ClarityAlliance

STACKING DAO (UPGRADE) SECURITY REVIEW

Conducted by:
KRISTIAN APOSTOLOV, ALIN BARBATEI (ABA)

JUNE 3RD, 2025

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Stacking DAO 4
5. Risk Classification 5
    5.1. Impact 5
    5.2. Likelihood 5
    5.3. Action required for severity levels 5
6. Security Assessment Summary 6
7. Executive Summary 7
8. Summary of Findings 8
8.1. Critical Findings 9
[C-01] Tracking Contract Vulnerable to Draining Due to Double-Counted Rewards 9
8.2. Low Findings 11
[L-01] Differentiate Allowed Staking Contracts in Rewards 11
[L-02] Reserve Contract for Supported Positions Lacks Sanity Checks 12
8.3. QA Findings 13
[QA-01] Incorrect Event Entry Name 13
[QA-02] Rewards Interval Length Should Be a Divisor of PoX Reward Cycle Length 14
[QA-03] Outdated Tracking Data Default Reserve Contract 15
[QA-04] Document the Correct Internal Contract Version 16
[QA-05] Outdated Behavioral Comments in the DAO Core Contracts 17
[QA-06] Migration of Third-Party Contracts May Permanently Block stSTXBTC Tokens 18
[QA-07] Deactivated Positions Particularities 19
[QA-08] Rewards Contract Optimization Opportunity 21
[QA-09] Potential Confusion with Commission Traits 22

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.

ClarityAlliance
Security Review

Stacking DAO (Upgrade)

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Stacking DAO 4
5. Risk Classification 5
   5.1. Impact 5
   5.2. Likelihood 5
   5.3. Action required for severity levels 5
6. Security Assessment Summary 6
7. Executive Summary 7
8. Summary of Findings 8
8.1. Critical Findings 9
[C-01] Tracking Contract Vulnerable to Draining Due to Double-Counted Rewards 9
8.2. Low Findings 11
[L-01] Differentiate Allowed Staking Contracts in Rewards 11
[L-02] Reserve Contract for Supported Positions Lacks Sanity Checks 12
8.3. QA Findings 13
[QA-01] Incorrect Event Entry Name 13
[QA-02] Rewards Interval Length Should Be a Divisor of PoX Reward Cycle Length 14
[QA-03] Outdated Tracking Data Default Reserve Contract 15
[QA-04] Document the Correct Internal Contract Version 16
[QA-05] Outdated Behavioral Comments in the DAO Core Contracts 17
[QA-06] Migration of Third-Party Contracts May Permanently Block stSTXBTC Tokens 18
[QA-07] Deactivated Positions Particularities 19
[QA-08] Rewards Contract Optimization Opportunity 21
[QA-09] Potential Confusion with Commission Traits 22

2. Disclaimer

This report is not, nor should be considered, an “endorsement” or “disapproval” of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any “product” or “asset” created by any team or project that contracts Clarity Alliance to perform a security assessment.

This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk. Clarity Alliance’s position is that each company and individual are responsible for their own due diligence and continuous security. Clarity Alliance’s goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree
to analyze.

The assessment services provided by Clarity Alliance are subject to dependencies and under continuing development. You agree that your access and/or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis.

Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives, and other unpredictable results. The services may access, and depend upon, multiple layers of third parties. Notice that smart contracts deployed on the blockchain are not resistant from internal/external exploit. Notice that active smart contract owner privileges constitute an elevated impact to any smart contract’s safety and security. Therefore, Clarity Alliance does not guarantee the explicit security of the audited smart contract, regardless of the verdict.

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Stacking DAO 4
5. Risk Classification 5
    5.1. Impact 5
    5.2. Likelihood 5
    5.3. Action required for severity levels 5
6. Security Assessment Summary 6
7. Executive Summary 7
8. Summary of Findings 8
8.1. Critical Findings 9
[C-01] Tracking Contract Vulnerable to Draining Due to Double-Counted Rewards 9
8.2. Low Findings 11
[L-01] Differentiate Allowed Staking Contracts in Rewards 11
[L-02] Reserve Contract for Supported Positions Lacks Sanity Checks 12
8.3. QA Findings 13
[QA-01] Incorrect Event Entry Name 13
[QA-02] Rewards Interval Length Should Be a Divisor of PoX Reward Cycle Length 14
[QA-03] Outdated Tracking Data Default Reserve Contract 15
[QA-04] Document the Correct Internal Contract Version 16
[QA-05] Outdated Behavioral Comments in the DAO Core Contracts 17
[QA-06] Migration of Third-Party Contracts May Permanently Block stSTXBTC Tokens 18
[QA-07] Deactivated Positions Particularities 19
[QA-08] Rewards Contract Optimization Opportunity 21
[QA-09] Potential Confusion with Commission Traits 22

