OPEN POSSIBILITIES.

Securing Private Keys in Edge Datacenters

Notices & Disclaimers
Intel technologies may require enabled hardware, software or service activation. No product or component can be absolutely secure. Your costs and results may vary.

OCP
GLOBAL
SUMMIT

NOVEMBER 9-10, 2021

SECURITY

Securing Private Keys in Edge Datacenters

Manish Dave, Principal Engineer, Intel

OPEN PLATINUM™

OPEN POSSIBILITIES.

OCP GLOBAL SUMMIT
NOVEMBER 9-10, 2021

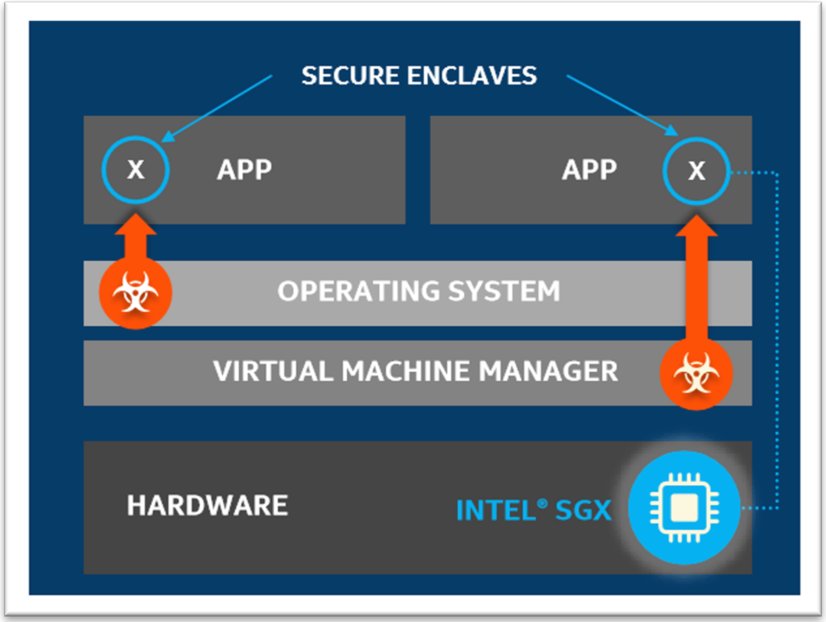
Problem Statement

SECURITY

- **Software based private key protection is not sufficient**
  - Platforms in Edge datacenters have larger attack surface
  - Hardware Security Modules (HSM) are cost prohibitive, do not scale easily, not suitable for every Edge Compute node
  - TPM based implementations may not provide full (in memory runtime) protections and have deployment limitations on Edge, Virtualization and multi-tenancy

SGX Hardware-based Trusted Execution Environment

SECURITY



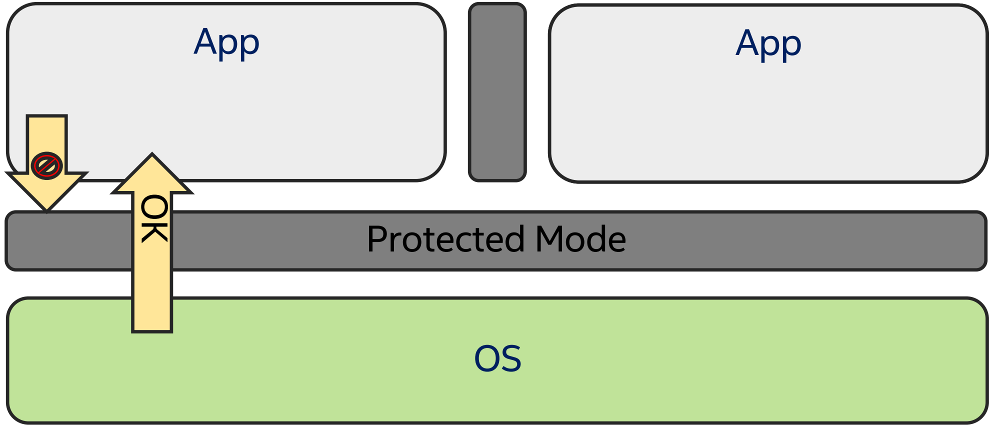
- Intel SGX removes the privileged software (OS, VMM, SMM, devices) and unprivileged software (Ring 3 applications, VMs, containers) from the trust boundary
- Encrypts memory to help protect against memory bus snooping and cold boot attacks for enclave code and data in host DRAM
- Provides Hardware Based Remote Attestation

