Switch Debian from legacy to UEFI boot mode

11.3.2017

Table of Contents

This note explains how to switch a legacy boot Debian/Ubuntu system into a UEFI boot system. Typical use case:

- switch a legacy boot installation into an UEFI one,
- reinstall a broken UEFI boot loader on Debian 7, Debian 8 or Debian 9.

> This manual has been tested on Debian 7 Wheezy, Debian 8 Jessie and Debian 9 Stretch

1. Boot a live system

1. Enable UEFI in bios.

2. Boot an recent Debian live[1] system on USB or DVD.

2. Prepare the harddisk

2.1. Back up your data

Back up your data!

2.2. Identify Debian's "/boot" partition

My lecagy boot system had a 243MiB ext2 partition mounted on /boot. This partition is never encrypted. It is where the grub files and Linux kernels reside. Check by double clicking on the partition icon on the live-disk-desktop and have a look inside.

```
# ls -l
total 21399
-rw-r--r-- 1 root root   155429 Sep 28 00:59 config-3.16-0.bpo.2-amd64
drwxr-xr-x 3 root root     7168 Nov  5 08:03 grub
-rw-r--r-- 1 root root 15946275 Nov  5 16:28 initrd.img-3.16-0.bpo.2-amd64
drwx------ 2 root root    12288 Nov 24  2012 lost+found
-rw-r--r-- 1 root root  2664392 Sep 28 00:59 System.map-3.16-0.bpo.2-amd64
-rw-r--r-- 1 root root  3126096 Sep 28 00:48 vmlinuz-3.16-0.bpo.2-amd64
```

```
# df -h
Filesystem             Size  Used Avail Use% Mounted on
...
/dev/sdb1              234M   28M  206M  13% /media/....
```

As you can see in the following partition table of the Debian legacy boot system my `/boot` partition is number 1 (`/dev/sdb1`).

Although 1 is the default value for standard debian installations better check!

The live system has identified this partition as `/dev/sdb`. The debian system on your harddisk could reference it differently.

[1] https://www.debian.org/CD/live/

Partition table of the Debian legacy boot system

```
# fdisk -l /dev/sdb
...
   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1   *        2048      499711       44032    7  HPFS/NTFS/exFAT
...
/dev/sdb5          501760   976771071   488134656   83  Linux
```

In legacy boot mode the `/boot` partition must have the `boot`-flag (*) set. This confirms our assumption: the `/boot` filesystem is on: `/dev/sdb1`.

```
# gdisk -l /dev/sdb
GPT fdisk (gdisk) version 0.8.5

Partition table scan:
  MBR: MBR only
  BSD: not present
  APM: not present
  GPT: not present
...
Number  Start (sector)    End (sector)  Size       Code  Name
   1            2048          499711   243.0 MiB   8300  Linux filesystem
   5          501760       976771071   238.2 GiB   8300  Linux filesystem
```

2.3. Create GPT partition table

Transform the partition table from MBR to GPT with

```
  #gdisk /dev/sdb

  r       recovery and transformation options (experts only)
  f       load MBR and build fresh GPT from it
```

2.4. Create an UEFI partition

A good graphical tool is the Gnome Partition Editor `gparted`:

```
# gparted /dev/sdb
```

1. Shrink the `/root` partition to 200MB in order to free 43MB (see partition 1 below).

2. Create a new 43MB partition for efi using `gparted` with partition code `EF00` (EFI system) and flag it **bootable**. Format the partition with a *fat32* [2] filesystem (see partition 2 below).
3. UEFI needs additionally [3] a *not* formatted 1MB partition . [4] (see partition 3 below).

Leave the other partitions untouched (see partition 5 below).

Here the result:

Partition table of the Debian UEFI boot system

```
# gdisk -l /dev/sdb
GPT fdisk (gdisk) version 0.8.5

Partition table scan:
  MBR: protective
  BSD: not present
  APM: not present
  GPT: present

Found valid GPT with protective MBR; using GPT.
Disk /dev/sdb: 976773168 sectors, 465.8 GiB

...

Number  Start (sector)    End (sector)  Size       Code  Name
   1            2048          411647   200.0 MiB   8300  Linux filesystem
   2          411648          499711   43.0 MiB    EF00  Efi partition
   3          499712          501759   1024.0 KiB  8300  Linux filesystem
   5          501760       976771071   465.5 GiB   8300  Linux filesystem
```

[5]

3. Mount the Debian filesystem

The next step differs whether the `root`-filesystem is encrypted or not.

[2] fat32=vfat in `/etc/fstab`

[3] I have not verified if the additional 1MB partition is really necessary. Omitting this step the following error message may occur: `GPT detected. Please create a BIOS-Boot partition (>1MB, unformatted filesystem, bios_grub flag). This can be performed via tools such as Gparted. Then try again.`

[4] Some say it should have the flag **bios_grub**, for me it works without.

