Fedora Silverblue With Disk Encryption:

How I Almost Lost Everything But Gained Much Wisdom (Side Story: Bmaptool And Ddrescue: Why One Should Never Ever Use Dd)

Marcel Ziswiler

Speaking today

Marcel Ziswiler

ROLE

Founder ZisiSoft GmbH

Consultant Software Engineer
Codethink Ltd.

TENURE

Joined Codethink
in 2024

PAST ENGAGEMENTS

Senior Linux Expert, System Engineer, Technical Project Leader
Noser Engineering

Platform Manager Embedded Linux
Toradex

EDUCATION

MS Computer Science
ETH Zurich

Certificate in Embedded Systems Technologies UCI

Contents

Disaster Struck Timeline

What happened?

- Testing different RADXA ROCK 5B board boot options
  - Booting from eMMC, NVMe, SPI, USB, uSD
- Inadvertently typed `sudo dd of=/dev/nvme0n1` on my notebook!
- Unfortunately the image was around 8 GB in size!
  - Doesn't sound that big given my 2 TB NVMe SSD
- Overwrote entire boot chain plus several GB well into the encrypted system partition
  - no backup of the LUKS header

lsblk (abbreviated)

```
nvme0n1         1.8T  0 disk
├─nvme0n1p1     600M  0 part
/boot/efi
├─nvme0n1p2       1G  0 part
/boot
└─nvme0n1p3     1.8T  0 part
└─luks-24d7de45-bb6d-49f7-914a-9
17810617f8f
```

Gosh, what happened

Disaster Struck – Timeline

```
Oct 02 09:32:41 fedora kernel: hub 8-1.2.4.4.4:1.0: 4 ports detected
=> docked in

Oct 02 09:57:27 fedora sudo[73444]:      zim : TTY=pts/9 ;
PWD=/var/home/zim/Downloads/aarch64/tmp ; USER=root ; COMMAND=/usr/sbin/losetup -P -f --show
disk.img
…
Oct 02 09:58:06 fedora sudo[73490]:      zim : TTY=pts/9 ;
PWD=/var/home/zim/Downloads/aarch64/tmp ; USER=root ; COMMAND=/usr/bin/mount /dev/loop0p1
/mnt
=> poked the disk image

Oct 02 09:59:23 fedora kernel: usb 2-1: new SuperSpeed USB device number 7 using xhci_hcd
Oct 02 09:59:23 fedora kernel: usb 2-1: New USB device found, idVendor=0781, idProduct=5591,
bcdDevice= 1.00
Oct 02 09:59:23 fedora kernel: usb 2-1: New USB device strings: Mfr=1, Product=2,
SerialNumber=3
Oct 02 09:59:23 fedora kernel: usb 2-1: Product:  SanDisk 3.2Gen1
=> plugged in USB stick which should have been written instead
```

Disaster Struck – Timeline

```
Oct 02 10:00:53 fedora sudo[73877]:      zim : TTY=pts/9 ;
PWD=/var/home/zim/Downloads/aarch64 ; USER=root ; COMMAND=/usr/bin/ls
=> dirty trick doing a sudo ls to prime shell for disaster?

Oct 02 10:00:59 fedora sudo[73889]:      zim : TTY=pts/9 ;
PWD=/var/home/zim/Downloads/aarch64 ; USER=root ; COMMAND=/usr/bin/dd conv=fsync
of=/dev/nvme0n1 bs=4M status=progress
=> here you go, disaster struck!

Oct 02 10:01:22 fedora udisksd[1379]: Error getting '/dev/nvme0n1p3' metadata_size:
Failed to initialize device: No such device (g-bd-crypto-error-quark, 1)
=> yep, udisksd kinda noticed it pretty quick

Oct 02 10:01:51 fedora sudo[73926]:      zim : TTY=pts/9 ;
PWD=/var/home/zim/Downloads/aarch64 ; USER=root ; COMMAND=/usr/sbin/blockdev
--flushbufs /dev/sda
=> yes, I flushed the USB stick :)
```

