General overview of AMD SEV-SNP and Intel TDX

Kevin Kollenda

ABSTRACT

Trusted execution environments (TEEs) have become increasingly common for the execution of security critical code. AMD SEV-SNP and Intel TDX are new hardware extensions developed to provide trusted execution for virtual machines. By providing additional integrity guarantees and building upon previous secure extensions, they enable confidential computing in cloud environments without risking sensitive user data. This work presents the additional components and processes which are used to accomplish these substantial security gains.

KEYWORDS

trusted computing, amd sev-snp, intel tdx, secure nested paging

1 INTRODUCTION

Today, software companies are increasingly moving their applications to cloud environments instead of hosting them on-premises. This may pose a risk to confidential user data, as cloud service providers (CSP) have direct access to the hardware running potentially security critical applications. TEEs provide a way to safely execute code without the risk of sensitive data being disclosed to nefarious actors and were previously created using Intel SGX. However, SGX works by partitioning an application into a trusted part secured by an enclave and an untrusted part being run normally. This degrades the development experience, as developers need to be aware of the security model and split their applications accordingly, whereas securing a VM requires no adjustments to application code.

To prevent information leakages and protect confidential data, the need for virtual machines (VM) with locked down access from hypervisors emerged. AMD and Intel now iterate on their previous trusted computing CPU extensions, AMD SEV-ES [4] and Intel SGX [17], to improve virtual machine integrity guarantees and minimize the trusted computing base (TCB). Cloud service users (CSU) are able to safely rely on applications running in cloud environments, leveraging the capabilities introduced by these new extensions.

AMD SEV-SNP (Secure Nested Paging) [2] and *Intel Trust Domain Extensions (TDX)* [17] are two hardware based CPU extensions that aim to increase the confidentiality and integrity of in-memory virtual machine data. They build upon earlier extensions (e.g. Intel SGX, AMD SEV-SEM), which could not provide sufficient integrity guarantees. These guarantees are implemented by allowing VMs to restrict write access to their memory pages and cryptographically verifying the output of security critical CPU instructions like `CPUID`. Additionally, virtual machines can now trust that the firmware versions reported by security critical components match the actually running version, which is essential to prevent rollback attacks making use of fixed issues.

Similar to their predecessor extensions AMD SEV-ES and Intel SGX, AMD SEV-SNP and Intel TDX provide remote attestation capabilities [24]. Generated attestation reports include the VM's state and custom data provided by the VM (e.g. a public key used for verification) [2] [17], allowing CSUs to confirm that their VM launched and executed correctly.

While AMD SEV-SNP is already available for every AMD EPYC 7003 series processor released since March 2021 [22], Intel TDX is not available for any current generation processors. Linux kernel support for both AMD SEV-SNP and Intel TDX was introduced with version 5.19 of July 2022 [13] and share some of the newly added code due to the architectural similarity.

2 BACKGROUND

In this section, previous VM extensions and their functionalities are explained. Additionally, the process of how memory is accessed on a modern system is outlined, due to its importance for memory integrity.

2.1 History of virtual machine CPU extensions

Previously multiple CPU extensions were developed by AMD and Intel to support more reliable and better performing virtual machines. This includes *AMD-V Nested Paging (NP)* [1] and *Intel Extended Page Table (EPT)* [15], which replaced slow software-based page virtualization with hardware accelerated nested paging.

Increased confidentiality guarantees were made possible by *AMD Secure Memory Encryption (SME)* [11] to provide memory encryption for general purpose computing and *AMD Secure Encrypted Virtualization (SEV)* [11] which brought SME to virtual machines accelerated by AMD-V Nested Paging. The required memory encryption keys are handled by an integrated low power AMD Secure (Co-)Processor (AMD SP) [11], as to further decrease the amount of trusted components. There is currently no hardware assisted VM memory encryption available for Intel processors, but it is covered in the upcoming Intel TDX extensions. While SGX could be used in virtual machines to secure virtualized applications (vSGX), it is not ideal as several VM features are unavailable once you enable it for a guest (e.g. VM suspend/resume, snapshots, ...)[27] [9]. AMD also iterated on their AMD SEV extensions by introducing AMD SEV-ES (Encrypted State), which encrypts the virtual machine's register state when the VM terminates [4]. As AMD SEV-ES already provides integrity guarantees for the VM's register contents no additional hardware support was required regarding the VM's state in AMD SEV-SNP.

Current generation processors will provide a (malicious) virtual machine manager (VMM) with the encrypted contents of guest's memory (by relying on SEV), but they do not prevent write accesses on VM pages. This enables the aforementioned malicious hypervisor to corrupt the VM's state and poses the risk for various replay attacks. Replay attacks are a type of attack vector, where the malicious actor retrieves ciphertext at a given moment and replaces the unprotected memory with this data at a later point in time.

