An Open Source Approach to System Security

Helena Handschuh,
RISCV Security Standing Committee Chair

RISCV Workshop, @ Zurich 06/12/2019

Rambus
Data · Faster · Safer

Why do we need an open source approach?

- sharing specifications with peers can advance development faster
- testing is more effective ;
- Compliance is critical to build up an ecosystem
- Interoperability
- No security by obscurity
- Open formal models allow to test for security issues
- Formalizing the security test tools and development tools
- Make tools available to entire community
- Reason on security models and functionality

Embedded System Security

→ Ways that the RISC-V Foundation and its Security Committees can help designers of secure cores.

- Let's start with a clean slate: RISCV open specifications

- Secure Processor Ingredients…
  - RISCV base Instruction Set Architecture (ratified)
  - Privilege specification defining privilege modes (Machine, Supervisor, Hypervisor, User)
  - Security Extensions
  - Crypto extensions (Richard Newell, Microchip) and a Trusted Execution Environment TG (Joe Xie, Nvidia)

- Allow to define Secure Processing:
  - Secure boot process
  - Cryptographic algorithms and keys
  - Trusted Execution Environment
  - Secure applications
  - Secure User API

RISCV Foundation Task Groups relating to Security

Crypto extensions Task Group

- Approach based on vector extensions
- AES instructions (1 round, full round)
- SHA-2 instructions (1 round, full round)
- Prototyping Public Key Crypto algorithms
  - Long integer arithmetic
  - Implementation proof of concept

TEE Task Group

- Secure boot specifications proposal
- PMP Physical Memory Protection proposal
- IO PMP proposal
- …

Security measures can benefit from an open approach

- Newer cache timing side-channels result from micro-architectural design flaws
- Cannot be directly fixed/eliminated at the ISA level but:
- Generic solutions can be added into RISCV based platform specifications
  - All internal resources should be either flushable or separable (AISA)
  - For example platform specific Flush instructions can be added
  - Access permissions can be defined to separate cache space, memory spaces

  - Security Standing committee discussing how to add to *platform specs*
    - Galois taking the lead: demonstrating first platform specs for a secure voting machine @ Defcon
    - Gernot Heiser Data61 closely following with platform specs based on SeL4

Taxonomy and Formal reasoning (Galois, SSITH projects)

- “Lando” : a formal specification for HW design with 4 sublanguages:
  - A system spec language
  - Architecture language
  - Product line engineering language
  - Security property specification language
- A domain model for specifying security properties.
  - formalization of the NIST CWEs related to buffer/memory errors
- BESSPIN: a tool suite for formal reasoning
  - GRIFT: subsystem of tool suite contributed to RISCV Formal Specs TG
- Platform specs and security-enriched ISA:
  - Secure voting machine platform spec includes security properties/guarantees
  - 6 other platform specs based on RISCV SoCs (Rocket, Boom, Piccolo, Flute, Bassoon, Riscy)

Still lots of work to BE doNE...

Come join our RISCV Foundation Security Committees

Thank you!