CE4062/CZ4062
Computer Security

Lecture 6: Operating System Protection

Tianwei Zhang

Outline

- **Confinement**
- **Reference Monitor**
- **System Integrity Protection and Verification**
- **Trusted Execution Environment**

Outline

- **Confinement**
- **Reference Monitor**
- **System Integrity Protection and Verification**
- **Trusted Execution Environment**

Confinement

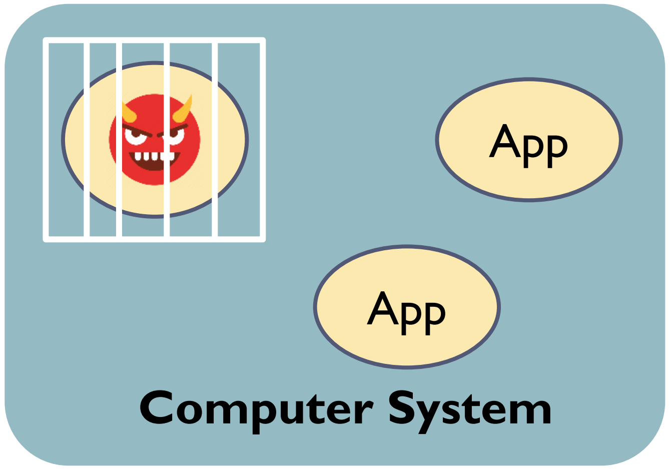
An important security strategy in OS protection

- When some component (e.g., application) in the system is compromised or malicious, we need to prevent it from harming the rest of system.
- Confinement: restrict its impact on other components.

Application scenario

- Cut off the propagation chain.
- Malware testing and analysis

Can be implemented at different levels



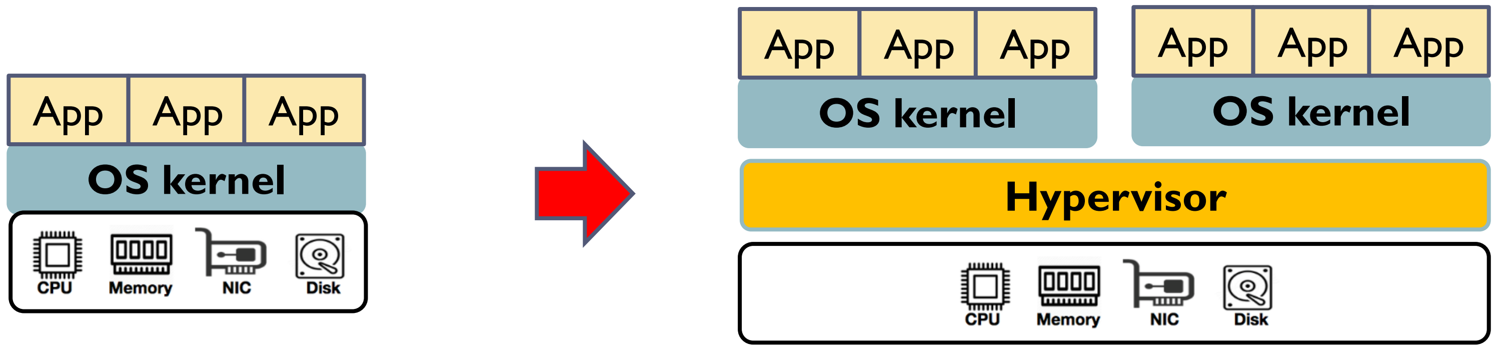
Virtual Machine – OS Level Isolation

Virtualization: the fundamental technology for cloud computing

- Different operating systems (virtual machines) run on the same machine
- Each virtual machine has an independent OS, logically isolated from others

Technical support

- Software layer: hypervisor or virtual machine monitor (VMM) for visualizing and managing the underlying resources, and enforcing the isolation
- Hardware layer: hardware virtualization extensions (Intel VT-x, AMD-V) for accelerating virtualization and improving performance



Visualization Technology

Pros

- Provide better efficiency in the usage of the physical resources as different VMs can share the same resources.
- Provide support for multiple distinct operating systems and associated applications on one physical system
- High isolation across different VMs so malware inside one VM will not affect other VMs or the hypervisor

Cons

- The introduction of hypervisor can incur larger attack surface
  - The hypervisor has big code base, and inevitably brings more software bugs
  - The hypervisor has higher privilege than the OS kernel. If it is compromised, then the attacker can take control of the entire system more easily.

Virtual Machine for Malware Analysis

Malware analysis: deploying the malware in the OS and observes its malicious behaviors.

- Deploying the malware in the native OS has the potential to compromise the entire system
- Virtual machine: an ideal environment for testing malware
  - The malware cannot cause damages outside of the VM
  - The malware's behavior can be observed from the hypervisor/host OS

