AMD

PROTECTING VM REGISTER STATE WITH AMD SEV-ES

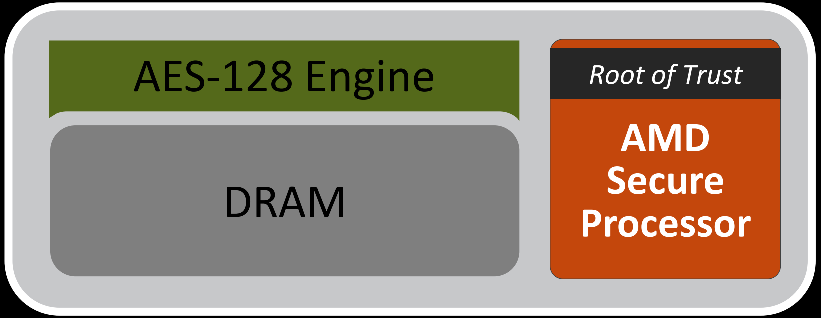
DAVID KAPLAN
LSS 2017

BACKGROUND-- HARDWARE MEMORY ENCRYPTION

AMD

AMD Secure Memory Encryption (SME) / **AMD Secure Encrypted Virtualization** (SEV)

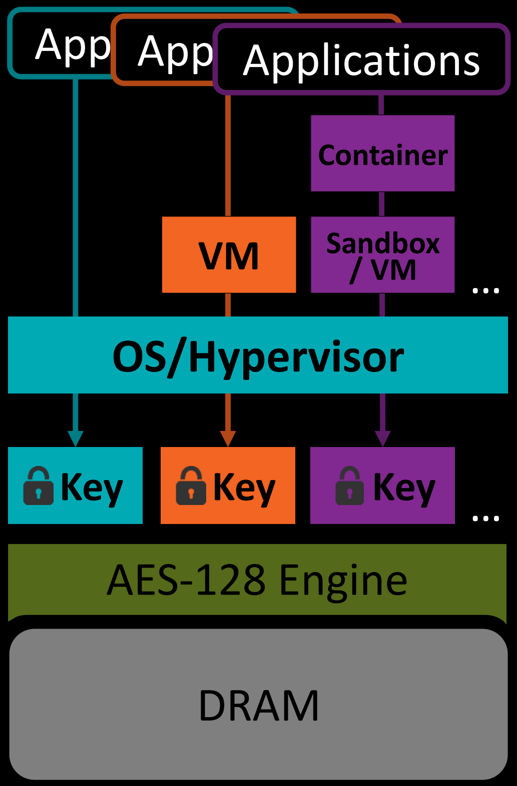
- Hardware AES engine located in the memory controller performs inline encryption/decryption of DRAM
- Minimal performance impact
  - Extra latency only taken for encrypted pages
- No application changes required
- Encryption keys are managed by the AMD Secure Processor and are hardware isolated
  - not known to any software on the CPU



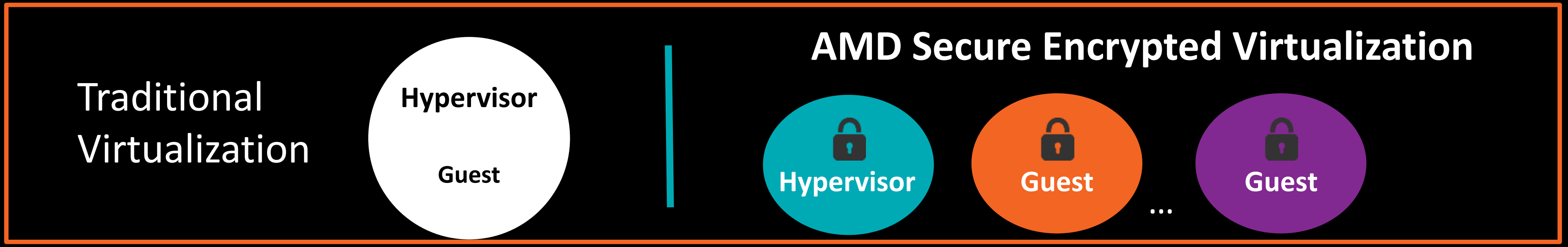
Defense against unauthorized access to memory

