OPERATING SYSTEM

BOOK & THESIS CHALLENGE

Pitchuka Venkata Ramana
03190444
EEIC 4th Year

MELTDOWN VULNERABILITY

(A)MELTDOWN VULNERABILITY ①INTRODUCTION

- Hardware vulnerability : Intel x86 microprocessors and some ARM-based microprocessors.
- Rogue process read memory
- January 2018; 3 years ago

(A)MELTDOWN VULNERABILITY ②INITIAL WORKAROUNDS

- Software fixes: Slows 5~30%
- March 2018: Intel redesign begins
- October 2018: Intel adds mitigations

(A)MELTDOWN VULNERABILITY ③MECHANISM

Background – modern CPU design

Features are particularly relevant to Meltdown:

- Virtual (paged) memory
- Privilege levels
- Instruction pipelining
- CPU cache

(A)MELTDOWN VULNERABILITY ③MECHANISM

Meltdown Exploits

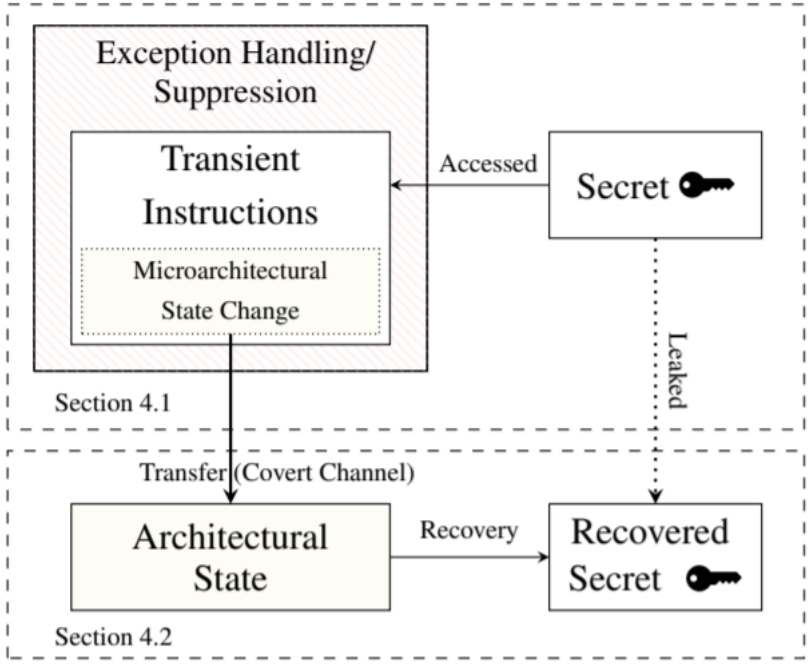
- Ordinarily, the mechanisms are considered secure.
- Exploits the mechanisms to bypass the CPU's fundamental privilege controls
- Accesses privileged and sensitive data from the O.S.

- CPU attempts to execute an instruction referencing a memory operand (address: Base+A)
- **Caching of the data at Base+A**, completed as a side effect of the memory access before the privilege check
- Act of caching -> leak of information
- Process executes a timing attack referencing memory operands directly

(A)MELTDOWN VULNERABILITY ③MECHANISM

How MELTDOWN leaks kernel space information.

- Step 1 The content of an attacker-chosen memory location, which is inaccessible to the attacker, is loaded into a register.
- Step 2 A transient instruction accesses a cache line based on the secret content of the register.
- Step 3 The attacker uses Flush+Reload to determine the accessed cache line and hence the secret stored at the chosen memory location.



The Meltdown attack uses exception handling or suppression, e.g., TSX, to run a series of transient instructions.

