2023-04-06

Created: 2023-04-26 23:54:14

Questions

- Virtualization `(230327.0P-TEE.txt)`
  - `[10:54]` Does OP-TEE support VMs? `(OP-TEE.0327.md)`
  - `[11:05]` Is the technology of OP-TEE supporting VMs mature? `(OP-TEE.0327.md)`
- Hypervisor & TrustZone `(230407.Arm.Hypervisor.txt)`
  - `[10:54]` Can hypervisor run in secure world? `(OP-TEE.0327.md)`
  - `[11:03]` What is difference of runnin hypervisor in secure world and normal world. Please also summarize it in a table. `(OP-TEE.0327.md)`
  - `[20:56]` Please compare TrustZone and Hypervisor and also summarize in a table. `(TrustZone.0329.md)`
- 2017 vTZ `(230406.Papers.OP-TEE.txt)`
  - `[11:33]` Paper: vTZ: Virtualizing ARM TrustZone `(Papers.OP-TEE.0406.md)`
- 2019 TrustedVIM `(230406.Papers.OP-TEE.txt)`
  - `[11:34]` Paper: Cloud & Edge Trusted Virtualized Infrastructure Manager (VIM) - Security and Trust in OpenStack `(Papers.OP-TEE.0406.md)`
- Compare 2017 vTZ and 2019 TrustedVIM `(230406.Papers.OP-TEE.txt)`
  - `[11:35]` What is difference between two papers? Please also summarize it in a table. `(Papers.OP-TEE.0406.md)`
  - `[11:36]` What is same between two papers? `(Papers.OP-TEE.0406.md)`
  - `[11:39]` What is difference of virtualization tech between two papers? `(Papers.OP-TEE.0406.md)`
- Hypervisor `(230407.Arm.Hypervisor.txt)`
  - `[16:07]` What is hypervisor? `(OP-TEE.0327.md)`
  - `[20:58]` Is hypervisor a software or a hardware? `(TrustZone.0329.md)`
- 2020 HETEE `(230406.Papers.OP-TEE.txt)`
  - `[16:33]` Title: Enabling Rack-scale Confidential Computing using Heterogeneous Trusted Execution Environment `(Papers.OP-TEE.0406.md)`
- 2020 SoK TrustZone `(230406.Papers.OP-TEE.txt)`
  - `[16:55]` Title: SoK: Understanding the Prevailing Security Vulnerabilities in TrustZone-assisted TEE Systems `(Papers.OP-TEE.0406.md)`
- 2020 RusTEE `(230406.Papers.OP-TEE.txt)`
  - `[17:27]` Title: RusTEE: Developing Memory-Safe ARMTrustZone Applications `(Papers.OP-TEE.0406.md)`
- 2020 Bao `(230406.Papers.OP-TEE.txt)`
  - `[20:36]` Paper: Bao: A Lightweight Static Partitioning Hypervisor for Modern Multi-Core Embedded Systems `(Papers.OP-TEE.0406.md)`
  - `[22:26]` What is static partitioning? `(Papers.OP-TEE.0406.md)`
- 2019 Research on TrustZone `(230406.Papers.OP-TEE.txt)`
  - `[20:39]` Paper: Research on ARM TrustZone `(Papers.OP-TEE.0406.md)`
  - `[21:15]` The below context comes from "Other Platforms: IoT, Drone, Cloud" of the paper "Research on ARM TrustZone" `(Papers.OP-TEE.0406.md)`
  - `[21:17]` In the context, vTZ run in normal world. Why does not vTZ run in normal world need to trust REE hypervisor? `(Papers.OP-TEE.0406.md)`
- 2019 TEEv `(230406.Papers.OP-TEE.txt)`
  - `[20:41]` Paper: TEEv: Virtualizing Trusted Execution Environmentson Mobile Platforms `(Papers.OP-TEE.0406.md)`
  - `[21:54]` Please compare TEEv and vTZ and also summarize them in a table. `(Papers.OP-TEE.0406.md)`

Q & A

Virtualization `(230327.OP-TEE.txt)`

Question: `[10:54]` Does OP-TEE support VMs? `(OP-TEE.0327.md)`

Answer:

Yes, OP-TEE supports working with virtual machines (VMs) in the sense that it can operate in a virtualized environment managed by a hypervisor. The hypervisor is responsible for managing VMs running in the normal world, while OP-TEE provides secure services in the secure world.

