Protecting Nested VMs with AMD SEV

Kazuki Takiguchi
Kyushu Institute of Technology
takiguchi@ksl.ci.kyutech.ac.jp

Kenichi Kourai
Kyushu Institute of Technology
kourai@ci.kyutech.ac.jp

1 Background

As cloud services that provide virtual machines (VMs) are widely used, sensitive data is now being handled in clouds. This increases the risk of sensitive data being stolen by cloud insiders. To address this issue, modern clouds offer confidential VMs using trusted execution environments (TEE) such as AMD SEV. SEV transparently encrypts the memory of a VM and decrypts it only within the VM. Therefore, even cloud insiders cannot eavesdrop on sensitive data stored in the memory of a VM.

In parallel, nested virtualization is used for various systems such as virtual clouds and secure monitoring systems. It allows a VM (L1 VM) to run nested VMs (L2 VMs) within it. However, SEV support is still limited to L2 VMs. Microsoft's patch [2] can apply SEV-SNP only to L2 VMs. Hecate [1] can apply SEV-SNP to both L1 and L2 VMs but support only one L2 VM. Both run on top of Hyper-V.

2 Nested SEV

We propose nested SEV, which enables an L1 VM protected by SEV to run L2 VMs protected by SEV. As shown in Figure 1, nested SEV can run the hypervisor inside an L1 VM. This makes it possible to apply SEV to L2 VMs running on it. Nested SEV supports both system configurations that use different keys for memory encryption of an L1 VM and L2 VMs and that use the same key. Using different encryption keys can protect L2 VMs from the hypervisor within the L1 VM, while using the same key permits the hypervisor within the L1 VM to access L2 VMs. Nested SEV also supports SEV-ES and SEV-SNP. These extensions provide greater security but also have overhead, so they can be used in different ways depending on the requirements.

To achieve these system configurations, nested SEV provides two methods: SEV virtualization and SEV passthrough. SEV virtualization applies virtual SEV to L2 VMs. For this purpose, a virtual AMD secure processor (AMD-SP) is provided to an L1 VM and manages the encryption keys of virtual SEV via the physical AMD-SP. In contrast, SEV passthrough directly applies SEV used in the L1 VM to L2 VMs, encrypting the memory of both L1 and L2 VMs. We have implemented nested SEV for various hypervisors as in Table 1.

To evaluate the performance of nested SEV, we sent requests to a web server within a VM. We measured the performance with SEV, SEV-ES, and SEV-SNP enabled and SEV disabled. The hypervisor used was KVM, and the CPUs used

Table 1: Implementation status.

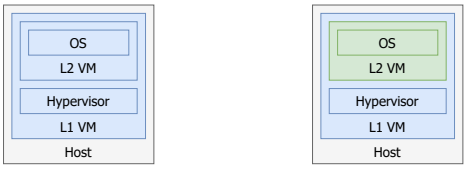
	SEV virtualization			SEV passthrough		
Hypervisor	SEV	ES	SNP	SEV	ES	SNP
KVM	✓	✓	✓	✓	✓	✓
Xen (para-virt)	-	-	-	✓	✓	✓
BitVisor	✓			✓		

were the 3rd and 4th generation AMD EPYC processors. The request processing performance of the web server is shown in Figure 2. The results indicate that the performance degraded in the order of SEV, SEV-ES, and SEV-SNP. Comparing the 3rd and 4th generations, the performance gap between SEV-ES and SEV-SNP has narrowed, suggesting that hardware implementation may have been improved.

References

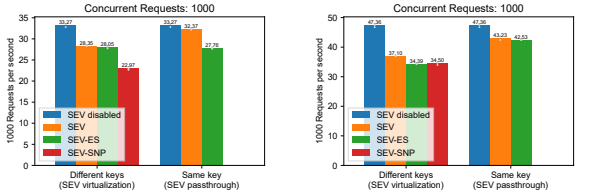
[1] Xinyang Ge, Hsuan-Chi Kuo, and Weidong Cui. 2022. Hecate: lifting and shifting on-premises workloads to an untrusted cloud. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security* (CCS '22), 1231–1242.

[2] Jeremi Piotrowski. [RFC PATCH v2 0/7] Support nested SNP KVM guests on Hyper-V. https://lore.kernel.org/lkml/20230213103402.1189285-1-jpiotrowski@linux.microsoft.com/, (2023).



(a) Same key (b) Different keys

Figure 1: System configurations using nested SEV.



(a) 3rd generation AMD EPYC (b) 4th generation AMD EPYC

Figure 2: The performance of the web server.