Crucial Security Programs

Ring -1 vs. Ring -2: Containerizing Malicious SMI Interrupt Handlers on AMD-V

Pete Markowsky
Senior Security Researcher
pmarkows@harris.com

You might remember me from such films as...

HARRIS

- Who Am I?
  - Pete Markowsky
    - Senior Security Researcher at Crucial Security Programs, Harris Corporation
    - transzorp on IRC and Twitter
    - Worked in
      - .edu
      - .com
      - .mil

What is this talk about?

HARRIS

- System Management Mode and System Management Mode Interrupts (SMIs)
- Virtualization on the AMD platform
- The interplay between virtualization extensions in AMD-V and System Management Mode (SMM)
  - which mode already has or can seize control of execution, when, and where

Why should I care?

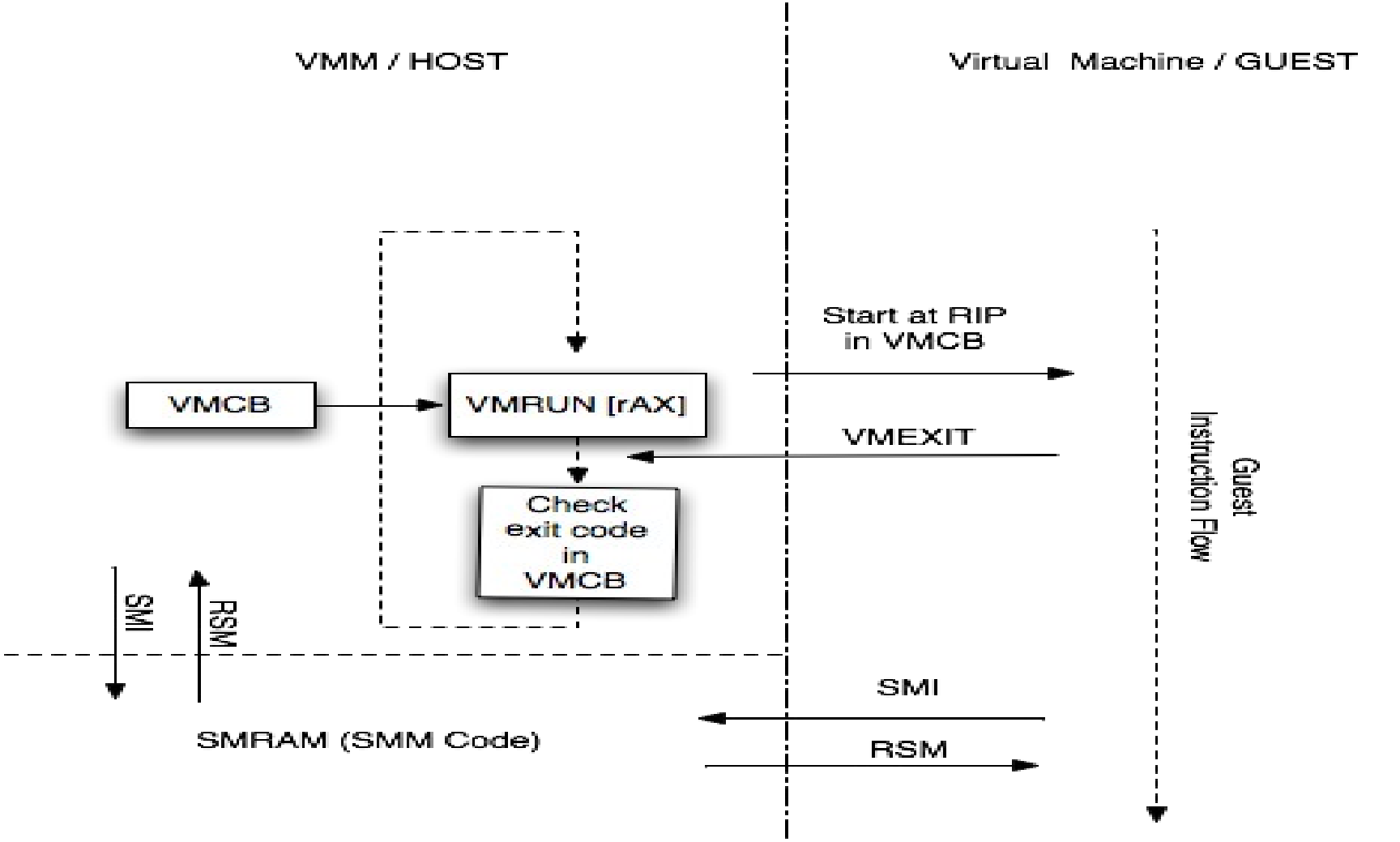
HARRIS

- If you're building a security mechanism based on a VMM on AMD-V
- Normally SMM code runs **outside** of the VMM's protections
- Malicious System Mode Interrupt (SMI) handlers can be used for:
  - bypassing Intel's TXT
  - rootkits
  - privilege escalation attacks