In a virtualized environment, OP-TEE works in conjunction with a hypervisor, like Xen or KVM on ARM, to provide secure services to multiple VMs running on the same hardware platform. The hypervisor ensures the secure world's isolation from the normal world and enforces proper access control policies to prevent unauthorized access to sensitive resources.

However, it is important to note that OP-TEE itself does not directly manage the VMs; it works with a hypervisor that is responsible for managing the VMs in the normal world.

Question: `[11:05]` Is the technology of OP-TEE supporting VMs mature? `(OP-TEE.0327.md)`

Answer:

The technology of OP-TEE supporting VMs is maturing but may not be considered fully mature yet. There have been significant efforts put into enabling OP-TEE to work with virtual machines and hypervisors. OP-TEE does provide support for virtualization, as described in its documentation, and some projects, such as XEN Project, have been working on integrating OP-TEE with their hypervisors.

However, the maturity of this technology can vary depending on the specific use cases, hardware platforms, and hypervisors in question. It is advisable to assess the maturity of the technology on a case-by-case basis and consult with experts or the community to ensure that the chosen solution meets the required security and performance needs.

Hypervisor & TrustZone `(230407.Arm.Hypervisor.txt)`

Question: `[10:54]` Can hypervisor run in secure world? `(OP-TEE.0327.md)`

Answer:

Yes, a hypervisor can run in the secure world, but it is less common. Typically, hypervisors run in the normal world, and they manage VMs in the normal world while ensuring their isolation. However, there are secure world hypervisors, like ARM TrustZone-based hypervisors, which are designed to run in the secure world and manage secure partitions or secure VMs.

A secure world hypervisor manages the secure partitions, including OP-TEE, and ensures their isolation from each other and the normal world. In this configuration, OP-TEE acts as one of the secure partitions managed by the secure world hypervisor. This approach allows multiple secure services to coexist in the secure world, each running in its own isolated partition or secure VM.

It's worth mentioning that having a hypervisor in the secure world adds complexity to the system, and its implementation must ensure that it does not introduce vulnerabilities or weaken the security guarantees provided by the secure world.

Question: `[11:03]` What is difference of runnin hypervisor in secure world and normal world. Please also summarize it in a table. `(OP-TEE.0327.md)`

Answer:

Aspect	Secure World Hypervisor	Normal World Hypervisor
1. Security	Enhanced security and isolation	Less secure environment
2. Trust boundaries	Well-defined trust boundaries for secure services	Trust boundaries managed within the normal world
3. Resource allocation	Resource constraints due to limited secure world size	More resources available for the hypervisor
4. Complexity	Increased development and maintenance complexity	Simpler development and maintenance process
5. Attack surface	Smaller attack surface due to isolation	Potentially larger attack surface
6. Isolation	Strong isolation between secure and normal world	Less strict isolation between guest OSs and hypervisor

Question: `[20:56]` Please compare TrustZone and Hypervisor and also summarize in a table. (`TrustZone.0329.md`)

Answer:

TrustZone and hypervisors serve different purposes in providing security and isolation in computer systems, but they can also be complementary. Here's a comparison between the two:

TrustZone: 1. Purpose: Provides hardware-based security and isolation between the secure and non-secure worlds. 2. Scope: Limited to ARM-based systems. 3. Implementation: Built into the ARM processor, enabled by the SoC vendor. 4. Isolation: Creates two separate execution environments (secure world and normal world). 5. Use Cases: Secure boot, key storage, cryptographic operations, and DRM.