3. Introduction

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

4. About Stacking DAO

Stacking DAO is the STX Stacking infrastructure powerhouse for the most prominent Bitcoin L2. The protocol currently offers 3 STX Stacking services:

- **stSTX**: A liquid representation of stacked STX that accrues in value in STX as Stacking rewards are auto-compounded daily. It can also be used across DeFi to earn additional yield and points.

- **stSTXbtc**: A liquid stacking token backed 1-to-1 with STX, and holders receive sBTC rewards daily that can be claimed at any moment. stSTXbtc can also be used across Stacks dApps.

- **Native Stacking**: Delegate STX and earn BTC rewards with zero fees while STX are locked during the two-week Stacking cycles.

ClarityAlliance
Security Review

Stacking DAO (Upgrade)

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Stacking DAO 4
5. Risk Classification 5
    5.1. Impact 5
    5.2. Likelihood 5
    5.3. Action required for severity levels 5
6. Security Assessment Summary 6
7. Executive Summary 7
8. Summary of Findings 8
8.1. Critical Findings 9
[C-01] Tracking Contract Vulnerable to Draining Due to Double-Counted Rewards 9
8.2. Low Findings 11
[L-01] Differentiate Allowed Staking Contracts in Rewards 11
[L-02] Reserve Contract for Supported Positions Lacks Sanity Checks 12
8.3. QA Findings 13
[QA-01] Incorrect Event Entry Name 13
[QA-02] Rewards Interval Length Should Be a Divisor of PoX Reward Cycle Length 14
[QA-03] Outdated Tracking Data Default Reserve Contract 15
[QA-04] Document the Correct Internal Contract Version 16
[QA-05] Outdated Behavioral Comments in the DAO Core Contracts 17
[QA-06] Migration of Third-Party Contracts May Permanently Block stSTXBTC Tokens 18
[QA-07] Deactivated Positions Particularities 19
[QA-08] Rewards Contract Optimization Opportunity 21
[QA-09] Potential Confusion with Commission Traits 22

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

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Stacking DAO 4
5. Risk Classification 5
   5.1. Impact 5
   5.2. Likelihood 5
   5.3. Action required for severity levels 5
6. Security Assessment Summary 6
7. Executive Summary 7
8. Summary of Findings 8
8.1. Critical Findings 9
[C-01] Tracking Contract Vulnerable to Draining Due to Double-Counted Rewards 9
8.2. Low Findings 11
[L-01] Differentiate Allowed Staking Contracts in Rewards 11
[L-02] Reserve Contract for Supported Positions Lacks Sanity Checks 12
8.3. QA Findings 13
[QA-01] Incorrect Event Entry Name 13
[QA-02] Rewards Interval Length Should Be a Divisor of PoX Reward Cycle Length 14
[QA-03] Outdated Tracking Data Default Reserve Contract 15
[QA-04] Document the Correct Internal Contract Version 16
[QA-05] Outdated Behavioral Comments in the DAO Core Contracts 17
[QA-06] Migration of Third-Party Contracts May Permanently Block stSTXBTC Tokens 18
[QA-07] Deactivated Positions Particularities 19
[QA-08] Rewards Contract Optimization Opportunity 21
[QA-09] Potential Confusion with Commission Traits 22

6. Security Assessment Summary

Scope

The following contracts were in the scope of the security review:

- `rewards-v5.clar`
- `ststxbtc-migration-v1.clar`

Additionally, all updates to the Clarity smart contracts in the repository included in the following pull request were reviewed:

https://github.com/StackingDAO/StackingDAO/pull/717

Initial Commit Reviewed:
db8f78b458cfaec3e6cbfb4d898743d753a907fb

Final Commit After Remediations:
36bd090cf955b9884bfa3817b61047a81f24260a

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Stacking DAO 4
5. Risk Classification 5
   5.1. Impact 5
   5.2. Likelihood 5
   5.3. Action required for severity levels 5
6. Security Assessment Summary 6
7. Executive Summary 7
8. Summary of Findings 8
8.1. Critical Findings 9
[C-01] Tracking Contract Vulnerable to Draining Due to Double-Counted Rewards 9
8.2. Low Findings 11
[L-01] Differentiate Allowed Staking Contracts in Rewards 11
[L-02] Reserve Contract for Supported Positions Lacks Sanity Checks 12
8.3. QA Findings 13
[QA-01] Incorrect Event Entry Name 13
[QA-02] Rewards Interval Length Should Be a Divisor of PoX Reward Cycle Length 14
[QA-03] Outdated Tracking Data Default Reserve Contract 15
[QA-04] Document the Correct Internal Contract Version 16
[QA-05] Outdated Behavioral Comments in the DAO Core Contracts 17
[QA-06] Migration of Third-Party Contracts May Permanently Block stSTXBTC Tokens 18
[QA-07] Deactivated Positions Particularities 19
[QA-08] Rewards Contract Optimization Opportunity 21
[QA-09] Potential Confusion with Commission Traits 22

7. Executive Summary

Over the course of the security review, Kristian Apostolov, Alin Barbatei (ABA) engaged with - to review Stacking DAO. In this period of time a total of **12** issues were uncovered.

Protocol Summary

Protocol Name	Stacking DAO
Date	June 3rd, 2025

Findings Count

Severity	Amount
Critical	1
Low	2
QA	9
Total Findings	**12**

ClarityAlliance
Security Review

Stacking DAO (Upgrade)

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Stacking DAO 4
5. Risk Classification 5
   5.1. Impact 5
   5.2. Likelihood 5
   5.3. Action required for severity levels 5
6. Security Assessment Summary 6
7. Executive Summary 7
8. Summary of Findings 8
8.1. Critical Findings 9
[C-01] Tracking Contract Vulnerable to Draining Due to Double-Counted Rewards 9
8.2. Low Findings 11
[L-01] Differentiate Allowed Staking Contracts in Rewards 11
[L-02] Reserve Contract for Supported Positions Lacks Sanity Checks 12
8.3. QA Findings 13
[QA-01] Incorrect Event Entry Name 13
[QA-02] Rewards Interval Length Should Be a Divisor of PoX Reward Cycle Length 14
[QA-03] Outdated Tracking Data Default Reserve Contract 15
[QA-04] Document the Correct Internal Contract Version 16
[QA-05] Outdated Behavioral Comments in the DAO Core Contracts 17
[QA-06] Migration of Third-Party Contracts May Permanently Block stSTXBTC Tokens 18
[QA-07] Deactivated Positions Particularities 19
[QA-08] Rewards Contract Optimization Opportunity 21
[QA-09] Potential Confusion with Commission Traits 22

Summary of Findings

ID	Title	Severity	Status
[C-01]	Tracking Contract Vulnerable to Draining Due to Double-Counted Rewards	Critical	Resolved
[L-01]	Differentiate Allowed Staking Contracts in Rewards	Low	Resolved
[L-02]	Reserve Contract for Supported Positions Lacks Sanity Checks	Low	Acknowledged
[QA-01]	Incorrect Event Entry Name	QA	Resolved
[QA-02]	Rewards Interval Length Should Be a Divisor of PoX Reward Cycle Length	QA	Resolved
[QA-03]	Outdated Tracking Data Default Reserve Contract	QA	Resolved
[QA-04]	Document the Correct Internal Contract Version	QA	Resolved
[QA-05]	Outdated Behavioral Comments in the DAO Core Contracts	QA	Resolved
[QA-06]	Migration of Third-Party Contracts May Permanently Block stSTXBTC Tokens	QA	Acknowledged
[QA-07]	Deactivated Positions Particularities	QA	Acknowledged
[QA-08]	Rewards Contract Optimization Opportunity	QA	Resolved
[QA-09]	Potential Confusion with Commission Traits	QA	Resolved

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Stacking DAO 4
5. Risk Classification 5
   5.1. Impact 5
   5.2. Likelihood 5
   5.3. Action required for severity levels 5
6. Security Assessment Summary 6
7. Executive Summary 7
8. Summary of Findings 8
8.1. Critical Findings 9
[C-01] Tracking Contract Vulnerable to Draining Due to Double-Counted Rewards 9
8.2. Low Findings 11
[L-01] Differentiate Allowed Staking Contracts in Rewards 11
[L-02] Reserve Contract for Supported Positions Lacks Sanity Checks 12
8.3. QA Findings 13
[QA-01] Incorrect Event Entry Name 13
[QA-02] Rewards Interval Length Should Be a Divisor of PoX Reward Cycle Length 14
[QA-03] Outdated Tracking Data Default Reserve Contract 15
[QA-04] Document the Correct Internal Contract Version 16
[QA-05] Outdated Behavioral Comments in the DAO Core Contracts 17
[QA-06] Migration of Third-Party Contracts May Permanently Block stSTXBTC Tokens 18
[QA-07] Deactivated Positions Particularities 19
[QA-08] Rewards Contract Optimization Opportunity 21
[QA-09] Potential Confusion with Commission Traits 22

