CLIP OS: Building a defense-in-depth OS with the Linux kernel and open source software

Timothée Ravier, Nicolas Godinho, Thibaut Sautereau

Agence nationale de la sécurité des systèmes d'information (ANSSI)

Ready for IT

May 20–22, 2019

About the ANSSI

- *Agence nationale de la sécurité des systèmes d'information*
- French authority in the area of cyberdefence, network and information security
- Provides its expertise and technical assistance to government departments and businesses and plays an enhanced role in supporting operators of vital importance.

CLIP OS project

CLIP OS?

- Linux distribution developed by the ANSSI
- Initially only available internally
- Now open source, mostly under the LGPL v2.1+
- Code and issue tracker hosted on GitHub[1][2]:
  - Version 4: available as reference and for upstream patch contribution
  - Version 5: currently developed version, alpha status, beta coming soon

[1] https://github.com/CLIPOS
[2] https://github.com/CLIPOS-Archive

CLIP OS?

Not yet another Linux distribution

- Not a generic/multi-purpose distribution

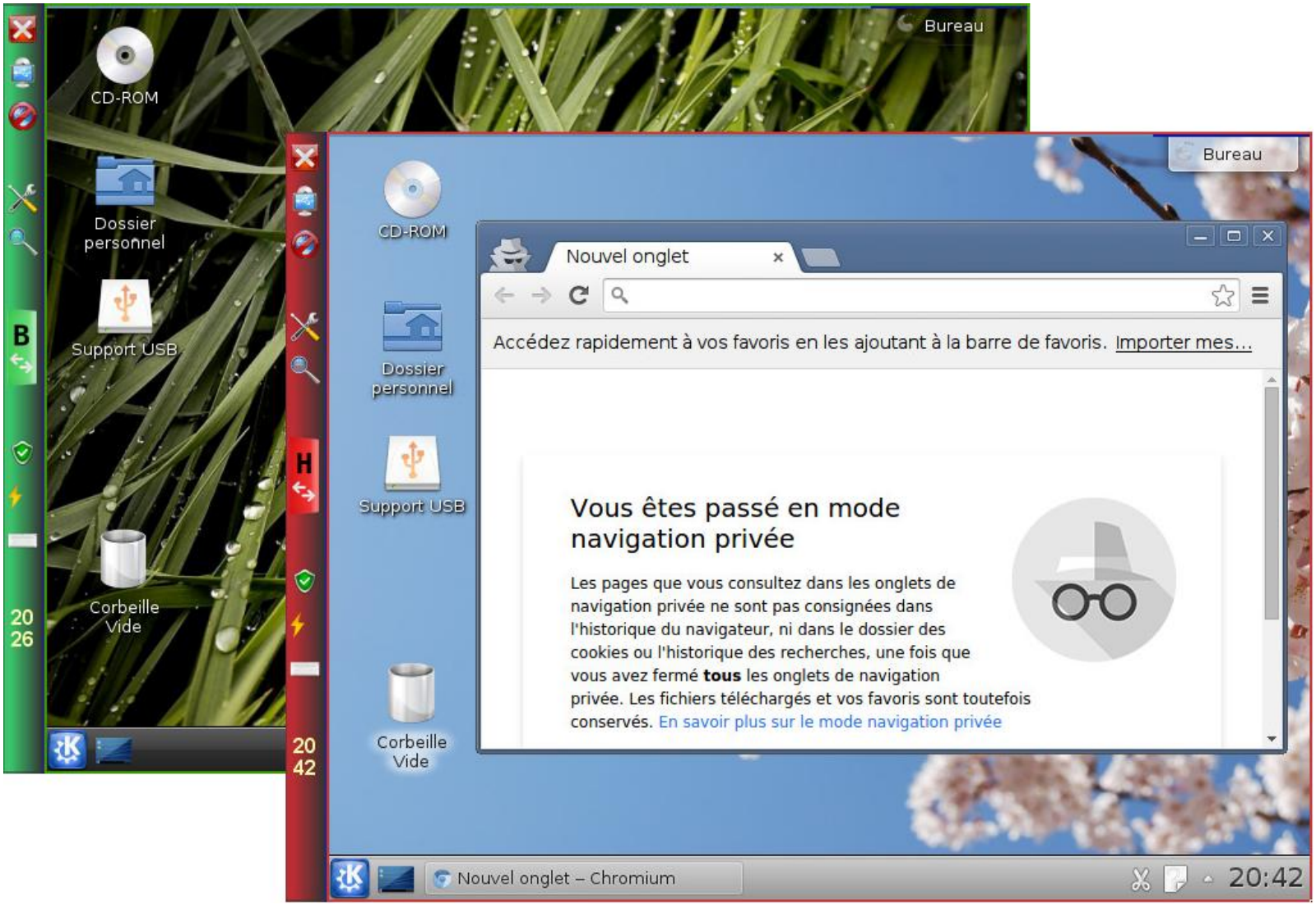
Targets three main use cases

- Office workstation
- Administration workstation
- IPsec gateway

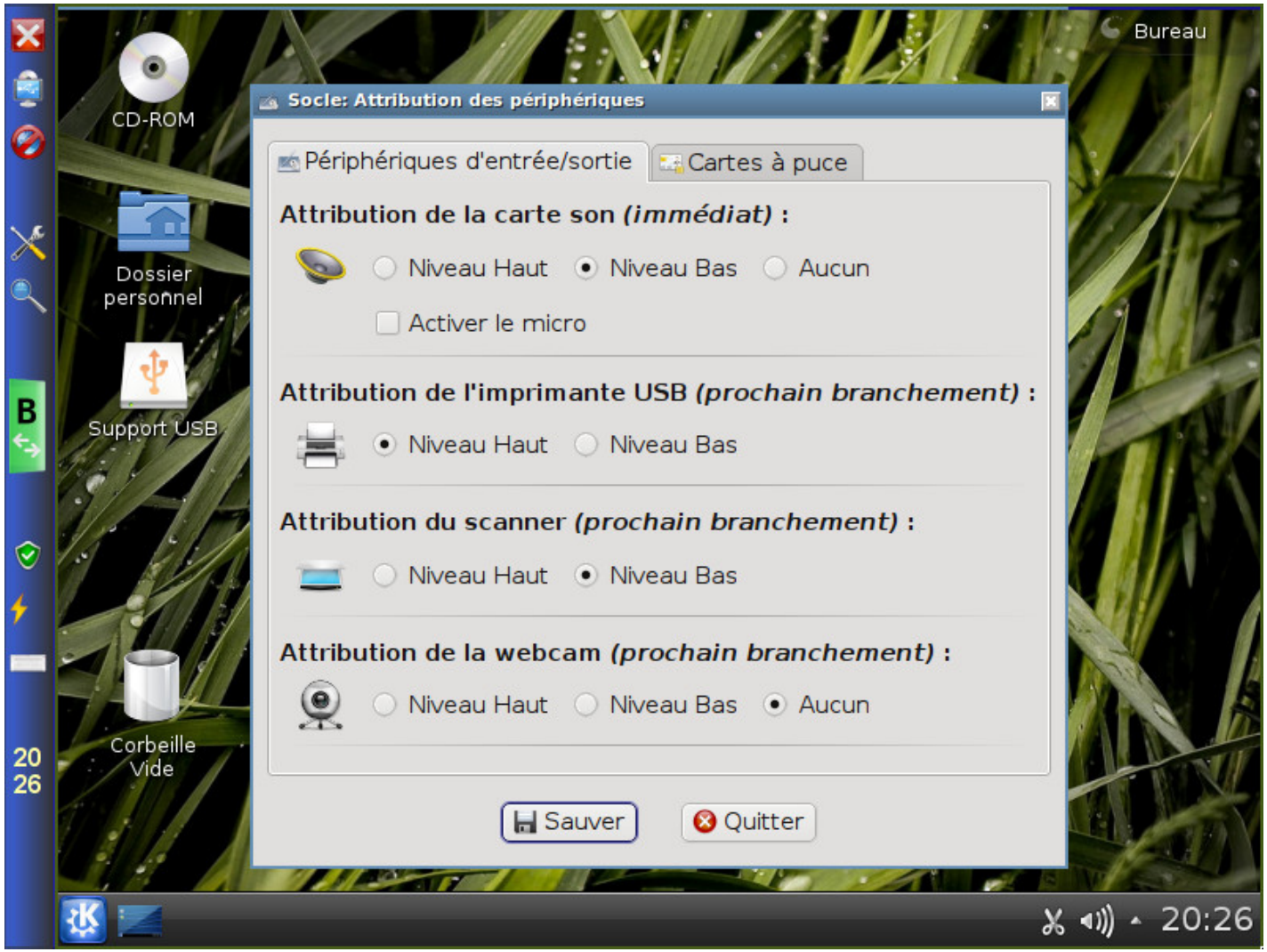
Hardened OS

- Based on Gentoo Hardened
- Hardened Linux kernel and confined services