BACKGROUND - SECURE ENCRYPTED VIRTUALIZATION (SEV)

- Protects VMs/Containers from each other, administrator tampering, and untrusted Hypervisor
- One key for Hypervisor and one key per VM, groups of VMs, or VM/Sandbox with multiple containers
- Cryptographically isolates the hypervisor from the guest VMs
- Integrates with existing AMD-V technology
- System can also run unsecure VMs



Enhances isolation of VMs

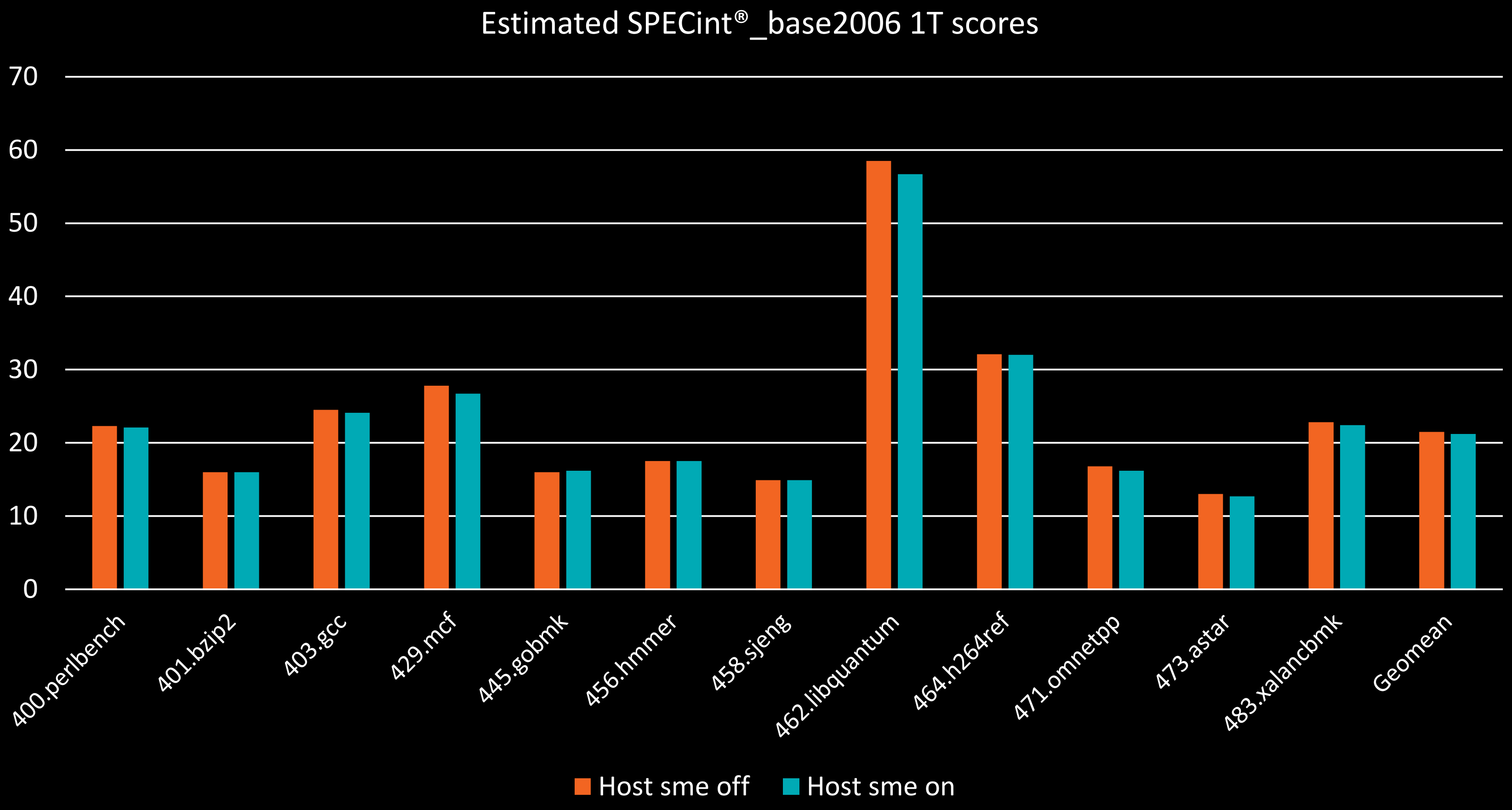

Traditional Virtualization
Hypervisor
Guest
AMD Secure Encrypted Virtualization
Hypervisor
Guest
…
Guest