Assumptions

- An attacker has the ability to install an SMI handler
  - can take advantage of BIOS overwrites or cache attacks
  - they have root in a guest
- An attacker may have subverted the BIOS (up to a point)

The problem


VMM / HOST
Virtual Machine / GUEST
Start at RIP in VMCB
VMCB
VMRUN [rAX]
VMEXIT
Check exit code in VMCB
Guest Instruction Flow
SMI
RSM
SMI
RSM
SMRAM (SMM Code)

Ring -2: System Management Mode (SMM)

HARRIS

- System Management Mode is
  - a 16-bit x86 operating mode like real-mode
  - transparent to the OS
  - supposed to be for power management
- SMM is entered **only** when a System Mode Interrupt (SMI) is asserted by hardware
- Called “Ring -2” because normally
  - there are no restrictions to the I/O ports or memory
  - it has the same privileges as in Ring 0
  - it can manipulate all of system memory

Ring -2: SMRAM

- SMM code in SMRAM
  - range of physical memory
- SMBASE MSR
  - defines start of SMRAM
- Default size on AMD64 64K
  - can range from 32K – 4GB

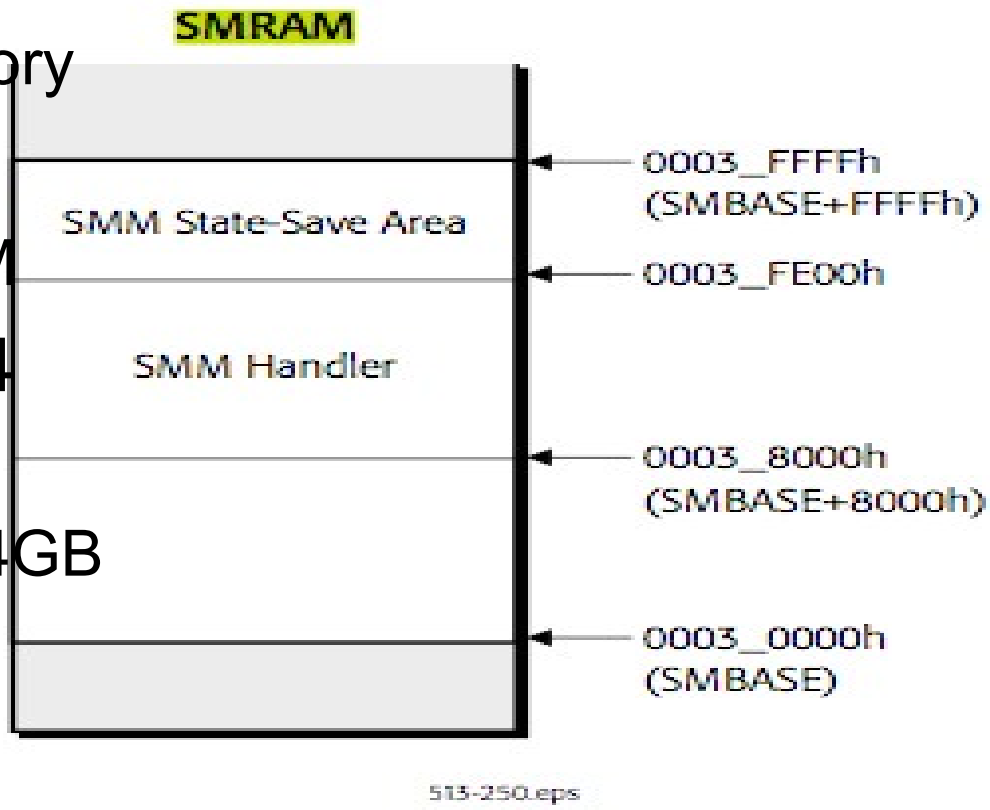


Figure 10-1. Default SMRAM Memo

Ring -2: Where did it come from?

