Virtually Impossible
The Reality of Virtualization Security

Gal Diskin / Chief Research Officer / Cyvera LTD.
/WhoAmI ?

- Chief Research Officer @ Cvyera LTD
- Formerly Security Evaluation Architect of the Software & Services Group @ Intel®
- Before that – Entrepreneur, Consultant, IDF
- Always a security “enthusiast” 😊

– Personal focus areas:
  - DBI, Fuzzing & Automated exploitation
  - Exploitation techniques & Mitigations
  - Vehicles & Traffic systems
  - Embedded systems
ThankZ & GreetZ

• My wife
  – For tolerating me doing security research

• Everyone at Cyvera, special thanks to:
  – Harel Baris for help with the presentation design
  – Gal Badishi and Ariel Cohen for reviewing

• All Intel security people
  – Especially my old team
What I will talk about today
Beyond why virtualization is virtually impossible to secure...

- Hardware assisted virtualization
- SW stacks and different virtualization approaches and related weaknesses
- The complexity in memory management and related weaknesses
- Computer platforms internals and related weaknesses
- Finally, I will present a small taxonomy of attacks against virtualization
- Special bonus – potential VM escape ;-)

 полуночник

 полуночник

 полуночник
What is Virtualization?

• In the context of this talk replacing the CPU and computer platform with a virtual environment

• A bit of history:
  – Turing’s universal computing machine
  – Popek and Goldberg virtualization requirements
Terminology

- A Virtual Machine Manager (VMM) is the software virtualizing privileged instructions and hardware
- A Virtual Machine (VM) is a software stack running under a VMM
- A Guest OS is the operating system of a VM
- A Host OS is the operating system controlling the VMM
- Root operation is when you execute inside a VMM
What is “secure” virtualization?

Security Goals:
- Prevent modification of VMM and host OS by guests
- Prevent guest OS from modifying another guest
- Prevent guest from subverting hardware or firmware*
- Prevent guest from stealing data from other guest OS / host OS / VMM*
- Prevent DOS by guest OS or getting unfair share of resources relative to other guests*
- Keep guest OS secure – don’t harm normal OS defenses*

* Depending on the hypervisor design, might be a non-goal
SOFTWARE STACKS
Piling different pieces of software
Software Stack
Type 1 Hypervisor

- Virtual Machine Manager (Hypervisor / VMM)
- Guest Operating System 1
- Guest Operating System 2
- Process 1
- Process 2
- Process 3
- Hardware

System Calls
VM Exits / Entries
Instruction Set

- Cyvera Cyber Defense Solutions
Software Stack
Type 2 Hypervisor

- Host Operating System
- VMM
- Guest Operating System
- Process 1
- Process 2
- Process 3

- Hardware
- System Calls
- VM Exits / Entries
- Virtualized System Calls

Instruction Set

Types of Software Stack:
- Type 2 Hypervisor
ISA emulation challenges

- A VMM needs to emulate every instruction or event it registers on.
- A VMM **must** register to a certain set of instructions and x86 events known as the “fixed-1 exits”
  - e.g.: CPUID, GETSEC, INVD, XSETBV and various VT ISA
- ISA emulation challenges
  - Specification
  - Corner cases
  - Deciding if the guest has the right privilege from root operation is hard
    - Confused deputy situation…
Software Stack
SMM with VMM

RSM
#SMI

SMM

Process 1
Process 2
Process 3

Guest OS 2

Guest OS 1

Virtual Machine Manager (Hypervisor / VMM)

Hardware

System Calls
VM Exits / Entries
Instruction Set

Cyvera Cyber Defense Solutions
Software Stack
SMM Transfer Monitor (STM)

SMM

Process 1
Process 2
Process 3

Guest OS 2
Guest OS 1

Virtual Machine Manager (Hypervisor / VMM)

Hardware

System Calls
VM Exits / Entries
Instruction Set

Cyvera Cyber Defense Solutions
Micro-VMMs

- Process 1
- Process 2
- Process 3

- Guest Operating System
- Micro VMM
- Hardware

- System Calls
- VM Exits / Entries
- Instruction Set
Section summary

• There are many ways to use hardware virtualization technology:
  – Type I, Type II, Micro-VMMs, …

• Each approach has its own unique challenges:
  – Full HW virtualization: Secure a big implementation of SW emulation for all HW
  – Para-virtualization: Secure the guest OS interface with the host OS
  – All implementations: Emulate ISA correctly and securely
  – Micro-VMMs: Defend from HW subversion

• SMM is too privileged and where are the STMs?
MEMORY

Where did I put that instruction?
Memory is simple, right?
Memory – Address Translations

- Linear addressing
- Guest physical
- System view
- DRAM

Paging translation → EPT translation → MCH mappings

* MCH = Memory Control Hub (MMU)
** Segmentation adds another translation and the cache adds a whole new translation path
Memory – point of view
Memory – MMIO

- Confidential -
Special address ranges
Cache!

Sorry, out of scope for today–there is no end to it once you start discussing cache and security.