UPDATES SINCE LSS 2016

- Hardware is available!
  - Ryzen/ThreadRipper support TSME/SME only
  - EPYC supports SEV as well
  - Demo of SEV in action: https://youtu.be/qgiUuTmXyGs (just search for “amd security”)
- Linux support underway
  - OVMF (BIOS) patches accepted 7-10-2017 (https://github.com/tianocore/edk2/commits?author=codomania)
  - SME Linux kernel patches accepted 7-18-2017 (likely to be included in 4.14)
  - SEV Linux kernel patches under RFC
- Please help review patches!

SOME PERFORMANCE DATA (SME)

Estimate based on data collected on config:
- OS Ubuntu 16.04 running stock kernel: 4.10
- BIOS WDL7628N, release Date: 06/26/2017
- EPYC 2.2 GHz fixed frequency, SMT on
- Host memory 512GB @ 2667MHz, 64GB per socket for host.
- Compiled with GCC 6.1



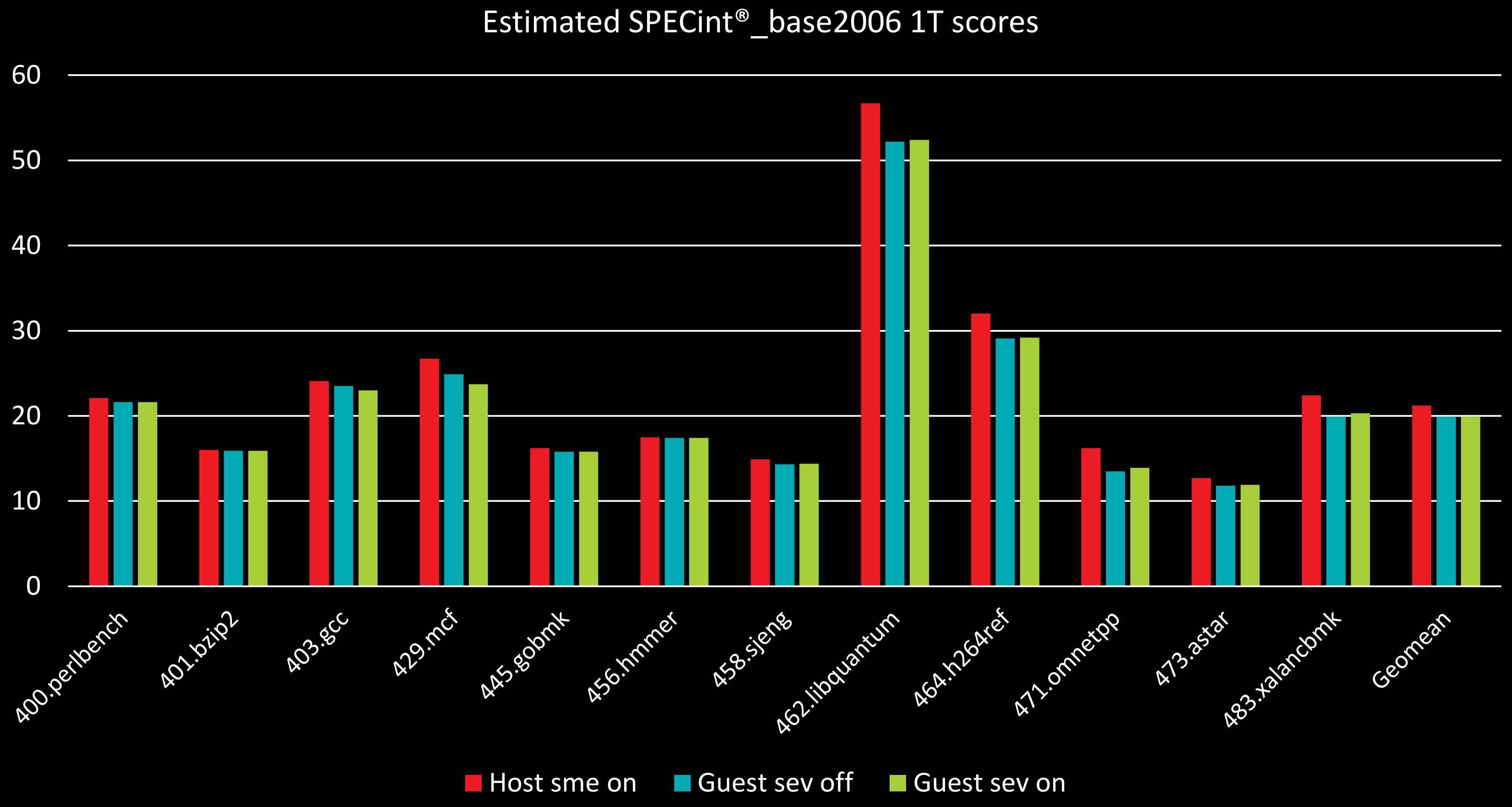
- Geomean: -1.40%
- Worst (mcf): -3.96%

More information about SPEC CPU ® 2006 can be found at http://www.spec.org

SEV PERFORMANCE

Estimate based on data collected on config:
- •OS Ubuntu 16.04 running stock kernel: 4.10
- •BIOS WDL7628N, release Date: 06/26/2017
- •EPYC 2.2 GHz fixed frequency, SMT on
- •Host memory 512GB @ 2667MHz, 64GB per socket for host.
- •Compiled with GCC 6.1
- •KVM/QEMU

SEV kernel:
https://github.com/AMDESE/AMDSEV



▶ Guest vs Host: -6.13% average (for both SEV on/off)

More information about SPEC CPU ® 2006 can be found at http://www.spec.org

STREAM PERFORMANCE

Host system setup

OS Ubuntu 16.04 running kernel: 4.13.0-rc1-sev-rfc-3-2

BIOS WDL7628N, release Date: 06/26/2017

EPYC Silicon at 2.2 GHz fixed frequency, SMT on

Host memory 512GB @ 2667MHz, 64GB per socket for host.