- No interactive *root* account available:
  $\Rightarrow$ "Unprivileged" admin, audit and update roles
- Automatic updates using A/B partition model (similar to Android 7+)
- Multilevel security:
  - Provide two isolated user environments
  - Controlled interactions between isolated environments

Multilevel from the end user point of view (v4)



Admin panel: devices assignment per level (v4)



Differences with Qubes OS

CLIP OS development began 5 years earlier than Qubes OS

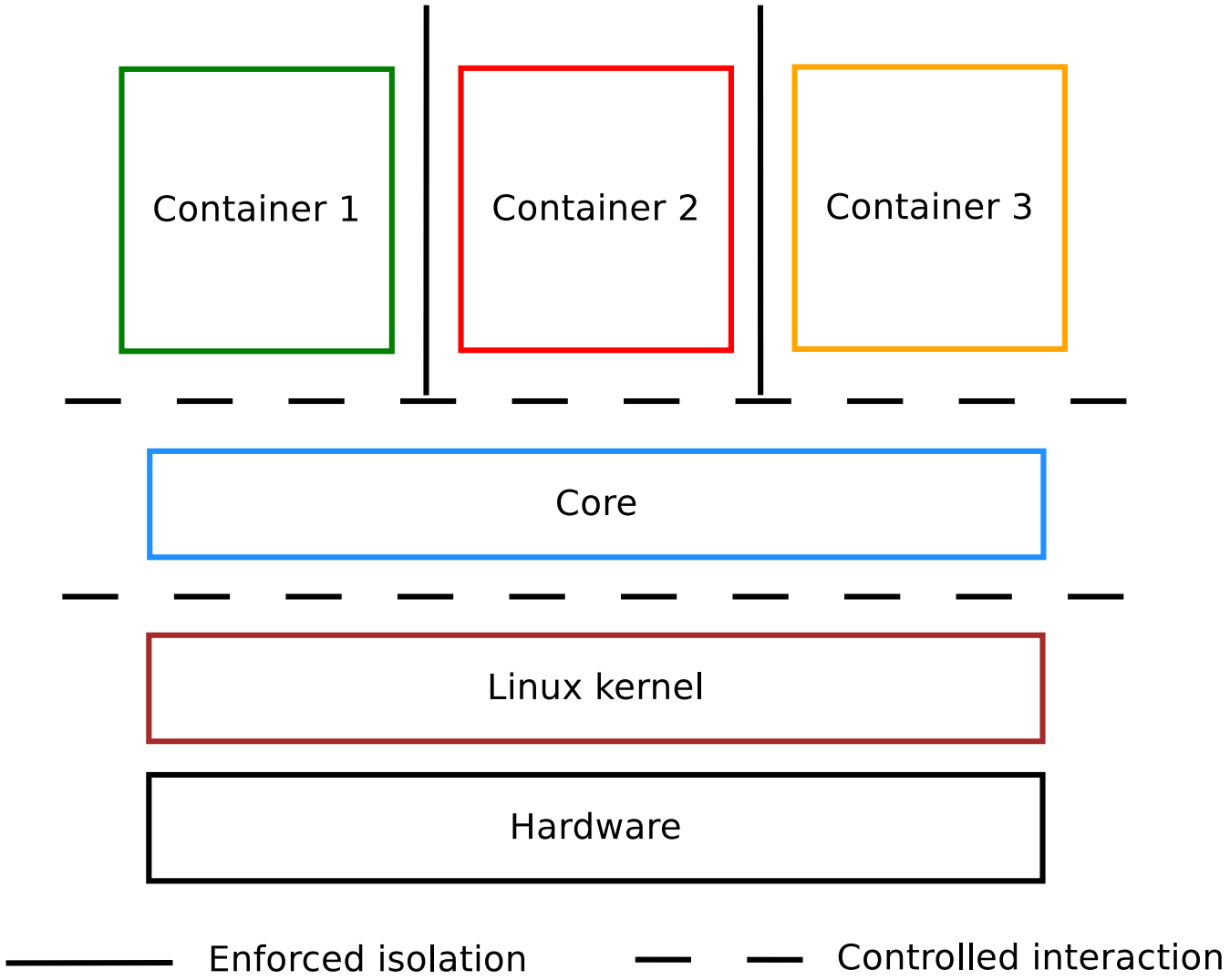
Main goals

- We target non-expert users
- Multilevel security model with two levels
- We favor a defense-in-depth approach

