Security Audit — 0.7.209 "Courtesy Flush"

Date: 2026-06-05 **Auditor:** Cowork session (Claude), for Sean **Scope:** Only the changes in this release — the SmackBack false-positive fix (Bug A + Bug B) and its hardening. Pre-existing code is reviewed only where this change touches or moves its trust boundary.

1. What changed

File	Change
`smack-back.php`	Re-init action now calls `smackback_resolve_breach('reinit')` when `$ok && $is_breach`; clean full-verify branch calls `resolve_breach('manual')` when `$is_breach`. (Bug B — a breach was previously only clearable via RESTORE, which reverts code.)
`smack-central/sc-release.php`	`sc_build_release_zip()` now builds `smackback-manifest.json` (hash/size/eof_signature per monitored php/css/js, from the exact bytes written to the zip) and adds it before `$dst->close()`. (Bug A — deployed zips carried no manifest, so the post-update re-baseline always no-op'd.)
`smack-update.php`	If `smackback_init_manifest()` finds no manifest, fall back to `smackback_init_from_disk()`. A stale breach is auto-cleared **only** on the signed-manifest path (`$smack_signed`).
`_spec/smackback-netalert-fixes-spec.md`	§1 reconciled to the traced reality. Doc only — no runtime effect.

2. Threat model for this change

The relevant assets are (a) the integrity baseline in `snap_file_manifest`, and (b) the breach state that gates admin access in lockout mode. The relevant actors are an unauthenticated web visitor, and an attacker who has compromised the GitHub repo or the release-delivery path. The Ed25519 package signature (verified before extract) is the trust root for all update content and is unchanged by this release.

3. Surfaces reviewed and findings

F1 — Integrity baseline now cryptographically anchored on deployed releases. [Improvement] The new `smackback-manifest.json` is added to the zip *before* the zip is hashed and signed (`sc-release.php` Step 4 signs `sha256(zip)`), so the manifest is inside the signed payload. Before this release, SC-packaged zips shipped no manifest at all and the update baseline was never refreshed — a control failure, now closed.

F2 — Breach auto-clear restricted to the signed path. [Hardening] The update flow can re-clear a stale breach. This is gated to `$smack_signed` — true only when the baseline came from the in-zip signed manifest. The disk-fallback path (`init_from_disk`) deliberately does **not** auto-clear; a breach there is left for admin review on `smack-back.php`. This prevents a manifest-less but validly-signed (legacy) package from silently blessing tampered-on-disk files via the breach reset.

F3 — `init_from_disk` trust basis. [Low — accepted] The fallback and the manual re-init both trust whatever is on disk. This is only reachable *after* Ed25519 package verification (fallback) or by an authenticated admin (re-init), i.e. the same trust basis as a fresh install, which has always used `init_from_disk`. The re-init UI explicitly warns not to use it while a breach is active. Accepted, no change.

F4 — No new injection / path traversal. [Info] Manifest keys are the zip-relative paths already guarded in the copy loop (entries containing `..` or a leading `/` are rejected, `sc-release.php` L322). File content is

hashed in memory from `ZipArchive::getFromIndex()`; no user input reaches the manifest. JSON is built with `json_encode`, not string concatenation.

F5 — Breach-clear handlers are auth- and CSRF-gated, and audit-logged. [**Info**] Both modified handlers are POST-only, behind `core/auth-smack.php` (admin auth), which calls `csrf_check()` globally (HTTP 403 on token mismatch). Every clear writes a row to `snap_smackback_log` via `smackback_resolve_breach()`, preserving the incident trail. No new endpoint, parameter, or auth path was introduced.

4. Follow-ups (out of scope for this push — flagged, not fixed here)

S1 — `smack-central/sc-skins.php` has the same missing-manifest gap. [**Med**] The SC skin packager downloads the GitHub archive and builds + signs per-skin zips but writes no `smackback-manifest.json`. Effect: SC-distributed skins are not integrity-monitored (`skin-registry.php` L302 `init_skin_manifest` gets no manifest → no skin baseline rows). This is a protection gap, not a false positive. Fix is the same pattern as `sc-release.php`, per skin — but skins are forkable deliverables, so it needs its own session with post-edit structure verification. Do not bulk-edit skins.

S2 — `smackback-manifest.json` exposed at web root. [**Low**] Post-0.7.209 the manifest lands at each install's web root fleet-wide. It discloses the monitored file inventory and SHA-256 hashes — both derivable from the public release, so low sensitivity. The runtime reads the manifest from the update **zip**, not from disk, so denying web access has no functional cost. Recommend an `.htaccess` deny rule in `core/htaccess-template` (rebuilt on update).

5. Verdict

The release closes a real integrity-control failure and is net-positive for security. No new web-facing attack surface; the one moved surface (breach auto-clear) is gated to the cryptographically-signed path. Ship. Track S1 and S2 as separate work.