CYVERA
CYBER DEFENSE SOLUTIONS

# 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 ;-)

# 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:

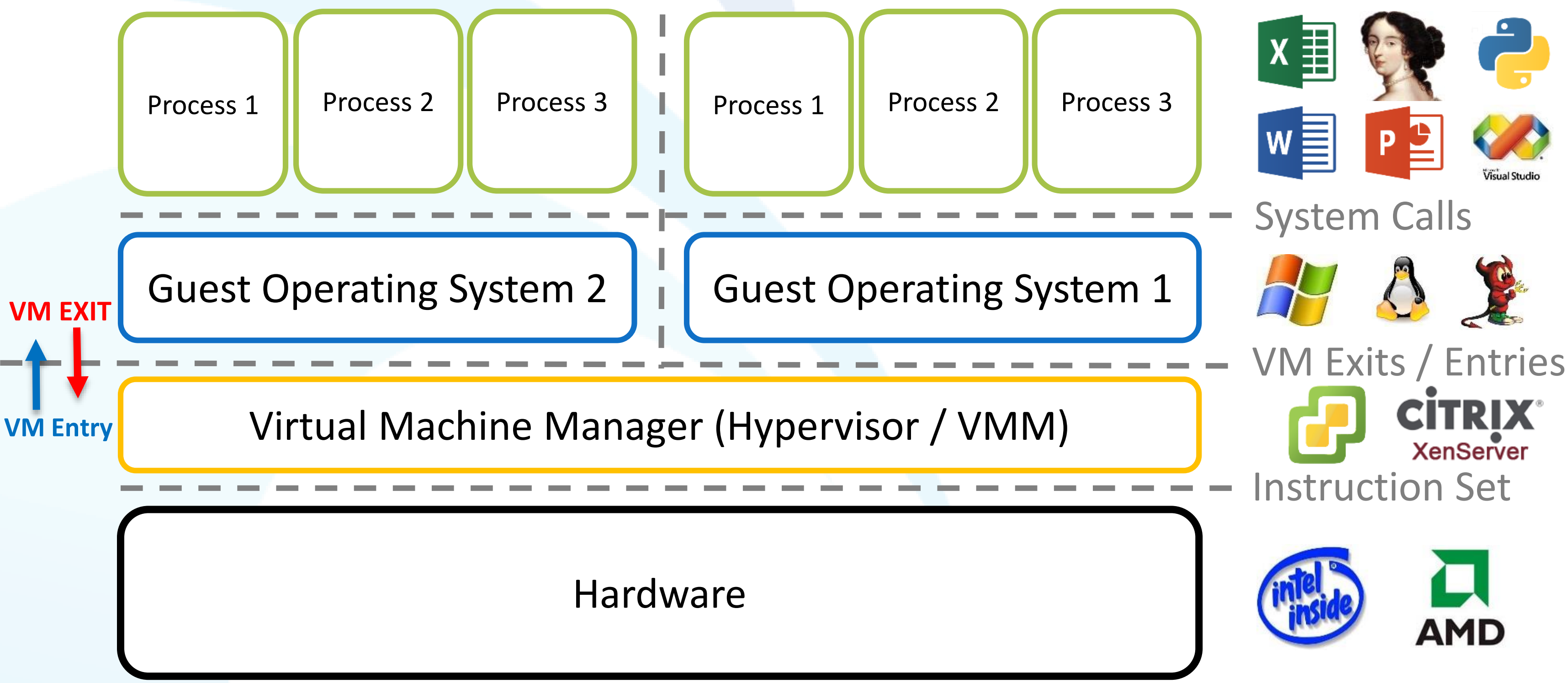
- Prevent modification of VMM and host OS by guests
- Prevent guest OS from modifying another guest
- Prevent guest from subverting hardware or firmware*
- Prevent guest from stealing data from other guest OS / host OS / VMM*
- Prevent DOS by guest OS or getting unfair share of resources relative to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

