A Survey of Trusted Execution Environment Security

Hyundo Yoon*, Junbeom Hur*

*Department of Computer Science and Engineering,
College of Informatics,
Korea University

Abstract

Trusted Execution Environment(TEE), such as Intel SGX, AMD Secure Processor and ARM TrustZone, has recently been a rising issue. Trusted Execution Environment provides a secure and independent code execution, hardware-based, environment for untrusted OS. In this paper, we show that Trusted Execution Environment's research trends on its vulnerability and attack models. We classify the previous attack models, and summarize mitigations for each TEE environment.

1. Introduction

Recently, a hardware-supported trusted execution environment for CPUs has been researched and has reached the mass market. Rather than trusting the operating systems and applications, TEE, such as ARM TrustZone and Intel SGX, allows the execution of the applications, or code, inside the *Enclaves* that are proven secure.

Trusted Execution Environment(TEE) has two separate "world": *normal world* and *secure world*. The division of the world was necessary for TEE, because, in practice, the normal world contains untrusted software or applications that are vulnerable to attack and are easily exploited. This isolation of the normal world from the secure world is done by hardware-enforced mechanisms.

Unlike the Purpose of the TEE, there still exists weaknesses that can be exploited by adversaries. In the past few years, researches on vulnerability have been made, and the needs for supplementation are clear. The researches have been made on TEEs shown below.

- **Trusted Execution Environment**
  - ARM TrustZone
  - AMD Secure Processor
  - Intel SGX

2. Backgrounds

2.1 ARM TrustZone Architecture

TrustZone is a security extension to the ARM architecture with modifications to the processor, memory, and I/O devices [1]. For secure workloads, it provides the system-wide isolated environment for execution. This isolation disables the access from the normal world processes, which gives more secureness.

2.2 AMD Secure Processor Architecture

AMD Secure Processor (AMD-SP) is a microcontroller coprocessor integrated within chipsets of AMD. It provides the functionality needed for security subsystem, secure booting, and corporate asset management. AMD-SP runs secure closed-source AMD-signed kernel code, and it provides functionality related to cryptography such as key generation and key management.

2.3 Intel SGX Architecture

Intel Software Guard Extensions (SGX) is the instruction set architecture that enables the creation of enclave, trusted execution environment. This ensures the integrity and the confidentiality of the code and the data of the program[2].

3. Attack Models and Vulnerabilities

In the past few years, vulnerabilities of different TEEs have been researched. Each of them has features that can be exploited by adversaries. Chart 1 below shows what features of each TEE is vulnerable and can be exploited.

<Chart 1> Each TEE's Vulnerabilities

ARM TrustZone	L1 Cache
AMD Secure Processor	Maseter Key
	RYZENFALL
	FALLOUT
Intel SGX	Page Table
	Processor Reserved Memory
	L1 Cache
	Branch Target Buffer

3.1 ARM TrustZone

- **TruSpy** attack exploits the cache contention between the normal world and the secure world as a cache timing side channel to extract sensitive information from the secure world [1]. This attack basically uses the technique of prime and probe [3] to learn the victim process' cache access pattern. The workflow of TruSpy attack follows five major steps. First step is identifying the memory for cache priming usage. Second step is filling out the cache with memory content from the address space of the attacker. Third step is triggering the execution of the victim process in the secure world. Fourth step is to measure the

change in cache configuration when victim's execution is done in the secure world. The last step is to analyze the gathered information and recover the secret information.

- **ARMageddon** attack uses malicious application that does not require any permission, can be executed in unprivileged userspace, and does not require a rooted device [4]. There are three main cross-core cache attack models: *Prime+Probe, Flush+Reload, Evit+Reload, and Flush+Flush.*

3.2 AMD Secure Processor

- **MASTERKEY** is a set of three vulnerabilities allowing three distinct pathways to bypass *Hardware Validated Boot* on EPYC and Ryzen and achieve arbitrary code execution on the Secure Processor itself [5]. Attacker installs persistent malware inside AMD-SP, disables the security features, and re-flash the BIOS. The malicious update of BIOS often passes the digital signature verification that is BIOS-specific.
- **RYZENFALL** is a vulnerability that is a set of design and implementation flaws inside the AMD Secure OS. RYZENFALL allows the malware running on the main processor. This vulnerability is only found and exploited on Ryzen, Ryzen Pro and Ryzen Mobile. The prerequisite for the exploitation is that an adversary need to run a program with local-machine elevated administrator privileges, thus able to break the hardware security seal.
- **FALLOUT** vulnerability is related to a set of design-flaw residing inside EPYC's Secure Processor's boot loader component. For the exploitation, the prerequisite is same with Ryzen. A program should run with local-machine elevated administrator privileges. FALLOUT has impact on VTL-1 memory write, disabling SMM protection, and VTL-1 memory read.

