Aeon

FDE is almost there

How do we tackle the last hurdles?

Richard Brown

About Me

Distribution Architect at SUSE

Former Sysadmin, Systems Manager, QA Engineer, Distribution/Release Engineer, openSUSE Board Member & Chair

10+ years at SUSE

20+ years in Open Source

Creator of Aeon

Agenda

Brief Introduction to Aeon Desktop

FDE on Aeon Desktop

Outstanding Challenges

Open Floor Discussion

Introducing Aeon Desktop

Reliable, Predictable & Immutable

Opinionated

- Supports GNOME only
- “Chromebook-like” experience
- Image-based installation

Minimal, yet Functional

- Printing, Gaming, Development and much more must all work

Works straight “out of the box”

- Additional configuration must not be needed before being able to get to work

FDE on Aeon Desktop

FDE on Aeon

July 2024 saw the release of Aeon RC3 with Full Disk Encryption

Enabled by Default, in one of two modes

- Default (TPM-backed automatic unlock)
- Fallback (Passphrase, for systems without sufficient TPM support)

Deployed using tik, a new installer which uses systemd-repart to configure FDE, pair with the TPM, and populate blocks/files.

TPM Measurements updated using sdbootutil

tik and FDE - Pre Deployment

- Probes system for TPM with PolicyAuthorizeNV
  - `tpm2_getcap commands | grep -q 'commandIndex: 0x192'`
- Checks for SecureBoot
  - `mokutil --sb-state | grep -q 'enabled'`
- Informs user which Encryption Mode they're getting
  - TPM with PolicyAuthoriseNV = Default Mode
  - TPM without PolicyAuthoriseNV = Fallback Mode
  - No TPM = Fallback Mode
  - Fallback Mode without SecureBoot = Fallback Mode with a grumpy warning

tik doesn't care if SecureBoot is enabled or disabled for Default Mode, but we do measure PCR 7 so whatever state it's in must stay the same

tik and FDE - Image Deployment

tik avoids doing as much encryption work as possible

- Creates random keyfile for use only during installation
  - `dd bs=512 count=4 if=/dev/urandom of=/tmp/tik.XXXXXXXXXX iflag=fullblock`

- Calls systemd-repart to deploy the image and conduct initial encryption
  - `systemd-repart --no-pager --pretty=0 --empty=force --dry-run=no --key-file=${tik_keyfile} --image=${image_file} --image-policy=root=unprotected ${image_target}`

systemd-repart config

00-esp.conf

```
[Partition]
Type=esp
Format=vfat
SizeMinBytes=4G
SizeMaxBytes=4G
```

50-root.conf

```
[Partition]
Type=root
CopyBlocks=auto
Encrypt=key-file
```

tik and FDE - Post Deployment Part 1

tik must mount the freshly deployed installation and finish configuring the OS

- Correct /etc/fstab and /etc/cmdline with the UUIDs of the ESP and Root partitions
- Populate /etc/crypttab
- Populate ESP with systemd-boot, kernels and initrd
  - `sdbootutil install`
  - `sdbootutil add-all-kernels`

tik and FDE - Post Deployment Part 2 (Default Mode)

For Default Mode there are the following additional steps

- Configure PCR policy
- Update TPM Predictions & Enrol to TPM
  - `sdbootutil update-predictions`
  - `systemd-cryptenroll --unlock-key-file=${tik_keyfile} --tpm2-device=auto ${cryptpart}`

tik and FDE - Post Deployment Part 2 (Fallback Mode)

For Fallback Mode there are the following additional steps

- Prompt for Encryption Passphrase

tik and FDE - Post Deployment Part 3

Both Encryption modes still benefit from having a generated recovery key

tik and FDE - Post Deployment Part 4

The temporary keyfile is wiped and unenrolled

- `systemd-cryptenroll --unlock-key-file=${tik_keyfile} --wipe-slot=0 ${cryptpart}`

