SnapSmack Security Audit

Third Review — Alpha 0.7.27 · April 26, 2026

Executive Summary

This third audit was conducted during the 0.7.27 development session, covering all changes made between Alpha 0.7.26 and 0.7.27. The review focused on Cloudflare Tunnel HTTPS detection, installer hardening, release packager changes, and code integrity controls introduced in this release cycle.

This audit identified 1 medium finding (credential overwrite in installer — fixed), 2 low findings (mobile skin path confusion, code integrity), and confirmed 6 changes as carrying no new security risk. The deferred HIGH finding (CSRF) from Audit 2 remains open. No authentication bypass, SQL injection, or privilege escalation vulnerabilities were found.

#	File	Severity	Finding	Status
1	install.php	MEDIUM	Installer admin INSERT could overwrite credentials	FIXED 0.7.27
2	index.php	LOW	Empty SNAPSMACK_MOBILE_SKIN causes path confusion	FIXED 0.7.27
3	Nine PHP files	LOW	Null bytes injected by edit tool — code integrity	FIXED 0.7.27
4	snap_is_https()	INFO	HTTPS detection via server vars — correct implementation	PASS
5	setup.php	INFO	File-by-file zip extraction — no new attack surface	PASS
6	install.php (passwords)	INFO	Show/hide toggle — client-side only, server validation intact	PASS
7	.htaccess	INFO	HTTPS redirect — dual-condition, no open redirect	PASS
8	sc-release.php	INFO	Exclude list tightened — reduces package attack surface	PASS
9	Multiple AJAX endpoints	HIGH	CSRF tokens missing (carried from Audit 2)	DEFERRED

Findings

1. Installer Admin User INSERT — Credential Overwrite Risk

MEDIUM · install.php line 756 · FIXED 0.7.27

The installer's admin account creation used `INSERT INTO ... ON DUPLICATE KEY UPDATE`, which updates the email, password hash, and role of an existing user matching the submitted username. On a partially installed server where install.php was still present, a third party who knew the admin username could overwrite the admin password by submitting the form with their own credentials.

Vulnerable code:

```
$stmt = $pdo->prepare("INSERT INTO `{$prefix}users` (...) VALUES (?, ?, ?, 'admin',
'midnight-lime') ON DUPLICATE KEY UPDATE email = VALUES(email), password_hash =
VALUES(password_hash), user_role = 'admin'");
```

Note: The installer does have a safety lock that checks for an existing installation, but this lock depends on snap_settings having rows — a partially installed server (schema created, settings not yet seeded) could bypass it.

Fix applied — changed to INSERT IGNORE:

```
$stmt = $pdo->prepare("INSERT IGNORE INTO `{$prefix}users` (...) VALUES (?, ?, ?, 'admin',
'midnight-lime'");
```

INSERT IGNORE silently skips if the username already exists. No credentials are changed. The admin can log in with their original password and change it from within the admin panel. This removes the credential-overwrite surface entirely.

2. Empty SNAPSMACK_MOBILE_SKIN Causes Path Confusion

LOW · index.php line 46 · FIXED 0.7.27

When SNAPSMACK_MOBILE_SKIN is an empty string (the previous default), the mobile skin check called `is_dir(__DIR__ . '/skins/')` — evaluating to true since the skins directory always exists. This set `$active_skin = ''`, causing include paths like `skins//skin-meta.php` which produced a fatal warning and 404 on mobile devices.

While not directly exploitable, the empty active_skin could in edge cases cause unintended file inclusion depending on server configuration and PHP include_path settings.

Vulnerable code:

```
if (snapsmack_is_mobile() && is_dir(__DIR__ . '/skins/' . SNAPSMACK_MOBILE_SKIN)) {
```

Fix applied:

```
if (snapsmack_is_mobile() && SNAPSMACK_MOBILE_SKIN !== '' && is_dir(__DIR__ . '/skins/' .
SNAPSMACK_MOBILE_SKIN)) {
```

Additionally, SNAPSMACK_MOBILE_SKIN now defaults to 'photogram' in both the repo's constants.php and the installer's generated constants.php template, since Photogram ships in every base release package.