Guest config

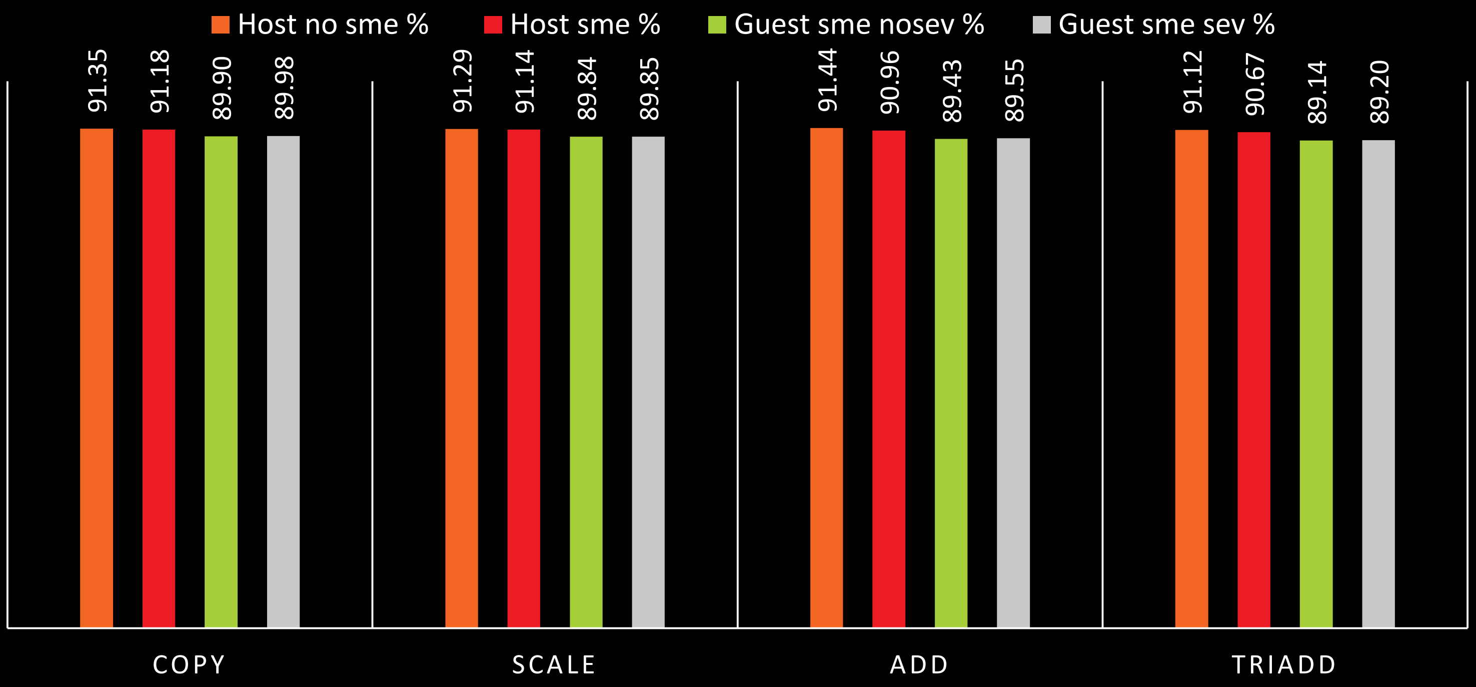
SW config same as host

HW: 4 vcpus, 95% of 20GB memory

qemu process bound to node 0 and die 0 cpus

each vcpu thread taskset to a unique core of die 0

STREAM 4T (% OF THEORETICAL MAX)



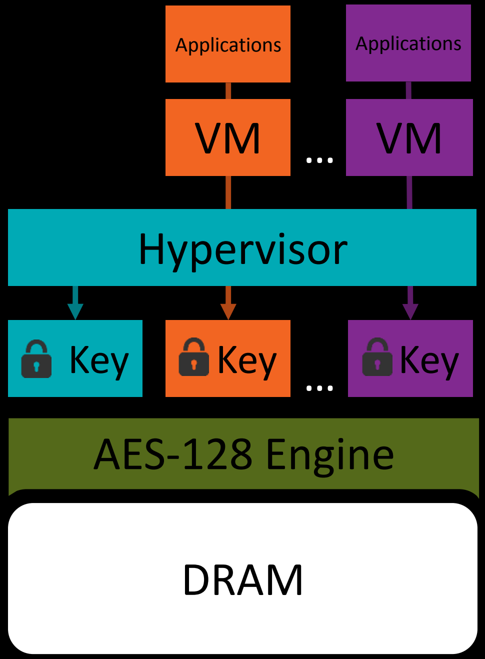
	Host SME	Guest NoSEV	Guest SEV
COPY	-0.17	-1.45	-1.37
SCALE	-0.15	-1.45	-1.44
ADD	-0.48	-2.01	-1.90
TRIADD	-0.45	-1.98	-1.92

INTRO TO SEV-ES

- SEV-ES (Encrypted State) provides additional VM security on top of AMD SEV memory encryption

- SEV protects guest memory using memory encryption
- SEV-ES protects guest register state
  - Register state is encrypted using guest memory encryption key
  - Only guest is allowed to modify its register state
  - Register state is integrity protected to prevent rollback attacks

- New architectural features allow guests to selectively allow HV access to state when needed for VM emulation purposes



THREAT MODEL

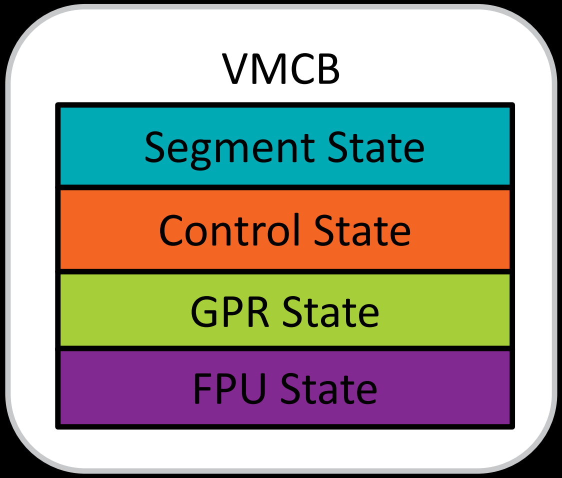
- Like SEV, SEV-ES requires trust in the HW, AMD Secure Processor, and guest VM

