ClarityAlliance

Zest Strategy Vault Security Review

Conducted by:
Kristian Apostolov

September 2nd, 2026

Contents

1. About Clarity Alliance

Clarity Alliance is a team of expert whitehat hackers specialising in securing protocols on Stacks.

They have disclosed vulnerabilities that have saved millions in live TVL and conducted thorough reviews for some of the largest projects across the Stacks ecosystem.

Learn more about Clarity Alliance at clarityalliance.org.

2. Disclaimer

This report is not, and should not be considered, an endorsement or disapproval of any project, team, product, or asset, nor does it reflect on their economics, value, business model, or legal compliance. It does not provide any warranty regarding the absolute bug-free nature or functionality of the analyzed technology.

Nothing in this report should be used to make investment or participation decisions. It does not constitute investment advice. Instead, it reflects an extensive assessment process intended to help clients improve code quality and reduce the inherent risks associated with cryptographic tokens and blockchain systems.

Blockchain technology presents ongoing and significant risk, and each company or individual remains responsible for their own due diligence and security posture. Clarity Alliance aims to reduce attack vectors and technological uncertainty but does not guarantee the security or performance of any system we review.

All assessment services are subject to dependencies and active development. Your access to and use of any services, reports, or materials is at your sole risk on an as-is and as-available basis.

Cryptographic tokens are emergent technologies with high technical uncertainty, and assessment results may include false positives, false negatives, or other unpredictable outcomes. Smart contracts may depend on multiple external parties, remain vulnerable to internal or external exploitation, and may carry elevated risks if owner privileges remain active. Accordingly, Clarity Alliance does not guarantee the explicit security of any audited smart contract, regardless of the reported verdict.

3. Introduction

A time-boxed security review of the Zest Strategy Vault, where Clarity Alliance reviewed the scope and provided insights on improving the protocol.

4. About Zest

What is Zest Protocol?

Zest Protocol is the leading Bitcoin lending protocol. BTC holders earn yield on their Bitcoin and borrow stablecoins against it.

Zest Protocol is the largest lending protocol on Bitcoin L2s, with $100M+ peak TVL and over two years of mainnet history. Built by one of the longest-running team in Bitcoin DeFi, building on Bitcoin since 2020.

Introducing Stacks Vaults

Stacks Vaults are automated yield strategies built on top of Zest Protocol's lending markets on Stacks. Users deposit a single asset into a vault, and the vault executes a yield strategy on their behalf: no position management, no rebalancing, no juggling markets.

Stacks Vaults mark the evolution of Zest Protocol from a lending market into yield infrastructure. Every yield source on Stacks becomes a strategy that can be automated and offered as a single-deposit product.

The first vault is the levered Bitcoin Staking vault, built around stBTC, the liquid Bitcoin Staking token from Stacking DAO.

The Levered Bitcoin Staking Vault

Users deposit BTC, sBTC, or stBTC into the vault. The vault uses stBTC as collateral on Zest Protocol's lending market to borrow sBTC, stakes the borrowed sBTC into stBTC, and repeats the process to build a leveraged staking position.

The result is compounded yield on top of the base Bitcoin Staking rewards, while the user holds a single vault position.

Target yield: 6 to 8% APY, derived from Bitcoin Staking on Stacks.

Deposit. Deposit BTC on Bitcoin L1, or sBTC or stBTC on Stacks. BTC and sBTC deposits are converted into stBTC on entry. The user receives a vault position representing their share of the strategy.

Automated leverage. The vault posts stBTC as collateral on Zest Protocol, borrows sBTC against it, stakes the borrowed sBTC into stBTC, and re-deposits. The loop runs until the target leverage is reached. Zest Protocol manages the loop automatically.

Yield source. All yield originates from Stacks Bitcoin Staking rewards, carried into the vault through stBTC's auto-compounding ratio. The strategy is profitable when the stBTC staking yield exceeds the sBTC borrow rate. No emissions, no incentives, no external yield sources.

Continuous monitoring. Zest Protocol monitors the position continuously: collateral ratio, borrow rates, and staking yield. The vault adjusts leverage as market conditions change.

