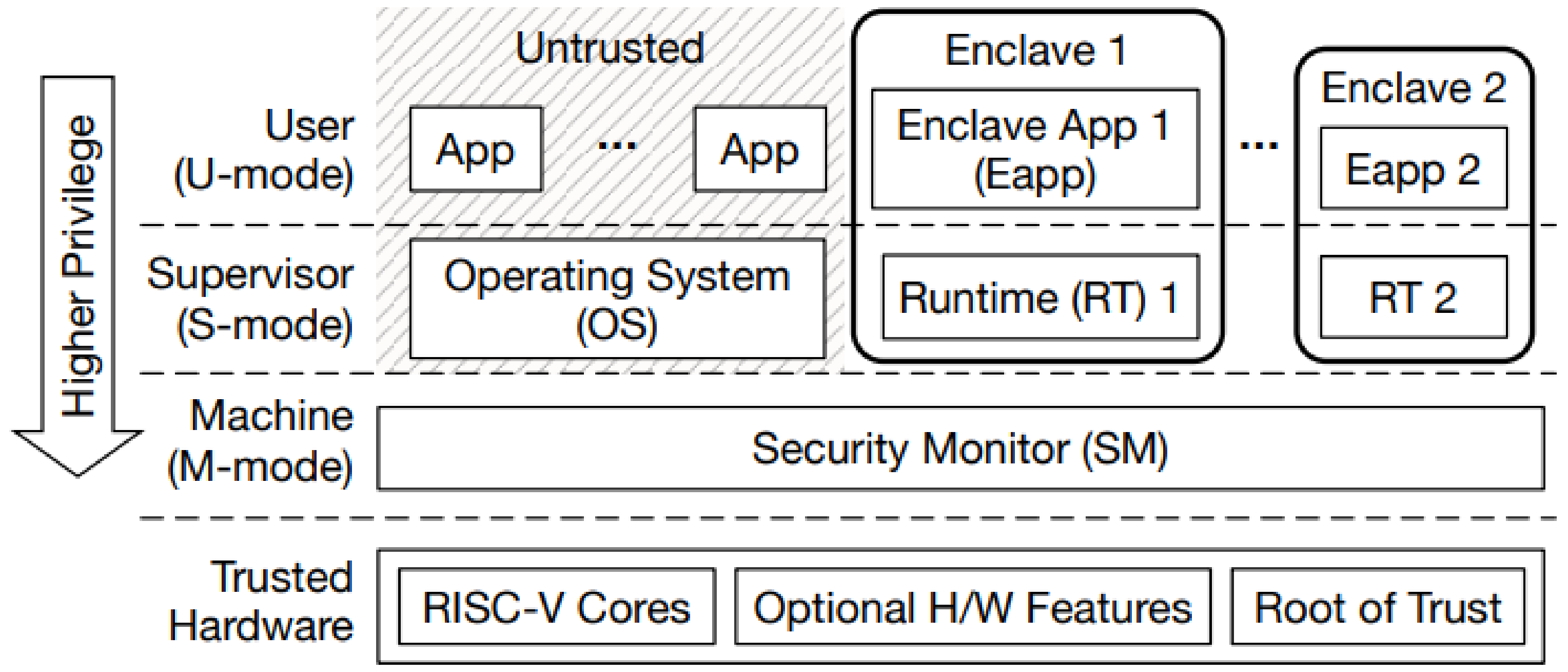
Mitigating I/O Vulnerabilities In Trusted Execution Environments

Pratham Hegde, Byeonggil Jun, Prof. Hokeun Kim

Introduction

Trusted Execution Environments (TEEs) protect sensitive code and data from untrusted systems. While existing TEEs like Intel SGX are closed and rigid, **Keystone** offers a flexible, open-source TEE for **RISC-V**, enabling customizable enclave designs. However, I/O pathways remain a potential attack surface. This project investigates Keystone's I/O vulnerabilities and explores strategies for securing enclave interactions.

