

CVE-2026-98616

Microsoft eth: fbnic: reset num_napi when the napi vectors are freed

In the Linux kernel, the following vulnerability has been resolved: eth: fbnic: reset num_napi when the napi vectors are freed fbn->num_napi is the count of live napi vectors, each of which owns an IRQ. The PM path had freed them without clearing the count. fbnic_pm_suspend() tears the datapath down via ndo_stop() and frees the IRQs, but leaves netif_running() true so resume knows to re-open.

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)

Resume rebuilds the datapath in __fbnic_pm_resume() and fbnic_reset_queues() sets num_napi and __fbnic_open() re-allocates the vectors. When the datapath is torn down but never rebuilt, num_napi is left pointing at freed vectors under 2 different scenarios: - a PCIe error recovery that fails (fbnic_err_slot_reset() -> __fbnic_pm_resume() returns an error -> PCI_ERS_RESULT_DISCONNECT), so .resume never runs; or - an __fbnic_open() that fails partway on resume and unwinds, freeing the vectors after fbnic_reset_queues() has already set num_napi. The netdev is then running with num_napi > 0 but napi[] freed, and the eventual remove/unbind close re-enters fbnic_down() -> fbnic_dbg_down() and dereferences the freed vectors: BUG: kernel NULL pointer dereference, address: 0000000000000210 RIP: fbnic_dbg_down+0x28 Clear num_napi when the vectors are freed: in the suspend teardown (a good resume re-establishes it before __fbnic_open()) and on the resume open failure. A redundant ndo_stop() then walks an empty napi[]. The normal ndo_stop() down/up cycle is untouched and keeps num_napi for the next ndo_open().

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/4bcc4a92c603fe7f062cea22e20da2e0ad6b12c3	Additional reference	git.kernel.org/stable/c/4f4b767ffea7653a00904383908536e754f8ef5b
Additional reference	git.kernel.org/stable/c/c5b5aee41b9f5ed0743344be1c7b544555fa3cb9	NVD	nvd.nist.gov/vuln/detail/CVE-2026-98616
ENISA EUVD	euvd.enisa.europa.eu/vulnerability/EUVD-2026-103661	Current data	evulnable.com/vulnerability-management/cve/CVE-2026-98616

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