Technical point of view

- Hypervisor (Qubes OS) vs. supervisor isolation (CLIP OS)
- CLIP OS: Limited access rights and capabilities, even for administrators

General architecture overview



Project status (v5)

- First alpha release in September 2018
- Now close to beta release
- Current use-case: server & virtualization (no graphical user interface)

```
This is clipos-qemu.unknown_domain (Linux x86_64 5.0.14-clipos) 14:07:12

Hint: Num Lock on

clipos-qemu login: root
clipos-qemu ~ # lsblk
NAME                          MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
vda                           254:0    0   20G  0 disk
|-vda1                        254:1    0  512M  0 part  /mnt/efiboot
`-vda2                        254:2    0 19.5G  0 part
  |-mainvg-core_5.0.0--alpha.1 253:0    0    4G  0 lvm
  | `-verity_core_5.0.0-alpha.1 253:3    0  177M  1 crypt /
  |-mainvg-core_state          253:1    0  512M  0 lvm
  | `-core_state_dif           253:4    0  474M  0 crypt
  |   `-core_state             253:5    0  474M  0 crypt /mnt/state
  `-mainvg-core_swap           253:2    0    1G  0 lvm
    `-swap                     253:6    0    1G  0 crypt [SWAP]
clipos-qemu ~ # uname -sr
Linux 5.0.14-clipos
clipos-qemu ~ # _
```

Security features

Goals

- High resistance to remote or local exploits
- Defense in depth: limit impact of successful exploits
- Limited options for attacker persistence

Currently available

- Minimal system and hardened applications
- Curated Linux kernel configuration and hardware profiles
- Confined services, user and roles
- No arbitrary code execution ($W \oplus X$) enforced system wide
- Full boot chain integrity with UEFI Secure Boot
- Password-less encrypted partitions with TPM 2.0 support
- Expected for beta: Automatic, atomic, in-background updates

Development and contribution

Development workflow:

- Install dependencies
- Retrieve sources
- Automated build steps
- Test in a QEMU virtual machine

See full documentation at `https://docs.clip-os.org`:

Conclusion

Open source project

- Sources: https://github.com/CLIPOS
- Bugs: https://github.com/CLIPOS/bugs
- Code review: https://review.clip-os.org

- Built to be reusable for other use cases
- Fell free to come and talk to us at the ANSSI stand!

Thanks!

