

CVE-2026-98613

Microsoft bpf: Bound ownership depth through local kptrs and graph roots

In the Linux kernel, the following vulnerability has been resolved: bpf: Bound ownership depth through local kptrs and graph roots Program-allocated objects can own other local objects through referenced kptrs. bpf_obj_free_fields() follows those pointers through __bpf_obj_drop_impl() synchronously, before the object storage is freed through RCU. A self-referential local kptr type therefore permits arbitrarily deep object chains, and dropping the head can exhaust the kernel stack.

EVRS 0 /100 Informational priority	CVSS — Not scored	EPSS 0.002 6.7th 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)

Long acyclic type chains have the same problem. btf_check_and_fixup_fields() still assumes referenced kptrs only point to kernel types and checks ownership through list and rbtree roots only. Its existing rule is sufficient for graph-only cycles: the target of each graph edge must contain a node, so every type in a cycle has both a root and a node. The rule rejects such a type owning another root, breaking every cycle. It also limits graph-only chains to three types, or two if the first type contains a node, and conservatively rejects longer acyclic chains. The missing local-kptr edges, rather than a missed graph-only cycle, are the bug introduced by support for bpf_kptr_xchg() into local kptrs. Replace that restriction with one bounded ownership walk covering graph roots and local referenced kptrs. Run it after all BTF records have been fixed up, reject cycles and paths deeper than eight record-bearing types, and cache each type's suffix depth while checking it against the remaining budget. This also permits the longer acyclic graph-only layouts rejected by the old rule; update their existing BTF tests accordingly. Keep the bound independent of MAX_CALL_FRAMES because recursive destruction can run below a BPF call chain. A plain local pointee without special-field metadata adds only a final non-recurring drop. Non-owning kptrs and kernel-BTF kptrs do not recurse through local records and remain outside the walk. Include local percpu-kptr edges too, although allocation of percpu objects with special fields is currently forbidden, so that relaxing that restriction cannot bypass the ownership bound. btf_check_and_fixup_fields() continues to initialize graph_root.value_rec, including for separately allocated map records. The ownership relationships belong to immutable program BTF and only need validation at BTF load time.



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/4105870a4b8afe852aeb24a6dd58e1836cc4ce58	Additional reference	git.kernel.org/stable/c/92aa0502c781d72bb9685245c3280efb3766a110
Additional reference	git.kernel.org/stable/c/9c11e543fec32e92b3aa3f693339ca3ee79ea9dd	Additional reference	git.kernel.org/stable/c/bfc888f04588f591851e95c974954cfca58e6c19
NVD	nvd.nist.gov/vuln/detail/CVE-2026-98613	ENISA EUVD	euvd.enisa.europa.eu/vulnerability/EUVD-2026-103658
Current data	evulnable.com/vulnerability-management/cve/CVE-2026-98613		

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