* Depending on the hypervisor design, might be a non-goal

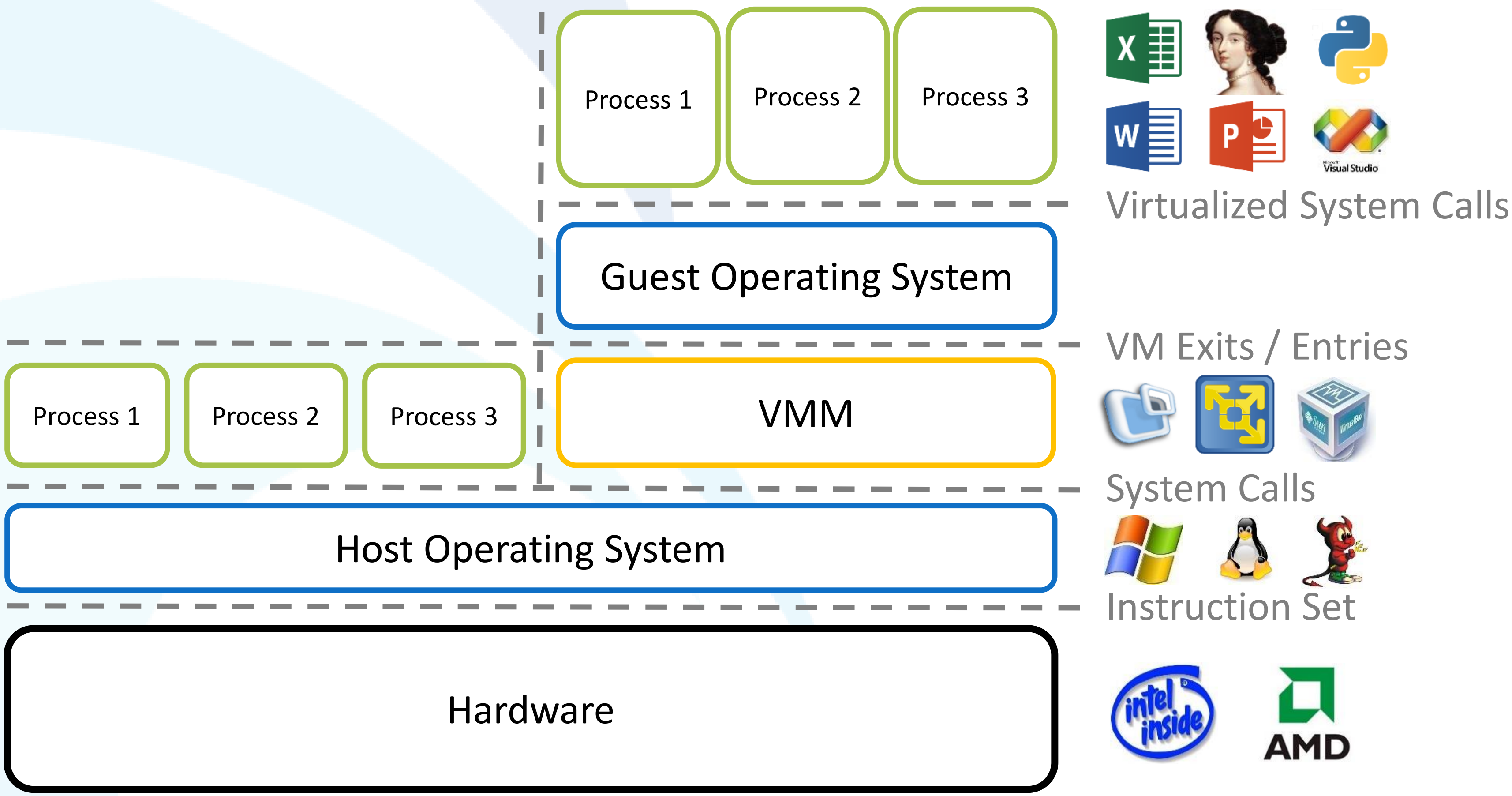
# SOFTWARE STACKS

## Piling different pieces of software

# Software Stack
# Type 1 Hypervisor



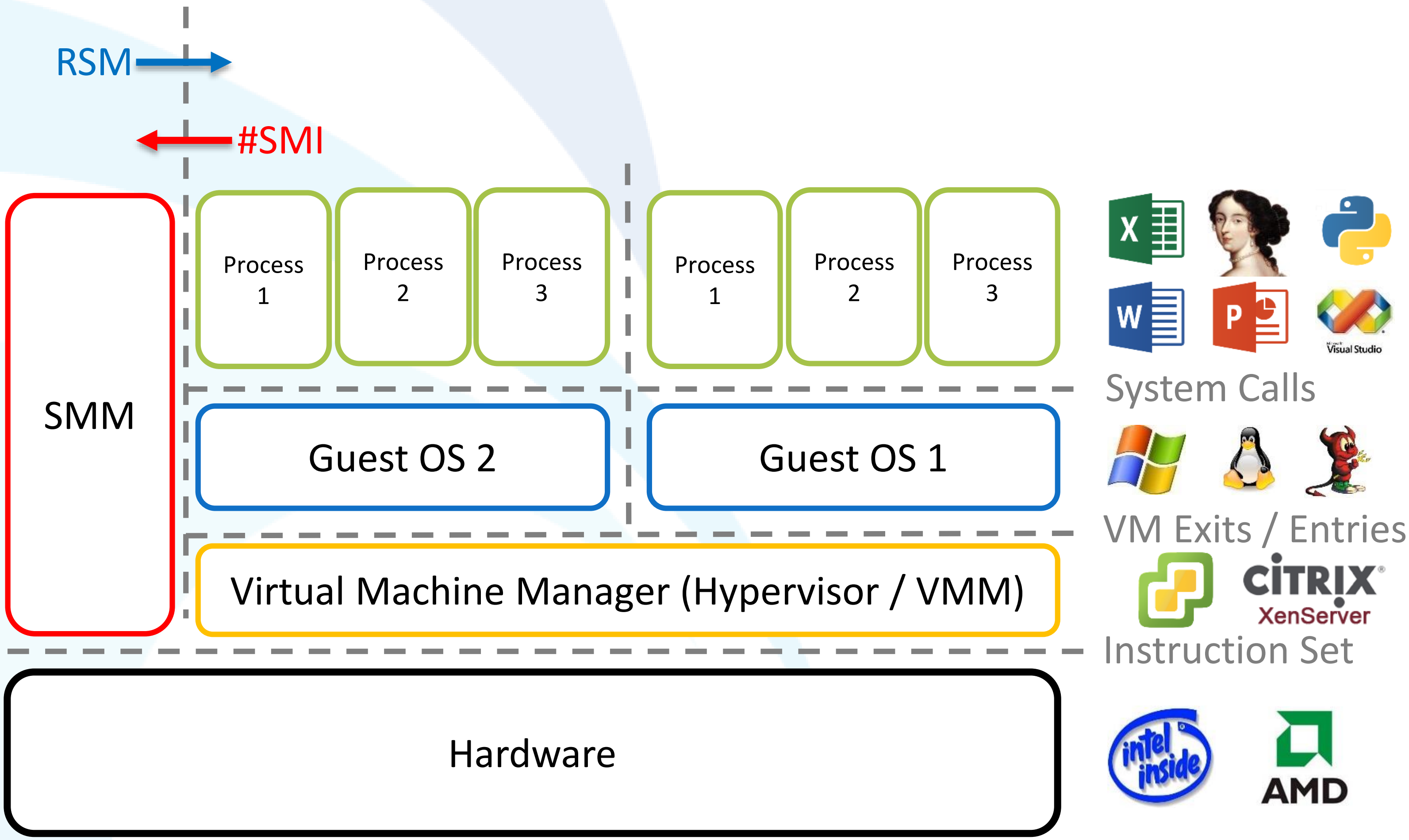
# Software Stack
# Type 2 Hypervisor



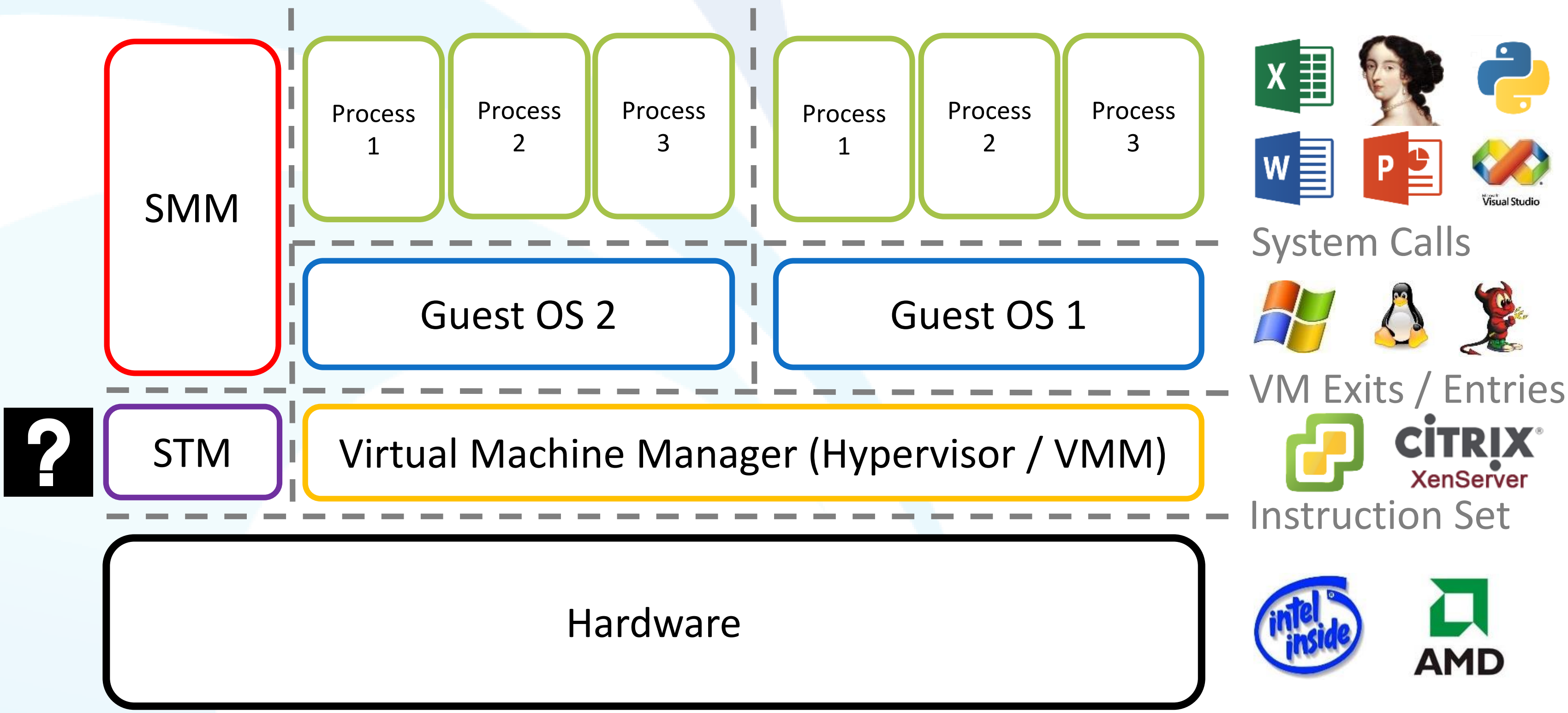
# ISA emulation challenges

- 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…

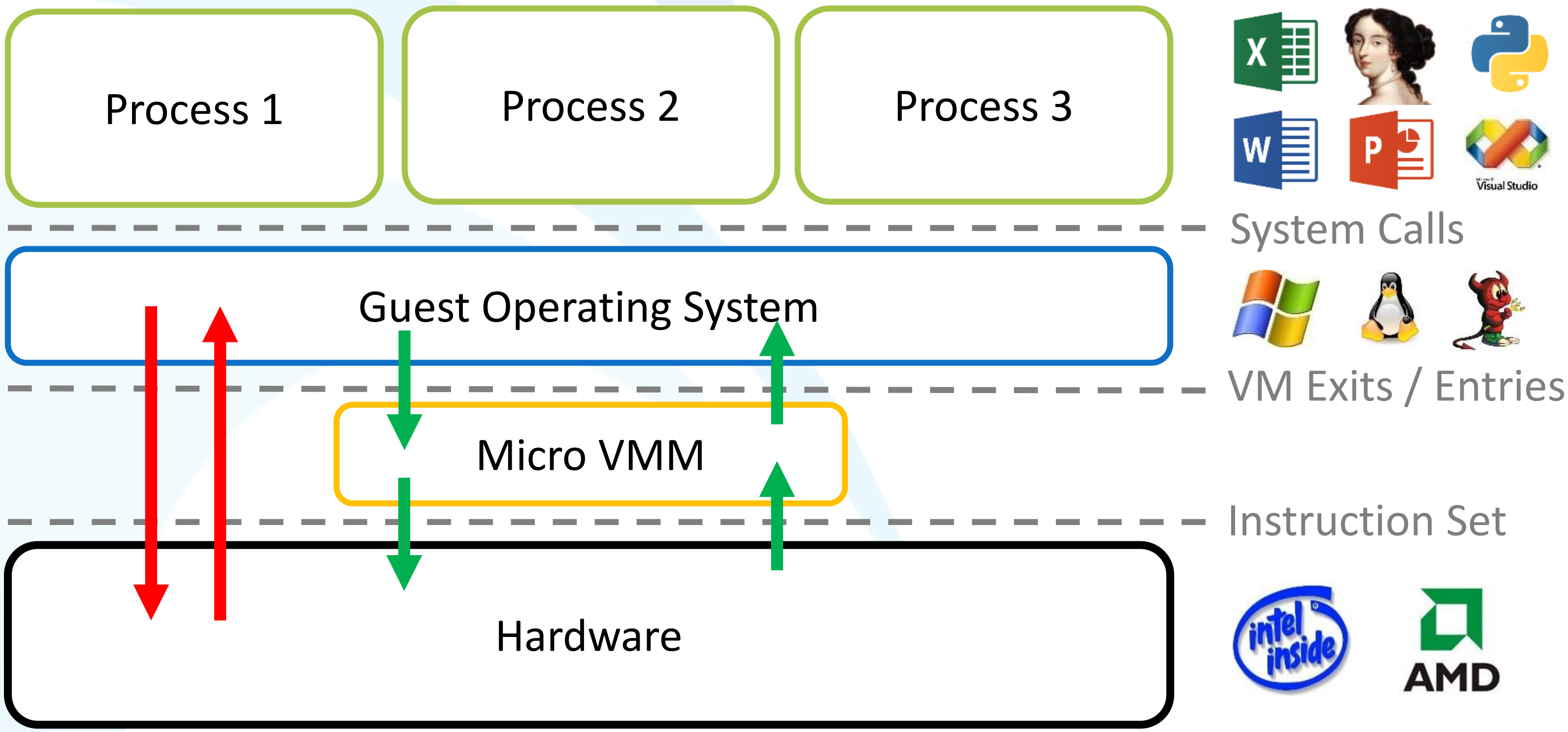
# Software Stack
# SMM with VMM



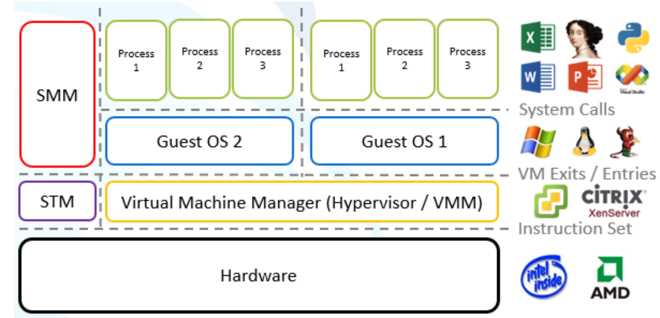
# Software Stack
# SMM Transfer Monitor (STM)



# Micro-VMMs


Process 1
Process 2
Process 3
Visual Studio
System Calls
Guest Operating System