Why Aren't Platforms Trustworthy?

Build 3

SECURITY

Protected Mode (rings) protects OS from apps ...

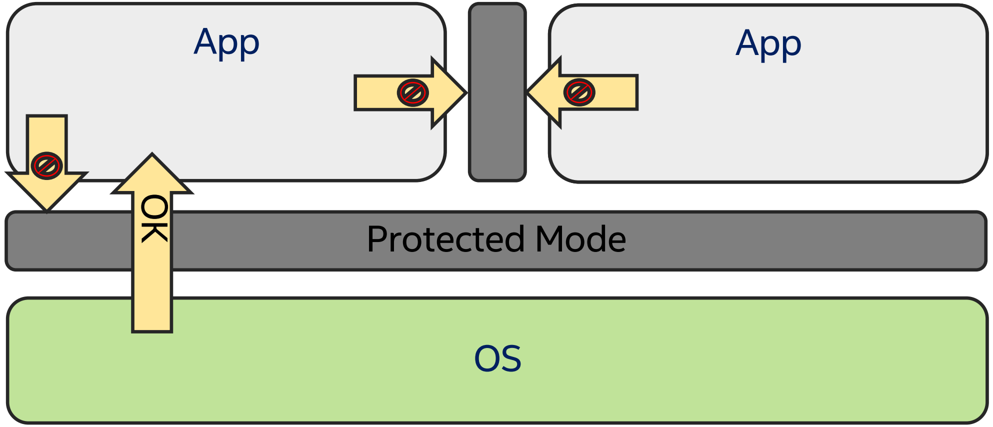


Why Aren't Platforms Trustworthy?

Build 3

SECURITY

Protected Mode (rings) protects OS from apps …



… and apps from each other

…

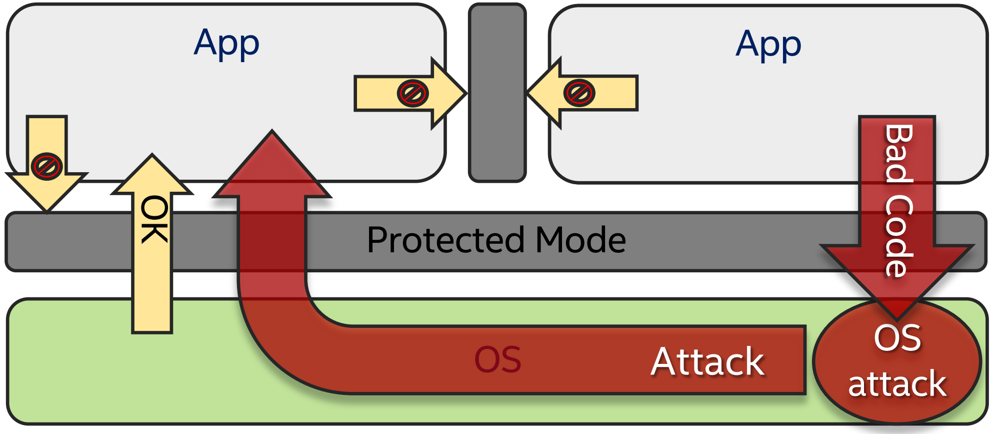
Why Aren't Platforms Trustworthy?

Build

3

SECURITY

Protected Mode (rings) protects OS from apps ...



... and apps from each other

...