Disaster Struck – Timeline

```
Oct 02 10:02:13 fedora kernel: usb 2-1: USB disconnect, device number 7
Oct 02 10:02:13 fedora systemd-homed[1376]: block device
/sys/devices/pci0000:00/0000:00:08.1/0000:c1:00.3/usb2/2-1/2-1:1.0/host0/target0:0:0/0:0:0:0
/block/sda/sda1 has been removed.
=> removed it and tried on my embedded system without much luck :(

Oct 02 10:06:35 fedora sudo[74347]:      zim : TTY=pts/9 ;
PWD=/var/home/zim/Downloads/aarch64 ; USER=root ; COMMAND=/usr/bin/dd conv=fsync
of=/dev/nvme0n1 bs=4M status=progress
=> so I actually re-tried it again a few minutes later (from command history)

Oct 02 10:06:58 fedora udisksd[1379]: Error getting '/dev/nvme0n1p3' metadata_size: Failed
to initialize device: No such device (g-bd-crypto-error-quark, 1)
=> yep, udisksd is still not happy

Oct 02 10:07:08 fedora sudo[74363]:      zim : TTY=pts/9 ;
PWD=/var/home/zim/Downloads/aarch64 ; USER=root ; COMMAND=/usr/sbin/blockdev --flushbufs
/dev/sda
=> :)
```

Disaster Struck – Timeline

```
Oct 02 10:07:15 fedora kernel: usb 2-1: USB disconnect, device number 9
…
Oct 02 10:07:15 fedora systemd-homed[1376]: block device
/sys/devices/pci0000:00/0000:00:08.1/0000:c1:00.3/usb2/2-1/2-1:1.0/host0/target0:0:0/0:0:0:0
/block/sda has been removed.
=> removed it again

Oct 02 10:07:21 fedora kernel: usb 2-1: new SuperSpeed USB device number 10 using xhci_hcd
Oct 02 10:07:21 fedora kernel: usb 2-1: New USB device found, idVendor=0951, idProduct=1666,
bcdDevice= 1.00
Oct 02 10:07:21 fedora kernel: usb 2-1: New USB device strings: Mfr=1, Product=2,
SerialNumber=3
Oct 02 10:07:21 fedora kernel: usb 2-1: Product: DataTraveler 3.0
Oct 02 10:07:21 fedora kernel: usb 2-1: Manufacturer: Kingston
=> trying another brand USB stick?

Oct 02 10:21:19 fedora sudo[74954]:      zim : TTY=pts/9 ;
PWD=/var/home/zim/Downloads/aarch64 ; USER=root ; COMMAND=/usr/bin/dmesg
=> wait a minute, …
```

Disaster Struck – Timeline

```
Oct 02 10:21:36 fedora sudo[74966]:      zim : TTY=pts/9 ;
PWD=/var/home/zim/Downloads/aarch64 ; USER=root ; COMMAND=/usr/bin/dmesg
=> something doesn't feel right

Oct 02 10:26:59 fedora kernel: usb 1-4: reset full-speed USB device number 2 using
xhci_hcd
=> undocked

Oct 02 10:27:04 fedora sudo[75226]:      zim : TTY=pts/2 ; PWD=/tmp ; USER=root ;
COMMAND=/usr/bin/ls -l /boot/efi/
…
Oct 02 10:27:51 fedora sudo[75284]:      zim : TTY=pts/2 ; PWD=/tmp ; USER=root ;
COMMAND=/usr/bin/tar cfj boot-efi.tar.bz /boot/efi
…
Oct 02 10:28:13 fedora sudo[75295]:      zim : TTY=pts/2 ; PWD=/tmp ; USER=root ;
COMMAND=/usr/bin/tar cfj boot.tar.bz /boot
=> poked and backed-up my boot chain
```

