Third-Party Security & Functionality Audit

qXRP Falcon Ledger Wallet - Bridge, Pool/AMM, Quick Swap & Limit Orders

Independent Security Review

2026-07-07

Contents

Figure 1: Falcon Logo

Classification: Informational - testnet application

Target: `beartec-jpg/Falcon-faucet-wallet` (qXRP Web Portal)

Network: Falcon Ledger testnet (ID 1001) + Sepolia EVM

Review cycle: Initial assessment -> remediation -> re-test

---

1 Third-Party Security & Functionality Audit

1.1 qXRP Falcon Ledger Wallet - Bridge, Pool/AMM, Quick Swap & Limit Orders

Report type	Independent third-party security & functionality review (remediation cycle)
Target	`beartec-jpg/Falcon-faucet-wallet` (qXRP Web Portal)
Scope	Bridge (in/out), AMM Pool (deposit/withdraw), Quick Swap, DEX Limit Orders
Network	Falcon Ledger **testnet**, Network ID `1001` (Falcon-512 post-quantum accounts); EVM side on **Sepolia**
Methodology	White-box source review, architecture & threat-model analysis, transaction-lifecycle and finality analysis, remediation verification
Classification	Informational - testnet application
Review cycle	Initial assessment -> remediation -> re-test

> **Disclaimer.** This report reflects a point-in-time review of the source code. It is not a penetration test and does not guarantee the absence of vulnerabilities. Findings are rated for the **testnet** context; several accepted risks would escalate to **High/Critical** before any mainnet or real-value deployment.

1.2 1. Executive Summary

The DeFi surface (bridge, AMM pool, instant swap, limit orders) inherits the wallet's strong **non-custodial, in-browser signing** model: the `falcon_secret` is decrypted and signed entirely in the browser via WASM and never leaves the device. The constant-product AMM math is correct, the instant-swap path re-quotes immediately before signing and enforces on-ledger `SendMax`/`DeliverMin` bounds, and the EVM bridge key is CSPRNG-generated and passkey-encrypted.

The **initial** assessment identified **three High-severity** cross-cutting issues plus a set of Medium/Low hardening items. All three High findings - and the two most material Mediums - have since been **remediated and verified**. This report documents the full lifecycle: initial issues, the fixes applied, and the re-tested final posture.

1.2.1 Outcome at a glance

Phase	Critical	High	Medium	Low	Info
Initial assessment	0	3	4	4	1
After remediation	0	**0**	**0**	5	1

Result: No Critical or High findings remain. **No residual Medium findings** remain: two Mediums were fully resolved and two were reduced to Low with the on-ledger/client bounds now in place. The remaining Low/Info items are documented, accepted testnet risks.

1.3 2. Scope & Architecture

1.3.1 2.1 In scope

- **Bridge-in:** `src/components/BridgeDepositPanel.tsx`, `src/lib/evm-bridge-client.ts`, `src/lib/bridge-config.ts`, `src/app/api/bridge/config/route.ts`, `src/lib/bridge-memo.ts`.
- **Bridge-out:** Falcon `Payment` + `sepolia-withdraw` memo (`src/lib/falcon-tx-sign.ts`), relay `WithdrawalReleased` watcher (`waitForWithdrawalRelease` in `src/lib/evm-bridge-client.ts`).
- **Pool / AMM:** `src/components/MarketLiquidityPanel.tsx`, `src/lib/swap/amm-math.ts`, `src/app/api/market/*`, `src/app/pool/page.tsx`.
- **Quick Swap:** `src/app/swap/page.tsx`, `src/lib/swap/quote.ts`, `src/lib/swap/*`.
- **Limit Orders:** `src/components/DexOrdersPanel.tsx`, `src/app/api/market/offers|orderbook`.
- **Cross-cutting write path:** `src/lib/wallet-submit.ts` (`submitWithSequenceRetry`), `src/lib/falcon-tx-sign.ts`

1.3.2 2.2 Trust model

- **Client (browser)** is the trust anchor: keys generated, encrypted, and used there.
- The **relay/server** is untrusted for confidentiality of secrets; it only relays public reads and pre-signed transaction blobs. The custodial bridge relay is a trust anchor for **liveness/value transfer across chains** (accepted, documented - see F-12).
- The **RPC/DEX read APIs** are untrusted for **integrity of returned amounts**; client code must range-validate parsed values.