... UNTIL a malicious app exploits an OS flaw to gain full privileges and then tampers with the OS or other apps

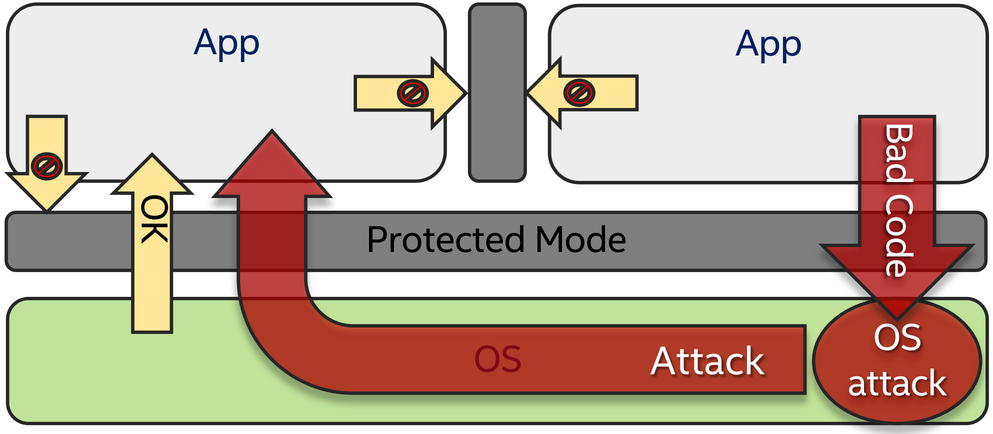
Why Aren’t Platforms Trustworthy?

Build

3

SECURITY

Protected Mode (rings) protects OS from apps ...



... and apps from each other

...

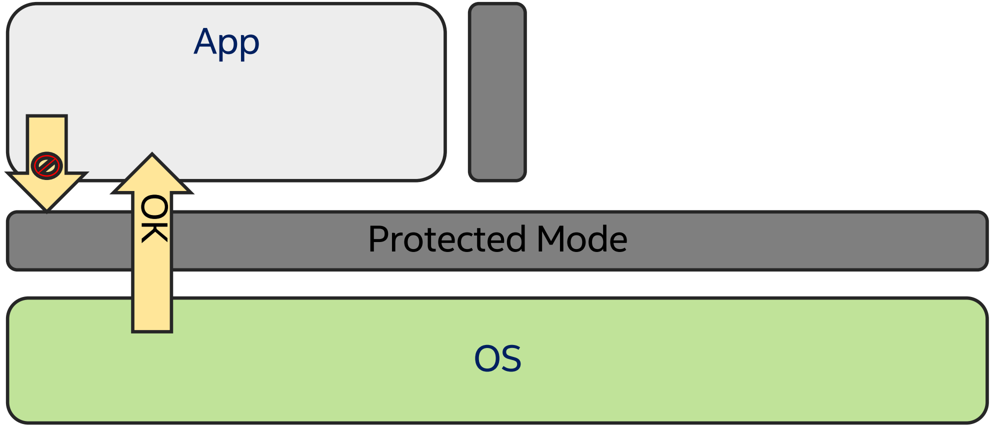
... UNTIL a malicious app exploits an OS flaw to gain full privileges and then tampers with the OS or other apps