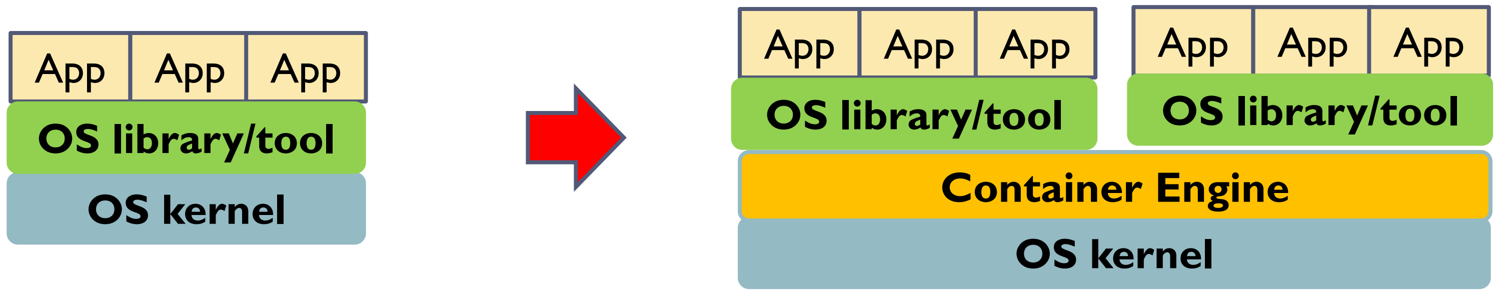
Hypervisor detection

- A smart malware can detect that it is running inside a VM, not the actual environment, and then behave like normal applications
  - Virtual machine is a simulated environment
  - Hypervisor introduces access latency variance.
  - Hypervisor shares the TLB with all the OSes.

Container – Process Level Isolation

A standard unit of software

- Package the code and all the dependencies (e.g., library) into one unit
- Each container is a lightweight, standalone, executable software package that includes everything needed to run the application
  - Code, system tools and libraries, configurations.
- Different applications in different containers are isolated
- A Container Engine is introduced to manage different containers
  - Docker: a popular container platform



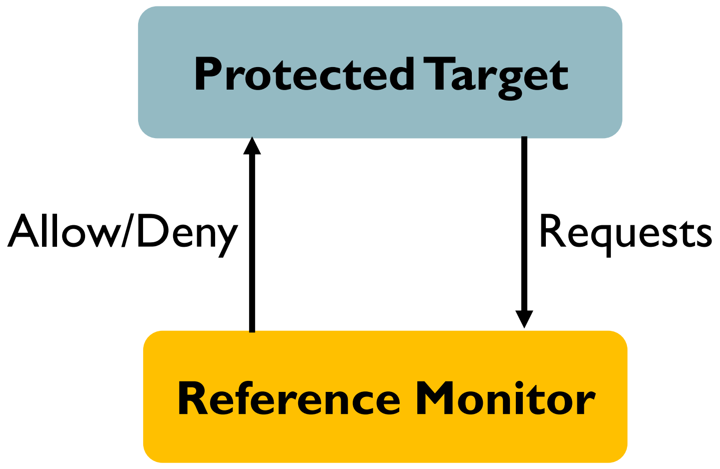
Outline

- Confinement
- **Reference Monitor**
- System Integrity Protection and Verification
- Trusted Execution Environment

Reference Monitor (RM)

A security mechanism that monitors and mediates requests from the protected targets at runtime

- Enforce some security policies, e.g., confinement
- When any security policy is violated, RM can deny the request.



Different Types of RMs

OS-based RM

- The OS monitors the system calls issued by the application.

Software-based RM

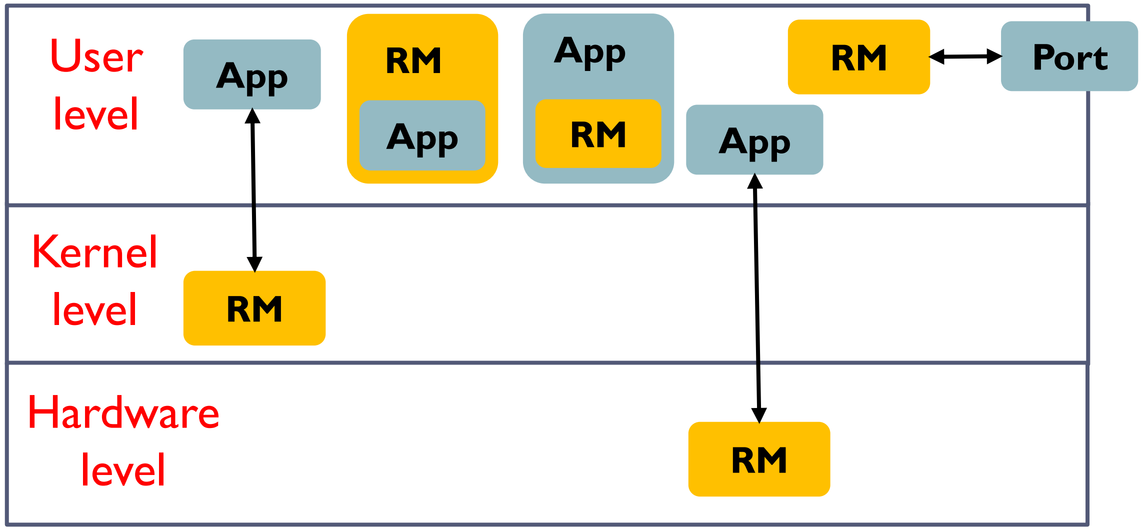
- Interpreter
- In-line RM: integrate RM into the program or rewriting compiled program

Hardware-based RM

- Memory address access

Network-based RM

- Firewalls



Requirements of RM

Function requirement

- The reference validation mechanism must always be invoked.
- RM is able to observe all the requests
- RM is able to deny the malicious requests

Security requirement

- The reference validation mechanism must be tamper-proof.

