Research Report for Secure EL2

authors in alphabetical order

Abstract

Some sorts of documents need abstracts. Others do not.

1 Introduction & Background

Under data explosion adn data becaming treasures, the cloud computing is valued and being developed to store and dig the data mine. In order to meet the various needs of cloud computing, virtualization has become an important technology of cloud computing. In the meanwhile, the securtiy of cloud is also to be taken more serious. Lots of techniques is used, including TEE.

Trusted Execution Environment (TEE) is a security domian in CPU who is isolated from the other, i.e. REE. TEE is capable of providing confidentiality and intergity for the codes and data stored in it. All accepted code have been authorized by TEE so that it is trusted. Besides, TEE need all code and assets load and start in a expected way so that it won't be tempered. A TEE should also responsible for isolation between TEE and REE. In addition, each TA can only access it own assets. A TEE is a important composition of trusted computing.

1.1 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 seperates 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-seucre 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.

1.2 Secure Guard Extension (SGX)

1.3 Virtualization of TEE Technology

Virtualization is a technology to divide the hardware resouce 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 year, 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 shoudld trust the same TEE, while there are amount of vulnerabilities discovered in major venders' TEE-kernel.

Basic idea to virtualize TEE. simulation base virtualize. hardware virtualize. an conclusion of vTZ and TEEv method.

Virtualized TEE in industry. Sierraware. Imagination Technologie (multi-trust TEE)

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

2 Hypervisor

2.1 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.

2.2 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 serveral 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

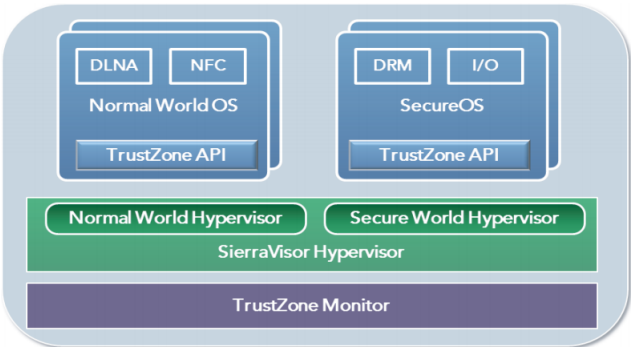

Figure 1: SierraVisor Architecture Overview

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.

2.3 SierraVisor

Proposed by Sierraware, a company dedicated to deliver the virtualization technologies and security solutions, SierraVisor is a hypervisor for ARM which is integrated with TrustZone and Android. Specifically, SierraVisor supports TrustZone Monitor as VMM on Cortex A9 and ARM11, i.e. SierraVisor use TrustZone Monitor and extend it as a hypervisor. The guest can continue to work without modification. And kernel can continue to run in supervisor mode. Guest OSes can run in their inidividual containers.(1)

Above is the architecture of SierraVisor. At the bottom there are TrustZone Monitor and SierraVisor Hypervisor. The SierraVisor has two hypervisor, normal world hypervisor and secure world hypervisor running in each world. Both hypervisor supports multiple OS in Normal World and Secure World.

As a product, SierraVisor is quite old because the supported SoC presenting on the their website is up to Cortex A15, a processors implementing the Armv7-A architecture. The timestamp of most web pages about SierraVisor freezes in 2012. But the idea of SierraVisor and its contribution on supporting multiple TEE-kernel is on the right direction.

(advantages: high compatibility, type-2 hypervisor, as long as linux support, mine supports.)

3 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 achieve by the software. The represensative is vTZ.

