

CVE-2026-98682

Microsoft smb: client: drain and invalidate before server-side copy/clone

In the Linux kernel, the following vulnerability has been resolved: smb: client: drain and invalidate before server-side copy/clone
cifs_file_copychunk_range() and the clone (FICLONE) path of cifs_remap_file_range() do not serialise the destination page cache against the server-side copy the way the other server-side range operations (smb3_zero_range(), smb3_punch_hole(), smb3_collapse_range() and smb3_insert_range()) do.

EVRS 0 /100 Informational priority	CVSS — Not scored	EPSS 0.002 4.3rd percentile · FIRST	CISA DEADLINE — No CISA deadline
---	---------------------------------------	---	--

Signals behind the score

☒ counted in the score ☐ not counted

evidence for 2 of 8 signals

☐ CISA 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)

cifs_file_copychunk_range() invalidates the destination range with filemap_invalidate_inode(), which takes and drops the mapping's invalidate_lock internally, so the lock is no longer held when the copychunk ioctl is issued. It also never drains in-flight netfs I/O on the target. The clone path never takes the invalidate_lock at all (only i_rwsem), uses a bare truncate_inode_pages_range() and likewise does not drain outstanding I/O. As a result an asynchronous destination writeback can complete after the server-side copy/clone has run and reinstate stale data over the region just written by the server, corrupting the file. This is the same class of corruption as commit d7d2adcd022b ("smb/client: flush dirty data before punching a hole") and has been seen randomly in generic/363 against Windows Server. Fix both paths to follow the established ordering: hold the target mapping's invalidate_lock across the flush and invalidation of the destination and the server ioctl, and call netfs_wait_for_outstanding_io() on the target to drain in-flight writes before the ioctl is issued. Only the target inode's invalidate_lock is required, as the source is merely flushed and not invalidated; i_rwsem (already held via lock_two_nondirectories()) is acquired before the invalidate_lock, matching the VFS lock ordering. Since filemap_invalidate_inode() takes that same lock internally, replace it with its own unmap/flush/invalidate steps instead of calling it, and return early on a zero-length copy to avoid a range underflow.



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/ c9b1194e8ec99d95db25db45edd2fbf690f043ab	Additional reference	git.kernel.org/stable/c/ dc71660f11ef2fecc0695a41e49cd496a9d255ab
NVD	nvd.nist.gov/vuln/detail/CVE-2026-98682	ENISA EUVD	euvd.enisa.europa.eu/vulnerability/ EUVD-2026-103936
Current data	evulnable.com/vulnerability-management/cve/ CVE-2026-98682		

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