



FULL-STACK SECURITY REVIEW

SECURITY REVIEW —

· Track T3 — Treasury-yield pool + factory (APR reward engine)

PROJECT



TRACK / TIER

T3 · Full

DATE

1 May 2026

CHAINS

Ethereum + EVM

CODE VERSION

locked at kickoff (pending)

CLASSIFICATION

PUBLIC SAMPLE

Disclaimer. This report describes a time-boxed technical security review of the materials identified in the Scope of Review as they existed at the referenced commit/version during the review window. It is a point-in-time assessment: subsequent changes to code, configuration, infrastructure, or dependencies are not covered. No security review can identify all vulnerabilities. This report does not constitute a guarantee, warranty, or insurance of any kind, and it is not an endorsement of the project, its business model, or its tokens. It is not investment, legal, or financial advice. The review does not include formal verification, continuous monitoring, or exhaustive novel economic attack modeling unless expressly stated in scope. ■■■■■ may publish or share this report in full, unmodified form, including this disclaimer. Partial reproduction that omits findings, statuses, or this disclaimer is not authorized. Vari accepts no liability for decisions made by third parties on the basis of this report.

00 EXECUTIVE SUMMARY

TreasuryYield is a 1:1 deposit/withdraw ERC20 (■■■■■) with an APR-per-token reward engine covering a governor-controlled “■■■■■” token and the pool’s own asset. Rewards accrue via a global index not diluted by supply (fixed-APR model) and are paid from the PoolFactory’s aggregated balance. The core index math is internally consistent and uses an accrue-first pattern that neutralizes several would-be phantom-reward bugs. The main risks are an unbounded, manager-settable APR that can both drain the shared factory reserve and permanently brick a pool via overflow, a KYC/whitelist not enforced on cToken transfers or principal withdrawal, and broad timelock-free upgrade authority.

CRITICAL

0

HIGH

1

MEDIUM

2

LOW

3

INFO

1

Highest-priority items: **PROTO-T3-01** Unbounded manager APR rate: drains shared factory reserve and can permanently brick a pool.

01 SCOPE OF REVIEW

TreasuryYield pool, PoolFactory, reward-asset library, utilities. Upgradeable (beacon). [REDACTED]'s own code.

TRACK :: T3 · TIER :: Full · CHAINS :: Ethereum + EVM · CODE :: locked at kickoff (pending)

Out of scope: third-party dependencies (base libraries, external protocols, bridge/oracle endpoints), off-chain keeper/relayer infrastructure, and key management, assumed correct unless expressly listed. Test, mock and deploy-script files were read for context but are not the subject of the findings.

02 METHODOLOGY

AI-assisted broad analysis over the full source followed by targeted human review of business logic, value-moving paths, roles/permissions and higher-risk areas; cross-layer reconciliation; and live-state verification of the deployed settings (see §05). The review is manual + AI-assisted + static; it does not include formal verification or runtime fuzzing unless stated. Severity reflects exploitability + impact, never effort: **Critical** (direct path to loss of funds / full compromise) · **High** (exploitable, realistic conditions, material impact) · **Medium** (limited impact or unusual preconditions) · **Low** (defense-in-depth / best practice) · **Info** (observation).

03 FINDINGS

ID	SEVERITY	LAYER	FINDING
PROT0-T3-01	HIGH	contracts	Unbounded manager APR rate: drains shared factory reserve and can permanently brick a pool
PROT0-T3-02	MEDIUM	contracts	KYC/whitelist bypass: cToken transfers and principal withdraw() are not eligibility-gated
PROT0-T3-03	MEDIUM	config	Broad, timelock-free upgrade and admin authority
PROT0-T3-04	LOW	contracts	Reward payout truncation: full 18-dec amount marked withdrawn but truncated amount paid
PROT0-T3-05	LOW	contracts	Unhandled ERC20 assumptions (FoT / rebasing / >18 decimals) break 1:1 accounting
PROT0-T3-06	LOW	contracts	Hardcoded reward-asset indices; asset == [REDACTED] token collision breaks rate setters
PROT0-T3-07	INFO	contracts	Default-0 address index aliases unregistered assets to the [REDACTED] slot (latent footgun)

HIGHPROTO-
T3-01**Unbounded manager APR rate: drains shared factory reserve and can permanently brick a pool**

LOCATION

```
pool/TreasuryYield.sol: changeAprRate / _accrueRewardsIndex; libraries/RewardAsset.sol
```

DESCRIPTION

changeAprRate is onlyManager with no upper bound. Two impacts: (1) drain — because the index accrues $\text{rate} \times \text{dt}$ per token (undiluted) and payout is pulled from the shared PoolFactory balance, a manager can set an absurd APR so this pool's depositors accrue and withdraw far beyond intended, draining reserves that also back other pools; (2) permanent brick — a large enough rate makes the accrual arithmetic overflow under checked math, and since _accrueRewardsIndex runs inside every transfer/deposit/withdraw, the pool locks. Crucially both rate setters begin by accruing, which itself overflows once time passes, so the rate cannot be reset in place — only a beacon upgrade recovers it.

