Review “TrustShadow”

Name: Wenhui Zhang

Paper Name：

Guan, L., Liu, P., Xing, X., Ge, X., Zhang, S., Yu, M., & Jaeger, T. (2017, June). TrustShadow: Secure execution of unmodified applications with ARM trustzone. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services* (pp. 488-501). ACM.

Contribution:

In this paper, Trustshadow, a mechanism is proposed and it is specifically designed for ARM platform to shield a trusted application from malicious OS. This work works with the ARM platform without relying on hypervisor, and requires no modification to legacy programs. TrustZone is used to achieve shielded execution for embedded system applications.

Motivation:

Kernel vulnerabilities are reported every month, and exiting software-based solutions, including kernel ASLR, kernel stack protection, real-time kernel protection, all seem to be inefficient. This motivates us to find a hardware-assisted solution to shield the trusted applications even if the OS is compromised. Furthermore, researchers have found that smartphones are also vulnerable to physical attacks such as cold boot attack.

Existing shielding work mainly focus on x86 platform, utilizing SGX or hypervisor. However, SGX is not available for ARM platform, and hypervisor is an overkill for embedded or IoT settings.

Related works:

Controlled-channel attacks [Oakland’15]; protection from Iago attack:Graphene [Eurosys’14]

Methodology:

ARM TrustZone provides two virtual cores, one for the normal world and the other for the secure world. Each world has its own user mode and privileged mode, except that there is an additional monitor mode in the secure world, which serves

As a gate for world switches. In its original design, applications in the normal world needs to invoke API across the world to call security services in the secure world. Therefore, it cannot protect legacy code. Trustshadow ensures program integrity by verifying the hash value when a page is loaded into the memory. During execution, the program runs in an isolated execution with the OS. As a result, a compromised OS cannot interfere with its execution: (1) Load code page & allocate secure page; (2) Install normal-world PTE Verify S-Page; (3) Install secure-world PTE; (4) Copy N-Page to S-Page; (5) Hash validation.

Results:

	TrustShadow	Intel SGX
Isolation	✓	✓
Measurement	✓	✓
Attestation	−	✓
Sealing	✓	✓
Memory Encryption	✓	✓
Flexibility	✓	−
Unmodified OS	−	−
Unmodified Program	✓	✓

Take away:

Memory isolation through page mapping is a good strategy. (e.g. ZONE_TZ_APP)