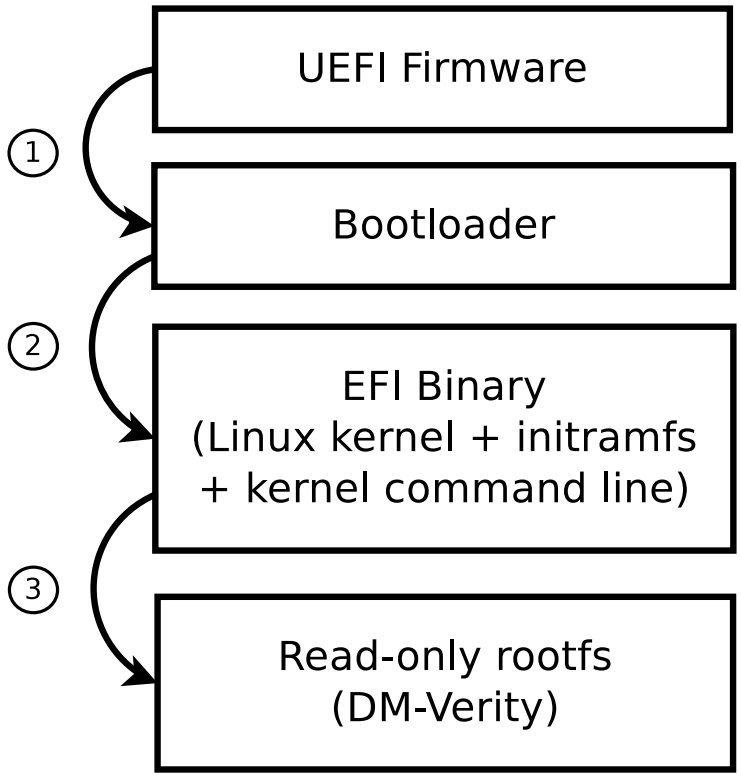
✉ clipos@ssi.gouv.fr

- Website: clip-os.org
- Docs: docs.clip-os.org
- Sources: github.com/CLIPOS
- Bugs: github.com/CLIPOS/bugs

Full boot chain integrity guarantee

Guarantee full system integrity even in the event of a system compromise

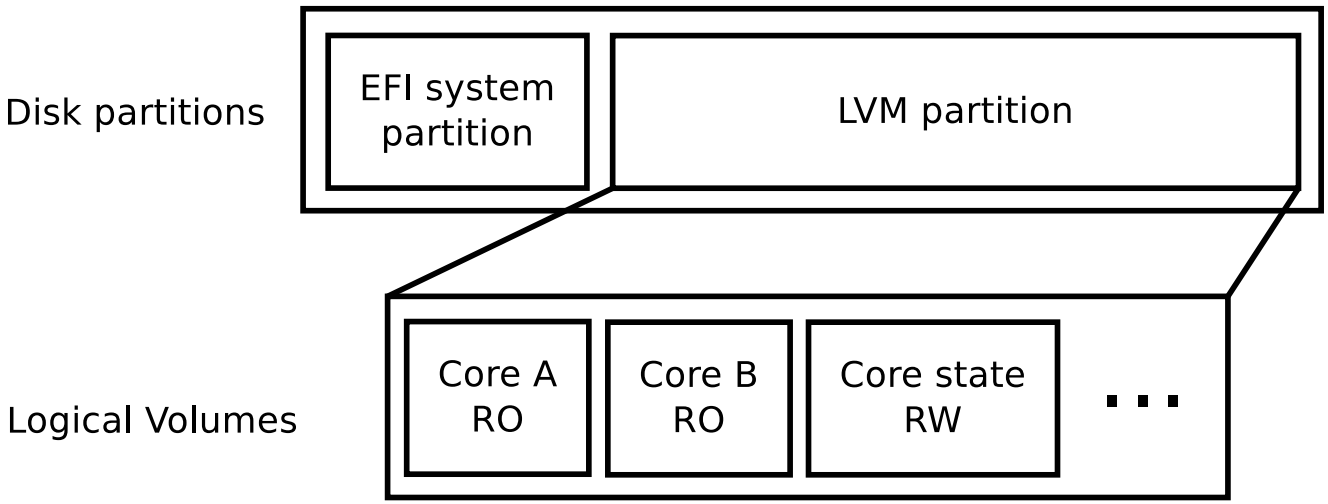
- Will only boot if the system's integrity can be cryptographically verified
- Based on UEFI Secure Boot feature:
  - Signed bootloader, initramfs, Linux kernel and its command line
  - Read-only system partition (Squashfs) protected by DM-Verity (with forward error correction)
  - Custom keys (i.e. not signed by Microsoft, requires enrollment in hardware)



No arbitrary code execution: W ⊕ X

Defense in depth and difficulty for an attacker to persist post compromission

- Strict split between:
  - Read Only: system executables, configuration and data (DM-Verity)
  - Read Write: runtime configuration, logs, user and application data (DM-Crypt+DM-Integrity)

