Citadel: A Trusted Reference Monitor for Linux using Intel SGX Enclaves

A.H. Bell-Thomas

Computer Laboratory, University of Cambridge

26th June, 2020

Background

Background

1. **Reference Monitor**

Background

1. **Reference Monitor**
   ⇝ *Information Flow Control*

Background

1. **Reference Monitor**
   $\rightsquigarrow$ *Information Flow Control*

2. **Intel SGX**

Information Flow Control

- Access Control specifics *who* can access resources. IFC also mediates *how* they can be used once opened.

- Construct an abstract system of *entities*;
  $\rightsquigarrow$ processes, files, sockets, etc.

- Each *entity* carries a *security context*, defining its granular ownership or restriction information.

- Aim: achieve *non-interference* between all *security contexts.*

- Decentralised IFC — let entities specify their own, *discretionary*, protection policy for assets they own. More flexible, and supports operations such as *declassification.*

Information Flow Control

Enforcement is implemented using a *reference monitor*, which provides;

- **Tagging**
  Entities must be uniquely and reliably identifiable to support decisions.

- **Tracking**
  Contexts are mutable to accommodate an evolving situation.

- **Policy Decisions**
  Is an operation acceptable given its consequences?
  c.f. Biba, Bell-LaPadula

Linux Security Modules

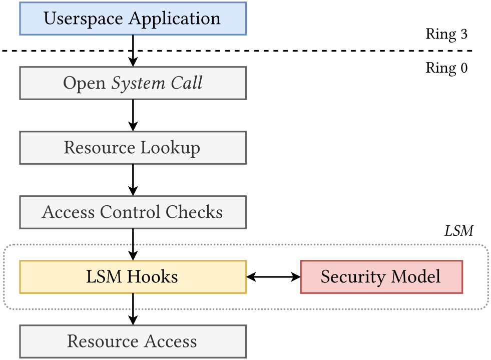


Figure: Core workflow of an LSM.

Intel SGX

A general-purpose *trusted execution environment* provided via x86 at the architectural level in modern processors.

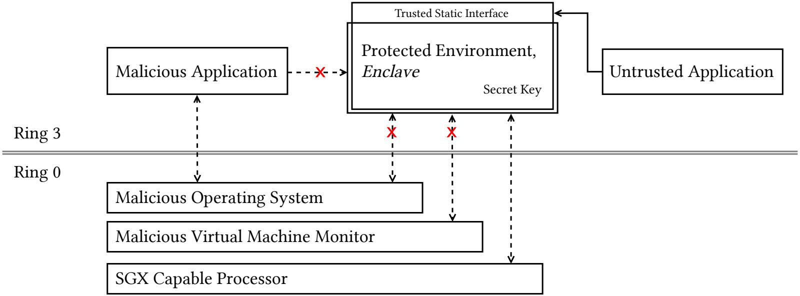


Figure: Abstract overview of SGX's protections.

Intel SGX

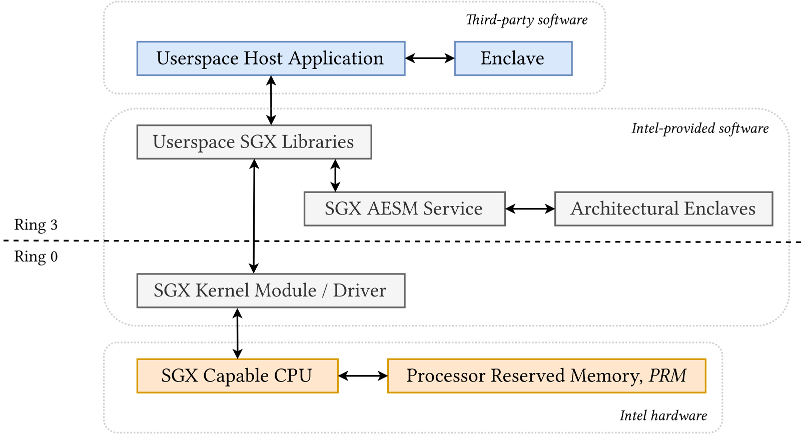


Figure: Components of the SGX platform.

Citadel

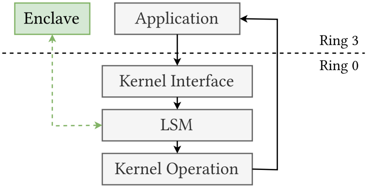
A prototype implementation of an SGX-protected reference monitor for Linux.