2.2 CPU capability self-reports and security sensitive registers

Modern processors provide various ways for a running system to gather information about the current CPU. This is mainly done by calling the `CPUID` instructions, which reports the hardware extensions available to the CPU, register sizes, and several other

configuration details. While emulating and adjusting the output of this instruction is often performed by a VMM to ease VM migrations and to limit a VM's features, this can also be abused by a malicious hypervisor. Although such interferences will only lead to a denial-of-service for the guest in most cases, it may also result in a buffer overflow when wrong values are reported for the extended save area included in x86. [2]. Model-specific-registers (MSR) are various control registers used by a processor to provide hardware debugging capabilities, performance monitoring/tracing data and additional information about available CPU features [21]. Unrestricted access to these registers is possible by the hypervisor, enabling potentially unwanted interventions in the VM's execution, such as forcing a debug breakpoint to interrupt the guest's control flow. Similarly, the CPU microcode patch level and firmware versions of CPU components used by hardware extensions can also be queried from software. If a trust domain (TD) cannot rely that security critical hardware components are running a certain predetermined version, there is no guarantee that previously resolved issues and bugs found in those aforementioned components aren't exploited.

2.3 Threat model

The threat model that earlier CPU extensions (e.g. AMD SEV and AMD SEV-ES) relied on includes more components that need to be trusted. This is caused by the lack of memory integrity guarantees, which enable a malicious entity to tamper with the VM's memory. As SEV-SNP and TDX prevent this, every component which would previously considered to be trusted due to its ability to write into VM memory (e.g. Hypervisor, Direct Memory Access (DMA) capable PCI devices) can now safely be assumed to be untrusted.

2.4 Paging

In typical x86_64 systems memory is accessed using pages. Paging is used to emulate a full virtual address space for each process, without requiring that much memory to be available on the computer [25]. It is also typically used for page-level memory protection, providing each process with its own set of pages. Thus user space processes cannot read and write to pages belonging to a different process or the system kernel. Paging capabilities are handled by the Memory Management Unit (MMU), which translates the virtual addresses used by applications to their underlying real physical addresses available to the hardware. Addresses are mapped to pages by splitting them up in different parts. Typically the most significant bits correspond to the page directory entry, the following bits indicate the page table entry and the remaining bits contain the offset within the page and various flags set by the operating system (e.g. read/write/execute enable). AMD SEV provides guest virtual machines to selectively protect and encrypt memory pages using a VM specific encryption key [11], restricting read access from outside sources (like the VMM). Similar memory encryption is guaranteed on a per application basis by Intel SGX [10].

3 ARCHITECTURE

Several hardware technologies are leveraged by SEV-SNP and TDX to provide a more secure TEE for VMs. How this is achieved and which components are involved in this process are described below.

3.1 Security critical components

The architectural improvements enabled by SEV-SNP and TDX are achieved by the introduction of newly created CPU integrated components and modules. While AMD's implementation of SEV-SNP relies on a CPU inbuilt secure processor to provide a safe environment for VMs [2], Intel TDX uses a multi-component architecture [17] consisting of the following items:

- Intel TDX Module offering a safe way to manage TDs while enforcing various security policies
- Intel Authenticated Code Module required for loading and verifying the TDX module in protected memory
- TD Quoting enclave (TDQE) created using Intel SGX for remote attestation

Those components offer the management interface needed by the hypervisor to create and configure VMs and are in charge of intercepting any attacks targeted at VMs protected by either SEV-SNP or TDX. Communication with AMD's SP is facilitated using a novel VM management API interface [6], while Intel relies on additional CPU instructions [18]. Data structures used by these components are inaccessible by external actors, e.g. software running on the system or DMA enabled devices, due to their security critical purpose. These data structures contain management information as required by the CPU's integrated secure modules. Such data structures include the encrypted virtual machine save area (VMSA) as provided by SEV-SNP or the virtual machine control structure (VMCS) issued for each TD by the VMMs using TDX.

3.2 Memory integrity protection

Memory integrity projection marks one of the major improvements introduced in AMD SEV-SNP and Intel TDX. Both SEV-SNP and TDX establish a split between shared (unencrypted) and private (encrypted) memory pages. Whether a page is considered shared or private is indicated by the most significant bit of the virtual address and verified in the page table walk. Shared pages may still be encrypted with a key belonging to the hypervisor [17], but should be considered untrusted from the VM's point of view. Private (guest) pages are always encrypted using a VM specific key, making the memory contents unreadable for outside observers [2] [17].

3.2.1 AMD SEV-SNP

The key principle of AMD SEV-SNP's integrity improvements is guaranteeing a VM, which has allocated private memory pages, that it will always read back the last memory value that it wrote into such pages. SEV-SNP not only guarantees this behaviour for regular memory reads and writes, but also when memory pages are swapped to persistent storage or the whole virtual machine is migrated to another host [2]. Thus replay attacks relying on replacing VM memory with previously read ciphertexts and denial of service attacks that corrupt VM memory by writing unrelated data to its pages, are no longer possible.

This is realized by adding a reverse map table (RMP) data structure shared across the system, which keeps track of the owner for each page that can be assigned to a VM. In the case of a RMP entry belonging to a page owned by a SNP enabled VM, it also saves the guest's physical address (gPA) that the page should be mapped to. Additionally, each RMP entry contains a validation bit, which is

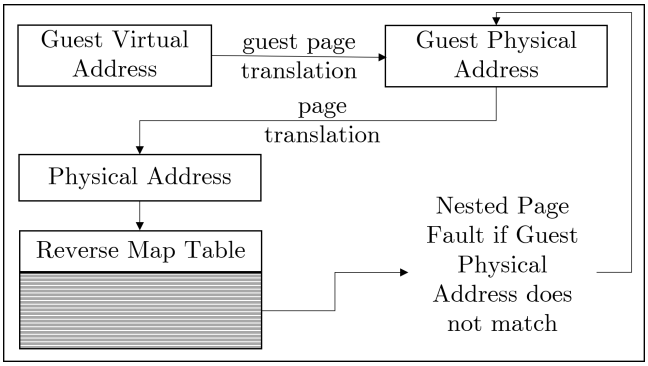

Figure 1: Nested VM PTW on AMD SEV-SNP enabled systems.