Hypervisor: 1. Purpose: Enables the virtualization of multiple operating systems or environments on a single physical host. 2. Scope: Applicable to various processor architectures (x86, ARM, etc.). 3. Implementation: Software-based, can be a Type-1 (bare-metal) or Type-2 (hosted) hypervisor. 4. Isolation: Creates multiple virtual machines (VMs) or containers, each running a separate OS or application. 5. Use Cases: Server consolidation, sandboxing, running multiple OSes on a single device, and cloud computing.

Summary in a table:

Feature	TrustZone	Hypervisor
Purpose	Hardware-based security	Virtualization
Scope	ARM-based systems	Multiple platforms
Implementation	Built into ARM processor	Software-based
Isolation	Secure and normal worlds	Virtual machines/containers
Use Cases	Secure boot, key storage, crypto operations, DRM	Server consolidation, sandboxing, multiple OSes, cloud computing

Keep in mind that TrustZone and hypervisors can be used together in some cases to enhance the overall security of a system. For example, a TrustZone-enabled secure world can host a secure hypervisor to further isolate and protect sensitive applications.

2017 vTZ (`230406.Papers.OP-TEE.txt`)

Question: `[11:33]` Paper: vTZ: Virtualizing ARM TrustZone (`Papers.OP-TEE.0406.md`)

Answer:

Paper Summary:

The paper titled "vTZ: Virtualizing ARM TrustZone" addresses the challenge of virtualizing ARM TrustZone, which is not

designed to be virtualizable, in order to securely share it among multiple virtual machines (VMs). ARM TrustZone is a security extension providing a secure world or trusted execution environment (TEE) for running security-sensitive code, primarily in mobile platforms. As ARM64 becomes more popular in server markets like cloud, TrustZone is expected to become a key component of cloud security.

The authors present vTZ, a solution that allows each guest VM to have a virtualized guest TEE using existing hardware. vTZ separates functionality from protection by maintaining a secure co-running VM to serve as a guest TEE, while utilizing hardware TrustZone for strong isolation among guest TEEs and the untrusted hypervisor. vTZ employs a tiny monitor within the physical TrustZone that securely interposes and virtualizes memory mapping and world switching. Additionally, it uses a Constrained Isolated Execution Environment (CIEE) with protected, self-contained code to ensure secure virtualization and isolation among multiple guest TEEs.

The authors implemented vTZ on Xen 4.8 on both ARMv7 and ARMv8 development boards and evaluated its performance using two common TEE-kernels, seL4 1 and OP-TEE. The results demonstrated that vTZ provides robust security with minimal performance overhead.

2019 TrustedVIM `(230406.Papers.OP-TEE.txt)`

Question: `[11:34]` Paper: Cloud & Edge Trusted Virtualized Infrastructure Manager (VIM) - Security and Trust in OpenStack `(Papers.OP-TEE.0406.md)`

Answer:

Paper Summary:

The paper titled "Cloud & Edge Trusted Virtualized Infrastructure Manager (VIM) - Security and Trust in OpenStack" discusses the security challenges faced in 5G mobile networks as they adopt virtualization technologies, such as Network Functions Virtualization (NFV), and rely on edge computing for improved performance. The scattered nature of devices and their varying locations make man-in-the-middle, denial-of-service attacks, and tampering easier. Additionally, the dynamic nature of the cloud calls for real-time threat prevention.

The authors propose a solution to address these challenges by incorporating trust into the virtualized edge infrastructure. They present their contributions in developing security services for platform authentication and integrity, which are hosted inside a trusted execution environment (TEE). The proposed solution leverages technologies such as OP-TEE, ARM TrustZone, Virtual Infrastructure Manager (VIM), and OpenStack to achieve the desired level of security.

The authors evaluate the performance overhead of their solution and suggest potential improvements for future work. Their approach aims to provide enhanced security and trust in virtualized cloud and edge computing environments, which are critical components of 5G mobile networks.

Compare 2017 vTZ and 2019 TrustedVIM `(230406.Papers.OP-TEE.txt)`

Question: `[11:35]` What is difference between two papers? Please also summarize it in a table. `(Papers.OP-TEE.0406.md)`

Answer:

The main difference between the two papers is their focus and proposed solutions. The first paper, "vTZ: Virtualizing ARM TrustZone," focuses on virtualizing ARM TrustZone for secure sharing among multiple virtual machines (VMs). The second paper, "Cloud & Edge Trusted Virtualized Infrastructure Manager (VIM) - Security and Trust in OpenStack," aims to address security challenges in 5G mobile networks by incorporating trust into virtualized edge infrastructure.

