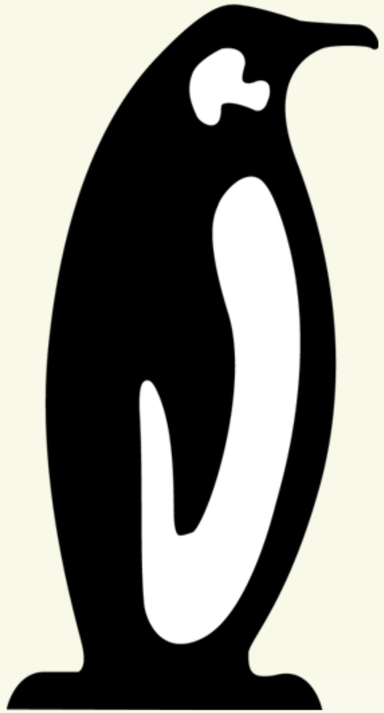


Linux Plumbers Conference

Richmond, Virginia | November 13-15, 2023

Linux
Plumbers
Conference | Richmond, VA | Nov. 13-15, 2023

Secure AVIC

Securing Interrupt Injection from a Malicious Hypervisor

Authors:

Kishon Vijay Abraham
Suravee Suthikulpanit

Agenda

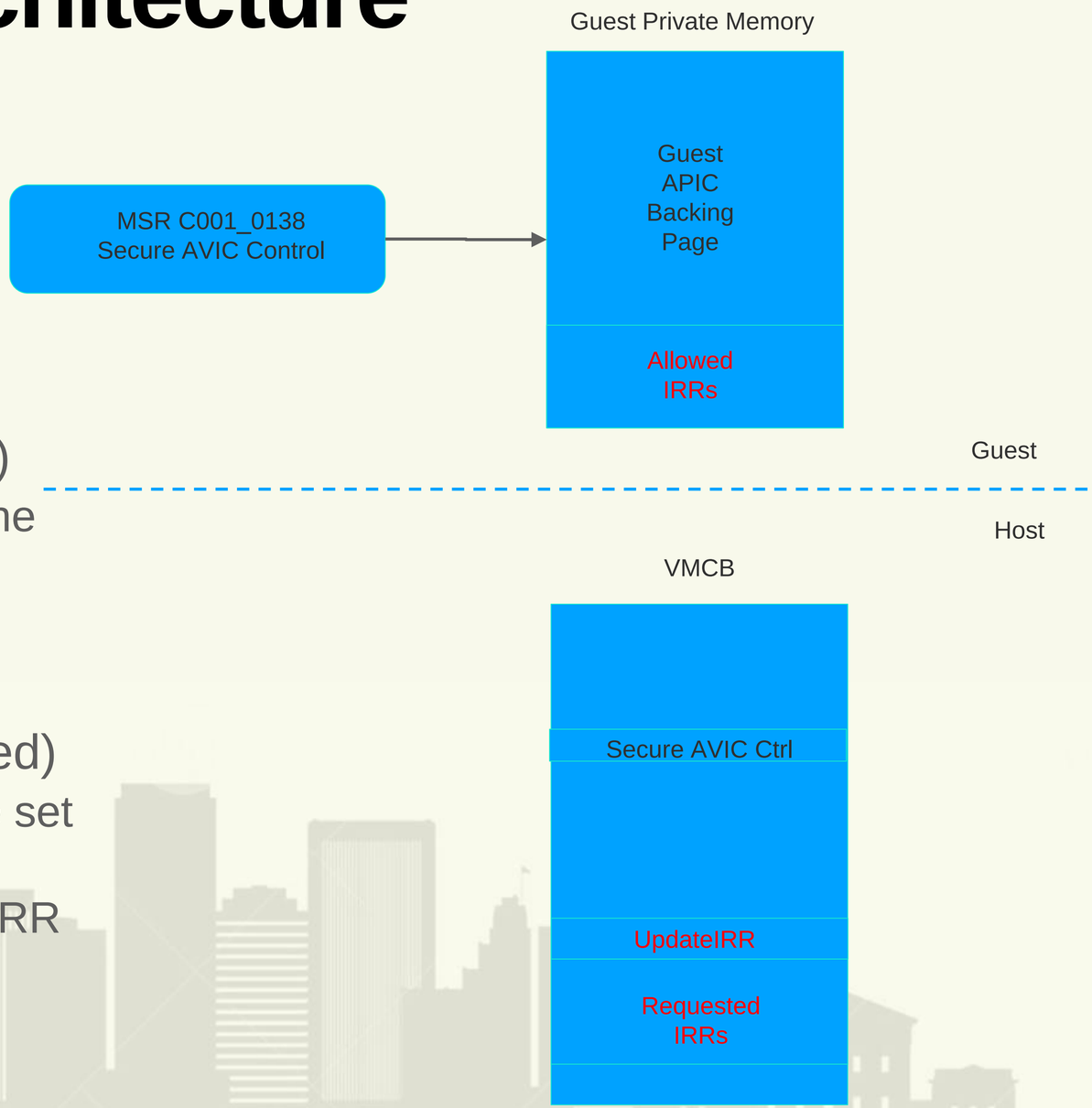
- Introduction
- Hardware & Software Architecture
- Linux Host / Guest Initialization
- Interrupt Injection Supports
  - IPI interrupts
  - Device interrupts
- Current Status & Issues