1.4 3. Severity Methodology

Ratings combine **impact** (loss of funds, stuck/mispriced transactions, availability) and **likelihood** (attacker position / preconditions), adjusted for the **testnet** context.

- **Critical** - remote key/fund compromise, low preconditions.
- **High** - fund loss or a wrong/failed on-ledger outcome under realistic conditions.
- **Medium** - meaningful weakening of a control; exploitable under additional conditions.
- **Low** - limited or defense-in-depth impact.
- **Informational** - hardening / hygiene.

1.5 4. Part I - Initial Findings (pre-remediation)

ID	Severity	Finding	Area
F-01	**High**	No slippage / min-output protection on AMM deposit & withdraw (unlike swap)	Pool/AMM
F-02	**High**	Sequence-number race - read-modify-write with no re-fetch/retry on `tefPAST_SEQ`/`terPRE_SEQ`	All writes

ID	Severity	Finding	Area
F-03	**High** (mainnet) / Med (testnet)	Bridge release watcher scans only 1 block back (`RELEASE_LOOKBACK_BLOCKS = 1`) - reorg/miss risk	Bridge-out
F-04	Medium	JavaScript `number` (IEEE-754 double) used for money; precision loss vs XRPL 16-digit IOU	Pool/AMM/Swap
F-05	Medium	Bridge-in uses 1-block confirmation (`tx.wait(1)`) for approve/deposit	Bridge-in
F-06	Medium	Untrusted RPC/server values parsed without range validation (NaN/Infinity/negative)	Pool, orders
F-07	Medium	Pool snapshot not refreshed before signing; stale ratio drives deposit matching	Pool/AMM
F-08	Low	No EIP-55 / XRPL checksum verification on bridge addresses	Bridge
F-09	Low	`lastLedgerSequence = current + 20` fixed (~60-100s) can expire under network lag	All writes
F-10	Low	No client-side upper-bound / safe-integer check on drops math	Swap, orders
F-11	Low	Limit orders can cross the AMM with no price cap	Limit orders
F-12	Info	Custodial relay is a full trust anchor; no on-chain attestation of memo parsing	Bridge

1.5.1 4.1 Detail - the three High findings

F-01 - No slippage/min-out on AMM liquidity ops. `AMMDeposit` (`tfTwoAsset`) treats `Amount`/`Amount2` as on-ledger *maximums* and `AMMWithdraw` (`tfLPToken`) has **no** on-ledger `min-received` field. Without a client-side guard, a pool that moves between quote and submission could consume a deposit at an unexpected ratio, or return less than expected on withdraw, with no abort. The instant-swap path had this protection (`SendMax`/`DeliverMin`); the liquidity path did not.

F-02 - Sequence-number race. Every write path (swap, orders, pool, bridge-out) built a transaction from a single cached account `Sequence`. Two writes in flight - or a stale cache - cause the second to be rejected `tefPAST_SEQ`/`terPRE_SEQ` with no recovery, surfacing as a spurious failure. Because signing is in-browser, any retry must re-fetch -> re-sign -> resubmit **client-side**.

F-03 - Bridge release watcher reorg-fragile. `waitForWithdrawalRelease` scanned only `tip - 1` block for the `WithdrawalReleased` event. A short reorg, a batched/late relay call, or slow polling could place the event outside the 1-block window, so the UI would never observe a release that actually happened. Low impact on an isolated testnet, High on mainnet.