Here is a summary table highlighting the key differences between the two papers:

Aspect	vTZ: Virtualizing ARM TrustZone	Cloud & Edge Trusted Virtualized Infrastructure Manager (VIM) - Security and Trust in OpenStack
Main Focus	Virtualizing ARM TrustZone	Enhancing security in virtualized edge infrastructure
Application Domain	Mobile platforms and cloud	5G mobile networks and edge computing
Proposed Solution	vTZ	Security services hosted inside a TEE
Key Technologies	ARM TrustZone, Xen 4.8	OP-TEE, ARM TrustZone, VIM, OpenStack
Implementation & Evaluation	ARMv7 and ARMv8 development boards	Performance overhead evaluation and future improvements

The first paper presents vTZ, a solution for virtualizing ARM TrustZone, while the second paper focuses on developing security services for platform authentication and integrity, hosted inside a TEE, for virtualized edge infrastructure in 5G networks.

Question: [11:36] What is same between two papers? (Papers.OP-TEE.0406.md)

Answer:

The similarities between the two papers lie in their overall goals, focus on security, and the use of Trusted Execution Environments (TEEs). Both papers:

1. Aim to enhance security: Both papers focus on improving security in their respective domains. The first paper addresses the challenge of securely sharing ARM TrustZone among multiple virtual machines, while the second paper aims to incorporate trust into virtualized edge infrastructure in 5G mobile networks.

2. Utilize Trusted Execution Environments (TEEs): Both papers leverage TEEs to provide secure environments for running security-sensitive code. The first paper uses ARM TrustZone as a TEE, while the second paper hosts security services inside a TEE.

3. Employ ARM TrustZone: ARM TrustZone is a key technology used in both papers. The first paper proposes a solution (vTZ) to virtualize ARM TrustZone, and the second paper uses ARM TrustZone as part of the trusted execution environment for hosting security services.

In summary, both papers share a common goal of enhancing security and use Trusted Execution Environments, specifically ARM TrustZone, to achieve their objectives.

Question: [11:39] What is difference of virtualization tech between two papers? (Papers.OP-TEE.0406.md)

Answer:

The main difference in virtualization technology between the two papers lies in their specific focus and approach to addressing security challenges in their respective domains.

1. "vTZ: Virtualizing ARM TrustZone": This paper is primarily concerned with virtualizing ARM TrustZone, a security extension that provides a secure world or Trusted Execution Environment (TEE) to run security-sensitive code. The proposed solution, vTZ, allows each guest virtual machine (VM) to have a virtualized guest TEE using existing hardware. vTZ separates functionality from protection by maintaining a secure co-running VM to serve as a guest TEE while utilizing hardware TrustZone for strong isolation among guest TEEs and the untrusted hypervisor. vTZ employs a tiny monitor within the physical TrustZone that securely interposes and virtualizes memory mapping and world switching.

2. "Cloud & Edge Trusted Virtualized Infrastructure Manager (VIM) - Security and Trust in OpenStack": This paper focuses on enhancing security in virtualized edge infrastructure in 5G mobile networks. The authors propose incorporating trust into the virtualized edge infrastructure by developing security services for platform authentication and integrity, hosted inside a Trusted Execution Environment (TEE). The paper does not specifically focus on virtualizing a particular technology but rather aims to improve security in a virtualized infrastructure.

In summary, the first paper targets virtualizing ARM TrustZone to enable secure sharing among multiple VMs, while the second paper addresses security challenges in virtualized edge infrastructure in 5G networks by incorporating trust and

providing security services hosted inside a TEE.