HARRIS

- Installed by the BIOS
  - BIOS sets up SMRAM (copies handler)
  - Configures SMM MSRs
  - Typically locks SMM registers by setting the HWCR.SMMLOCK bit
    - Prevents access to SMRAM

Ring -2: SMM security

HARRIS

- Two segments of physical memory protected for SMM code
  - ASeg: `0xA0000` – `0xBFFFF`
  - TSeg: defined by SMM_ADDR and SMM_MASK
- Bits in SMM_MASK MSR set protection for segments
- Protected segments can only be accessed in SMM

Ring -2: System Mode Interrupt (SMI)

HARRIS

- Special Non-Maskable Interrupt
- Takes priority over other interrupts
- Two types
  - internal: generated by hardware in response to IOIO instructions
    - which IOIO depend on motherboard logic
  - external: generated by hardware
- Causes switch to SMM

Ring -2: In what modes can an SMI occur?

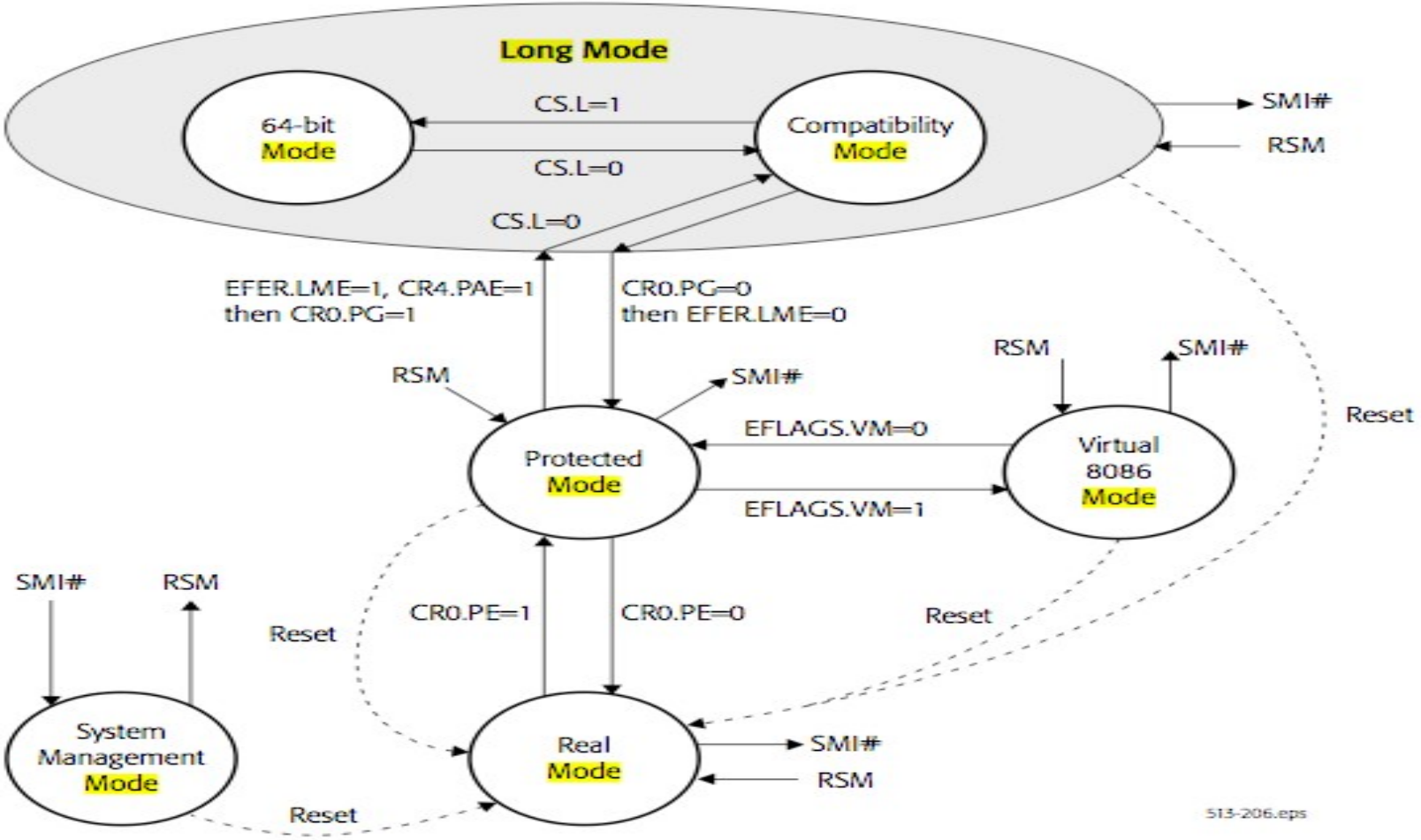


Figure 1-6. Operating Modes of the AMD64 Architecture

Ring -2: What happens during an SMI

HARRIS

- Stops fetching instructions
- Processor waits for currently executing instructions and buffered memory writes to finish
- CPU state is saved in the SMRAM save state region
- RIP changed to the start of the SMI handler