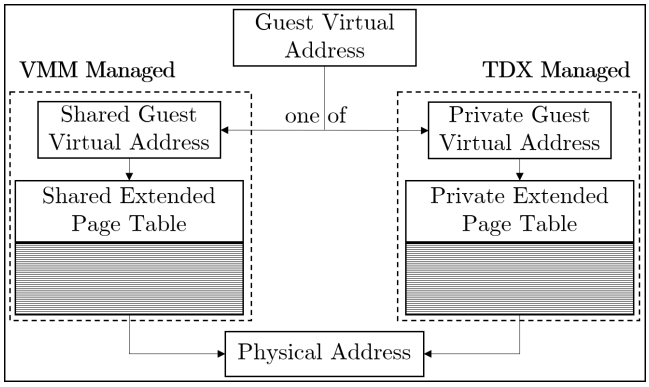

Figure 2: Page table walk on Intel TDX enabled systems.

cleared before a page is created for a VM [2]. Whenever a memory write access to a SNP-VM's page is performed, either by the VMM or a VM itself, a RMP check is performed to ensure that only the entity who owns that page can modify it. Memory read accesses from external actors, e.g. the virtual machine manager or other guests, do not need to be validated using the RMP, as the memory of private pages is always encrypted using the VM's specific private key.

The hypervisor's table walk is unaffected by the RMP, as long as the accessed page does not belong to a SEV-SNP enabled guest. If a malicious hypervisor tries to overwrite the memory contents of such a page, the table walk will result in a page fault (*#PF*) and the addressed page is unaffected. RMP checks are performed at the end of the regular page table walk (PTW), and its entries are indexed using the system's physical address. The changed table walk is the same as in Figure 1, however the first page translation is not necessary as there is no guest involved.

For SEV-SNP enabled virtual machines the page table walk is more complex. Using the accelerated nested page table walk provided by AMD-V, the initial address translation from a guest virtual address (gVA) to the guest's physical address and finally to the host systems physical address (sPA) is performed [1]. Afterwards the RMP check is invoked, which verifies that the page:

- belongs to a VM and not the hypervisor.
- is owned by the specific guest that initiated the table walk.
- mapped at the correct gPA.

This modified version of the PTW with the additional nested page table walk is visualized in Figure 1. Further information on page table states can be found in the appendix section A.

Page re-mapping attacks are prevented by these memory integrity guarantees, if a guest properly validates its private pages. To ensure that these integrity violations are not possible and caught by RMP checks, the guest must ensure that any gPA is only validated once. Achieving this can be done by performing all page validation on VM launch or by the guest keeping track of all previously verified gPAs. If this injective property between gPAs and sPAs holds true, any malicious nested page table change initiated by a compromised hypervisor will lead to the guest receiving a VM communication (*#VC*) exception when trying to access a remapped gPA (as `RMPUPDATE` clears the validation). Guests should treat *#VC* exceptions with great care, since it is a sign of an attack taking place. Typically the guest then tries to performs any necessary steps required to protect itself from a malicious VMM or terminates completely. [2]

3.2.2 *Intel TDX*

Intel TDX enables similar memory related integrity capabilities as AMD SEV-SNP. The CPU's inbuilt TDX Module provides an interface for the hypervisor to manage VMs indirectly and offers new instructions, `VMLAUNCH-VMX` and `VMRESUME`, for starting and resuming a VM. Instead of keeping track of the owner for each memory page on a system wide basis, TDX relies on keeping one shared extended page table for the hypervisor and multiple private extended page tables for each virtual machine (called TD by Intel) [18]. Due to these inherent architectural differences, the PTW initiated from inside a TD works quite differently to the one used by SEV-SNP. As can be observed in Figure 2, systems incorporating TDX resolve gVAs by choosing the appropriate EPT for a given address.

Initialization of the private EPT is handled by the Intel TDX module, which transforms VMM provided memory pages into private pages required by the TD. Integrity protection for pages is offered analogous to SEV-SNP. However, instead of relying on the guest to validate its private pages itself, Intel TDX tracks that the mapping of pages to their associated gPAs is unique inside a TD and additionally across TD boundaries, to avoid any memory aliasing related vulnerabilities [18]. Differentiating between shared and private memory is realized by including a 1-bit TD identifier for each cache line and optionally a 28-bit message authentication code (MAC) that includes the 1-bit identifier to ensure that any unauthorized changes to the memory are detected. If an attacker were to write to pages protected by the TDX module, a MAC-verification-failure would occur the next time the TD tries to read from the affected memory. If logical-integrity-mode without a crpytographic MAC is used, such read accesses will instead result in a TD-ownership check failure. Contrary to AMD's implementation, the guest has no impact on the way these failures would be handled and they will be terminated by the TDX module. The VMM or other VMs are not affected by the forceful termination of a guest due to one of these integrity violations.

3.3 Privilege levels and access control

AMD SEV-SNP and Intel TDX rely on different kinds of privilege levels for their integrity guarantees.

