Confidential VMs Explained: An Empirical Analysis of AMD SEV-SNP and Intel TDX

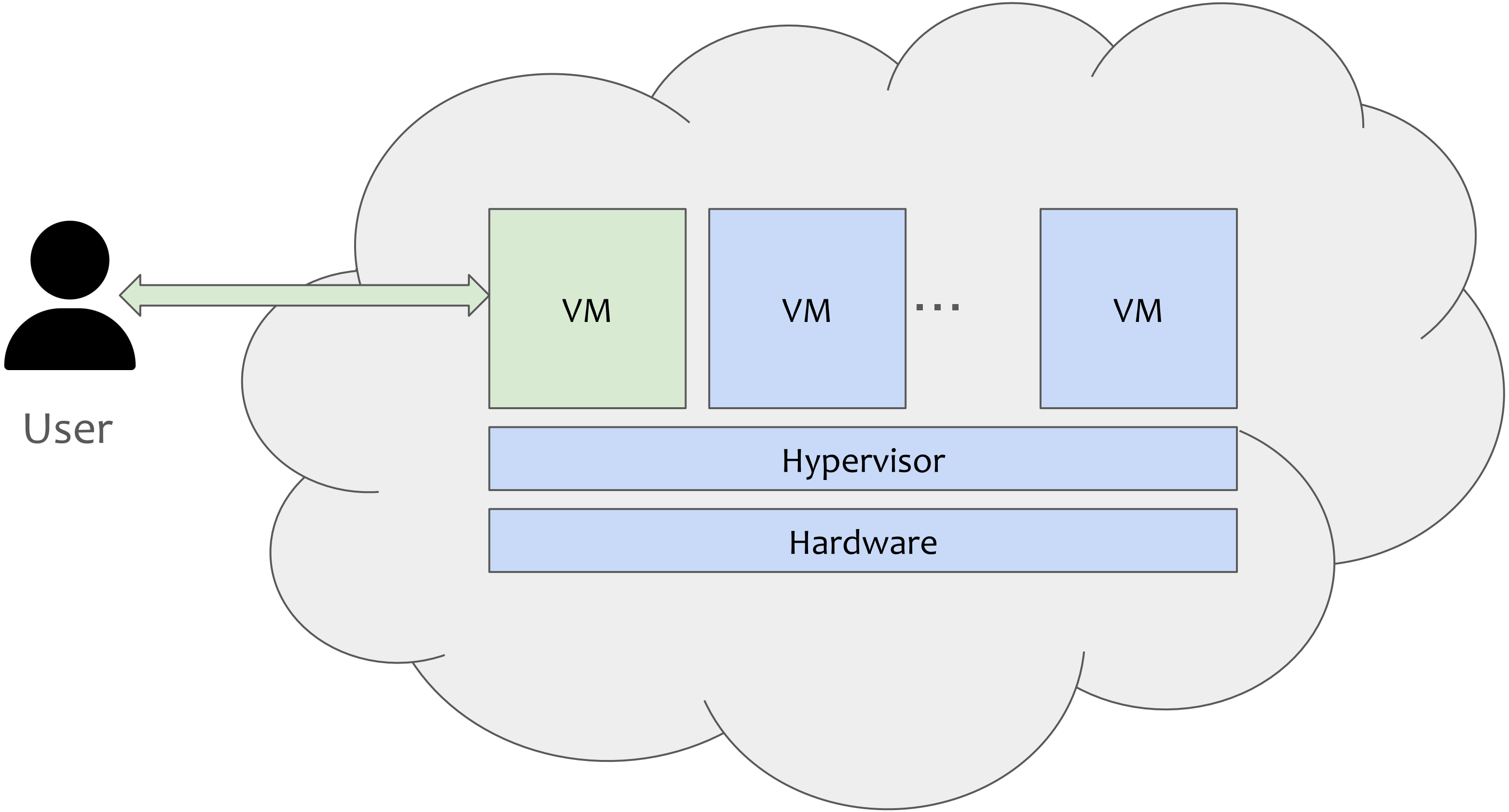
Masanori Misono, Dimitrios Stavrakakis, Nuno Santos*, Pramod Bhatotia

System Research Group, TU Munich, Germany
*SysSec Research Team, INESC-ID Lisbon/IST, Portugal

