

# Bilko Landing Lead-Capture Chain — Mechanism, the Two-Month Outage, and the Traps

# Bilko Landing Lead-Capture Chain — Mechanism, the Two-Month Outage, and the Traps

**MCs covered:** #106193 (bilko.cloud KV, genesis), #106358 (bilko.io / bilko.company KV), #106359 (preview namespace isolation), #106405 (dead Turnstile widgets), #106408 (correction to a false "fixed" report), #106413 (HR cookie-banner click interception, OPEN), #106418 (CF Pages drift, structural root cause, OPEN), #106403 (was anything lost before today, OPEN) | **Status:** all three landings proven capturing leads end-to-end as of 2026-07-28 | **Verified against:** azdo/main @ 50e4223a (2026-07-28) + live Cloudflare state read at deploy time | **Last updated:** 2026-07-28

**One-line summary:** a single 2026-06-17 session broke the storage layer and the anti-spam widget on two of three landing sites at once, in a way that produced no error a human would see for the next six weeks; bilko.company did not capture a single lead in its entire existence until it was fixed today.

## 1. The full path a lead takes

Three separate Cloudflare Pages projects, one per market, each a static HTML form (Next.js-exported for two of them) with its own Pages Function:

| Market               | App directory    | Domain      | CF Pages project |
|----------------------|------------------|-------------|------------------|
| io (Serbia, primary) | apps/landing-io/ | bilko.io    | bilko-io         |
| hr (Croatia)         | apps/landing-hr/ | bilko.cloud | bilko-cloud      |

| Market      | App directory    | Domain        | CF Pages project |
|-------------|------------------|---------------|------------------|
| ba (Bosnia) | apps/landing-ba/ | bilko.company | bilko-company    |

```
Visitor fills the form
→ Turnstile (invisible widget, static <script src="../../../turnstile/v0/api.js">) issues a token
→ POST /api/lead { name, email, company, phone, message, cf-turnstile-response }
→ functions/api/lead.js (per project):
  - origin check (must be the site's own https:// origin)
  - field validation (name, email, message length)
  - REJECTS if cf-turnstile-response is empty → "Anti-spam provjera nije završena."
  - POST response to challenges.cloudflare.com/turnstile/v0/siteverify with TURNSTILE_SECRET
  - REJECTS if siteverify fails → "Anti-spam provjera nije prošla."
  - storeRecord(env, key, value) → env.BILKO_LEADS.put(...) (KV, 1-year TTL)
  - sendLeadNotification(...) → Slack chat.postMessage to #ceo (C0AFJDP9V6U), independent cf
  - responds success only if AT LEAST ONE of {KV, Slack} accepted the record
```

**Not obvious, and someone will assume otherwise:** all three projects write into **ONE shared KV namespace** (`BILKO_LEADS`, id `06a6b6112e2e4d17a590db72b3e4b390`), distinguished only by a market-prefixed key — `io_<ts>_<rand>`, `hr_<ts>_<rand>`, `ba_<ts>_<rand>` — and by a `market` field inside the stored JSON. There are not three namespaces. This is a deliberate, pre-existing design (the namespace already held `io_*` keys from May 2026, months before the outage), not an accident of the recent fixes; splitting it would need a migration and is explicitly left as an open question in `fix-verdict.md`, not a decision made here.

**bilko.cloud (hr) only** also exposes an unauthenticated `conversion_event` path on the same endpoint (`type: "conversion_event"`), used for click-tracking (CTA clicks, form-submit-success pings). It has **no Turnstile gate at all** — see §4 for why that matters to how a fix gets verified.

## 2. How it was broken for about two months — the failure modes ARE the documentation

A single session on **2026-06-17** did two unrelated things to two of the three sites at once, and the combination is what made it invisible:

### 2.1 The KV binding was silently dropped

`wrangler.toml` was introduced for `landing-io` (commit `ef7265b9`) and `landing-ba` (commit `44e6fd7a`) — the same commit also touched `landing-hr`'s config. None of the three declared a `kv_namespaces` block. Because a `wrangler.toml` with `pages_build_output_dir` makes the file the **sole source of**

**truth** for a Pages project's bindings (dashboard-configured bindings are ignored the moment such a file exists), this silently unset `env.BILKO_LEADS` at runtime on the very next deploy of each project — with no visible error, because the handler at the time guarded the write with `if (env.BILKO_LEADS)` and still answered `{"success":true}` regardless (fixed in the current handler, §3).