(A)MELTDOWN VULNERABILITY ④IMPACT

- CPU
- OS
- Virtual Machine
- Embedded Device, IoT

- **Impacts Cloud Platforms** like AWS and GCP
- Affects Para virtualized Machines and Containers

(A)MELTDOWN VULNERABILITY
⑤THESIS: READING KERNEL MEMORY FROM USER SPACE

Optimizations and Limitations

- Inherent bias towards 0
- Optimizing the case of 0
- Single-bit transmission.
- Exception Suppression using Intel TSX
- Dealing with KASLR.

https://meltdownattack.com/meltdown.pdf
https://lwn.net/Articles/738975/

(A)MELTDOWN VULNERABILITY
⑤THESIS: READING KERNEL MEMORY FROM USER SPACE

EVALUATION

- Leakage and Environments
  - Linux
  - Linux with KAISER Patch
  - Microsoft Windows
  - Android
  - Containers (Docker, LXC, and OpenVZ)
  - Uncached and Uncacheable Memory

Table 1: Experimental setups.

Environment	CPU Model	Cores
Lab	Celeron G540	2
Lab	Core i5-3230M	2
Lab	Core i5-3320M	2
Lab	Core i7-4790	4
Lab	Core i5-6200U	2
Lab	Core i7-6600U	2
Lab	Core i7-6700K	4
Lab	Core i7-8700K	12
Lab	Xeon E5-1630 v3	8
Cloud	Xeon E5-2676 v3	12
Cloud	Xeon E5-2650 v4	12
Phone	Exynos 8890	8

(A)MELTDOWN VULNERABILITY
⑤THESIS: READING KERNEL MEMORY FROM USER SPACE

- Meltdown Performance
  - Core i7-8700K Reading rate: 582 KB/s Error Rate 0.003%
  - Core i7-6700K Reading rate: 569 KB/s Error Rate 0.002 % | L3 Data Cache RR:12.4 KB/s ER: 0.02 %
  - Xeon E5- 1630 Reading rate: 491 KB/s Error Rate 10.7 % but Slower version (137 KB/s Error Rate->0)

- Limitations on ARM and AMD
  - Successfully leak kernel memory on different Intel CPUs and a Samsung Exynos M1 processor
  - Failed on ARM cores and AMD

(A)MELTDOWN VULNERABILITY
⑤THESIS: READING KERNEL MEMORY FROM USER SPACE

Countermeasures

- Hardware
  - Meltdown bypasses the hardware-enforced isolation of security domains
  - As Meltdown exploits out-of-order execution,
    →disable out-of-order execution completely.
    →however, performance impacts would be devastating

- KAISER
  - Kernel modification to not have the kernel mapped in the user space
  - Intended to prevent side-channel attacks breaking KASLR
  - Due to the design of the x86 architecture, several privileged memory locations are required to be mapped

VIRTUAL MACHINES

(B)VIRTUAL MACHINES
①INTRODUCTION

- Emulation of a computer system
- Based on computer architectures and provide functionality of a physical computer

Broadly 2 types

- System virtual machines
  - Provide a substitute for a real machine.
  - They provide functionality needed to execute entire operating systems.
- Process virtual machines
  - execute computer programs in a platform-independent environment

(B)VIRTUAL MACHINES
①INTRODUCTION

- USES
  - Test applications in a safe, sandboxed environment.
  - Used for server virtualization, enabling teams to consolidate their computing resources and improve efficiency.
- Advantages
  - Run multiple OS on a single computer
  - Windows host can run Linux on VM
- Disadvantages
  - Can result in unstable performance
  - Less efficient and generally run slower

(B)VIRTUAL MACHINES
②VIRTUALIZATION

- Types of Virtualization
  - Hardware Virtualization
  - Software Virtualization
  - Storage Virtualization
  - Network Virtualization
  - Desktop Virtualization
- Container vs virtual machine
  - Similar as they both run isolated applications on a single platform
  - VM make a computer but containers package up just a single app
  - Containers are faster

(B)VIRTUAL MACHINES
②VIRTUAL MACHINES VS EMULATORS (SLIGHTLY OFF-TOPIC)

- The purpose of a virtual machine is to create an isolated environment.
- The purpose of an emulator is to accurately reproduce the behavior of some hardware.