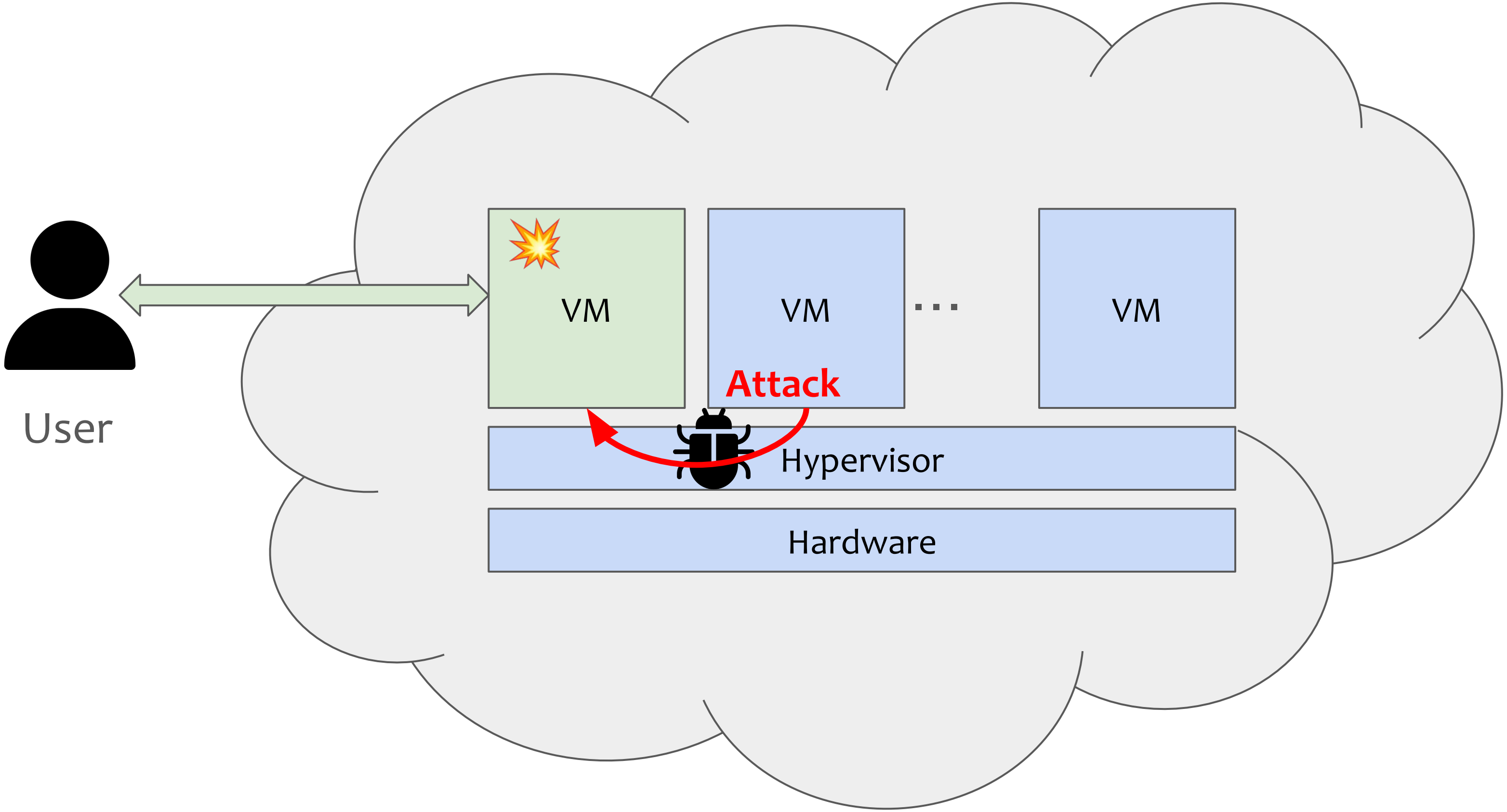
ACM SIGMETRICS 2025, New York, USA

2025-06-10

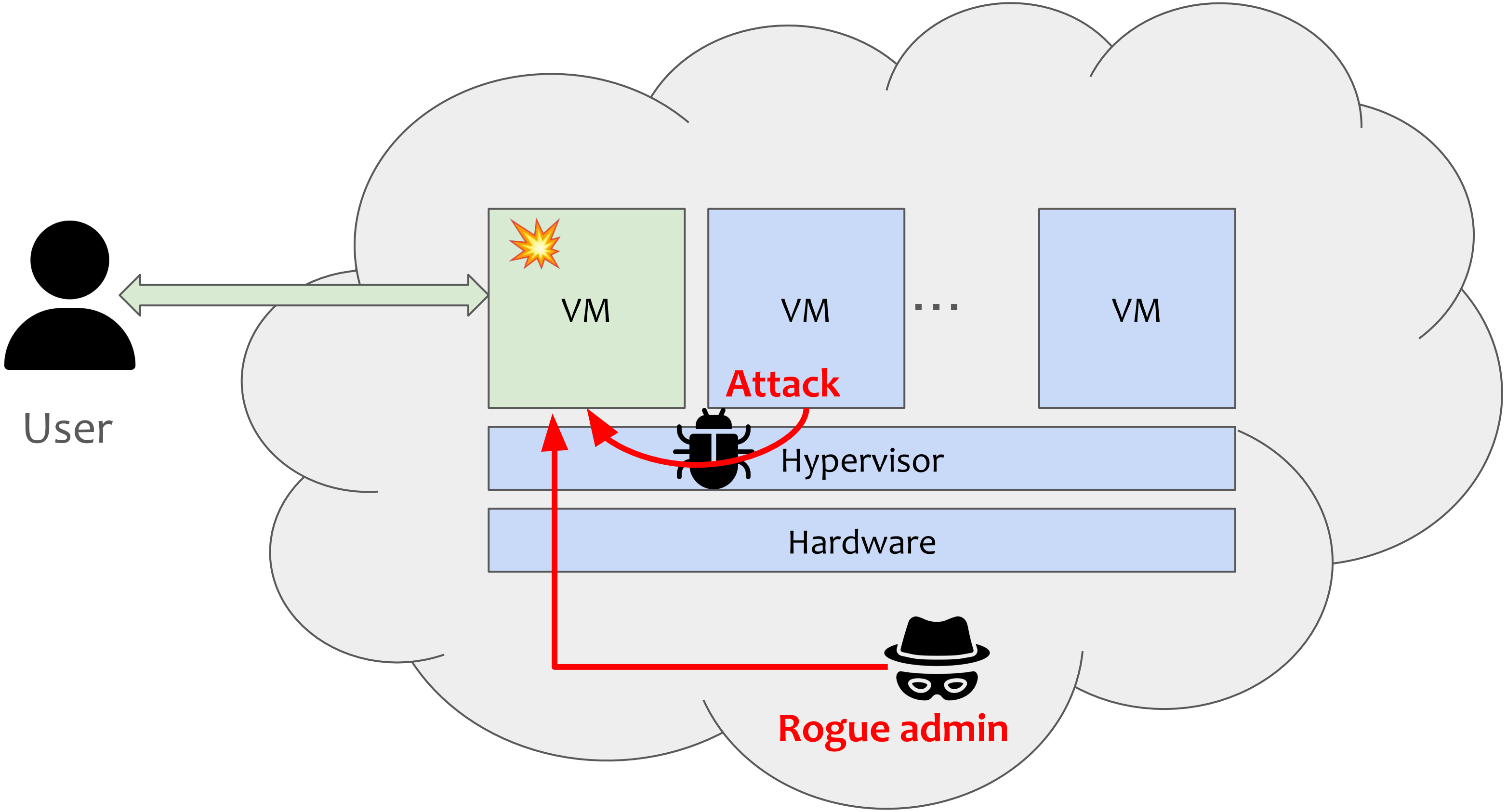
Security threat in the cloud



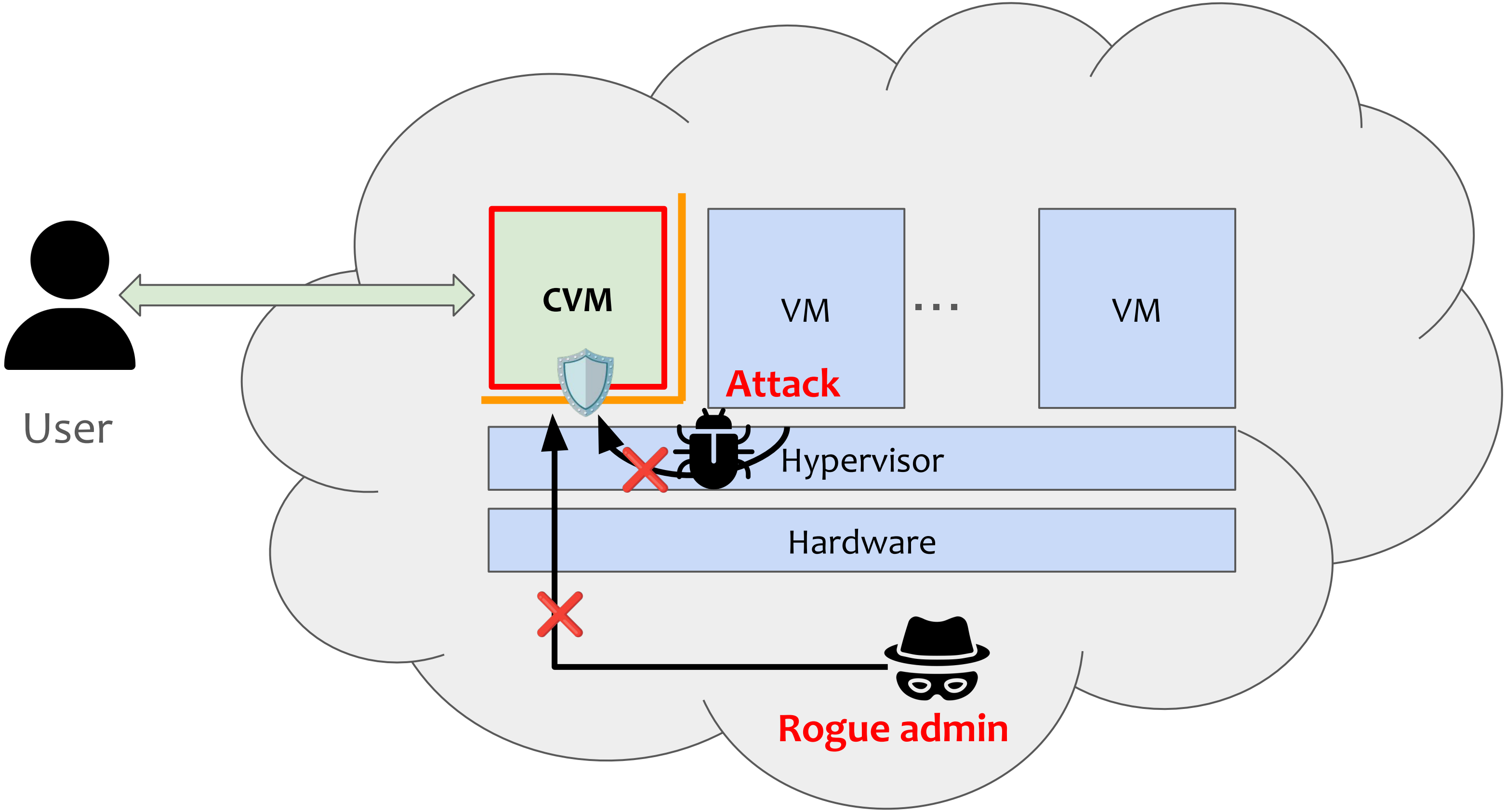
Security threat in the cloud



Security threat in the cloud



Confidential virtual machines (CVMs)