1.6 5. Part II - Fixes Applied

1.6.1 F-01 -> Fixed: client-side slippage guards on both liquidity ops

- **Withdraw min-out guard.** Before signing, the live LP position is re-fetched and the now-expected `FALCON/F-USDC` outputs are compared against the user's slippage-derived minimums (`applySlippage(..., 'min')`); the submission **aborts** with a clear "pool moved" message if either falls short. `src/components/MarketLiquidityPanel.tsx:345-360`, `src/lib/swap/amm-math.ts:applySlippage`.
- **Deposit ratio-drift guard.** The live pool ratio is re-fetched immediately before signing and compared to the ratio the deposit was sized against; if drift exceeds tolerance the deposit is **aborted** rather than executed at a shifted ratio. `src/components/MarketLiquidityPanel.tsx:264-283`.
- A named `FP_COMPARE_EPSILON` (`1e-9`) absorbs floating-point rounding in the comparisons.

1.6.2 F-02 -> Fixed: transparent sequence re-fetch/re-sign/resubmit

- New `submitWithSequenceRetry` (`src/lib/wallet-submit.ts`) re-fetches a fresh `Sequence` + validated ledger index, **re-signs in-browser**, and resubmits on `tefPAST_SEQ`/`terPRE_SEQ` (`SEQUENCE_RETRY_RESULTS`), up to 3 attempts with incremental backoff. The `falcon_secret` stays inside the signing closure and is never sent to the server.
- Wired into **every** write path: `src/app/swap/page.tsx`, `src/app/pool/page.tsx`, `src/app/rewards/page.tsx`, `src/app/wallet/page.tsx`, `src/components/MarketLiquidityPanel.tsx`, `src/components/DexOrdersPanel.tsx`, `src/components/BridgeDepositPanel.tsx`.

1.6.3 F-03 -> Fixed: chain-aware lookback + overlapping re-scan + de-dup + env override

- Replaced the fixed `RELEASE_LOOKBACK_BLOCKS = 1` with a chain-appropriate default (`defaultReleaseLookbackBlocks`: mainnet `200`, testnets `50`) and a per-call / config override (`release_lookback_blocks`, env `RELEASE_LOOKBACK_BLOCKS`). `src/lib/evm-bridge-client.ts`, `src/lib/bridge-config.ts`, `src/app/api/bridge/config/route.ts`.
- The watcher now advances a cursor but always keeps a `lookback` **overlap** each poll (`fromBlock = min(scanFrom, latest - lookback)`), so events landing during a reorg or between polls are re-scanned rather than dropped. A `seen` set keyed on `txHash:logIndex` de-duplicates across overlapping windows and the newest matching release is returned. `src/lib/evm-bridge-client.ts:waitForWithdrawalRelease`.

1.6.4 Mediums addressed

- **F-06 -> Fixed.** User-supplied and derived amounts are `Number.isFinite`/`> 0` (and range-) validated before signing across the liquidity and order paths (`src/components/MarketLiquidityPanel.tsx:135,187,241,314`; `src/lib/swap/quote.ts:55,110,123,136`; `src/components/DexOrdersPanel.tsx` amount/price guards), rejecting `NaN`/`Infinity`/negative inputs.
- **F-07 -> Fixed.** The stale-snapshot risk is closed by the **same pre-submit freshness re-fetch** introduced for F-01: the pool is re-read immediately before signing on both deposit and withdraw (`fetchLpPosition()` re-fetch), so matching is driven by a fresh ratio, not a stale snapshot.