Assurance requirement

- The reference validation mechanism must be small enough to be analyzed and tested.

Hardware-based Reference Monitors

OS is the arbitrator of access requests. It is itself an object of access control.

Hardware-based RMs are introduced to monitor the OS

- Memory access: each process has a dedicated virtual memory address. The hardware is responsible for checking each memory access
- Distinct user and kernel modes:
  - At any time, CPU is in one mode.
  - Privileged instructions can only be issued in kernel mode.
  - When the user mode wants to execute these functions, it switches to kernel mode (e.g., system call, interrupt) for execution. When finished, it goes back to the user mode. – controlled invocation.

Outline

- Confinement
- Reference Monitor
- **System Integrity Protection and Verification**
- Trusted Execution Environment

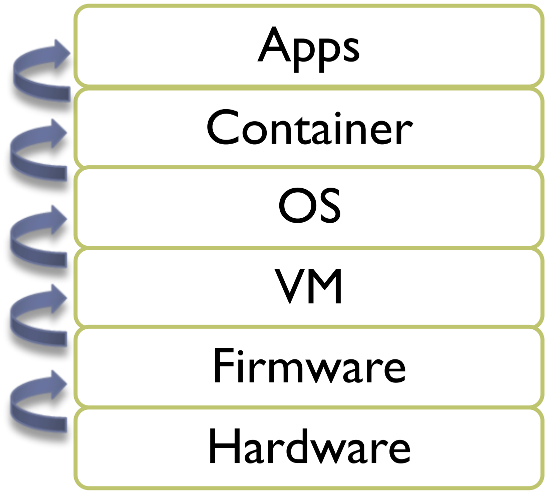
Computer System

A hierarchic view: layered system

- Different scenarios may have different layers
- Lower layers have higher privileges and can protect higher layers.
- Lower layers need to be better protected

Chains of trust: establish verified systems from bottom to top

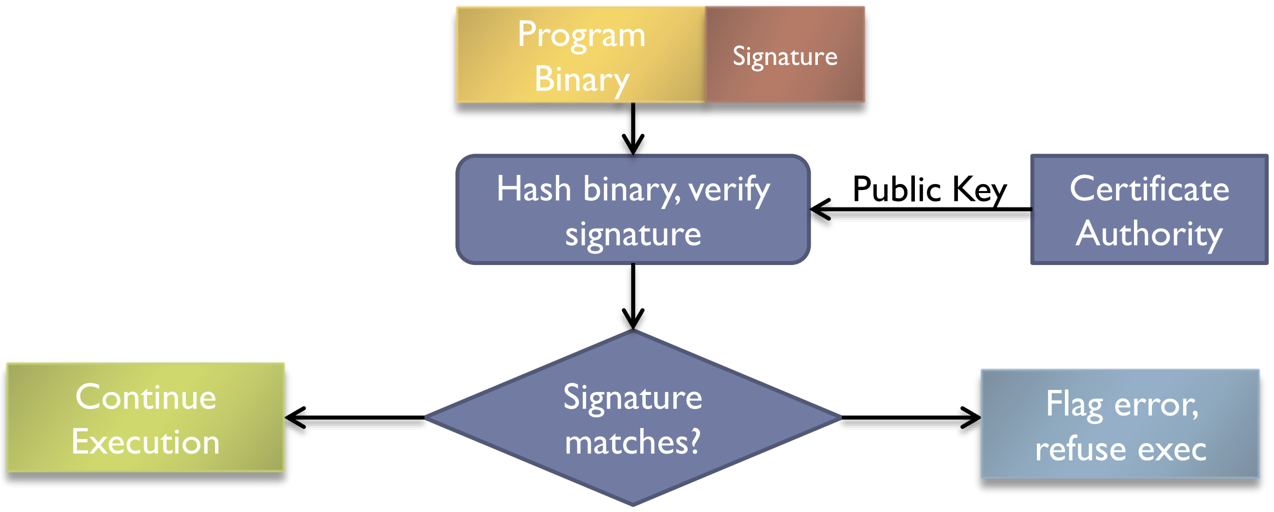
- The bottom layers validate the integrity of the top layers
- If the verification passes, then it is safe to launch it.
- Each layer is vulnerable to attack from below if the lower layers are not secured appropriately



Integrity Verification

Only execute code signed by an entity we trust

- Load the bootloader in the firmware
- Reads and verifies the kernel
- Only loads kernel if the signature is verified
- Similarly, kernel can run only verified applications



Root of Trust

Software is not always trusted

- Can be tampered with at any phase easily
- There can still be vulnerabilities when using software to protect software.

Hardware is more trusted

- After the chip is fabricated, it is hard for the attacker to modify it. The integrity of hardware can be guaranteed.
- It is also very hard for the attacker to peek into the chip and steal the secret (e.g., encryption key). The confidentiality of hardware can also be guaranteed.
- Hardware is a perfect component as the start of the chain of trust. It is regarded as secure for initializing the integrity verification.

