Solid State Drives

Installing Slackware 14.1 on a SSD drive

While some of this could be performed after an installation (changing the LVM settings), I'll assume a new installation, because most parts can't be easily performed afterwards.

Partition creation

Create your partitions with fdisk, as cfdisk doesn't automatically aligns your partitions correctly. Parted seems also to do this right and also has a function to check the alignments of partitions. To do this start parted and use the `align-check opt` command:

```
# parted
(parted) align-check opt <partition number>
```

parted now displays `<partition number> aligned` if the partition is aligned correctly, or `<partition number> not aligned` if not.

Compare: https://www.gnu.org/software/parted/manual/parted.html#align_002dcheck

SSDs and LUKS

Here are some points to be observed. First, SSDs and encryption are not a really perfect combination. For best security there should be no information about the contents of the encrypted container, but SSDs internally “load-balance” the write cycles so that the memory is used equally (if you delete something, the device gets notified about the new free space). This method is called TRIM. LUKS intentionally prevents this by never telling the drive which parts of it are in use and which are free for balancing.

This behavior can be turned off, but naturally this has an impact on the security of the encryption. For more information about this, read: http://asalor.blogspot.com/2011/08/trim-dm-crypt-problems.html

In the following I'll show the steps that are required to enable TRIM with LUKS, if you decide to use it that way. It is assumed that a LUKS+LVM setup is used, as described in `README_CRYPT.TXT` section “Combining LUKS and LVM” to which I'll refer for the commands issued.

Formatting the LUKS container

If you have decided to use TRIM, you can skip the part of writing random data to the partition. Format the LUKS container as described in the mentioned section of `README_CRYPT.TXT` :

```
# cryptsetup -s 256 -y luksFormat /dev/sdx2
```

Opening the LUKS container

When opening the encrypted partition, you have to use the parameter `--allow-discards`:

```
# cryptsetup --allow-discards luksOpen /dev/sdx2 lukssda2
```

Modifing the initrd script

Then continue as described in `README_CRYPT.TXT` with creating the LVM volumes. The next part where care has to be taken is creating the initrd. Here you have to manually patch the

```
initrd-tree/init
```

script to assure that the LUKS containers are opened with the

```
--allow-discards
```

parameter.

Populate `initrd-tree` by executing `mkinitrd` as described in `README_CRYPT.TXT`. Then, open `initrd-tree/init` with the editor of your choice and insert `--allow-discards` at these commands:

```
/sbin/cryptsetup ${LUKSKEY} luksOpen ${LUKSDEV} ${CRYPTDEV} </dev/tty0
>/dev/tty0 2>&1
```

changes to

```
/sbin/cryptsetup --allow-discards ${LUKSKEY} luksOpen ${LUKSDEV} ${CRYPTDEV}
</dev/tty0 >/dev/tty0 2>&1
```

there should be two occurrences of luksOpen in the file. Now with the file changed, call mkinitrd again, but this time without the `-c` option to prevent the deletion of your modifications.

Continue as described, with modifying your lilo configuration. Because LVM is used in this setup, read also the next section (which could also be performed after rebooting).

SSDs and LVM

The TRIM functionality mentioned in the LUKS section must also specifically enabled for LVM. To activate the issuing of these commands, change

```
issue_discards = 0
```

in `/etc/lvm/lvm.conf` to

```
issue_discards = 1
```

and reboot.

howtos, ssd