Jalapeno

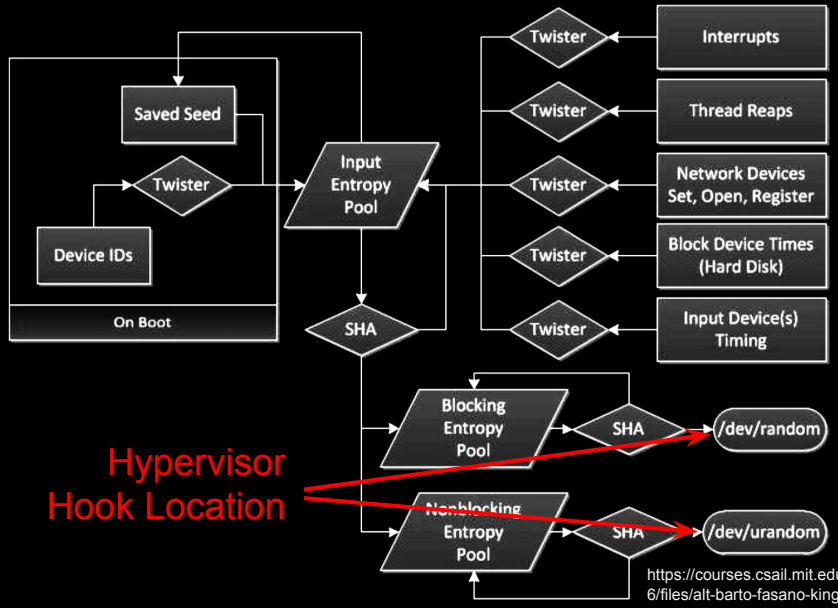
(joll-ah-pee-no)

Jeremy Erickson and
Timothy Trippel

Motivation

- We created a malicious *hypervisor* that attacks VM RNGs
- Works against both Linux kernel and Apache2 -- nearly invisible
- Imagine an AWS scenario -- Since keys can be remotely predicted, no need to exfiltrate
- Any cyber superpowers that may have the capabilities to do this in the real world?

Linux Kernel RNG



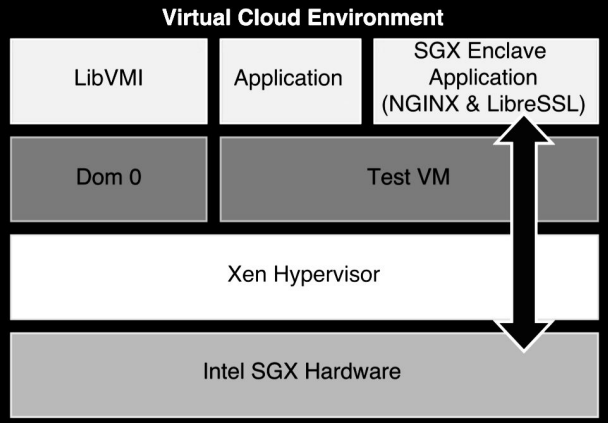
https://courses.csail.mit.edu/6.857/2016/files/alt-barto-fasano-king.pdf

Real random bytes

```
# dd if=/dev/urandom count=1 bs=10 2>/dev/null | xxd
 362b 3f69 cdb8 fce9 64f1                  6+?i....d.
# dd if=/dev/urandom count=1 bs=10 2>/dev/null | xxd
 6666 6666 6666 6666 6666                  ffffffffff
```

Not-so-random bytes...

Approach Overview



- Intel Software Guard Extensions (SGX)
- Perform RNG and private key storage in hardware-supported secure enclave
  - Even hypervisor cannot inspect/modify!
- Expose APIs to application to perform privileged operations (sign, decrypt, etc.) without accessing key material
  - Private keys guaranteed to never leave the enclave
    - *(we have ideas for how to replicate them across machines without leaking them)

We're building an open-source SGX enclave crypto library -- with a focus on remote attestation that keys are *unleakable* and *unpredictable*.

Current Status

- ☑ Create working SGX Enclave
- ☑ Generation of asymmetric key pairs
- ☑ Sealing of key pairs to disk for secure persistent storage + recovery on failure
- ☑ Implementation of core asymmetric crypto functions (sign, decrypt)
- ☐ Generation of symmetric key material (with PFS, derived from multiple parties)
- ☐ Implementation of core symmetric crypto functions (encrypt, decrypt)
- ☐ Convert existing “untrusted” application into library
- ☐ Build test application that measures performance overhead

End Goal

For this class:

- Working crypto library to be used by applications in untrusted environments
- Test application that compares performance of crypto operations in and out of enclave
- Performance evaluation detailing overhead incurred by using SGX enclaves for crypto operations

Ultimately:

- Secure Webserver: full integration with LibreSSL / NGINX
  - Will require substantial rewriting of how NGINX performs its TLS operations
  - Fully backwards-compatible with existing TLS client implementations, just adds new security guarantees
- Generic SGX Crypto API for integration into future applications