Introduction

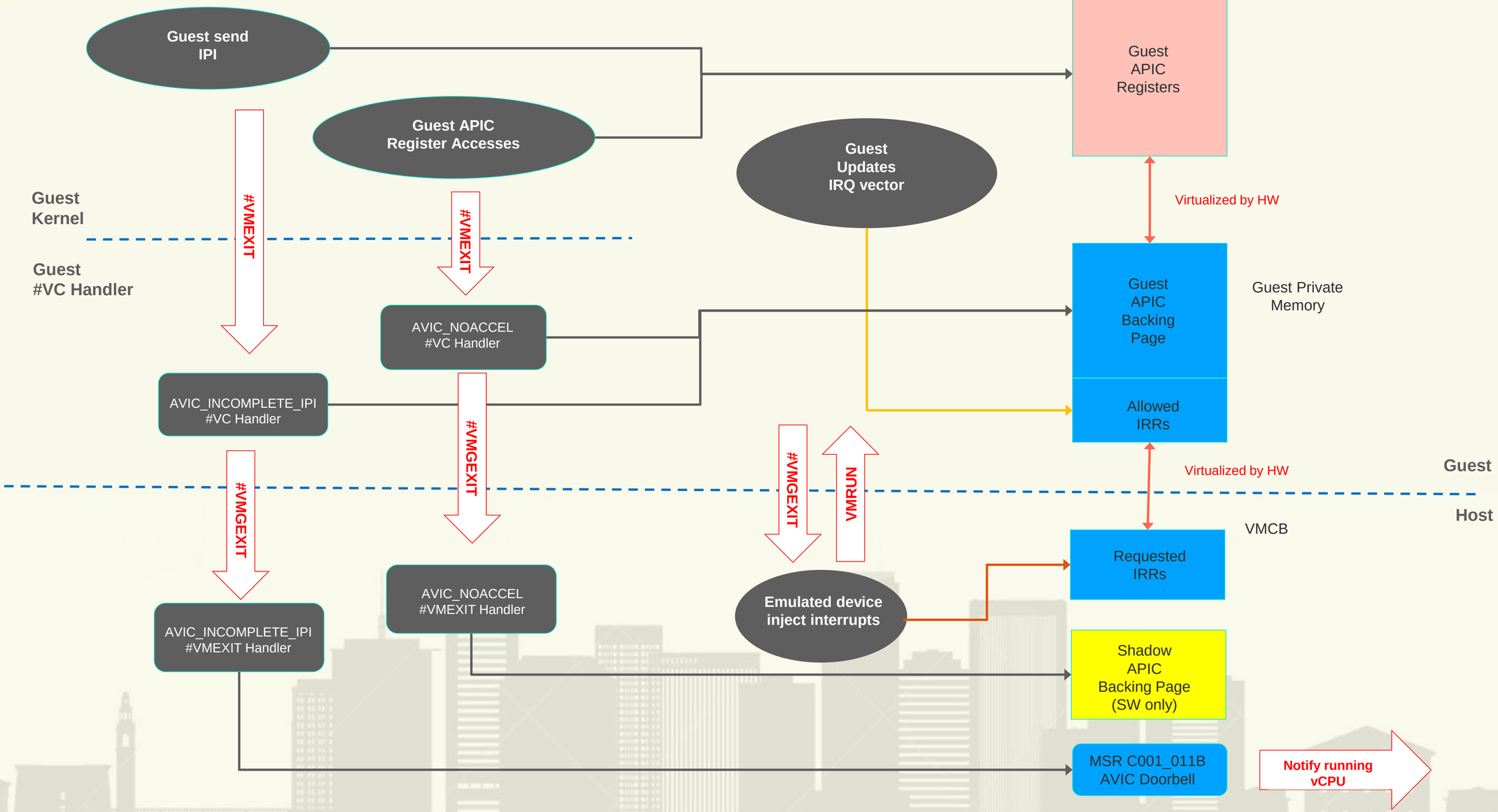
- Added security for guest APIC registers for SEV-SNP guests
- HW acceleration for performance sensitive APIC register accesses
  - Initially support only Self-IPI and EOI virtualization
- Single vs. Multi-VMPL usage models
  - Currently leverage single-VMPL w/ enlightened guest.
  - #VC handler is responsible for emulating additional functionality
- Only support x2APIC mode via x2APIC MSRs