CVM technologies

AMD SEV-SNP (2021-)

<AMD SEV (1st gen) (2019-)>

Intel TDX (2023-)

<Public release (2024-)>

ARM CCA (Arm v9)

<emulator available>

CoVE

<emulator available>

Major CPU vendors offer CVM technologies

Cloud vendor support

Azure

aws

Google Cloud

ORACLE Cloud

Alibaba Cloud

IBM Cloud

Major cloud vendors start offering CVMs as a service

Motivation

- Each CVM technology has the same goal but works differently
- Newly system components introduced

Understanding characteristics and limitations is crucial for adoption

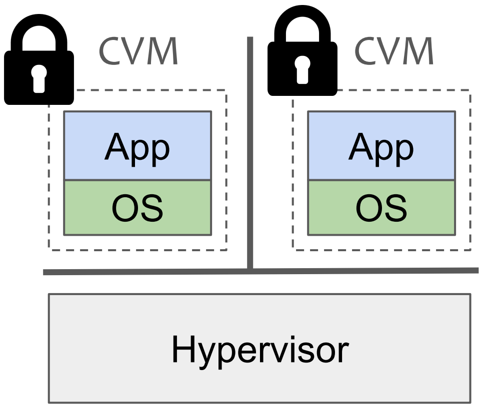
This work provides a comprehensive empirical analysis of CVMs:
<u>AMD SEV-SNP and Intel TDX</u>

Outline

- ~~Overview~~
- Background
- Evaluation
- Summary

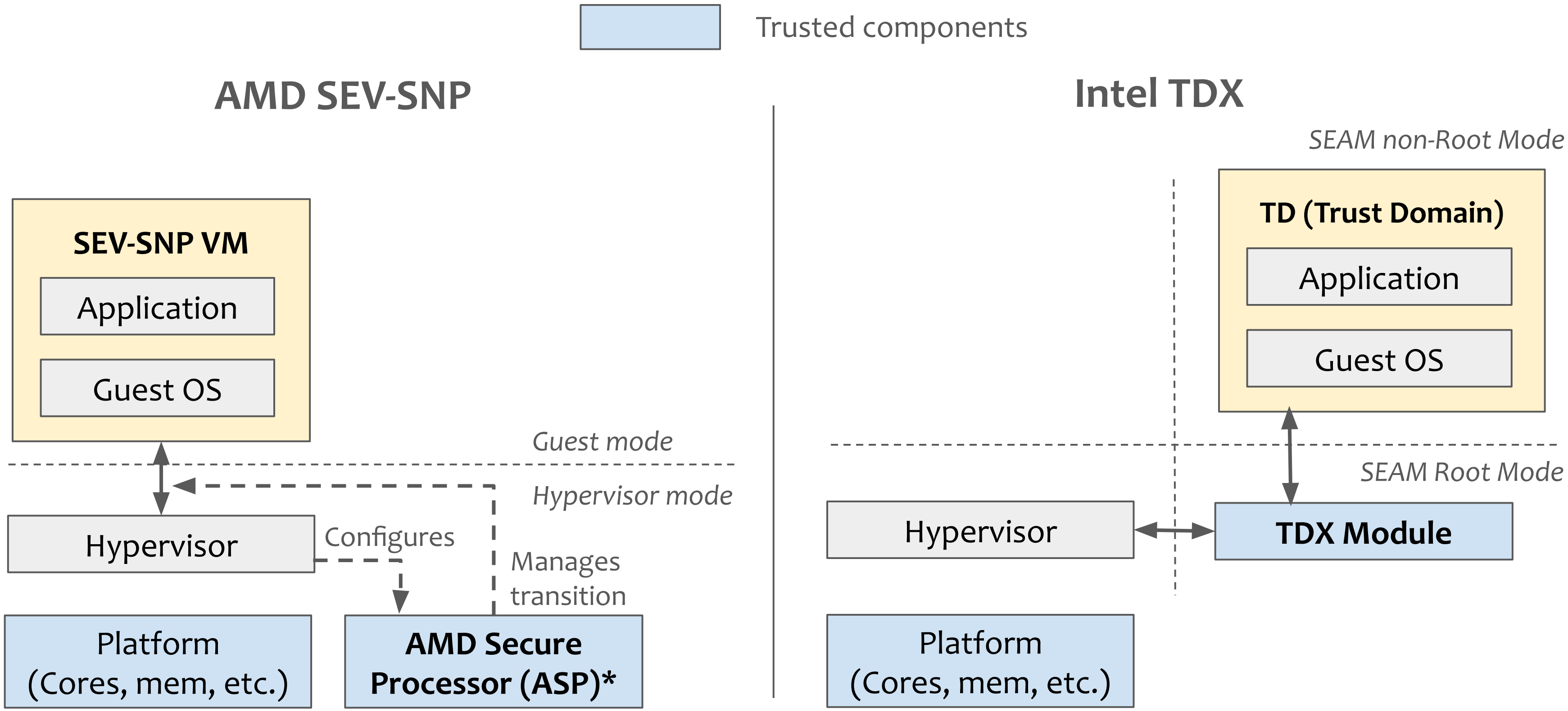
Confidential virtual machines (CVMs)

- **VM-level** Trusted Execution Environment (TEEs)
  - **Confidentiality**
    - VM state and memory is kept hidden
  - **Integrity**
    - No other entity can modify the VM state and memory
  - **Attestability**
    - Remote attestation to ensure the state of CVMs
- Easy to use than application-level TEEs (e.g., Intel SGX)
  - 👍 Programmability (use the existing software stacks)
  - 👍 Deployability (run unmodified applications)



