TECHNICAL ACCURACY REVIEW · SAMPLE SECTION

Hardening-guide review — Content Security Policy

Olha Shevchenko · 2026-08-21 · One section of the deliverable format. The excerpt under review is illustrative, written for this sample; every claim about LibreChat is checked against public upstream sources, cited below.

SCOPE	VERDICT	VERIFICATION	SOURCES CITED
1 of 5 review areas	**Accurate but incomplete**	**Documentation** not runtime-tested	**4**

CLASSIFICATION	Critical	**Important**	Future
	Must be fixed before publication	Correct before an enterprise reader relies on it	Improvement, not a defect

SECTION REVIEWED

Content Security Policy (Helmet) for a self-hosted LibreChat deployment.

EXCERPT UNDER REVIEW · ILLUSTRATIVE

“Enable Content Security Policy through Helmet, using restrictive directives such as `default-src 'self'`, `script-src 'self'`, `object-src 'none'`, and `frame-ancestors 'self'`.”

1 WHAT IS CORRECT

CSP as defence in depth against script injection is sound. OWASP recommends server-delivered CSP and treats nonce- or hash-based policies as the preferred model; Helmet supports directives, per-request nonces, and report-only mode.

2 WHAT IS MISSING

The instruction reads as if a fixed Helmet policy is a safe generic hardening step. LibreChat’s own history says otherwise. `PR #7377` added `helmet()` with a hardcoded CSP in response to issue `#7110` and was reverted in `PR #7419` (“Temporarily Remove CSP until Configurable”) because the fixed allow-list broke deployments depending on which optional features were enabled. As of issue `#14438`, still open, configurable CSP remains unresolved, and that issue proposes shipping the CSP-independent Helmet headers separately, with CSP explicitly disabled. The missing requirement is deployment-specific policy discovery and staged enforcement.

3 CORRECTION — REPLACEMENT WORDING

> Configure Content Security Policy for the actual LibreChat deployment rather than applying a fixed generic allow-list. Enable the CSP-independent security headers (HSTS, X-Frame-Options, X-Content-Type-Options) separately so they are not lost while CSP remains deployment-specific. Build the candidate policy from the deployment’s real browser-facing surface, including enabled authentication providers, embedded tooling and external model endpoints, and deploy it in Report-Only mode first. Exercise every enabled workflow, review the violations, and move to enforcement only after the required resource graph is understood. Prefer nonce- or hash-based script controls over widening `script-src`.

4 WHY IT MATTERS

A CSP can be syntactically valid and operationally wrong in both directions: too permissive, and it protects less than the guide implies; too restrictive, and legitimate functionality fails at enforcement. LibreChat’s upstream history already contains the second failure. “Enable CSP” overstates the portability of the recommendation.

SOURCES

LibreChat issue #7110, Missing Content Security Policy (closed by PR #7377) `github.com/danny-avila/LibreChat/issues/7110`

LibreChat issue #14438, Add baseline non-CSP security headers (open; documents the #7377 implementation and the #7419 revert) `github.com/danny-avila/LibreChat/issues/14438`

OWASP Content Security Policy Cheat Sheet `cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html`

Helmet CSP documentation `github.com/helmetjs/helmet/blob/main/middlewares/content-security-policy/README.md`