

# Debian

## Debian 32bit Sarge Firecracker Raid1 Howto+grsecurity Kernel 2.6.11.12

Diese Anleitung basiert auf der Anleitung von Fastwebpace24 aus dem Webperoni Kundenforum.

<https://www.webperoni.de/forum/viewtopic.php?p=700#700>

Partitionen formatieren:

```
[code]
mke2fs -j /dev/md1
mke2fs -j /dev/md0
mkswap /dev/hda3
mkswap /dev/hdc3
swapon /dev/hda3
swapon /dev/hdc3
[/code]
```

Verzeichnisse erstellen und Partitionen mounten:

```
[code]
mkdir -p /mnt/rd/mnt
mkdir -p /mnt/rd/mnt/debian
mount /dev/md1 /mnt/rd/mnt/debian
mkdir -p /mnt/rd/mnt/debian/daten
[/code]
```

Basis Paket herunterladen:

```
[code]
cd /mnt/rd/mnt/debian/daten
Dies ist eine Zeile->wget http://http.us.debian.org/debian/dists/woody/main/
disks-i386/base-images-current/basedebs.tar<-Zeile Ende
Dies ist eine Zeile->/usr/sbin/debootstrap --arch i386 --unpack-tarball
/mnt/rd/mnt/debian/daten/basedebs.tar woody /mnt/rd/mnt/debian<-Zeile Ende
```

```
[/code]
```

Bootpartition mounten und ab ins chroot:

```
[code]
chroot /mnt/rd/mnt/debian/ /bin/bash
mkdir -p /mnt/rd/mnt/debian/boot
mount /dev/md0 /mnt/rd/mnt/debian/boot
[/code]
```

Wer sich mit dem vi nicht auskennt, der drückt Einfg, fügt den gewünschten text ein, einmal Esc, danach :wq

Ich nehme da immer den MidnightComander (mc) apt-get install mc. Der Befehl zum öffnen lautet dann mcedit /etc/fstab

/etc/fstab anlegen:

```
[code]
vi /etc/fstab
/dev/md0      /boot        ext3         user         0 0
/dev/md1      /             ext3         usrquota,grpquota 1 1
/dev/hda3     swap         swap         pri=42       0 0
```

Seite 1 / 5

# Debian

```
/dev/hdc3      swap          swap      pri=42          0 0
devpts         /dev/pts        devpts     mode=0620,gid=5  0 0
proc           /proc           proc        defaults         0 0
usbfs          /proc/bus/usb   usbfs       noauto           0 0
sysfs          /sys            sysfs       noauto           0 0
```

[/code]

proc mounten:

[code]

```
mount -t proc proc /proc
```

[/code]

Tastatur einstellen:

[code]

```
dpkg-reconfigure console-data
```

[/code]

/etc/modules anlegen:

[code]

```
vi /etc/modules
```

```
via-rhine
```

[/code]

Netzwerk einrichten:

[code]

```
vi /etc/network/interfaces
```

```
auto lo
```

```
iface lo inet loopback
```

```
auto eth0
```

```
iface eth0 inet static
```

```
address 83.133.xx.xxx
```

```
network 83.133.xx.0
```

```
netmask 255.255.255.0
```

```
broadcast 83.133.xx.255
```

```
gateway 83.133.xx.1
```

[/code]

resolv.conf anlegen:

[code]

```
vi /etc/resolv.conf
```

```
nameserver 83.133.80.9
```

```
nameserver 83.133.80.10
```

[/code]

Hostnamen anlegen:

[code]

```
echo meinhostname > /etc/hostname
```

[/code]

hosts Datei anlegen:

[code]

```
vi /etc/hosts
```

```
127.0.0.1 localhost
```

```
83.133.xx.xxx meinhostname.webperoni.de
```

```
83.133.xx.xxx meinhostname
```

# Debian

[/code]

Debian Konfigurieren:

[code]

/usr/sbin/base-config

[/code]

Noch ein paar Programme installieren:

[code]

```
apt-get install less wget w3m vim libncurses5-dev make gcc mbr bzip2 lilo mdadm  
mc ssh
```

[/code]

Sources.list ergänzen:

[code]

