Running sensitive workloads on untrusted hosts: Project Enarx

Mike Bursell
Office of the CTO
Red Hat
The Problem
Who do you trust?
Who do you trust?

- Your Cloud Service Provider?
- Your service provider’s sysadmins?
- Your internal servers?
- Your sysadmins?
What do you trust?

- Your Cloud Service Provider?
- Your service provider’s sysadmins?
- Your internal servers?
- Your sysadmins?

- The Operating System?
- The entire software stack?
  - At provisioning
  - After it’s been sitting there for months
- All the other workloads?
- All the firmware?
- All the hardware?
The Need for Confidentiality and Integrity

- Banking & Finance
- Government & Public Sector
- Telco
- IoT
- HIPAA
- GDPR
- Sensitive enterprise functions
- Defense
- Human Rights NGOs
- ...
Virtualization Stack

- Application
- Middleware
- Userspace
- Kernel
- Bootloader
- Hypervisor
- Firmware
- BIOS | EFI
- CPU | Management Engine
THE MODERN TECH STACK

- Compromised by a customer
- Compromised by a former employee
- Compromised by a current employee
- Compromised by bitcoin miners
- Compromised by unknown hackers
- Compromised by our own government
- Compromised by a foreign government
- Massive undiscovered hardware vulnerability

https://xkcd.com/2166/
The Plan

Application

Middleware (inc. Enarx)

Userspace

Kernel

Linux

Hypervisor

BIOS | EFI

CPU | Firmware
Don’t trust the host
Don’t trust the host owner
Don’t trust the host operator
All hardware cryptographically verified
All software audited and cryptographically verified
The Fit

- Well suited to microservices
- Well suited to sensitive data or algorithms
- Easy development integration
- Simple deployment
- Standards based: WebAssembly (WASM)

Don’t trust the **host**
Don’t trust the host **owner**
Don’t trust the host **operator**
All **hardware** cryptographically verified
All **software** audited and cryptographically verified
Trusted Execution Environments
What’s a TEE?

- Application
- Middleware
- Userspace
- Kernel
- Bootloader
- Hypervisor
- Firmware
- BIOS | EFI
- CPU | Management Engine
What's a TEE?

Only the CPU has access
What's a TEE?

What happens when other layers try to access?

Only the CPU has access
What’s a TEE?

What happens when other layers try to access? *Blocked by CPU.*

Only the CPU has access

- Application
- Middleware
- Userspace
- Kernel
- Bootloader
- Hypervisor
- Firmware
- BIOS | EFI
- Management Engine
Trusted Execution Environments

TEE is a protected area within the host, for execution of sensitive workloads.
Trusted Execution Environments

TEE is a protected area within the host, for execution of sensitive workloads

TEE provides:
- Memory Confidentiality
- Integrity Protection
- General compute
- HWRNG
TEE provides:
- Memory Confidentiality
- Integrity Protection
- General compute
- HWRNG

Q. “But how do I know that it’s a valid TEE?”
Q. “But how do I know that it’s a valid TEE?”
A. Attestation

TEE provides:
- Memory Confidentiality
- Integrity Protection
- General compute
- HWRNG
TEE provides:
- Memory Confidentiality
- Integrity Protection
- General compute
- HWRNG

Attestation includes:
- Diffie-Hellman Public Key
- Hardware Root of Trust
- TEE Measurement
TEE provides:
- Memory Confidentiality
- Integrity Protection
- General compute
- HWRNG

Attestation includes:
- Diffie-Hellman Public Key
- Hardware Root of Trust
- TEE Measurement

Trusted Execution Summary

Attestation
On which technology do I build my application?
Introducing Enarx
Enarx Principles

1. We don’t trust the host owner
2. We don’t trust the host software
3. We don’t trust the host users
4. We don’t trust the host hardware
   a. ... but we’ll make an exception for CPU + firmware
Enarx Design Principles

1. Minimal Trusted Computing Base
2. Minimum trust relationships
3. Deployment-time portability
4. Network stack outside TCB
5. Security at rest, in transit and in use
6. Auditability
7. Open source
8. Open standards
9. Memory safety
10. No backdoors
The Enarx 5-bullet overview
The Enarx 5-bullet overview

Uses TEEs (SGX, SEV, etc.) for confidential workloads
The Enarx 5-bullet overview

- **Uses TEEs** (SGX, SEV, etc.) for confidential workloads
- **Easy** development and deployment
The Enarx 5-bullet overview