- In SEV-ES, the HV is less trusted than SEV as it is only able to do the following
  - Run guest VMs (SEV/SEV-ES do not protect against DOS)
  - Manage memory allocation (maintain nested tables)
  - Inject interrupts/exceptions into guest
  - Emulate devices/services as requested by the guest

- In particular, SEV-ES protects against attacks such as
  - Exfiltration (HV observing guest register state during exits)
  - Control flow (modifying guest register state to change control flow)

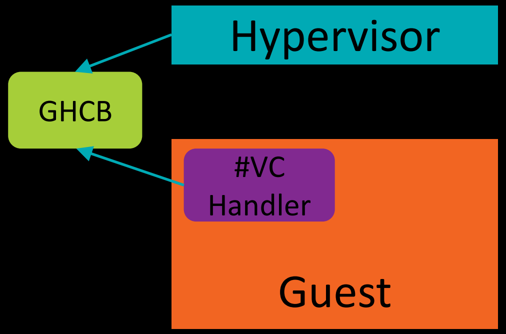
ARCHITECTURE AT A GLANCE

- World switches now swap ALL register state
  - Includes all segment registers, GPRs, FPU state (see Table B-4 i APM Vol2)
  - All register state is encrypted with the guest encryption key
  - Integrity value is calculated and stored in a protected page



- The guest is notified by a new exception (#VC) when certain events occur
  - The guest decides what state (if any) to share with the HV
  - The guest invokes the HV to perform the required tasks
  - The guest updates its state based on the output from the HV

- The guest and HV use a special structure to communicate
  - Guest-Hypervisor Communication Block (GHCB)
  - Location set by guest, mapped as unencrypted memory page