## 2.2 Two of three Turnstile widgets were recreated, not edited — and the HTML was never updated

The same evening, Cloudflare Turnstile dashboard activity shows the `bilko.cloud` and `bilko.company` widgets were **deleted and recreated** as new widgets with new sitekeys (both timestamped ~21:34-21:35 UTC), while `bilko.io`'s widget was merely **edited in place**, keeping its original sitekey (timestamped ~21:25 UTC, ten minutes earlier). Nobody updated the `data-sitekey` attribute in `landing-hr/index.html` or `landing-ba/index.html` to point at the new widgets. A sitekey naming a deleted widget makes Cloudflare's own API answer *"Trying to access a deleted widget"* — `api.js` never issues a token, the hidden `cf-turnstile-response` field stays empty forever, and the handler correctly rejects with *"Anti-spam provjera nije završena."* before the request ever reaches KV. `bilko.io` **survived this evening entirely by accident** — it happened to be edited rather than recreated.

## 2.3 The combined effect, and why nobody noticed for six weeks

`bilko.company`'s form could not be submitted by anyone from 2026-06-17 onward — and, per the KV record, had **never captured a single lead in its entire existence** before this was fixed on 2026-07-28. `bilko.cloud`'s form was equally dead from the same date, though its KV binding alone had been noticed and "fixed" once already (MC #106193, 2026-07-27) — see §4 for why that fix did not actually restore the form. `bilko.io` kept working the whole time because its widget survived, but it too lost its KV binding on 2026-06-17 (fixed alongside `bilko.company` in MC #106358).

No error was visible anywhere that anyone was routinely looking: the endpoint kept answering `200/success:true` until the KV-binding fix landed (which then correctly started reporting failure, which is what eventually surfaced this), and the Turnstile failure produces a page-level inline message a visitor sees but nobody internally does. There is no dashboard, alert, or count that would have caught this sooner (see §5 on why `#106403` — whether real leads were lost — is still open and may be unanswerable).

---

### 3. Current state of the storage-failure handling

The handler in all three `functions/api/lead.js` now:

- throws a typed `StorageError` (`kv_binding_missing` / `kv_write_failed` / `kv_readback_failed` / `kv_write_unconfirmed`) instead of silently no-op'ing on a missing binding;
  - logs the failure and falls back to the Slack notification as a second, independent channel;
  - reports `persisted:"kv"` or `persisted:"slack_only"` to distinguish a durable write from a fallback — it no longer claims a KV write it did not make;
  - returns **503**, not a fabricated success, only if *neither* KV nor Slack accepted the record — a visitor is never thanked for a submission that was completely dropped.
- 

## 4. The traps — each is the reason this outage is worth a page, not just a fix

### 4.1 Deploying from outside the app directory ships static assets with NO Functions

`wrangler pages deploy <path>` run from anywhere other than inside the project directory (e.g. `wrangler pages deploy apps/landing-io` from the repo root) uploads the static files but does **not** bundle `functions/`. The result is not an error — the site loads fine and `/api/lead` just returns **405**, which looks like a routing problem, not a deploy problem. Always `cd` into the app directory and deploy `.`:

```
cd apps/landing-io
wrangler pages deploy . --project-name=bilko-io --branch=main
```

The log must contain **both** `"[ ] Compiled Worker successfully"` **and** `"[ ] Uploading Functions bundle"`. Either line missing means the Function did not ship, regardless of what the site's homepage looks like.

### 4.2 `wrangler kv key delete --force` is not a valid flag in wrangler 4.83

It prints usage help and deletes nothing, with no error exit code that would make a script notice. Use the Cloudflare REST API for deletes during test cleanup (`DELETE /accounts/:acct/storage/kv/namespaces/:ns/values/:key`), and always **re-list the namespace afterward** to confirm the count actually dropped — a delete that silently no-op'd looks identical to one that worked if you only check the HTTP status of the delete call itself.