IMPACT

A compromised/malicious pool manager can drain cross-pool reward reserves and/or permanently lock all depositor principal in a pool.

RECOMMENDATION

Enforce a sane MAX_RATE bound in the rate setters (and in RewardAsset.insert); separate reward reserves per pool/manager; make accrual overflow-tolerant; add an admin reset path that does not first call the overflowing accrual.

Status: OPEN · Confidence: inference — code-read; overflow and cross-pool drain reasoned from control flow

MEDIUMPROTO-T3-
02**KYC/whitelist bypass: cToken transfers and principal withdraw() are not eligibility-gated**

LOCATION

```
pool/TreasuryYield.sol: withdraw / _update; TreasuryYieldInfo.onlyEligible
```

DESCRIPTION

onlyEligible is applied only to deposit() and reward withdrawal. ERC20 transfers go through _update with no eligibility check, and withdraw() has no onlyEligible. In a KYC pool, a whitelisted lender can transfer [REDACTED] to any non-whitelisted address, and that recipient can withdraw the underlying — bypassing the whitelist for holding and redeeming; a de-whitelisted address can still exit principal.

IMPACT

Compliance/KYC gating is not enforced on the transferable share token or on principal redemption.

RECOMMENDATION

Enforce onlyEligible in _update (both from and to when KYC required) and on withdraw(), or make [REDACTED] non-transferable in KYC pools; decide explicitly whether blacklisted users may exit principal.

Status: OPEN · Confidence: repro — no eligibility check exists on transfer/withdraw

MEDIUM**PROTO-T3-03 Broad, timelock-free upgrade and admin authority**

LOCATION

```
factory/PoolFactory.sol: changePoolBeacon / setLenderStatus / [REDACTED];  
pool/TreasuryYield.sol: updateTotalAccumulated / changeManager
```

DESCRIPTION

The factory owner controls the pool beacon and can therefore upgrade the implementation of ALL pools at once with no timelock, and solely controls whitelist, visibility, [REDACTED] rate and manager assignment.

updateTotalAccumulated overwrites a reward-reporting field with an arbitrary value with no invariant checks.

IMPACT

A single key can upgrade or reconfigure all pools and corrupt reward reporting; no on-chain delay for users to react.

RECOMMENDATION

Put the factory owner behind a timelock/multisig; delay and emit beacon changes; remove/constrain updateTotalAccumulated (it should be derived).

Status: OPEN · Confidence: repro — code-read of privileged functions

LOW**PROTO-T3-04 Reward payout truncation: full 18-dec amount marked withdrawn but truncated amount paid**

LOCATION

```
factory/PoolFactory.sol: withdrawReward (fromWei); pool/TreasuryYield.sol:  
updateRewardWithdrawals
```

DESCRIPTION

Pools record withdrawals in full 18-dec precision; the factory pays a decimal-truncated amount. The truncated remainder is recorded as withdrawn but never transferred, so the user is slightly under-paid each time (zero when reward decimals == 18).

IMPACT

Dust-level permanent under-payment of rewards, accumulating over many claims.

RECOMMENDATION

Record withdrawn amounts in the units actually transferred, or carry the sub-unit remainder forward.

Status: OPEN · Confidence: repro — rounding path clear

LOW**PROTO-T3-05 Unhandled ERC20 assumptions (FoT / rebasing / >18 decimals) break 1:1 accounting**

LOCATION

```
pool/TreasuryYield.sol: deposit/withdraw/cash; utils/utils.sol
```

DESCRIPTION

deposit() mints the requested amount before verifying received balance, so a fee-on-transfer asset credits more shares than assets received and late withdrawers hit insufficient liquidity; decimals>18 underflows in the wei helpers and the cached decimals is stale if the asset changes it.

IMPACT

Non-standard assets cause share/asset mismatch or revert; principal accounting can break for the last withdrawers.

RECOMMENDATION

Measure received amount via balance delta; restrict supported assets to standard ≤18-dec non-FoT/non-rebasing tokens and document it.

Status: OPEN · Confidence: repro — code-read

LOWPROTO-T3-
06**Hardcoded reward-asset indices; asset == [REDACTED] token collision breaks rate setters**

LOCATION

`pool/TreasuryYield.sol: __init / changeAprRate / [REDACTED]`

DESCRIPTION

__init inserts [REDACTED] at index 0 and the pool asset at index 1 and the rate setters hardcode those indices; the second insert's success is not checked. A pool whose asset equals the [REDACTED] token leaves a single-element array and changeAprRate then reverts out-of-bounds.