Withdrawals. Users can withdraw at any time. The vault unwinds the required share of the position: repaying borrowed sBTC and releasing stBTC, and returns the underlying assets to the user's wallet.

Stacks Vaults are fully non-custodial. The vault contract can only execute strategy actions on Zest Protocol's lending markets. It cannot move funds anywhere else, and only the user can withdraw their position.

Key features:

- **Non-custodial** – The vault contract is restricted to strategy actions on Zest Protocol. Only the user can withdraw their funds
- **One deposit, one position** – Deposit BTC, sBTC, or stBTC and hold a single position while the strategy runs itself
- **Pure Bitcoin yield** – Target 6 to 8% APY derived entirely from Bitcoin Staking on Stacks, with no token emissions
- **Built on live markets** – The vault runs on Zest Protocol's lending markets: two years in production, over a thousand liquidations processed without bad debt
- **First of many strategies** – STX-based strategies, stablecoin and credit-based strategies, and structured yield products will follow on the same foundation. In the future, external curators will be able to manage their own strategies on Stacks Vaults

Stacks Vaults are separate from Bitcoin Collateral Vaults, Zest Protocol's upcoming flagship product that allows users to borrow against native BTC on any chain (e.g. Ethereum).

Risks

Leveraged staking carries risks that unlevered holding does not:

- **Rate risk.** If the sBTC borrow rate rises above the stBTC staking yield, the loop becomes unprofitable until the vault deleverages
- **Liquidation risk.** The vault maintains conservative collateral ratios, but a sharp move in the stBTC/sBTC ratio could trigger deleveraging
- **Smart contract risk.** The vault inherits the risk of the underlying contracts: Zest Protocol's lending market and Stacking DAO's stBTC

Stacks Vaults will launch alongside stBTC, before Stacks Bitcoin Staking goes live. More details at zestprotocol.com.

5. Risk Classification

Severity	Impact: High	Impact: Medium	Impact: Low
Likelihood: High	Critical	High	Medium
Likelihood: Medium	High	Medium	Low
Likelihood: Low	Medium	Low	Low

5.1. Impact

- High - leads to a significant material loss of assets in the protocol or significantly harms a group of users.
- Medium - only a small amount of funds can be lost (such as leakage of value) or a core functionality of the protocol is affected.
- Low - can lead to any kind of unexpected behavior with some of the protocol's functionalities that's not so critical.

5.2. Likelihood

- High - attack path is possible with reasonable assumptions that mimic on-chain conditions, and the cost of the attack is relatively low compared to the amount of funds that can be stolen or lost.
- Medium - only a conditionally incentivized attack vector, but still relatively likely.
- Low - has too many or too unlikely assumptions or requires a significant stake by the attacker with little or no incentive.

5.3. Action required for severity levels

- Critical - Must fix as soon as possible (if already deployed)
- High - Must fix (before deployment if not already deployed)
- Medium - Should fix
- Low - Could fix

6. Security Assessment Summary

Scope

Repository: `Zest-Protocol/zest-core/contracts/strategy-vault`

The scope includes all the smart contracts in the repository.

Contracts

- `zv-traits.clar`
- `zvstBTC.clar`
- `zv-state-stbtc-0.clar`
- `zv-ops-stbtc-0.clar`
- `zv-engine-stbtc-0.clar`

Initial Commit Reviewed

`77795979b5f007c5f87f6ff3378a0d2af561afec`

Final Commit Reviewed

`8b982f31289bce3159e47451ab98acdfdefb6111`

The final commit reviewed is located in the production repository.

7. Executive Summary

Over the course of the security review, Kristian Apostolov engaged with Zest Protocol to review Zest. In this period of time a total of **7** issues were uncovered.