(B)VIRTUAL MACHINES
③THESIS THE ARCHITECTURE OF VIRTUAL MACHINES

- Architected interfaces
  - Instruction set architecture.
  - Application binary interface.
  - Application programming interface.

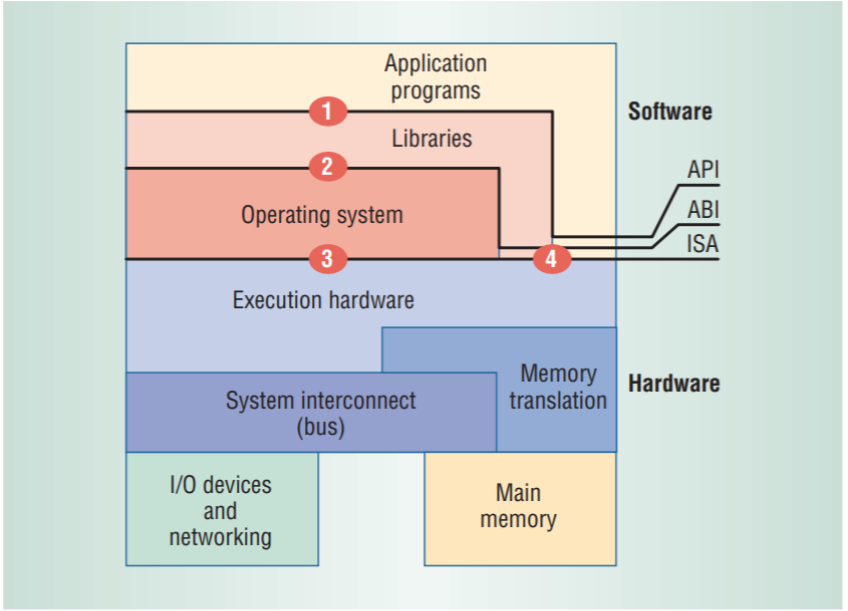


Figure 2. Computer system architecture. Key implementation layers communicate vertically via the instruction set architecture (ISA), application binary interface (ABI), and application programming interface (API).

(B)VIRTUAL MACHINES
③THESIS THE ARCHITECTURE OF VIRTUAL MACHINES

- Process and system VMs
  - Perspective of a process
    - executing a user program, the machine consists of a logical memory address space assigned to the process along with user-level instructions and registers that allow the execution of code belonging to the process
  - Perspective of the operating system and the applications it supports
    - The entire system runs on an underlying machine.

- The virtualizing software in a system VM is typically referred to as the virtual machine monitor (VMM)