3.3.1 *AMD SEV-SNP's privilege concept*

As hypervisor operations modifying the state or memory of SEV-SNP enabled VM's are no longer permitted by default, the SP needs to assess these requests. Such operations may include standard VM managing commands, e.g. launching and resuming or suspending and terminating a VM, which were previously handled completely by the hypervisor [6]. Remote attestation and quoting is also performed by the SP, using the appropriate keys for the VM that the attestation process has been initiated for.

Apart from moving security sensitive operations to the secure processor, SEV-SNP also supports four additional optional virtual machine privilege levels (VMPLs). They are numbered from VMPL0 to VMPL3, with VMPL0 indicating the highest and VMPL3 the least access rights. These can be used to further divide the newly gained access controls provided by the RMP, for example to enable hardware assisted address space isolation inside a SEV-SNP enabled guest. Each virtual CPU (vCPU) assigned to a guest runs within one VMPL, where each VMPL can only grant permissions equivalent to the ones it currently possesses. This is done by invoking the `RMPADJUST` instruction, which updates the necessary RMP entries accordingly. Initial page validation using `PVALIDATE` only grants full read, write and execute rights to VMPL0. The restrictive nature of page table permission checks performed during the nested table walk imply that multiple page permissions need to be equivalent for a page to be accessible by a guest. Page permission validation is therefore handled in both the guest-managed page table and the hypervisor-managed nested page table provided by AMD-V, also additionally through the RMP table managed by the higher privileged VMPL. [2] [5]

Utilizing this finer-grained access control made available by these different VMPLs allows SEV-SNP guests to create a more restrictive emulation environment. APIC virtualization, which was previously handled by the hypervisor, could be performed by software running in VMPL0 and passing through the results to lower privileged VMPLs running inside the guest. Further tasks can be delegated to an intermediate layer executing with the highest privilege level, such as handling *#VC* exceptions which occurred in other vCPUs, thus providing additional abilities to support SEV-SNP unaware software inside a guest. Combining all those features made available by VMPLs, it is possible to nest legacy (non SEV-SNP) VMs inside a SEV-SNP guest containing intermediate code in VMPL0. Although there is a slight performance decrease due to the additional emulation performed by this translation, it can enable legacy workloads to run securely. [2] An architectural overview of this is outlined in Figure 3.

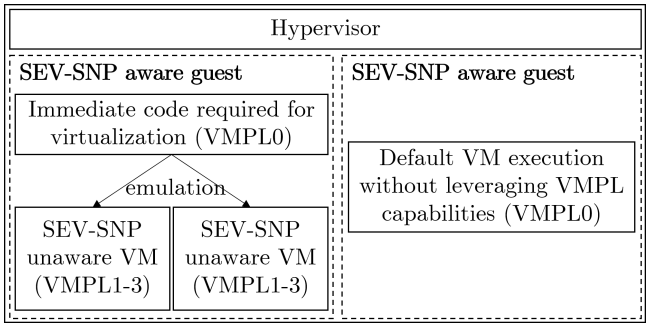

Figure 3: Nested virtualization as enabled by optional VMPLs.

3.3.2 *Intel TDX components required for access control*

Due to the multi-component architecture used by Intel TDX, there are more steps involved for VM management and communication. The Intel TDX module is hosted in an environment protected by the CPU's Secure-Arbitration Mode (SEAM), allocating as much reserved memory as is configured in the SEAM-range-register (SEAMRR). Confidentiality and integrity guarantees for SEAM memory are enforced similar to Intel TDX secured VMs. Memory accesses from any external actor, e.g. all software (regardless of whether it is run inside a VM or the VMM) or DMA capable devices, are prohibited [17]. The CPU in SEAM mode is not fully privileged though, as it is not permitted to access other secured memory regions, such as those used by Intel SGX enclaves or the CPU's system-management mode.

Installation and loading of this TDX module is performed by the newly introduced SEAM Loader (SEAMLDR), an additional authenticated code module (ACM) integrated into Intel's Trusted Execution Technology (TXT) stack. It is responsible for verifying and subsequently loading the Intel TDX module into the aforementioned SEAM memory range (SEAMMR), after being invoked by the VMM [18]. Additionally, the SEAMLDR passes the security-version number (SVN) through hardware-measurement registers inside the SEAMMR and places the TDX module into SEAM-VMX root mode. Following this initialization process, the VMM can communicate and give control over to the TDX module by executing the `SEAMCALL` instruction. The `SEAMRET` instruction is used by the TDX module to return execution to the hypervisor, ensuing the requested operation (e.g. creation, deletion or launching of a TD). Starting or resuming a virtual machine is accomplished using `VMLAUNCH` and `VMRESUME` instructions, which move the TDX module into SEAM-VMX non-root operation before passing control over to the TD [18].

In contrast to AMD's singular module architecture, remote attestation is handled by a separate SGX TDQE [18].

3.4 Secure capability reports

As mentioned in subsection 2.2, there are several issues with a VMM being able to control the information provided by `CPUID` or read from MSRs using `RDMSR` [21]. SEV-SNP and TDX seek to prevent a hypervisor from supplying VMs with incorrect CPU capability information as retrieved using one of the instructions mentioned above.