1.6.5 Mediums reduced to Low (bounded, accepted for testnet)

- **F-04 (float for money) -> Low.** No change to the `number` representation, but every value-moving path now enforces **explicit output bounds** - swaps via on-ledger `SendMax`/`DeliverMin` (`src/app/swap/page.tsx:295-321`) and liquidity ops via the new client min-out/ratio guards - so double-rounding cannot silently produce a materially worse fill. Migrating to integer/decimal money types remains recommended before mainnet.
- **F-05 (1-conf bridge-in) -> Low.** The cross-chain finality/liveness decision ultimately rests with the custodial relay (already a documented trust anchor, F-12), and the bridge-out watcher is now reorg-tolerant (F-03). On testnet the residual 1-confirmation window is accepted; deeper confirmations are recommended for mainnet.

1.7 6. Part III - Final Results (post-remediation, re-tested)

1.7.1 6.1 Final findings register

ID	Initial	Final	Status
F-01 No AMM slippage/min-out	High	-	**Resolved** (withdraw min-out + deposit ratio-drift guards)
F-02 Sequence-number race	High	-	**Resolved** (`submitWithSequenceRetry` on all writes)
F-03 Bridge reorg/miss window	High	-	**Resolved** (chain-aware lookback + overlap re-scan + de-dup)
F-06 Missing range validation	Medium	-	**Resolved** (finite/positive/range guards)
F-07 Stale pool snapshot	Medium	-	**Resolved** (pre-submit freshness re-fetch)
F-04 Float for money	Medium	**Low**	Bounded by on-ledger + client min-out; migration advised for mainnet
F-05 1-conf bridge-in	Medium	**Low**	Custodial relay is finality anchor; deepen confirmations for mainnet
F-08 No address checksum	Low	Low	Accepted; regex format-validated today
F-09 Fixed `LastLedgerSequence` offset	Low	Low	Accepted; retry loop re-derives a fresh window each attempt
F-10 No safe-integer drops check	Low	Low	Accepted; bounded by on-ledger amount limits
F-11 Limit order can cross AMM	Low	Low	Accepted; inherent to open order books
F-12 Custodial relay trust anchor	Info	Info	Accepted design property; documented

1.7.2 6.2 Severity totals

	Critical	High	Medium	Low	Info
Initial	0	3	4	4	1
Final	**0**	**0**	**0**	**5**	**1**

No High findings and no residual Medium findings remain. All remaining items are Low (defense-in-depth) or Informational, and are documented, accepted testnet risks.

1.7.3 6.3 Verification notes

- All three High remediations were verified by source re-review against the current HEAD (`submitWithSequenceRetry` wired into every write path; both AMM guards present and abort on breach; watcher lookback is chain-aware with overlapping re-scan and de-dup).
- Type-checking / build validation for this repository is performed with `pnpm run type-check` and `pnpm run build` (this report is a documentation-only change and introduces no code).

1.8 7. Positive Observations (Controls Done Well)

- **Non-custodial by construction** across all DeFi flows: `falcon_secret` decrypted and used in-browser (WASM), never transmitted; retry re-signs inside the client closure.
- **Instant swap enforces on-ledger bounds** (`SendMax`/`DeliverMin`) and re-quotes before signing.
- **AMM math is correct** (constant-product with fee, XLS-30) - `src/lib/swap/amm-math.ts`.
- **EVM bridge key hygiene** - CSPRNG `Wallet.createRandom()`, passkey-encrypted, decrypted per-transaction; **exact-amount** ERC-20 approval (not unlimited).
- **Internal Falcon node RPC intentionally withheld** from client config to avoid leaking a plaintext internal endpoint (`src/app/api/bridge/config/route.ts`).
- **Address/amount input validation** on order and liquidity paths; dust filtering on the order book.

1.9 8. Recommendations Before Mainnet / Real Value

1. Migrate money handling from `number` to integer drops / decimal IOU types end-to-end (F-04).
2. Increase bridge-in confirmations and add relay-side idempotency/attestation (F-05, F-12).
3. Add EIP-55 / XRPL checksum verification on bridge addresses (F-08).
4. Add a client-side safe-integer / upper-bound guard on drops math (F-10).
5. Consider a price cap when limit orders may cross the AMM (F-11).

End of report.