Applications are not protected from privileged code attacks

Build 3

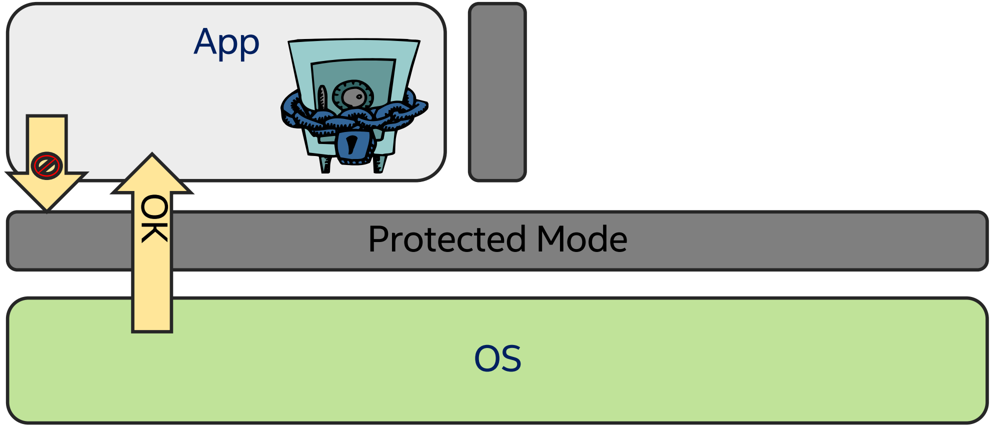
SECURITY

Allowing App Developers to Secure Data



Allowing App Developers to Secure Data

Intel® SGX provides a safe place for code and data in the application

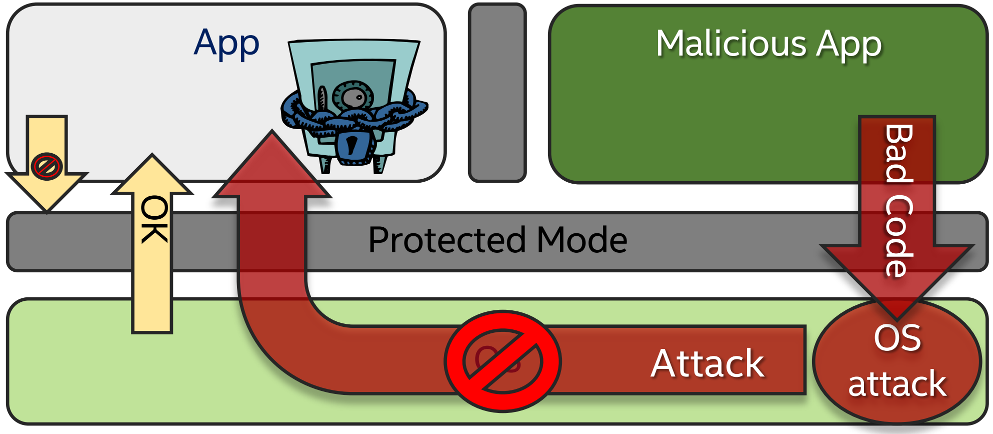


Build
3

SECURITY

Allowing App Developers to Secure Data

Intel® SGX provides a safe place for code and data in the application



Undetected malicious software cannot access secrets

Secrets are protected from bad actors with access to the platform

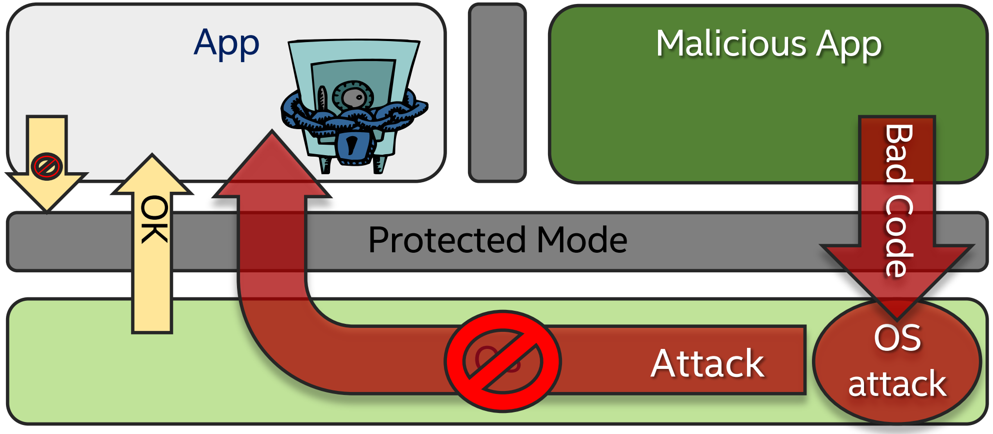
Build

3

SECURITY