Hypervisor `(230407.Arm.Hypervisor.txt)`

Question: `[16:07]` What is hypervisor? `(OP-TEE.0327.md)`

Answer:

A hypervisor, also known as a virtual machine monitor (VMM), is a type of software that enables the creation and management of virtual machines (VMs) on a physical host machine. The hypervisor is responsible for abstracting the underlying hardware resources of the host and allocating them to the virtual machines, allowing multiple VMs to run simultaneously on a single host.

There are two main types of hypervisors:

1. Type 1 hypervisor (Bare-metal): This type of hypervisor runs directly on the host's hardware without the need for an underlying operating system. It provides strong isolation between VMs and efficiently manages hardware resources. Examples of Type 1 hypervisors include VMware ESXi, Microsoft Hyper-V, and Xen.

2. Type 2 hypervisor (Hosted): This type of hypervisor runs on top of an existing operating system, which acts as the host. The VMs are created and managed within the host OS. Type 2 hypervisors typically have higher overhead and lower performance compared to Type 1 hypervisors, but they can be easier to set up and manage. Examples of Type 2 hypervisors include VMware Workstation, Oracle VirtualBox, and Parallels Desktop.

Hypervisors enable efficient use of computing resources, making it possible to run multiple operating systems and applications on the same physical hardware, improving resource utilization, and enabling greater flexibility in managing workloads. They are widely used in data centers, cloud computing environments, and for virtualization in general.

Question: `[20:58]` Is hypervisor a software or a hardware? `(TrustZone.0329.md)`

Answer:

A hypervisor is a software component that enables virtualization, allowing multiple operating systems or environments to run on a single physical host. Hypervisors can be classified into two types:

1. Type-1 hypervisors (bare-metal hypervisors): These hypervisors run directly on the hardware, providing better performance and more efficient resource management. Examples include VMware ESXi, Microsoft Hyper-V, and Xen.

2. Type-2 hypervisors (hosted hypervisors): These hypervisors run on top of a host operating system, sharing the system's resources with the guest operating systems. Examples include VMware Workstation, Oracle VirtualBox, and Parallels Desktop.

In summary, hypervisors are software components that enable virtualization and are not hardware themselves. However, some modern processors have hardware-assisted virtualization features that can improve the performance and security of hypervisors.

2020 HETEE `(230406.Papers.OP-TEE.txt)`

Question: `[16:33]` Title: Enabling Rack-scale Confidential Computing using Heterogeneous Trusted Execution Environment `(Papers.OP-TEE.0406.md)`

Answer:

Paper Summary:

The paper titled "Enabling Rack-scale Confidential Computing using Heterogeneous Trusted Execution Environment" addresses the limitations of today's Trusted Execution Environments (TEEs) in supporting large-scale confidential computing involving high-throughput accelerators like GPUs, FPGAs, and TPUs. Existing solutions that extend CPU-like enclaves to GPUs require chip-level changes, may introduce new security risks due to side-channel leaks, and are constrained by CPU TEE resources.

The authors present HETEE, a novel Heterogeneous TEE design that supports large-scale compute or data-intensive (CDI) computing without requiring chip-level changes. HETEE is a device for centralized management of all computing units (e.g., GPUs and other accelerators) in a server rack. It works with today's data centers and clouds, leveraging resource pooling technologies to dynamically compartmentalize computing tasks, enforce strong isolation, and reduce Trusted Computing Base (TCB) through hardware support.

HETEE uses PCIe ExpressFabric to allocate accelerators to server nodes for non-sensitive CDI tasks and moves them back into a secure enclave when confidential computing is required. A thin TCB stack for security management runs on a security controller (SC), while other software components run on integrated microservers that operate enclaves. Enclaves are physically isolated through hardware and verified by the SC at their inception, with microservers and computing units restored to a secure state upon termination.

The authors implemented HETEE on a real hardware system and evaluated its performance using popular neural network inference and training tasks. The evaluation results show that HETEE can easily support real-world scale CDI tasks, with a maximum throughput overhead of 2.17% for inference and 0.95% for training on ResNet152.

