

CVE-2026-98683

Microsoft smb: client: flush dirty data before zeroing a range

In the Linux kernel, the following vulnerability has been resolved: smb: client: flush dirty data before zeroing a range
smb3_zero_range() emulates FALLOC_FL_ZERO_RANGE by invalidating the page cache over the target range with truncate_pagecache_range() and then issuing FSCTL_SET_ZERO_DATA to the server. Dirty data was only flushed conditionally, when the range reached or extended EOF.

EVRS 0 /100 Informational priority	CVSS — Not scored	EPSS 0.002 5.5th 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)

For a purely interior zero range that does not reach EOF, no flush happened, so a dirty folio overlapping the range could be written back to the server after the FSCTL and refill the range that was just zeroed with stale data. Fix this by unconditionally flushing and waiting for dirty data in the range before invalidating the page cache and issuing FSCTL_SET_ZERO_DATA, exactly as was done for smb3_punch_hole() in commit d7d2adcd022b ("smb/client: flush dirty data before punching a hole"); the two paths are structurally identical here. This is observed as generic/363 randomly reading stale data where a zeroed range is expected against Windows Server.

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/8f8d32c1044f70e46e87550e039a15ff20a4b21c	Additional reference	git.kernel.org/stable/c/cecf4114069fc19629d284710fd783d56bbca862
Additional reference	git.kernel.org/stable/c/e8ba21c7e6ef3d4cb617858760673877b2e4a2bb	NVD	nvd.nist.gov/vuln/detail/CVE-2026-98683
ENISA EUVD	euvd.enisa.europa.eu/vulnerability/EUVD-2026-103937	Current data	evulnable.com/vulnerability-management/cve/CVE-2026-98683

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