TLDR Actionizer — Security Exemption for the Flood Cap (MC #106196)

TLDR Actionizer — Security Exemption for the Flood Cap (MC #106196)

Status: Shipped 2026-07-22 · commit `b8e712b093` · file `~/system/daemons/tldr-actionizer.js`

Problem

The TLDR actionizer runs daily (07:30) and classifies each morning-briefing insight as `TASK` or `SUGGEST`. `TASK`-classified insights create an MC task titled `[TLDR] Implement: …`.

To stop those tasks piling up unactioned (original fix MC #102890), `checkFloodGuard()` enforces `FLOOD_CAP = 3`: if 3 open/review `[TLDR] Implement%` tasks already exist, **every new TASK insight is silently dropped** (a `warn` log + Slack line, no task).

On 2026-07-17/18 three auto-generated `[TLDR] Implement` tasks (owner `agentforge`) sat `open` and unactioned for 4–5 days, permanently holding the cap at 3/3. As a result, on 2026-07-21 two **security** insights — WordPress Core (500M sites) and an OpenSSL memory-disclosure CVE — were both dropped by the `flood_cap` guard. A real CVE could be buried behind three stale generic-news tasks. (CEO surfaced the gap: "recommendations but no task.")

Fix

Security-relevant insights are now **exempt from the flood cap** — they always reach the task-create path even when the cap is full. The **dedup guard still applies** to them, so no duplicate tasks are created.

Key elements in `tldr-actionizer.js` (≈ L357–426):

- `SECURITY_KEYWORD_RE = /\b(CVE|vulnerabilit|exploit|RCE|patch|zero-day|malware|supply.chain)\b/i`
- `isSecurityInsight(insight, owner)` → true if `owner === 'securion'` (the cleanest existing signal, reused from the owner router) **or** the insight/apply text matches the regex.
- `checkFloodGuard(insightTitle, isSecurity)` → cap check becomes `if (openCount >= FLOOD_CAP && !isSecurity)`. When at cap **and** security, it falls through to the (unchanged) dedup check and returns `{ skip:false, securityBypass:true }`.
- `createMcTask()` computes `isSecurity` before the guard and, on bypass, emits a distinct `info` log line `"Security insight — bypassing flood cap"` (auditable, separate from the normal `warn` "flood guard triggered" skip).

Non-security `TASK` insights: cap behaviour at 3 is **byte-for-byte unchanged**.

Verification (evidence `~/system/evidence/106196/`)

- `node --check` → exit 0.
- Real `--dry-run --date 2026-07-21`: both CVE insights classify `TASK`/`HIGH`, route `owner=securion`, and reach `[DRY-RUN] Would create MC task`.
- Focused sim harness (execSync stubbed to `openCount=3`) → 4/4 PASS:
  1. security insight at cap → `{skip:false, securityBypass:true}`
  2. non-security at cap → `{skip:true, reason:"flood_cap…"}`
  3. security at cap **with** existing dup → `{skip:true, reason:"dedup…"}` (dedup still applies)
  4. real WordPress CVE text at cap → `{skip:false, securityBypass:true}`

Operating notes

- **Unjamming the cap:** if legitimate `[TLDR] Implement` tasks stall unactioned, close them (`mc.js bulk-close …`) so the cap frees up. Zombie holders block *non-security* creation by design; security now bypasses regardless.
- **Follow-ups (not done here, out of "surgical" scope):** per-owner caps / widening `FLOOD_CAP` semantics; and the MC evidence-gate is hardcoded to `/tmp/evidence-<id>/` which is wiped across reboots — durable evidence lives in `~/system/evidence/<id>/`.
- Fix was committed on branch `feature/email-auto-forward-102448` (working-tree file is what the daemon executes, so it is live); a clean main-merge is separate hygiene.

Revision #1
Created 2026-07-22 21:37:44 UTC by John
Updated 2026-07-22 21:37:44 UTC by John