2020 SoK TrustZone `(230406.Papers.OP–TEE.txt)`

Question: `[16:55]` Title: SoK: Understanding the Prevailing Security Vulnerabilities in TrustZone-assisted TEE Systems `(Papers.OP–TEE.0406.md)`

Answer:

Paper Summary:

The paper titled "SoK: Understanding the Prevailing Security Vulnerabilities in TrustZone-assisted TEE Systems" investigates the security vulnerabilities and limitations that affect existing TrustZone-assisted Trusted Execution Environment (TEE) systems, the challenges in building them correctly, and potential contributions from the research community to address these issues. TEEs, which are used to protect security-critical applications and OS components, are often assumed to be highly secure. However, various successful attacks on TEEs have been reported in recent years, revealing security flaws in TEE systems.

The authors conduct a security analysis of popular TrustZone-assisted TEE systems, targeting Cortex-A processors, developed by Qualcomm, Trustonic, Huawei, Nvidia, and Linaro. They study publicly documented exploits and vulnerabilities and reverse-engineer the TEE firmware to identify critical vulnerabilities across existing systems. The findings raise legitimate concerns about the security of commercial TEE implementations.

This paper provides valuable insights into the types of vulnerabilities that affect TrustZone-assisted TEE systems and highlights the need for robust security measures to protect TEEs from potential attacks. The research contributes to a better understanding of TEE security and helps inform future improvements in TEE design and implementation.

2020 RusTEE `(230406.Papers.OP–TEE.txt)`

Question: `[17:27]` Title: RusTEE: Developing Memory-Safe ARMTrustZone Applications `(Papers.OP–TEE.0406.md)`

Answer:

Paper Summary:

The paper titled "RusTEE: Developing Memory-Safe ARM TrustZone Applications" addresses the issue of vulnerabilities in trusted applications running in ARM TrustZone's Trusted Execution Environment (TEE). While TrustZone provides isolation to protect trusted applications from malicious software in the untrusted rich execution environment (REE), it cannot prevent vulnerabilities within trusted applications themselves. Many trusted applications are written in memory-unsafe languages like C, leading to reported memory corruption vulnerabilities.

The authors propose RusTEE, a trusted application mechanism that uses Rust, a memory-safe programming language, to enhance the security of trusted applications. They address several challenges in adopting Rust for mobile TEEs, as Rust does not currently support TrustZone-assisted TEE systems. The paper presents the following contributions:

1. Extending the Rust compiler: The authors extend the existing Rust compiler to provide support for TrustZone-assisted

TEE systems.

2. Implementing comprehensive security mechanisms: They resolve two security issues of trusted applications, namely, securely invoking high-privileged system services and securely communicating with untrusted REE.

3. Developing a prototype: The authors implement a prototype of RusTEE as an SDK for trusted applications, supporting both emulators and real hardware devices.

Experimental results show that RusTEE can compile applications with close-to-C performance on the evaluated platforms, demonstrating the potential of using Rust to enhance the security of trusted applications in ARM TrustZone TEE systems.

2020 Bao `(230406.Papers.OP-TEE.txt)`

Question: `[20:36]` Paper: Bao: A Lightweight Static Partitioning Hypervisor for Modern Multi-Core Embedded Systems `(Papers.OP-TEE.0406.md)`

Answer:

Paper Summary:

The paper titled "Bao: A Lightweight Static Partitioning Hypervisor for Modern Multi-Core Embedded Systems" addresses the need for a virtualization solution in increasingly complex and mixed-criticality modern embedded systems. Current hypervisors like KVM and Xen were not designed for embedded constraints and requirements. The static partitioning architecture used by Jailhouse seems to address embedded concerns, but it still relies on Linux to boot and manage its virtual machines (VMs).

The authors present Bao, a minimal, standalone, and clean-slate implementation of the static partitioning architecture for Armv8 and RISC-V platforms. Bao is a lightweight hypervisor designed specifically for modern multi-core embedded systems, offering strong spatial and temporal isolation. The main contributions of the paper include:

1. A minimal, standalone implementation: Bao is a clean-slate implementation of the static partitioning architecture, removing the dependency on Linux for booting and managing VMs.