VM Exits / Entries
Micro VMM
Instruction Set
Hardware
intel inside
AMD

# Section summary

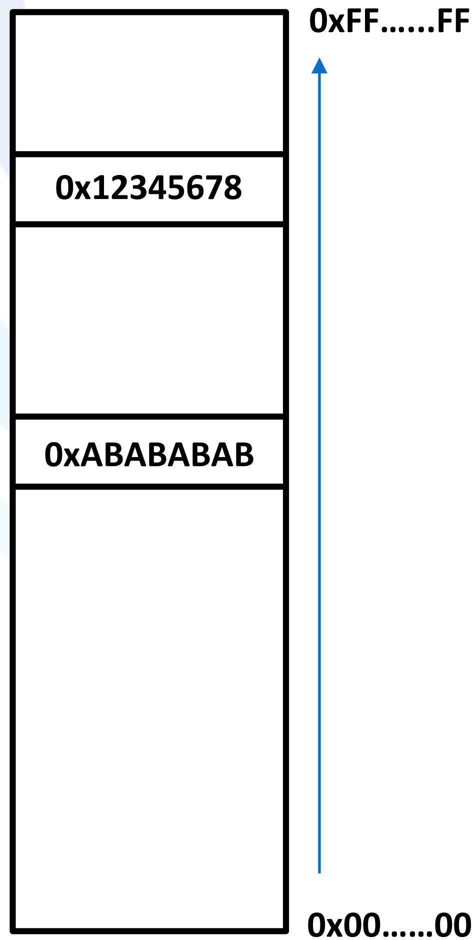


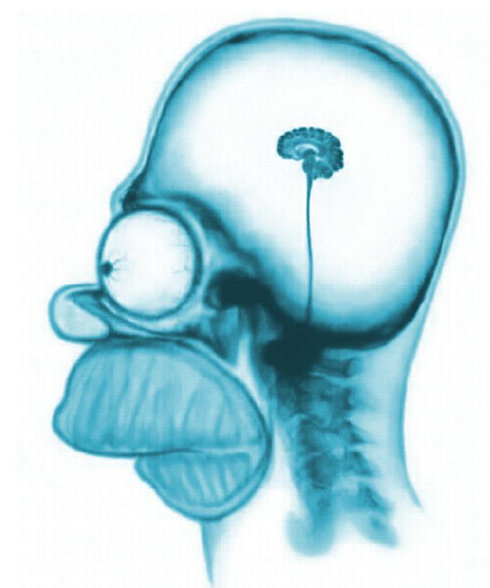

- There are **many** ways to use hardware virtualization technology:
  - Type I, Type II, Micro-VMMs, …
- Each approach has its own unique challenges:
  - Full HW virtualization: Secure a big implementation of SW emulation for all HW
  - Para-virtualization: Secure the guest OS interface with the host OS
  - All implementations: **Emulate ISA** correctly and securely
  - Micro-VMMs: Defend from **HW subversion**
- **SMM** is too privileged and where are the **STM**s?