HW Architecture

- Guest APIC Backing Page
  - Allocated and managed by Guest
  - Host APIC backing page allocation still exist
    - Cache pending interrupts in IRR
- AllowedIRR[0..7] Registers (Guest-controlled)
  - New field to indicate the interrupt vectors which the guest
    allows the hypervisor to send.
- RequestedIRR[0..7] Registers (Host-Controlled)
  - Each set bit in RequestedIRR registers, would be set in the APIC backing page IRR registers by the microcode if the same bits are set in the AllowedIRR registers.



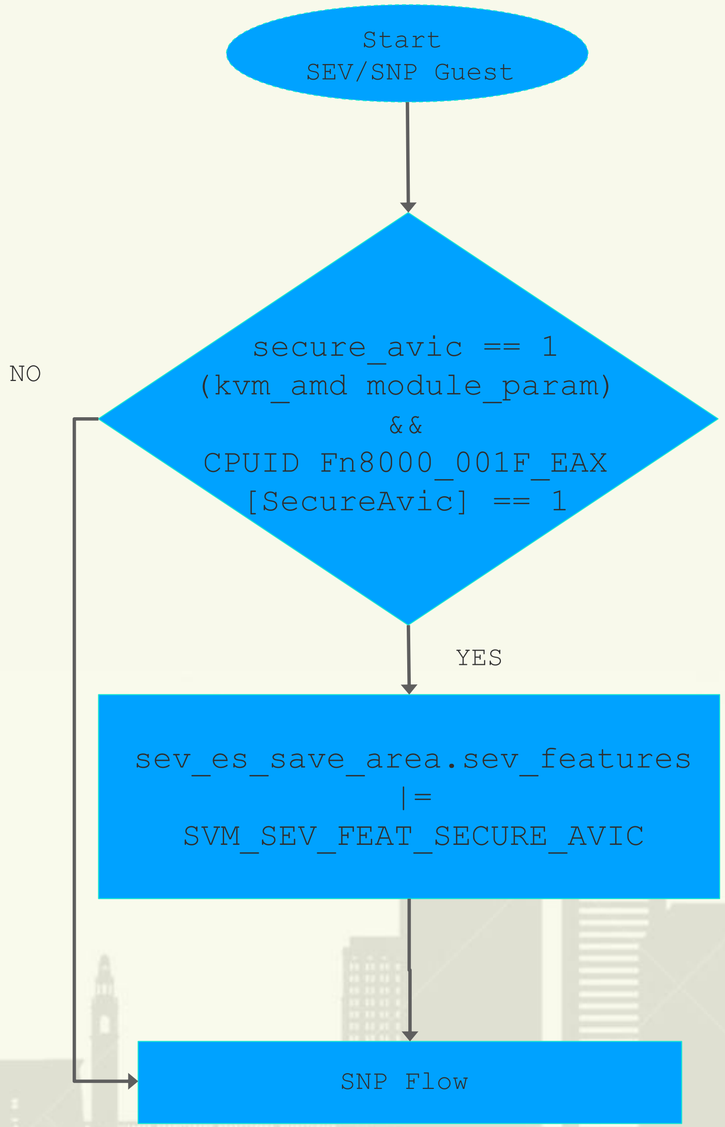
SW Architecture



Linux APIC Virtualization Comparison

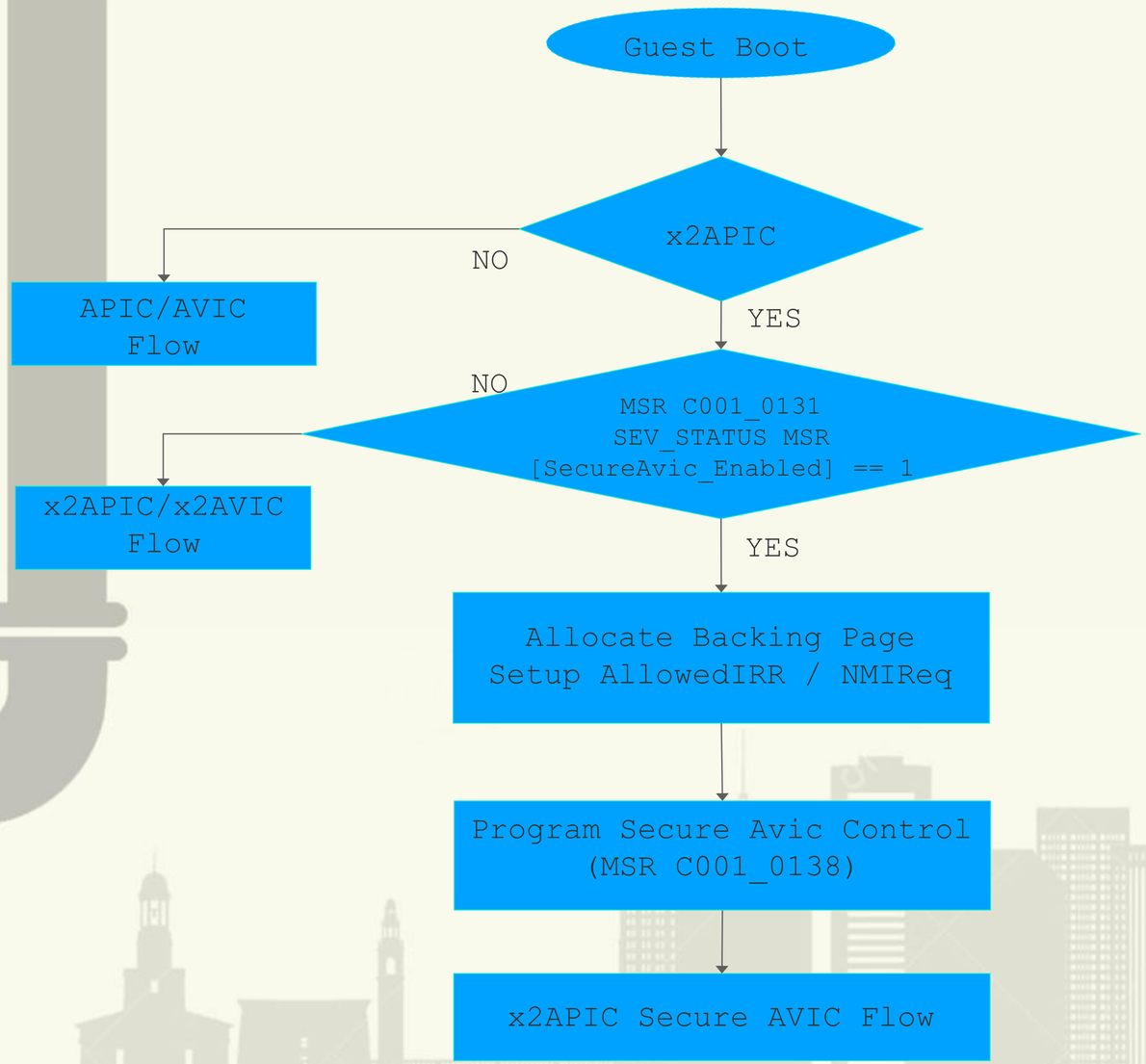
	Emulated x2APIC	x2AVIC	Secure AVIC
APIC Backing Page	Owned by Hypervisor	Owned by Hypervisor	**Owned by Guest**
Register Access	All Read and Write: VMEXIT and Handled by hypervisor	• Most Reads are accelerated by HW except extended APIC register • Most writes results in VMEXIT except ICR, TPR, EOI register	• **All accesses to ICR, TPR and EOI registers are accelerated.** • **For other registers, access must be handled by #VC handler**