- **Uses TEEs** (SGX, SEV, etc.) for confidential workloads
- **Easy** development and deployment
- **Strong security** design principles
The Enarx 5-bullet overview

Uses TEEs (SGX, SEV, etc.) for confidential workloads

Easy development and deployment

Strong security design principles

Cloud-native → Openshift, kubernetes
The Enarx 5-bullet overview

- Uses TEEs (SGX, SEV, etc.) for confidential workloads
- Easy development and deployment
- Strong security design principles
- Cloud-native → Openshift, Kubernetes
- Open source: project, not production-ready (yet)
The Enarx 5-bullet overview

- Uses TEEs (SGX, SEV, etc.) for confidential workloads
- Easy development and deployment
- Strong security design principles
- Cloud-native → Openshift, kubernetes
- Open source: project, not production-ready (yet)
Enarx Architecture

Application

Language Bindings (libc, etc.)

WASI

WebAssembly

Process-Based Keep

SGX
Sanctum

VM-Based Keep

SEV
PEF
MKTME

W3C standards
Keep - process or VM-based

- Core Keep
- Platform-specific
  - Hardware (CPU): silicon vendor
  - Firmware: silicon vendor
  - Software: Enarx

Architecture varies between VM/Process-based platforms
- WebAssembly (WASM)
  - W3C standard
  - Stack Machine ISA
  - Sandboxed
  - Supported by all browsers
  - Exploding in the “serverless” space
  - Implementations improving rapidly
    - cranelift and wasmtime
WebAssembly System API (WASI)

- W3C Standards Track
- Heavily inspired by a subset of POSIX
- Primary goals:
  - Portability
  - Security
- libc implementation on top
- Capability-based security:
  - No absolute resources
  - Think: openat() but not open()
Language Bindings (libc, etc.)

Compilation targets and includes, e.g.

- Rust: --target wasm32-wasi
Application

- Written by
  - Tenant (own development)
  - 3rd party vendor
- Standard development tools
- Compiled to WebAssembly
- Using WASI interface
Enarx is a Development Deployment Framework

Choose Your Language / Tools → Develop Application → Compile to WebAssembly

Choose Host → Instance Configuration
Enarx is a Development Deployment Framework

(Example components)

Choose Your Language / Tools → Develop Application → Compile to WebAssembly

Choose Host

Openshift/Kubernetes

Instance Configuration

Dev tooling

IBM Cloud, Azure, AWS, ...
Process flow
Enarx architectural components

Host

Application
Enarx runtime
Enarx host agent
CPU + firmware

Keep

Client

Enarx client agent

CLI
Orchestrator (e.g. Openshift/k8s, Openstack)

Code + Data (Encrypted)

Client/host agent comms

Attestation

1, 5

3, 7

2, 4

6
Enarx attestation process diagram

1. Request workload placement
2. Request Keep
3. Create Keep, load Enarx runtime
4. Measurement of Keep + Enarx runtime
5. OK/not-OK
6. Code + Data (encrypted)
7. Load Code + Data into Keep
What happens?

“Server”

Host

AMD firmware

Secure VM

“Client”

Tenant
What happens?

“Server”

Host

Secure VM

“Client”

Tenant

Attestation handshake

AMD firmware
What happens?

"Server"

Host

AMD firmware

Secure VM

"Client"

Tenant

Attestation handshake

Code + data delivery (encrypted)
What happens?

"Server"

Host

- AMD firmware

Secure VM

- Code runs

"Client"

Tenant

Attestation handshake

Code + data delivery (encrypted)
What happens? (SEV)

“Server”

Host

- AMD firmware

Secure VM

- Code runs

Enarx Keep
What happens? (SGX)

Host

Secure Enclave

Intel firmware

Attestation handshake

Code+ data delivery (encrypted)

“Server”

Tenant

“Client”

Code runs
What is Enarx?

Enarx Keep

TEE instance

Code runs

H/w vendor firmware

Host

“Server”
We Need Your Help!

Website: https://enarx.io

Code: https://github.com/enarx

Gitter: https://gitter.im/enarx/

License: Apache 2.0

Language: Rust
What should I do now?

When I leave this room
● Check out https://enarx.io/

When I get back to the office
● Evaluate our existing deployments of sensitive workloads

When my boss asks me what I learnt from the RSAC webinar
● Disclose existing weaknesses, explain the opportunities
● Ask them to let me get involved in the Enarx project

When the CTO asks my boss what I learnt from the RSAC webinar
● Ask them about joining the Confidential Computing Consortium
Questions?

https://enarx.io