

CVE-2026-98468

Microsoft KVM: arm64: Fix spurious warning for benign stage 2 teardown race

In the Linux kernel, the following vulnerability has been resolved: KVM: arm64: Fix spurious warning for benign stage 2 teardown race kvmtool was used to establish an L1 guest with 8 CPUs and 8 GiB of RAM, an L2 guest with 4 CPUs and 4 GiB of RAM and an L3 guest with 2 CPUs and 2 GiB of RAM, all of which was then exited. Under memory pressure in the L0 host warnings were observed due to migration triggered by compaction: WARNING: arch/arm64/kvm/mmu.c:336 at __unmap_stage2_range+0x64/0x80, CPU#5: kcompactd0/66 Which

EVRS 0 /100 Informational priority	CVSS — Not scored	EPSS 0.002 9.3rd percentile · FIRST	CISA DEADLINE — No CISA deadline
---	---------------------------------------	---	--

Signals behind the score

☒ counted in the score ☐ not counted

evidence for 2 of 8 signals

☐ CISA KEY	Not currently CISA KEY-listed	☐ Vendor exploitation	Active-exploitation status not yet checked - scored at the population base rate (none) until known
☐ Ransomware	Ransomware association unavailable	☐ KEV recency	CISA KEY listing date unavailable
☐ EPSS	0.002 probability	☐ Pre-patch disclosure	Disclosure status unavailable
☐ CVSS	CVSS base score unavailable	☐ Microsoft Exploitability Index	Not a Microsoft CVE

Remediation

Monitor Untagged reference available; patch status is not confirmed

NVD lists one or more references for this CVE that aren't tagged as a patch or vendor advisory. Review them directly to confirm patch availability and applicability before making a remediation decision.

Basis: NVD reference (untagged)

was, in turn, triggered by an MMU notifier for the host invalidation: mmu_notifier_invalidate_range_start() -> ... -> kvm_mmu_notifier_invalidate_range_start() -> kvm_mmu_unmap_gfn_range() -> kvm_unmap_gfn_range() -> kvm_nested_s2_unmap() -> kvm_stage2_unmap_range() -> __unmap_stage2_range() -> stage2_apply_range() <- -EINVAL, triggering a WARN_ON() Racing with LO's teardown of stage 2 page tables: exit_mm() -> mmpu() -> __mmpu() -> exit_mmap() -> mmu_notifier_release() -> ... -> kvm_mmu_notifier_release() -> kvm_flush_shadow_all() -> kvm_arch_flush_shadow_all() -> kvm_free_stage2_pgd() -> [acquire kvm->mmu_lock for write] -> mmu->pgt = NULL [among other tasks] -> [release kvm->mmu_lock for write] It turns out there is a benign race resulting in a spurious warning: Thread A - notify: migration | Thread B - notify: release -----|----- < kvm->mmu_lock held > | stage2_apply_range() | get mmu->pgt, check !NULL | ... | kvm_arch_flush_shadow_all() cond_resched_rwlock_write(); | < contend, sleep kvm->mmu_lock > < drop kvm->mmu_lock > | < acquire kvm->mmu_lock > | ... | kvm_free_stage2_pgd() | mmu->pgt = NULL | < invalidate MMU > | ... | < release kvm->mmu_lock > [scheduled] | stage2_apply_range() | < loop to next > | get, mmu->pgt, check !NULL | is NULL, return -EINVAL | __unmap_stage2_range() | WARN_ON(-EINVAL) <--- entirely spurious - the race was handled correctly. Fix the spurious warning by updating stage2_apply_range() to no longer treat concurrent PGT teardown on lock release as an error - whether the walker is tearing down page tables or doing something else this is a legitimate reason to abort the operation without error. This keeps the warning in place for all other circumstances. In practice only __unmap_stage2_range() actually does anything with the error so this only impacts that.



Affected and fixed versions

PACKAGE	AFFECTED	FIXED IN	SOURCE
Not identified by the connected sources	–	no fixed version listed	NVD

Sources

Reference	git.kernel.org/stable/c/330346dda3bf5f8a3b06723024f7d906326270cf	Additional reference	git.kernel.org/stable/c/38b70fc453c3112f1a62583b89903ae41116cc27
Additional reference	git.kernel.org/stable/c/c23ed110d27ba99cf3a881986514f79aacedf476	Additional reference	git.kernel.org/stable/c/f44a92633136b409ebb9b7036fceb7e11de60a20
NVD	nvd.nist.gov/vuln/detail/CVE-2026-98468	ENISA EUVD	euvd.enisa.europa.eu/vulnerability/EUVD-2026-103708
Current data	evulnable.com/vulnerability-management/cve/CVE-2026-98468		

Snapshot at generation time. EPSS can move quickly. A public exploit module is not evidence of active exploitation.