MultiHopper Bounty Submission / Security Disclosure

Bounty: "Break It Before Users Do: MultiHopper Agentic Flow Bugs & Fixes" (Superteam Earn) **Channels:** Superteam Earn listing (bounty) · security@multihopper.com (security disclosure). This report contains **no secrets** — the `mh_test_` API key, signer private key, and wallet recovery material are NOT included and were never shared (per your support notice).

Responsible disclosure: all testing was on **devnet** with a self-served `mh_test_` key. No mainnet value moved. Please keep private until remediated.

Requirements compliance

Explicit confirmation against the bounty's stated requirements:

Requirement	How this submission complies
Use only the approved **test environment / keys / wallets / balances**; no mainnet or live user funds unless authorized in writing	All active testing was on **devnet** via a self-served `mh_test_` key and a devnet wallet with test SOL. **No mainnet transactions were sent and no live user funds were touched.** The single mainnet-related finding (UPGRADE-AUTH) is based solely on **read-only inspection of public on-chain program metadata** (`getAccountInfo`/`getProgramAccounts` — no auth, no transactions, no funds) and public program bytecode; that is public-ledger data, not testing against live funds.

Submission must concern **MultiHopper API usage in an automated/agentic workflow** (generic UI feedback not eligible)	Every finding targets the agent-facing REST flow (`create → prepare → sign/broadcast → confirm-broadcast → monitor`), the MCP surface, webhooks, and the client-side signing/recovery rules — exercised by an **automated TypeScript harness**, not a browser. No UI-only feedback is included.
Clear **reproduction path**: environment, request flow, payloads (secrets removed), expected vs actual, evidence	Each finding lists the endpoint/method, steps, a redacted payload, expected vs actual outcome, and links captured evidence in `evidence/`. The harness + `curl` repros are runnable; each surfaces a fresh `requestId`.
Practical suggested fix / mitigation / doc improvement for each issue	Every finding includes a concrete Fix; the 19 documentation issues each include a proposed doc correction.
Never expose private keys, seed phrases, API keys, or third-party secrets	Verified: no secrets in the repo or this report — the `mh_test_` key and signer key are `.env`-only (gitignored); only `.env.example` placeholders are committed. All sample payloads are redacted.
Do **not** exploit, drain, spam, disrupt, or publicly disclose before review	Testing was non-destructive and rate-limit-respecting (test-mode only, no real value moved, no DoS). This report and repo are kept **private** and shared only with MultiHopper until reviewed/remediated.
Submissions in **English**	This report is in English.
Submit in an **accessible Google Doc / Notion / etc.**	Provided as importable HTML (`MultiHopper-Submission-Full.html` → Google Drive → Open with Google Docs) alongside this Markdown; happy to share as a Google Doc/Notion link on request.

Environment / debug context (adapted for an API + autonomous-agent integration)

Field	Value
Where it happened	`https://devnet.multihopper.com/api/v1` (devnet REST API) + `dev-docs.multihopper.com/mcp` + on-chain (mainnet program config)
Integration / provider wallet	`GfcpRxboaev5cHDLUR86t1Bh5c5TtGGCP2pQFQKf9KGv`
Source / signer wallet (devnet)	`3BqHVNkHyLRu2wN2MdzKyUxUJm3iKDuLdEKE8C7iwMDH`
API key mode	`mh_test_` (devnet dashboard)
Client	Automated agent + `curl` (not a browser) — so browser/OS/Phantom-version fields are N/A
Time window	2026-07-10, ~14:00–18:00 UTC
Tooling	Interface-based TypeScript harness + Schemathesis + Solana RPC + sBPF disassembly (repo attached)
Sample request IDs (MH_090 auth/usage 500s)	`1750664f-6555-a9eb-7604-5a4453ba072a`, `8ef6ce7b-684a-a487-0ea0-885704811d33`, `6e90c9f6-1417-8bf2-258c-fea014a4458a`

Request IDs are per-request; each finding below includes a one-line repro that surfaces a fresh `requestId` in the response body.

Findings (severity-ordered; full detail + evidence in `SUBMISSION.md` and `evidence/`)

For each: **where**, **steps**, **evidence/request-id**, **fix**. Plain-English summaries in `SUBMISSION.md`.

