TrustZone Understanding

Mukam Augusta

TrustZone

TrustZone (ARM)

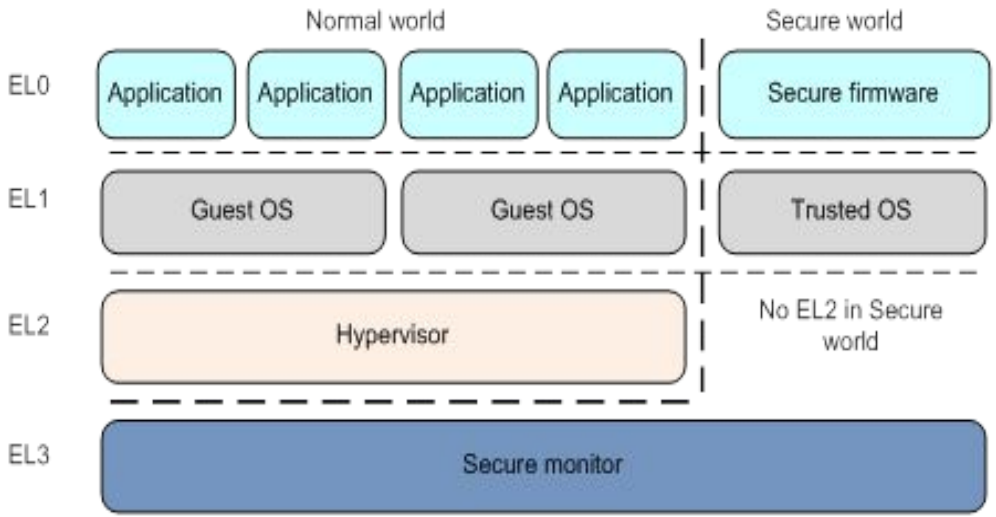
Introduction

- Developers need to secure systems beginning at the **lowest levels**
- **TrustZone** : hardware security extension technology
  - provide secure execution environment by splitting computer resources
  - **normal world (REE)** and **secure world (TEE)** corresponding to virtual cores on a CPU
- Hardware barriers (memory regions, busses, peripherals, interrupts, etc) for the normal world
  - The memory system prevents it from accessing regions of the physical memory designated as secure (which is not restricted)
- **Monitor mode:** The mechanism to context switch between the two worlds (interruption)
  - triggered by executing a dedicated instruction: **Secure Monitor Call (SMC)**

TrustZone

CPU Mode

- Only trusted applications running on a TEE (Secure World) have complete access to the main processor, peripherals and memory
- The current processor mode is determined by the mode field (M) of the current program state register (CPSR)
- The main memory is also flagged with the **Non-Secure bit**

