Phoenix5981 Security — Security Research Summary

Contact: willworachat@gmail.com | Report date: 2026-08-13 | Report type: Audit Coverage Summary

Target	Quantus Network — post-quantum Layer 1 blockchain (Substrate)
Platform	Immunefi — Audit Competition (competition live 12–25 August 2026; $20,000 USDC pool)
Engagement dates	2026-08-13 (single-session)
Reports submitted	0
Outcome	**NO FINDINGS — CLEAN RESULT ACROSS 10 ANALYSED SURFACES**
Estimated coverage	Approximately 25–30% of the ~300,000-line in-scope corpus, deliberately concentrated rather than uniform: the post-audit code delta was prioritised to near-complete coverage (vesting pallet, ML-DSA-65 signature scheme, QPoW consensus and difficulty adjustment, wormhole address derivation, and the leaf plus private-batch ZK circuit soundness chain), while heavily pre-audited and vendored regions received proportionally less time. Engagement was concluded on a reward-ceiling judgement, not on scope exhaustion.

Scope Tested

Quantus Network is a proof-of-work Layer 1 built on Polkadot's Substrate framework, designed to resist quantum adversaries by replacing elliptic-curve cryptography with lattice-based signatures. Five repositories were in scope, each analysed at its competition-pinned `audit-comp-ready` commit: the Substrate node and runtime (`3b243b87`, 296,294 lines), the ML-DSA/Dilithium signature and HD-wallet implementation (`94dfe6e6`, 59,267 lines), the Plonky2 zero-knowledge circuits (`2a6914ed`, 23,337 lines), the Poseidon2 hash implementation (`f885ad5d`, 3,584 lines), and the Flutter wallet and SDK (`d5fc4d65`, 3,338 lines).

The system's distinguishing components are ML-DSA-87 (NIST Level 5) with ML-DSA-65 as a recently added second transaction signature scheme; a Poseidon2 hash used both on-chain and in-circuit; a QPoW consensus layer; and an EIP-7503-style "zk-wormhole" providing private transfers, in which deposits to keyless addresses are later withdrawn by proving knowledge of a spend secret against an append-only Merkle tree of every transfer on the chain.

Methodology Applied

- **Prior-audit baseline before hunting.** Three Equilibrium Group audit reports shipped inside the repository were read first — a Proof-of-Work and Poseidon review (October 2025), a wormhole and cryptography review (March 2026), and a Substrate runtime and node review (May 2026) — to establish a known-issues baseline and avoid re-reporting disclosed findings. A fourth audit (Neodyme) was cited by the programme but not present in the repository.
- **Commit-history analysis to locate the post-audit delta.** The 628-commit history was mined to separate audited code from code written after the last audit round. Recent commits proved to be explicit audit-response bundles ("V12 audit round 2: assess all findings, fix the 32 confirmed"; "Harden FRAME primitives against audit findings"), which allowed effort to be redirected away from saturated regions and toward code still changing two days before the competition opened.
- **Recovery of deleted design documentation from version-control history.** A 278-line wormhole soundness design document removed by a recent commit was recovered from git history and read in full. It specified the security model and the on-chain invariant the project had previously enforced, and materially reframed the remainder of the engagement.
- **Hypothesis-driven review with explicit refutation.** Each surface was approached as a falsifiable claim ("the event scanner double-records vesting payouts", "the batch circuit's weaker dummy sentinel admits a forged exit") and carried to a definitive verified conclusion against source, rather than being marked inspected. Every hypothesis in this engagement was refuted on evidence.
- **Field-arithmetic soundness analysis of zero-knowledge constraints.** Circuit inequalities were not accepted at face value; the underlying Goldilocks field arithmetic was analysed for wrap-around soundness, including deriving the concrete bounds under which a range-check-based inequality can and cannot be subverted.
- **Cross-layer consistency checking.** Constraints were traced across the circuit, pallet, and runtime-extension boundaries — verifying, for example, that a value left unconstrained in one circuit is made unprovable by a constraint in another, and that on-chain checks and in-circuit checks agree on the same invariant.

Coverage

Ten distinct surfaces were carried to a definitive conclusion. In the runtime: the newly added pull-based vesting pallet, reviewed both for wormhole-leaf double-recording and for payout arithmetic, quantum alignment, and pot-solvency invariants; the high-security-account restriction model, including the full enumeration of origin-rewriting wrappers (`batch`, `as_derivative`, `as_recovered`, `dispatch_as`, multisig) against the call whitelist; and the ML-DSA-65 signature scheme addition, tested for scheme confusion, downgrade, and public-key-to-account binding failures.

