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

Masanori Misono
Dimitrios Stavrakakis
masanori.misono@in.tum.de
dimitrios.stavrakakis@tum.de
Technical University of Munich
Munich, Germany

Nuno Santos
nuno.m.santos@tecnico.ulisboa.pt
INESC-ID & Instituto Superior Técnico, University of Lisbon
Lisbon, Portugal

Pramod Bhatotia
Technical University of Munich
Munich, Germany
pramod.bhatotia@tum.de

Abstract

Confidential Virtual Machines (CVMs) strive to alleviate the programmability and usability challenges of the previously proposed enclave-based trusted computing technologies, promoting easier deployment in cloud infrastructures. However, differing microarchitectural features, interfaces, and security properties among vendors complicate the evaluation of CVMs for different use cases. Understanding the performance implications, functional limitations, and security guarantees of CVMs is a crucial step toward their adoption.

This work presents a detailed empirical analysis of two leading CVM technologies: AMD Secure Encrypted Virtualization–Secure Nested Paging (SEV-SNP) and Intel Trust Domain Extensions (TDX). We review their microarchitectural components and conduct a thorough performance evaluation across various aspects, including memory management, computational and I/O performance, and attestation primitives. We further present a security analysis through a trusted computing base (TCB) evaluation and Common Vulnerabilities and Exposures (CVE) analysis. Our key findings demonstrate, among others, the effect of CVMs on boot time, memory management and I/O, and identify inefficiencies in their context switch mechanisms. We further provide insights into the performance implications of CVMs and highlight potential room for improvement.

CCS Concepts

• **General and reference** → **Surveys and overviews**; • **Security and privacy** → **Virtualization and security**.

Keywords

confidential computing, virtual machine, AMD SEV-SNP, Intel TDX

ACM Reference Format:
Masanori Misono, Dimitrios Stavrakakis, Nuno Santos, and Pramod Bhatotia. 2025. Confidential VMs Explained: An Empirical Analysis of AMD SEV-SNP and Intel TDX. In *Abstracts of the 2025 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS Abstracts '25), June 9–13, 2025, Stony Brook, NY, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3726854.3727280

*This is a summary of our full paper [16].

SIGMETRICS Abstracts '25, Stony Brook, NY, USA

ACM ISBN 979-8-4007-1593-8/2025/06
https://doi.org/10.1145/3726854.3727280

1 Introduction

Context. Confidential computing [6] has become essential in cloud environments to protect data and code in use. While the traditional application-level enclave-based Trusted Execution Environments (TEEs) [8, 13] offer strong security properties, they have limited adoption by cloud users because of high programmability overheads and compatibility issues, among other factors.

To overcome the limitations of enclave-based TEEs, major hardware vendors have proposed new trusted VM-level ISA extensions, namely Confidential Virtual Machines (CVMs) [3, 4, 10–12, 18, 20]. CVMs provide a confidential computing abstraction at the level of a virtual machine, which allows unmodified applications to run on it with existing software stacks while protecting data in use from unauthorized access, even from the cloud provider. To this end, several major cloud providers are already offering CVM instances [7, 15]).

Motivation. However, CVMs from different vendors vary in microarchitectural features, hardware interfaces and security properties, despite their aim to offer the same VM-based abstraction for trusted computing. Such differences mandate new VM management schemes, leading to changes in the system software stack for both the host and guest environments. Therefore, understanding the performance characteristics, functional limitations, and security features of CVMs is crucial for their adoption. Although several studies analyze CVMs, they mainly perform literature reviews [1, 5, 13, 19] or focus on evaluating a single CVM technology (e.g., AMD SEV(-SNP)) [2, 9, 14, 17, 21]. Our paper fills the gap by providing a practical and comprehensive empirical analysis of CVMs.

Our Approach. We conduct a detailed empirical analysis of two widely-used commercial CVM technologies, AMD SEV-SNP [3], and Intel TDX [12] (Figure 1). Initially, we thoroughly review their (micro)architectural components to highlight their functionalities and how they interact to achieve the security goals of CVMs. Following, we demonstrate our experimental results, obtained through a series of micro- and macrobenchmarks, to identify the performance characteristics and implications of CVMs on various workload scenarios and use cases. We examine several CVM aspects, including memory management, computational performance, storage and network stacks, and attestation primitives. Lastly, we analyze the security features of these CVM technologies and their trusted computing base (TCB) size and present our Common Vulnerabilities and Exposures (CVE) analysis.

