SnapSmack Security Audit
022 — Canonical Schema Remote Fetch Hardening

Date: 2026-06-07

Auditor: Cowork session (Claude), for Sean

Release: 0.7.214 "Flush Protocol"

Scope: updater_canonical_diff() — remote fetch gate and signature enforcement

Prior audit: 021A (2026-06-05) — hub/spoke attack surface addendum

1. Background

0.7.213 wired the canonical schema URL from latest.json through the update pipeline, enabling updater_canonical_diff() to fetch the authoritative snapsmack_canonical.sql directly from snapsmack.ca at update time. The SC packager signs the file with the same Ed25519 private key used for release zips. However, the signature verification path contained a gap: if the sodium PHP extension was absent, the check was silently skipped and the remote SQL was applied without verification.

Sean identified this as a new attack surface introduced in 0.7.213. This audit covers the gap analysis and the hardening applied in 0.7.214.

2. What the attack surface was

The canonical SQL is fetched over HTTPS from snapsmack.ca. An attacker with a MITM position or write access to snapsmack.ca could serve a malicious snapsmack_canonical.sql. The apply function (updater_apply_canonical_diff) would then execute the attacker-supplied SQL against the target database using the already-authenticated local PDO connection.

Blast radius is structurally limited: updater_apply_canonical_diff only executes CREATE TABLE IF NOT EXISTS and ALTER TABLE ... ADD COLUMN. It cannot DROP tables, DELETE or UPDATE rows, or modify existing column definitions. An attacker could add backdoor tables or columns but could not destroy data.

The gap was in core/updater.php, updater_canonical_diff(), lines 1343-1362 (pre-patch). The relevant branch:

```
if ($canonical_sig_url !== '') {
    $sig_hex = ...
    if ($sig_hex !== '' &&
function_exists('sodium_crypto_sign_verify_detached')) {
        // verify ...
    }
    // ← if sodium absent, falls through here with $fetched still set
```

```
}
```

If function_exists() returned false (sodium absent), the inner block was skipped entirely and $fetched remained set. The subsequent check if ($fetched !== false) then accepted the remote SQL without any verification.

3. What changed

File	Change
core/updater.php	updater_canonical_diff() — remote fetch gate rewritten with explicit three-condition check
core/constants.php	Version bump to 0.7.214
CHANGELOG.md	Security entry for 0.7.214

The new gate requires all three conditions before accepting remote SQL:

```
(a) SNAPSMACK_SIGNING_ENFORCED is true — real Ed25519 pubkey installed, not the
all-zeros placeholder
(b) canonical_schema_sig URL is present in the manifest (latest.json /
latest-dev.json)
(c) function_exists('sodium_crypto_sign_verify_detached') — sodium extension
available
```

Any condition unmet: remote copy is rejected, disk fallback used, specific error logged. When SNAPSMACK_SIGNING_ENFORCED is false (placeholder pubkey — dev installs, fresh installs before pubkey is configured), the remote fetch is skipped entirely: no point fetching content that cannot be authenticated.

4. Findings

F1 — Silent sodium-skip closed. [Fixed — 0.7.214]

The pre-patch code silently skipped signature verification when sodium was absent and accepted the remote SQL. The patch restructures the block so every verification failure mode (sodium absent, empty sig response, SodiumException, sig mismatch) explicitly sets $accepted = false. Remote SQL is only used if $accepted is explicitly set to true after a successful verify call. No silent pass-through path exists.

F2 — SIGNING_ENFORCED=false path hardened. [Fixed — 0.7.214]

Previously, an install with the placeholder pubkey (SIGNING_ENFORCED=false) would still attempt a remote fetch if a canonical_schema_url was present — it would then attempt sig verification with the all-zeros key, fail, and fall back to disk. This round-trip was unnecessary and left a window (however small) where the fetched content existed in memory before rejection. The patch gates the

entire remote fetch on SNAPSMACK_SIGNING_ENFORCED, so placeholder-key installs never fetch the remote file at all.

F3 — Sodium is a hard installer requirement. [Info — no action needed]

The hardening includes a fallback for sodium-absent installs (reject remote, use disk). On examination, install.php already checks function_exists('sodium_crypto_sign_verify_detached') and sets $all_pass = false if absent — same gate as PDO MySQL and GD. An install cannot complete without sodium. Therefore the sodium-absent fallback path in updater_canonical_diff is defense-in-depth (server migration, extension loss) rather than a realistic runtime condition on any legitimately installed site.

F4 — SC must be at 0.7.213+ before a build is triggered. [Low — operational]

The sc-release.php that publishes and signs snapsmack_canonical.sql was introduced in 0.7.213. If a build is triggered on an SC instance still running 0.7.212 or earlier, canonical_schema_sig will be absent from the manifest. Installs will fall back to the on-disk copy — correct and safe behavior — but the remote verification path will never be exercised. Update SC to 0.7.213+ before triggering the first post-0.7.213 build.

F5 — Disk fallback trust basis unchanged. [Info]

The disk copy at database/schema/snapsmack_canonical.sql is extracted from the release zip, which is Ed25519-verified before extraction (smack-update.php stage_verify). The disk copy therefore carries the same trust level as the zip itself. No new trust assumption is introduced by falling back to disk.

5. Threat model summary

Threat	Pre-0.7.214	Post-0.7.214
MITM on canonical SQL fetch (sodium present)	Sig mismatch → disk fallback [Protected]	Sig mismatch → disk fallback [Protected]
MITM on canonical SQL fetch (sodium absent)	Sig check skipped → remote applied unverified [GAP]	Remote rejected → disk fallback [Fixed]
Compromised snapsmack.ca (malicious canonical SQL)	Sig would fail (attacker lacks private key) [Protected]	Sig would fail [Protected]
Placeholder pubkey install	Remote fetched, sig fails (wrong key), disk used	Remote never fetched [Hardened]

6. Follow-ups

None. No residual open items from this audit. F4 is an operational note, not a code defect.

7. Verdict

The gap was real: a sodium-absent PHP installation (not normally possible on a legitimate SnapSmack install, but theoretically achievable via server migration or misconfiguration) could have had a malicious canonical schema applied without signature verification. The fix closes the gap cleanly. The hardening is net-positive with no usability cost — the disk fallback is always authoritative and always present. Ship 0.7.214.