# MEMORY

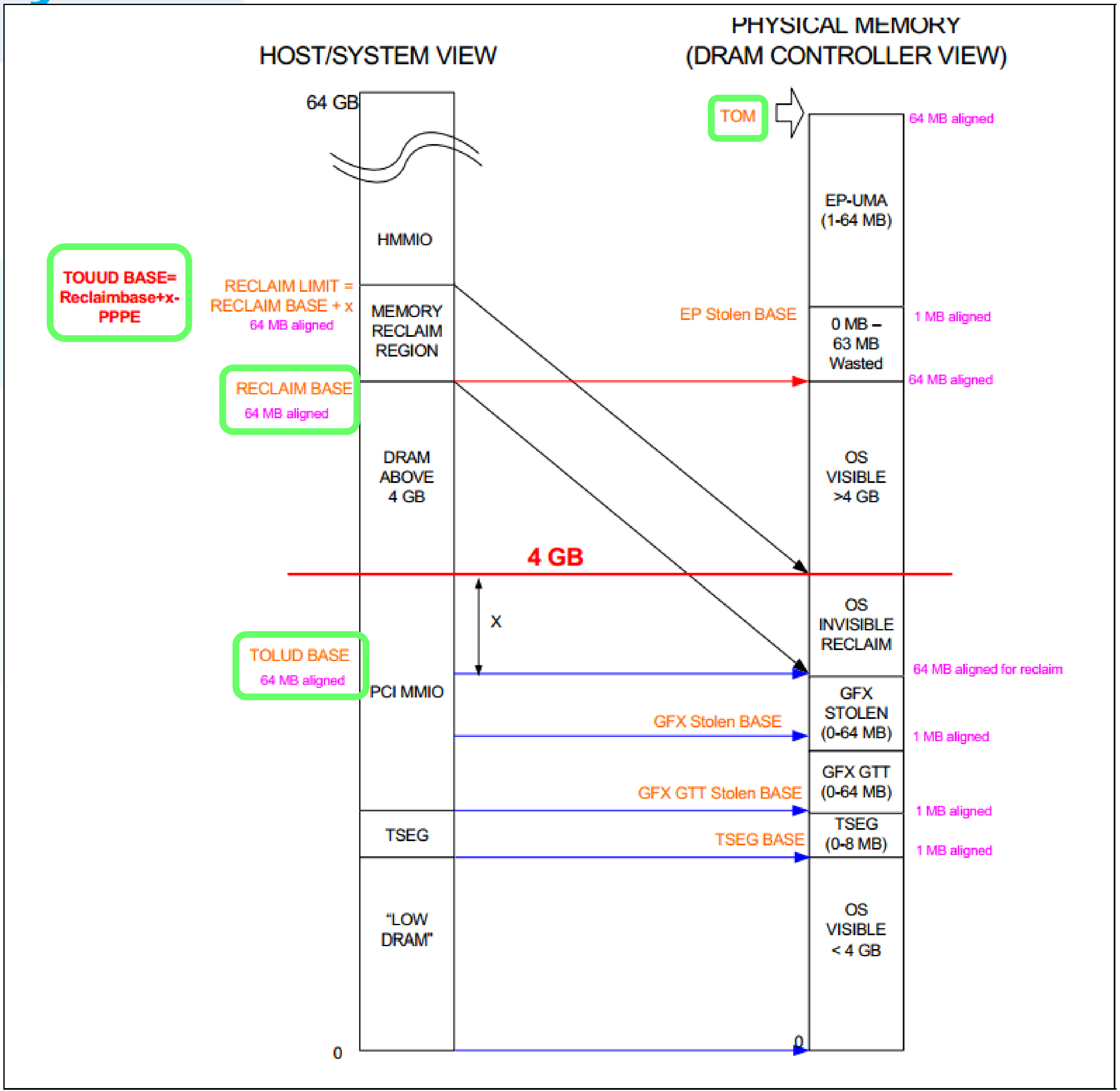
Where did I put that instruction?