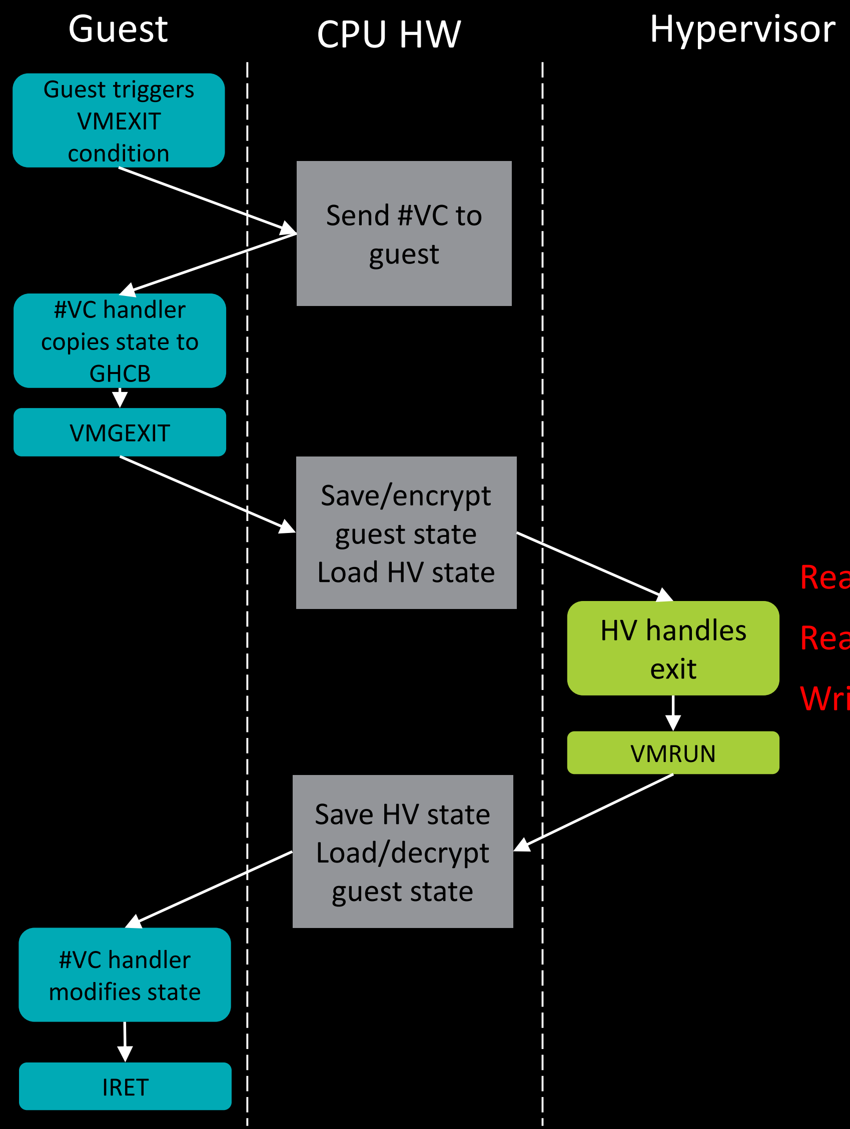


TYPES OF EXITS

- **Automatic Exits (AE)**
  - Events that occur asynchronously to the guest (e.g. interrupts)
  - Events that do not require exposing guest state (e.g. HLT)
  - Nested page faults not due to MMIO emulation
  - AE events save all state and exit to HV
  - Only action HV can do is just resume the guest w/o modifications
- **Non-Automatic Exits (NAE)**
  - All other exit events
  - NAE events cause a #VC instead of a VMEXIT
  - Guest handler may invoke the HV via VMGEXIT instruction

Code	Name	Notes	HW Advances RIP
52h	VMEXIT_MC	Machine check exception	No
60h	VMEXIT_INTR	Physical INTR	No
61h	VMEXIT_NMI	Physical NMI	No
63h	VMEXIT_INIT	Physical INIT	No
64h	VMEXIT_VINTR	Virtual INTR	No
77h	VMEXIT_PAUSE	PAUSE instruction	Yes
78h	VMEXIT_HLT	HLT instruction	Yes
7Fh	VMEXIT_SHUTDOWN	Shutdown	No
8Fh	VMEXIT_EFER_WRITE_TRAP	Write to EFER	Yes
90h-9Fh	VMEXIT_CR[0-15]_WRITE_TRAP	Write to CRx	Yes
400h	VMEXIT_NPF	Only if PFCODE[3]=0 (no reserved bit error)	No
403h	VMEXIT_VMGEXIT	VMGEXIT instruction	Yes
-1	VMEXIT_INVALID	Invalid guest state	-

NAE FLOW EXAMPLE



Guest

CPU HW

Hypervisor

CPUID

Guest triggers VMEXIT condition

Send #VC to guest

Read ErrorCode => CPUID

Write CPUID_EXIT to GHCB

Copy RAX to GHCB

#VC handler copies state to GHCB

VMGEXIT

Save/encrypt guest state Load HV state

HV handles exit

Read GHCB => see CPUID_EXIT

Read RAX, emulate CPUID

Write RAX/RBX/RCX/RDX to GHCB

VMRUN

Save HV state Load/decrypt guest state

Copy RAX/RBX/RCX/RDX to register state

#VC handler modifies state