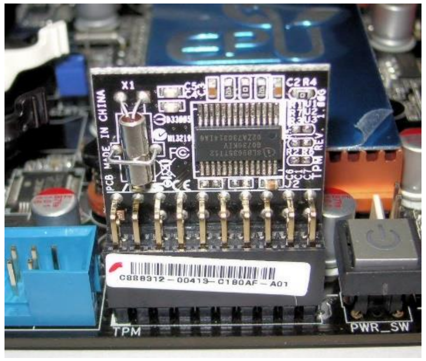
Trusted Platform Module

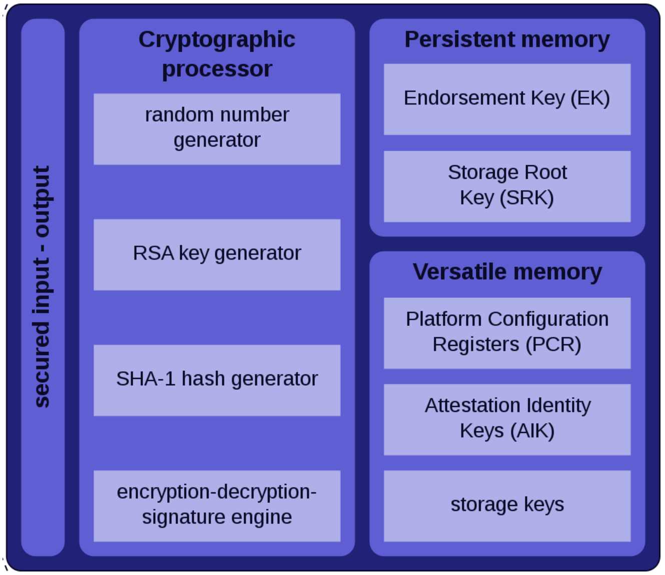
A chip integrated into the platform

- A separated co-processor
- Its state cannot be compromised by malicious host system software

Inside the chip

- Random number and key generators
- Crypto execution engine
- Different types of crypto keys.





Development and Implementation

Designed by Trusted Computing Group (TCG)

- A non-profit industry organization, which develops hardware and software standards. Funded by many IT companies, e.g., IBM, Intel, AMD, Microsoft…
- First version: TPM 1.1b, released in 2003.
- An improved version: TPM 1.2, developed around 2005-2009
  - Equipped in PCs in 2006 and in servers in 2008
  - Standardized by ISO and IEC in 2009
- An upgraded version: TPM 2.0, released on 9 April 2014.

Application of TPM

- Intel Trusted Execution Technology (TXT)
- Microsoft Next-Generation Secure Computing Base (NGSCB)
- Windows 11 requires TPM 2.0 as a minimal system requirement
- Linux kernel starts to support TPM 2.0 since version 3.20
- Google includes TPMs in Chromebooks as part of their security model
- Google Compute Engine offers virtualized TPM in the cloud services.
- VMware, Xen, KVM all support virtualized TPM.
- ……

Security Functionality – Platform Integrity

Building a chain of trust

- Establish a secure boot process from the TPM, and continue until the OS has fully booted and applications are running.

Application

- Digital right management
- Enforcement of software license, e.g., Microsoft Office and Outlook
- Prevention of cheating in online games

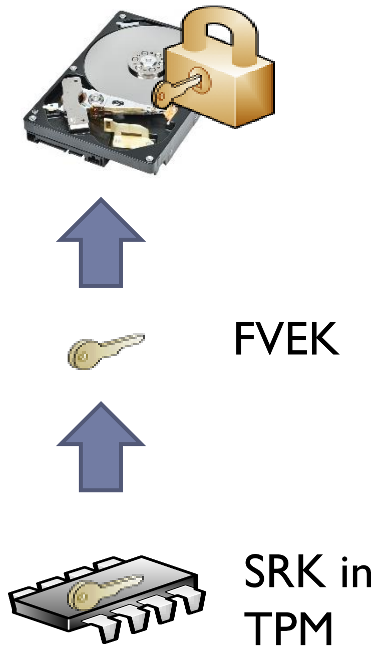
Security Functionality – Data Encryption

Full disk encryption

- Encrypt the data with the key in the hardware.
- It is difficult for any attacker to steal the key, which never leaves the chip.
- TPM can also provide platform authentication before data encryption

Application: Windows BitLocker

- Disk data are encrypted with the encryption key FVEK.
- FVEK is further encrypted with the Storage Root Key (SRK) in TPM.
- When decrypting the data, BitLocker first asks TPM to verify the platform integrity. Then it asks TPM to decrypt FVEK with SRK. After that, BitLocker can use FVEK to decrypt the data
- With this process, data can only be decrypted on the correct platform with the correct software launched.



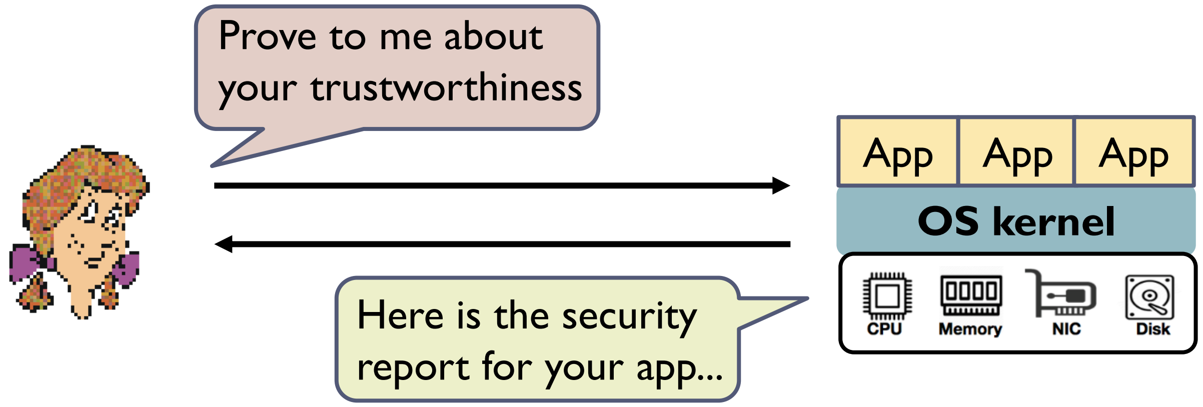
Security Functionality – Remote Attestation

Scenario

- Alice launches an application on a remote server. How can she know whether this application is executing securely on a trusted platform?