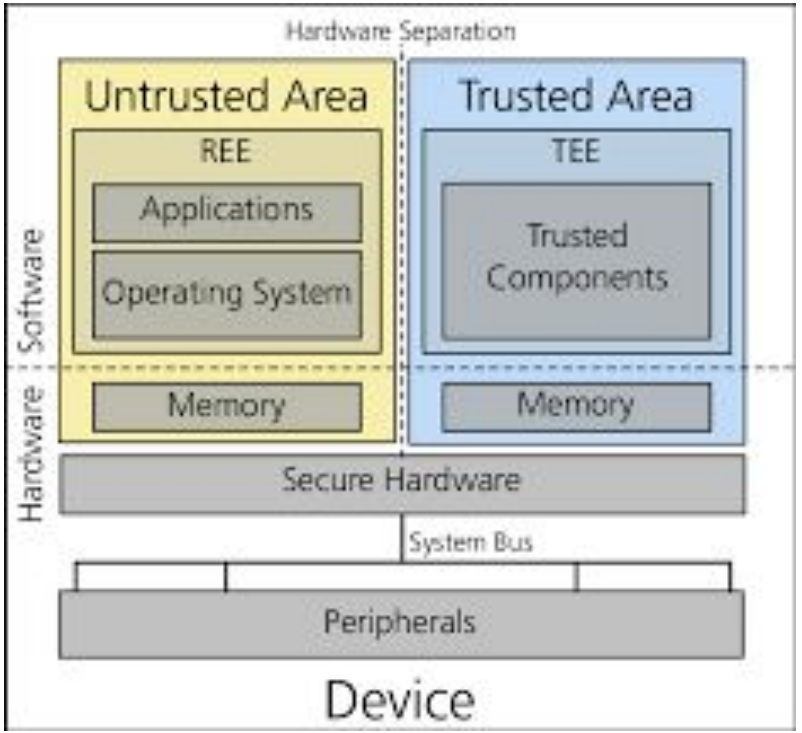


Trusted Execution Environment

Trusted Execution Environment

Principle

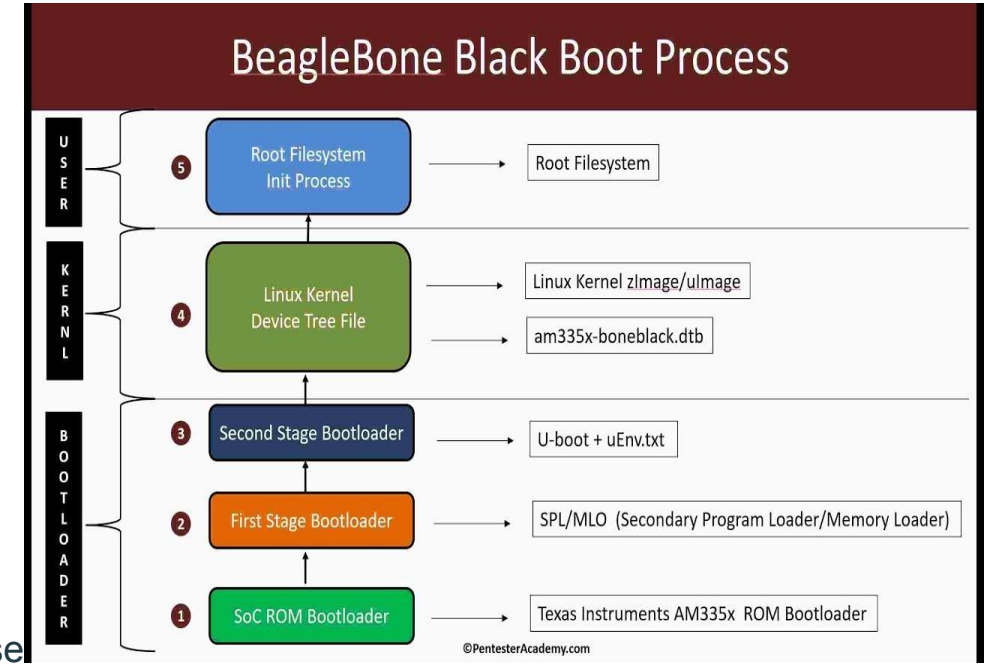
- Security subsystem running in the secure world in parallel to the REE
- Trusted applications (TAs) and associated data are completely isolated from the REE and their applications.
- TAs must run in isolation from other TAs and from the TEE itself.
- imagine an untrusted application running on Linux that wants a service from a TA
  - use an API to send the request to the Linux kernel
  - use the TrustZone drivers to send the request to the TEE OS via SMC instruction
  - the TEE OS will pass along the request to the TA



Trusted Execution Environment

Secure boot

- Build a chain of trust
- Each step is cryptographically verified
  - Binairies
  - System kernel and root filesystem
  - Applications
- Each element is signed and this signature can be verified through a certificate chain placed directly after the signature.
- This chain of trust can be validated thanks to the root certificate of which a SHA256 is placed in a fuse (QFuse), ensuring its integrity.



TEE usages

- **Storage and management of the device encryption keys** that could be used to verify the integrity of the operating system.
- **Biometric authentication methods** (facial recognition, fingerprint sensor and voice authorization), isolating resources within a device to store the biometric algorithm, user credentials and associated data.
- In **mobile e-commerce** applications like mobile wallets, peer-to-peer payments or contactless payments to store and manage credentials and sensitive data.
- TEE is also a suitable environment for **protecting digital copyrighted information** (books, movies, audio, etc) on connected devices such as smartphones, tablets and smart TVs.

