

Research Report for Secure EL2

Shiqing Liu

South University of Science and Technology
Shenzhen, China

Email: liusq2017@mail.sustech.edu.cn

Yinhua Chen

South University of Science and Technology
Shenzhen, China

Email: 11612127@mail.sustech.edu.cn

Abstract—The abstract goes here.

I. INTRODUCTION & BACKGROUND

Trusted Execution Environment (TEE) is a security domain in CPU who is isolated from the other. TEE is capable of providing confidentiality and integrity for the codes and data stored in it.

A. TrustZone

TrustZone is the first commercial hardware architecture to support TEE. After development in these years, TrustZone is widely used in mobile devices to provide trusted computing. TrustZone separates the hardware and software resources, such as memories, I/O, and interrupts, into two parts, Secure world and Non-secure world (i.e. Normal world). Each world has their own kernel space and user space, associated with cache, memory, and etc. Non-secure world is not capable of accessing the resources in Secure world, but Secure world has access rights to resources in Non-secure world. In addition, each world has its own privileges level, represented in exception levels. Note that there is only Non-secure world having exception level 2 (EL2). Correspondingly, processors have two state, Secure state and Non-secure state, to handle affairs in two different worlds.

B. Virtualization of TEE Technology

Virtualization is a technology to divide the hardware resource and create virtual instances who act like a real system. Since 1960s, virtualization has been always popular in computer science. The trend of virtualization is more intense these years, especially after cloud computing becomes more and more popular.

However, TrustZone is not designed to be virtualizable in the past few years. All virtual machines managed by the same hypervisor should trust the same TEE, while there are amount of vulnerabilities discovered in major vendors' TEE-kernel.

Basic idea to virtualize TEE.

Virtualized TEE in industry. Sierraware. Imagination Technologies

The arrangement of the report. Firstly talk about the software-simulation based virtualization(vTZ). second hardware-assisted based virtualization(not SEL2)

II. HYPERVISOR

A. KVM

vTZ is proposed by Christoffer Dall and Jason in 2014 on ASPLOS. As ARM CPUs become increasingly common in mobile devices and servers, there is a growing demand for providing the benefits of virtualization for ARM-based devices. At that time, ARM Xen has already been proposed, but there are some disadvantages on ARM Xen. For example, ARM Xen requires much maintenance effort since it is a hypervisor interacting with hardware by itself. So it needs to write its own code to be compatible with different ARM-based platform. However, KVM does not have this disadvantage. KVM is just a kernel module of Linux. It only provides hypervisor-related functions in KVM code and gives Linux full privilege to interact with hardware. Since Linux is implemented on all ARM-based platforms, KVM can be adapted to all ARM-based platforms by a good-defined API between KVM and Linux.

The reason we try to understand KVM design is that in S-EL2 project we need a hypervisor running in NWd to build the whole model. And the preferred choice is KVM. Understanding the design and implements can help us to build the model and the following modification on KVM.

B. HypSec

HypSec is proposed by Shih-Wei Li, John S. Koh, and Jason Nieh in 2019. HypSec is a new hypervisor design scheme. With HypSec, we can retrofit an existing hypervisor. After retrofitting the hypervisor, its TCB will decrease with several orders of magnitude and its performance overhead is still acceptable. In conclusion, this is a hypervisor design pattern which can improve security of a hypervisor with a cost of small overhead.

The reason that we read this paper is that there are two hypervisors in our S-EL2 model. A normal world hypervisor like KVM in NWd and a secure world hypervisor like Hafnium in SWd. In future we need to build a cross-world communication channel between these two hypervisors. And then there is a problem, the TCB might be too large. We might need to include KVM, hafnium and hafnium code lines into TCB. The TCB can be over millions lines of code, which is not acceptable. With HypSec, we can reduce KVM and hafnium's TCB by a several orders of magnitude, so a much smaller TCB can be gained.

III. SOFTWARE SIMULATION BASED TEE VIRTUALIZATION(?)

The software simulation base TEE virtualization is to create a virtual TEE and simulate a part of, even all of TEE functionalities in a software way. The important component of TEE, such as interrupt controller and memory controller, is achieved by the software. The representative is vTZ.

A. vTZ

vTZ is proposed by Zhichao Hua in 2017 on usenix conference. TrustZone does not provide any hardware support for virtualization at that time. All VMs in normal world should share the same TEE and it leads to a serious security issue for breaking the isolation of VMs. vTZ aims to virtualize TrustZone and provide each VM an independent TEE. Different from other jobs, it makes use of the existed TEE architecture instead of design a new one. vTZ separates the functionality and security by running another VM in normal world as guest TEE to serve the VM. The security is ensured by the TrustZone hardware isolation. In details, a tiny monitor in secure world controls the memory mapping and world switching. A Constrained Isolated Execution Environment (CIEE) provides isolation.