Remote attestation

- A remote platform provides unforgeable evidence about the security of its software to a client.
- A common strategy to prove the OS and applications running on the platform are intact and trustworthy.


Prove to me about your trustworthiness
Here is the security report for your app...
App
App
App
OS kernel
CPU
Memory
NIC
Disk

Major Components for Remote Attestation

Integrity measurement architecture:

- Must be able to provide reliable and trustworthy security report
- Solution: TPM provides the platform integrity measurement service, and the measurement cannot be compromised by any malicious OS or apps.

Remote attestation protocol

- The attestation report must be transmitted to the client without being modified by any attacker in the OS, apps or network.
- Cryptographic protocols can be used to secure the report.
- Solution: TPM is equipped with crypto keys that can be used to encrypt and sign the messages, making the integrity measurement unforgeable.

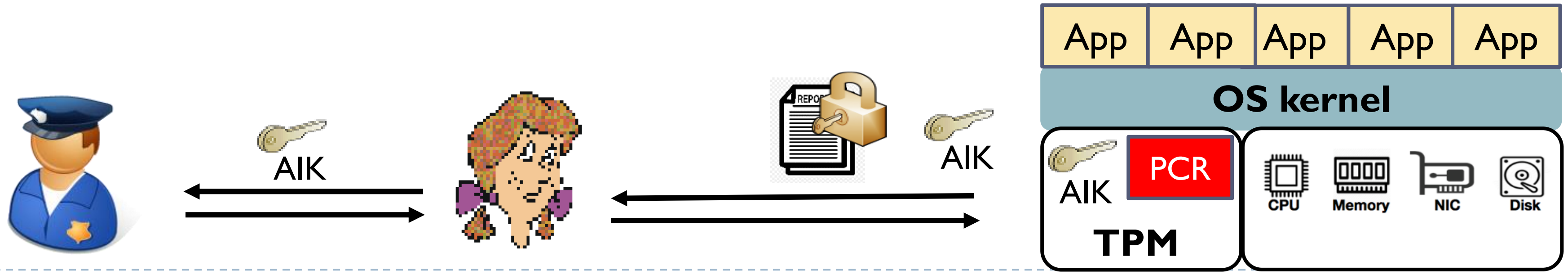
Remote Attestation with TPM

Integrity measurement architecture:

- TPM measures the hash values of the code of each loaded software, as the integrity report.
- The hash values are stored in the Platform Configuration Registers (PCR) in TPM.

Remote attestation protocol

- TPM generates an Attestation Identity Key (AIK), to sign the hash values.
- The hash values together with the AIK will be sent to Alice.
- A trusted third party, Privacy Certification Authority (PCA) is called to verify this AIK is indeed from the correct platform.
- Alice uses this AIK to verify that received hash values are authentic.
- By checking the hash values, Alice knows if the loaded software is correct or not



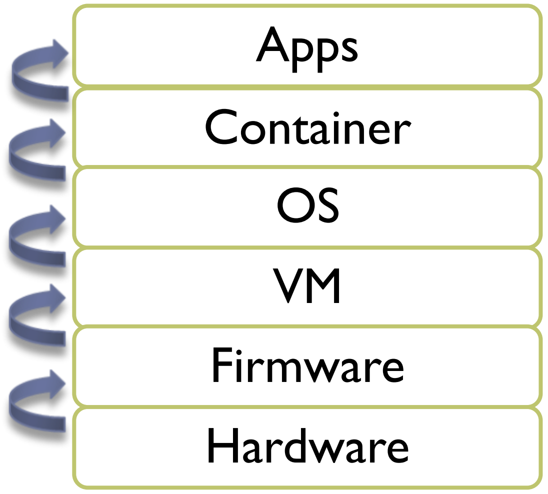
Outline

- Confinement
- Reference Monitor
- System Integrity Protection and Verification
- **Trusted Execution Environment**

Untrusted Privileged Software

Chains of Trust

- The secure execution of an application relies on the security of underlying software and hardware layers.
- Easy to ensure the security of hardware
- More challenging to ensure the security of privileged software (OS, hypervisor)



Security of privileged software

- TPM can guarantee the privileged software is intact at loading time, but not the security at runtime.
- Privileged software usually has very large code base, which inevitably contains lots of vulnerabilities.
- Once the privileged software is compromised, the attacker can do anything to any apps running on it.

