IAG Journal — Security Verification Checklist (macOS)

What this is: A step-by-step protocol for independently verifying IAG Journal's security claims — code identity, the local-first architecture, and the safety of TastyTrade AutoSync — using free tools and observable evidence. No trust required; every claim gets tested.

Origin: Executed in full against the shipping build in July 2026 by a subscriber (iOS engineer / options trader) prior to connecting a live brokerage account. All checks passed. Reference results are included at each step so you know what "pass" looks like.

Time: ~90 minutes, mostly passive monitoring. **Tools:** Terminal (built in), LuLu (free, open-source firewall from Objective-See), a web browser.

Phase A — Verify the binary (before granting anything)

Claim under test: the app you downloaded is the app the developer built, signed by a verified identity, and scanned by Apple.

1. Install the app by dragging it to `/Applications` and **eject the DMG**. Always run from `/Applications` — running from the mounted disk image triggers macOS app translocation and breaks updates.
2. Verify the code signature:

```
codesign -dv --verbose=4 "/Applications/Investing Against the Grain Journal.app"
```

Pass: - `Authority=Developer ID Application: Daniel Lemmon (C86X522W2S)` — an Apple-identity-verified developer certificate - `flags=0x10000(runtime)` — hardened runtime enabled

3. Verify Apple notarization:

```
spctl -a -vv "/Applications/Investing Against the Grain Journal.app"
```

Pass: `accepted` + `source=Notarized Developer ID` — Apple's notarization service scanned this exact build for malware.

Fail conditions (stop and contact support): unsigned or ad-hoc signature, a different signer identity, or `rejected` from spctl.

> *Re-run step 3 after every fresh download. It takes five seconds and re-verifies the supply chain.*

Phase B — Watch the network (the local-first claim)

Claim under test: "Your trades never leave your machine."

4. Quit IAG Journal completely. Install LuLu and approve its network extension (System Settings → General → Login Items & Extensions → Network Extensions → enable LuLu → Allow the filter prompt).
5. Configure LuLu (menu bar icon → Settings): - **Mode:** Passive Mode **OFF** (you want alerts) - **Rules: UNCHECK "Allow Already Installed Programs"** — this is the critical one; if left on, LuLu silently whitelists the app and the entire test observes nothing - **Rules:** "Allow Apple Programs" ON (suppresses OS noise), DNS + localhost allowed - View Rules → delete any pre-existing rules for the app
6. Launch IAG Journal and record every alert bearing its name. For each alert, expand **Details & Options** and use the process-ancestry icon to confirm the connection truly originates from the app. **Reference result:** exactly **one** destination at launch — the app's account/entitlement backend, port 443.
7. Log in when prompted. **Reference result:** login is browser-based with a localhost callback — the app never handles your password. No new destinations beyond the backend.
8. Let the app idle **15+ minutes**. **Reference result:** silence. (Observed: 2+ hours, zero new connections.)
9. **The litmus test — CSV import.** Import a transaction CSV while watching LuLu. **Pass: zero new outbound connections** at the moment trade data enters the app. **Reference result:** completely silent on the wire. Trade data renders from the local database with no network involvement — observable proof of the local-first claim.
10. Identify every destination you allowed. LuLu often shows a bare IP (many cloud hosts publish no reverse DNS — normal). Look up ownership at ARIN: `https://search.arin.net/rdap/?query=<IP>` **Reference results (July 2026):**

Destination	ARIN registrant	Purpose
64.29.17.65	Vercel, Inc.	App account / entitlement backend
(after Phase D) 170.76.246.66	tastyworks, Inc.	AutoSync — direct to broker

Any destination that doesn't map to the published stack (Vercel/Supabase/Stripe/GitHub per the privacy policy) or to tastytrade deserves a question before allowing.

> *Tip: when creating allow rules, choose **Endpoint** scope rather than Process scope. Process-scope rules whitelist all future destinations for the app, which blinds later observations.*

Phase C — Data at rest

11. Confirm the database is local:

```
ls ~/Library/Application\ Support/ | grep -i iag
```

Reference result: `iag-journal` — your trades live in that folder, on your disk.

12. Confirm full-disk encryption:

```
fdesetup status
```

Pass: `FileVault is On.` If not, enable it before importing brokerage data — the local database is the asset the local-first model protects, and it flows into Time Machine backups too.

Phase D — TastyTrade AutoSync (the credential decision)

Claims under test: the app never sees your password, access is read-only, and brokerage data flows direct to tastytrade — not through IAG servers.

13. Follow the in-app AutoSync instructions to create a personal OAuth client at tastytrade. At the scope screen: **check `read` only. Leave `trade` unchecked.** This is enforced server-side by tastytrade — a read-only token is architecturally incapable of placing orders, no matter who holds it.
14. Store the `client_secret` and `refresh_token` in a password manager only (shown once; never in notes/files/screenshots).
15. Paste into IAG Journal and connect. Per the app's guidance, set backfill start date appropriately if you've already imported history via CSV.
16. **Post-connect verification (three checks):** - **Traffic:** the sync fires a new LuLu alert. Identify the IP via ARIN. **Pass:** registrant is **tastyworks, Inc.** — your brokerage data goes Mac → tastytrade, direct. **Fail (revoke immediately):** sync traffic terminating at the app vendor's own infrastructure would mean brokerage data proxying through their servers. - **Keychain:** open Keychain Access, search "iag". **Reference result:** `iag-journal-tt` entries in the **login** keychain (local, FileVault-protected, *not* iCloud-synced). - **Kill switch:** tastytrade → Manage → My Profile → API → **Manage OAuth Grants** — confirm your grant is listed. This page revokes the app's access instantly and unilaterally, forever. Know where it is.

Ongoing posture

- Leave LuLu resident. Any **new** destination from the app after an update gets identified (ancestry → ARIN) before allowing.
- After each app update: re-run the `spctl` check (step 3) and glance at LuLu.
- The `read` scope is permanent. Treat any future request to expand permissions as a fresh security review.
- Keep your own system of record. A journal is an analytics layer; your data discipline shouldn't depend on any one tool.

Verdict from the July 2026 run

Signed and notarized binary · two total network destinations across launch, login, hours of idle, CSV import, and live sync · zero traffic correlated with trade data entering the app · sync direct-to-broker (ARIN-confirmed) · passwordless OAuth with server-enforced read-only scope · credentials in the local keychain · unilateral revocation available at tastytrade.

Every published claim was observed, not just believed. Approved — connected to a live account.