Examples of TrustZone

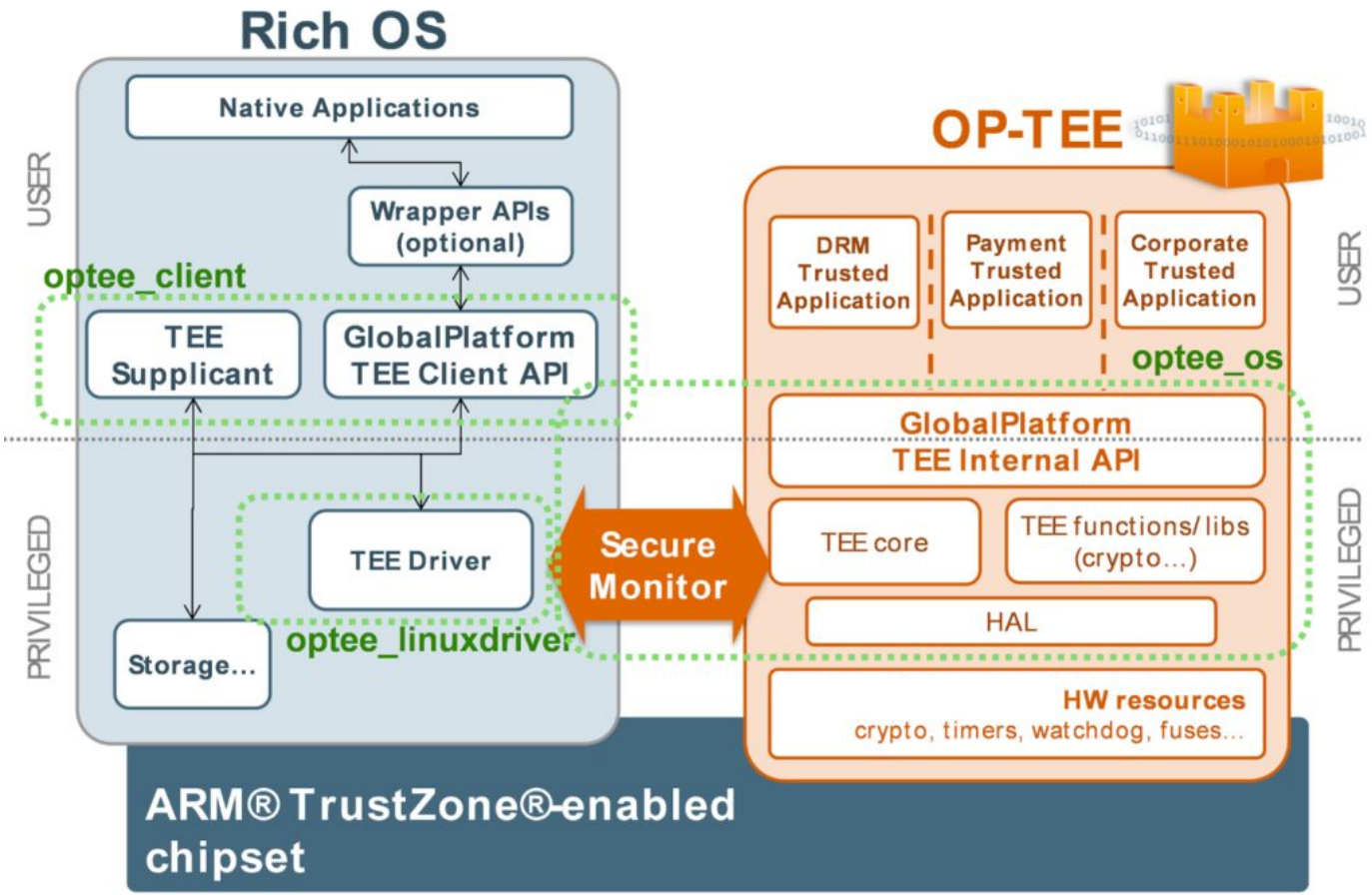
- **Arm's TrustZone** technology offers an efficient, system-wide approach to security with hardware-enforced isolation built into the CPU.
- **MultiZone Security**
- The **AMD Platform Security Processor** (PSP), officially known as **AMD Secure Technology**
- **Intel Software Guard Extensions** (SGX)

- Apple uses a dedicated processor called **SEP** (**Secure Enclave Processor**) for features like data protection, Touch ID, and Face ID. The SEP is responsible for handling keys and other inform ation such as biometrics that is sensitive enough to not be handled by the application processor.
- Google also has a similar solution called **Titan M**, an external chip available on some Android Pixel devices to implement a TEE and handle features like secure boot, lock screen protection, disk encryption, etc.

Examples of TEE

- **Kinibi** is the TEE implementation from Trustonic that is used to protect application-level processors, such as the ARM Cortex-A range, and are used on several smartphone devices like the Samsung Galaxy S series.
- **TEEGRIS** of Samsung
- Qualcomm has its own TEE implementation called **Qualcomm Secure Execution Environment** (QSEE) that is also used on a lot of smartphone devices.
- **iTrustee** is the Huawei implementation of a TEE operating system for ARM's TrustZone.
- **Trusty** is an open source project from Google that implements a TEE for Android. It is compatible with ARM's TrustZone and Intel's Virtualization Technology.
- **OP-TEE** (Open Portable Trusted Execution Environment) is an open source TEE designed as a companion to a non-secure Linux kernel running on ARM Cortex-A cores using the TrustZone technology.

OPTEE


Rich OS
Native Applications
Wrapper APIs (optional)
optee_client
TEE Supplicant
GlobalPlatform TEE Client API
TEE Driver
Storage...
optee_linuxdriver
USER
PRIVILEGED
Secure Monitor
OP-TEE
DRM Trusted Application
Payment Trusted Application
Corporate Trusted Application
optee_os
GlobalPlatform TEE Internal API
TEE core
TEE functions/ libs (crypto...)
HAL
HW resources
crypto, timers, watchdog, fuses...
USER
PRIVILEGED
ARM® TrustZone®-enabled chipset

Conclusion

Conclusion

- Virtualization effect
- A TEE implementation is just another layer of security and has its own attack surfaces that could be exploited.
- And numerous vulnerabilities were already found in different implementations of a TEE using TrustZone!

References

References

- https://sefcom.asu.edu/publications/trustzone-explained-cic2016.pdf
- https://embeddedbits.org/introduction-to-trusted-execution-environment-tee-arm-trustzone/
- https://blog.quarkslab.com/introduction-to-trusted-execution-environment-arms-trustzone.html

TrustZone Understanding

Mukam Augusta
mukamaugusta5@gmail.com