[5] I noticed on my system the code EF00 changed somehow to 0700. Why?

3.1. Mount a non-encrypted `root` -filesystem

1. Mount the `/` (root) filesystem.
   - For non-encrypted root filesystems a simple `mount` will do.

```
# mount -t ext4 /dev/sdb5 /mnt
```

3.2. Mount an encrypted `root` -filesystem

- For encrypted root filesystems the mounting procedure can be a little tricky especially when the root filesystem resides inside a logical volume which is encrypted. This section shows how to mount and unmount an encrypted `root` -filesystem.

> The recovery mode of the Debian 9 Stretch installer disk automates all following steps. Try this first. If it does not work follow the rest of this section.

Find the device and partition of the to be mounted logical volume

1. Connect the disk with `host-system` and observe the kernel messages in `/var/log/syslog`

```
root@host-system:~# tail -f /var/log/syslog
sd 3:0:0:0: [sdb] 976773168 512-byte logical blocks: (500 GB/465 GiB)
sd 3:0:0:0: [sdb] Write Protect is of manually.
sd 3:0:0:0: [sdb] Mode Sense: 43 00 00 00
sd 3:0:0:0: [sdb] Write cache: enabled, read cache: enabled, doesn't
support DPO or FUA
 sdb: sdb1 sdb2 sdb3 sdb5
sd 3:0:0:0: [sdb] Attached SCSI disk
```

   The to be mounted device is `/dev/sdb` .

2. Find the partition

```
root@host-system:~# gdisk -l /dev/sdb
GPT fdisk (gdisk) version 0.8.5
...
```

```
Number  Start (sector)    End (sector)  Size       Code  Name
   1            2048          411647   200.0 MiB   8300  Linux
 filesystem
   2          411648          494821   43.0 MiB    0700
   3          494822          501759   1024.0 KiB  8300  Linux
 filesystem
   5          501760       976771071   465.5 GiB   8300  Linux
 filesystem
```

The to be mounted logical volume of `disk-system` resides on `/dev/sdb5`.

Mount encrypted logical volume

1. Open decryption layer.

```
root@host-system:~# lvscan
  ACTIVE            '/dev/host-system/root' [231.03 GiB] inherit
  ACTIVE            '/dev/host-system/swap_1' [7.20 GiB] inherit
```

Logical volume is not registered yet. Do so.

```
root@host-system:~# cryptsetup luksOpen /dev/sdb5 sdb5_crypt
Enter passphrase for /dev/sdb5:
```

Enter disk password.

```
root@host-system:~# lvscan
  inactive          '/dev/disk-system/root' [457.74 GiB] inherit
  inactive          '/dev/disk-system/swap_1' [7.78 GiB] inherit
  ACTIVE            '/dev/host-system/root' [231.03 GiB] inherit
  ACTIVE            '/dev/host-system/swap_1' [7.20 GiB] inherit
```

Logical volume of `disk-system`is registered now. It contains one `root` partition (line 1) and and one `swap` partition (line 2).

2. Activate logical volumes

```
root@host-system:~# lvchange -a y disk-system
```

Check success.

```
root@host-system:~# lvscan
  ACTIVE            '/dev/disk-system/root' [457.74 GiB] inherit
  ACTIVE            '/dev/disk-system/swap_1' [7.78 GiB] inherit
  ACTIVE            '/dev/host-system/root' [231.03 GiB] inherit
  ACTIVE            '/dev/host-system/swap_1' [7.20 GiB] inherit

root@host-system:~# ls /dev/mapper
control  disksystem-root  disksystem-swap_1  hostsystem-root
 hostsystem-swap_1  mymapper  sdb5_crypt
```

3. Mount logical volume

```
root@host-system:~# mount -t ext4 /dev/mapper/disksystem-root /mnt
```

Check success.

```
root@host-system:~# ls /mnt
bin   etc         initrd.img.old  lib64       mnt   proc  sbin     sys
  var
boot  home        lib             lost+found  mnt2  root  selinux  tmp
  vmlinuz
dev   initrd.img  lib32           media       opt   run   srv      usr
  vmlinuz.old
```