SEV-SNP enabled guests may instruct the AMD SP to verify that the `CPUID` data passed through by the hypervisor does not contain capabilities unavailable on the host and that safety critical size information is correct. This filtering may either be performed on-the-fly whenever `CPUID` is called or during the initial VM launch. When this should be handled on launch, two special pages are inserted into the guest's memory by the SP. One page contains the

encryption keys used for the communication between guest and SP, while the other holds the verified CPUID values. The former page is securely encrypted by the guest's private memory encryption key to prevent any unwanted accesses by the VMM [6]. As the verification process is only done once on VM launch and not every time CPUID is invoked, it is the better performing solution [2].

On guests using Intel TDX CPUID validation is done by default and does not need to be explicitly configured. Analogous to SEV-SNP the TDX module prevents the VMM from reporting greater capabilities than the host system actually supports. However, TDX guests can enable a virtualization exception unconditionally being thrown whenever CPUID is executed, allowing the VM's operating system (OS) to fully control how software inside the VM receives the requested CPUID information [18].

SEV-SNP and TDX both prevent interference with guest's MSRs, e.g. hardware debugging registers, by prohibiting the VMM's ability to write into these security sensitive registers [18]. This is handled automatically, without requiring any modifications to the guest's OS.

3.5 TCB rollback prevention

Rollback attacks rely on downgrading the version of components included in the trusted computing base (TCB) or maliciously reporting an older version of the component and subsequently exploiting bugs, which were already fixed in recent versions.

On SEV-SNP systems the SP ensures that its firmware may not be downgraded below the currently running version [6]. Additionally, the firmware version of each TCB component, such as the SP, is cryptographically merged with the Chip Endorsment Key (CEK) fused into the processor [2]. Due to these improvements, guest owners can now reliably trust that their VMs will not launch with a misreported firmware version lower than the minimum version threshold they have set beforehand.

A TCB managed by a TDX module is only considered to be up-to-date if each component included in the TCB reports a SVN higher than a threshold set by the component's author [17]. These SVNs are loaded from hardware registers into memory managed by the SEAM Loader, which is inaccessible for anyone but the TDX module, keeping it safe from manipulations of external actors. Thus any downgrade of such a module below this previously set version can result in the TCB losing it's up-to-date status. If a VM's launch policy does require a modern TCB version without it being present, the VM will not launch. [19]

3.6 Interrupt and exception injection

Injecting interrupts and exceptions is traditionally possible at all times by the hypervisor. Generally, this does not lead to problems for VMs, as all major VM operating systems support proper interrupt and exception handling. Some of these VM OSs contain in-built presumptions about how and when interrupts and exceptions can occur, caused by the fact that VMs typically try to emulate bare-metal hardware as close as possible. A guest OS may assume that no unknown opcode exceptions (*#UD*) are thrown after executing a valid instruction, as is the case for real hardware. However, they could be injected by the VMM at any time. [2]

Preventing yet to be discovered issues in the handling of such uncommon events in operating systems, AMD SEV-SNP provides two optionally configurable modes, which guests can enable to restrict the commonly unprotected interrupt and exception interface [4]. Alternate injection provides the default virtualized interrupt injection and queuing interfaces as they are typically used by the hypervisor, but only allow them to be called from within the VM itself. This prevents a malicious hypervisor from interfering with the guest's OS altogether, because the fields used to save the interrupt information are only accessible by other entities that can already work with the guest's data. Similar to how nested virtualization is realized using VMPL0 as an intermediate layer, as described in subsection 3.3.1, interrupt and exception handling can be facilitated by software running with VMPL0 privileges. However, guests may not want to fully block the hypervisor from injecting interrupts and exceptions, therefore SEV-SNP provides an additional interface called restricted injection. Guests that have enabled this mode prevent virtual interrupt queuing from the VMM altogether and heavily limit which interrupts can be injected. Instead of necessary interrupts being passed through to the VM directly, a newly introduced exception called hypervisor injected exception (*#HV*) [5] notifies the guest of the VMM's interrupt. Communicating extended information required by the VM to handle this event can be passed between the hypervisor and the guest using shared memory pages [2].

Intel TDX similarly keeps track of virtual interrupt information and APIC data inside the virtual machines control structures, protected from the hypervisor. Pages used for saving this information are provided on TD launch using the VM's associated private key. Intel VM-X was adjusted to prevent the delivery of exceptions into trust domains as virtual interrupts and injected interrupts are managed by CPU hardware [18]. Ensuring assumptions made by the VM's OS about interrupt prioritization and masks is also handled by TD's virtual interrupt virtualization. All of these guarantees are provided without requiring any modifications to the guest's operating system [17].

3.7 Remote attestation

Remote attestation is an essential feature required by CSUs as they want to verify that their deployed VMs are running as the CSU intended them to and were not tampered with [24]. Attestation reports are generated by the CPU in-built components, e.g. the SP on AMD systems or the TDQE on Intel hosts, containing various information (collected after launch and during runtime) about the VM itself. Previous extensions only allowed for the attestation to be performed after a trusted execution environment was launched, but SEV-SNP and TDX also enable runtime attestation.