Self IPI	VMEXIT and injected via event injection	Accelerated by HW	Accelerated by HW
Broadcast IPI (All Including Self)	VMEXIT and injected via event injection	• Accelerated by HW if target vCPU is running • VMEXIT AVIC_INCOMPLETE_IPI to reschedule vCPU.	• **Self IPI is accelerated by HW** • #VC handler for all other target vCPU • VMGEXIT to ring doorbell or wakeup target vCPU (by hypervisor)
Broadcast IPI (All Excluding Self)	VMEXIT and injected via event injection	• Accelerated by HW if target vCPU is running • VMEXIT AVIC_INCOMPLETE_IPI to reschedule vCPU.	• #VC handler for all target vCPU • VMGEXIT to ring doorbell or wakeup target vCPU (by hypervisor)
Emulated Device Interrupt Injection	VMEXIT and injected via event injection	• HV inject by updating APIC IRR register • HV ring doorbell or wakeup target vCPU	• **Guest updates AllowedIRR** • **HV updates RequestedIRR and UpdateIRR** • HV ring doorbell or wakeup target vCPU
Multiple Pending Interrupt	VMEXIT for each interrupt using VINTR EXIT	HW invokes ISR for each set IRR	**HW invokes ISR for each set bits of [Allowed | Requested] IRRs.**

Initialization: Secure AVIC (Host)



