

CVE-2026-98845

Microsoft tpm: Disable TPM on null key name mismatch

In the Linux kernel, the following vulnerability has been resolved: tpm: Disable TPM on null key name mismatch The null key name check exists to protect against TPM reset attacks, so a mismatch should stop the device from serving further requests. Currently it does not disable the chip when it finds a mismatch. The mismatch is logged: tpm tpm0: null key integrity check failed but the chip keeps serving commands: / # tpm2_getcap -c properties-fixed TPM_PT_FAMILY_INDICATOR: as UINT32: 0x08322e3000 as string: "2.0"

EVRS 0 /100 Informational priority	CVSS — Not scored	EPSS 0.002 4.4th 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)

... tpm2_load_null() where the null key name check is run sets the chip as disabled only if the rc is non zero. When the mismatch is seen, rc is zero at that point and it returns success. The other issue is that the caller expects the null key handle to be populated when the function returns 0 which it does here without writing the handle and proceeds assuming the null key handle is valid. During the test, I noticed that tpm2_start_auth_session() uses the uninitialized stack value as the key handle since tpm2_load_null() returns 0 despite the integrity failure and proceeds with TPM2_CC_START_AUTH_SESS with this value as salt key handle. Set rc to -ENODEV on the mismatch. The error path then disables the chip and returns the correct code to the caller. Tested in QEMU with swtpm and CONFIG_TCG_TPM2_HMAC=y by making TPM2_CC_CONTEXT_LOAD fail with TPM2_RC_INTEGRITY and changing the name of the re-created null key. The chip is now disabled on the mismatch.

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/015fb29a748342a186f37b602ac70017098c8251	Additional reference	git.kernel.org/stable/c/96c46c75b0d470b4204f9fe1bdb075cd2ebaebfa
Additional reference	git.kernel.org/stable/c/a03d0c64cd827a622578623a78defd66b6593b82	Additional reference	git.kernel.org/stable/c/f6599d8830fe7ddab7334cc17846fd9ac9a7fa50
NVD	nvd.nist.gov/vuln/detail/CVE-2026-98845	ENISA EUVD	euvd.enisa.europa.eu/vulnerability/EUVD-2026-103785
Current data	evulnable.com/vulnerability-management/cve/CVE-2026-98845		

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