Protocol Summary

Protocol Name	Zest
Repository	Zest-Protocol/zest-core/contracts/strategy-vault
Report Date	September 2nd, 2026

Findings Count

Severity	Amount
Medium	7
Total Findings	**7**

Summary of Findings

ID	Title	Severity	Status
[M-01]	Fee Settlement Erases Pending Reserve Allocations	Medium	Resolved
[M-02]	Funded Claims Lack Aggregate Liability And Custody Reservation	Medium	Resolved
[M-03]	stBTC Close Ignores Actual Outputs and Delayed Withdrawals	Medium	Resolved
[M-04]	Guardian Pause Leaves Risk-Increasing Strategy Actions Active	Medium	Resolved
[M-05]	Fee-Aware Close Can Report Success With Residual Debt	Medium	Resolved
[M-06]	Lossy Unwinds Leave Vault NAV Overstated	Medium	Resolved
[M-07]	Guardian Pause Can Disable Every Vault-Funded Unwind Path	Medium	Resolved

8. Findings

8.1. Medium Findings

[M-01] Fee Settlement Erases Pending Reserve Allocations

Location:
`contracts/strategy-vault/stbtc-0/zv-state-stbtc-0.clar#L333-L346`

Description:
`zv-state-stbtc-0::settle-fees` is expected to move each pending allocation into its corresponding permanent accounting bucket while reducing shareholder assets by the same combined amount. However, the stBTC state contract adds `pending-fees` to `accrued-fees` and then clears `pending-rf` without crediting it to `reserve-fund`.

For example, settling 10 units of fees and 5 units of reserve reduces `total-assets` by 15, but only the 10-unit fee is recorded. The 5-unit reserve remains in custody but is no longer represented in accounting, effectively becoming unowned value and reducing the recorded protection available to absorb future losses.

Recommendation: In `zv-state-stbtc-0::settle-fees`, add `pending-rf` to `reserve-fund` before clearing it, while preserving the existing fee accrual behavior, `total-assets` reduction, authorization checks, and atomic rollback semantics.

```
(var-set reserve-fund (+ (var-get reserve-fund) prf))
```

[M-02] Funded Claims Lack Aggregate Liability And Custody Reservation

Location: `contracts/strategy-vault/stbtc-0/zv-engine-stbtc-0.clar#L218-L263`

Description : `zv-engine-stbtc-0::fund-claim` is expected to convert burned shares into a funded liability backed by collateral reserved for the specified recipient. However, the function subtracts the gross claim assets from `total-assets` and records the claim amount without (1) increasing any aggregate liability or (2) restricting `transfer-collateral-to-ops`. After the cooldown expires, any caller can make the claim uncancellable while the same custody remains available for deployment. If those assets later become impaired or illiquid, the payout may fail, and the contracts do not define any priority between the claimant and remaining shareholders.

Recommendation: Track unpaid net claims as an aggregate liability, reserve matching custody to prevent deployment, and reduce the liability only after `redeem` completes a recipient-verified payout.

```
(var-set funded-claim-liability
  (+ (var-get funded-claim-liability) assets-net))
(asserts! (>= balance (+ amount
  (var-get funded-claim-liability))) ERR-INSUFFICIENT-LIQUIDITY)
(var-set funded-claim-liability
  (- (var-get funded-claim-liability) assets-net))
```

[M-03] stBTC Close Ignores Actual Outputs and Delayed Withdrawals

Location:
`contracts/strategy-vault/stbtc-0/zv-ops-stbtc-0.clar#L88-L124`

Description:
`close-position` should unwind a strategy position based on the stBTC actually redeemed, the sBTC actually repaid, and any delayed withdrawal that remains pending. Instead, it relies on a caller-supplied `repay-amount`, withdraws the entire ops stBTC balance, ignores the results of the withdrawal and repayment, and does not record any continuation state. As a result, a fee or delayed conversion can cause the transaction to revert while debt continues to accrue. If redemption produces surplus sBTC, that value remains in ops without being returned to permanent state or re lected in accounting.

Recommendation:
Deprecate `close-position`. Repay indexed debt using `repay-only`, require the debt to be zero, remove collateral, and then transfer any residual sBTC to permanent state.