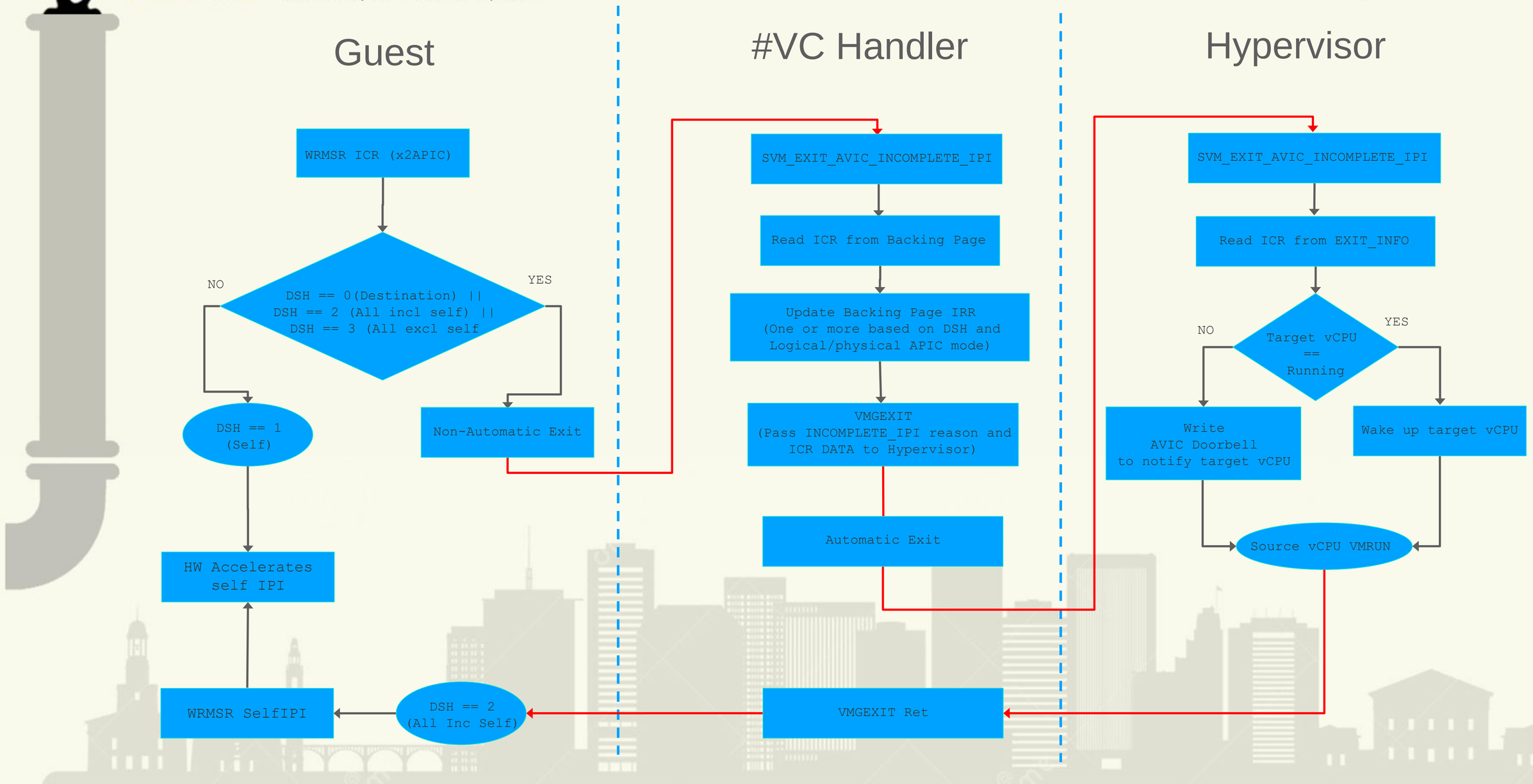
- New kvm_amd kernel module option:
  - kvm_amd.secure_avic
    (mutual exclusive to the avic parameter)
- Checking CPUID for the feature support
- Enabling the feature
  - Set the SEV_FEATURES[SecureAvic] bit in the VMSA.

