Implementing AMD SEV in Xen hypervisor

VATES

AGENDA

VATES

1. AMD SEV technology
2. AMD Secure Processor
3. Platform finite State Automata
4. Guest finite State Automata
5. Xen Hypercalls
6. I/O Emulation & Instruction decode
7. What’s next…

AMD SEV Technology

- Confidential Computing technology from AMD wich targets VM environnement
- an extension to AMD-V
- Comes in building blocks

AMD SEV Technolgy

SME

- 1 AES Key
- C-bit
- Encrypt in Place

SEV

- Amd Secure Processor
- VEK (bound to ASID)
- shared/private pages for CR.PAE
- ins fetch & HW ptwalk are encrypted
- encryption is bound to PA
- no DMA to private pages

SEV-ES

- All CPU/FPU context is encrypted (VMSA)
- Automatic/NonAutomatic Exits
- #VC handler in guest
- VMGEXIT instruction
- GHCB protocol

SEV-SNP

- Integrity Control
- RMP table (and ISA extensions)
- Bunch of other stuff

SEV-TIO

- PCI SIG TDISP
- ASP-TIO interface (SPDM)

AMD Secure Processor

VATES

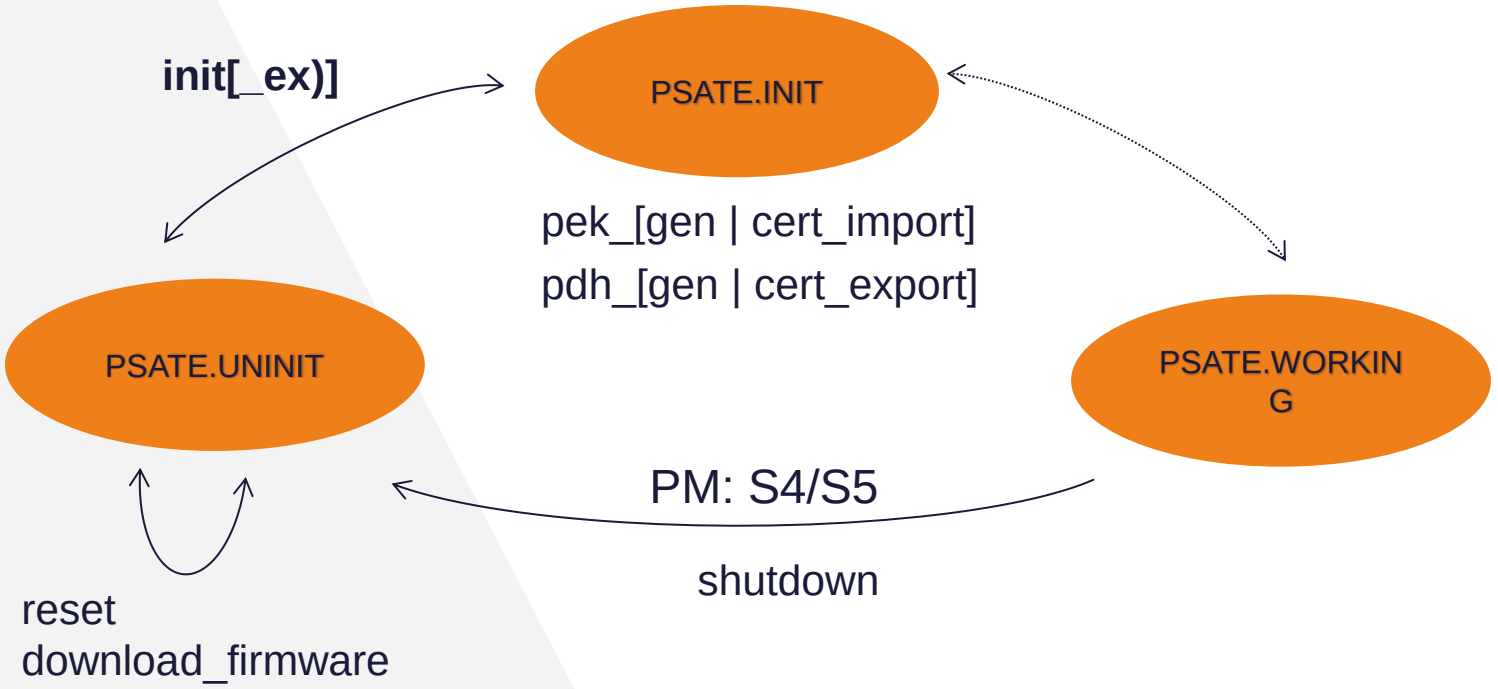
ASP:

- PCI device
- multiple Interfaces (one bdf)
- mailbox protocol

API:

- platform management
- guest management
- remote attestation support
- guest migration
- misc: copy/swap