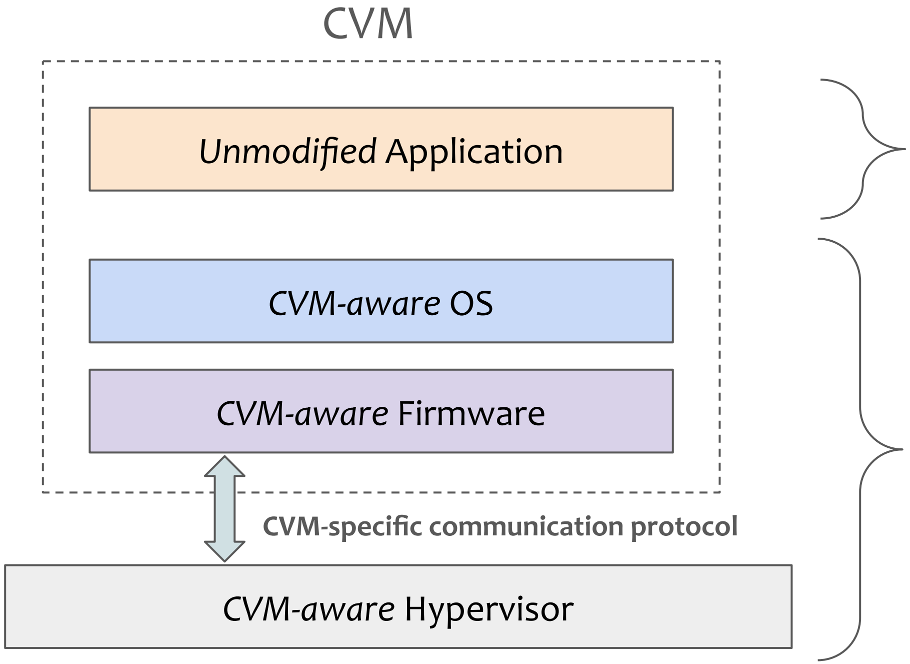
CVMs are attractive for various industry applications

Overview of AMD SEV-SNP / Intel TDX



*Also known as PSP
(Platform Security Processor)

Software stacks



Protecting data outside of CVM (e.g., using TLS) is **app's responsibility**

Both **guest OS and Host needs to be CVM-aware** for management

(AMD and Intel develop software stacks based on Linux and QEMU)

CVMs require new system software stacks + additional management

Outline

- ~~Overview~~
- ~~Background~~
- Evaluation
- Summary

Evaluation item

- Memory performance
- Boot time
- VMEXIT latency
- Application performance
  - System benchmark (Unixbench)
  - HPC (NPB), 3D rendering (Blender)
  - AI/ML (TensorFlow (BERT), PyTorch (AlexNet))
- I/O performance
  - Network
    - TCP/UDP (iperf)
    - Nginx, memcached
  - Storage (fio)
- Attestation primitives
- Security analysis
  - TCB size
  - CVE survey

Today's focus

- **Q: What is the basic overhead of CVMs?**
  - Memory performance
  - VMEXIT latency
- **Q: When is the CVM overhead significant?**
  - AI/ML (TensorFlow (BERT))
  - TCP/UDP (iperf)
- **Q: How actually secure is the current CVMs?**
  - CVE survey

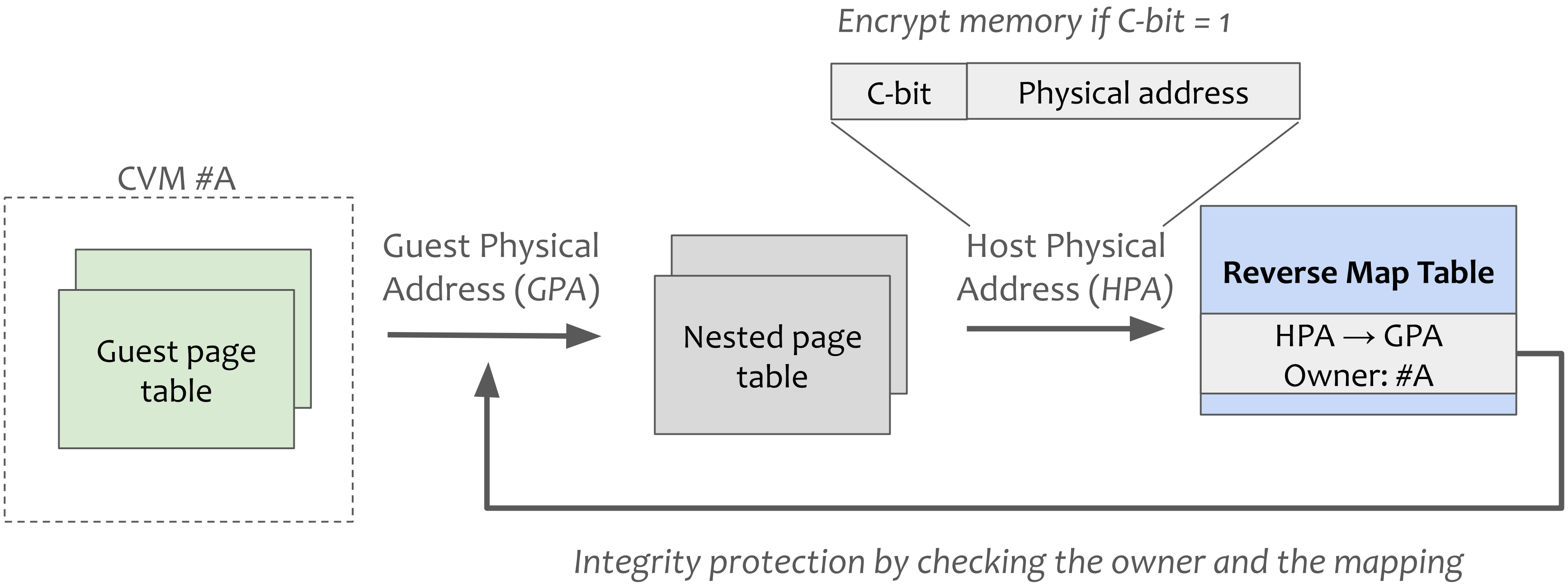
[Please refer to the paper for all evaluation]

Evaluation environment

	AMD SEV-SNP	Intel TDX
CPU	4th Gen AMD EPYC 9654P x 2	5th Gen Intel Xeon Platinum 8570 x 2
Memory	768 GB (SK Hynix DDR5 4800 MT/s 64 GB x 12)	1024 GB (Samsung DDR5 4800 MT/s 64 GB x 16)
Hypervisor	QEMU 8.2	QEMU 8.2
OS (Host/Guest)	Linux 6.8 / Linux 6.8	Linux 6.8 / Linux 6.8
Guest Firmware	OVMF	TDVF

- Disable hyperthreading, Turbo-boost, C-state
- Each vCPUs is pinned to a dedicated pCPU
- All measurement done in one NUMA node

Memory protection (AMD SEV-SNP)

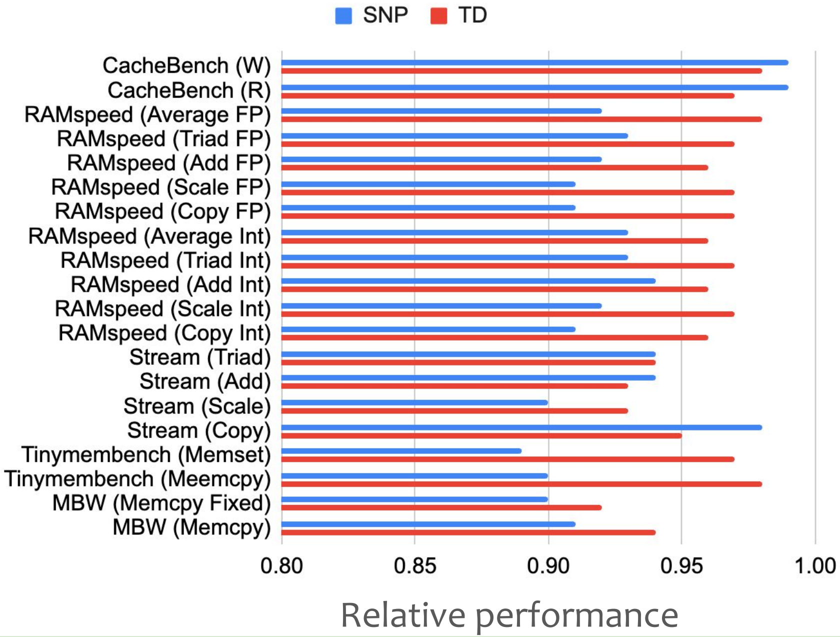


CVM has memory overhead due to memory encryption and integrity protection

Memory performance (Phoronix Memory Test Suite)