Life with FDE

Mostly painless

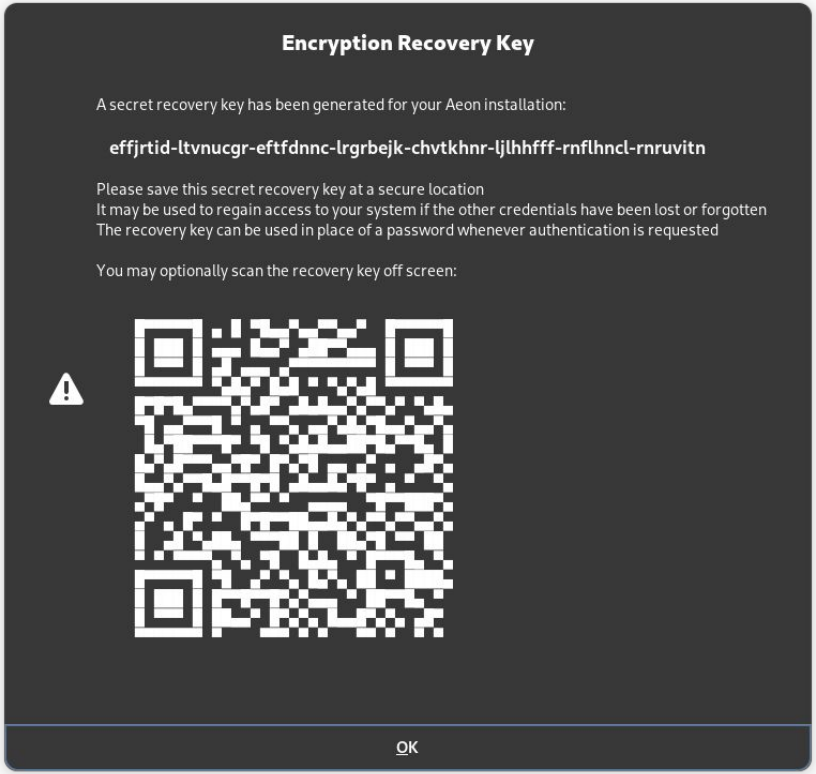
sdbootutil updates predictions automatically with package updates & system rollbacks

Custom aeon-check tool needed every now and again to adjust PCR Policy defaults & update predictions

“It just works”

Outstanding Challenges

Keyboard Layouts Suck



Multi-Factor Authentication

TPM+PIN is great, but scary:

Note that incorrect PIN entry when unlocking increments the TPM dictionary attack lockout mechanism, and may lock out users for a prolonged time, depending on its configuration. The lockout mechanism is a global property of the TPM, **systemd-cryptenroll** does not control or configure the lockout mechanism

Are we dependant on TPM manufacturers to enable TPM+FIDO2 or TPM+FIDO2+PIN?

PCR Policy

Aeon started measuring PCR's 0,4,5,7,9

- 0 - Core UEFI Firmware Executables
- 4 - Boot Loader and Drivers
- 5 - GPT Partition Table
- 7 - SecureBoot State
- 9 - initrd + kernel + cmdline

Worked mostly alright, except when fwupd updated the UEFI

Following the advice in systemd-cryptsetup docs now measuring only 4,5,7,9

Is this enough?

Probably Not - PCR 15?

https://oddlama.org/blog/bypassing-disk-encryption-with-tpm2-unlock/

Strongly recommends the use of PCR 15 to confirm you're actually booting a volume with the correct encryption key, machine-id, UUID & Label.

Much of the existing systemd tooling for measuring values into PCR 15 assume use of UKIs/systemd-stub

Are UKI's really the one-true-path to a secure boot chain?

Open Floor Discussion

Possible Topics

- What to do about PolicyAuthorizeNV or TPM-less hardware?
- Are UKIs the one true path forward?
- Should there be a recommended list of PCR's to measure?
- Should PCR 15 be considered mandatory?
- How can we make Recovery Keys more friendly?