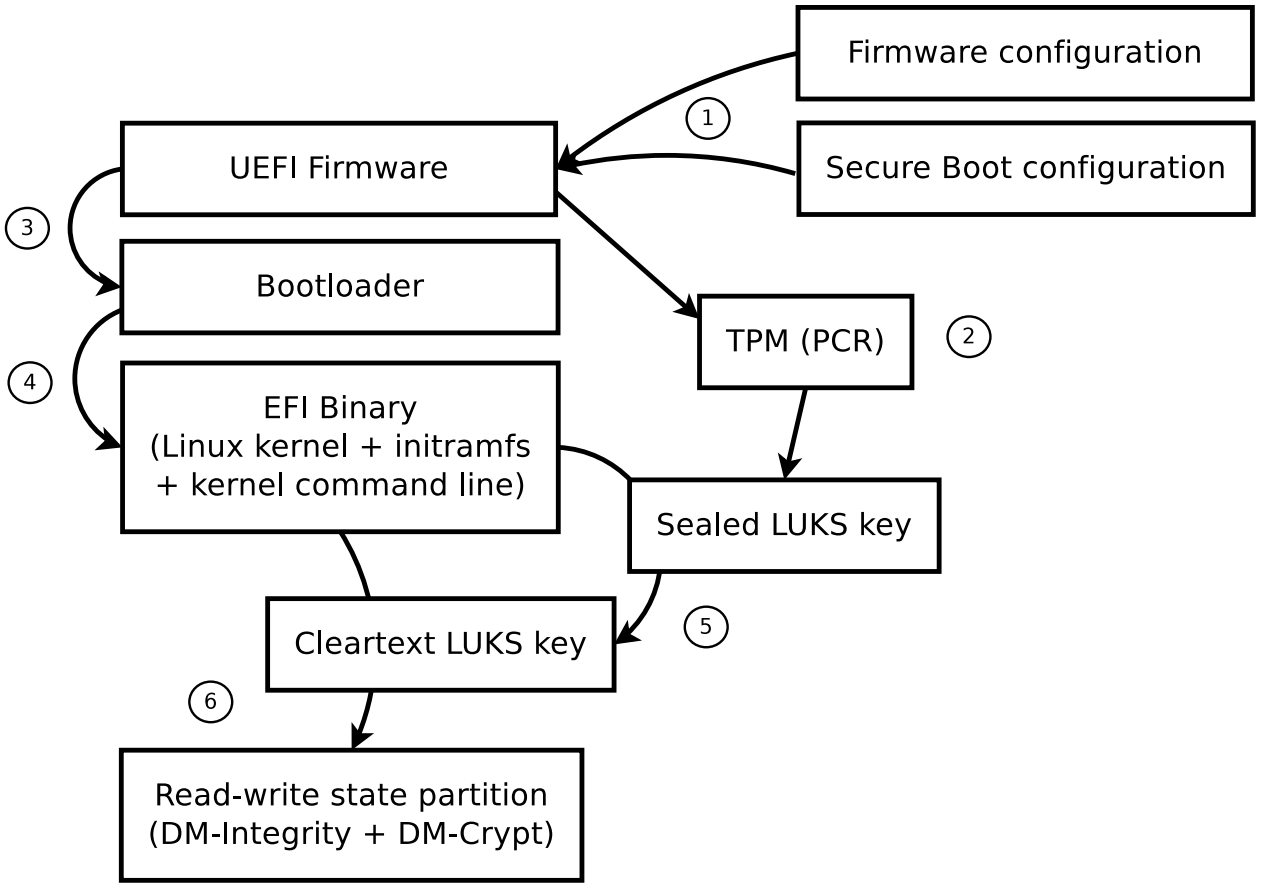


- Interpreter support (Bash, Python, etc.) currently in progress upstream[3]

[3] See the talk at Kernel Recipes 2018, Paris (https://clip-os.org/en/talks)

Password-less encrypted partitions

- Automatic secret sealing & unsealing with a TPM 2.0
- Based on boot chain integrity measurements



Hardened Linux kernel and curated hardware profiles

Hardened Linux kernel

- Based on latest upstream stable kernel
- Includes hardening patches: lockdown, linux-hardened, stackleak
- Security focused build configuration (KCONFIG)
- Security focused runtime configuration (sysctl)

Curated hardware profiles

- Per hardware profile selection of firmware and kernel modules
- Currently available hardware profiles (easily extended):
  - QEMU/KVM virtual machine
  - Lenovo X260

Roadmap: Beta

Completed

- "Unprivileged" admin, audit and update roles
- SSH server (for audit, admin and debug)

In progress

- Client for automatic updates:
  - Unattended, in background, updates (i.e. effective on reboot)
  - User controlled rollback at boot time
- Confined IPsec client
- Basic network (DHCP, static IP) and firewall (static rules) support

Roadmap: 5.0 stable

- Confined user environments (GUI)
- Multilevel support (Vserver-like LSM)
- Automated installation using PXE
- etc.