle EFI-Informationen verloren sind === EFI Variablen schreiben === Die folgenden Shell-Variablen gemäß Ausgabe von

```
ls /boot/efi/EFI/
```

und der Datei `sicherung.efivars` setzen:

```
efi_label='Name der Distribution'
distro='DISTRIBUTION'
boot_device='/dev/sda'
esp_partition_id=2
efibootmgr --create --disk "$boot_device" --part "$esp_partition_id" --label
'UEFI OS' --loader '\EFI\BOOT\BOOTX64.EFI'
efibootmgr --create --disk "$boot_device" --part "$esp_partition_id" --label
"$efi_label" --loader "\\EFI\\$distro\\SHIMX64.EFI"
```

=== Grub2 konfigurieren === In CentOS 8 BLSCFG abschalten:

[/etc/default/grub](#)

```
...  
GRUB_ENABLE_BLSCFG=false  
...
```

TODO: Lösung mit BLSCFG finden == CentOS ==

```
grub2-mkconfig -o /boot/efi/EFI/$distro/grub.cfg
```

== Debian, Ubuntu ==

```
dpkg-reconfigure grub-efi-amd64
```

oder

```
update-grub2
```

²⁷⁾

aus dem [archlinux-Wiki](#): *Keep in mind that if your Boot-Manager is GRUB, it needs a BIOS Boot Partition. If your MBR Partitioning Layout isn't too old, there is a good chance that the first partition starts at sector 2048 for alignment reasons. That means at the beginning will be 1007 KiB of empty space where this bios-boot partition can be created. To do this, first do the mbr→gpt conversion with gdisk as described above. Afterwards, create a new partition with gdisk and manually specify its position to be sectors 34 - 2047, and set the EF02 partition type.*

²⁸⁾

aus der [GRUB Doku](#): *When creating a BIOS Boot Partition on a GPT system, you should make sure that it is at least 31 KiB in size. (GPT-formatted disks are not usually particularly small, so we recommend that you make it larger than the bare minimum, such as 1 MiB, to allow plenty of room for growth.) You must also make sure that it has the proper partition type. Using GNU Parted, you can set this using a command such as the following:*

```
parted /dev/disk set partition-number bios_grub on
```

If you are using gdisk, set the partition type to '0xEF02'. With partitioning programs that require setting the GUID directly, it should be '21686148-6449-6e6f-744e656564454649'.

²⁹⁾

möglicherweise kann man `/etc/mtab` auch einfach löschen

³⁰⁾

Vermeidet die Probleme von tar und gzip bei Datenfehlern

³¹⁾

erfordert laufenden rsyncd auf dem Zielsystem server

From:
<https://wiki.lab.linuxhotel.de/> - **Linuxhotel Wiki**

Permanent link:
<https://wiki.lab.linuxhotel.de/doku.php/lpi1:systemsicherung>

Last update: **2021/12/25 16:56**