IRET

GUEST-HYPERVISOR COMMUNICATION BLOCK (GHCB)

- To facilitate HV/OS interoperability, AMD is working on defining a GHCB format/contract
  - GHCB layout will mirror the VMCB layout
  - Guest OS is expected to supply certain values on certain exceptions (e.g. RDMSR requires RCX)
- GHCB specification is in development, will be open for comments shortly

- A new MSR defines the location of the GHCB, value is per-guest
  - On boot, this MSR will contain information about the SEV configuration
  - Once ready, the guest will write the MSR with the guest physical address of the GHCB

MMIO

- SEV-ES assumes that MMIO pages are marked with a reserved bit set in the nested tables
  - This is what KVM does today
  - Other page faults (e.g. not present) are handled as AEs

- A guest d-side access that encounters a reserved page fault throws a #VC
  - Guest #VC handler must read RIP and determine what access is required
  - Guest #VC handler calls HV to read/write MMIO bytes as required
  - Hypervisor does not crack/emulate instruction since guest #VC handler does this

#VC HANDLER OPTIMIZATIONS

- The #VC handler may be used to reduce total world switches needed

- Example: Avoid VMEXIT for static values
  - After first CPUID, remember results and use them in the future
  - Avoids user programs from taking CPUID VMEXITs

- Example: Fine-grained MMIO traps
  - #VC handler checks page offset and decides if it merits a VMEXIT
  - Could allow for write coalescing (group many MMIO updates into one VMEXIT)
  - Optimize MMIO reads with static results