Allowing App Developers to Secure Data

Intel® SGX provides a safe place for code and data in the application



Undetected malicious software cannot access secrets

Secrets are protected from bad actors with access to the platform

Need a safe as well as guards

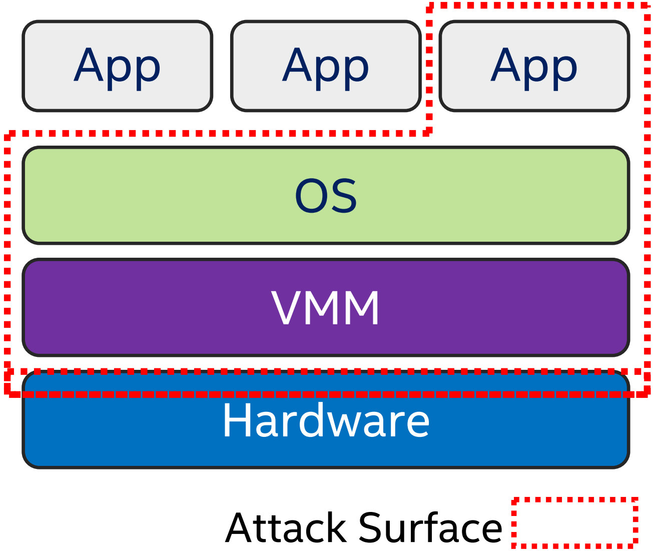
Build

3

Reducing the Attack Surface with Intel® SGX

SECURITY

Attack surface for legacy platforms



App

App

App

OS

VMM

Hardware

Attack Surface

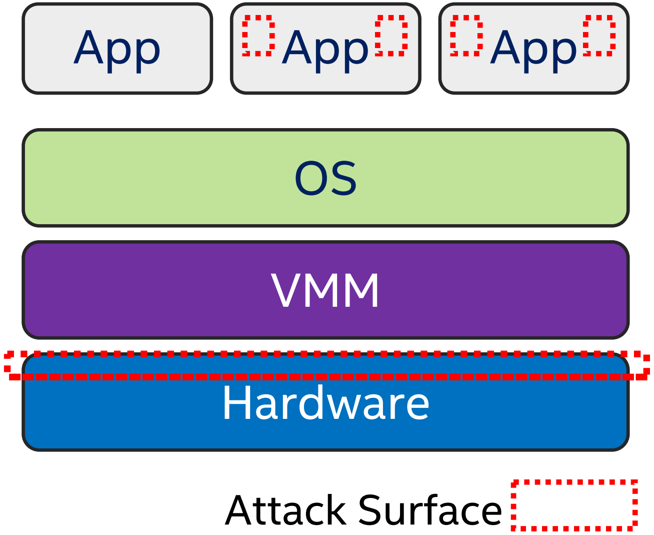
Build

3

Reducing the Attack Surface with Intel® SGX

SECURITY

- Application can defend its own secrets
- Small attack surface (Application's private areas + HW)