Contributions. To the best of our knowledge, this work is the first systematic study of modern confidential computing architectures

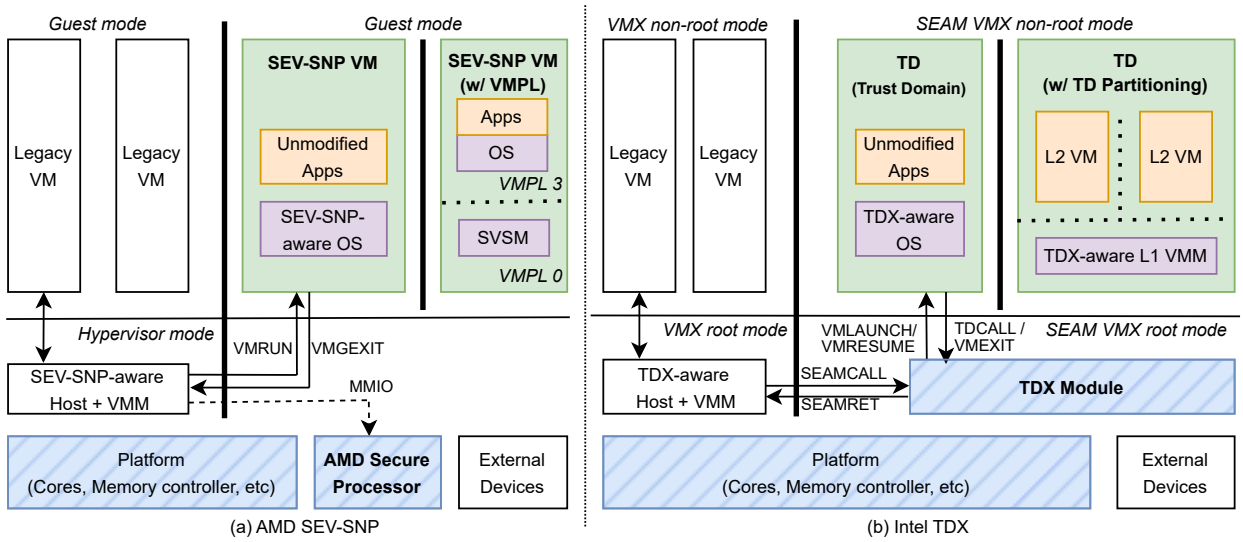

Figure 1: The architecture of AMD SEV-SNP (left) and Intel TDX (right). The green regions denote CVMs, with unmodified software (orange) and modified software (purple). The blue-hatched regions are trusted components. The thick line indicates that each CVM is isolated from the host and other VMs with encryption. We present an in-depth empirical analysis of these two technologies.

that are widely used in current cloud infrastructures. Importantly, we conduct both an extensive architectural and security analysis of CVMs, and practical experiments on real hardware. More specifically, our paper makes the following contributions:

(1) We present in-depth the architectural characteristics of AMD SEV-SNP and Intel TDX, the core confidential virtual machine technologies of the x86-64 platforms.
(2) We empirically evaluate the performance of AMD SEV-SNP and Intel TDX using real hardware (4th-generation AMD EPYC Processors, 5th-generation Intel Xeon Scalable Processors) across multiple dimensions to cover various use cases and application scenarios.
(3) We present a thorough security analysis of AMD SEV-SNP and Intel TDX, examining the TCB size and the reported CVM-related CVEs.
(4) We make our evaluation framework publicly available to facilitate further research endeavors in CVMs. The evaluation code is available at https://github.com/TUM-DSE/CVM_eval.

2 Our Key Findings

Through our extensive architectural analysis and experimental evaluation, we identify the following key findings for the examined CVM technologies:

- **Slow boot time:** Booting a CVM can take over twice as much as booting a standard VM. In addition to the additional procedure to launch a CVM, the host-side memory management predominantly affects boot time.
- **Memory allocation tax:** The CVM-specific *memory acceptance* operation is performance-heavy, and our memory allocation microbenchmarks show up to 92% increase for the *initial* memory allocation of the CVMs.
- **Costly context switch:** Frequently repeating the vCPU-sleep events (HLT) can cause a significant performance drop (e.g., NPB shows 431% overhead in the worst case). We further show that guest-side polling can mitigate this issue.
- **I/O overhead:** When the CPU utilization is high, the I/O overhead can be significant due to the internal implementations of the default I/O software stacks using bounce buffers. We observe up to 60% performance drop for heavy network processing benchmarks (i.e., iperf TCP).
- **Large TCB:** The TCB size of a CVM typically includes millions of Lines of Code (LoC) as it contains the full-fledged operating system (OS) of the guest, thus increasing the attack vectors.
- **New security attack vectors:** 39% of the CVEs related to AMD SEV-SNP and Intel TDX are attributed to improper validation mechanisms, while 54% of the CVEs are associated with vulnerabilities in the underlying firmware. Additionally, 8 CVEs refer to attacks from the guest to the host and from the host to the host.

