SECAUDIT 046 - Process Correction, 524-to-525 Delta Review, and Invalidated Simulation

SECURITY AUDIT / PUBLIC REMEDIATION RECORD

- **Date:** 2026-08-14
- **Requested work:** Live penetration test of the current SnapSmack release
- **Work initially delivered:** Simulated multi-agent source review against stale code
- **Valid work retained:** Line-by-line security review of the actual 0.7.524D-to-0.7.525D release delta
- **Valid delta result:** No findings
- **Publication status:** Public process and audit record; superseded by SECAUDIT 047 and the genuine live test recorded in SECAUDIT 048

Executive summary

SECAUDIT 046 records both a valid security result and a significant process mistake.

Claude was asked to perform a live penetration test. Instead, the work product was a simulated, multi-agent source review. It did not demonstrate attacks against a running current installation, and it was performed against a branch on the 0.7.124 lineage - roughly 400 versions behind the 0.7.525D release then under review. Calling that work a live penetration test would have been inaccurate. Treating its findings as findings against the shipping product would also have been inaccurate.

The mistake was identified. The stale-branch conclusions were invalidated rather than quietly relabelled or published as current vulnerabilities. The useful portion of the work - a separate line-by-line review of the real 0.7.524D-to-0.7.525D release delta - remained valid and produced no findings. The security backlog suggested by the stale simulation was then rechecked against current code in SECAUDIT 047. A real running installation was subsequently attacked in SECAUDIT 048.

This report is published because an honest security ledger should record failed audit processes as well as vulnerabilities. A missing number would conceal the mistake; reusing the number for unrelated work would rewrite the record.

What was requested

The requested task was a live penetration test of the current SnapSmack release. A live penetration test requires a running target and evidence tied to that target: the tested build, threat positions, accounts and roles, requests made, responses observed, controlled exploit attempts, and verification after remediation.

What actually happened

Claude orchestrated an eight-dimension review covering authentication and sessions, updater and signing behaviour, public-input injection, uploads and code execution, multisite key custody, authorization, database and cryptographic controls, and skins and companion tools.

That exercise was a simulation and source-level review, not a live penetration test. The resulting record did not establish a current running target or provide live request-and-response evidence. More seriously, the reviewed branch, `claude/audit-or-security-test-41d53b`, belonged to the 0.7.124 lineage and had diverged during the recovered-branch incident. The shipping development line was approximately 400 versions newer.

The simulation could still generate useful hypotheses, but it could not establish vulnerabilities in 0.7.525D. Some alleged conditions were already corrected in current code; other files and control paths had changed substantially. Patches

prepared against the stale branch could not safely be released because applying them risked reverting hundreds of versions of later work.

Why this mattered

The process failure created three risks:

- **False assurance:** A simulated review could be mistaken for proof that a running installation had resisted attack.
- **False findings:** Defects found in stale code could be presented as vulnerabilities in the current product even when already fixed or structurally irrelevant.
- **Unsafe remediation:** Applying stale-branch patches could restore old code and introduce new regressions while appearing to improve security.

The correct response was to invalidate the unsupported claims, preserve only independently valid work, and rerun the security work against the correct target.

Valid work retained: the 524-to-525 release delta

Separately from the invalidated simulation, the complete shipping delta between `v0.7.524D` and `v0.7.525D` was reviewed. The review covered 649 changed lines across 21 code files after excluding marketing pages, help content, and tests. Each non-cosmetic change was traced from input to effect.

Result: no findings

Approximately 95 percent of the delta renamed the public label SMACKVERSE to Fediverse. The internal state key remained unchanged, so the display update did not alter routing, authorization, stored data, or protocol behaviour.

The substantive changes were also cleared:

Change	Assessment
Schema self-heal for federation and sensitivity columns	Column definitions were fixed literals, not user input. The calls remained behind the relevant authenticated write gates and were correctly placed outside transactions because schema alteration can implicitly commit.
Administrative `unbury_batch` maintenance action	The query used fixed matching patterns, the action was covered by the administrative CSRF gate, and the reflected count was integer-cast before output. The operation was idempotent.
New gram-post ordering changed from buried to normal	Feed-ordering correction only; no security boundary changed.

The valid conclusion was narrow: the 0.7.524D-to-0.7.525D release delta introduced no identified security defects. It was not proof that the entire application had no vulnerabilities.

Invalidated simulation output

The stale review raised hypotheses involving multisite trust, login protection, two-factor guessing resistance, upload handling, per-site secrets, role enforcement, CSRF protection, updater enforcement, skin signatures, and installation cleanup.

None of those hypotheses was accepted as a finding against the live release on the strength of SECAUDIT 046. They became a re-verification backlog only. The distinction matters: a useful question is not the same thing as a demonstrated vulnerability.

Corrective action

The following corrective sequence was used:

1. The stale-branch simulation was labelled invalid for claims about current or live code.
2. Its hypotheses were retained only as a checklist for verification against the current branch.
3. SECAUDIT 047 performed the broad security review against the actual 0.7.525D development code and documented the findings and remediation there.
4. SECAUDIT 048 attacked a disposable running installation as an unauthenticated visitor and as a compromised content-only editor. That report contains the live evidence and the flaws found only when the software was exercised as a running system.
5. Future reports must identify whether their method is source review, simulation, release-delta review, or live penetration testing. Those terms are not interchangeable.

Process controls adopted

For security work to be described as a live penetration test, the report must now record:

- the exact deployed version and target environment;
- confirmation that the target is isolated from production data and networks where appropriate;
- the attacker positions and account roles used;
- the live requests, responses, headers, status codes, and controlled payload outcomes needed to support each finding;
- remediation deployment details and a live retest; and
- limitations and untested surfaces.

Before any source audit begins, the checked-out branch and version must be compared with the intended target. If they do not match, the audit stops. Findings from stale code may be retained as research leads, but they may not be reported as current findings without reproduction against current code.

Final disposition

- **SECAUDIT 046 delta review:** Valid; no findings in the 524-to-525 release delta.
- **SECAUDIT 046 simulated broad review:** Invalid as a live or current-code audit; retained only as historical process evidence and a re-verification checklist.
- **SECAUDIT 047:** Correct broad source-level audit against current code.
- **SECAUDIT 048:** Actual live penetration test against a disposable running installation.

The error is part of the public record. The numbering is preserved, the unsupported claims remain invalidated, and the later reports show how the work was rerun correctly.