The design of vTZ is shown in figure. There are four secure modules in vTZ, Secure Memory Mapping (SMM), Secure World Switching (SWS), Control Flow Lock (CFlock), and Constrained Isolated Execution Environment (CIEE). Running in secure world, SMM controls all memory mappings. To achieve that, the hypervisor is not allowed to contain the sensitive instruction to access and control the translation table, not to mention the first level translation and second level translation. Instead, the sensitive instruction is replaced by the invocation to SMM. SMM will check whether the request is legal. SWS is responsible for the world switching and it is another module in secure world. It will check the VID of the VMs and restore the correct VM. As long as SWS ensures the correctness of the switch from VMs to the hypervisor and the hypervisor to VMs, all switches in vTZ will be covered, because switches between VMs are composed of both switches. As a module in normal world hypervisor, CFlock is used to prevent the execution control flow from tempering by protecting exception vector table. CFlock can be used to ensure SWS will eventually handle all switches. The last module is CIEE, a region in normal world EL2, running with the hypervisor, serves the guest-TEE. CIEEs contain the software implementations of TrustZone. In other words, vTZ does not utilize all the hardware in consideration of the support of the virtualization. To protect the isolation and security of CIEE, CIEE is designed to execute atomically and independently. The CIEE itself is also verifiable. As a result of it, CIEE can't be falsified.

Based on these four modules, vTZ reaches the goal of secure boot, memory isolation, and privilege isolation.

The advantages of vTZ are high efficiency and high security. In vTZ evaluation, the performance overhead of applications is low compared to the native environment, TrustZone, and Xen hypervisor. For security, vTZ has a TCB of thousands

lines of code, which is far less than one who putting an entire hypervisor and guest TEE into secure world. And vTZ utilizes TrustZone to protect itself. In details, the privilege mode and secure world of TrustZone make the isolation and security of CIEE, SWS and SMM. In a result, vTZ successfully defends several kinds of attacks, such as code-reuse attacks, DMA attacks, debugging attacks and code tempering attacks.

However, vTZ also has some disadvantages. vTZ is not compatible to the existing commercial hypervisor. In vTZ, the hypervisor is not allowed to have sensitive instructions related to address translations. In addition, the module, CFlock should run in hypervisor to support vTZ. So the hypervisor must modify their source code in order to run on vTZ. That is a huge disadvantage to the promotion and application.

In conclusion, vTZ is mostly close to our goals Secure EL2 project. vTZ is also a close competitor of our project. We want to compare the performance with vTZ and prove the advantage of our one.

B. TEEv

TEEv is another TEE virtualization architecture to provide restricted TEEs, i.e. vTEEs. These vTEEs are concurrent in runtime and isolated from each other. (More about vTEE). A tiny hypervisor, TEE-visor, running in Secure World to manage these vTEEs and to provide isolation between vTEEs. It allows each vTEE from different manufacturers to host their own TA.

To address the challenge of the lack of TrustZone virtualization support, TEEv deploys the TEE-visor and vTEEs at the same privilege level, secure EL1. In addition, TEEv modified the commercial TEE by scanning vTEE's binary to remove the MMU-related instructions, so that TEE-visor can exclusively control the MMU. Controlling the MMU, TEE-visor can protect itself in security and confidentiality, and isolate vTEEs by protecting each vTEE's entries. As a result of removing the sensitive instruction, TEE-visor is responsible for the communication between vTEE and corresponding client application (CA). In details, TEE-visor handles the request from CAs and verifies the page access from TA to CA. For isolation of vTEEs and REEs, TEE-visor manages the page table exclusively in order that each vTEE and each REE can only access their own page. Similar to vTZ, TEE-visor protects the kernel of vTEEs by setting their kernel as read only.

The advantage of TEEv

IV. INTRODUCTION

This demo file is intended to serve as a "starter file" for IEEE conference papers produced under L^AT_EX using IEEEtran.cls version 1.8b and later. I wish you the best of success.

mds

August 26, 2015

A. Subsection Heading Here

Subsection text here.

1) Subsubsection Heading Here: Subsubsection text here.

V. CONCLUSION

The conclusion goes here.

ACKNOWLEDGMENT

The authors would like to thank...

REFERENCES

- [1] H. Kopka and P. W. Daly, *A Guide to L^AT_EX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.