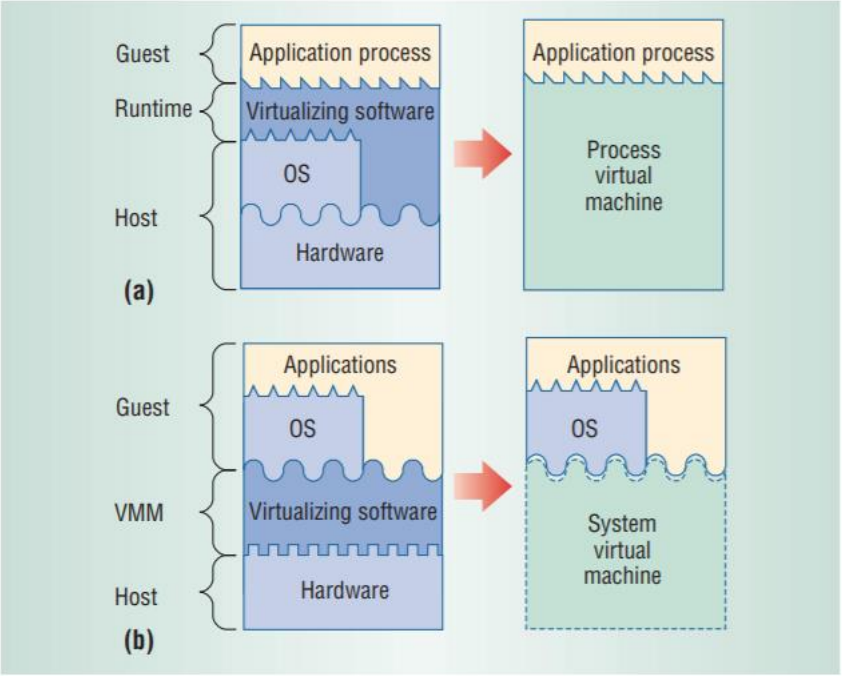


Figure 3. Process and system VMs. (a) In a process VM, virtualizing software translates a set of OS and user-level instructions composing one platform to those of another. (b) In a system VM, virtualizing software translates the ISA used by one hardware platform to that of another.

- Process VM
  - Process VMs provide a virtual ABI or API environment for user applications. In their various implementations, process VMs offer replication, emulation, and optimization.
  - Multiprogrammed systems
  - Emulators and dynamic binary translators
  - Same-ISA binary optimizers
  - High-level-language VMs

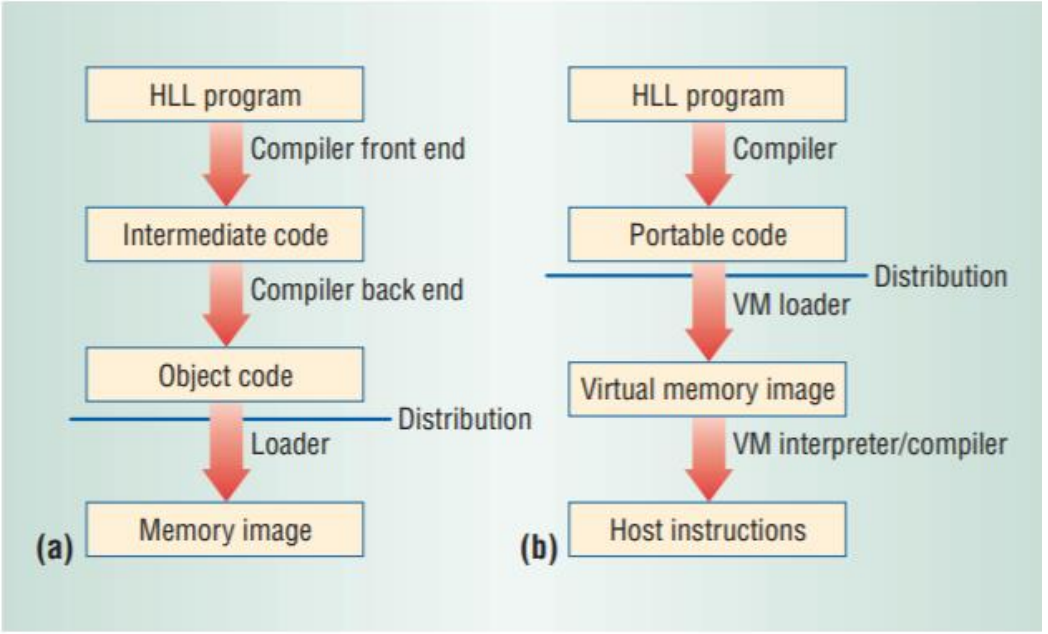


Figure 4. High-level-language environments. (a) Conventional environment where platform-dependent object code is distributed. (b) HLL VM environment where a platform-dependent VM executes portable intermediate code.

(B)VIRTUAL MACHINES
③THESIS THE ARCHITECTURE OF VIRTUAL MACHINES

- **System VM**
  - A system VM provides a complete environment in which an operating system and many processes, possibly belonging to multiple users, can coexist.