1. **[High] Single-key program upgrade authority (centralization / trust-model breach).** Where: mainnet programs `GLWfkHgg…` (Core / permanent delegate), `4TPCmR2u…` (Orchestrator), `5GbQ8AM…` (Transfer Hook Guard). Steps: `getAccountInfo` → all three upgradeable by one System-owned wallet `2QTAuHxo…` (not a multisig). Contradicts the docs' "you don't need to trust the MultiHopper team." Fix: multisig/timelock or immutability. (`evidence/onchain-upgrade-authority-and-pricing.md`)
2. **[High] Auth-lookup 500 leaks the `api_keys`/`integrations` schema.** Where: `POST /transfers/estimate` (intermittent; also GET endpoints). Steps: send a normal request; on the intermittent fault the 500 body returns the raw `select "api_keys"…"key_hash","key_prefix"…` query. Evidence: 6 consecutive 500s captured (`evidence/api-keys-auth-leak.md`); request IDs above. Fix: error-boundary mapping DB errors to a generic code with no query text.
3. **[High] Numeric-overflow SQL/schema leak.** Where: `POST /transfers`. Steps: `amountRaw="9223372036854775808"` → 500 leaking the `orchestrator_steps` INSERT + columns. Repro: see `evidence/A13-sql-schema-leak.json`. Fix: bounded-BigInt validation before DB write.
4. **[High] Webhook SSRF.** Where: `POST /webhooks`. Steps: register `url:"http://169.254.169.254/latest/meta-data/"` → accepted (persisted `isActive:true`). Fix: block loopback/RFC1918/link-local/metadata; re-validate at delivery.
5. **[High] Webhook signature verification broken.** Where: `api-reference/webhooks/events` + live delivery. Server signs header **`x-mh-signature`** over the **raw body**, but the docs' verifier reads `x-multihopper-signature` and has no `timingSafeEqual` length guard → integrators reject/crash on every webhook. Evidence: `evidence/webhook-delivery.md`. Fix: correct header + length-guarded raw-body verify.
6. **[High] Fee mispricing via client `tokenDecimals`.** Where: `estimate`/`create`. Steps: `tokenDecimals=6` → `tier=premium, usd=150000, 50bps` vs `=9` → `standard,150,5bps` (10× fee / 1000× value). Fix: resolve decimals from the mint.

7. **[High] Idempotency TOCTOU race.** Where: `POST /transfers/:id/confirm-broadcast`. Steps: 6 concurrent identical requests (same Idempotency-Key) → all bypass the lock (0× `MH_072`). Fix: atomic acquire (INSERT … ON CONFLICT).
8. **[High] Duplicate transfer race.** Where: `POST /transfers`. Steps: 2 concurrent creates, same `externalId` → 2 transfers created (want 1). Fix: unique constraint / atomic dedup on externalId.
9. **[Medium] CORS reflects any Origin with credentials.** Where: all endpoints. Steps: `Origin: https://evil.com` → `Access-Control-Allow-Origin: https://evil.com` + `Allow-Credentials: true`. Fix: origin allowlist / drop credentials.
10. **[Medium] `MH_040` quote-stale for boundary amounts.** Where: `create`→`prepare`. Steps: 10 SOL → create quotes `50bps/2000flat`, immediate prepare says `5bps` → `MH_040` re-quote loop. Fix: single pricing source.
11. **[Medium] `amountRaw`/`amountTokens` not cross-validated.** Contradictory amounts accepted (protocol uses `amountRaw`; `amountTokens` is stored-but-unused → misleading). Fix: validate or derive one from the other.
12. **[Medium] Interrupted deploy stuck with no error/recovery signal** (`status:"active"`, `lastError:null`, `recovery:null`, past deadline). Fix: surface a resume hint / post-funding deadline.
13. **[Low/Med] Docs (19 findings) contradict the API** — broadcast order (keeper last vs first), `arrivalSeconds` min, undocumented `status:"active"` + on-chain error codes, screening-fee-in-estimate, www 308 auth-strip, signing warning, quickstart Idempotency-Key omission, etc. See `DOCS-ISSUES.md`.

Confirmed secure (tested, held): IDOR/BOLA scoping, webhook-delete IDOR, mass-assignment, input validation (overflow/type-confusion/prototype-pollution/injection), auth parity, verb tampering, `keeper_share_bps<=10000`, and the transfer hook's delegate-only gate (verified at the bytecode level).

For the team to run (need source / mainnet / a rival keeper): `MAINNET-TESTS.md` (T1 wrapper burn/close bypass, T3 keeper reward theft, T6 mint/burn invariant + hook `is_transferring`, T9 compliance-proof bypass).

Attachments

Private repo (access granted on request):
https://github.com/lpsmurf/multihopper-flow-audit Contains the harness, 54 offline tests,

`evidence/`, `SUBMISSION.md`, `ATTACK-MATRIX.md`, `DOCS-ISSUES.md`, `MAINNET-TESTS.md`, `TOOLING.md`. Every finding has a runnable repro. No secrets are committed (`.env` is gitignored; only `.env.example` placeholders are included).

Contact: please reply to this email; happy to grant repo access and walk through any repro.