## 4.3 Preview inherits the production KV binding by default — this is the normal state, not a misconfiguration

`kv_namespaces` is a [non-inheritable key](#) in Cloudflare's own terms, which — counter-intuitively — means the opposite of what "non-inheritable" suggests: with **no** explicit `[env.preview]` override, a `wrangler pages deploy` from any non-production branch applies the **top-level** `kv_namespaces` block to preview too. So by default, every preview deployment of every one of these three projects writes preview traffic straight into the **production** leads namespace. This is exactly what happened to `bilko-cloud` on 2026-07-27 (MC #106359) from an ordinary preview deploy — proven, not theorized: a real preview write was shown landing in the production namespace before the fix, and in a dedicated preview namespace (`BILKO_LEADS_PREVIEW`, id `dcd1f732b2d24850b6c81179beef9909`) after it.

**Check this on every project** before relying on preview being isolated — it is not the default, and the fact that `landing-io`/`landing-ba` declare `[env.preview] kv_namespaces = []` (no preview binding at all, falls back to Slack-only) while `landing-hr` gets a real preview namespace is a deliberate, stated inconsistency (HR is the only landing with the unauthenticated `conversion_event` path, which made a real isolation test possible there) — not an oversight to "fix" toward uniformity without a reason.

## 4.4 The HR cookie banner does not block Turnstile — it intercepts the CLICK (MC #106413, OPEN)

**Only** `landing-hr` has a cookie-consent banner; it is entirely absent from `landing-ba` and `landing-io` (confirmed: zero matches for `cookie-banner` in either file). The first diagnosis of the resulting symptom was wrong and is worth recording as a lesson in itself: it is **not** that the banner blocks Turnstile from loading — Turnstile is not consent-gated at all, and loads and runs its challenge with no consent choice made. What actually happens: the banner is `position:fixed; bottom:0; z-index:999` and occupies the bottom ~84px of the viewport. When a visitor scrolls the form to its natural end position, the submit button's rect lands entirely inside that bottom strip, and the click goes to the banner element, not the button. **The submit handler never runs, so there is no error, no spinner, and no message of any kind** — the single worst failure shape, because the

visitor has no reason to try again and there is no signal on our side either. Dismissing the banner (either choice) removes the overlay and the exact same click then works. Status: open, not yet fixed — proposed remedy is bottom padding equal to the banner height while visible, or restricting the banner's clickable area to its own buttons.

## 4.5 CF Pages projects are not reliably git-integrated — merging to main can deploy nothing, and manual deploys bypass the merge gate entirely (MC #106418, OPEN)

**This is the structural root cause that allowed the whole chain in §2 to happen and go unnoticed** — not just one more fact about this system. GitHub Actions workflow files nominally exist for auto-deploy on push to each project's `main`, but the GitHub mirror of this repository (`gh-mirror-stale/gh-ssh`) is itself frozen roughly 300 commits behind `azdo/main` (documented separately, MC #106042) — so those workflows, even where they still fire, are not deploying current content. The **actual** mechanism in practice is a person running `wrangler pages deploy` by hand from a freshly synced `azdo/main` checkout after a PR merges. That means:

- Merging a PR to `azdo/main` deploys **nothing** by itself — confirmed live: PR 201 (a cosmetic Turnstile attribute cleanup) has been in `azdo/main` since before this outage was fixed, and was still not live on any of the three sites at time of writing, purely because nobody had run the manual deploy step since it merged.
- `wrangler pages deploy` run manually bypasses the azdo PR/CI merge gate entirely — it is possible to ship code to these three production domains that never went through review.
- The only reliable way to know what commit is actually live is `deployment_trigger.metadata.commit_hash` on the Pages deployment record (via the Cloudflare API), **not** `git log azdo/main` and not the GitHub Actions run history.

Open, undecided as of this writing: either give these three projects real git integration (merge → deploy, matching the rest of Bilko's pipeline discipline) or add an explicit deploy stage to the azdo pipeline; failing either of those, at minimum build a drift detector that compares the commit each Pages project is actually serving against `azdo/main`'s tip and alerts on divergence. A deliberate decision was made NOT to deploy PR 201 immediately after these fixes landed, specifically to avoid re-risking freshly-proven-working forms for a cosmetic change — that is a one-time judgment call, not a substitute for the structural fix.

