CVE-2026-32145

Multipart form body parser bypasses body size limits in wisp

« Back to all CVEs

See on OSV.dev »

Weakness Type (CWE)

CWE-770 — CWE-770 Allocation of Resources Without Limits or Throttling

CAPEC

CAPEC-130 — CAPEC-130 Excessive Allocation

CVSS 4.0 Score

8.7

HIGH

CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:N/VI:N/VA:H/SC:N/SI:N/SA:N

Vulnerability description

Allocation of Resources Without Limits or Throttling vulnerability in gleam-wisp wisp allows a denial of service via multipart form body parsing.
The `multipart_body` function bypasses configured `max_body_size` and `max_files_size` limits. When a multipart boundary is not present in a chunk, the parser takes the `MoreRequiredForBody` path, which appends the chunk to the output but passes the quota unchanged to the recursive call. Only the final chunk containing the boundary is counted via `decrement_quota`. The same pattern exists in `multipart_headers`, where `MoreRequiredForHeaders` recurses without calling `decrement_body_quota`.

An unauthenticated attacker can exhaust server memory or disk by sending arbitrarily large multipart form submissions in a single HTTP request.

This issue affects wisp: from 0.2.0 before 2.2.2.

Affected

pkg:hex/wisp

Module	Source File	Routine
wisp	src/wisp.gleam	wisp:multipart_body/7 wisp:multipart_headers/5

Status	Type	Version	Changes / Fixed in
Affected	semver ⓘ	0.2.0	< 2.2.2

pkg:github/gleam-wisp/wisp

Module	Source File	Routine
wisp	src/wisp.gleam	wisp:multipart_body/7 wisp:multipart_headers/5

Status	Type	Version	Changes / Fixed in
Affected	git ⓘ	d8e722e22c	< 7a978748e1

Workarounds

Deploy a reverse proxy (such as nginx or HAProxy) in front of the wisp application and configure it to enforce request body size limits.

References

- https://github.com/gleam-wisp/wisp/security/advisories/GHSA-8645-p2v4-73r2 vendor-advisory related
- https://osv.dev/vulnerability/EEF-CVE-2026-32145 related
- https://github.com/gleam-wisp/wisp/commit/7a978748e12ab29db232c222254465890e1a4a90 patch

Credits

- **Finder**: John Downey
- **Remediation developer**: Louis Pilfold

CVE record as JSON: `GET /cves/CVE-2026-32145.json`
OSV record as JSON: `GET /osv/EEF-CVE-2026-32145.json`