# Memory is simple, right?

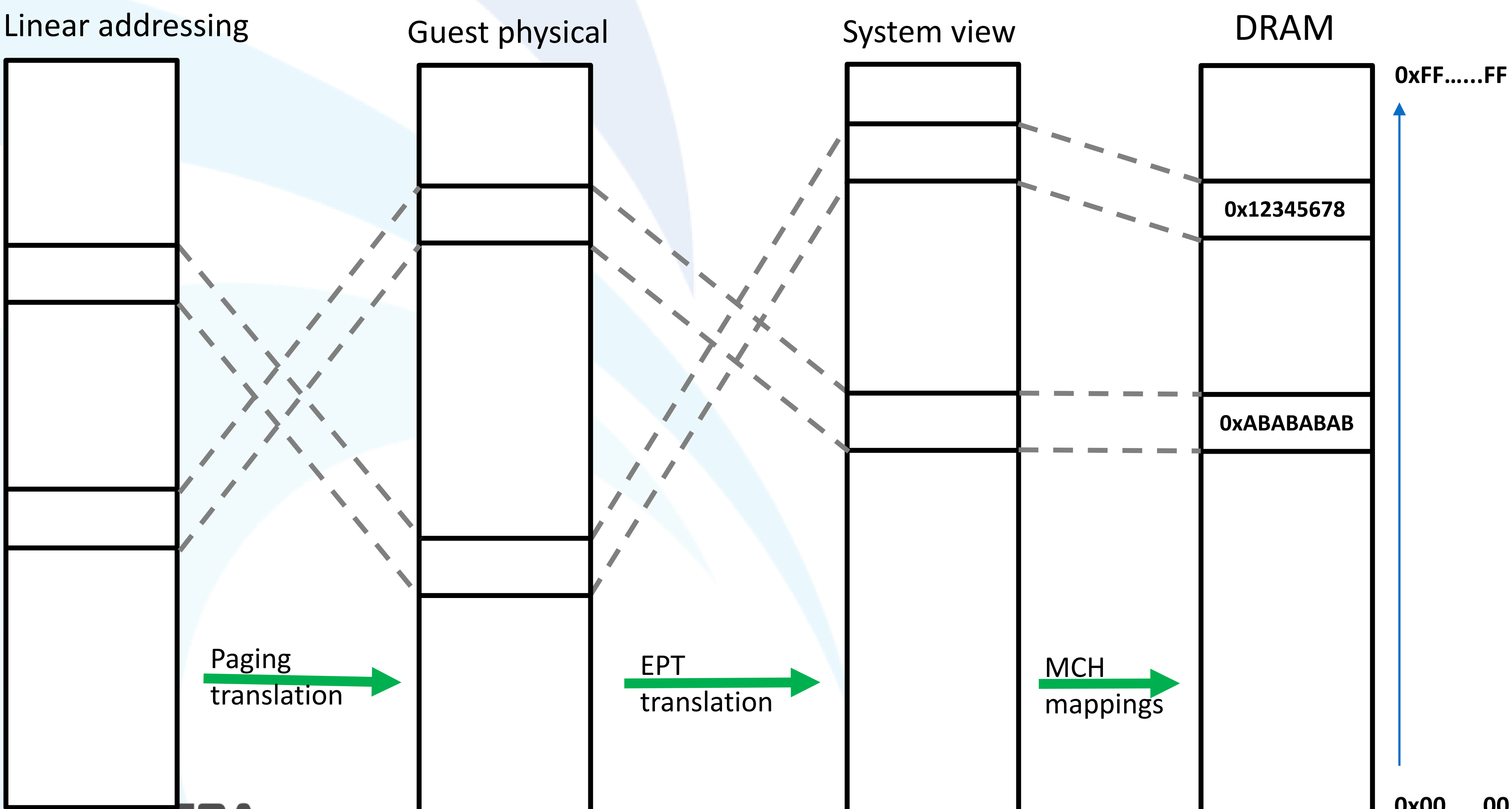


# Memory – Intel manual



# Memory – Address Translations

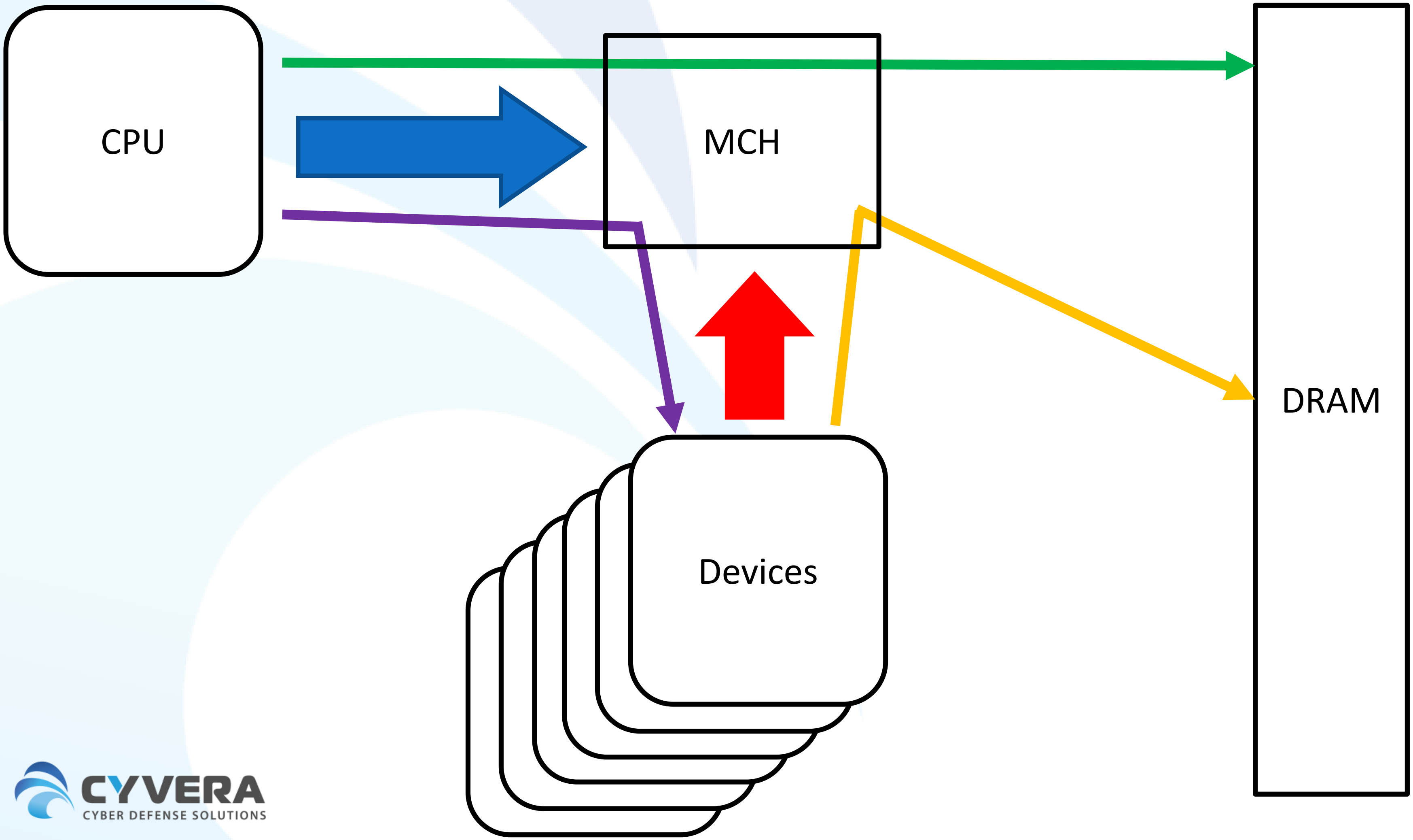




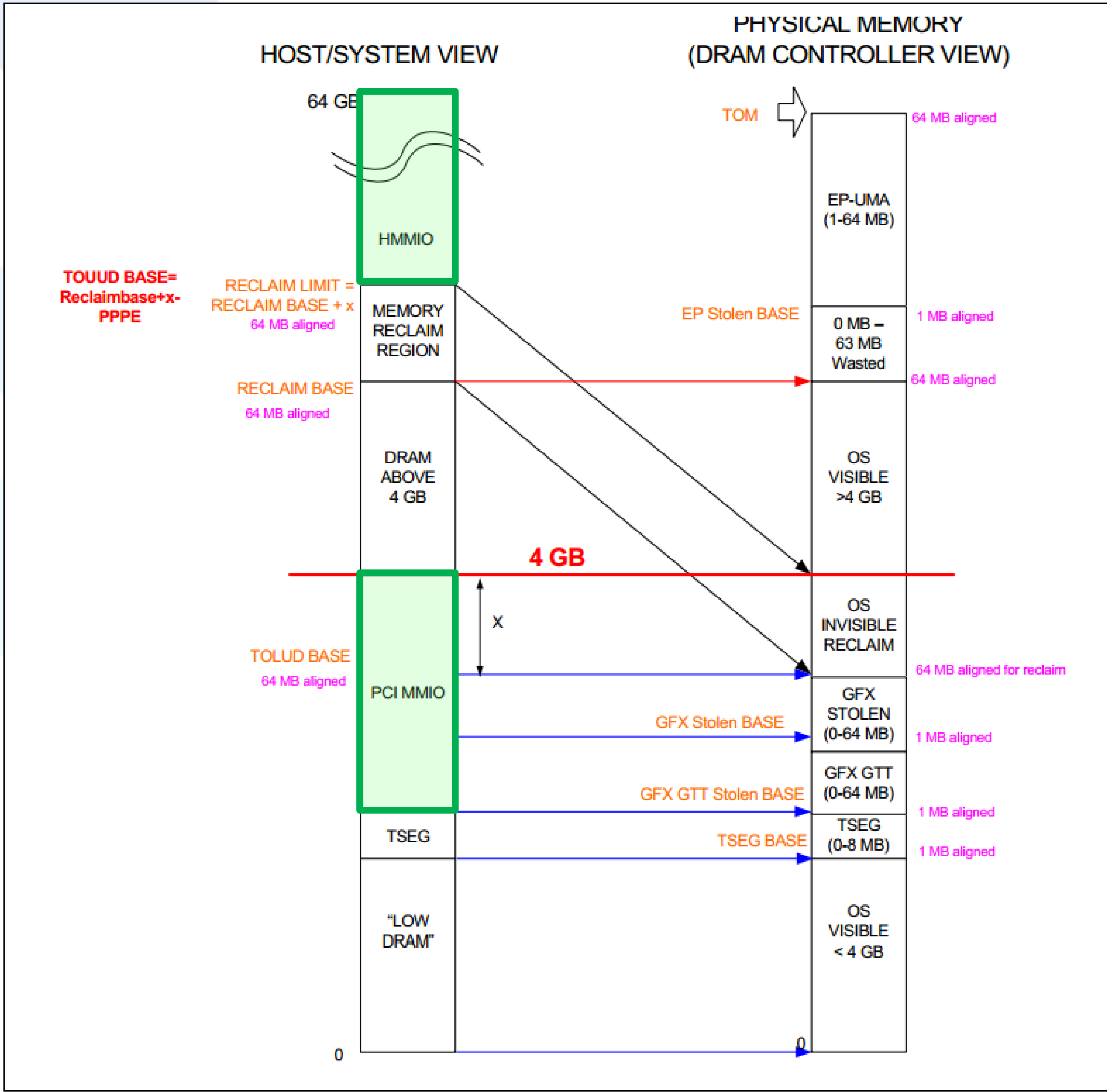

* MCH = Memory Control Hub (MMU)