Baseline: normal VM w/o any memory protection

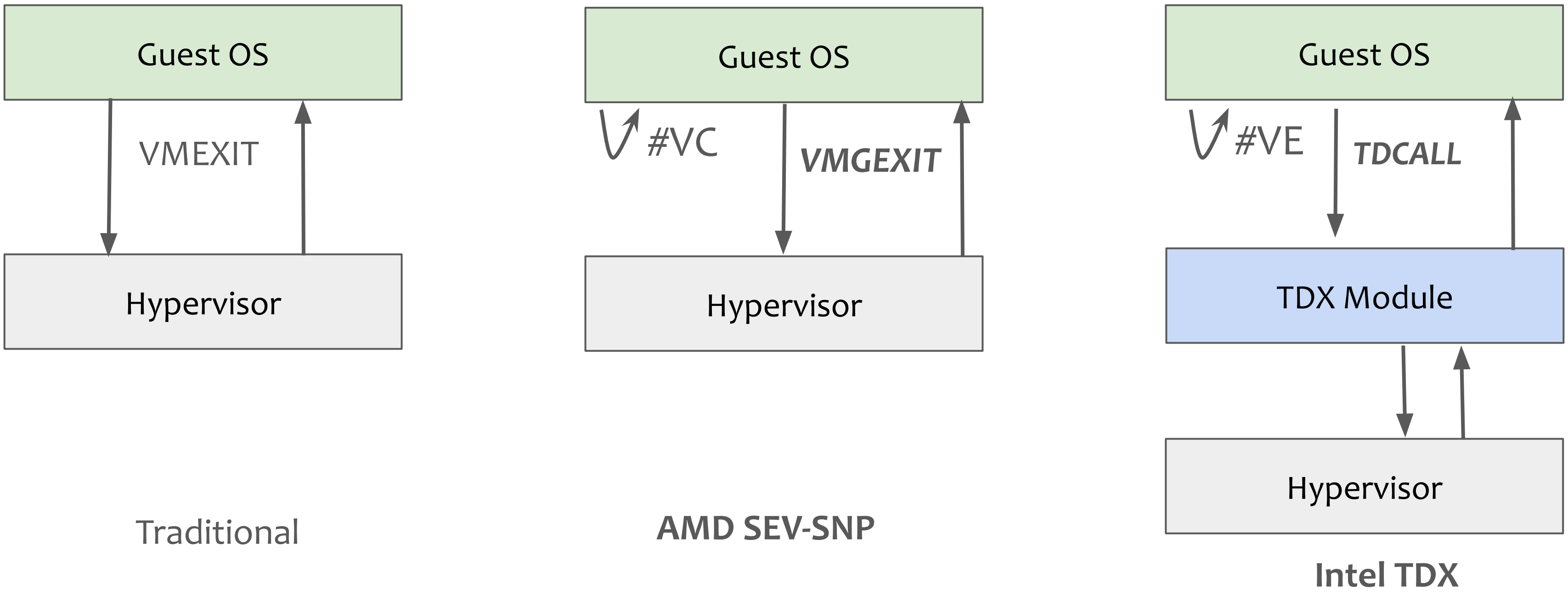
Compare SNP with a VM on the AMD machine, TDX with a VM on the Intel machine



→ **Right is better**

CVMs introduces <u>7.29%</u> (SEV-SNP) / <u>4.06%</u> (TDX) overhead on average

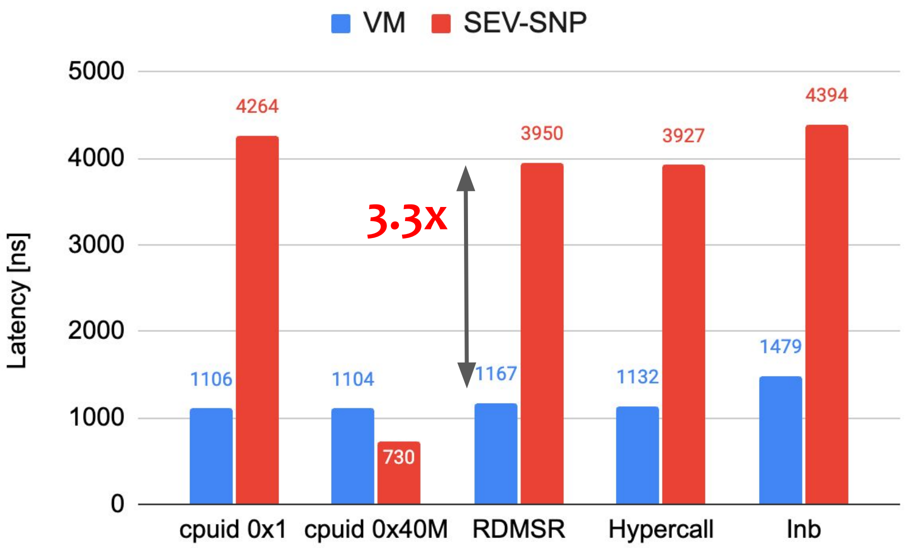
VMEXIT overhead



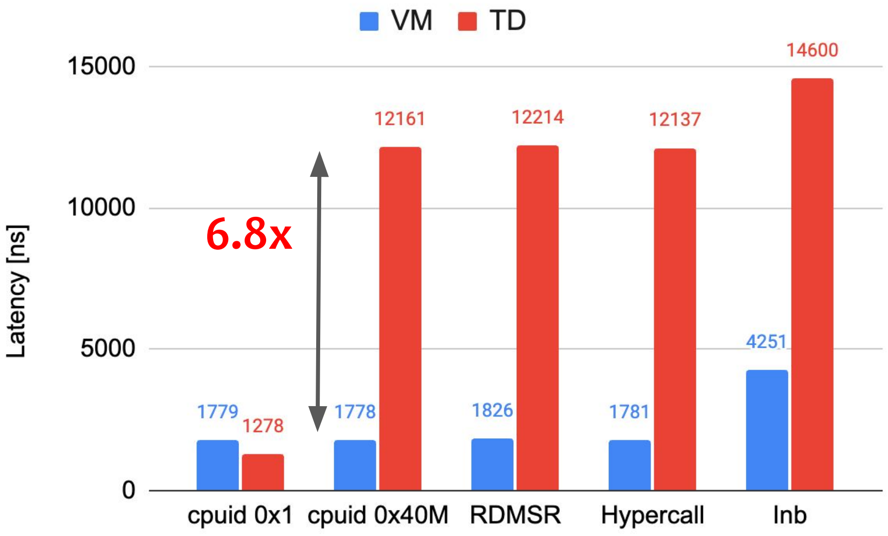
VMEXITs entail additional communication overhead

