SNAPSMACK SECURITY AUDIT

Remediation Addendum — #024A · Caption XSS Fix + UZ-03 Status Correction

---

Date: 2026-06-12 · **Supersedes status in:** Audit #024 (same day)
Components: core 0.7.254 → fix targets 0.7.255 · Unzucker importer
Scope: Remediation of the one actionable finding from #024 (**UZ-11**) plus a correction to the **UZ-03** status. Files: `core/snap-tags.php`; verification read of `tools/unzucker/ig_parser.py`.
Auditor: Claude (Opus 4.8), Cowork session — for Sean McCormick

SUMMARY

Following audit #024, the single actionable finding — **UZ-11**, a MEDIUM stored-XSS vector in caption rendering — has been **remediated at the sink**. A re-read of current source also shows **UZ-03** (export-root path traversal) is **already mitigated** in `ig_parser.py`; #024 listed it as open in error, and that record is corrected here. No new findings. The deferred backlog items (UZ-01 SFTP host-key, UZ-04 admin CSRF) are unchanged and still recommended for dedicated work.

EOF MARKER / TRUNCATION CHECK

File	EOF Marker	Status
`core/snap-tags.php`	`// ===== SNAPSMACK EOF =====`	✓ OK
`tools/unzucker/ig_parser.py`	`# ===== SNAPSMACK EOF =====`	✓ OK

Verified via the Read tool. `php -l` / `py_compile` remain the operator's gate (no PHP in the build sandbox).

REMEDIATION

UZ-11 — Caption stored-XSS (strip_tags attribute passthrough) FIXED

File: `core/snap-tags.php — snap_render_caption_html()`

The function previously ran `strip_tags($text, '<a>…')`, which keeps whitelisted tags *with their attributes* — so `<a href="javascript:…">` and event-handler attributes such as `onmouseover` survived into the page. Rewritten to escape all markup first, then re-introduce only the formatting the function itself generates:

```
// BEFORE (unsafe)
$allowed = '<p><br><strong><em><u><a><ul><ol><li><blockquote>';
$safe = strip_tags($text, $allowed);   // keeps attributes on allowed tags

// AFTER (safe)
```

```
$safe = htmlspecialchars($text, ENT_QUOTES, 'UTF-8');  // neutralise ALL markup
$safe = nl2br($safe);                                  // line breaks we generate
// ... existing hashtag linkifier below builds its own escaped <a> anchors
```

The hashtag linkifier (unchanged) already emits `htmlspecialchars`-escaped display text and a `rawurlencode`'d href, so it adds no new sink. Verified the escape entities (`&#039;` etc.) cannot match the hashtag pattern, so escaping does not corrupt link detection.

Blast radius — fixed for all three callers at once (the sink is shared): `skins/the-grid/layout.php` L247, `skins/photogram/feed.php` L246, `skins/photogram/layout.php` L265. All render Instagram-imported captions.

Intended behaviour change: captions no longer render *embedded* HTML formatting (`<strong>`, `<a>`, lists) — they display as literal text. This is correct for the callers (IG captions are plain text). If formatted captions are ever desired, replace escaping with a DOM-based sanitizer that keeps a tag whitelist but strips all attributes (and validates `<a href>` to http/https/mailto). Not needed today.

Deploy: core file — requires a core release. 0.7.254 is live, so this rides the next bump (0.7.255), versioned/tagged/released by the operator.

UZ-03 — ig_parser path traversal (status correction)

ALREADY MITIGATED

File: `tools/unzucker/ig_parser.py — parse(), L146–156`

Audit #024 (and #023) listed UZ-03 as open. Current source already constrains every resolved media path to the export root and skips anything that escapes it:

```
abs_path = os.path.normpath(os.path.join(export_root, uri))
if not abs_path.startswith(os.path.normpath(export_root) + os.sep) \
        and abs_path != os.path.normpath(export_root):
    result.errors.append(f"... URI escapes export root (possible path traversal) —
skipped: {uri}")
    missing_images += 1
    continue
```

This is exactly the containment assertion the audits recommended (absolute-URI escape and `../` traversal both rejected). The "open" status was stale; no change required. Record corrected.

STATUS AFTER ADDENDUM

ID	From #024	Now	Note
UZ-11	MEDIUM / OPEN	FIXED	Sink hardened; rides core 0.7.255
UZ-03	LOW–MED / OPEN	ALREADY MITIGATED	Guard already in source; #024 status was stale
UZ-01	MEDIUM / OPEN	OPEN	SFTP AutoAddPolicy — needs UX call (known_hosts)

UZ-04	LOW–MED / OPEN	OPEN	Admin CSRF — own focused pass; do not keep deferring
UZ-08	INFO / unverified	VERIFIED CLEAN	snap_sync_tags() fully parameterised — no SQLi

> **Verdict.** UZ-11 — the only actionable finding from #024 — is closed in source and awaits the next core release (0.7.255) to go live. UZ-03 was already mitigated; UZ-08 is confirmed clean. The remaining open items, UZ-01 (SFTP host-key verification) and UZ-04 (admin CSRF), are unchanged and should each get dedicated, deliberate work rather than another deferral.