3. Null Bytes in PHP Files — Code Integrity

LOW · Nine files · FIXED 0.7.27

The Cowork VM edit tool was found to silently inject null bytes (0x00) into PHP files on save. PHP treats null bytes as string terminators in some contexts, causing garbled output, broken page rendering, and in some cases fatal parse errors. The following files were found to contain null bytes and were stripped:

File	Null bytes removed
install.php	1,738
smack-help.php	1,316
smack-edit.php	1,160
smack-post-carousel.php	942
smack-edit-carousel.php	865
smack-central/sc-release.php	193
core/meta.php	43
core/auth.php	39
core/community-session.php	39

Two files (sc-release.php and setup.php) were additionally found to be truncated — missing their closing script blocks and PHP includes. The truncation caused changelog JavaScript to never execute and the setup deployer to render without its Install button. Both were restored from git history.

Mitigation — a pre-commit hook (.githooks/pre-commit) was added that scans all staged PHP files for null bytes and runs php -l syntax validation before allowing any commit. Activate with: `git config core.hooksPath .githooks`. The hook blocks the commit and prints the strip command if dirty files are found.

Items Reviewed — No New Risk Found

snap_is_https() Helper

INFO

Added to core/constants.php and inlined in install.php for fresh-install availability. Checks $_SERVER['HTTPS'], HTTP_X_FORWARDED_PROTO, and HTTP_X_FORWARDED_SSL in correct order. No user input involved. Correctly returns bool — no output, no headers, no side effects.

setup.php File-by-File Extraction

INFO

Switched from ZipArchive::extractTo() to a per-file loop to prevent silent failures when directory entries don't precede file entries in the zip. Path traversal check (strpos against realpath) runs before any extraction. setup.php itself is skipped in the loop — PHP cannot overwrite a running script. Added mkdir() before file_put_contents() — uses only server-local paths derived from validated zip entries.

Password Show/Hide Toggle

INFO

Client-side JavaScript only. Toggles input type between 'password' and 'text'. Server-side validation (minimum length, mismatch check) runs independently of client-side state and cannot be bypassed by manipulating the toggle. The submit button disable is a UX convenience — the server rejects mismatched passwords regardless.

.htaccess HTTPS Redirect

INFO

Dual-condition redirect: HTTP:X-Forwarded-Proto != https AND HTTPS != on. Rewrites to %{HTTP_HOST} which is the server's own host — no open redirect possible. R=301,L flags are correct. Works on both direct HTTPS and Cloudflare Tunnel proxied installs without redirect loops.

Release Packager Exclude List

INFO

secaudits/, migrations/, database/, data/, .well-known/ and maintenance scripts added to always_exclude. Skin screenshots excluded from all packages — gallery previews are served from sc-assets, not the install directory. These changes reduce the attack surface of the install package; no new exposure introduced.

Deferred Finding — Carried from Audit 2

CSRF Tokens Missing on State-Changing Endpoints

HIGH · Multiple files · DEFERRED

State-changing POST endpoints (liking posts, setting reactions, posting community comments) continue to accept requests without validating a CSRF token. SameSite=Lax cookie attribute provides partial mitigation for cross-site requests initiated by top-level navigation, but does not cover all CSRF vectors.

This finding was intentionally deferred due to implementation scope. It remains the highest-priority open item for a future release.

Overall Assessment

The 0.7.27 release cycle focused on installer reliability and Cloudflare Tunnel compatibility rather than new feature development. From a security perspective, the changes are neutral to positive: one medium credential-overwrite risk in the installer was closed, a path confusion bug on mobile was fixed, and a systematic null byte injection problem was identified, stripped from nine files, and prevented from recurring via a pre-commit hook.

The codebase continues to use parameterised queries consistently throughout, session management is sound, Ed25519 signature verification on release packages is in place, and security response headers are sent on every request. No authentication bypass, SQL injection, or privilege escalation vulnerabilities were identified in this review.

CSRF remains the most significant open item. The pre-commit null byte hook should be activated on all developer machines (git config core.hooksPath .githooks) to prevent code integrity issues from recurring.

Finding	File	Severity	Status
Installer credential overwrite	install.php	MEDIUM	FIXED
Mobile skin path confusion	index.php	LOW	FIXED
Null bytes in PHP files (9 files)	Multiple	LOW	FIXED
CSRF missing on AJAX endpoints	Multiple	HIGH	DEFERRED