VMEXIT latency

↓ Lower is better



AMD SEV-SNP

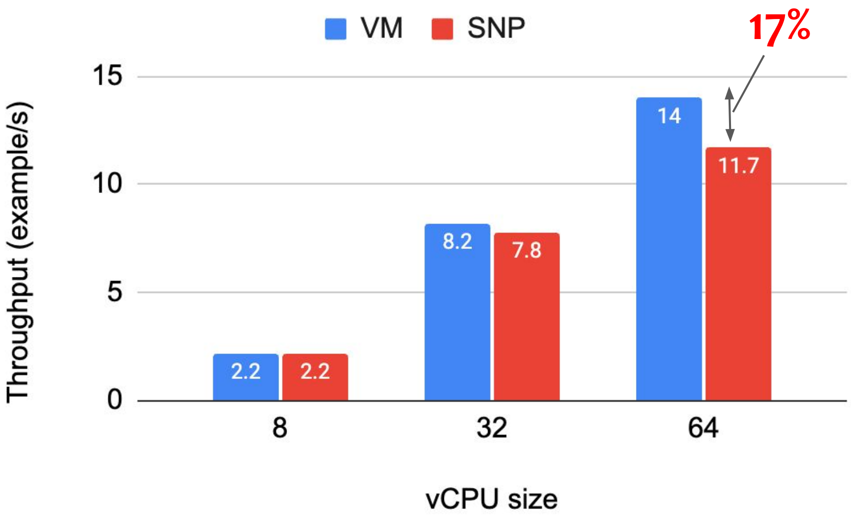


Intel TDX

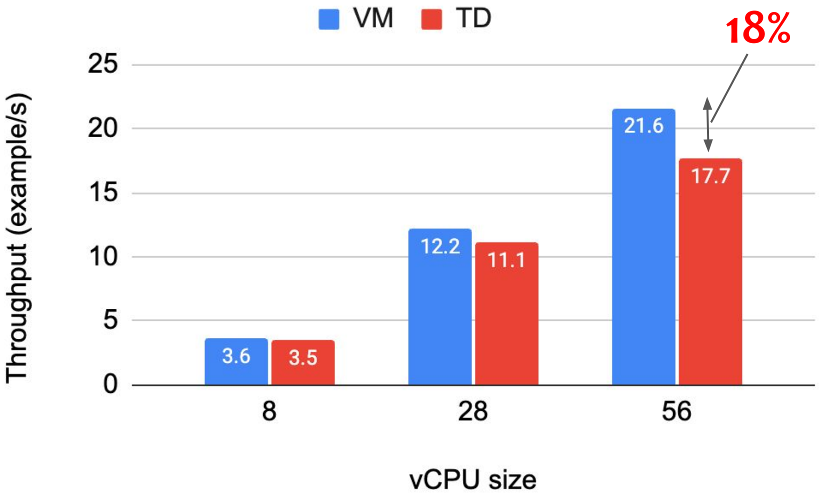
VMEXIT is costly for CVMs, showing up to 6.8x latency (TDX)

TensorFlow (BERT)

↑ **Upper is better**



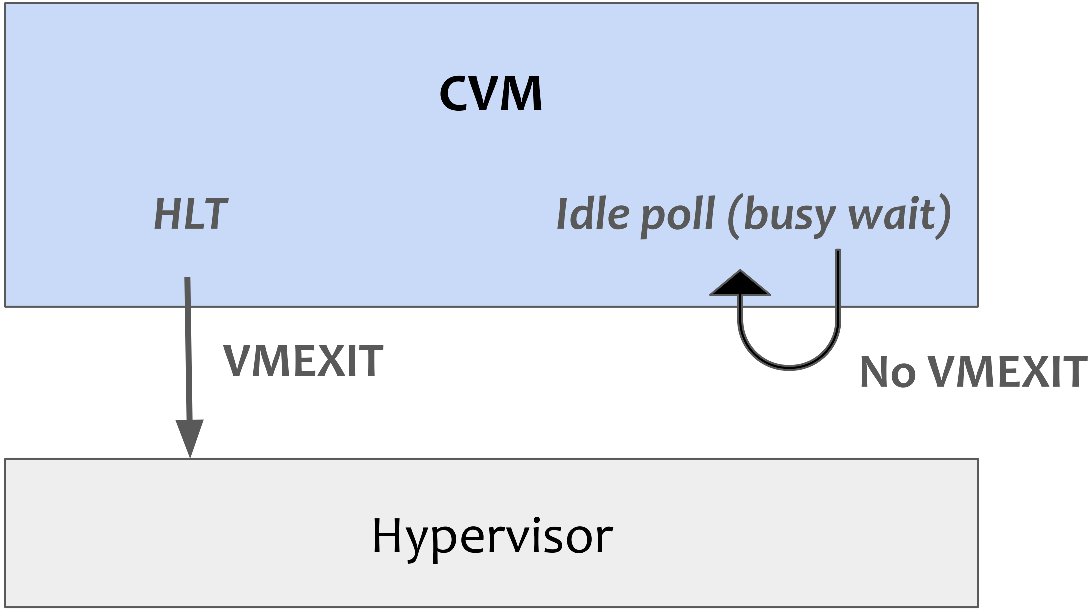
AMD SEV-SNP



Intel TDX

Memory overhead alone cannot explain this high overhead

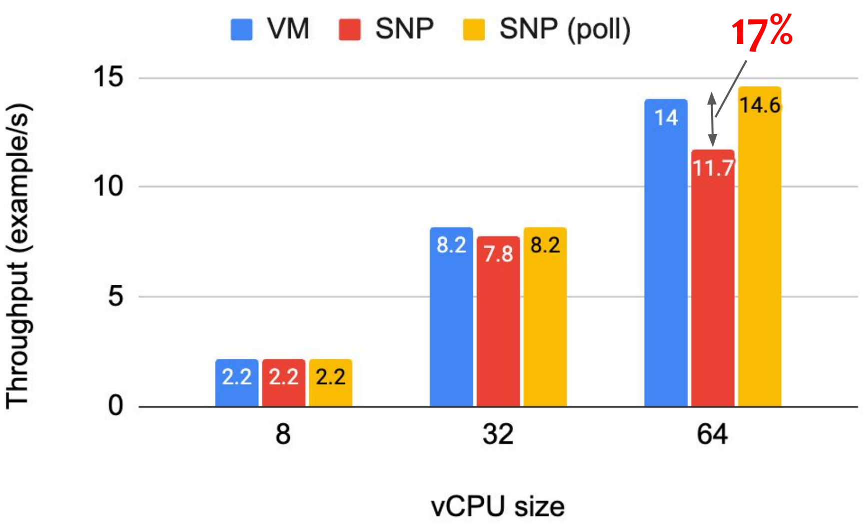
Guest-side idle polling



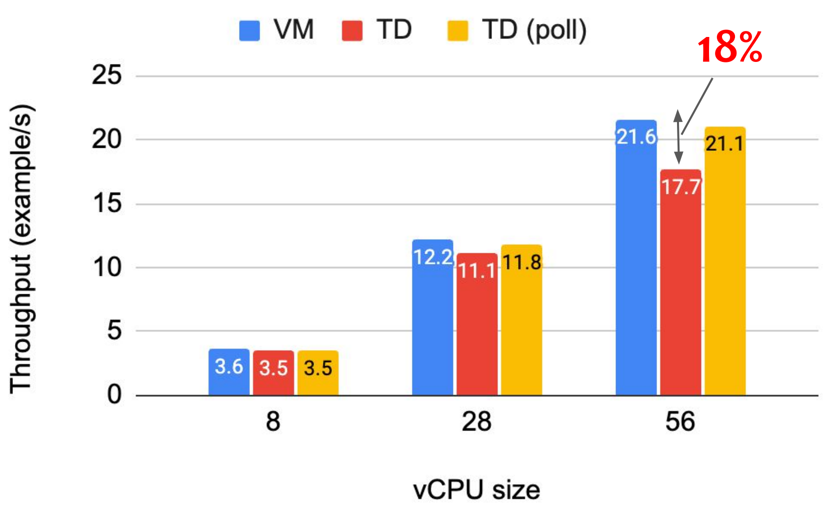
Polling is trade-off between CPU cycles and VMEXITs

