

CVE-2026-98617

Microsoft eth: fbnic: use the Rx queue napi pointer to find the napi vector

In the Linux kernel, the following vulnerability has been resolved: eth: fbnic: use the Rx queue napi pointer to find the napi vector The queue management ndos pick the napi vector for an Rx queue with: `nv = fbn->napi[idx % fbn->num_napi]`; The issue is this is only correct in the cases where there are no standalone Tx vectors. In those cases we were allocating the Tx vectors first and then the Rx so the queues would be pointing to Tx NAPI vectors instead of the Rx ones.

EVRS 0 /100 Informational priority	CVSS — Not scored	EPSS 0.002 8.1st 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)

The mapping the ndos want is already recorded. `fbnic_set_netif_napi()` publishes it with `netif_queue_set_napi()`, which stores the napi pointer in `netdev_rx_queue.napi`, and `fbnic_reset_netif_napi()` clears it again. Both run under the `netdev` instance lock that the queue management ndos also hold, so the pointer can be read directly. Use it and drop the divide. The pointer is NULL exactly while the datapath is down, so `fbnic_queue_mem_alloc()` can reject that case rather than reaching into freed state: `netdev_rx_queue_restart()` calls it before it tests `netif_running()`, and `fbnic_pm_suspend()` leaves `netif_running()` true across a PCIe recovery that never completes, so a queue restart can arrive after `fbnic_stop()` has freed the rings and the vectors. `fbnic_stop()` clears the association in `fbnic_reset_netif_queues()` before `fbnic_free_napi_vectors()`, so the NULL is always published first. `fbnic_queue_start()` and `fbnic_queue_stop()` need no check of their own, as `netdev_rx_queue_reconfig()` only reaches them once `fbnic_queue_mem_alloc()` has succeeded under the same instance lock.

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/1fb569cf3bf97e552f465b2c719e36b48ae7c18b

NVD nvd.nist.gov/vuln/detail/CVE-2026-98617

Current data evulnable.com/vulnerability-management/cve/CVE-2026-98617

Additional reference git.kernel.org/stable/c/b5d9e9d4d0c13bc8b60d8d97e7a07fb25fea639e

ENISA EUVD euvd.enisa.europa.eu/vulnerability/EUVD-2026-103662

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