8. Findings

8.1. Critical Findings

[C-01] Tracking Contract Vulnerable to Draining Due to Double-Counted Rewards

Description

In the new `ststxbtc-tracking-v2` contract, when a new underlying `stSTXbtc-v2` transfer occurs, the rewards are not immediately transferred. Instead, they are saved and added to the user's claim.

A critical flaw in the reward-saving logic results in users' rewards being double-counted, leading to inflated reward amounts.

When rewards for a user are saved through the `ststxbtc-tracking-v2::save-pending-rewards` call, pending rewards are gathered via a `get-pending-rewards` call, and existing saved rewards are also retrieved using a `get-saved-rewards` call.

```
(define-public (save-pending-rewards (holder principal) (position principal))
  (let (
    (pending-rewards (unwrap-panic (get-pending-rewards holder position)))
    (existing-rewards (get-saved-rewards holder position))
  )
    (if (>= pending-rewards u1)
      (begin
        (map-set saved-rewards { holder: holder, position: position }
          (+ existing-rewards pending-rewards))
```

If pending rewards exist, they are added to the `saved-rewards` map.

The problem arises because the `get-pending-rewards` function again adds the saved rewards to the pending amount.

```
(define-read-only (get-pending-rewards (holder principal) (position principal))
  (let (
    ;; ... code ...
    (rewards-saved (get-saved-rewards holder position))
    ;; ... code ...
  )
    (if is-holder-position
       (ok u0)
       (ok (+ rewards rewards-saved))
    )
```

As a result, users' rewards are over-inflated, allowing holders to extract more funds than intended from the contract. This vulnerability enables an attacker to completely drain any rewards allocated to the `ststxbtc-tracking-v2` contract.

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Stacking DAO 4
5. Risk Classification 5
   - 5.1. Impact 5
   - 5.2. Likelihood 5
   - 5.3. Action required for severity levels 5
6. Security Assessment Summary 6
7. Executive Summary 7
8. Summary of Findings 8
8.1. Critical Findings 9
- [C-01] Tracking Contract Vulnerable to Draining Due to Double-Counted Rewards 9

8.2. Low Findings 11
- [L-01] Differentiate Allowed Staking Contracts in Rewards 11
- [L-02] Reserve Contract for Supported Positions Lacks Sanity Checks 12

8.3. QA Findings 13
- [QA-01] Incorrect Event Entry Name 13
- [QA-02] Rewards Interval Length Should Be a Divisor of PoX Reward Cycle Length 14
- [QA-03] Outdated Tracking Data Default Reserve Contract 15
- [QA-04] Document the Correct Internal Contract Version 16
- [QA-05] Outdated Behavioral Comments in the DAO Core Contracts 17
- [QA-06] Migration of Third-Party Contracts May Permanently Block stSTXBTC Tokens 18
- [QA-07] Deactivated Positions Particularities 19
- [QA-08] Rewards Contract Optimization Opportunity 21
- [QA-09] Potential Confusion with Commission Traits 22

Recommendation

There are two potential solutions to address this issue:

1. Retain the current implementation of `get-pending-rewards` the `save-pending-rewards` function to update the `saved-rewards` only if the difference between `pending-rewards` and `existing-rewards` greater than 0. This adjustment is necessary because `pending-rewards` will always be positive if there is a saved amount. The focus should be on newly generated pending reward amounts. With this solution, save the `pending-rewards` in the `saved-rewards`.

2. Remove the addition of saved rewards from the `get-pending-rewards` function and incorporate the saved rewards into the amount in the `claim-pending-rewards` function.

Note that the second approach is simpler but leaves `ststxbtc-tracking-v2` the contract with only the `get-saved-rewards` function as a helper. There are no `-many` or `-iter` type helpers, and from a third-party perspective, the `claim-pending-rewards` function will not return all pending rewards, only those stored `pending` as without those stored as `saved`.