Ring -2: SMI handler

HARRIS

- Basically real mode x86 code
  - segments are extended up to 4GB
- RSM instruction returns to previous execution mode
- Address translation somewhat difficult
  - read CR3 from save state area and do translation by hand

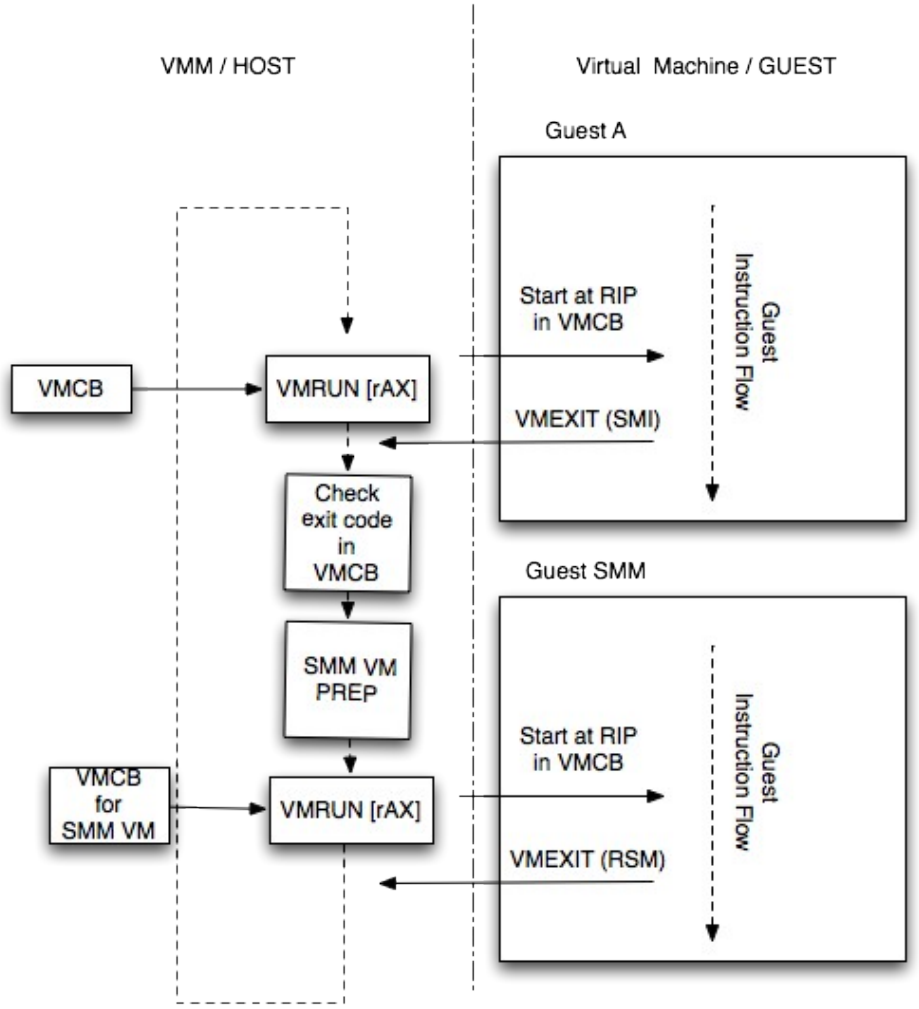
Ring -2: Known methods of installing an SMI handler

HARRIS

- Install a modified BIOS
  - write code to DRAM and set up appropriate MSRs
- Find a bug in the SMI handler and exploit it
  - Invisible Things Labs Q35 remapping attack
- x86 cache manipulation
  - Duflot's attack

