Security Audit — 028 "Package Dev-File Leak + SMACKBACK Blind Spot"

Date: 2026-06-20 **Auditor:** Cowork session (Claude, Opus 4.8), for Sean **Severity:** Medium (info-leak + integrity-noise / availability); High *conditional* on Finding C **Status:** Packager remediated this session (allowlist). Cleanup of existing installs and the integrity-model items remain (see §6).

1. Summary

Two structural packaging/monitoring weaknesses, surfaced while auditing live spokes after the 0.7.274 strict-detection rollout:

1. The release packagers use a hand-maintained **denylist**, so dev/doc files that nobody remembered to exclude (`.githooks/`, `BRANCHING.md`, `PHASE1-GIT-COMMANDS.md`, two `.docx` specs, the whole `outputs/` scratch dir) shipped into every install root.
2. `smack-central/` — the hub/release control plane — is **excluded from SMACKBACK monitoring on every install**, which is correct on the hub but a blind spot everywhere else, including a place to hide a dropped file.

Same denylist fragility previously leaked the AURORA skin fleet-wide in 0.7.263. The fix is to invert the packager to a **default-deny allowlist**.

2. Finding A — dev/doc files ship into installs (Medium)

The packagers (`smack-central/sc-release.php`, active; `tools/_build/build-release.php`, local/future) excluded files by enumerated name/dir. Three `.md` files were named and **no `.docx` rule existed**, so any other doc rode along. Confirmed shipping into install root:

- `BRANCHING.md`, `PHASE1-GIT-COMMANDS.md`
- `snapsmack-tools-reference.docx`, `the-grid-spec-for-review.docx`
- the entire `outputs/` dev-scratch directory (e.g. `outputs/coauthorship-worked-example-DRAFT.md`)
- `.githooks/pre-commit` (tracked; no `.gitattributes export-ignore` strips it from the GitHub tarball the packager builds from), `package-lock.json`

Impact: public exposure of internal repo structure and working docs; and — see Finding C — these untracked-by-baseline files trip SMACKBACK as UNEXPECTED, contributing to fleet lockouts. No secrets were in the leaked files themselves.

Remediated — a default-deny **allowlist gate** added to both packagers. A web install now ships only runtime types (`php css js txt png jpg jpeg gif svg ico webp`), with three runtime non-allowlist files force-included (`database/schema/snapsmack_canonical.sql`, `protected_paths.json`, `core/htaccess-template`). `outputs/` added to the dir excludes. `.php` cruft (`backfill-*`, `release-pubkey.php`, `install.php`/`setup.php`) is still removed by the existing name/dir excludes — the allowlist only backstops the non-`.php` leaks.

3. Finding B — SMACKBACK blind to `smack-central/` on spokes (Medium)

`smackback_should_monitor()` lists `smack-central/` in its excluded dirs. Correct on the hub (where it lives and changes legitimately); on a spoke it means a `smack-central/` folder — which should not exist at all — is invisible to integrity monitoring, and is a ready hiding spot for a planted file. A leftover `smack-central/` was in fact found on a spoke (lightafterdark.ca), reduced to a single inert `sc-version.php` (stale `0.7.265D`, no secrets). It did **not** arrive via any package (both packagers exclude `smack-central/`); it was an out-of-band upload remnant.

Open fix (integrity model): make the `smack-central/` exclusion **role-aware** (legit only where this install IS the hub), and treat the *presence* of hub-only paths on a spoke as a breach rather than ignoring them.

4. Finding C — strict detection amplifies the leak into lockouts (Medium, availability)

0.7.274's strict unknown-file detection flags any monitored on-disk file absent from the signed baseline. The Finding-A files are on disk but never in the baseline -> UNEXPECTED -> contributes to LOCKOUT on spokes that updated to 274. The allowlist fix removes them from future installs at source; existing installs need a one-time cleanup (§6).

5. Conditional finding — historical hub-secret exposure (High, UNCONFIRMED)

The real (gitignored) `smack-central/sc-config.php` defines `SMACK_RELEASE_PRIVKEY` (the Ed25519 release **signing** secret), SC/STE DB passwords, the forum mod key, and a GitHub PAT. The signing key is the linchpin of "a compromised hub cannot forge a signed package." On the spoke inspected, only `sc-version.php` was present — **not** `sc-config.php` — so this scenario is not realized there. **Action:** confirm via FTP/access logs whether the *full* `smack-central/` (with `sc-config.php`) was ever on a public spoke earlier and later trimmed. If so, treat the signing key as potentially compromised: rotate the Ed25519 keypair and redeploy `core/release-pubkey.php` fleet-wide; rotate the SC/STE DB passwords, forum mod key, and GitHub PAT.

6. Residuals / follow-ups

- **Clean existing installs:** remove `outputs/`, root `.md`/`.docx`, and any leftover `smack-central/` from spokes (site-by-site, operator). `outputs/` confirmed safe to delete — zero runtime references.
- **`.gitattributes export-ignore`** would strip dev files at the git-archive level too, but must be cross-checked against `always_include` (canonical SQL must still reach the tarball); deferred as a belt-and-suspenders layer to the allowlist.
- **Role-aware `smack-central/` monitoring + presence-as-breach** (Finding B fix) — part of the open SMACKBACK integrity model (`_continuity/smackback-integrity-model-2026-06-20.md`).
- **Lint before release:** `php -l smack-central/sc-release.php tools/_build/build-release.php` on the build box (sandbox PHP is unavailable / mount may be stale).