

CVE-2026-98510

Microsoft drm/xe/vm: nuke PTs only after unlinking contested VMAs

In the Linux kernel, the following vulnerability has been resolved: drm/xe/vm: nuke PTs only after unlinking contested VMAs In `xe_vm_close_and_put()`, external-BO VMAs are queued on the contested list for deferred destruction via `xe_vma_destroy_unlocked()`. However, `xe_vm_pt_destroy()` was previously invoked before processing contested VMAs, destroying `vm->pt_root` while those VMAs were still linked to their respective buffer objects (`vm_bo->list.gpuva`).

EVRS 0 /100 Informational priority	CVSS — Not scored	EPSS 0.002 9.2nd percentile · FIRST	CISA DEADLINE — No CISA deadline
--	---------------------------------------	---	--

Signals behind the score

☒ counted in the score ☐ not counted

evidence for 2 of 8 signals

☐ CISA KEV	Not currently CISA KEV-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 KEV 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)

If a concurrent thread evicts one of those shared buffer objects, `xe_bo_trigger_rebind()` holding only `bo->resv` walks the BO's VMAs and, in fault mode, calls `xe_vm_invalidate_vma()` -> `xe_pt_zap_ptes()`. Because `vm->pt_root[tile->id]` is already NULL, dereferencing `pt->level` causes a NULL ptr deref. Fix this by deferring `xe_vm_free_scratch()` and `xe_vm_pt_destroy()` until after all contested VMAs have been unlinked and destroyed. User is reporting hitting a NULL ptr deref in `xe_pt_zap_ptes()`, which could be explained by this race. (cherry picked from commit `c2863648959489767f08892fd6e90577d2ea0b6a`)

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/24a22fb3c731474b68e986af6804db450fb88617	Additional reference	git.kernel.org/stable/c/70fddc065ac80845ffd33095b3635d59883cf7e6
Additional reference	git.kernel.org/stable/c/ba678c2b02e0d36b53deb5a6d5e5ea044d36def7	Additional reference	git.kernel.org/stable/c/bddaa5e4094c134452c29e20a26c4718827e4cb1
NVD	nvd.nist.gov/vuln/detail/CVE-2026-98510	ENISA EUVD	euvd.enisa.europa.eu/vulnerability/EUVD-2026-103750
Current data	evulnable.com/vulnerability-management/cve/CVE-2026-98510		

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