SEV-ES IN LINUX

- First priority is to finalize GHCB software format/conventions

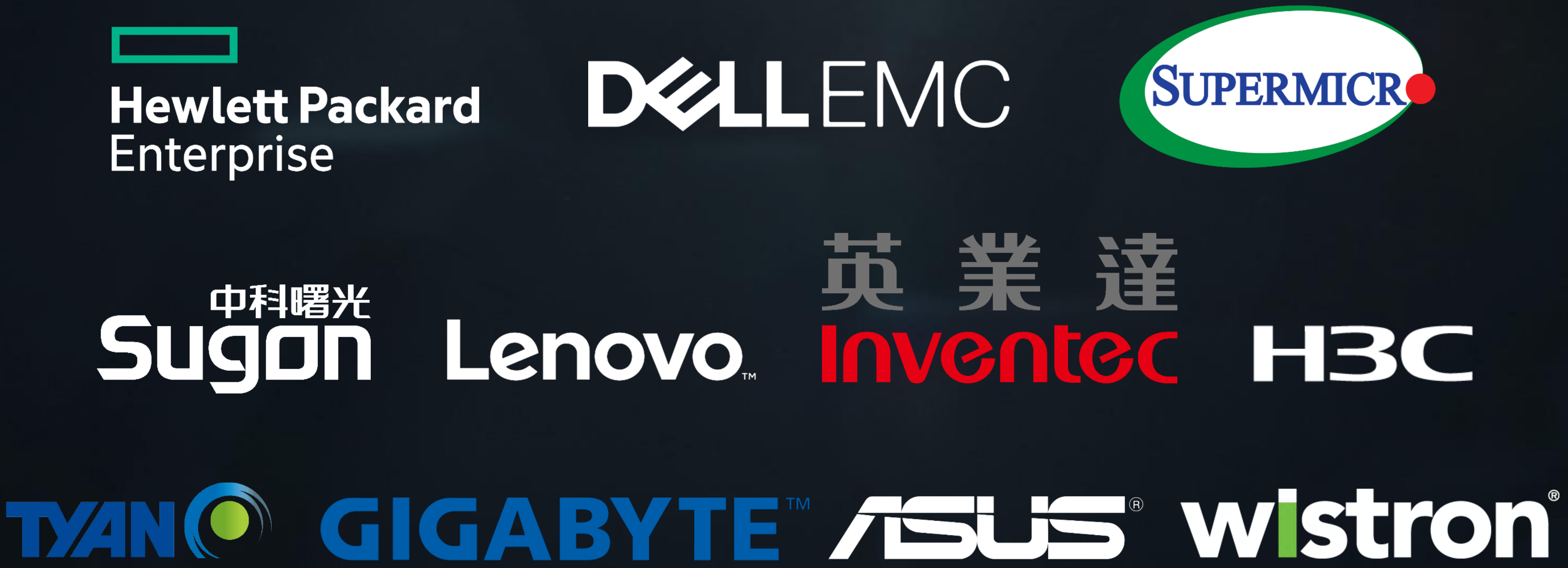
- KVM
  - Support for atomic world switch
  - Read/write register values from GHCB instead of VMCB
  - Support for new exits (e.g. read/write MMIO)
  - Call to AMD Secure Processor to initialize/measure initial VMCB state

- (Guest) Kernel
  - New #VC exception handler
  - Instruction cracking for #VC handler

DOCUMENTATION

- Whitepapers
  - SEV-ES: http://support.amd.com/TechDocs/Protecting%20VM%20Register%20State%20with%20SEV-ES.pdf
  - SEV: http://developer.amd.com/wordpress/media/2013/12/AMD_Memory_Encryption_Whitepaper_v7-Public.pdf
- Technical Documentation
  - AMD64 Manual (vol2): http://support.amd.com/TechDocs/24593.pdf
    - SEV: Section 15.34
    - SEV-ES: Section 15.35
  - Key Management API: http://support.amd.com/TechDocs/55766_SEV-KM%20API_Specification.pdf
- Code
  - GitHub: https://github.com/AMDESE/AMDSEV

EPYC PLATFORM ECOSYSTEM



DISCLAIMER & ATTRIBUTION

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions and typographical errors.

The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION.

AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY DIRECT, INDIRECT, SPECIAL OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

ATTRIBUTION