Attack surface with Intel®SGX

Reducing the Attack Surface with Intel® SGX

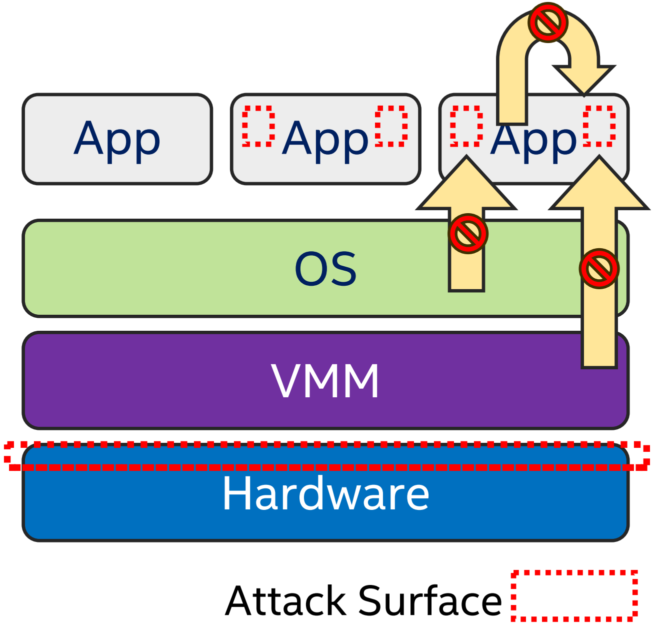
Build

3

SECURITY

- Application can defend its own secrets
- Small attack surface (Application's private areas + HW)
- Malware that subverts any other SW component unable to steal app secrets in private areas

Attack surface with Intel®SGX



Edge Platform Key Protection

SECURITY

Use-Case: Private key Protection on Edge Compute platforms using Hardware TEE

Hardware: Intel Icelake Platform with Intel Software Guard Extension (SGX) enabled

Software:

1. Existing Applications consuming keys (for e.g.: NGINX)
2. PKCS #11 Interface on standard crypto library, e.g.: OpenSSL)
3. SGX Enablement (UEFI BIOS, OS)^1
4. Crypto API Toolkit for Intel® SGX based on SoftHSMv2 ^2

^1 - only for provisioning, resource allocation, management, outside of trust boundary
^2 - https://github.com/intel/crypto-api-toolkit - reference implementation

Architecture SGX Based Key Protection

SECURITY

Platform Attestation and Private Key Flow

Infrastructure Enclaves attest Platform, CPU, TCB Level, Identity and integrity of key protection enclave along with hash of session public key ^1

1. TEE (on Edge Compute Node) generates attestation quote and sent to server over secure channel
2. After successful verification, attestation server wraps private keys ^2and sends resulting wrapped keys over secure channel to be used by TEE on Edge Compute Node
3. TEE on Edge Compute Node unwraps keys and secures inside the enclave

Private Key Operations always executed inside TEE

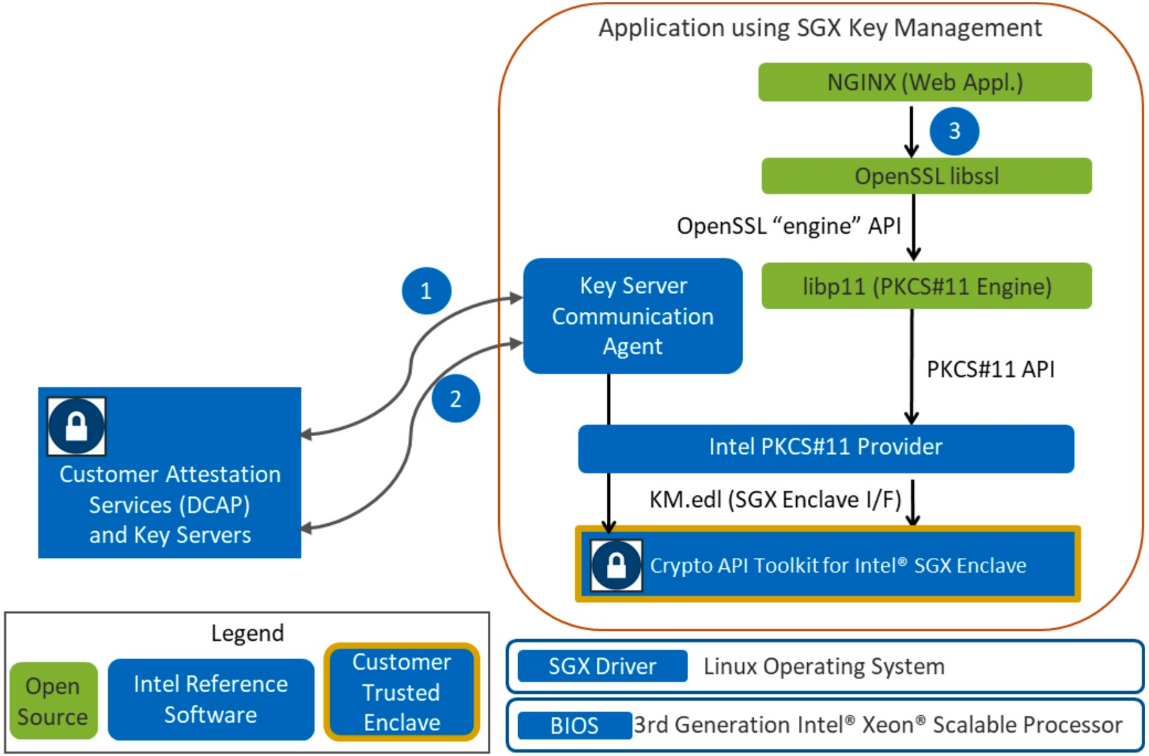
- Key pair tokens provisioned and stored in TEE after successful attestation, authentication and authorizations: Private key is never exposed in the clear outside of TEE
- Application (e.g.: NGINX) request use of the key via OpenSSL Libp11 engine (PKCS#11 API)