Initialization: Secure AVIC (Guest)

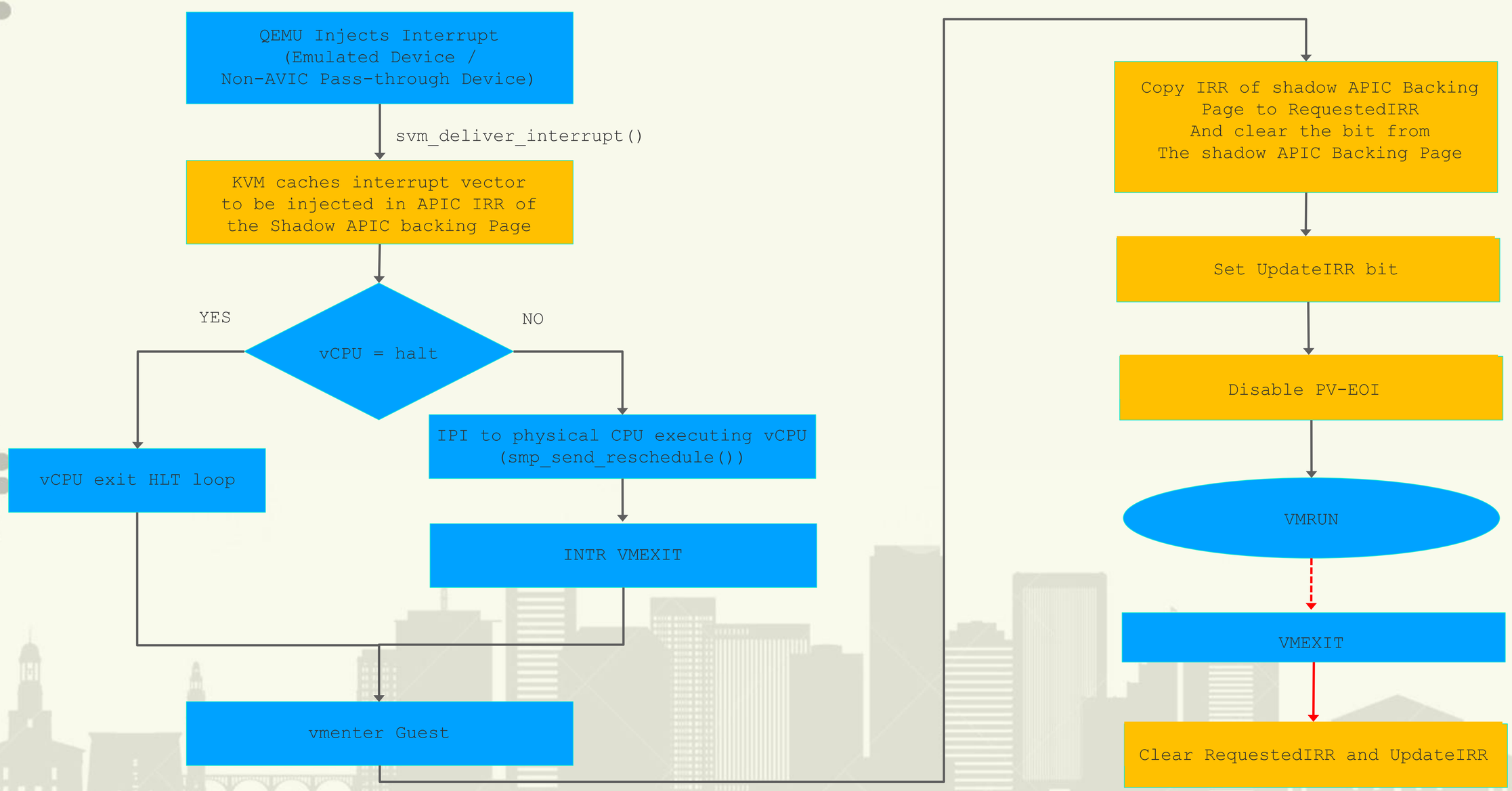


- Check MSR_C0010131 SEV STATUS [SecureAvicEn] bit
  - Indicates Secure AVIC support for guest
- Allocate Guest APIC Backing Page
  - Program backing page 4k-aligned GPA into the MSR_C001_0138[GuestApicBackingPagePtr]
  - Pinned in while in Guest mode (during vmrun)
  - Page is marked private (encrypted) in RMP table.
- Enabling Secure AVIC feature
  - Set MSR_C0010138[SecureAvicEn] bit