Disaster Struck – Timeline

```
Oct 02 10:28:13 fedora kernel: EXT4-fs warning (device nvme0n1p2):
ext4_dirblock_csum_verify:406: inode #23: comm tar: No space for
directory leaf checksum. Please run e2fsck -D.
Oct 02 10:28:13 fedora kernel: EXT4-fs error (device nvme0n1p2):
htree_dirblock_to_tree:1083: inode #23: comm tar: Directory block failed
checksum
=> /boot partition starts going bad

Oct 02 10:29:45 fedora flatpak[75414]: libostree pull from 'flathub' for
app/com.slack.Slack/x86_64/stable complete
=> updates running in the background don't help
```

Disaster Struck – Timeline

```
Oct 02 10:30:01 fedora kernel: BTRFS error (device dm-0): bad tree block start, mirror
1 want 762494976 have 11287069813007085624
Oct 02 10:30:01 fedora kernel: BTRFS error (device dm-0): bad tree block start, mirror
2 want 762494976 have 16203133442273220086
Oct 02 10:30:01 fedora kernel: BTRFS error (device dm-0 state A): Transaction aborted
(error -5)
Oct 02 10:30:01 fedora kernel: BTRFS: error (device dm-0 state A) in
__btrfs_free_extent:3235: errno=-5 IO failure
=> now system partition went bad

Oct 02 10:30:01 fedora kernel: BTRFS info (device dm-0 state EA): forced readonly
Oct 02 10:30:01 fedora kernel: BTRFS error (device dm-0 state EA): failed to run
delayed ref for logical 34041208832 num_bytes 12288 type 178 action 2 ref_mod 1: -5
Oct 02 10:30:01 fedora kernel: BTRFS: error (device dm-0 state EA) in
btrfs_run_delayed_refs:2223: errno=-5 IO failure
=> the end of the journal as we know it
```

Disk Encryption

Block Device Encryption

- Encrypts/decrypts data transparently
- Underlying block device sees only encrypted data
- Passphrase required to mount
- Additional security beyond existing OS security mechanisms
- Protects device contents even if physically removed from system

During installation in Anaconda

Checking "Encrypt System" checkbox on "Automatic Partitioning" screen

Better safe than sorry

Using dm-crypt/LUKS

LUKS (Linux Unified Key Setup)

- specification for block device encryption
- on-disk format for the data
- passphrase/key management policy

LUKS uses kernel device mapper subsystem via dm-crypt module

- low-level mapping handling encryption and decryption
- user-level operations (creating/ accessing encrypted devices) using cryptsetup utility

system startup

– presented with passphrase prompt

– enter correct passphrase

– system will continue to boot normally

Seamless.

LUKS Overview

- Encrypts entire block devices
- Protecting contents of mobile devices
  - Removable storage media
  - Laptop disk drives
- Underlying contents of encrypted block device are arbitrary
- Useful for encrypting swap devices
- Can be useful with certain databases using specially formatted block devices for data storage
- Uses existing device mapper kernel subsystem (like LVM, so well tested)
- Provides passphrase strengthening
- Protects against dictionary attacks
- LUKS devices contain multiple key slots
- Allows users to add backup keys/passphrases

LUKS not well-suited

- for applications requiring many (>8) users to have distinct access keys to same device
- for applications requiring file-level encryption

Tooling

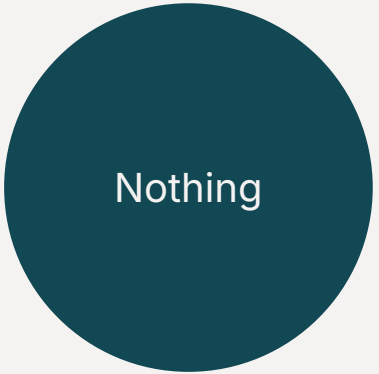
cryptsetup

Command-line tool for managing LUKS-encrypted disks

systemd-cryptsetup

Automatically unlocking encrypted disks at boot (e.g. using a keyfile)

ToDO



So far so good.