Suffice to say that it adds another translation layer and that it is complex and performance oriented.
Section summary

• Memory is complex!
• Attackers with access to MMIO or physical memory addresses can compromise anything on the system
• Access to special address ranges is also dangerous
• EPT can help mitigate some of the problems
  – If you can configure it correctly, if it is available
COMPUTER PLATFORMS

The insides
What is a computer?

- A very complex device internally
- The logical software architecture can be complex
- Every modern computer system is also a complex high speed network of interconnecting hardware components using many communication protocols.
Computer Platform (1)
Computer platform (2)
Device virtualization

- XEN and KVM use a modified QEMU – No vulnerabilities there, right?
VT-d (IOMMU)

- Used for virtualizing chipset components
- DMA remapping
  - Paging for devices
  - Nested translations
- Interrupt remapping
  - Allows directing interrupts coming from hardware

There is a good paper that explains the need for it by Rafal Wojtczuk and Joanna Rutkowska:

- *Following the White Rabbit: Software attacks against Intel® VT-d technology*

- What about older systems where you don’t have VT-d?
Section summary

- Computer hardware is complex!

- Emulating necessary components is hard:
  - Multiple CVEs already found in ACPI and APIC virtualization as well as QEMU

- VT-d helps virtualizing DMA and hardware interrupts
  - If used correctly
ATTACK VECTORS TAXONOMY
Potential and practical ones
Basic Vectors

• **ISA Implementation**
  – Emulating x86 isn’t easy…

• **Performance monitoring**
  – Classic side channels
  – Real Time Instruction Tracing – new feature coming up

• **Old systems**
  – New defenses were introduced with the latest HW

• **New features**
  – Approach to new features (CPU/PCH)
    • Whitelist or Blacklist?
Address Space Attacks

• **IO Address Space**
  - How many IO ports are there in x86?
  - What happens if we configure port overlaps?

• **MMIO Overlaps**
  - As discussed during the presentation

• **Special memory ranges access and overlaps**
  - What happens when a guest can access special ranges?

• **MSR address space**
  - MSRs define system configuration and behavior
Privileged Software

• **Corrupted ACMs**
  – ACMs run in a very high privilege, if you can compromise one…

• **CPU/PCH Firmware(s)**
  – Compromise of the CPU or PCH firmware naturally allows an attacker to control any VM

• **BIOS & SMM**
  – The BIOS is a common component of the platform and controls both configuration and SMM code
Other Interesting Vectors

• **Intentional misconfiguration**
  – It is possible to misconfigure PCIe config space, MSRs or MMIO constants in order to create unexpected situations for the VMM

• **Server platforms (are fun!)**
  – Platforms and CPUs for the server market have special features, all of those usually run in very high privilege

• **Errata**
  – What if there is an Errata in the CPU/PCH behavior we rely on to emulate something - sucks, right? 😃
Bonus: Interesting Errata

• The below Errata appears in the June 2013 revision for 2nd generation core CPUs
• Sounds like an exploitable issue if you can prevent reload of CR3 with 32bit value

**BK124. The Upper 32 Bits of CR3 May be Incorrectly Used With 32-Bit Paging**

**Problem:** When 32-bit paging is in use, the processor should use a page directory located at the 32-bit physical address specified in bits 31:12 of CR3; the upper 32 bits of CR3 should be ignored. Due to this erratum, the processor will use a page directory located at the 64-bit physical address specified in bits 63:12 of CR3.

**Implication:** The processor may use an unexpected page directory or, if EPT (Extended Page Tables) is in use, cause an unexpected EPT violation. This erratum applies only if software enters 64-bit mode, loads CR3 with a 64-bit value, and then returns to 32-bit paging without changing CR3. Intel has not observed this erratum with any commercially available software.

**Workaround:** Software that has executed in 64-bit mode should reload CR3 with a 32-bit value before returning to 32-bit paging.

**Status:** For the stepping(s) affected, see the Summary Tables of Changes.
Summary

- Computer platforms are complex
- There are several approaches to virtualizing HW, each with its own inherent weaknesses
  - Full hardware virtualization: slower and uses SW emulation, therefore prone to SW vulnerabilities
  - Direct hardware access: prone to malicious HW manipulations (micro-VMMs)
- Better defenses are available only with new and sometimes also high end HW
THE END

Contact me at: http://www.cyvera.com/contact
Extended materials: http://cyvera.com/virtually-impossible/
BACKUP
Useful tools and info for people interested in virtualization research

- LOLA by Jeff Forristal (free)
  - Because a Python interface to the HW rocks!
- 8 series PCH manuals
- 2nd generation core Errata
- Intel software developer manuals
  - Volume 3 contains most information about VT
  - Other volumes are also useful to understand what is emulated
- Patience!
  - Hardware debugging, reading long technical manuals
How different virtualization SW works

• VMware Player
  – Emulates 440BX motherboard (15 years old)
  – Monitors PCIe configuration, at least IO ports, to some degree but because of Win95 and Win3.1 compatibility, not security!
Disclaimer

• All product logos and names used in this presentation are the property of their respective owners. I make no claim for ownership on those. I am merely using them as examples of such products.