Contribution ID: **205** Type: **not specified**

Removing guest memory from the host kernel's direct map

Thursday, 19 September 2024 12:30 (30 minutes)

Since the discovery of Spectre and Meltdown in 2018, transient execution attacks are being discovered regularly, both in old and new hardware. Mitigation involves applying specific patches for each vulnerability, and is often costly in terms of performance, leading to cloud computing providers to seek more general mitigations.

The majority of these attacks are based on the presence of a machine's entire physical memory in host kernel address space. Carefully crafted malicious software may influence CPU execution by mistraining branch predictor units so that the CPU speculatively accesses data in the kernel context and leaves non-architectural side effects of that activity, such as loading certain data in the CPU cache, which can be observed by the attacker to infer sensitive content.

We propose mitigating these attacks by removing page table mappings of sensitive memory regions from kernel address space, thus preventing malicious speculative loads and their side effects altogether. This makes memory immune to a large class of both known and not-yet-discovered transient execution attacks. We will discuss KVM patch series for securing the entirety of a virtual machine's memory against these types of issues, by extending KVM's guest_memfd to remove its memory from the kernel's direct map. guest_memfd is a fd-based backend for guest memory (as opposed to the traditional VMA-based backend) introduced in Linux 6.8, inspired by confidential compute technologies such as Intel TDX and AMD SEV-SNP, which we are interested in extending to the non-CoCo usecase.

Primary authors: KALYAZIN, Nikita (AWS); ROY, Patrick (Amazon UK)

Co-authors: GRAF, Alexander; GOWANS, James (Amazon EC2)

Presenter: ROY, Patrick (Amazon UK)

Session Classification: KVM Microconference

Track Classification: KVM Microconference