Fedora Silverblue Lessons Learned

During/After Installation

/boot partition is way too small (1 GB)

– limits (pinned) deployments

– disaster case overwrote important data way beyond

– suggest using at least 16 GB

Backup LUKS header

```
sudo cryptsetup luksHeaderBackup
/dev/nvme0n1p3 --header-backup-file
luks-header.img
```

– in case it ever gets corrupted

– typically only 16 MB

and

– backup your keys

Don't wait for disaster to strike.

Fedora Silverblue as a Daily Driver

- Use flatpak and toolbox aka toolbx :)
  - just relax - it works
  - if not flatseal might help
  - okay, worst case just copy your favourite/important tool from a toolbx to ~/.local/bin (;-p)
  - see below under bmaptool usage
- python
  - just use venv
- Blindness?
  - `gsettings set org.gnome.mutter experimental-features "['scale-monitor-framebuffer']"`

use upstream flathub

```
flatpak remove --all
flatpak remote-delete fedora
flatpak remote-add --if-not-exists
flathub
https://flathub.org/repo/flathub.
flatpakrepo
```

So far so good.

Libostree (previously OSTree)

- `sudo rpm-ostree status`
- `sudo ostree log fedora:fedora/41/x86_64/silverblue`
- `sudo rpm-ostree deploy 41.20250129.0`
- `sudo ostree remote refs fedora`
- `sudo rpm-ostree rebase fedora:fedora/41/x86_64/silverblue`
- ostree pinning
  - `sudo ostree admin pin 0`
  - `sudo ostree admin pin --unpin 3`

what is it?

– system for versioning updates of Linux-based operating systems

– "git for operating system binaries"

Atomic.

More Advanced Tricks

how about wireshark?

– gosh, did you seriously miss my USB talk (see references below)?

```
- sudo ~/.local/bin/tcpdump -i wlp1s0
  -U -w - | flatpak run
  --filesystem=host --file-forwarding=
  host --share=network
  org.wireshark.Wireshark -k -i -
```

and meld?

```
- alias meld='flatpak run
  org.gnome.meld'
```

ToDO

Watch my other talks :)

Anything can be made to work.

Bonus Track: DHCP/TFTP Servers (e.g. for Embedded use)?

```
sudo nmcli con add type vlan ifname devnet dev eth0 id 91 ip4 192.168.10.1/24 gw4
192.168.10.1
sudo nmcli con up vlan-devnet
sudo firewall-cmd --add-masquerade --permanent
sudo firewall-cmd --add-service=dhcp --permanent
sudo firewall-cmd --add-service=dns --permanent
sudo firewall-cmd --add-service=tftp --permanent
sudo firewall-cmd --zone=FedoraWorkstation --add-interface=devnet --permanent
sudo firewall-cmd --reload
sudo nmcli connection modify vlan-devnet ipv4.method shared
sudo nmcli connection modify vlan-devnet ipv6.method shared
sudo systemctl restart NetworkManager
sudo chmod -R 777 /var/lib/tftpboot
sudo chcon -Rt tftpdir_t /var/lib/tftpboot
[zim@fedora ~]$ cat /etc/NetworkManager/dnsmasq-shared.d/dhcp-tftp.conf
enable-tftp
tftp-root=/var/lib/tftpboot
dhcp-boot=flash.bin
dhcp-host=00:14:2d:67:28:6a,192.168.10.2
dhcp-range=192.168.10.2,static
```

Bmaptool

Generic Tool

- creating block map (bmap) for a file
- copying files using block map

faster and more reliable

- large files
  - raw system image files

- originally created for "Tizen IVI" project
- written in Python

faster depends on various factors

- write speed
- image size
- how full an image aka sparseness
- proves 5-7 times faster

Got it.