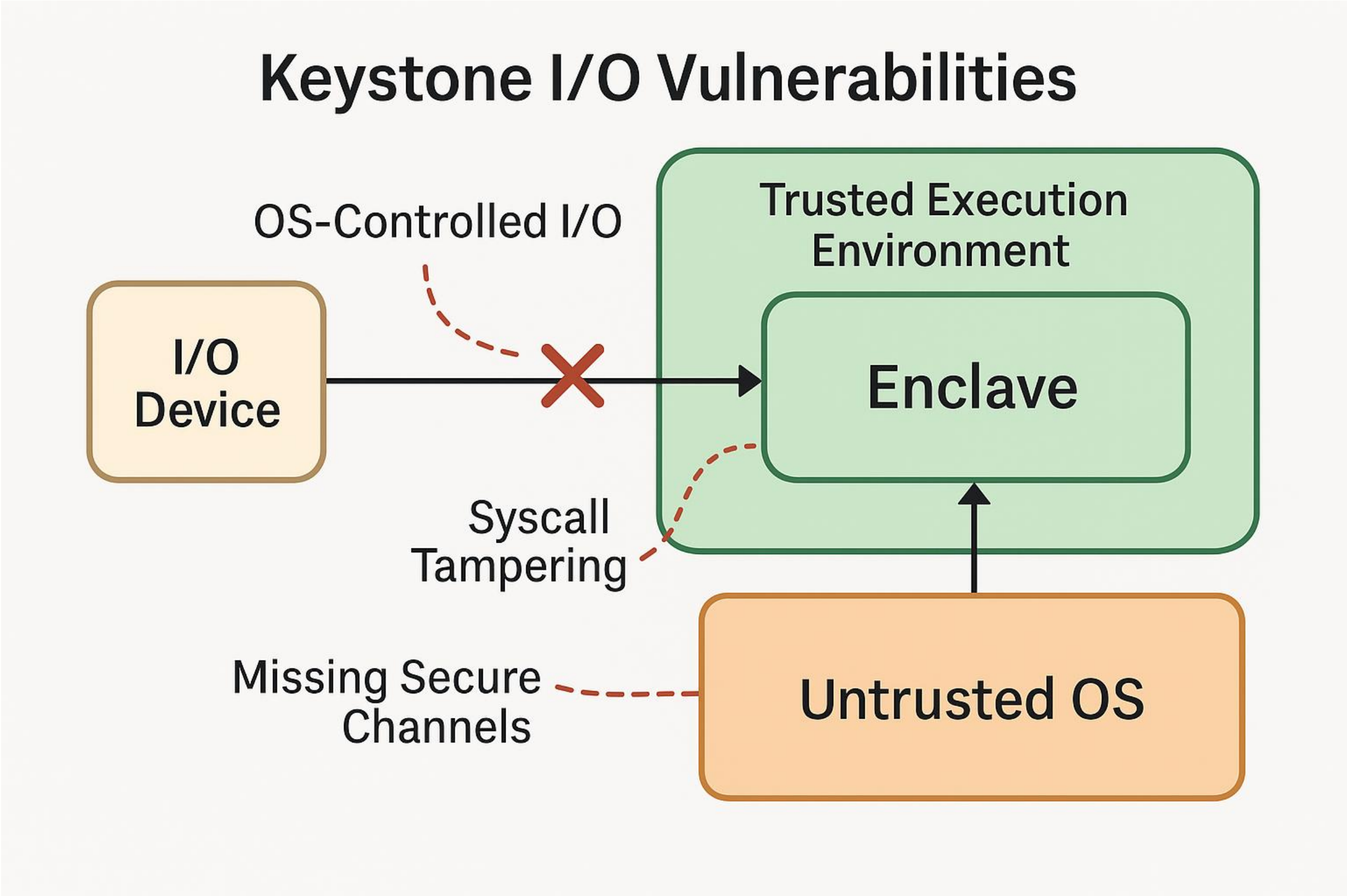


Acknowledgements

- D. Lee, M. Yun, R. Xu, S. Arora, I. Lebedev, R. Xu, and S. Devadas, "Keystone: An Open Framework for Architecting Trusted Execution Environments," https://doi.org/10.48550/arXiv.1907.10119
- QEMU - https://www.qemu.org/

Research Methods

- Deploying **Keystone** with **FireMarshal** and **Chipyard** on **AWS EC2** using RISC-V toolchain and FireSim.
- Building and testing Keystone with **QEMU (Quick EMUlator)** using Buildroot and custom defconfigs.
- Conducted literature review on **TEE models**, Keystone internals, and enclave benchmarks.
- Plan to explore **enclave memory** and **I/O isolation** through **PMP** and **edge call mechanisms**.
- Plan to analyze **attestation flow**, secure boot procedures, and syscall security within the security monitor.


Keystone I/O Vulnerabilities

Progress/Achievements So Far

- •Built toolchains and dependencies using Buildroot.
- •Successfully compiled host-binutils and toolchain after cleaning stale builds.
- •Patched CMakeLists to support keystone-sdk configuration.
- •Managed PATH, environment, and overlay issues during image build.

Existing Vulnerabilities

- **OS-Controlled I/O** – Data paths to I/O devices remain exposed to OS tampering and snooping.
- **Syscall Tampering** – I/O syscalls can be intercepted or altered by a malicious OS.
- **Missing Secure Channels** – TEEs lack built-in secure I/O, relying on vulnerable custom implementations.

Future Plans

- Fix g++ linker to finalize SDK build, validate enclaves on QEMU, and benchmark performance/security.
- Integrate FireSim for profiling and analyze I/O vulnerabilities with architectural fixes.
- Resolving QEMU and cross-compilation issues (GLIBC, pkg-config) due to outdated model.