[M-04] Guardian Pause Leaves Risk-Increasing Strategy Actions Active

Location: `contracts/strategy-vault/stbtc-0/zv-state-stbtc-0.clar#L453-L480`

Description : `zv-state-stbtc-0::pause-vault` provides the guardian with an immediate emergency brake; however, it only clears `vault-enabled`, while strategy entry points check `trading-enabled`. As a result, the trader can still call `open-position`, `borrow-more`, and the debt-increasing `remove-collateral` during the response window. For example, if the trader key is compromised, an attacker could borrow additional sBTC immediately after the guardian pauses the vault, increasing the vault's exposure before the owner disables trading. Although health checks reduce the risk of insolvency, they do not enforce the containment boundary the guardian pause is intended to provide.

Recommendation: Clear both pause flags and require `trading-enabled` in `remove-collateral`. Keep collateral addition, repayment, close, and cancellation available to support recovery.

```
(var-set vault-enabled false)
(var-set trading-enabled false)
```

[M-05] Fee-Aware Close Can Report Success With Residual Debt

Location: `contracts/strategy-vault/stbtc-0/zv-ops-stbtc-0.clar#L107-L149`

Description : `close-position` should either repay the requested debt in full or revert. Instead, it burns `repay-amount` stBTC, repays only `min(repay-amount, sbtc-out)`, removes the caller-supplied collateral amount, and returns `(ok true)` without verifying that the remaining debt is zero or enforcing a minimum conversion rate. In a reproduced position with 5 stBTC collateral and 3 sBTC debt at a 0.95 withdrawal ratio, removing 1 stBTC succeeds after repaying 2.85 sBTC and leaves 0.15 sBTC of residual debt. Attempting to remove all 5 stBTC returns the market `ERR-UNHEALTHY` and rolls back. The added regression test does not disprove this issue because it validates ops balances rather than checking for residual debt in the market.

Recommendation: Make `close-position` full-close-or-revert by enforcing a minimum unwind rate and requiring zero remaining indexed debt before allowing collateral removal. If partial deleveraging is intended, expose it via a separate function and emit an event that reports the actual repayment and remaining debt.

[M-06] Lossy Unwinds Leave Vault NAV Overstated

Location: `contracts/strategy-vault/stbtc-0/zv-ops-stbtc-0.clar#L119-L149` & `contracts/strategy-vault/stbtc-0/zv-engine-stbtc-0.clar#L364-L482`

Description : A successful unwind should reconcile any realized conversion loss before pricing subsequent deposits or redemptions. Instead, `close-position` can burn more stBTC than the debt value it removes while leaving `total-assets` unchanged. In the reproduced 0.95-ratio close, residual debt remains while NAV stays at its pre-close value. Reporting the resulting 1.5% loss immediately triggers `ERR-REWARD-TOO-LARGE` under the default 10 bps cap; additionally, the 7 bps deviation limit and daily update window prevent timely correction via smaller reports. As a result, early redemptions can shift the loss onto remaining holders. Claim reservation, empty ops balances, and eventual keeper reports do not correct NAV during this interval.

Recommendation: Update `total-assets` atomically based on each unwind's verified net profit or loss. Derive net P&L from a tracked cost basis or a verified position-value delta; until this is implemented, keep deposits and redemptions disabled whenever an unwind loss remains unreconciled.

[M-07] Guardian Pause Can Disable Every Vault-Funded Unwind Path

Location: `contracts/strategy-vault/stbtc-0/zv-ops-stbtc-0.clar#L273-L289`

Description : A guardian pause should prevent risk expansion while still preserving a way to repay debt using vault-owned assets. However, the new `trading-enabled` assertion disables `unstack-sbtc-from-state`, and `close-position` requires collateral removal and re-deposits any positive sBTC residual into StackingDAO. In a reproduced position with 5 stBTC collateral and 3 sBTC debt at a 1.1 ratio, disabling StackingDAO deposits and then pausing causes `close-position` to return `ERR-DEPOSITS-DISABLED`, while the fallback unstack path returns `ERR-PAUSED`. Smaller partial closes do not help because any surplus still triggers the disabled deposit. Requiring the owner to re-enable deposits weakens the emergency boundary during a time-sensitive liquidation scenario.

Recommendation: Add a pause-safe, atomic `unstack-and-repay` path that applies the actual sBTC output directly to debt repayment without collateral removal or a new StackingDAO deposit. Preserve claim custody, report the actual repayment amount, and send any irreducible surplus to permanent sBTC custody.