SNAPSMACK_EOF_HEADER Last non-empty line of this file MUST be the canonical EOF marker for this file type: an HTML comment containing five equals, space, the literal string 'SNAPSMACK EOF', space, five equals. Missing or different = truncated/corrupted. Restore before saving.

SECAUDIT 041 - `download_url` scheme trust, and the IndieWeb identity surface

SECURITY AUDIT / PUBLIC REMEDIATION RECORD

Date: 2026-08-07 Scope: `core/indieweb.php` (shipped in 0.7.506D) and the download-link surface it sits beside - `core/social-dock.php`, `core/download-overlay.php`, `core/threeacross-api.php`, `smack-backfill.php` Status: **all findings closed in this pass**

Why this audit happened

The IndieWeb semantics shipped in 0.7.506D publish owner identity as machine- readable markup (`rel=me`, `h-card`, `h-entry`). Anything that turns stored settings into `href` attributes on every public page deserves a look before it has been live long. The review of that code found it sound. The surface it touches did not survive as well.

Finding A - `snap_images.download_url` was never scheme-checked (CLOSED)

Severity: stored XSS, one visitor click, site origin.

`download_url` is written by desktop importers over a scoped API key and later rendered as an `href` on the public photo page. Neither end checked what kind of URL it was.

Write paths, both unvalidated:

Path	What it did
`core/threeacross-api.php:842`	`substr(trim($body['download_url']), 0, 512)`
`smack-backfill.php` (`action=update`, sybu key)	`trim($_POST['download_url'])`

Length and `trim()` are not validation. `javascript:alert(1)` is nineteen characters and contains no whitespace. The backfill endpoint is the worse of the two, because on the same request it also sets `global_downloads_enabled = 1` - the exact condition `download-overlay.php` requires before it renders the button. A bad URL arrived with its own switch.

Render paths, both unsafe, in different ways:

`core/download-overlay.php` ran the value through `htmlspecialchars()`. That is real escaping and it does stop attribute breakout - but it escapes `< > & " '`, and a `javascript:` URL contains none of them. **Escaping is not scheme validation.** The two get confused constantly, and this is what that confusion looks like in production: code that appears defended, is defended against the wrong thing, and reads as fine in review.

`core/social-dock.php` was blunter. It re-extracted the href with `preg_match('/href="([^"]*)"/', ...)` and echoed the capture with **no escaping at all**. Attribute breakout happens to be impossible there - the capture stops at the first ", and the source string had already been escaped - but the scheme passes through untouched, so the outcome is identical.

Fix. `snap_api_safe_link()` - the helper SECAUDIT 040 built for exactly this shape of problem, a stored URL later rendered as an `href` - is now applied at all four points. Both writers reject a non-`http(s)` value before storing it; both renderers re-check rather than trusting the database. The overlay treats an unacceptable URL as absent, so it falls through to the internal tokenised download instead of losing the button. The dock output is escaped as well.

Defence at the write side alone would have been insufficient: rows already stored before this release are still in the database.

Finding B - the Social Dock's own identity links (CLOSED IN 0.7.506D)

Worth recording because it is the same class and was fixed by the IndieWeb work rather than by an audit. Before 0.7.506D the dock rendered `$settings['social_dock_*']` straight into `href` attributes with only `htmlspecialchars()` - the same escaping-is-not-scheme-validation gap as Finding A, reachable by an admin pasting or importing a `javascript:` profile URL. `core/social-dock.php` now routes every one through `snapsmack_indieweb_url()`, which enforces an `http(s)` allowlist.

The IndieWeb change closed a latent issue on its way past. Noted so nobody "simplifies" that call back out later.

The IndieWeb module itself - no findings

`core/indieweb.php` was reviewed line by line and is sound:

- `snapsmack_indieweb_url()` validates with `FILTER_VALIDATE_URL` **and** an

explicit `http`/`https` scheme allowlist - validation, not just escaping.

- Every emitted value goes through `htmlspecialchars(..., ENT_QUOTES, 'UTF-8')`.
- `u-photo` builds `BASE_URL . ltrim($file, '/')`. The `ltrim` is doing real work:

it stops a stored `//evil.com/x.jpg` from becoming a protocol-relative URL that leaves the origin. Worth not removing.

- `dt-published` guards on `strtotime(...) !== false` (strict - a `0` timestamp

is a valid date and is not silently dropped).

- A **disabled** Social Dock publishes no `rel=me` at all. That is the right

default: `rel=me` is an identity assertion, and upgrading SnapSmack must never start publishing profiles the owner chose to keep unpublished.

- No endpoints, no remote fetches, no Webmention/IndieAuth/Micropub. The passive

boundary in the module header is real, not aspirational.

One non-security note: `snapsmack_indieweb_url()` omits the control-character and length checks that `snap_api_safe_link()` has. `FILTER_VALIDATE_URL` rejects those cases anyway, and every consumer escapes on output, so there is no bug - but the two helpers now do nearly the same job with slightly different rules, and that is how they drift. A later consolidation would be tidy.

Regression

`tests/download-url-scheme-regression.php` - **34 checks**, and it was verified to FAIL against the pre-fix tree (6 failures) before being accepted. A guard test that has never seen the bug it guards against is a guess.

It pins:

- 9 hostile schemes rejected, including case variants, control-character

smuggling and leading whitespace;

- 5 real Drive/OneDrive/plain URLs still accepted - a validator that rejects

everything is an outage with good intentions;

- both write paths call the validator;
- both render paths check or escape;
- the dock's identity links stay scheme-checked;
- a disabled Social Dock publishes no `rel=me`;
- `smack-backfill.php` still enables downloads, so that if that behaviour ever

moves, this finding gets re-examined rather than silently invalidated.

Not addressed

- Consolidating `snapsmack_indieweb_url()` and `snap_api_safe_link()`.
- The desktop importers themselves were not audited in this pass. Unzucker in

particular is unaudited, and FLKR FCKR's `config.py` states it was forked from Unzucker's - so SECAUDIT 040's four findings may still be live in the original. That is the next audit.