** Segmentation adds another translation and the cache adds a whole new translation path

# Memory – point of view



# Memory – MMIO




CYVERA
CYBER DEFENSE SOLUTIONS

# Special address ranges

**Figure 2-3. Main Memory Address Range**

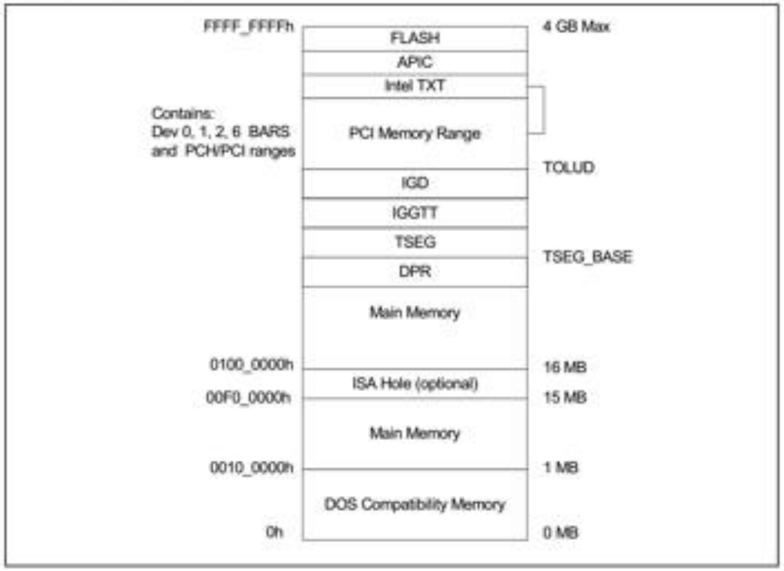



**Figure 7-4. PCI Memory Address Range**

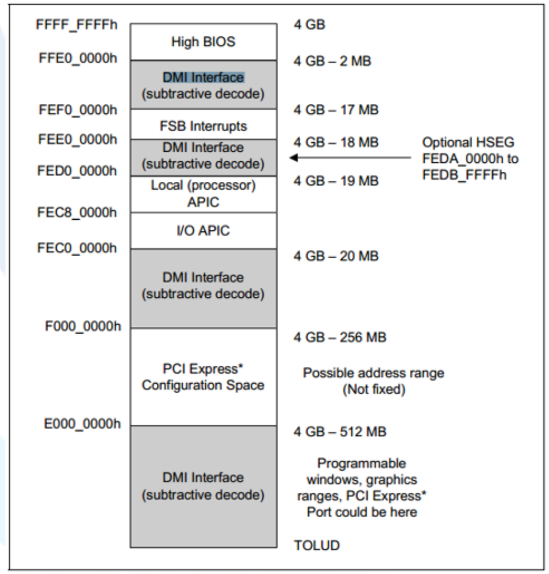



**Figure 2-1. System Address Range**

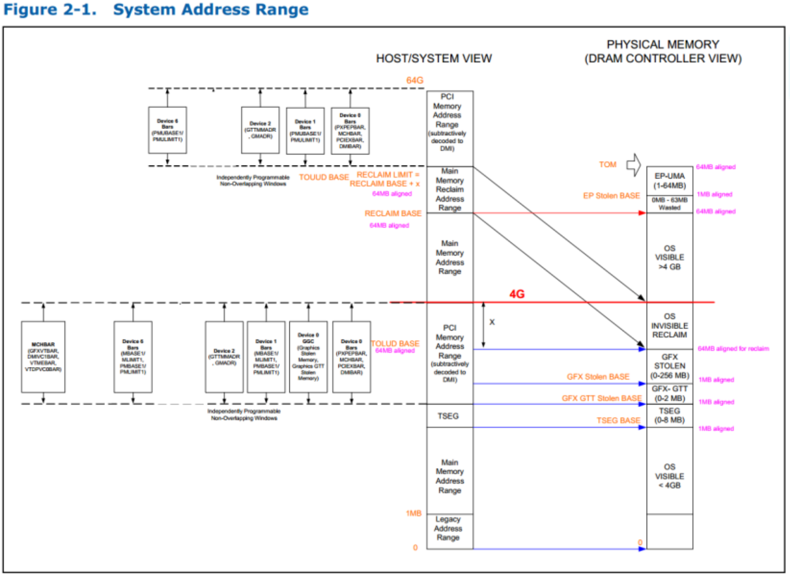
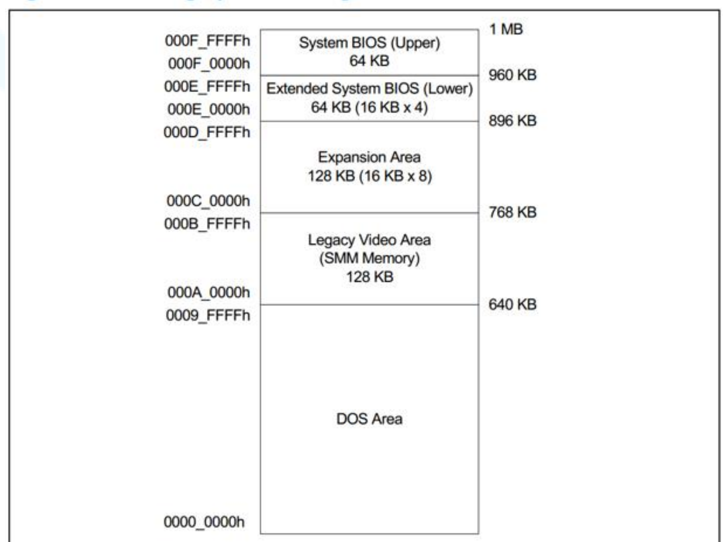