Reference monitors must be;

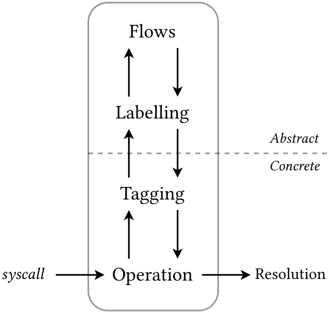
- Always invoked.
- Evaluable.
- Tamper proof.

— *in theory, a perfect use case for SGX.*

Architecture?

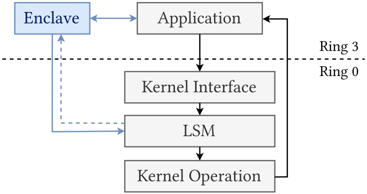


(a) Naive enclave integration.

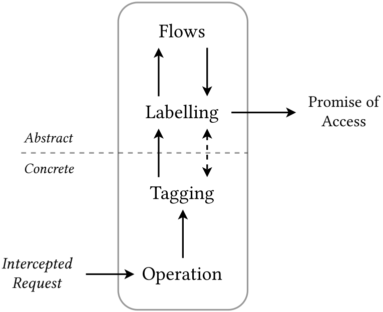


(b) Traditional reference monitor decision flow.

Architecture

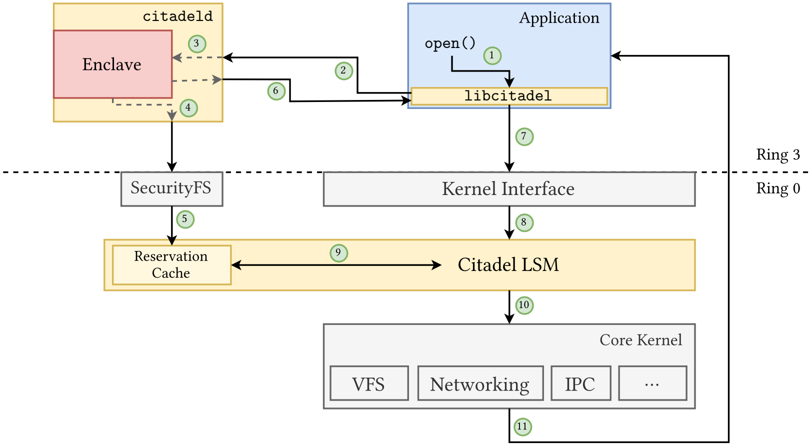


(a) High level Citadel dataflow. Backflow from the LSM to the enclave is asynchoronus.



(b) Citadel IFC decision flow. Decision provides a *promise* of access; permission propagates asynchronously to the LSM.

Architecture



Results

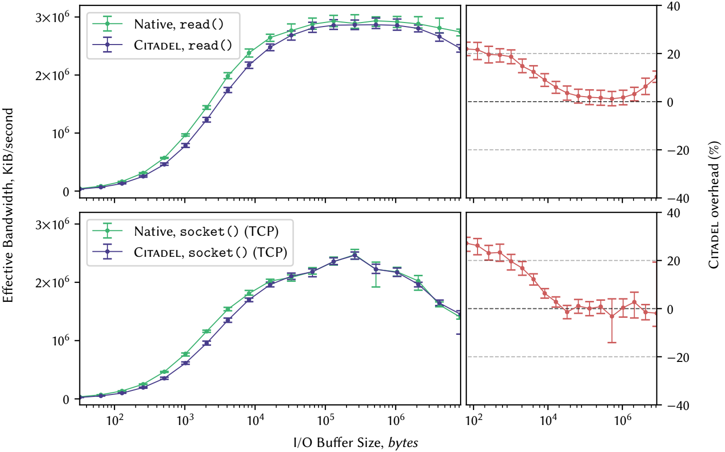


Figure: Effective operation bandwidths between two processes.

Results

- Median *syscall* overhead of $43\mu s$ $(1 - 2\mu s$ amoritsed).
- $20 - 25\%$ effective throughput decrease for IPC.
- Real-world benchmarks using NGINX;
  - Low latency trials: 24% median overhead.
  - High bandwidth file transfers: $\sim 0\%$ median overhead.
- Security characteristics — *promising*.

Conclusion

- Citadel — a modular, enclave-backed reference monitor to securely and verifiably implement IFC methods in the Linux kernel.
- Implemented using enclaves, an LSM, and an auxiliary library for unobstrusive application integration.
- Real-world performance overhead of $20 - 25\%$ observed using Nginx and microbenchmarks.
- Demonstrates potential viability of a symbiotic enclave-kernel relationship for security implementations.

References