2. Support for Armv8 and RISC-V platforms: Bao is designed to work with both Armv8 and RISC-V platforms, making it suitable for a wide range of embedded systems.

3. Low virtualization overhead: Preliminary results regarding size, boot, performance, and interrupt latency demonstrate that Bao incurs minimal virtualization overhead.

The authors plan to make Bao publicly available, aiming to engage both industry and academia in improving Bao's safety, security, and real-time guarantees. This paper presents a novel approach to virtualization in embedded systems, addressing the unique constraints and requirements of these systems while minimizing overhead.

Question: `[22:26]` What is static partitioning? `(Papers.OP-TEE.0406.md)`

Answer:

Static partitioning is a method of resource allocation in which system resources, such as memory, CPU time, and I/O devices, are divided into fixed, non-overlapping partitions. Each partition is assigned to a specific application, task, or virtual machine. The allocation of resources is determined at design time or during system initialization, and it remains fixed during the runtime of the system.

Static partitioning is often used in real-time and safety-critical systems, where strong spatial and temporal isolation between tasks is required to ensure predictable system behavior and to prevent interference between tasks. This isolation helps in maintaining the performance, safety, and security of the system.

In the context of hypervisors, static partitioning refers to creating fixed, isolated partitions for virtual machines (VMs) or operating system instances. Each partition is allocated a fixed set of resources, such as CPU cores, memory, and I/O devices. This approach contrasts with dynamic partitioning, where resource allocation can be adjusted during runtime based on the changing needs of the system. Static partitioning in hypervisors can help achieve better isolation, predictability, and reduced overhead compared to dynamic partitioning.

2019 Research on TrustZone `(230406.Papers.OP-TEE.txt)`

Question: [20:39] Paper: Research on ARM TrustZone (`Papers.OP-TEE.0406.md`)

Answer:

Paper Summary:

The paper titled "Research on ARM TrustZone" discusses ARM TrustZone, a hardware-based security feature providing a high-privilege and isolated execution environment known as the Trusted Execution Environment (TEE). TrustZone technology was proposed in 2002, but it gained widespread adoption in 2009 when Apple used it to protect Touch ID in the iPhone 5s. TrustZone ensures that user fingerprint data remains secure even if the iOS is compromised.

In 2017, Google mandated TEE for all Android devices with a fingerprint scanner. Today, almost all mobile phones and tablets have TEE deployed. ARM has integrated TrustZone into ARM64 and ARMv8-M to support a broader range of platforms, including servers and IoT devices.

The paper provides valuable insights into the ARM TrustZone technology, discussing its importance in providing a secure and isolated execution environment. It highlights the adoption of TrustZone by major tech companies like Apple and Google and emphasizes its relevance in the growing field of servers and IoT devices.

Question: [21:15] The below context comes from "Other Platforms: IoT, Drone, Cloud" of the paper "Research on ARM TrustZone" (`Papers.OP-TEE.0406.md`)

Answer:

In this section, the authors discuss how ARM TrustZone technology is applied to different types of platforms, such as IoT devices, drones, and cloud computing. They provide examples of various projects and research that utilize TrustZone to improve the security of these platforms:

1. Ditio: This project aims to enhance the security of IoT devices by recording sensor activity logs using TrustZone and virtualization support. Ditio captures access to sensor registers without modifying the Rich Execution Environment (REE) operating system. This generates a complete log of sensor activities that can be audited and checked for compliance with specific policies. Ditio also offers a tool for easier log analysis.

2. PROTC: This project utilizes TrustZone on drone platforms to protect peripherals. PROTC deploys a monitor in the secure world to enforce a secure access control policy for some peripherals on the drone, ensuring that only authorized applications can access specific peripherals.

3. vTZ: This research addresses the limitation of TrustZone, which can only provide one secure world, meaning multiple virtual machines (VMs) on a server must share the single secure world. vTZ virtualizes TrustZone to offer each VM an isolated secure world. With vTZ, software in a VM can use the Secure Monitor Call (SMC) instruction to switch between the REE and a virtual TEE. This preserves all TrustZone features after virtualization. The implementation requires an REE hypervisor but does not need to trust it.