**Figure 2-2. DOS Legacy Address Range**



# 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

Performance

Security

# 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

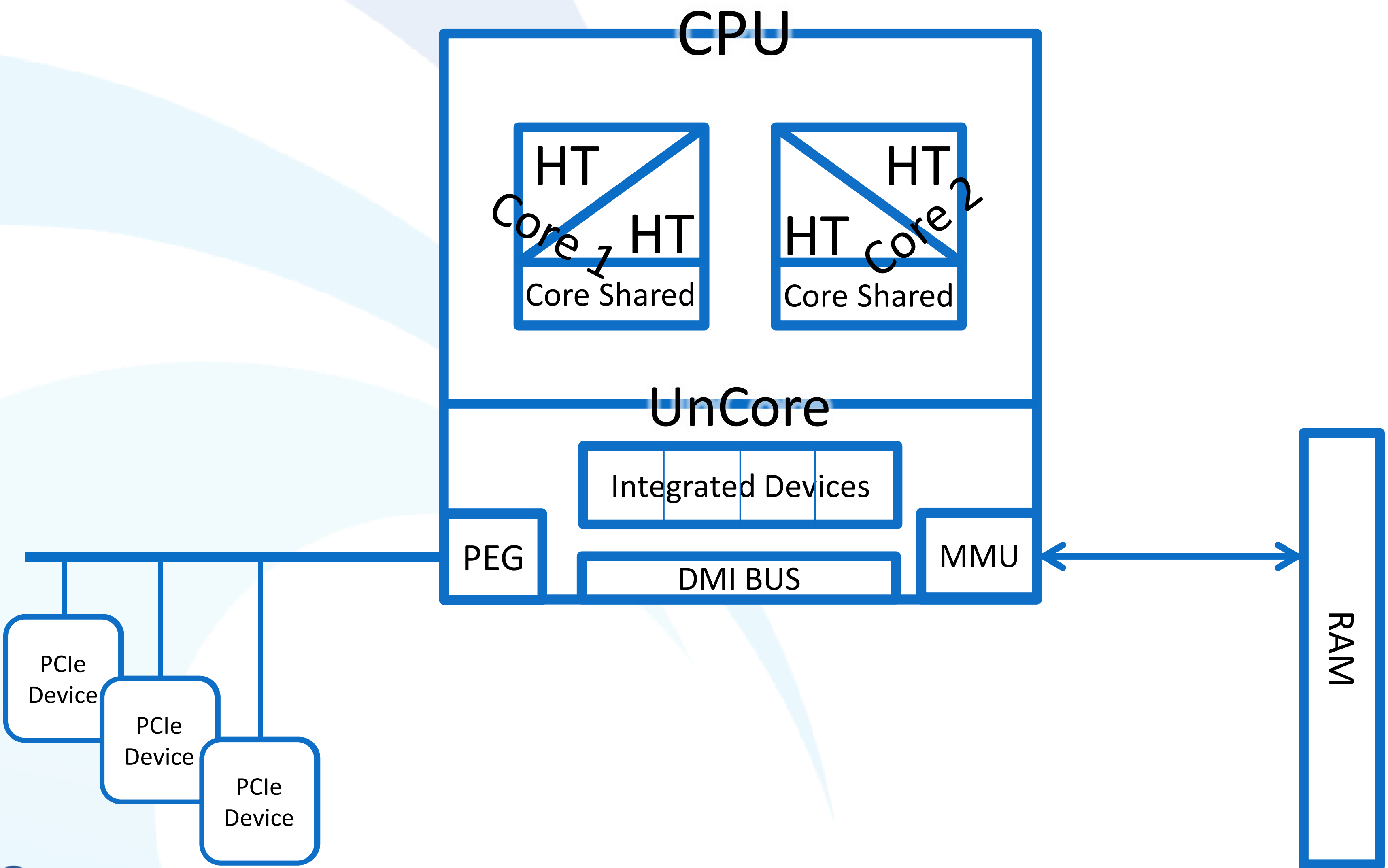
# COMPUTER PLATFORMS

## The insides

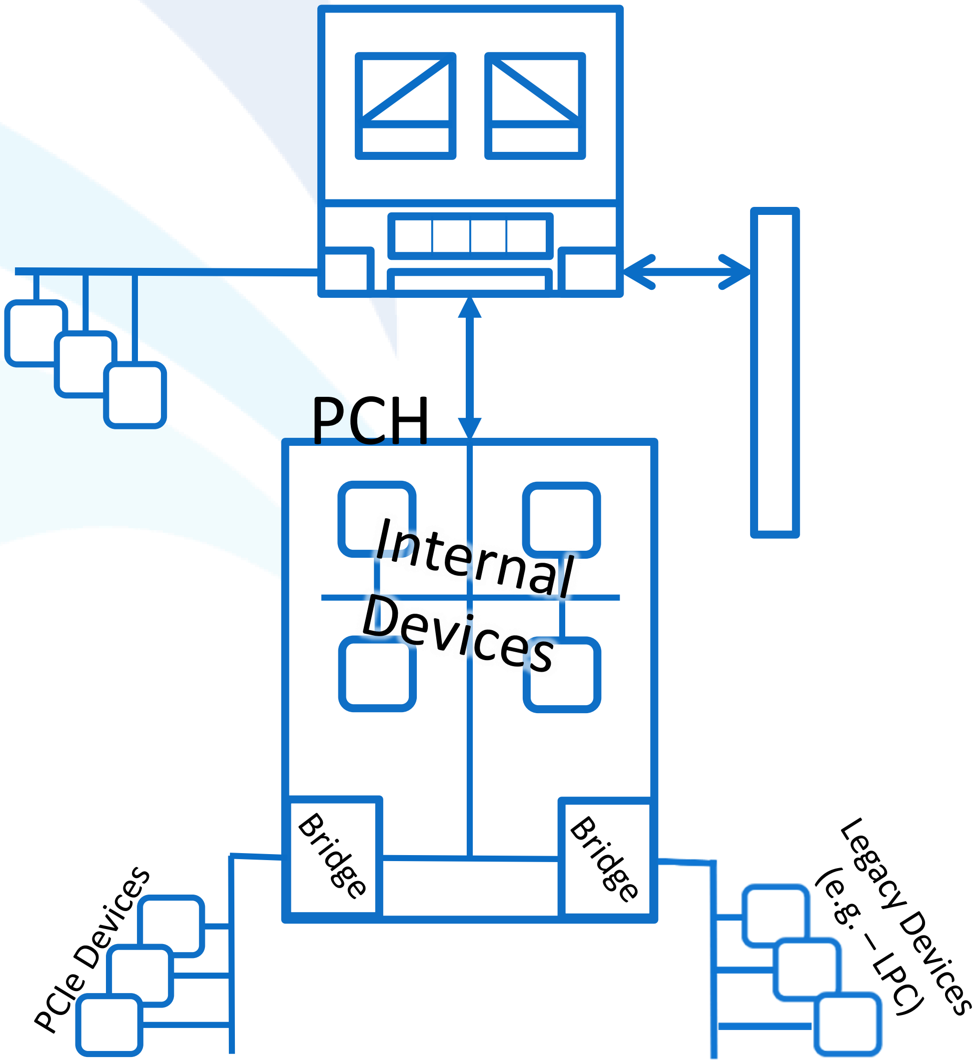
# What is a computer?

- 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?

| # | CVE ID | CWE ID | # of Exploits | Vulnerability Type(s) | Publish Date | Update Date | Score | Gained Access Level | Access | Complexity | Authentication | Conf. | Integ. | Avail. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | CVE-2013-4377 | 399 | | DoS | 2013-10-11 | 2013-10-15 | 2.3 | None | Local Network | Medium | Single system | None | None | Partial |
| | Use-after-free vulnerability in the virtio-pci implementation in Qemu 1.4.0 through 1.6.0 allows local users to cause a denial of service (daemon crash) by "hot-unplugging" a virtio device. | | | | | | | | | | | | | |
| 2 | CVE-2013-4344 | 119 | | Overflow +Priv | 2013-10-04 | 2013-10-23 | 6.0 | None | Local | High | Single system | Complete | Complete | Complete |
| | Buffer overflow in the SCSI implementation in QEMU, as used in Xen, when a SCSI controller has more than 256 attached devices, allows local users to gain privileges via a small transfer buffer in a REPORT LUNS command. | | | | | | | | | | | | | |
| 3 | CVE-2013-2007 | 264 | | | 2013-05-21 | 2013-08-22 | 6.9 | None | Local | Medium | Not required | Complete | Complete | Complete |
| | The qemu guest agent in Qemu 1.4.1 and earlier, as used by Xen, when started in daemon mode, uses weak permissions for certain files, which allows local users to read and write to these files. | | | | | | | | | | | | | |
| 4 | CVE-2012-6075 | 119 | | DoS Exec Code Overflow | 2013-02-12 | 2013-10-10 | 9.3 | None | Remote | Medium | Not required | Complete | Complete | Complete |
| | Buffer overflow in the e1000_receive function in the e1000 device driver (hw/e1000.c) in QEMU 1.3.0-rc2 and other versions, when the SBP and LPE flags are disabled, allows remote attackers to cause a denial of service (guest OS crash) and possibly execute arbitrary guest code via a large packet. | | | | | | | | | | | | | |
| 5 | CVE-2012-3515 | 20 | | +Priv | 2012-11-23 | 2013-10-10 | 7.2 | None | Local | Low | Not required | Complete | Complete | Complete |
| | Qemu, as used in Xen 4.0, 4.1 and possibly other products, when emulating certain devices with a virtual console backend, allows local OS guest users to gain privileges via a crafted escape VT100 sequence that triggers the overwrite of a "device model's address space." | | | | | | | | | | | | | |
| 6 | CVE-2012-2652 | | | | 2012-08-07 | 2013-04-18 | 4.4 | None | Local | Medium | Not required | Partial | Partial | Partial |
| | The bdrv_open function in Qemu 1.0 does not properly handle the failure of the mkstemp function, when in snapshot node, which allows local users to overwrite or read arbitrary files via a symlink attack on an unspecified temporary file. | | | | | | | | | | | | | |
| 7 | CVE-2011-2527 | 264 | | | 2012-06-21 | 2012-06-26 | 2.1 | None | Local | Low | Not required | Partial | None | None |
| | The change_process_uid function in os-posix.c in Qemu 0.14.0 and earlier does not properly drop group privileges when the -runas option is used, which allows local guest users to access restricted files on the host. | | | | | | | | | | | | | |
| 8 | CVE-2011-2212 | 119 | | DoS Overflow +Priv | 2012-06-21 | 2012-06-26 | 7.4 | None | Local Network | Medium | Single system | Complete | Complete | Complete |
| | Buffer overflow in the virtio subsystem in qemu-kvm 0.14.0 and earlier allows privileged guest users to cause a denial of service (guest crash) or gain privileges via a crafted indirect descriptor related to "virtqueue in and out requests." | | | | | | | | | | | | | |
| 9 | CVE-2011-1751 | 20 | | DoS Exec Code | 2012-06-21 | 2013-02-13 | 7.4 | None | Local Network | Medium | Single system | Complete | Complete | Complete |
| | The pciej_write function in hw/acpi_piix4.c in the PIIX4 Power Management emulation in qemu-kvm does not check if a device is hotpluggable before unplugging the PCI-ISA bridge, which allows privileged guest users to cause a denial of service (guest crash) and possibly execute arbitrary code by sending a crafted value to the 0xae08 (PCI_EJ_BASE) I/O port, which leads to a use-after-free related to "active qemu timers." | | | | | | | | | | | | | |
| 10 | CVE-2011-1750 | 119 | | DoS Overflow +Priv | 2012-06-21 | 2012-06-26 | 7.4 | None | Local Network | Medium | Single system | Complete | Complete | Complete |
| | Multiple heap-based buffer overflows in the virtio-blk driver (hw/virtio-blk.c) in qemu-kvm 0.14.0 allow local guest users to cause a denial of service (guest crash) and possibly gain privileges via a (1) write request to the virtio_blk_handle_write function or (2) read request to the virtio_blk_handle_read function that is not properly aligned. | | | | | | | | | | | | | |
| 11 | CVE-2011-0011 | 287 | | Bypass | 2012-06-21 | 2012-06-21 | 4.3 | None | Local Network | High | Not required | Partial | Partial | Partial |
| | qemu-kvm before 0.11.0 disables VNC authentication when the password is cleared, which allows remote attackers to bypass authentication and establish VNC sessions. | | | | | | | | | | | | | |
| 12 | CVE-2010-0297 | 119 | | DoS Exec Code Overflow | 2010-02-12 | 2010-08-21 | 7.2 | None | Local | Low | Not required | Complete | Complete | Complete |
| | Buffer overflow in the usb_host_handle_control function in the USB passthrough handling implementation in usb-linux.c in QEMU before 0.11.1 allows guest OS users to cause a denial of service (guest OS crash or hang) or possibly execute arbitrary code on the host OS via a crafted USB packet. | | | | | | | | | | | | | |
| 13 | CVE-2009-3616 | 399 | | Exec Code | 2009-10-23 | 2009-12-19 | 8.5 | None | Remote | Medium | Single system | Complete | Complete | Complete |