3.3 Intel SGX

- **CacheZoom** is an attack tool that virtually track all memory accesses of SGX enclaves with high spatial and temporal precision [6]. It uses *Prime+Probe* technique for attack. The attacker can retrieve accesses that are secret dependent by the target. The implementation of CacheZoom considers how the noise sources are limited and how one increases the time resolution to obtain clean traces. In addition, the CacheZoom attack can retrieve secret keys of various AESs running inside an enclave. Retrieval of secret keys can be done by the S-BOX implementation and T-Table implementation.
- **SGX-BOMB** attack locks the processor by triggering the defense mechanism of the MEE of SGX intentionally. SGX-BOMB has three major steps for the workflow. First step is to find the row addresses that reside in the same bank. Second step is to find interleaved row addresses. Last step is the trigger the bit flipping. The SGX-BOMB attack is easy to succeed by the Rowhammer attack for the DRAM.
- **Dark-ROP** attack is related to memory corruption vulnerability. The target of this attack is emulation of SGX enclave and the crypto-key of enclave. During the attack, three oracles are exploited in order to gain the hints with respect to the code in the unknown binary in the enclave.

4. Mitigations

For AMD Secure Processor, the mitigations are not known yet for vulnerabilities mentioned above.

For ARM TrustZone, the vulnerabilities can be prevented or eased by using hardware instructions for certain cases. If that is not the case, then a software-only bit -sliced implementation should be employed.

In order to avoid attacks in Intel SGX, there are some SGX-specific features be considered. Deploying the CFI in the enclave is one of the options. By using one of the general registers to point to the reference table that defines allowed target blocks, it can be easily bypassed by the attacker manipulating the context in trapped thread. Usage of DRAM that is Rowhammer-free is an simple mitigation for SGX-BOMB attack, because the root-cause of the attack is vulnerable DRAM, not the implementation of SGX.

5. Conclusion

As different Trusted Execution Environment become a useful tool for implementation of various application, it is important that TEE's vulnerabilities should be fixed and need a further research.

6. Acknowledgement

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2019-0-00533, Research on CPU vulnerability detection and validation)

References

[1] N. Zhang, K. Sun, W. Lou, D. Shands, W. J. Lou, and Y. T. Hou, "TruSpy: Cache Side-Channel Information Leakage from the Secure World on ARM Devices", 2016

[2] J. H. Lee, J. S. Jang, Y. J. Jang, N. H. Kwak, Y. S. Choi, C. H. Choi, T. S. Kim, M. Peinado, and B. B. Kang, "Hacking in Darkness: Return-oriented Programming against Secure Enclaves", 2017

[3] D. A. Osvik, A. Shamir, and E. Tromer, "Cache attacks and countermeasures: the case of aes," in Topics in Cryptology–CT-RSA 2006, pp. 1–20, Springer, 2006

[4] M. Lipp, D. Gruss, R. Spreitzer, C. Maurice, S. Mangard, "Armageddon: Cahce Attacks on Mobile Devices", In USENIX Security Symposium(2016)

[5] CTS Lab. 2018. Severe Security Advisory on AMD Processors. https://safefirmware.com/amdflaws_whitepaper.pdf

[6] A. Moghimi, G. Irazoqui, and T. Eisenbarth. CacheZoom: How SGX Amplifies The Power of Cache Attacks. Technical report, arXiv:1703.06986 [cs.CR], 2017. https: //arxiv.org/abs/1703.06986.

[7] Y. Jang, J. Lee, S. Lee, T. Kim, "SGX-Bomb: Locking down the processor via Rowhammer attack", *SysTEX*, 2017.