In consensus: the rewritten QPoW algorithm, including whether the proof-of-work commits to block contents, whether the Poseidon2 byte-to-field encoding admits nonce collisions, and whether the difficulty retarget reads a stale timestamp or is manipulable by a block author. In the wormhole and zero-knowledge stack: the derivation separating wormhole deposit addresses from ordinary signing accounts, and a full soundness pass over the leaf circuit and the private-batch aggregation circuit covering recursive verifier-data pinning, deposit-amount binding, the per-leaf fee inequality, dummy-slot handling, exit-account grouping, and intra-batch nullifier uniqueness.

Outcome / Findings Summary

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

- **Zero-knowledge circuit soundness chain intact.** Recursive aggregation pins the inner circuit's verifier data as a constant rather than accepting it as a witness, closing the standard recursion-substitution break. A leaf's deposit amount is committed in the same hash preimage the chain uses, Merkle-verified to a root bound to a real block that the pallet independently re-checks. The per-leaf fee inequality was confirmed sound under field arithmetic: all amounts are 32-bit range-checked, placing both sides of the comparison far below the threshold at which a wrapped difference could satisfy the range check.
- **A genuine asymmetry between two circuits was found to be safe.** The aggregation circuit classifies padding slots using a strictly weaker condition than the leaf circuit does — precisely the shape of an exploitable gap. The discrepant case proved unprovable at the leaf layer, because a nonzero output forces a binding that would require a hash preimage of zero. The result holds, but it rests on a cross-circuit invariant: the aggregator's safety depends on a constraint defined in a different circuit, and would break silently if the leaf's condition were ever weakened. This is noted as a structural observation, not a vulnerability.
- **Value conservation in exit aggregation confirmed.** Exit-account grouping sums each account's amounts once and zeroes subsequent duplicate slots, and pairwise distinctness of live nullifiers is enforced inside the circuit, defeating the replay-one-deposit-across-many-slots inflation path. The same protection is independently duplicated in the pallet, which deliberately does not trust the circuit's constraints.
- **Wormhole and ordinary accounts are properly domain-separated.** Because the transfer tree records a leaf for every transfer on the chain, including to ordinary accounts, the system's soundness depends on ordinary-account leaves being unspendable. Deposit addresses are derived through a salted double-hash under one byte-to-field encoding, while ordinary accounts use a different construction with different salt, encoding, and padding; claiming an ordinary account as a wormhole account would require a hash preimage attack.
- **Signature-scheme confusion closed.** With two lattice signature schemes live simultaneously, verification derives the account from the embedded public key and rejects any mismatch against the claimed signer, and the two schemes cannot collide because of differing key lengths under a length-injective hash. Notably, account derivation applies no explicit scheme domain-separation and is safe only by virtue of that length difference.
- **Origin-rewriting privilege bypass closed.** A previously patched bypass class was re-examined after discovering that the patch's recursive resolver had subsequently been reverted. The surviving flat check nonetheless fails closed, because the restricted-account whitelist contains no call wrappers at all, and every path by which a third party can synthesise a restricted account's origin carries its own independent check.

Risk change identified for the vendor (not a submittable vulnerability): a recent commit removed the on-chain wormhole circuit-breaker — an invariant capping cumulative wormhole withdrawals against the pool of balances that could legitimately have been deposited — together with its supporting counters and a migration deleting them from storage. The stated rationale, that the mechanism rate-limited rather than stopped an attacker and rested on a lossy heuristic, is defensible. The consequence is nonetheless a real change in risk posture: the circuits are sound as written, so nothing is presently exploitable, but there is now no on-chain ceiling on wormhole minting, and any future regression in the proving system would mint unbacked native currency uncapped. This is recorded here as constructive feedback appropriate for disclosure to the project.

$0. No reports submitted to the platform. The engagement was concluded on an expected-value judgement rather than scope exhaustion: the codebase had passed four audits and two dedicated fix rounds in the two months preceding the competition, with defensive measures observably targeted at the specific attack constructions attempted here, while the reward ceiling was a $20,000 pool distributed by rank across a competitive field.