  - Classic system VM
  - Hosted VMs
  - Whole-system VMs
  - Multiprocessor virtualization
  - Codesigned VMs

(B)VIRTUAL MACHINES
③THESIS THE ARCHITECTURE OF VIRTUAL MACHINES

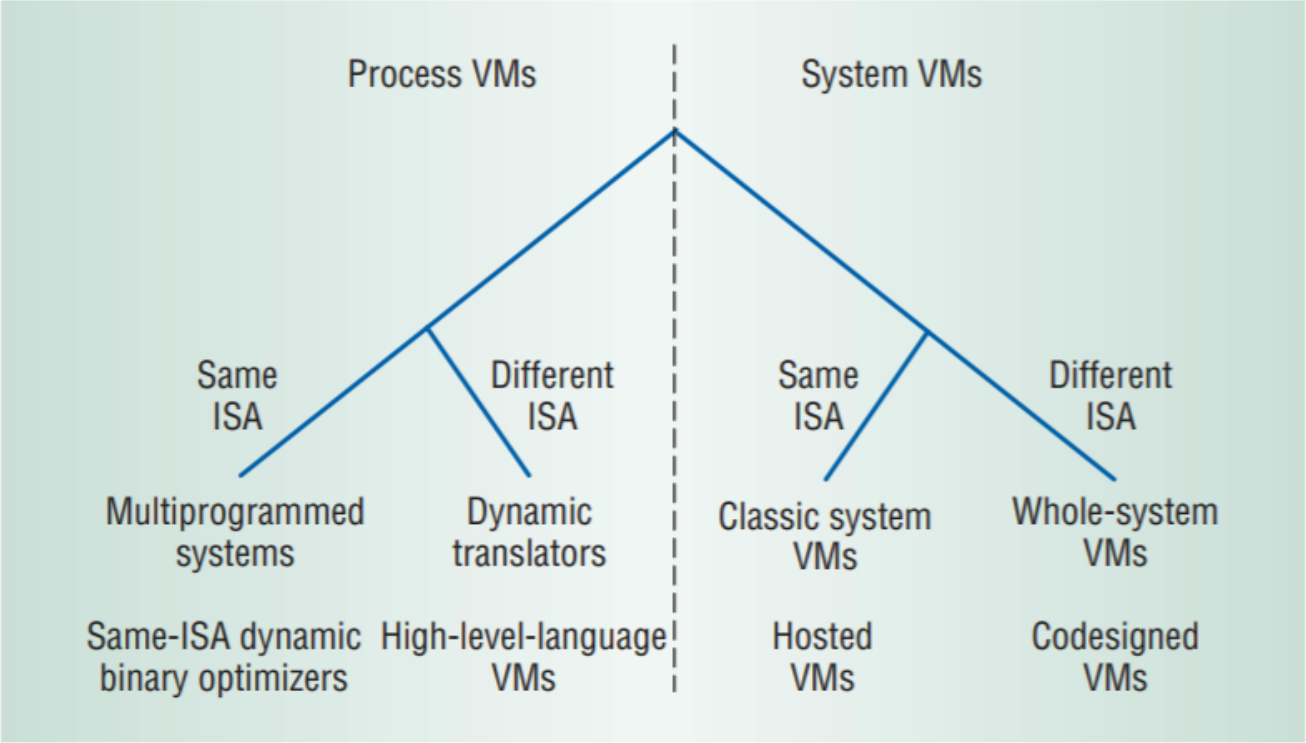


Figure 5. Virtual machine taxonomy. Within the general categories of process and system VMs, ISA simulation is the major basis of differentiation.

ANNOTATIONS/SOURCES

- MELTDOWN
  - https://meltdownattack.com/meltdown.pdf
  - https://lwn.net/Articles/738975/
  - https://lkml.org/lkml/2017/11/22/956
  - https://gruss.cc/files/kaiser.pdf
- VM
  - https://minds.wisconsin.edu/bitstream/handle/1793/11154/file_1.pdf?sequence=1&origin=publicationDetail
  - https://zoo.cs.yale.edu/classes/cs422/2014/bib/goldberg74architecture.pdf
  - http://mvapich.cse.ohio-state.edu/static/media/publications/abstract/huangwei-ics06.pdf