Improving the Security of Edge Computing Services

Update status of the support for AMD and Intel processors

FOSDEM 2020

Piotr Król
Who am I

Piotr Król
Founder & Embedded Systems Consultant

- open-source firmware
- platform security
- trusted computing

@pietrushnic
piotr.krol@3mdeb.com
linkedin.com/in/krolpiotr
facebook.com/piotr.krol.756859
- **S-RTM** doesn't address all issues
  - requires platform reset to establish trusted state
    - we cannot assume that everyone will reboot machine each time they want to do something requiring known security state
- PCRs update when using LUKS+TPM
  - PCRs value prediction was already solved
  - but we have to re-provision those values to TPM (or any other tamper-proof storage of measurements)
  - when and how to do that securely?
- vendor specific (NXP HAB, Intel Secure Boot/Boot Guard, AMD HVB) requires proprietary tools and NDAs
  - we tried many of them and have to say those tools are terrible
  - well-established in IBV environment
- hard to reestablish trust and/or re-own platform
  - platforms fused with vendor specific **S-RTM** cannot be re-owned
How D-RTM improve Edge Security

- reestablishing root of trust dynamically and on-demand e.g. before performing critical infrastructure management operations
- very useful in cloud environment when server uptime is critical
- important in bare-metal cloud environment, where trust in hardware have to be verified when switching between customers
- remote attestation
- secure firmware re-flashing to mitigate software supply chain risk
Open-source ecosystem-wide framework for launch integrity
  - Its goal is to make D-RTM first class citizen across the open-source projects
  - D-RTM should work out-of-the-box on all Linux distros

**NOTE**: Following slides discuss AMD TrenchBoot status, previous status was presented at **OSFC2019** and **PSEC2019**

OSFC2019: [https://osfc.io/talks/trenchboot-open-drtm-implementation-for-amd-platforms](https://osfc.io/talks/trenchboot-open-drtm-implementation-for-amd-platforms)
What components are involved:

- **hardware**: SKINIT-capable AMD, TXT-capable Intel (not on diagram)
- **firmware**: coreboot or proprietary UEFI-based implementation
- **bootloader**: GRUB2
- **DRTM Configuration Environment (DCE)**: Landing Zone (LZ)
- **DLME**: Linux kernel and u-root
Summary

- TB is tested on PC Engines (AMD GX-412TC, 4-core, 1GHz) network appliance
- There are no Intel TXT capable platform at ~100EUR price point
- Closest equivalent are Intel Broadwell based platforms, but do not support DRTM at that price point
- Intel TXT requires ACM BIOS and ACM INIT - those components are very unlikely to be open-source and its redistribution at this point is prohibited

Our plans

- IOMMU improvements (more info on LandingZone slide)
- so far we built SPI content in coreboot
  (coreboot+GRUB2+LZ+Linux+u-root), we want to move to Yocto meta-trenchboot
- try new AMD platforms and check if new PSP or UEFI firmware makes difference

Summary
- Diffstat with upstream: 8 files changed, 1713 insertions(+), 4 deletions(-)
- Further development and upstreaming on AMD ground is blocked due to lack of Intel TXT RFC

New things
- Relocator patch was sent but no feedback
- zeropage (Linux kernel struct boot_params) address corrected - previously we used address obtained before relocation and we were lucky that nothing overwrote that
- We changed location of lz_header, thanks to that SKINIT does not measure pointer to zeropage and SHA not depend on state before SKINIT

Our plans
- merge everything upstream to GRUB2 project

GRUB2 source code for AMD: https://github.com/3mdeb/grub2/tree/move_header
• Summary
  ○ Code size (SLOC): C: 2316, sh: 373, asm: 153
  ○ Diffstat with upstream: 12 files changed, 552 insertions(+), 30 deletions(-)

• New things
  ○ DEV DMA protection mechanism (bit array with access rights to 4k blocks) is removed on newer platforms
  ○ correct Linux kernel size reading
  ○ support for SHA256 measurements
  ○ support for 32bit builds - size dropped to 12kB (from ~60kB)
  ○ lots of bugfixes and optimizations (Kudos to Andrew Cooper and the Xen Project)
Our plans

- DMA protection: most probably IOMMU configuration for LZ
- we also consider unset bus mastering bit based on recent Matthew Garret blog post and kernel work
- sent patches and merge everything upstream to TrenchBoot project

Matthew Garret blog post: [https://mjg59.dreamwidth.org/54433.html](https://mjg59.dreamwidth.org/54433.html)
LZ source code for AMD: [https://github.com/3mdeb/landing-zone/tree/fosdem2020](https://github.com/3mdeb/landing-zone/tree/fosdem2020)
• Summary
  ◦ At PSEC 2019 we show demo where we successfully kexec'ed Xen and booted pfSense-based virtual firewall
  ◦ More information about patches in Intel part of this presentation
• Our plans
  ◦ reproducible builds Linux kernel and u-root
3mdeb for over 4 years is PC Engines apu-series maintainer in coreboot
Open-source software and hardware development: customized, application specific, edge computing appliances with long term support
BIOS/UEFI/open-source firmware extensions, Embedded Linux (like Yocto/OpenEmbedded) or hypervisors (Xen, Bareflank)
We produce Open Hardware: TPM modules, OpenViszla, MuxPi, RTE
If you looking for support feel free to contact us: contact@3mdeb.com
Q&A