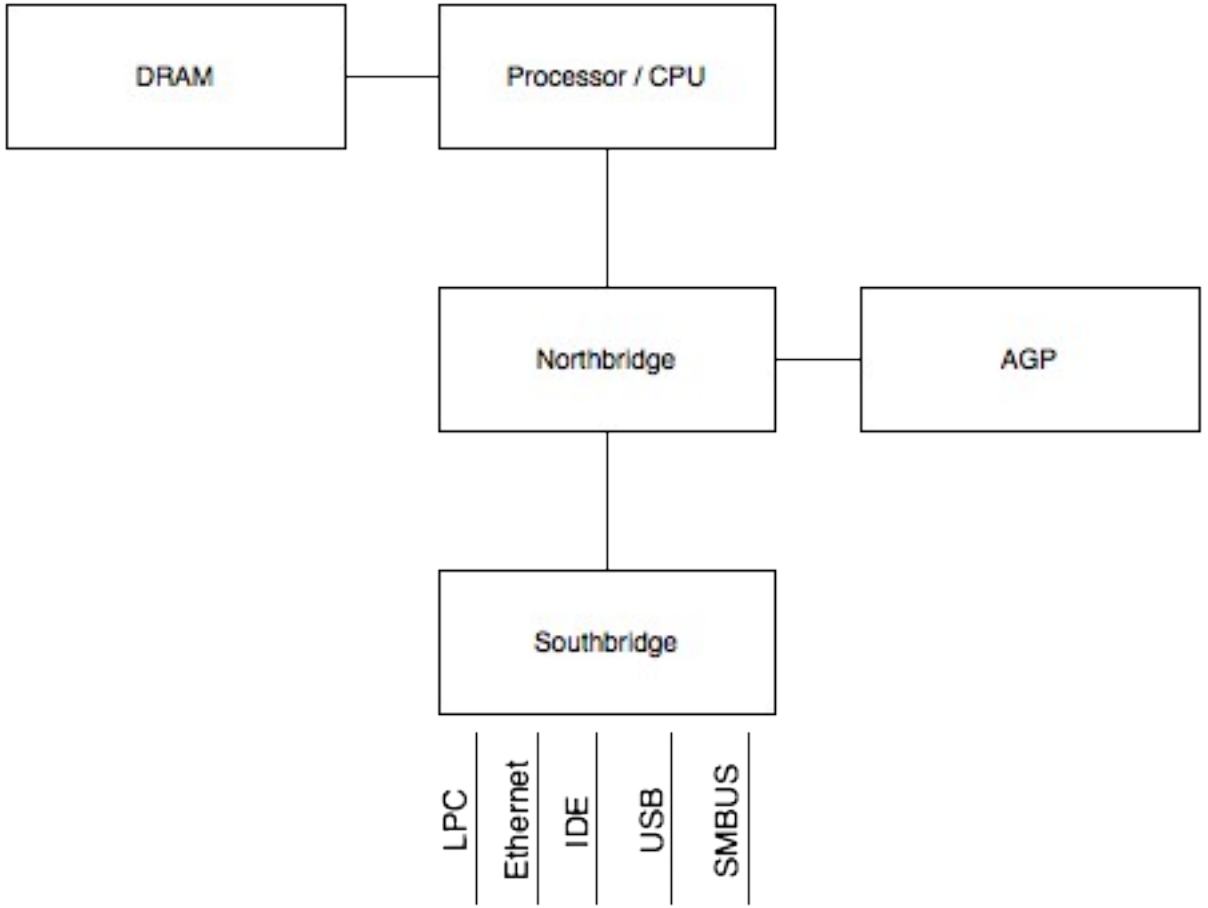
Goal

- Put all of the SMM code into a virtual machine so that its execution can be constrained by the VMM