These examples showcase the versatility of TrustZone technology and its potential for improving security on various platforms beyond mobile phones.

Question: [21:17] In the context, vTZ run in normal world. Why does not vTZ run in normal world need to trust REE hypervisor? (`Papers.OP-TEE.0406.md`)

Answer:

vTZ is designed to virtualize ARM TrustZone to offer each virtual machine (VM) an isolated secure world. Although vTZ runs in the normal world, it does not need to trust the REE hypervisor because of the hardware-level isolation provided by ARM TrustZone.

TrustZone ensures a strong isolation between the secure world (trusted execution environment) and the normal world (rich execution environment). When vTZ is implemented, it leverages the security features provided by TrustZone to maintain a separation between the virtual TEEs and the REE hypervisor.

As a result, even though vTZ operates in the normal world and interacts with the REE hypervisor, the secure world components (virtual TEEs) remain protected from potential attacks or vulnerabilities in the REE hypervisor. This isolation allows vTZ to run securely without having to trust the REE hypervisor.

2019 TEEv `(230406.Papers.OP-TEE.txt)`

Question: `[20:41]` Paper: TEEv: Virtualizing Trusted Execution Environmentson Mobile Platforms (`Papers.OP-TEE.0406.md`)

Answer:

Paper Summary:

The paper titled "TEEv: Virtualizing Trusted Execution Environments on Mobile Platforms" discusses the challenges of transitioning from vendor-controlled, single-purpose Trusted Execution Environments (TEEs) to open TEEs that host Trusted Applications (TAs) from multiple sources with independent tasks. This transition can lead to an enlarged attack surface and a lack of trust or isolation among TAs and the TEE.

To address these challenges, the authors propose TEEv, a TEE virtualization architecture that supports multiple isolated, restricted TEE instances (vTEEs) running concurrently. TEEv relies on a tiny hypervisor called TEE-visor, allowing TEE instances from different vendors to run in isolation on the same smartphone and host their own TAs. This ensures that a compromised vTEE cannot affect its peers or the Rich Execution Environment (REE), and TAs no longer have to run in untrusted or unsuitable TEEs.

The paper presents an implementation of TEEv on a development board and a real smartphone, running multiple commercial TEE instances from different vendors with minimal porting effort. The evaluation results show that TEEv can isolate vTEEs and defend against all known attacks on TEE with only mild performance overhead.

Question: `[21:54]` Please compare TEEv and vTZ and also summarize them in a table. (`Papers.OP-TEE.0406.md`)

Answer:

TEEv and vTZ are two different approaches to virtualize Trusted Execution Environments (TEEs) on mobile platforms. Here is a comparison and summary of both in the form of a table:

Aspect	TEEv	vTZ
Objective	Virtualize TEEs to support multiple isolated, restricted TEE instances	Virtualize TrustZone to offer each VM an isolated secure world
Architecture	Relying on a tiny hypervisor (TEE-visor)	Requires an REE hypervisor but does not need to trust it
Implementation	Runs multiple commercial TEE instances from different vendors	Uses SMC instruction to switch between REE and a virtual TEE
Security	Isolates vTEEs and defends against known attacks on TEE	Preserves all TrustZone features after virtualization
Performance	Mild performance overhead	Not explicitly mentioned

TEEv is a TEE virtualization architecture that supports multiple isolated, restricted TEE instances (vTEEs) running concurrently. It relies on a tiny hypervisor called the TEE-visor, allowing TEE instances from different vendors to run in isolation on the same smartphone and host their own Trusted Applications (TAs). TEEv can isolate vTEEs and defend against all known attacks on TEEs with only mild performance overhead.

vTZ aims to virtualize TrustZone to provide each VM an isolated secure world. This approach requires an REE hypervisor but does not need to trust it, as the hardware-level isolation provided by TrustZone is leveraged to maintain separation between the virtual TEEs and the REE hypervisor. With vTZ, software in a VM can use the SMC instruction to switch between the REE and a virtual TEE, just as it does in a non-virtualized environment, preserving all the features of TrustZone after virtualization.