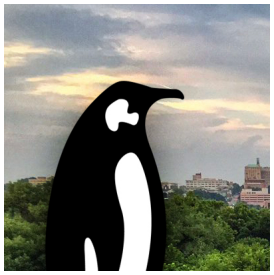

Contribution ID: **117** Type: **not specified**

Taming the Incoherent Cache Issue in Confidential VMs

Tuesday, 14 November 2023 17:00 (15 minutes)

It is well known that in AMD CPUs prior to Milan, cache lines within a confidential VM are incoherent with those outside of confidential VMs. SME_COHERENT is a feature introduced by AMD 3rd gen EPYC to improve cache coherency in their confidential computing environment. However, as testing demonstrates, SME_COHERENT does not support cache coherence between CPU and devices. This means that guest pages which were previously used for DMA may still contain dirty caches incoherent with cache lines generated by the CPUs. Since KVM does not track the page provenance, skipping the cache flush may lead to memory corruptions at host level when the guest pages are freed.

This problem is even worse when malicious or subverted host userspace could leverage the existing KVM API to fill confidential pages with dirty caches and free them without flushing. The security problem was recorded as CVE-2022-0171 [1]. Because of the limitation of SME_COHERENT, this vulnerability was not just limited to 2rd gen EPYC but 3rd gen EPYC and later. Upstream Linux solves this issue by flushing the cache unconditionally when a guest page mapping was removed from a VM's NPT.

Flushing the cache lines is a principled approach working with a confidential VM since confidential guest pages are pinned and thus cannot be moved by the host OS i.e., the guest pages are not leaving the VM until the VM dies. However, due to limitations of MMU API, there is no API telling KVM a guest page is deallocated. The most relevant APIs are the mmu_notifiers which were invoked when the guest page mappings were removed during the VM lifetime. In fact, guest mapping could be removed due to various legitimate reasons: changing page granularity from one to another, e.g., from 2M/1G to 4K and vice versa; page migration due to NUMA balancing, defragmentation and others like KSM.

The host capability to change the mapping of the guest VM creates performance problems with the existing upstream solution for CVE-2022-0171, as KVM may unnecessarily flush the cache lines in some of the scenarios mentioned above. This is reported at [2][3] when running with SEV-SNP VMs.

In this talk, we will discuss the prospective solutions to this problem, such as how we should nicely flush cache lines without introducing performance bottlenecks. One solution would be to use VMPAGE_FLUSH MSR instead of wbinvd, the latter of which requires the whole machine wise cache flush. This should improve performance in general with the cost of slowing down the shutdown of individual VMs. On the other hand, it might be feasible to leverage the "reason"parameter of mmu_notifiers to conditionally flush the cache. We will discuss further details during the talk.

References

[1] https://bugzilla.redhat.com/show_bug.cgi?id=2038940

[2] https://lore.kernel.org/all/876a7707-c9b9-0985-af00-c7fc461ada02@windriver.com/

[3] https://lore.kernel.org/kvm/YzJFvWPb1syXcVQm@google.com/T/#mb79712b3d141cabb166b504984f6058b01e30c63

Primary authors: ZHANG, Mingwei (Google); LI, Jacky (Google); CHRISTOPHERSON, Sean (Google)

Presenters: ZHANG, Mingwei (Google); LI, Jacky (Google); CHRISTOPHERSON, Sean (Google)

Session Classification: Confidential Computing MC

Track Classification: LPC Microconference: Confidential Computing MC