TensorFlow (BERT) (revisited)

↑ Upper is better



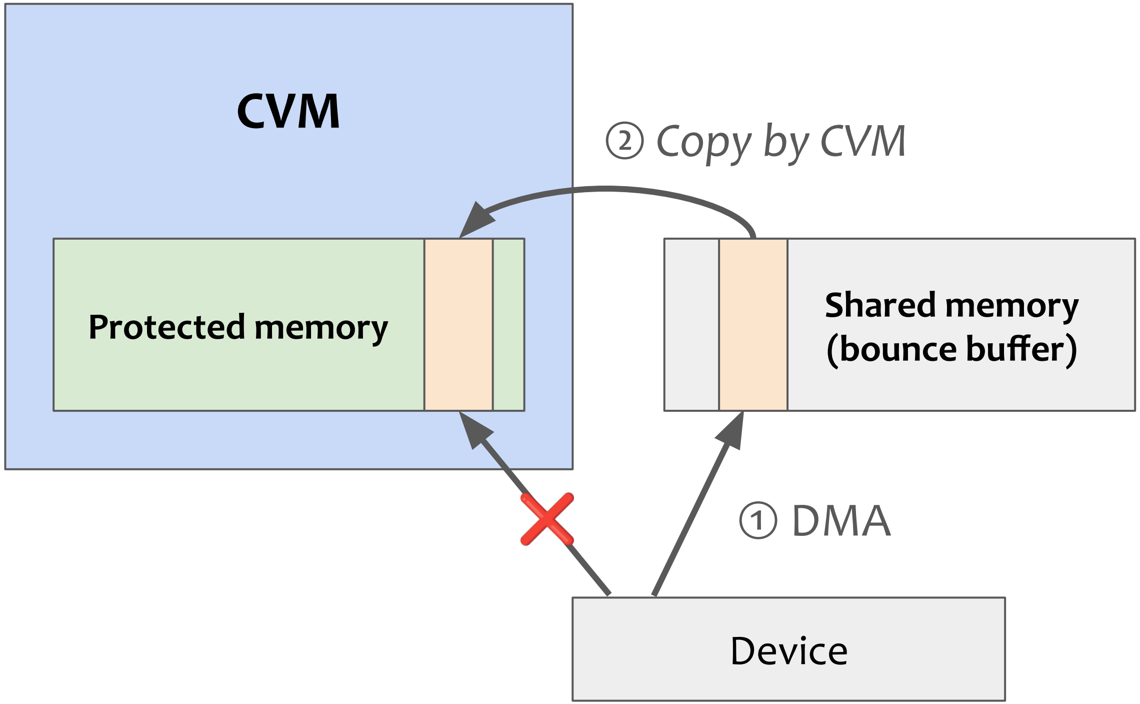
AMD SEV-SNP



Intel TDX

TensorFlow (BERT) shows up to <u>18% overhead</u> for a large VM
<u>Guest-side idle polling</u> mitigates the issue

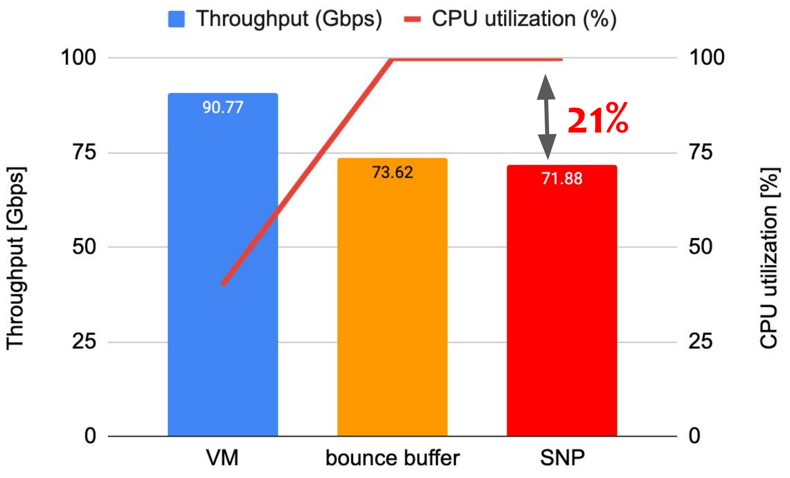
I/O in CVM



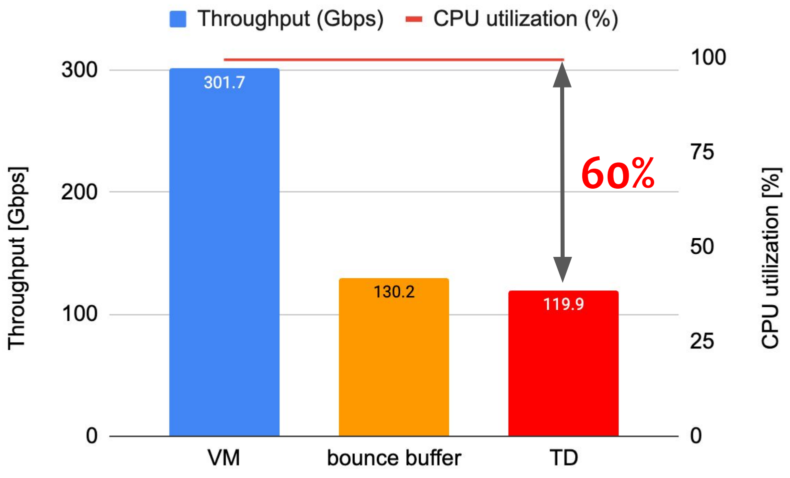
CVM
② Copy by CVM
Protected memory
Shared memory
(bounce buffer)
① DMA
Device

Network performance (1) High CPU load (TCP)