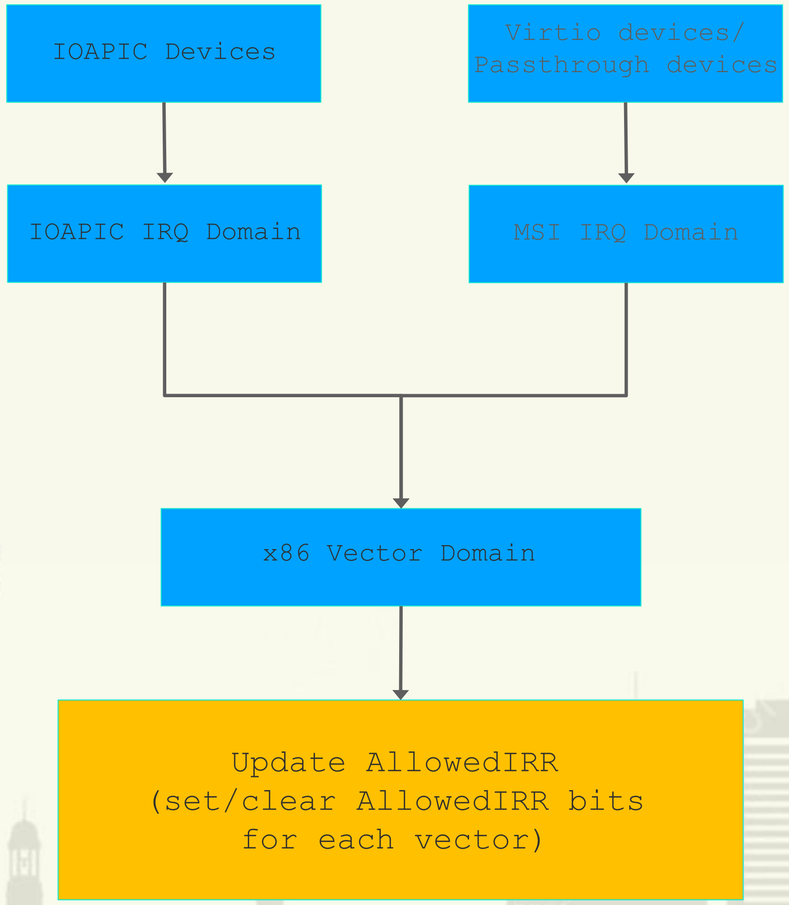
IPI Injection



Host Device Interrupt Injection



Guest Device Interrupt Injection



- **arch/x86/kernel/apic/vector.c is modified to update the AllowedIRR registers.**
- Microcode sets IRR bit in APIC backing page based on AllowedIRR and RequestedIRR
- Microcode sets ISR bit in APIC backing page (guest) after interrupt handler is invoked
- When Guest Kernel writes to EOI register, microcode clears ISR bit and invokes interrupt handler of the next set IRR

Status

- Phase #1 : (Currently work-in-progress)
  - IPI Interrupt Injection support
  - Emulated device interrupt injection support
  - Successfully boot a VM w/ 128 vCPUs
- Phase #2 :
  - NMI Emulation
  - LAPIC Timer Emulation

Issues

- Secure AVIC hardware requires PTE entry for guest APIC backing page to always be present in the nested page table
  - There should be no NPF Exception when HW accesses backing page.
- KVM MMU zaps PTE frequently causing NPF resulting in VMEXIT_BUSY and prevents the vcpu to resume.
- Temporary Workaround
  - Invoke kvm_tdp_mmu_map() to populate PTE of backing page in the page table before VMRUN.
  - However, this workaround does not always guarantee the page to be present.

Q & A