The attestation process is always initiated by the guest. SEV-SNP VMs ask the SP to generate the report for them by using the SNP_GUEST_REQUEST firmware API call [6], while TDX guests invoke the TDCALL instruction of the TDX module. SEV-SNP guests only need to communicate with the SP and no other entities to process the attestation. This differs on TDX TDs, which would start by asking the TDX module for a general attestation report and subsequently requesting the VMM to quote it using the TDQE. The quoting enclave cannot be used by the TDX module directly,

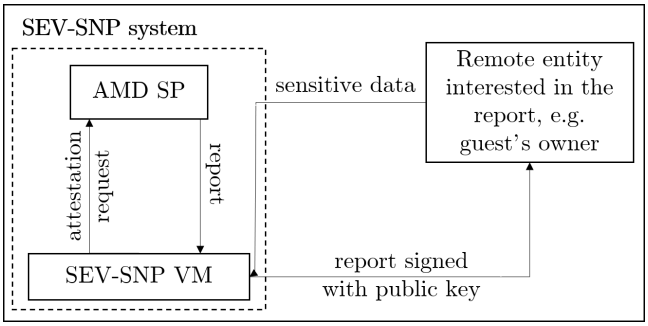

Figure 4: Remote attestation process for SEV-SNP guests adapted from AMD SEV-SNP whitepaper [2].

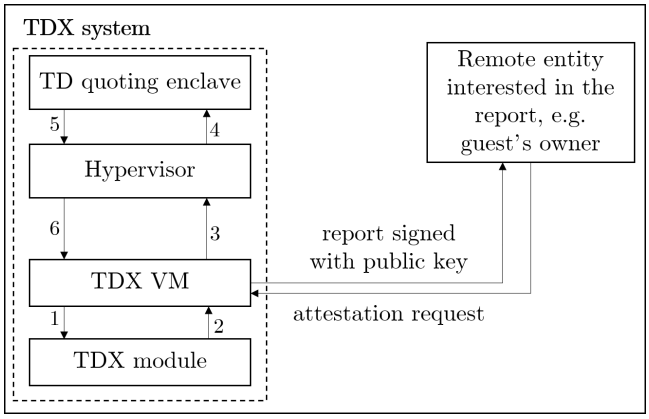

Figure 5: Remote attestation process for TDX guests adapted from Intel TDX whitepaper [17] and Intel TDX architecture specification [18].

because it is running inside a SGX enclave, inaccessible to SEAM. Typically the information includes metadata collected during the VM's launch, general system information, versions of security sensitive components (such as the TDX module or the SP's firmware) and arbitrary data provided by the TD [6] [2] [17]. On SEV-SNP enabled systems, the guest's owner can additionally issue a signed identity block (IDB) after launch to differentiate between guests and validate the boot using a provided checksum. TDX attestation reports also include fields identifying the TD's owner inside the `TDINFO_STRUCT` created during the attestation process [18]. The arbitrary data filled by the VM is often used to provide the other party with the VM's public key to communicate in a secure manner. CSUs can trust that the attestation report was generated properly, as the version for each component involved in the attestation and the status of security sensitive CPU features (e.g. simultaneous multithreading, SEM) is included [6]. The report is signed using either the versioned chip endorsement key (VCEK) that is unique for each AMD chip performing the SEV-SNP attestation or by relying on the signing key provided by the provisioning certification enclave (PCE) for Intel TDX guests [18]. Afterwards such reports can be verified by the party that requested the attestation by validating the report using company provided signatures. The process required to generate and transfer an attestation report is outlined in Figure 4 for SEV-SNP guests.

Due to the TDX's multi component architecture described above, there are more steps required for successful transfer of the attestation report as detailed in Figure 5. Initially, the TDX VM tasks the TDX module with generating the attestation report (Figure 5.1), analagous to how a SEV-SNP guest requests the report from the SP. Using this newly created report (Figure 5.2) the TD passes it upwards to the hypervisor for the signing process (Figure 5.3). The VMM relays the attestation data to the TDQE (Figure 5.4), responsible for the cryptographic verification of the report. Subsequently the signed information is passed back down to the hypervisor (Figure 5.5) and to the trust domain (Figure 5.6), which finally delivers the report back to the remote party [17] [18].

3.8 VM Migration and sealing

As trust domains may want to save data across VM executions, means to securely write data to persistent storage are part of SEV-SNP and TDX. SEV-SNP guests can simply ask the SP to generate local sealing keys, which the VM can trust to not have been tampered with by a malicious actor [2] and use to encrypt data that is not protected by SEV-SNP's memory guarantees.

SEV-SNP introduces a new CPU component called the migration agent (MA) to perform guest migrations. This agent runs inside a SEV-SNP VM on the same host and is responsible for verifying that a migration can be performed securely. As the agent is active on a per-system basis, it needs to run on both the source physical machine that the guest should be transferred away from and on the destination machine that it should be migrated to. TDX does also confidential VM migration using a specific service trust domain called the migration TD (MigTD), which handles the process in a secure fashion similar to AMD's MA [20]. Nearly all modern CSPs support the migration of (running) VM's, to ease maintenance efforts and to allow for dynamic load balancing. Live migration is managed by the agents running on both systems, which handle the re-encryption of data moved from source to destination [6]. Guests need to ensure that their resting data, e.g. data not being in memory, is encrypted by other means like full disk encryption, as neither TDX nor SEV-SNP handles the protection of such data.

4 RELATED WORK

While the introduced integrity guarantees provided by SEV-SNP solve multiple issues discovered in the predecessor SEV-ES extensions, new attack vectors have emerged targeting SEV-SNP systems. Fault injection vulnerabilities attacking the SP can be exploited to extract SEV-SNP secrets and decrypt memory assumed to be private. As this severely impacts the remote attestation process, it cannot be relied on for secure report creation on current generation AMD processors [8]. Additional side channel attacks are possible, because SEV-SNP does not prevent read access to (encrypted) VM private pages, which can be used to leak guest register values or recover secret keys [23]. There are currently no known vulnerabilities targeting Intel TDX, as no implementing hardware exists yet.