Ring -1: AMD64 organization

- Memory controller moved into the CPU

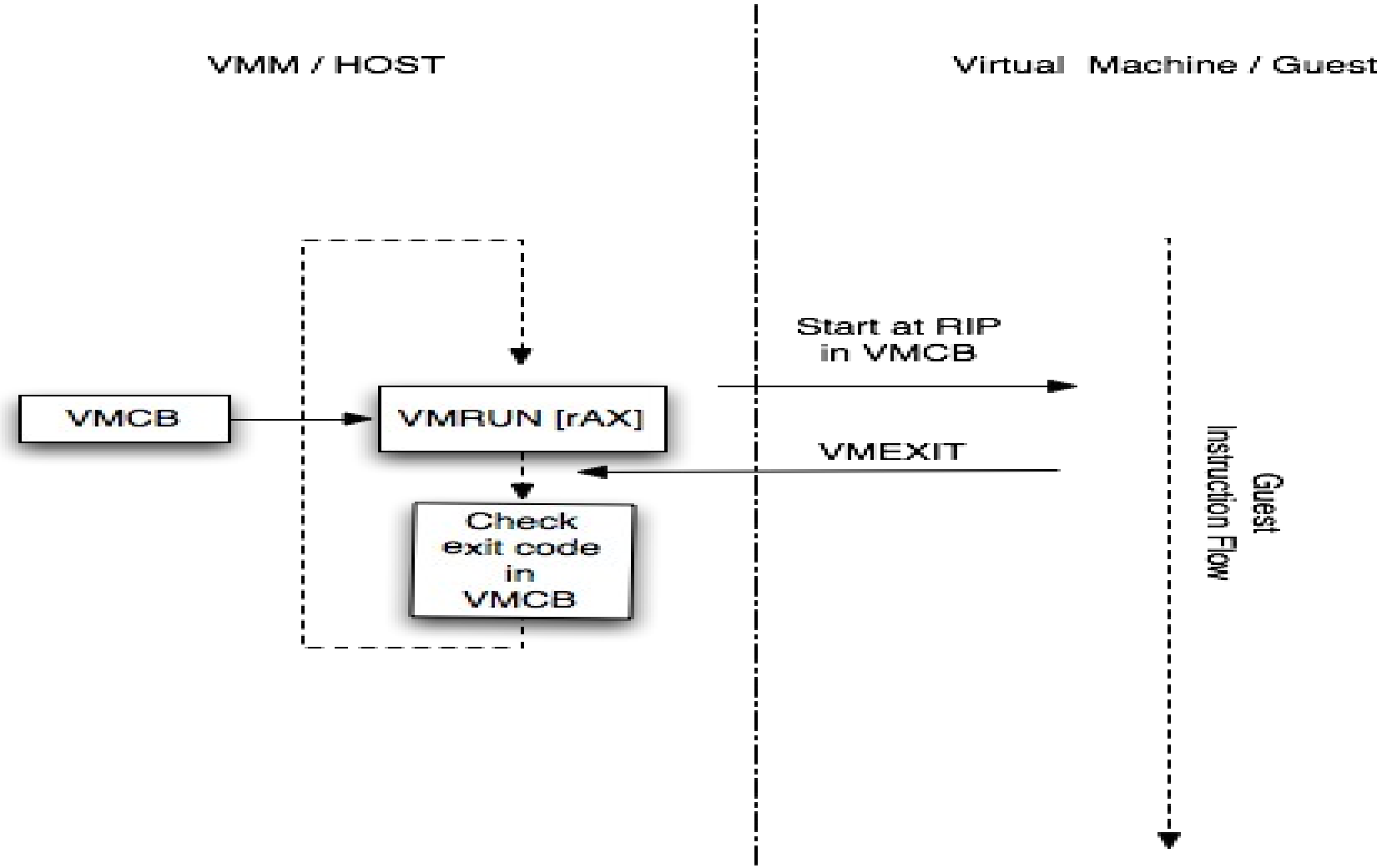


Ring -1: AMD-V quick overview

HARRIS

- x86-64 virtualization extensions
  - based around the VMRUN instruction and VMEXITs
- Hypervisor defines VMEXIT conditions in a page of memory called the Virtual Machine Control Block (VMCB)
  - these conditions are called interceptions
- One VMCB per guest
- A VMCB defines processor state at resumption of execution (after a VMRUN)
- A VMCB contains the state of the guest at the point of VMEXIT

Ring -1: AMD-V quick overview



Ring -1: Key features of AMD-V

HARRIS

- Global Interrupt Flag (GIF)
  - when set to 1 allows all interrupts
  - when set to 0 blocks all interrupts including SMI
- Paged Real Mode
  - run real mode code with paging protections applied
- SMM_CTL register
  - allows VMM code to map SMRAM

Ring -1: VMCB

- VMCB has two parts
  - state save area: contains guest CPU state
  - control area: defines intercept conditions
- Intercepts we care about:
  - SMI (bit 98)
  - RSM (bit 115)
  - interception of IOIO instructions (bit 123)
  - MSR_PROT (bit 124)

Ring -1: Dealing with SMIs

HARRIS

- Three solutions offered in AMD64 System Programmer's Manual
  - do nothing
    - SMM runs outside of VMM protections
  - intercept all IOIO instructions that can cause an SMI
    - complicated
    - doesn't stop replaced handler
  - containerize the SMI Handler
    - run the SMI handler inside a virtual machine

Containerizing SMI handlers

HARRIS

- Two solutions
  - SMM hypervisor
    - rewrite SMRAM to contain a VMM that places the original SMM handler in a virtual machine
  - write code in the VMM to emulate switching to and from SMM, then run SMRAM code in a VM

Ring -1: Intercepting SMIs

- The SMI intercept causes SMIs to transfer control to the VMM before the SMI handler
- SMI intercept is ignored when the SMMLOCK is 1

Ring -1: How to containerize an SMI handler

HARRIS

- To containerize an SMI Handler:
  1. Ensure that HWCR.SMMLOCK is 0
  2. Get access to SMRAM
  3. Configure a shadow page table for SMRAM
  4. Allocate and configure a VMCB for the SMM VM
  5. Configure guest VMCBs to intercept SMI
  6. Write code to copy guest state to and from the SMM VM's save state area