Implications on the microarchitectural hardware and software stack. There is a need to reconsider fundamental confidential computing concepts even though the hardware, firmware, and supporting software stacks constantly evolve to support more functionalities and extend their security guarantees. Our study can serve as a stepping stone toward enhancing cloud environments' security, applicability, and performance utilizing CVMs. In particular, based on the our study, we consider the following areas for improvement:

- **Reducing and optimizing VMEXIT impact:** In the context of CVMs, VMEXITs are expensive. Reducing the number of VMEXITs and minimizing VMEXIT processing time is critical for performance. From a software perspective, implementing a sophisticated and adaptable polling policy is desirable, while optimizing VMEXIT processing is essential from a hardware perspective.
- **Designing new boot scheme:** Improving the bootup times of CVMs through specially designed HW/SW co-designed system stacks is essential. Such an improvement will increase the applicability of CVMs in the cloud and facilitate further use cases for CVMs (e.g., serverless computing).
- **Optimizing I/O stacks:** I/O stacks are vulnerable points of CVMs, as all the data transfers must be checked and validated. However, current approaches incur significant performance degradation. Therefore, designing optimized CVM-aware I/O stacks should become a priority.
- **Unifying attestation primitives:** Each CVM technology has its own, sometimes quite convoluted, attestation primitives. A collaborative effort to unify the attestation process is necessary and would help to standardize a trustworthy CVM deployment process in the cloud.
- **Reducing and hardening TCB:** CVMs have inherently large TCBs, which can widen their attack surface due to the high number of inputs from the host. Hardening and reducing the TCB size is crucial to minimize security risks (e.g., hardened Linux kernel, minimal LibOS).
- **Open-sourcing firmware:** Making the platform firmware (ASP firmware / TDX module) ecosystem fully open-source would be beneficial for transparency and bug-fixing reasons. Furthermore, since it is written in C, it would be advantageous to use a memory-safe language such as Rust.
- **Testing new interfaces:** New CVM software stacks can introduce an attack from the guest to the host and the host to the host as well, and testing that interface is also essential.

Full version. The full version of this work presents the details of each evaluation with background information and discussion [16].

Acknowledgement. We thank our shepherd, Prof. Adwait Jog, and the anonymous reviewers for their helpful comments. We thank Intel for providing them with access to a TDX-enabled machine and thank Benny Fuhry for the assistance in the TDX machine access. We thank Shiny Sebastian for the valuable technical discussions on Intel TDX. We thank Robert Schambach and Luca Mathias for their initial contribution to the experiments. This work was partially supported by an ERC Starting Grant (ID: 101077577) and the Chips Joint Undertaking (JU), European Union (EU) HORIZON-JU-IA, under grant agreement No. 101140087 (SMARTY). The authors acknowledge the financial support by the Federal Ministry of Education and Research of Germany in the programme of "Souverän. Digital. Vernetzt.". Joint project 6G-life, project identification number: 16KISK002. This work was also supported by the Fundação para a Ciência e Tecnologia (FCT) under grant UIDB/50021/2020, and by IAPMEI under grant C6632206063-00466847 (SmartRetail).

References

[1] Ayaz Akram, Venkatesh Akella, Sean Peisert, and Jason Lowe-Power. 2022. SoK: Limitations of Confidential Computing via TEEs for High-Performance Compute Systems. In *Proceedings of the 2022 IEEE International Symposium on Secure and Private Execution Environment Design*. IEEE. doi:10.1109/SEED55351.2022.00018

[2] Ayaz Akram, Anna Giannakou, Venkatesh Akella, Jason Lowe-Power, and Sean Peisert. 2021. Performance Analysis of Scientific Computing Workloads on General Purpose TEEs. In *Proceedings of the 2021 IEEE International Parallel and Distributed Processing Symposium*. IEEE. doi:10.1109/IPDPS49936.2021.00115

[3] AMD. 2020. AMD SEV-SNP: Strengthening VM Isolation with Integrity Protection and More. Retrieved 2024-07-24 from https://www.amd.com/system/files/TechDocs/SEV-SNP-strengthening-vm-isolation-with-integrity-protection-and-more.pdf

[4] ARM. [n. d.]. Arm Confidential Compute Architecture. Retrieved 2024-07-24 from https://www.arm.com/architecture/security-features/arm-confidential-compute-architecture

[5] Pau-Chen Cheng, Wojciech Ozga, Enriquillo Valdez, Salman Ahmed, Zhongshu Gu, Hani Jamjoom, Hubertus Franke, and James Bottomley. 2024. Intel TDX Demystified: A Top-Down Approach. *ACM Comput. Surv.* (mar 2024). doi:10.1145/3652597

[6] The Confidential Computing Consortium. 2022. Confidential Computing: Hardware-Based Trusted Execution for Applications and Data: November 2022, V1.3. Retrieved 2024-07-24 from https://confidentialcomputing.io/wp-content/uploads/sites/10/2023/03/CCC_outreach_whitepaper_updated_November_2022.pdf