Integrity/Usability

Integrity

– verifies data integrity while flashing

– possible data corruptions noticed immediately

Usability

– read images directly from remote server

– no need to download images and save them locally

protects user's data

– mistake like specifying wrong block device name

– less likely destroy your data

– protection mechanisms: prevent writing to mounted block device

Aha!

Usage

```
zim@fedora:~$ sudo PYTHONPATH=~/.local/lib/python3.12/site-packages/ ~/.local/bin/bmaptool
--help
Place your finger on the fingerprint reader
^C[sudo] password for zim:
usage: bmaptool [-h] [--version] [-q] [-d] {create,copy} ...

A tool for creating block maps (bmap) and copying disk images using bmap files.
Documentation can be found here:
source.tizen.org/documentation/reference/bmaptool

options:
  -h, --help      show this help message and exit
  --version       show program's version number and exit
  -q, --quiet     be quiet
  -d, --debug     print debugging information

commands:
  {create,copy}
    create        generate bmap for an image file (which should be a sparse file)
    copy          write an image to a block device using bmap
```

Concepts

Sparse files

– filesystem maps blocks of file data to disk sectors
– disk index points to mapped blocks
– "free" area not mapped yet
– unmapped, contain "holes"

Block map

– xml file contains list of mapped areas

Raw images

– system images for target block devices
e.g. for embedded systems

okay

– mapping none-holes basically

Interesting.

Sparse files

- help save disk space as holes do not occupy any disk space
- reading data from a hole returns zeroes
- writing data to a hole destroys it by forcing file system to map corresponding file areas to disk sectors
- filesystems usually block based so mapping adheres to block boundary

not all tooling can handle sparse files (e.g. scp)

– some tooling requires a parameter (e.g. --sparse for tar and rsync)

Sweet.

Sparse Files – Multiple Implementations in Linux

- 2 ioctls to find mapped and unmapped areas
  - FIBMAP: old, widely supported, rather limited, requires root privileges
  - FIEMAP: more advanced, does not require root privileges
- lseek() system call with whence argument of SEEK_HOLE and SEEK_DATA
  - Allows positioning to next hole and next mapped area of a file
- fallocate() system call with FALLOC_FL_PUNCH_HOLE mode
  - punching holes: unmap aligned area, turn it into a hole

but

not all file systems support all or even one such mechanism

Welcome to the "real" world.

Ddrescue

GNU ddrescue

`ddrescue`

– data recovery tool

– copies data from one file or block device to another

– trying to rescue good parts first in case of read errors

`ddrescuelog`

– auxiliary tool

– manipulate ddrescue mapfiles

– basic operation fully automatic

and

No need to wait for an error, stop/restart, etc

Sounds promising.

Why One Should Never Ever Use Dd

Why One Should Never Ever Use Dd

Doesn't do any sanity checks

– overwrite mounted partition
– writing beyond partition boundary

Just stops encountering any error

– left with corrupt/partial data
– no easy way to continue

Worst-case

– forgot what happened?