SW	Line of codes
Linux Kernel 5.12	28.8M
Windows 10	50M
VMWare	6M
Xen	0.9M

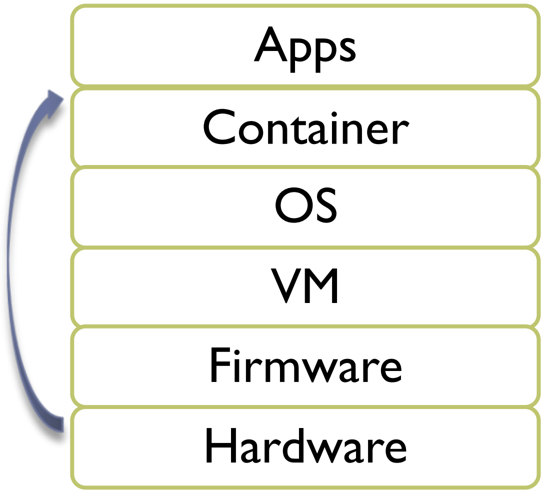
Commercial software typically has 20 to 30 bugs for every 1k lines of code

How to protect the applications with the untrusted privileged OS or hypervisor?

Trusted Execution Environment (TEE)

Solution: TEE

- Introduce new hardware to protect the apps from untrusted OS or hypervisor.
- The OS or hypervisor can support the execution of apps, but cannot compromise them
  - Cannot read the data of the apps.
  - Cannot change the code of the apps.



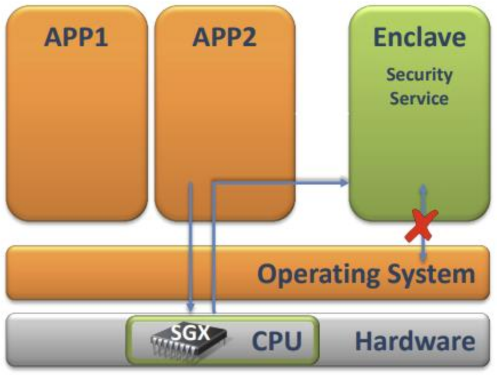
Different processors have been developed to support TEE

- Intel Software Guard Extensions (SGX)
- AMD Secure Encrypted Virtualization (SEV)
- ARM TrustZone

Intel Software Guard Extensions (SGX)

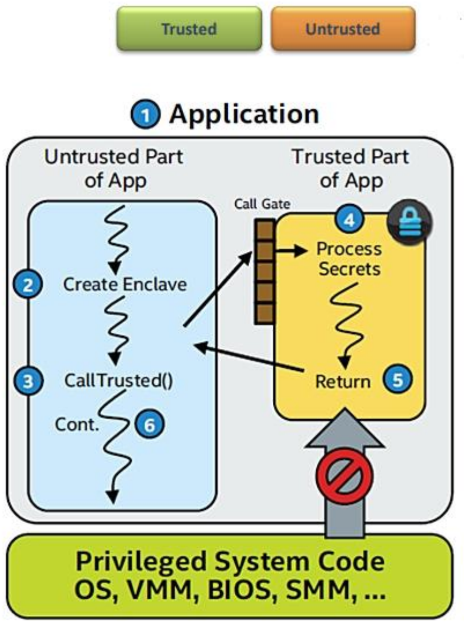
Enclave

- An isolated and protected region for the code and data of an application
- Data in the enclave are encrypted by the processor when they are stored in the memory
  - Only the processor can access the data. Attempts from other apps or OS will be forbidden and invoke exception



Application execution with enclave.

- An application is divided into a trusted part and an untrusted part.
- The untrusted part creates an enclave, and puts the trusted part into it.
- When the trusted code needs to be executed, the processor enters the enclave. In this mode, only the trusted code can be executed and access the data.
- After the code is completed, the processor exits from the enclave.



Intel Software Guard Extensions (SGX)

Attestation

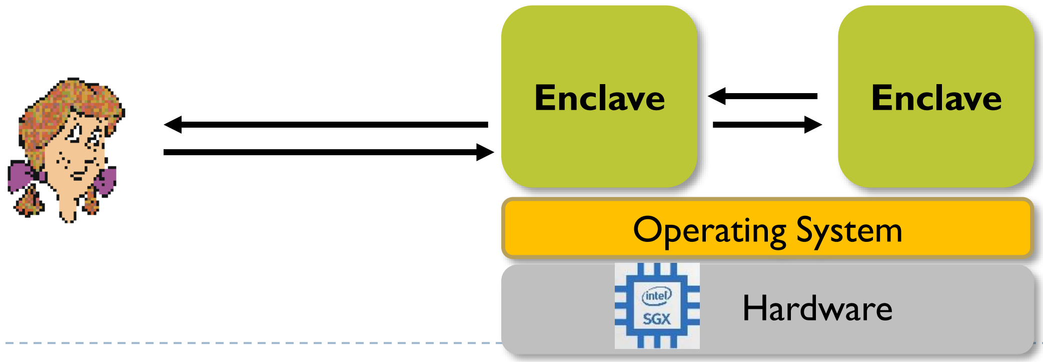
- Similar as TPM, SGX also provides the attestation service, which enables another party to attest what code is running inside the enclave.
- Integrity measurement architecture: enclave measurement of the code, data, stack, heap, security flags, location of each page…
  - Attestation protocol: attestation key.

Remote attestation

- A remote client attests the integrity of the code in the enclave.

Local attestation

- In some scenarios, multiple enclaves collaborate on the same task, exchanging data at runtime.
- The two exchanging enclaves have to prove to each other that they can be trusted.