Ring -1: Ensuring that the SMMLOCK is 0

- Bit 0 of the HWCR is the SMMLOCK
  - if it's 0 all SMM related MSRs may be written to in CPL 0
- SMM_KEY
- Need some cooperation from BIOS

Example:

```
if (rdmsr(HWCR) & 1){
//locked use SMM_KEY
  wrmsr(SMM_KEY,
    passwd);
  if (rdmsr(HWCR)& 1)
    halt(); //WRONG PASS
}
```

Ring -1: Get access to SMRAM

HARRIS

- Problem: SMRAM only accessible from SMM
- Solution: SMM_CTL MSR
- First map SMRAM region using SMM_CTL MSR
  - `wrmsr(SMM_CTL, SMI_ENTER)`
  - causes SMRAM to be mapped

Ring -1: Allocate and configure a VMCB for the SMM VM

HARRIS

- In the VMCB for the virtual machine you need to:
  - set VMCB CR0 field to CR0.PG=1 CR0.PE=0
    - Paged Real Mode
  - set VMCB CR3 field to the physical address of the guest page table
  - set VMCB exception vector to intercept #PF
  - set VMCB RSM intercept
  - set VMCB shutdown intercept
  - intercept all SMM MSR writes

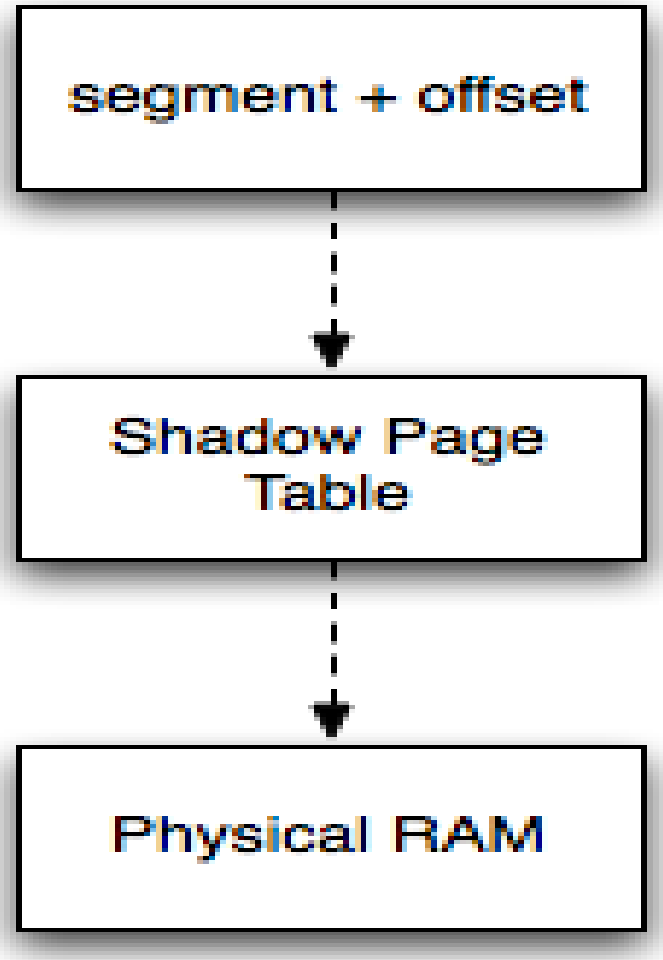
Ring -1: Allocate and configure a guest VMCB

HARRIS

- Prevent attempts to write MSRs by configuring VMCB MSRPM to:
  - intercept writes to
    - HWCR
    - SMMBASE
    - SMM_ADDR
    - SMM_MASK
    - SMM_CTL
    - SMM_KEY

Ring -1: Allocate a shadow page table

- Paged real mode requires a shadow page table
- Maps a guest linear address to a host physical address
  - (segment+offset) = guest linear address
- Must create entries for SMRAM