IMPACT

A misconfigured pool (asset == [REDACTED]) has a broken APR lever; brittle to future reward-asset changes.

RECOMMENDATION

Look up reward assets by address rather than positional index; require the second insert succeeds and asset != [REDACTED] in __init.

Status: OPEN · Confidence: repro — reasoned from RewardAsset.insert

INFOPROTO-T3-
07**Default-0 address index aliases unregistered assets to the [REDACTED] slot (latent footgun)**

LOCATION

`pool/TreasuryYield.sol: _accumulativeRewardOf; libraries/RewardAsset.sol`

DESCRIPTION

An unregistered asset's index defaults to 0 (the [REDACTED] slot); currently neutralized by the accrue-first ordering and zero stored index, but any refactor removing that ordering would expose phantom reward withdrawals for arbitrary assets.

IMPACT

No current exploit; a fragile invariant that could regress into a fund-draining bug.

RECOMMENDATION

Require the asset is registered in the reward paths; avoid relying on default-0 aliasing.

Status: OPEN · Confidence: repro — traced; currently non-exploitable

04 WHAT'S SOLID

- Accrue-first pattern: rate-change and withdraw functions accrue before computing rewards, neutralizing the default-index aliasing and preventing phantom rewards on unregistered assets.
- Reward accounting invariant (withdrawable = accumulative – withdrawn, accumulative computed live) holds; CEI ordering means an ERC777-style re-entry sees zero withdrawable.
- nonReentrant on deposit/withdraw; rate changes are non-retroactive; the 1:1 fixed share model avoids first-depositor inflation; storage gaps present on the info contracts.
- Reward and principal solvency separated: APR rewards are paid from factory balance, not pool cash, so principal withdrawal is not cannibalized by reward accrual (absent fee-on-transfer).

05 LIVE ON-CHAIN VERIFICATION

Pending deployed addresses. Governance/config findings must reflect the deployed state, not the source. The items below are queued to be read live (via Etherscan V2 proxy `eth_call`) once [REDACTED] provides the deployed addresses + chains per system; each will be reported PASS / ACTION / VERIFY / INFO. Until then they are marked VERIFY.

- PoolFactory.owner() and whether it is a timelock/multisig; the UpgradeableBeacon and its owner (upgrades all pools).
- Per-pool manager addresses and trust level — they hold the unbounded APR lever.
- isPublic / kycRequired per pool and the live whitelist for KYC pools.
- Live [REDACTED] and APR rates per pool vs intended scale; factory on-chain balances of each reward asset vs total outstanding accrued-minus-withdrawn (reward solvency).
- [REDACTED] decimals and standard-ness (no FoT/rebasing, ≤ 18 dec); confirm no pool was created with `asset == [REDACTED]` token.

06 ACTION POINTS

PRIORITY	ID	ACTION	STATUS
P0	PR0T0-T3-01	Unbounded manager APR rate: drains shared factory reserve and can permanently brick a pool	◦ OPEN
P1	PR0T0-T3-02	KYC/whitelist bypass: cToken transfers and principal withdraw() are not eligibility-gated	◦ OPEN
P1	PR0T0-T3-03	Broad, timelock-free upgrade and admin authority	◦ OPEN
P2	PR0T0-T3-04	Reward payout truncation: full 18-dec amount marked withdrawn but truncated amount paid	◦ OPEN
P2	PR0T0-T3-05	Unhandled ERC20 assumptions (FoT / rebasing / >18 decimals) break 1:1 accounting	◦ OPEN
P2	PR0T0-T3-06	Hardcoded reward-asset indices; <code>asset == [REDACTED]</code> token collision breaks rate setters	◦ OPEN
P2	PR0T0-T3-07	Default-0 address index aliases unregistered assets to the [REDACTED] slot (latent footgun)	◦ OPEN

P0 = before deploy / now · P1 = soon · P2 = hardening. Status column is filled at the re-review round (report goes to v1.1: ◦ OPEN · ✓ FIXED · ~ ACKNOWLEDGED · ✗ NOT FIXED).

07 RESIDUAL RISK

After the recommended fixes, residual risk concentrates in the privileged roles (owner/manager/strategist/governor keys) and in operational posture — key custody, multisig thresholds, timelocks, and monitoring — which code alone cannot guarantee. Because this system holds user funds, a companion Economic Modeling Addendum accompanies this report and models the solvency and adverse-market mechanics summarized above. This is a point-in-time review; re-verify after any change to code, configuration, or deployed settings.

VARI — EVERY LAYER. ONE REVIEW.

██████ · T3 ██████ · 1 May 2026

CLASSIFICATION :: PUBLIC SAMPLE — CLIENT IDENTITY REDACTED · CONTACT :: TELEGRAM @va_rinder ·
VARIEREVIEW.COM