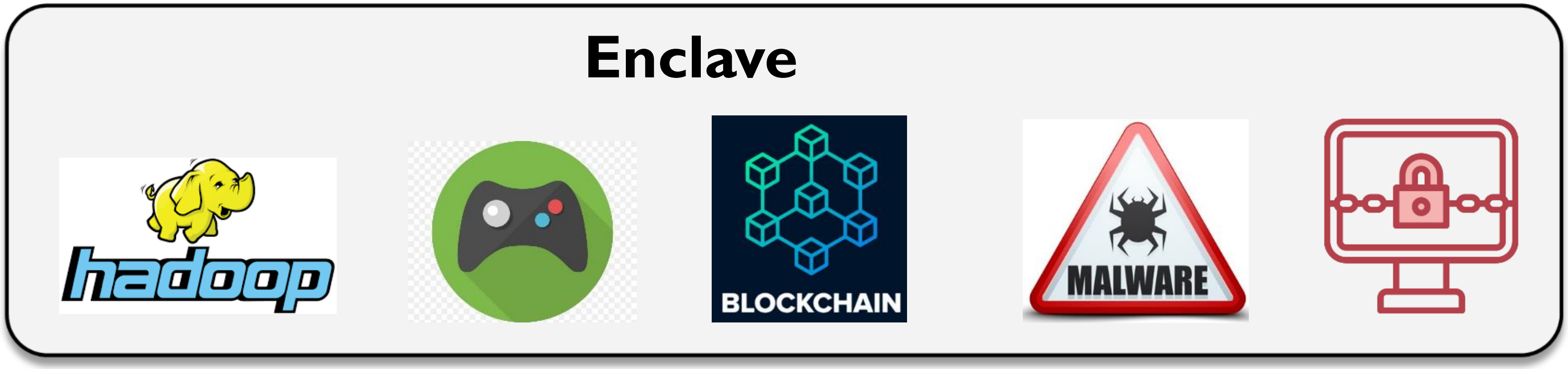
Application of SGX: Double-edged Sword

Positive usage

- Cloud computing: you do not need to trust the cloud provider
- Digital right management
- Cryptocurrency

Negative usage: adversaries leverage SGX to hide malicious activities for stealthier attacks

- Malware
- Ransomware



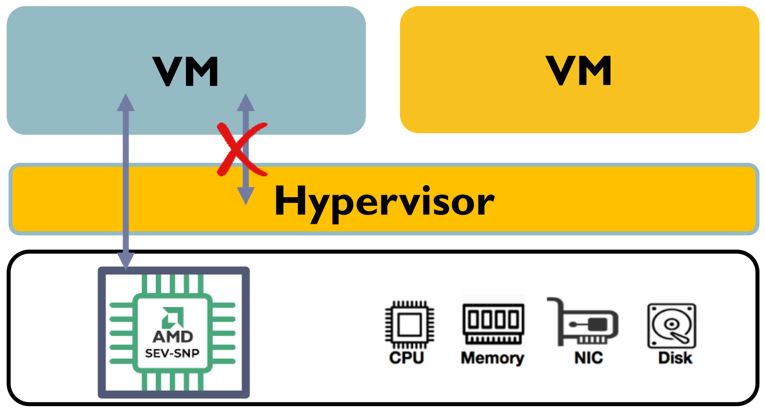
AMD Secure Encrypted Virtualization (SEV)

TEE for the virtualized platform

- In the conventional platform, hypervisor has privilege to control all the VMs. A compromised hypervisor can do anything to VMs running on top of it.
- AMD SEV: a hardware extension to protect the VMs against the hypervisor and other VMs.

Mechanism

- The processor encrypts the data (memory page, registers, configurations) of the guest VMs, so the hypervisor is not allowed to access the data.



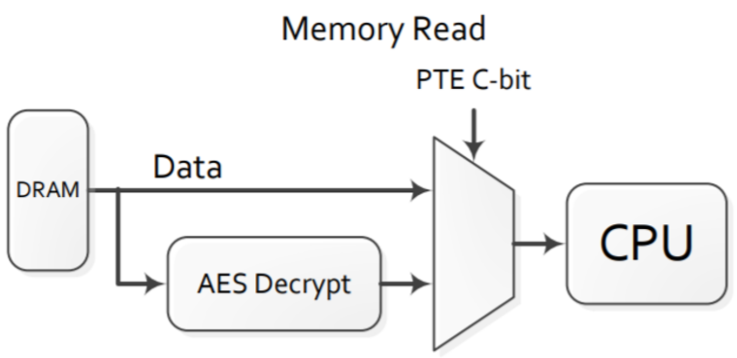
AMD Secure Memory Encryption (SME)

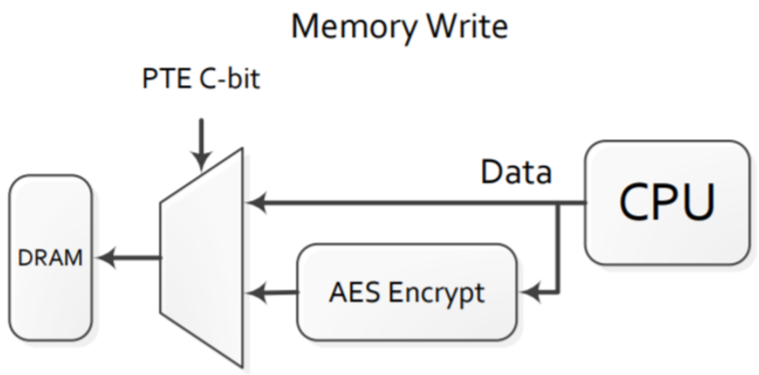
Virtual memory encryption is realized by SME