High demand for running applications in a TEE without the need to modify them also lead to the creation of Gramine and Secure

CONtainer Environment (SCONE). Both provide a secure environment by relying on SGX enclaves, making use of the confidentiality guarantees. Gramine is a library OS which can encapsulate an application, while keeping a low memory footprint [26] and handling all OS functionality that the application might require. SCONE allows for applications to be run inside a secure Docker container and keeps the TCB small by providing a C library [7].

IBM is working on providing comparable integrity guarantees, as ensured by SEV-SNP and TDX, for Power ISA based computers using a VM-based TEE called protected execution facility. It is supported in IBM POWER9 chips available since 2017 and increases access control restrictions and relies on several existing technologies, as it leverages secure boot and a trusted platform module [14].

5 CONCLUSION

SEV-SNP and TDX build on the memory confidentiality guarantees offered by SEV-SEM and SGX by enforcing additional integrity assurances. The switch to running applications in VM based TEEs is driven by the reduced development efforts, as no adjustments to application code are required. These new technologies allow secure processing of sensitive user data on systems administrated by external actors and mark an important milestone in the ongoing push towards trusted execution environments in the cloud.

REFERENCES

[1] AMD. 2008. AMD-V™ Nested Paging. http://developer.amd.com/wordpress/media/2012/10/NPT-WP-1%201-final-TM.pdf

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

[3] AMD. 2020. Secure Encrypted Virtualization API Version 0.24. https://www.amd.com/system/files/TechDocs/55766_SEV-KM_API_Specification.pdf

[4] AMD. 2021. AMD EPYC™ 7003 series processors. https://www.amd.com/en/processors/epyc-7003-series

[5] AMD. 2022. AMD64 Architecture Programmer's Manual Volume 3: General-Purpose and System Instructions. https://www.amd.com/system/files/TechDocs/24594.pdf

[6] AMD. 2022. SEV Secure Nested Paging Firmware ABI Specification. https://www.amd.com/system/files/TechDocs/56860.pdf

[7] Sergei Arnautov, Bohdan Trach, Franz Gregor, Thomas Knauth, Andre Martin, Christian Priebe, Joshua Lind, Divya Muthukumaran, Dan O'Keeffe, Mark L. Stillwell, David Goltzsche, David Eyers, Rüdiger Kapitza, Peter Pietzuch, and Christof Fetzer. 2016. SCONE: Secure Linux Containers with Intel SGX. In *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation* (Savannah, GA, USA) *(OSDI'16)*. USENIX Association, USA, 689–703. https://dl.acm.org/doi/10.5555/3026877.3026930

[8] Robert Buhren, Hans Niklas Jacob, Thilo Krachenfels, and Jean-Pierre Seifert. 2021. One Glitch to Rule Them All: Fault Injection Attacks Against AMD's Secure Encrypted Virtualization. https://arxiv.org/pdf/2108.04575.pdf

[9] Alpus Chen. 2022. *Configuring Virtual Intel Software Guard Extensions vSGX) in VMware ESXi on Lenovo ThinkSystem Servers.* Retrieved 16 December, 2022 from https://lenovopress.lenovo.com/lp1639.pdf

[10] Victor Costan and Srinivas Devadas. 2016. Intel SGX explained. *Cryptology ePrint Archive* (2016). https://eprint.iacr.org/2016/086.pdf

[11] Tom Woller David Kaplan, Jeremy Powell. 2016. AMD Memory encryption. https://amd.wpenginepowered.com/wordpress/media/2013/12/AMD_Memory_Encryption_Whitepaper_v9-Public.pdf

[12] Nathaniel McCallum et al. 2022. *Rust library exposing APIs for the AMD SEV platform.* Retrieved 19 December, 2022 from https://github.com/virtee/sev

[13] Torvards et al. 2022. *Linux Kernel changelog 5.19.* Retrieved 17 December, 2022 from https://cdn.kernel.org/pub/linux/kernel/v5.x/ChangeLog-5.19"

[14] 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 *(EuroSys '21)*. Association for Computing Machinery, New York, NY, USA, 294–310. https://doi.org/10.1145/3447786.3456243

[15] Intel. 2016. Intel® 64 and IA-32 Architectures Software Developer's Manual Volume 3C: System Programming Guide, Part 3. https://www.intel.de/content/dam/www/public/us/en/documents/manuals/64-ia-32-architectures-software-developer-vol-3c-part-3-manual.pdf

[16] Intel. 2021. Intel® Trust Domain Extensions (Intel® TDX) Module Architecture Application Binary Interface (ABI) Reference Specification. https://cdrdv2.intel.com/v1/dl/getContent/733579

[17] Intel. 2021. White Paper: Intel Trust Domain Extensions. https://cdrdv2.intel.com/v1/dl/getContent/690419

[18] Intel. 2022. Architecture Specification: Intel® Trust Domain Extensions (Intel® TDX) Module. https://cdrdv2.intel.com/v1/dl/getContent/733568

