SnapSmack Security Audit 049 - CMS compliance, image handling, and post-model discovery

SECURITY AUDIT / PUBLIC REMEDIATION RECORD

Assessment target: dev at 0.7.533, with disposition reviewed through 0.7.536D **Review opened:** 2026-08-17 **Review closed:** 2026-08-18 **Status:** CLOSED AS AN ASSESSMENT RECORD. Remediation status is stated per finding below. **Reviewers:** Claude Code, web-Claude, OpenAI Codex, with product and live-fleet facts supplied by Sean McCormick.

Executive conclusion

Audit 049 found two different classes of problem:

1. A conventional CMS and image-handling security review found strong controls around SQL parameters, escaping, CSRF, authentication, signed updates, secret generation, upload naming, and raster derivative generation. It also found several real gaps, chiefly public error disclosure, inconsistent HTTP-security-header delivery, no universal pre-decode pixel budget, and uneven upload-directory execution guards.
2. A separate post-model investigation exposed a foundational architectural defect: SnapSmack did not have one enforced definition of a publishable content unit. A photograph could be both a media asset and the post itself, while newer features used `snap_posts`. Publication state, dates, ownership, engagement identifiers, collections, and federation identity consequently developed competing meanings.

The second finding should have been identified earlier when the CMS architecture was questioned. Audit 049 maps the defect and records the canonical direction, but it is not a complete comparative architecture assessment. That work is assigned to SECAUDIT 050.

Important correction: the stable track is intentionally dormant

An earlier draft treated the old stable build as a patch-delivery failure. That conclusion was wrong.

- Sean deliberately stopped publishing to stable.
- No production SnapSmack installation runs on stable.
- The production fleet runs on the development/live track.
- A stale, unused track is therefore not an exposed fleet and is not a security finding.

The legitimate operational question is narrower: does each active site receive the live-track build completely, and can the fleet prove the deployed version and required controls? The 2026-08-17 header sweep showed uneven results and remains valid evidence of a delivery-verification problem. It does not revive the abandoned stable track as a risk.

Scope and method

The review combined:

- Static inspection of current source at 0.7.533.
- A written minimum compliance standard covering CMS controls, image handling, and delivery.
- Mechanical repository searches for every major reader and writer of `snap_images`, `snap_posts`, `snap_post_images`, `post_id`, `img_slug`, `img_status`, and `img_date`.
- A live response-header sweep across 26 named sites.
- Independent review of the corrected report by Codex.

It did not include a restored-database row audit, full live penetration test, exhaustive architecture comparison with mature CMS products, or fleet-wide post-remediation verification. Those limitations are material.

Corrected scorecard

The standard contains **24 requirements**, not 23: A1-A13 (13), B1-B10 (10), and C1 (1).

ID	Requirement	Verdict at close	Disposition
A1	Bound database parameters	MEETS	Retain regression coverage
A2	Output escaping	MEETS	Retain regression coverage
A3	CSRF on state changes	MEETS	Central POST and state-changing GET guards remain authoritative
A4	Authentication and capability checks	MEETS	Central administrator boundary retained
A5	Modern login handling	MEETS	bcrypt, session rotation, throttling, and expiring reset tokens verified in source
A6	Per-install secrets	MEETS	Random per-install secrets and self-heal path present
A7	Signed updates	MEETS	Ed25519 verification fails closed
A8	No public error disclosure	MEETS IN CURRENT SOURCE	Fixed after 0.7.536D; seven public entry points now log details and return generic errors. Release/live verification pending
A9	Direct-access guards on includes	GAP	TRANSFERRED to 050 trust-boundary inventory; direct endpoints must be distinguished from include-only files
A10	Security headers delivered live	FAILS	OPEN verification; 9/26 passed and 17/26 failed the 2026-08-17 sweep despite header code existing
A11	No dangerous functions on request data	MEETS	Offline/build shell use excluded from web request paths
A12	Protected updater paths	MEETS	Delivery caveat retained: protected files cannot carry fixes to existing installs
A13	Security-event logging	MEETS	Login, ban, and selected administrative events recorded
B1	Content-based image type checking	MEETS	Primary ingest paths inspect bytes/dimensions
B2	Image type allowlist	MEETS	Raster allowlists present

ID	Requirement	Verdict at close	Disposition
B3	No execution in upload directories	MEETS IN CURRENT SOURCE	One canonical Apache 2.4/2.2-compatible guard now covers `img_uploads`, `media_assets`, and `assets/img` from install, recovery, media management, and Maintenance repair. Release/live and nginx verification pending
B4	Filename sanitisation and traversal prevention	MEETS	Server-generated stored names and confined paths
B5	Pre-decode byte and pixel limits	FAILS	OPEN remediation; byte limits exist, universal pixel-budget enforcement does not
B6	SVG rejection or sanitisation	GAP	OPEN LOW/MEDIUM product decision for administrator-uploaded logo/favicon SVG
B7	Correct serving type and `nosniff`	MEETS	Download path and global header code present
B8	ImageMagick policy	N/A	SnapSmack uses GD
B9	EXIF/GPS policy is deliberate	MEETS	Preservation is an explicit photographer-first decision, not a defect
B10	Safe derivative generation	MEETS	Raster derivatives are re-encoded through GD
C1	Fixes reach active installs and are verified	GAP	Stable is irrelevant; active-fleet delivery and post-update verification remain the requirement

Corrected 0.7.533 tally: 16 MEETS, 4 GAPS, 3 FAILS, 1 N/A. Current-source remediation changes the A8 disposition but does not rewrite the historical score.