- An AMD architectural capability for main memory encryption
- Performed via dedicated hardware in the memory controllers
- Use AES engine to encrypt data and control with C-bit in Page Table Entry

C-bit

- Locate at physical address bit 47
- Set this bit to 1 to indicate this page is encrypted.
- Allow users to encrypt full memory of the VM, or selected memory pages

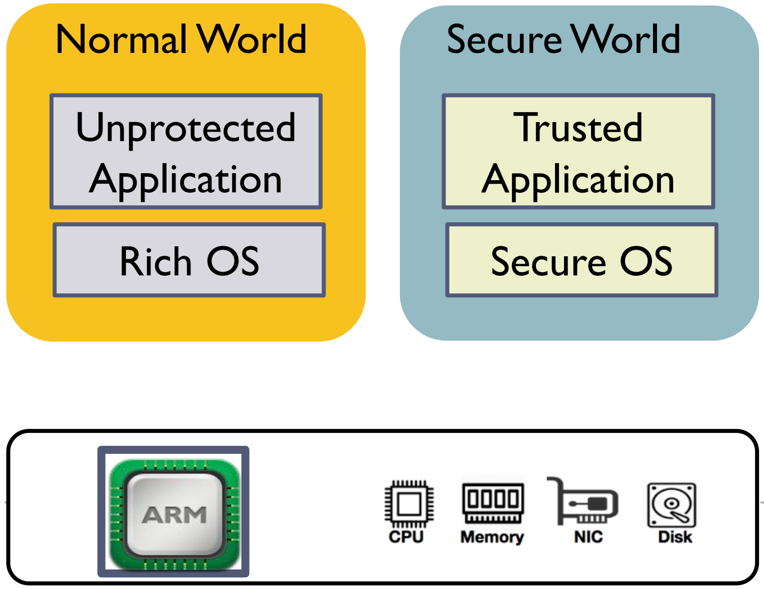




ARM TrustZone

The first commercial TEE processor

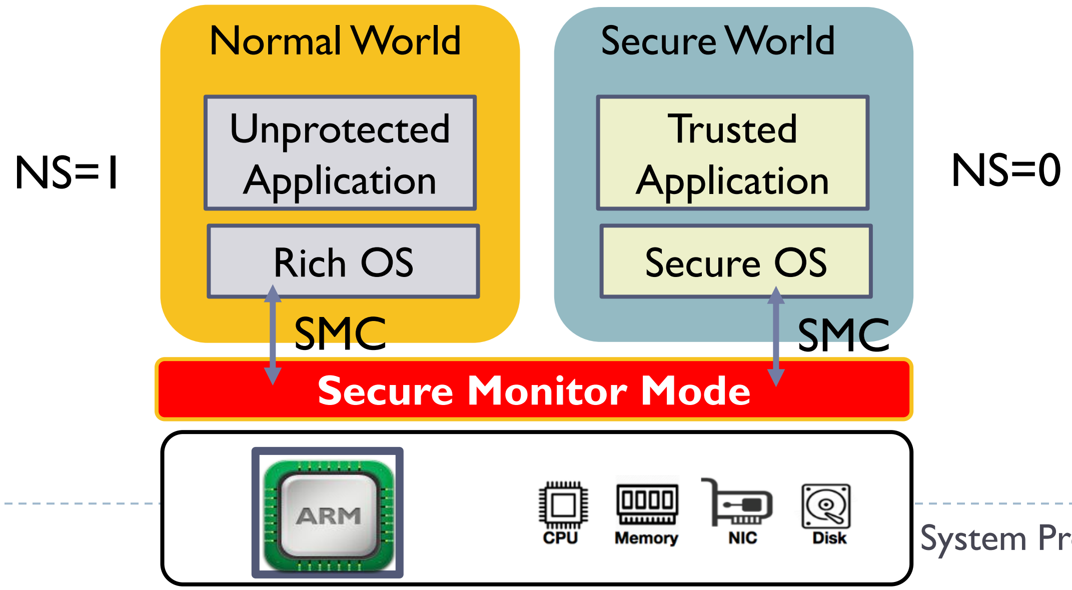
- Create two environments that can run simultaneously on the same processor. Each world has an independent OS
- Normal world: runs the normal unprotected applications and a rich OS. They have restricted access to the hardware resources in the secure world
- Secure world: runs the sensitive protected applications and a smaller secure OS, isolating them from the untrusted world. They have full access to the hardware resources in the normal world.



ARM TrustZone

Context switch

- The Non-secure bit in the Secure Configuration Register is used to determine which world the processor is currently running.
- A third privilege mode: secure monitor, in addition to user and kernel.
- When the processor wants to switch the world, it first issues a special instruction Secure Monitor Call (SMC) to enter the secure monitor mode. Then it performs some cleaning works and enter the other world.


Normal World
Unprotected Application
Rich OS
SMC
NS=1
Secure World
Trusted Application
Secure OS
SMC
NS=0
Secure Monitor Mode
ARM
CPU
Memory
NIC
Disk