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Stacking DAO 4
5. Risk Classification 5
   5.1. Impact 5
   5.2. Likelihood 5
   5.3. Action required for severity levels 5
6. Security Assessment Summary 6
7. Executive Summary 7
8. Summary of Findings 8
8.1. Critical Findings 9
[C-01] Tracking Contract Vulnerable to Draining Due to Double-Counted Rewards 9
8.2. Low Findings 11
[L-01] Differentiate Allowed Staking Contracts in Rewards 11
[L-02] Reserve Contract for Supported Positions Lacks Sanity Checks 12
8.3. QA Findings 13
[QA-01] Incorrect Event Entry Name 13
[QA-02] Rewards Interval Length Should Be a Divisor of PoX Reward Cycle Length 14
[QA-03] Outdated Tracking Data Default Reserve Contract 15
[QA-04] Document the Correct Internal Contract Version 16
[QA-05] Outdated Behavioral Comments in the DAO Core Contracts 17
[QA-06] Migration of Third-Party Contracts May Permanently Block stSTXBTC Tokens 18
[QA-07] Deactivated Positions Particularities 19
[QA-08] Rewards Contract Optimization Opportunity 21
[QA-09] Potential Confusion with Commission Traits 22

8.2. Low Findings

[L-01] Differentiate Allowed Staking Contracts in Rewards

Description

The `rewards-v5::process-rewards` function processes rewards by taking four traits as input. The contracts associated with these traits are verified to be protocol-approved, and the stSTX and stSTXBTC commission contracts are checked to ensure they are correctly passed on a 1:1 basis. However, this verification does not extend to the reserve and staking contracts.

Although these contracts are expected to adhere to the trait implementation, this may not be sufficient for the `staking-contract`. Officially, StackingDAO uses `staking-v0` as input. However, the development of a `staking-v1` contract suggests that this could change in the future.

If a v1 contract is deployed, it might be possible to distribute rewards without actual staking by bypassing v1 and using v0 instead. Additionally, while a trait is used, the `staking-trait` is quite basic, featuring only a single `add-rewards` function:

```
(add-rewards (uint uint) (response uint uint))
```

Within the StackingDAO ecosystem, there are several non-staking-related contracts that implement an `add-rewards` function. Currently, none of these match the staking trait:

- `ststxbtc-tracking.clar` : `(add-rewards (uint) (response bool uint))`
- Any reward contract: `(add-rewards (principal uint) (response bool uint))`

If, in the future, a contract is added to the protocol with the same function prototype, the rewards contract could become vulnerable to contract confusion.

Recommendation

Specify the exact staking contract address to be used in the `rewards-v5` contract and ensure it is explicitly checked in the `process-rewards` function.

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Stacking DAO 4
5. Risk Classification 5
   5.1. Impact 5
   5.2. Likelihood 5
   5.3. Action required for severity levels 5
6. Security Assessment Summary 6
7. Executive Summary 7
8. Summary of Findings 8
8.1. Critical Findings 9
[C-01] Tracking Contract Vulnerable to Draining Due to Double-Counted Rewards 9
8.2. Low Findings 11
[L-01] Differentiate Allowed Staking Contracts in Rewards 11
[L-02] Reserve Contract for Supported Positions Lacks Sanity Checks 12
8.3. QA Findings 13
[QA-01] Incorrect Event Entry Name 13
[QA-02] Rewards Interval Length Should Be a Divisor of PoX Reward Cycle Length 14
[QA-03] Outdated Tracking Data Default Reserve Contract 15
[QA-04] Document the Correct Internal Contract Version 16
[QA-05] Outdated Behavioral Comments in the DAO Core Contracts 17
[QA-06] Migration of Third-Party Contracts May Permanently Block stSTXBTC Tokens 18
[QA-07] Deactivated Positions Particularities 19
[QA-08] Rewards Contract Optimization Opportunity 21
[QA-09] Potential Confusion with Commission Traits 22

[L-02] Reserve Contract for Supported Positions Lacks Sanity Checks

Description

When the protocol opts to support specific third-party positions, it invokes the `ststxbtc-tracking-v2::set-supported-positions` function, which includes a reserve principal among its parameters.

Currently, there are no validations performed on the principal, allowing it to be any arbitrary value. Incorrectly set contracts could affect the availability of position rewards.