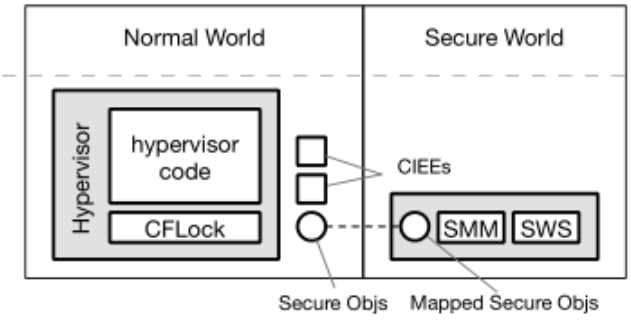

Figure 2: vTZ system designs

3.1 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 shared 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 make use of the existed TEE architecture instead of design a new one. vTZ seperate the funcutionality and security by running another VM in normal world as guest TEE to serve the VM. The security is ensure by the TrustZone hardware isolation. In details, a tiny monitor in secure world control 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 module 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 allow 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 to 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 ensure the correction 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 execption control flow from tempering by protecting exception vector table. CFlock can 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 contiain the software implementations of TrustZone. In other words, vTZ does not utilize all the hardware in considerations 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.

Base on these four modules, vTZ reaches the goal of secure boot, memory

isolation, and privilege isolation.

The advantages of vTZ is 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 a 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 defenses 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 exisiting commercial hypervisor. In vTZ, the hypervisor is not allow to have sensitive instructions related to address translations. In addition, the module, CFLock should run in hypervisor to support vTZ. So the hypervisor must modified 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 competiter of our project. We want to compare the performance with vTZ and proof the advantage of our one.

3.2 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. The vTEEs is not like traditional TEEs who can access the REE's resources. They can only access their own assest(More about vTEE). A tiny hypervisor, TEE-visor, running in Secure World to manage these vTEEs and to provide isolation between vTEEs. TEE-visor 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. Controling 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 verify the page access from TA to CA. For isolation of vTEEs and REEs, TEE-visor manage the page table exclusively in order that each vTEE and each REE can only access their own page. Similar to vTZ, TEE-visor protect the kernel of vTEEs by mapping their kernel's page as read only. (other details)

The advantage of TEEv is high security and high flexibility. The vTEE in TEEv is not a traditional TEE and it is restricted. If a vTEE is malicious, it can not affect the other vTEEs and REEs. Its TCB is also low. The TEE-visor only contians 3800 lines of C and assembly code. Each TEE should only trust their own code base and TEE-visor. (CVEs and attacks) For the flexibility, TEEv allows to install multiple vTEEs and support each own TA. ()

TEE	hardware supported	modify TEE-kernel	support multiple types of TEE-kernel	modify hypervisor	transparent to TA and REE	s
vTZ	yes	yes	yes	yes	yes	
TEEv	No	yes	yes	yes	yes	
Secure EL2	yes	no	yes	yes (a litttle)	yes	
TDX	yes	-	-	-		
SEV						
CCA						

Table 1: Comparison with current TEE virtualization architectures.

4 Virtualization of Hardware-based TEE Technology

4.1 vTPM

Trusted Platform Module (TPM) is a international standard for cryptoprocessors. Similar to TrustZone, it is a secure hardware extensions and designed for trusted computing. Trusted Computing Group (TCG) suggests every system should have a TPM. It provides the functionalities of attestation of system state, generation and conservation of the cryptographic data, identification of platform. Since it is a hardware standard, it is naturally tamper resistanced from software attacks.

For virtual machine, vTPM is a solution to virtualize TPM into multiple vTPM instances. Each vTPM instance will provide security support for its respective virtual machine just like a physical TPM.

5 Design of S-EL2

Primary design of S-EL2.

6 Discussion

we discussion the differences between S-EL2 and related works. List at least three innovations.

The VM and vTEE of vTZ run in NW, but every time the page table is changed, the world must be switched. It is not necessary in SEL2. vTZ has multiple VMs and multiple vTEEs, switching between VMs, switching between vTEEs, VMs Switching between vTEE and vTEE requires a world switch. But in SEL2, only the switching between VM and vTEE requires world switching, and vTZ needs to modify the binary of the NW hypervisor.

7 Conclusion

The conclusion about the research report.

References

[1] 2014. Sierraware Overview. https://www.sierraware.com/sierraware_tee_hypervisor_overview.pdf.