and best-case

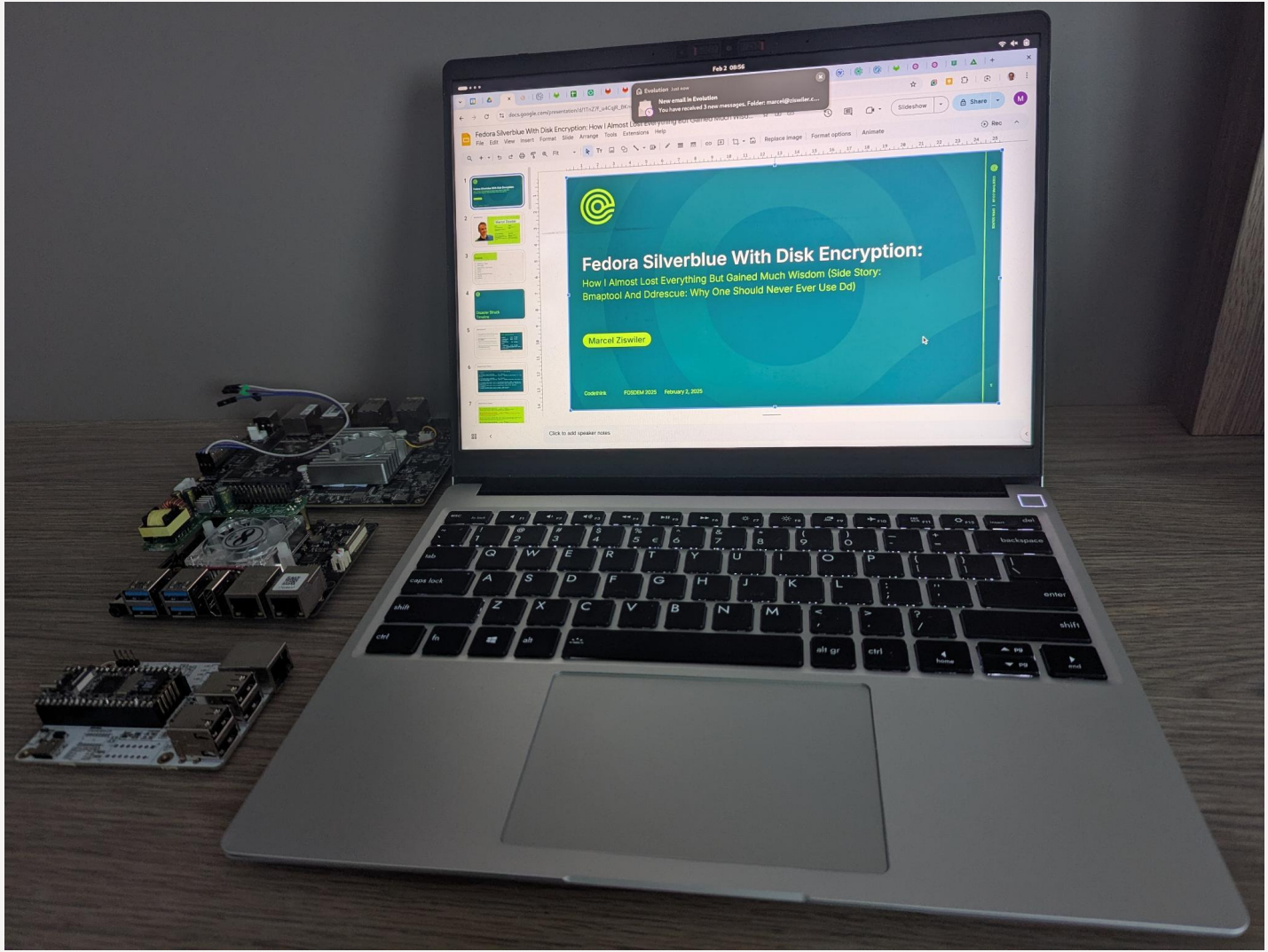
– it's just slow

– dirt slow

Let the matter rest!

Live Demo

Live Demo


Fedora Silverblue With Disk Encryption:
How I Almost Lost Everything But Gained Much Wisdom (Side Story:
Bmaptool And Ddrescue: Why One Should Never Ever Use Dd)
Marcel Ziswiler
CodeThink
FOSDEM 2025
February 2, 2025

References

References

- Fedora Disk Encryption User Guide
  https://docs.fedoraproject.org/en-US/quick-docs/encrypting-drives-using-LUKS
- Wireshark Usage (USB on Embedded Linux Systems Deep Dive - Debugging USB)
  https://youtu.be/QjULbx7oUVg?feature=shared&t=2103
- Bmaptool
  https://github.com/yoctoproject/bmaptool
- Ddrescue
  https://www.gnu.org/software/ddrescue

Thank You.

Codethink Ltd.

3rd Floor Dale House,
35 Dale Street,
MANCHESTER,
M1 2HF,
United Kingdom