Phoenix5981 Security — Security Research Summary

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

Target	Telcoin — Peg Stability Vault (PSV)
Platform	HackenProof (DualDefense Audit / FlashBounty format)
Engagement dates	2026-07-16 (single-session)
Reports submitted	0
Outcome	**NO FINDINGS — COMPREHENSIVE CLEAN RESULT**
Estimated coverage	100% of the single-file in-scope surface (`programs/solana-psv/src/lib.rs`, 1873 lines) read line-by-line; deep vulnerability-hunting pass covered ~90% of the attack surface across 8 independent lead categories, with the remainder being trivial single-field admin-gated setters (pause/unpause/set_fees/set_rate_limits) of low residual risk

Scope Tested

Solana/Anchor smart contract enabling 1:1 swaps between a stable token and a "gem" token at parity minus configurable basis-point fees. Single-file scope, `programs/solana-psv/src/lib.rs`. Full instruction set reviewed: `initialize`, `buy_gem`, `sell_gem`, `preview_buy_gem`, `preview_sell_gem`, `set_fees`, `set_rate_limits`, `add_whitelist`, `remove_whitelist`, `pause`, `unpause`, `withdraw`, `close_vault`, `propose_admin`/`accept_admin`, `propose_treasury`/`accept_treasury`, `set_pauser`, `set_unpauser`.

Methodology Applied

- P1/P4/P12 — read program documentation (`README.md`, 420 lines) and the project's own 947-line `invariants.md` specification in full before touching source code
- H15 blockchain-infra / Solana-discriminator-bypass checklist — PDA seed derivation, account-substitution attacks, missing owner/signer checks, CPI reentrancy surface
- Token-2022 mixed-interface analysis — traced every `transfer_checked` CPI call site (9 total) to confirm no `remaining_accounts` mechanism exists anywhere in the program, closing out a hypothesized TransferHook-reentrancy vector via direct source proof rather than assumption
- Cross-referenced every candidate finding against the program's explicit reward-eligibility bar (stolen/locked funds must exceed 2% of protocol TVL or 1% of a single user's deposit; front-running attacks explicitly out of scope) before investing further analysis time
- Full constraint-by-constraint reading of every `#[derive(Accounts)]` context struct, not reliance on function names or comments alone

Coverage

Eight independent lead categories were carried to a definitive conclusion: Token-2022 TransferHook reentrancy (closed via source-level proof — no `remaining_accounts` usage exists anywhere in the file), `Withdraw` and `CloseVault` account-substitution (closed — requires trusted-signer cooperation, out of scope under the program's rogue-privileged-user exclusion), rate-limit function drift between the mutating and read-only implementations (closed — logically identical), whitelist fee-bypass across six sub-angles (cross-user PDA borrowing, cross-vault reuse, `Option::Some` forgery, fee-math rounding, missing-treasury-account handling, redistribution-by-a-legitimate-party — all closed), preview-vs-actual swap output drift (closed — the one real difference, unchecked reserves in preview, is explicitly documented and fails safe),

`close_vault`'s full handler body (closed — sweep ordering, canonical-ATA pinning, and precondition enforcement all correct), and the two-step admin/treasury role-transfer state machine (closed — cleanest of the eight, defensive zero-address/self/no-op guards plus complete pending-state cleanup). The program has already been through two prior professional audits (Trail of Bits — a fix is cited inline as `TOB-TELCO-04` — and Hacken, June 2026), consistent with the depth of hardening observed.

Outcome / Findings Summary

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

- The program's own transfer CPI calls were confirmed, by direct code inspection, to never forward extra accounts into any Token-2022 CPI — a mint with a `TransferHook` extension would simply cause every swap to revert, closing off a reentrancy hypothesis without requiring a live proof-of-concept build.
- The whitelist fee-exemption mechanism was confirmed to be address-derivation-scoped to the exact `(vault, user)` pair via PDA seeds tied to the actual transaction signer — no cross-user or cross-vault reuse of another party's fee-exempt status is possible.
- Ceiling-division fee math was confirmed to never round to a free (zero-fee) swap for any non-zero, non-whitelisted transaction — the smallest possible fee is always at least 1 unit, and amounts too small to produce positive output revert rather than granting a free swap.
- `close_vault`'s fund-sweep and precondition logic (pause required, empty whitelist required, dual has_one checks on admin and treasury) were confirmed correct against the project's own 947-line invariants specification.

$0. No reports submitted to the platform.