^1 Session Keys used for wrapping are destroyed after unwrapping when the session ends

^2 Could use a centralized Key Management Service, Private HSM or Cloud HSM

Reference Implementation

SECURITY



1. SGX Enclave Launch with Attestation
2. Customer Key Delivery into Enclave
3. Application (e.g.: NGINX) uses Key Protected Keys inside Enclave

OPEN POSSIBILITIES.

Summary and Recommendations

SECURITY

Summary

- Scalability: Solution can scale to any number of Edge Compute Nodes
- Performance: Since only key operations are moved to TEE, data path is not impacted (Throughput latency, connections etc.)
- Security: HW TEE Protection removes most attack scenarios (vs. keys in clear in memory during runtime), Reduces attack surface and removes most of SW, privilege FW, OS/Kernel etc. from trust boundary

Operational Recommendations

- Keep all security (for example key protection code base) as small as possible and secure (remember app itself is still in trust boundary!)
- Follow secure software development guidelines
- Test for software attacks and side channel resistance

Call to Action and Additional Information

SECURITY

Get Involved: Opensource libraries and reference implementation links below

Opensource reference Software and Hardware Platforms: Available Now

https://github.com/intel/crypto-api-toolkit - Crypto API Toolkit for SGX

https://github.com/intel/SGXDataCenterAttestationPrimitives - SGX Attestation Libraries

https://github.com/cloud-security-research/sgx-ra-tls - SGX remote attestation with TLS connection setup

https://github.com/intel/sgx-ra-sample - Remote Attestation Sample

https://github.com/intel/intel-sgx-ssl - SGX SSL reference implementation

https://01.org/key-management-reference-application-kmra - Reference Implementation Source Code

Additional Information

https://download.01.org/intel-sgx/latest/linux-latest/docs/ - SGX Documents

https://networkbuilders.intel.com/solutionslibrary/intel-software-guard-extensions-intel-sgx-key-management-on-the-3rd-generation-intel-xeon-scalable-processor-technology-guide - Key Management Reference Guide

https://software.intel.com/content/www/us/en/develop/articles/intel-sdm.html#combined – Intel® 64 and IA-32 Architectures Software Developer Manuals

https://01.org/sites/default/files/downloads/intelsgxnginxprivatekey3rdgenintelxeonspuserguide634677v1.pdf

https://01.org/sites/default/files/downloads/intelsgxkeymanagement3rdgenintelxeonsptechguide635272v1-1.pdf

Acknowledgement

Kapil Sood: Principal Engineer, Intel - CPU, Platform and Security Architect
Lead for Key Management Reference Architecture (KMRA) project at Intel

Key Contributors
Veronika Karpenko
Jon Strang
David Lu
Darragh Coen

Thank you!

OCP GLOBAL SUMMIT

NOVEMBER 9-10, 2021