Ring -1: Copying to and from guest state

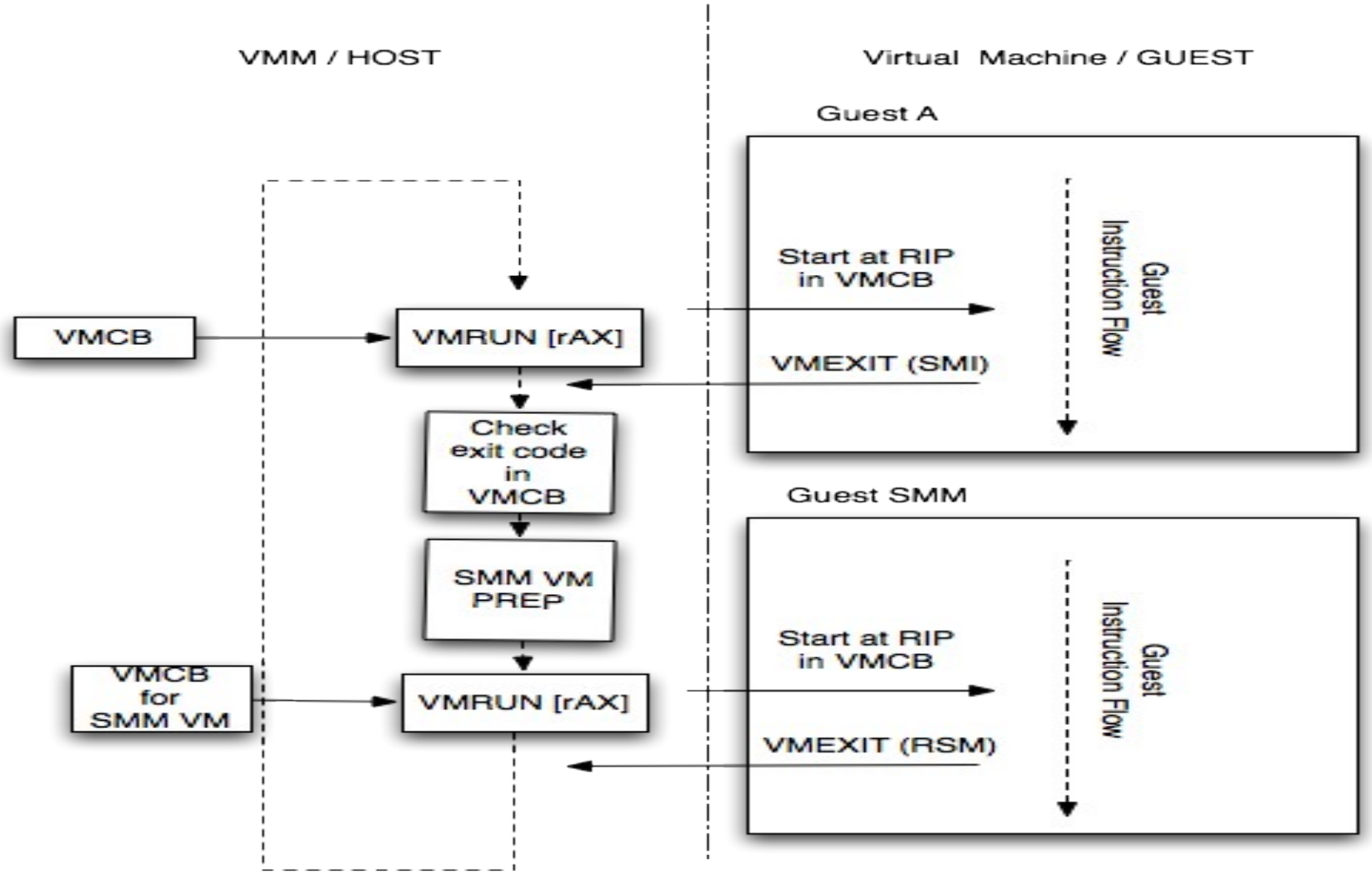
HARRIS

- SMM save state area
  - 512 bytes at the end of SMRAM
  - contains processor state at the time of the SMI
- Copy from SMM VM and guest VMCB
- Filter out known values before returning to guest

Demo

- Modified MAVMM to containerize the handler

What happened?



Aside: SKINIT

HARRIS

- SKINIT (Secure Kernel Init)
  - new instruction in AMD-V to securely load a VMM
  - similar to Intel's SENTER
- What it does
  - stops all other processors than the bootstrap processor
  - reinitializes processor state as if a reset has been asserted
  - uses DEV if set up in secure loader
  - sets GIF to 0
  - loads an Secure Loader (SL) into RAM
  - verifies integrity of Secure Loader in TPM

Differences from Intel VT

HARRIS

- Conceptually AMD-V and Intel VT are similar
- Intel VMCS is a variable length data structure
- Intel does not have an explicit intercept for SMI
  - dual monitor treatment
    - put another hypervisor in SMRAM
    - not all VT hardware supports it.
- Uses virtual 8086 mode to virtualize real mode code
  - sometimes needs other code to virtualize real mode code
    - vmxassist (from Xen)

Thank You!

- Thank you for attending my presentation and coming to ShmooCon 2010!

Questions?

- I’m happy to answer as many questions as I can in the time left