---

## 5. How to verify a fix here — and what does NOT count

Two specific mistakes already happened in this chain and are worth naming so a third does not:

- **Automated browsers cannot pass Turnstile** (Playwright, headless or headed, real Chrome via CDP — all return `TurnstileError 600010`, the automation refusal). An automated probe of the lead form therefore proves nothing about whether a real visitor can submit it — it can only ever demonstrate that the anti-spam check is doing its job.
- The `conversion_event` path on `bilko.cloud` has **no Turnstile gate at all**. Verifying a fix through that path proves only that KV writes work in general — exactly the part that was **not** broken. This is precisely the mistake that produced a false "HR lead capture is fixed" report (MC #106193 → corrected the next day as MC #106408): the verification exercised the unprotected event-tracking path and never touched the actual, still-dead lead form.

**A real fix is verified by a human driving the actual form in a real browser**, then reading the stored record back out of KV and matching a discriminator field (e.g. a company name like `MC106405-HR-DELETE-ME` chosen for the test) against what was typed — followed by deleting the test key and re-listing the namespace to confirm both the write and the cleanup actually happened. This was the exact standard applied on 2026-07-28: a human (team-lead, through the CEO's real Chrome) submitted all three forms for real; every record was read back verbatim; all test keys were removed and the namespace count returned to its pre-test value.

## 6. Related open tickets

| MC #    | Status  | What it is                                                                                                                                                                  |
|---------|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| #106413 | OPEN, H | HR cookie-banner intercepts the submit click with zero user-visible feedback — §4.4. Not fixed; a proposed remedy exists but is not implemented.                            |
| #106418 | OPEN, H | CF Pages drift is structural, not a one-off — §4.5. Two remedies proposed (git integration or a drift detector), neither implemented; PR 201 deliberately not yet deployed. |

| MC #    | Status  | What it is                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|---------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| #106403 | OPEN, H | Whether any real customer enquiries arrived during the outage and were lost is <b>currently unanswerable</b> : the Slack channel (#ceo, C0AFJDP9V6U) shows zero lead-shaped messages in its last 200, and the newest pre-fix <code>io_*</code> KV key predates the outage by three weeks (2026-05-26) — but there is no third, independent source (e.g. Cloudflare Web Analytics request counts) to distinguish "no enquiries arrived" from "enquiries arrived and both capture channels silently dropped them". Whether the Slack bot token even has write permission to that channel had not been separately verified as of this writing. |

## 7. Source

Verified against `azdo/main` @ `50e4223a` (2026-07-28) plus live Cloudflare account/API state read at fix and deploy time (account `d0ac2afb6bb5b298723b85a114151a04`).

- `apps/landing-io/functions/api/lead.js`, `apps/landing-io/wrangler.toml`, `apps/landing-io/index.html`
- `apps/landing-hr/functions/api/lead.js`, `apps/landing-hr/wrangler.toml`, `apps/landing-hr/index.html`
- `apps/landing-ba/functions/api/lead.js`, `apps/landing-ba/wrangler.toml`, `apps/landing-ba/index.html`
- `DEPLOY-MAP.md` — "CF Pages projects" section carries the operational recipe and the per-project binding/Turnstile tables this page summarizes; that file is the source for exact command syntax, this page is the narrative and the trap history. Note: at time of writing its "Status per project" table still reads bilko-io/bilko-company as "pending deploy" — that is stale; both were deployed and proven live the same day (§5), and this page reflects the current, corrected state.

Evidence: `~/system/evidence/106193/`, `~/system/evidence/106358/fix-verdict.md` + `deploy-verdict.md`, `~/system/evidence/106359/fix-verdict.md`, `~/system/evidence/106405/fix-verdict.md`.

Personal-data note: any stored lead record quoted in the evidence above has its submitter IP redacted; test submissions used internal `flowforge+<mc>@alai.no` addresses, not real customer data.