Unmount encrypted logical volume

This subsection is only for completeness. Skip it.

```
root@host-system:~# umount /mnt

root@host-system:~# lvscan
  ACTIVE            '/dev/disk-system/root' [457.74 GiB] inherit
  ACTIVE            '/dev/disk-system/swap_1' [7.78 GiB] inherit
  ACTIVE            '/dev/host-system/root' [231.03 GiB] inherit
  ACTIVE            '/dev/host-system/swap_1' [7.20 GiB] inherit

root@host-system:~# lvchange -a n disk-system
root@host-system:~# lvscan
  inactive          '/dev/disk-system/root' [457.74 GiB] inherit
  inactive          '/dev/disk-system/swap_1' [7.78 GiB] inherit
  ACTIVE            '/dev/host-system/root' [231.03 GiB] inherit
  ACTIVE            '/dev/host-system/swap_1' [7.20 GiB] inherit
```

```
root@host-system:~# cryptsetup luksClose sdb5_crypt
root@host-system:~# lvscan
  ACTIVE            '/dev/host-system/root' [231.03 GiB] inherit
  ACTIVE            '/dev/host-system/swap_1' [7.20 GiB] inherit
```

3.3. Mount the remaining filesystems

Either this way…

```
# mount /dev/sdb1 /mnt/boot
# mount /dev/sdb2 /mnt/boot/efi
# for i in /dev/ /dev/pts /proc /sys ; do mount -B $i /mnt/$i ; done
```

or this way, both commands do the same…

```
# mount /dev/sdb1 /mnt/boot
# mount /dev/sdb2 /mnt/boot/efi
# mount --bind /sys /mnt/sys
# mount --bind /proc /mnt/proc
# mount --bind /dev /mnt/dev
# mount --bind /dev/pts /mnt/dev/pts
```

Internet access

For internet access inside chroot:

```
# cp /etc/resolv.conf /mnt/etc/resolv.conf
```

4. Update debians /etc/fstab

Update the entries in `/mnt/etc/fstab` to reflect the partition changes above. Compare the UUID’s there with the ones listed here:

```
# ls /dev/disk/by-uuid
```

Add the new UEFI partition (see last line in `/etc/fstab` below) in order to get it mounted permanently on `/boot/efi`.

```
# cat /mnt/etc/fstab
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
```

```
/dev/mapper/koobue1-root /      ext4    errors=remount-ro 0       1
# /boot was on /dev/sdb1 during installation
UUID=040cdd12-8e45-48bd-822e-7b73ef9fa09f  /boot   ext2  defaults 0  2
/dev/mapper/koobue1-swap_1 none swap    sw              0       0
/dev/sr0        /media/cdrom0   udf,iso9660 user,noauto     0       0
#Jens: tmpfs added for SSD
tmpfs           /tmp            tmpfs   defaults,nodev,nosuid,size=500m 0
 0
tmpfs           /var/lock       tmpfs
 defaults,nodev,nosuid,noexec,mode=1777,size=100m 0 0
tmpfs           /var/run        tmpfs
 defaults,nodev,nosuid,noexec,mode=0775,size=100m 0 0
UUID=19F0-4372  /boot/efi       vfat    defaults        0       2
```

I use `/dev/mapper` for the encrypted file system and `tmpfs` because I have an SSD disk.

5. Inside the `chroot` environment

5.1. Preparation

Enter with:

```
# chroot /mnt
```

Check

```
# cat /etc/fstab
```

for not yet mounted entries and mount them manually e.g.

```
# mount /tmp
# mount /run
# mount /var/lock
...
```

5.2. Install grub-efi

```
# apt-get remove grub-pc
# apt-get install grub-efi
```

```
# grub-install /dev/sdb
```

Check presence of the efi file:

```
# file /boot/efi/EFI/debian/grubx64.efi
/boot/efi/EFI/debian/grubx64.efi: PE32+ executable (EFI application)
 x86-64 (stripped to external PDB), for MS Windows
```

A Debian entry should be listed here:

```
# efibootmgr
BootCurrent: 0000
Timeout: 0 seconds
BootOrder: 0000,2001,2002,2003
Boot0000* debian
Boot2001* EFI USB Device
Boot2002* EFI DVD/CDROM
Boot2003* EFI Network
```

Exit chroot environment.

```
exit
```

Reboot the system.

6. Validate the debian bootloader in UEFI Bios

The bios will not accept the bootloader by default, because `/EFI/debian/grubx64.efi` is not the default path and because the file has no Microsoft signature.

This is why `grubx64.efi` has to be validated manually in the UEFI bios setup. In my InsydeH20 Bios I selected:

Security → Select an UEFI file as trusted → Enter

Then browse to

```
/EFI/debian/grubx64.efi
```

in order to insert the grub boot loader in the trusted bootloader bios database.

On my Acer E3-111 the bios menu entry was disabled by default. To enable it I had to define first a supervisor password.
Security → Set Supervisor Password → Enter

7. References

Tanguy

Tanguy: *Debian: switch to UEFI boot.* http://tanguy.ortolo.eu/blog/article51/debian-efi. April 2012.

Vulcan

Vulcan, Silviu: *Linux on the Acer E3-111 – Aspire E3-111-C5FN*. http://www.sgvulcan.com/linux-on-the-acer-e3-111-aspire-e3-111-c5fn/ . 09/2014.