/etc/apt/sources.list

deb <http://ftp.debian.de/debian> sarge main contrib

deb-src <http://ftp.debian.de/debian> sarge main contrib

deb <http://ftp.debian.de/debian-non-US> sarge/non-US main contrib

deb-src <http://ftp.debian.de/debian-non-US> sarge/non-US main contrib

deb <http://ftp.debian.de/debian/> sarge-proposed-updates main

deb-src <http://ftp.debian.de/debian/> sarge-proposed-updates main

deb <ftp://ftp.de.debian.org/debian/> testing main

deb-src <ftp://ftp.de.debian.org/debian/> testing main

deb <http://security.debian.org/> testing/updates main contrib

deb <ftp://ftp2.de.debian.org/debian/> testing main

deb-src <ftp://ftp2.de.debian.org/debian/> testing main

deb <ftp://debian.tu-bs.de/debian/> testing main

deb-src <ftp://debian.tu-bs.de/debian/> testing main

deb <ftp://ftp.uni-erlangen.de/pub/Linux/debian/> testing main

deb-src <ftp://ftp.uni-erlangen.de/pub/Linux/debian/> testing main

deb <ftp://ftp.uni-stuttgart.de/debian/> testing main

deb-src <ftp://ftp.uni-stuttgart.de/debian/> testing main

deb <ftp://sunsite.informatik.rwth-aachen.de/pub/Linux/debian/> testing main

deb-src <ftp://sunsite.informatik.rwth-aachen.de/pub/Linux/debian/> testing main

[/code]

Updates durchführen:

[code]

```
apt-get update
```

```
apt-get upgrade
```

```
apt-get clean
```

```
apt-get dist-upgrade
```

[/code]

Lilo Bootloader anpassen:

[code]

```
vi /etc/lilo.conf
```

```
boot=/dev/md0
```

```
root=/dev/md1
```

```
install=/boot/boot-menu.b
```

```
delay=3
```

```
lba32
```

```
image=/vmlinuz
```

# Debian

initrd=/initrd.img

label=Linux

[/code]

Standard Kernel installieren:

[code]

apt-get install kernel-image-2.4-386

[/code]

Bootsector schreiben:

[code]

lilo -b /dev/hda -D Linux

zum Debuggen:

lilo -v -v -b /dev/hda -D Linux

[/code]

Recovery Modus im Webperoni Panel abschalten und nicht! neu starten lassen.  
Mindestens 2 min warten!

chroot beenden:

[code]

exit

umount /mnt/rd/mnt/debian/boot

umount /mnt/rd/mnt/debian/

shutdown -r now

[/code]

Das wars! Wenn alles geklappt hat bootet der Rechner jetzt :-)

Wer jetzt noch Lust hat, der baut sich noch einen eigenen aktuellen Kernel.

Kernel backen incl. grsecurity Kernel 2.6.11.12 patch:

Infos zum grsecurity Patch findete Ihr hier: <http://www.grsecurity.net>

Zuerst den alten Kernel entfernen:

[code]

apt-get remove kernel-image-2.4-386

[/code]

Aus der lilo.conf entfernen:

#initrd=/initrd.img

Kernel runterladen:

[code]

cd /usr/src

wget <http://ftp.nl.kernel.org/pub/linux/kernel/v2.6/linux-2.6.11.12.tar.bz2>

tar -jvxf linux-2.6.11.12.tar.bz2

ln -s linux-2.6.11.12 linux

# grsecurity (optional)

wget <http://www.grsecurity.net/grsecurity-2.1.6-2.6.11.12-200506141713.patch.gz>

# Debian

```
cd linux
```

```
zcat grsecurity-2.1.6-2.6.11.12-200506141713.patch.gz | patch -p1
```

```
cd /usr/src/linux
```

```
[/code]
```

Nun muss der Kernel compiliert werden.

Wichtig dabei ist, das die Module via-rhine, raid1 Support und der grsecurity Kernel aktiviert werden.

Nicht als Module, sondern direkt in den Kernel laden!

```
[code]
```

```
make menuconfig
```

```
make bzImage
```

```
cp arch/i386/boot/bzImage /boot/vmlinuz-2.6.11.12
```

```
rm /boot/map
```

```
rm /boot/System.map
```

```
cp System.map /boot/System.map-2.6.11.12
```

```
ln -s /boot/System.map-2.6.11.12 /boot/System.map
```

```
rm /boot/.config
```

```
cp .config /boot/.config-2.6.11.12
```

```
ln -s /boot/.config-2.6.11.12 /boot/.config
```

```
make modules
```

```
make modules_install
```

```
rm /vmlinuz
```

```
ln -s /boot/vmlinuz-2.6.11.12 /vmlinuz
```

```
lilo -v -v -b /dev/hda -D Linux
```

```
[/code]
```

Recovery Modus im Webperoni Panel abschalten und nicht! neu starten lassen.

Mindestens 2 min warten!

chroot beenden:

```
[code]
```

```
exit
```

```
umount /mnt/rd/mnt/debian/boot
```

```
umount /mnt/rd/mnt/debian/
```

```
shutdown -r now
```

```
[/code]
```

Kernel überprüfen:

```
cat /proc/version
```

```
Linux version 2.6.11.12-grsec (root@xxxxxxx) (gcc version 3.3.5 (Debian 1:3.3.5-13)) #1 SMP Fri Jul 15 00:13:58 CEST 2005
```

Eindeutige ID: #1023

Verfasser: Michael Stender

Letzte Änderung: 2007-08-08 03:11