[7] Google. 2019. Google Cloud Confidential Computing. Retrieved 2024-07-24 from https://cloud.google.com/confidential-computing/

[8] Roberto Guanciale, Nicolae Paladi, and Arash Vahidi. 2022. SoK: Confidential Quartet - Comparison of Platforms for Virtualization-Based Confidential Computing. In *Proceedings of the 2022 IEEE International Symposium on Secure and Private Execution Environment Design*. IEEE. doi:10.1109/SEED55351.2022.00017

[9] Benjamin Holmes, Jason Waterman, and Dan Williams. 2024. SEVeriFast: Minimizing the Root of Trust for Fast Startup of SEV microVMs. In *Proceedings of the 29th International Conference on Architectural Support for Programming Languages and Operating Systems*. ACM. doi:10.1145/3620665.3640424

[10] Guerney D. H. Hunt, Ramachandra Pai, Michael V. Le, Hani Jamjoom, Sukadev Bhattiprolu, Rick Boivie, Laurent Dufour, Brad Frey, Mohit Kapur, Kenneth A. Goldman, Ryan Grimm, Janani Janakirman, John M. Ludden, Paul Mackerras, Cathy May, Elaine R. Palmer, Bharata Bhasker Rao, Lawrence Roy, William A. Starke, Jeff Stuecheli, Enriquillo Valdez, and Wendel Voigt. 2021. Confidential Computing for OpenPOWER. In *Proceedings of the 16th European Conference on Computer Systems*. ACM. doi:10.1145/3447786.3456243

[11] IBM. [n. d.]. IBM Documentation – Introducing IBM Secure Execution for Linux. Retrieved 2024-07-24 from https://www.ibm.com/docs/en/linux-on-systems?topic=virtualization-secure-execution

[12] Intel. [n. d.]. Intel® Trust Domain Extensions (Intel TDX). Retrieved 2024-07-24 from https://www.intel.com/content/www/us/en/developer/articles/technical/intel-trust-domain-extensions.html

[13] Patrick Jauernig, Ahmad-Reza Sadeghi, and Emmanuel Stapf. 2020. Trusted Execution Environments: Properties, Applications, and Challenges. *IEEE Security & Privacy* 18, 2 (2020), 56–60.

[14] Dingji Li, Zeyu Mi, Chenhui Ji, Yifan Tan, Binyu Zang, Haibing Guan, and Haibo Chen. 2023. Bifrost: Analysis and Optimization of Network I/O Tax in Confidential Virtual Machines. In *Proceedings of the 2023 USENIX Annual Technical Conference*. USENIX. https://www.usenix.org/conference/atc23/presentation/li-dingji

[15] Microsoft. 2019. Azure Confidential Computing. Retrieved 2024-07-24 from https://azure.microsoft.com/en-us/solutions/confidential-compute

[16] Masanori Misono, Dimitrios Stavrakakis, Nuno Santos, and Pramod Bhatotia. 2024. Confidential VMs Explained: An Empirical Analysis of AMD SEV-SNP and Intel TDX. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 8, 3 (2024). doi:10.1145/3700418

[17] Saeid Mofrad, Fengwei Zhang, Shiyong Lu, and Weidong Shi. 2018. A Comparison Study of Intel SGX and AMD Memory Encryption Technology. In *Proceedings of the 7th International Workshop on Hardware and Architectural Support for Security and Privacy*. ACM. doi:10.1145/3214292.3214301

[18] Wojciech Ozga, Guerney D. H. Hunt, Michael V. Le, Elaine R. Palmer, and Avraham Shinnar. 2023. Towards a Formally Verified Security Monitor for VM-Based Confidential Computing. In *Proceedings of the 12th International Workshop on Hardware and Architectural Support for Security and Privacy*. ACM. doi:10.1145/3623652.3623668

[19] Ravi Sahita, Dror Caspi, Barry Huntley, Vincent Scarlata, Baruch Chaikin, Siddhartha Chhabra, Arie Aharon, and Ido Ouziel. 2021. Security Analysis of Confidential-compute Instruction Set Architecture for Virtualized Workloads. In *Proceedings of the 2021 International Symposium on Secure and Private Execution Environment Design*. IEEE Computer Society. doi:10.1109/SEED51797.2021.00024

[20] Ravi Sahita, Vedvyas Shanbhogue, Andrew Bresticker, Atul Khare, Atish Patra, Samuel Ortiz, Dylan Reid, and Rajnesh Kanwal. 2023. CoVE: Towards Confidential Computing on RISC-V Platforms. In *Proceedings of the 20th ACM International Conference on Computing Frontiers*. ACM. doi:10.1145/3587135.3592168

[21] Mingjie Yan and Kartik Gopalan. 2023. Performance Overheads of Confidential Virtual Machines. In *Proceedings of the 31st International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*. IEEE. doi:10.1109/MASCOTS59514.2023.10387607