Platform Finite State Automata

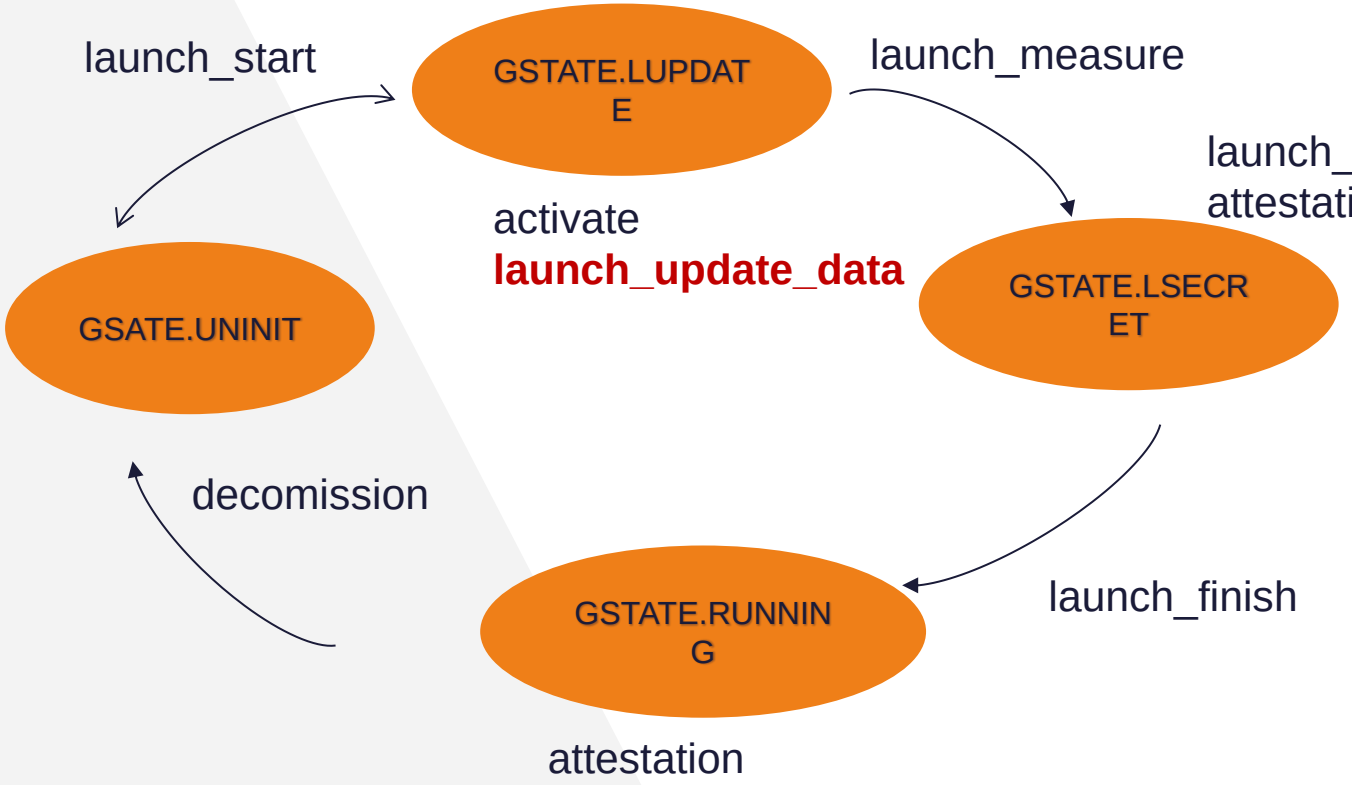


PCTX

NVRAM

- PEK key pair
- PDH key pair
- PDH Cert
- …

Guest Finite State Automata



GCTX

- state
- handle
- ASID
- policy
- VEK
- nonce (for TC)
- Master secret
- LaunchDigest

Xen Hypercalls & Shared Memory

- **hypercall new ABI :** no more linear addresses
  - copy_[from | to]_user]_hvm
- **hypercall page :** dynamically constructed
  - hardcoded/work in progress (need probably better solution for this)
- **hypercall buffers :** need to be SHARED memory
  - work in progress (SEV-guest)
- **shared memory need to be SEV-wise shared**
  - work in progress (SEV-guest)

Instruction Decode & I/O Emulation

I/O space accesses (VMEXIT_IOIO):

- simple:
  - *in [al | ax | eax], imm8*
  - *in [al | ax | eax], DX*
  - *out imm8, [aL | ax | eax]*
  - *out DX, [aL | ax | eax]*
- string:
  - ins/outs : virtual addresses in RSI/RDI

Decode Assist:

EXITINFO1: port, dir, size, str

EXITINFO2: nrip

MMIO accesses (VMEXIT_NPF) :

- 1 mem operand: (~mov)
  - NPF faulting GPA(the mem operand)
  - error code

Decode Assist:

- instruction length
- instruction bytes[]

some corner cases can't be handled!

(page tables, ins fetch, smap/smep errate, ..)

- 2 mem operand

movs(x), movdir64,… push, pop, call,…

Instruction Decode & I/O Emulation

Exceptions: (VMEXIT_EXCEPTION_XXX)

traped: DB, BP, **NM**, PF, AC, **UD**, MC

#UD - can't handle: must disable

Instructions:

MOV CRx :

- Decode assist: dir, #gpr

INVLPG:

- Decode Assist: linear address

INT n:

- Decode Assist: #n

Instructions:

CPUID/RDTSC/ RDMSR/WRMSR: OK

INVD/WBINVD/HLT: OK

TASK SWITCH/…: - can't handle

Event Injection:

SW & External Interrupts, HW exceptions, ….

OK if **SVM_FEATURE_NRIPS**

What's next

The Goal:

- run Linux in SEV-ES environnement

Major Steps:

- adapt Xen-dependant Linux kernel parts
- adapt ASP driver for SEV-ES platform & guest management
- implement GHCB protocol in XEN
- adapt misc Xen parts (emulation, hvm_function_table, …)

VATES

THANKS!

Any questions?

You can find me at andrei.semenov@vates.tech