[19] Intel. 2022. Device Attestation Model in Confidental Computing Environment. https://cdrdv2.intel.com/v1/dl/getContent/742533

[20] Intel. 2022. Guest Hypervisor Communication Interface (GHCI) for Intel Trust Domain Extensions (Intel TDX) 1.5. https://cdrdv2.intel.com/v1/dl/getContent/726792

[21] Intel. 2022. Intel® 64 and IA-32 Architecture Software Developer's Manual Volume 4: Model-Specific Registers. https://www.intel.com/content/www/us/en/develop/download/intel-64-and-ia-32-architectures-software-developers-manual-volume-4-model-specific-registers.html

[22] David Kaplan. 2017. Protecting VM register state with SEV-ES. https://www.amd.com/system/files/documents/amd-epyc-7003-series-datasheet.pdf

[23] Mengyuan Li, Luca Wilke, Jan Wichelmann, Thomas Eisenbarth, Radu Teodorescu, and Yinqian Zhang. 2022. A Systematic Look at Ciphertext Side Channels on AMD SEV-SNP. In *2022 IEEE Symposium on Security and Privacy (SP)*. 337–351. https://doi.org/10.1109/SP46214.2022.9833768

[24] Jämes Mé nétrey, Christian Göttel, Anum Khurshid, Marcelo Pasin, Pascal Felber, Valerio Schiavoni, and Shahid Raza. 2022. Attestation Mechanisms for Trusted Execution Environments Demystified. In *Distributed Applications and Interoperable Systems*. Springer International Publishing. https://doi.org/10.1007/978-3-031-16092-9_7

[25] Timothy Merrifield and H Reza Taheri. 2016. Performance implications of extended page tables on virtualized x86 processors. In *Proceedings of the12th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments*. https://dl.acm.org/doi/abs/10.1145/2892242.2892258

[26] Chia-Che Tsai, Kumar Saurabh Arora, Nehal Bandi, Bhushan Jain, William Jannen, Jitin John, Harry A. Kalodner, Vrushali Kulkarni, Daniela Oliveira, and Donald E. Porter. 2014. Cooperation and Security Isolation of Library OSes for Multi-Process Applications *(EuroSys '14)*. Association for Computing Machinery, New York, NY, USA, Article 9, 14 pages. https://doi.org/10.1145/2592798.2592812

[27] VMware. 2021. *vSGX Overview.* Retrieved 16 December, 2022 from https://docs.vmware.com/en/VMware-vSphere/7.0/com.vmware.vsphere.security.doc/GUID-EF552A5E-744B-4FD4-85AB-07B3CB22EF3E.html

A PAGE TABLE STATES

All pages tracked in the RMP are categorized using a page state attribute. Pages can only be in one of the following states shown in Table 1.

Pages in the `Hypervisor` state correspond to the aforementioned shared (unencrypted) pages, which can be freely accessed by the VMM or by SEV-SNP VMs. *Guest-Valid* pages are assigned to a SEV-SNP VM and were successfully validated by the guest and are thus marked as private pages. Transitioning between the different states is possible by calling new CPU instructions, e.g. `PVALIDATE` or `RMPUPDATE`, or by using the VM management API included in the AMD secure processor [2]. `PVALIDATE` is used within guests to move pages from the *Guest-Invalid* to the *Guest-Valid* state, verifying them in the process. This is typically done by the guest after receiving pages assigned using the `RMPUPDATE` instruction, as it clears the validated bit, marking the page as untrusted. Assigning pages from the hypervisor to a SEV-SNP enabled guest or from a guest back to the VMM can be done using the previously mentioned `RMPUPDATE` instruction. Modifying the RMP directly from software is forbidden by design and is only possible by the AMD Secure Processor, as it is essential for memory access control [6].

State	Description
Hypervisor	Default state for unassigned memory
Guest-Invalid	Unvalidated page assigned to guest
Guest-Valid	Page validated by guest
Pre-Guest	Page owned by AMD-SP before guest assignment
Pre-Swap	Used to prepare page swap
Firmware	Used for pages not yet configured by AMD-SP
Metadata	Metadata necessary for page swap
Context	Used by AMD-SP to hold VM metadata

Table 1: Table adapted from AMD SEV-SNP whitepaper [2]

Moving a page into the `Metadata`, `Firmware` or `Context` state requires calling into the AMD SP's management API, which makes the page immutable and configures it to the requested state.

B DEVELOPMENT EXPERIENCE

To develop software making use of these newly introduced capabilities, as provided by SEV-SNP and TDX, requires a stable function and instruction specification and hardware supporting these extensions. While AMD already offers processors with enabled SEV-SNP support, Intel still lacks any hardware support in their current generation chips. However, they have already issued an application binary interface (ABI) for TDX [16], enabling developers to start the software development process. AMD exposes API access to the SP using a rust library, which supports SEV-SNP since October 2022 [12]. Making use of these libraries and tools allows the creation of VM's and communication with the necessary secure components. Certificates for validating attestation reports are issued by the chip maker and are publicly available for each chip generation.

Performance may decrease due to the new components used during the page table walk, as they perform additional access verification for memory writes to guest private pages. The real world impact has not yet been measured, as the very recent and ongoing development efforts by AMD and Intel are still affecting the implementation details. Previously discussed CPU measurement instructions, such as `CPUID`, are also getting slower due to additional validation efforts provided by the SP and TDX module.

Received 9 January 2023