# VT-d (IOMMU)

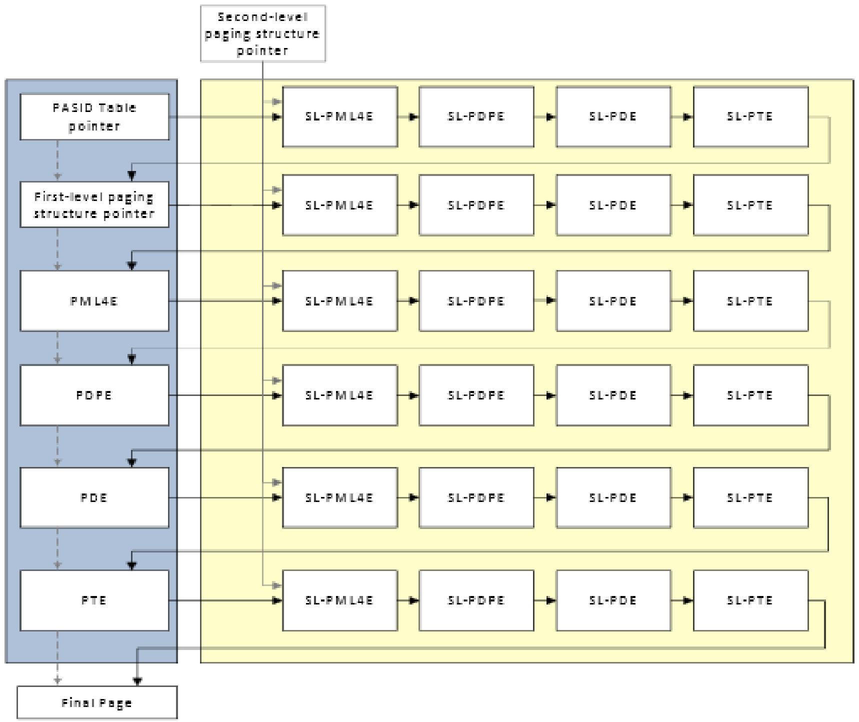


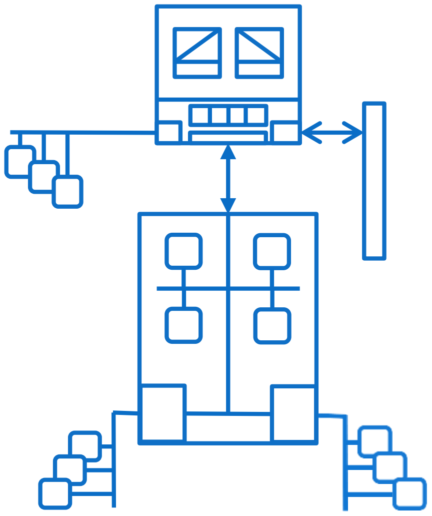

**Figure 3-12. Nested Translation with 4-KByte pages**

- 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?

CYVERA
CYBER DEFENSE SOLUTIONS

# Section summary

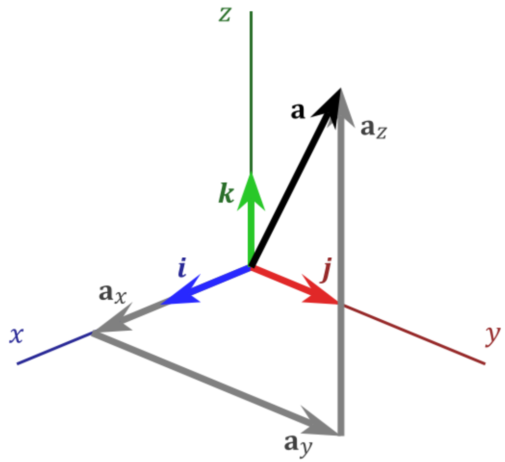
- Computer hardware is complex!
- Emulating necessary components is hard:
  - Multiple CVEs already found in ACPI and APIC virtualization as well as QEMU
- VT-d helps virtualizing DMA and hardware interrupts
  - If used correctly



# ATTACK VECTORS TAXONOMY

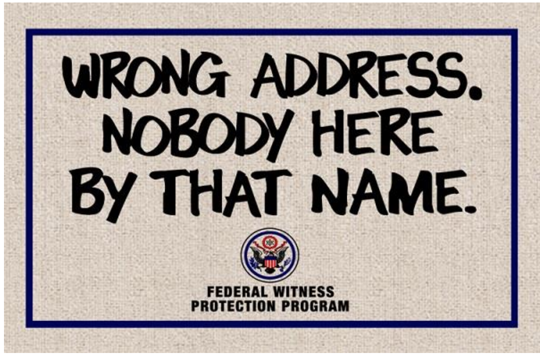
## Potential and practical ones

# Basic Vectors




- <u>ISA Implementation</u>
  - Emulating x86 isn't easy…
- <u>Performance monitoring</u>
  - Classic side channels
  - Real Time Instruction Tracing – new feature coming up
- <u>Old systems</u>
  - New defenses were introduced with the latest HW
- <u>New features</u>
  - 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


- <u>Corrupted ACMs</u>
  - ACMs run in a very high privilege, if you can compromise one…
- <u>CPU/PCH Firmware(s)</u>
  - Compromise of the CPU or PCH firmware naturally allows an attacker to control any VM
- <u>BIOS & SMM</u>
  - The BIOS is a common component of the platform and controls both configuration and SMM code

# Other Interesting Vectors

- <u>Intentional misconfiguration</u>
  - It is possible to misconfigure PCIe config space, MSRs or MMIO constants in order to create unexpected situations for the VMM
- <u>Server platforms (are fun!)</u>
  - Platforms and CPUs for the server market have special features, all of those usually run in very high privilege
- <u>Errata</u>
  - 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**

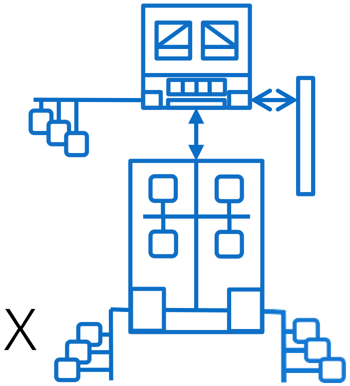
Problem: When 32-bit paging is in use, the processor should use a page directory located at the 32-bit physical address specified in bits 31:12 of CR3; the upper 32 bits of CR3 should be ignored. Due to this erratum, the processor will use a page directory located at the 64-bit physical address specified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

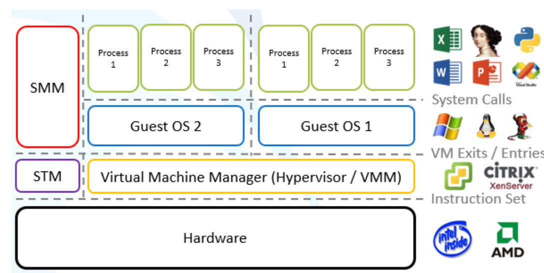
Workaround: Software that has executed in 64-bit mode should reload CR3 with a 32-bit value before returning to 32-bit paging.

Status: For the steppings 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