↑ **Upper is better** ※ local network, 8 queues, with vhost



AMD SEV-SNP

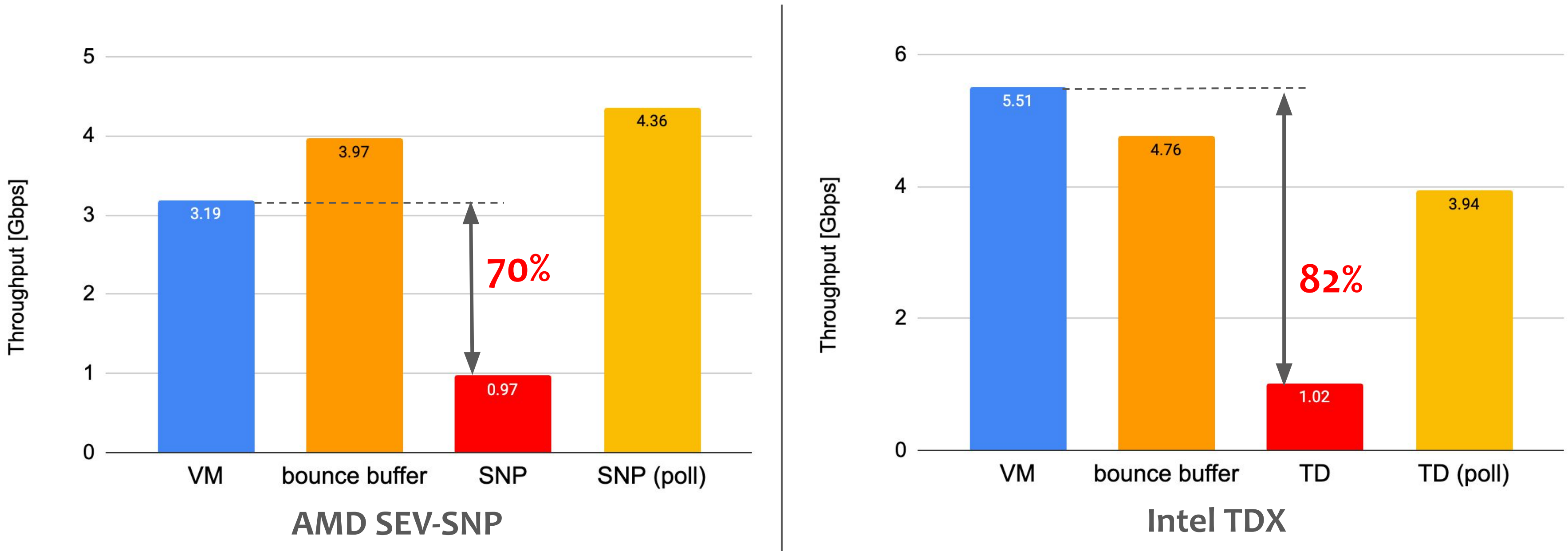


Intel TDX

Under high CPU load, <u>bounce buffer</u> dominates the performance drop

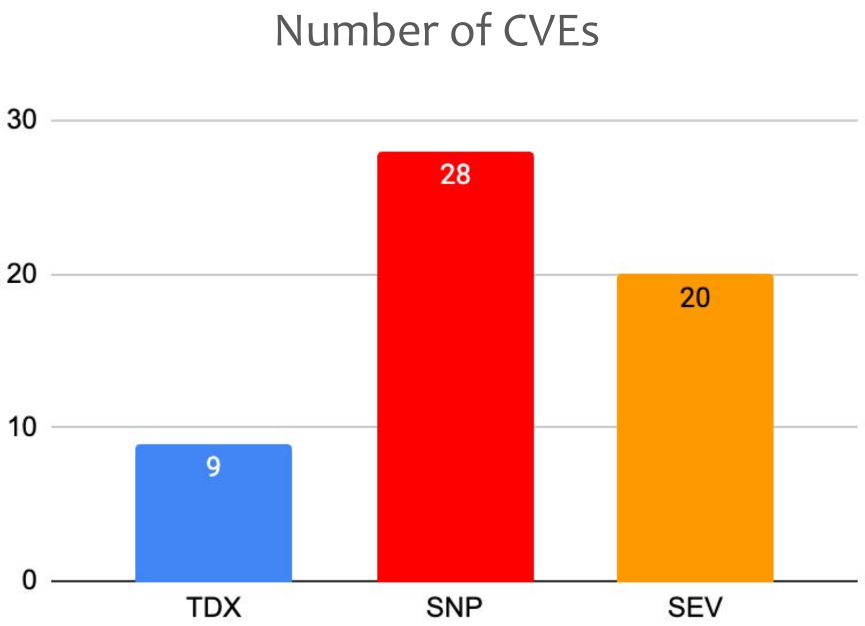
Network performance (2) non-CPU intensive (UDP)

↑ Upper is better ※ local network, single queues



For UDP (non-CPU intensive case), idle polling is also effective

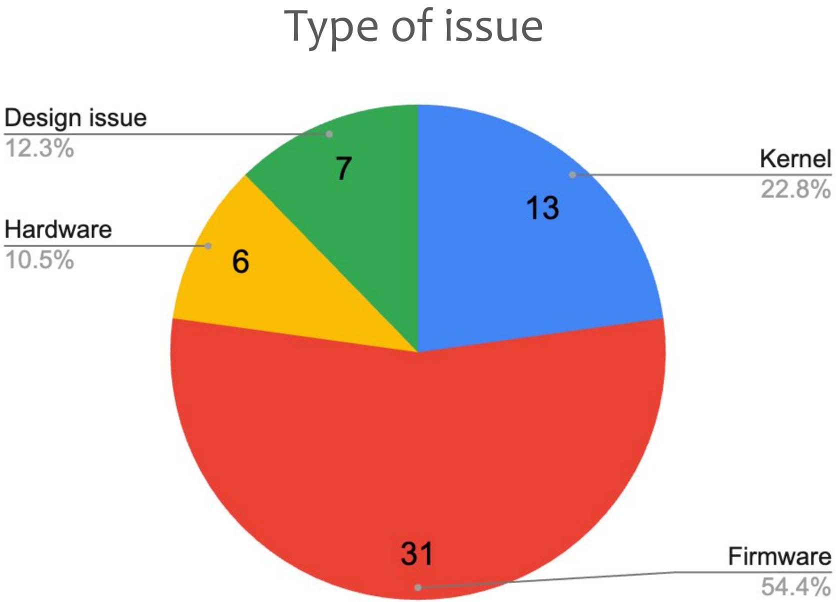
Security analysis: Found CVEs on SEV-SNP and TDX



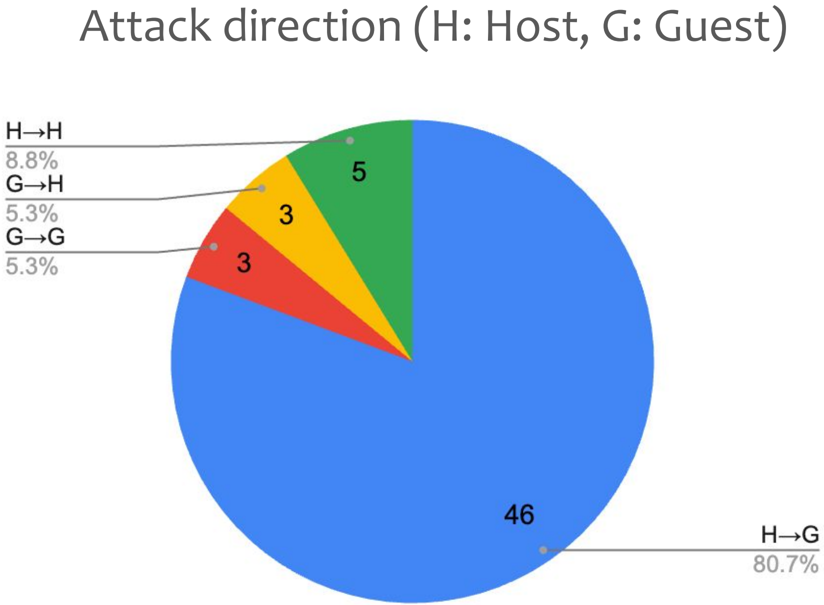
※ SEV(-SNP) predates TDX by ~5 years

CVMs are not free from CVEs

CVEs detail



* TDX-module/ASP firmware



CVM introduces new attack surfaces and vectors

Outline

- ~~Overview~~
- ~~Background~~
- ~~Evaluation~~
- Summary

Summary

Present a detailed empirical analysis of two leading CVMs: AMD SEV-SNP and Intel TDX

Call for the action

- Reducing VMEXIT impact (guest-side polling mitigates the issue)
- Optimizing I/O stacks (bounce buffer overhead is non-negligible)
- Testing additional software and new interfaces (new attack vectors introduced)

Evaluation code: https://github.com/TUM-DSE/CVM_eval

✉️ Masanori Misono <masanori.misono@in.tum.de>