Finding dispositions

1. Public error disclosure - FIXED IN CURRENT SOURCE, MEDIUM

Seven public entry points forced PHP error display, and several returned raw exceptions. Current source now disables visitor error display, keeps full reporting in server logs, returns HTTP 500, and shows only a generic message. A regression test covers all seven entry points. The fix still requires a numbered release and live verification before the fleet disposition becomes VERIFIED.

2. Uneven live HSTS/CSP delivery - OPEN VERIFICATION, MEDIUM

The 2026-08-17 sweep found HSTS and CSP on 9 of 26 sites and neither header on 17. Source contains a shipping header module, but source presence is not proof of live delivery. Re-sweep the active fleet after deployment and record named-site results. The intentional dormant stable track is unrelated.

3. No universal pre-decode pixel budget - OPEN, MEDIUM

Several GD decode paths validate compressed byte size but do not reject pathological dimensions before decoding. The remediation must protect worker memory without rejecting legitimate high-resolution photography. SECAUDIT 050 must establish the resource-budget invariant; implementation should use a documented high ceiling and/or a calculation tied to available decode memory.

4. Upload-directory execution guards - FIXED IN CURRENT SOURCE, LOW

Current source now uses one extension-scoped execution guard across `img_uploads`, `media_assets`, and `assets/img`. Installer, media management, recovery, and Maintenance diagnosis/repair all use the same writer. The rule supports both Apache 2.4 and the scoped Apache 2.2 fallback, avoids `php_flag` incompatibility under PHP-FPM, and still allows legitimate images to be served. A regression test verifies the rule and directory coverage. A numbered release, live Apache verification, and equivalent nginx server configuration remain delivery work rather than source-code findings.

5. Administrator-uploaded SVG - OPEN PRODUCT DECISION, LOW/MEDIUM

Logo and favicon SVG files are stored without sanitisation and can be opened directly. Choose one explicit policy: sanitise SVG, rasterise it, reject it, or serve it through a restrictive non-executable response path.

6. Include-only direct access and community-token storage - TRANSFERRED TO 050

These are architecture and trust-boundary questions rather than isolated patch lines. Audit 050 must distinguish executable endpoints from libraries and document which credentials must be recoverable versus one-way hashed.

Post-model finding

What failed

SnapSmack had no database-enforced content invariant. Legacy solo publishing treated `snap_images` as the publishable object; newer group and longform publishing treated `snap_posts` as the object. Both models survived simultaneously.

Consequences include:

- Two publication-status and publication-time authorities.
- Two image-to-post ownership representations.
- Engagement columns named `post_id` that can contain image IDs.
- Collection rows labelled `post` while pointing at images.
- Federation identities split between `/ap/note/i/` and `/ap/note/p/`.
- Readers and writers that can omit, duplicate, or misidentify the same photograph.

This was an obvious architectural limitation in hindsight and should have been raised during earlier "anything structurally wrong?" reviews. Component-level correctness was repeatedly assessed without first defining system-level invariants.

What 0.7.534D-0.7.536D added

- `modcheck.php` provides a read-only, per-site inventory of old-shape content and ambiguous references.
- Maintenance gained a transactional, repeat-safe CONVERT PHOTOS TO POSTS action.
- The repair creates one `snap_posts` row and one cover pivot per bare image, attaches the image, preserves the human permalink and lifecycle metadata, and rolls the whole run back on any failure.
- Existing likes, reactions, and community comments deliberately remain unchanged during the readers-first transition.

What remains

The repair button is not architectural closure. Readers, writers, scheduling, imports, federation, engagement, collections, deletion, schema constraints, migration receipts, and fleet verification still require coordinated convergence. Historical ActivityPub image identity is intentionally not preserved because SnapSmack remains alpha; authored photographs and human-facing permalinks must be preserved.

Why Audit 050 exists

Audit 049 was primarily a security/image-handling review that discovered and mapped an architecture failure. It did not compare the complete CMS design against mature implementations or prove system invariants across every lifecycle.

SECAUDIT 050 is therefore the dedicated CMS architecture assessment. It must evaluate the whole product against documented patterns from established CMS, publishing, media-library, and federated systems; produce an invariant catalogue and domain model; test the schema and transaction boundaries; trace every reader and writer; and issue explicit PASS, DEBT, or BLOCKER findings with evidence.

Closure statement

SECAUDIT 049 is closed as a corrected and independently reviewed assessment record. The dormant stable-track claim is withdrawn, the scorecard is corrected, the architecture limitation is explicitly acknowledged, and every unresolved finding has a named disposition rather than being erased.

This closure does **not** certify the CMS architecture, close the post-model migration, or erase the remaining open and transferred security/hardening items. Those claims require evidence from their remediation releases and the SECAUDIT 050 architecture assessment.