Phoenix5981 Security — Security Research Summary

Contact: willworachat@gmail.com | Report date: 2026-07-16 | Report type: **Audit Coverage Summary**

Target	Zynk Protocol — zynk-core + zynk-orbit
Platform	HackenProof
Engagement dates	2026-07-16 (single-session)
Reports submitted	0
Outcome	**NO FINDINGS — COMPREHENSIVE CLEAN RESULT**
Estimated coverage	100% of both in-scope programs read line-by-line (zynk-core 1537 lines, zynk-orbit 386 lines) at their respective program-scope-pinned commits; every instruction handler and account-context struct analyzed

Scope Tested

Two Solana/Anchor programs providing just-in-time liquidity and settlement infrastructure for cross-border payment companies. **zynk-core** (pinned `cc81cc7a`) — liquidity deployment, settlement, order lifecycle, beneficiary whitelisting, and a timelock/consensus governance system with four roles (admin/manager/guardian/attester). **zynk-orbit** (pinned `ebdbf987`, the repository's own HEAD) — a minimal LP fund-flow rail moving liquidity between LP wallets and the Zynk Operational Vault (ZOV), including the program's single stated unprivileged entry point, `deposit()`.

Methodology Applied

- P4 — read all 6 prior QuillAudits reports (Core v1 through v4.a, Orbit v1, spanning March 2025 to June 2026) in full before hunting, to establish a known-issues baseline and avoid re-reporting already-disclosed findings
- Commit-level scope discipline — zynk-core's program-scope pin was confirmed to be 5 commits behind the repository's HEAD; analysis was performed against the exact pinned commit via a dedicated git worktree, not the repository's current state
- Re-verification of previously-audited-and-fixed code paths against the live pinned source rather than trusting audit-report status alone — specifically the Ed25519 signature-verification syscall (`num_signatures` and offset-validation gaps from a prior audit round) and the `attest_order` message-construction / delimiter-injection guards
- Full Solana/Anchor toolchain installed and used to build and run a live, executable proof-of-concept (Solana CLI, Anchor CLI, yarn) against one candidate finding, rather than relying on static source analysis alone for severity determination
- Systematic constraint-by-constraint reading of every account-context struct, which is what surfaced the escape-hatch constraint that ultimately disproved the session's one Critical-shaped hypothesis

Coverage

Every instruction across both programs was analyzed. zynk-orbit's `deposit()` (the program's own stated sole unprivileged entry point) was confirmed soundly gated by PDA-existence-based whitelisting tied to the actual transaction signer. zynk-core's timelock/consensus role-separation machinery, the two `OrderTracker` PDA namespace patterns, `close_orders`'s batch-close remaining-accounts handling, and a hypothesized signature-amount-binding gap in `pull_and_create_order`'s transient-execution path were all carried to a definitive conclusion — the last of these via full proof-of-concept construction, described below.

Outcome / Findings Summary

No vulnerabilities of any severity were identified. Notable negative results, each independently verified rather than assumed:

- A candidate Critical-severity finding was identified during source review: `pull_and_create_order`'s transient-execution attester signature binds only the beneficiary/vault addresses, not the transfer amount, with no on-chain nonce or expiry. A full, executable Anchor/TypeScript proof-of-concept was built to test this empirically (requiring standing up the Solana CLI, Anchor CLI, and yarn toolchain). Running the PoC surfaced an account constraint that had been read past too quickly: the same fund movement is already fully reachable by the `manager` role through the protocol's ordinary non-transient path, with no attester involvement required at all. The missing amount-binding therefore does not grant any new fund-movement capability — it is an audit-trail/non-repudiation gap, not a fund-theft escalation, and does not clear this program's severity bar. This is disclosed here specifically because catching and correcting an overstated hypothesis before submission is itself a meaningful part of the research process.
- The Ed25519 signature-verification syscall used across the program's attestation and transient-execution paths was independently re-verified to correctly enforce a single-signature constraint and pin all instruction-introspection offset fields, closing a class of gap a prior audit round had found and fixed.
- `attest_order`'s signed-message construction was confirmed to bind the attested amount and to explicitly reject the delimiter sequence used to separate message fields within any caller-supplied string field, preventing message-ambiguity/injection.
- zynk-orbit's role addresses (ZOV/admin/manager) were confirmed to be immutable program constants rather than mutable configuration state, eliminating any on-chain role-hijack surface in that program entirely.

$0. No reports submitted to the platform.