Recommendation

Consider implementing checks for the `reserve` principal, such as:

- Creating a specific trait for position reserves
- Ensuring it is a protocol-owned address

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Stacking DAO 4
5. Risk Classification 5
   5.1. Impact 5
   5.2. Likelihood 5
   5.3. Action required for severity levels 5
6. Security Assessment Summary 6
7. Executive Summary 7
8. Summary of Findings 8
8.1. Critical Findings 9
[C-01] Tracking Contract Vulnerable to Draining Due to Double-Counted Rewards 9
8.2. Low Findings 11
[L-01] Differentiate Allowed Staking Contracts in Rewards 11
[L-02] Reserve Contract for Supported Positions Lacks Sanity Checks 12
8.3. QA Findings 13
[QA-01] Incorrect Event Entry Name 13
[QA-02] Rewards Interval Length Should Be a Divisor of PoX Reward Cycle Length 14
[QA-03] Outdated Tracking Data Default Reserve Contract 15
[QA-04] Document the Correct Internal Contract Version 16
[QA-05] Outdated Behavioral Comments in the DAO Core Contracts 17
[QA-06] Migration of Third-Party Contracts May Permanently Block stSTXBTC Tokens 18
[QA-07] Deactivated Positions Particularities 19
[QA-08] Rewards Contract Optimization Opportunity 21
[QA-09] Potential Confusion with Commission Traits 22

8.3. QA Findings

[QA-01] Incorrect Event Entry Name

Description

In the `rewards-v5::add-rewards-sbtc` contract, the `print` command emits the `sbtc-amount` incorrectly labeled as `stx-amount`. View the code here.

This mislabeling can cause minor confusion off-chain.

Recommendation

Update the `stx-amount` entry to `sbtc-amount` in the `print` command.

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Stacking DAO 4
5. Risk Classification 5
   5.1. Impact 5
   5.2. Likelihood 5
   5.3. Action required for severity levels 5
6. Security Assessment Summary 6
7. Executive Summary 7
8. Summary of Findings 8
8.1. Critical Findings 9
[C-01] Tracking Contract Vulnerable to Draining Due to Double-Counted Rewards 9
8.2. Low Findings 11
[L-01] Differentiate Allowed Staking Contracts in Rewards 11
[L-02] Reserve Contract for Supported Positions Lacks Sanity Checks 12
8.3. QA Findings 13
[QA-01] Incorrect Event Entry Name 13
[QA-02] Rewards Interval Length Should Be a Divisor of PoX Reward Cycle Length 14
[QA-03] Outdated Tracking Data Default Reserve Contract 15
[QA-04] Document the Correct Internal Contract Version 16
[QA-05] Outdated Behavioral Comments in the DAO Core Contracts 17
[QA-06] Migration of Third-Party Contracts May Permanently Block stSTXBTC Tokens 18
[QA-07] Deactivated Positions Particularities 19
[QA-08] Rewards Contract Optimization Opportunity 21
[QA-09] Potential Confusion with Commission Traits 22

[QA-02] Rewards Interval Length Should Be a Divisor of PoX Reward Cycle Length

Description

In the `rewards-v5` contracts, rewards are distributed using an interval system, where the reward amount is allocated per interval:

```
(total-intervals (/ (get-reward-cycle-length)
  (var-get rewards-interval-length)))
```

Intervals are determined by dividing the total cycle length by the reward distribution interval length. With the current defaults, this calculation is `2100/70 = 30`.

The `rewards-interval-length` variable can be modified through `set-rewards-interval-length` call to any arbitrary value.

If the rewards interval length is set to a value that is not a divisor of the PoX epoch length, users will experience periods within the reward cycles where they receive no benefits.

For example, changing the rewards cycle to 85 results in a total interval length of 24.705. This means that the period between the 2040th block and the 2100th block, when a new PoX cycle begins, yields no rewards for users. Consequently, users must wait for a new epoch to start and an additional 85 blocks, totaling a wait of 145 blocks instead of 85 for the last interval.

Recommendation

When calling `rewards-v5: :set-rewards-interval-length`, ensure the interval length perfectly divides the PoX reward cycle length at that time.

An on-chain check might overly constrain the interval value. Therefore, acknowledge this issue and ensure, off-chain, that the specified length is either a complete divisor of the PoX length or has the smallest possible division remainder.

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Stacking DAO 4
5. Risk Classification 5
   5.1. Impact 5
   5.2. Likelihood 5
   5.3. Action required for severity levels 5
6. Security Assessment Summary 6
7. Executive Summary 7
8. Summary of Findings 8
8.1. Critical Findings 9
[C-01] Tracking Contract Vulnerable to Draining Due to Double-Counted Rewards 9
8.2. Low Findings 11
[L-01] Differentiate Allowed Staking Contracts in Rewards 11
[L-02] Reserve Contract for Supported Positions Lacks Sanity Checks 12
8.3. QA Findings 13
[QA-01] Incorrect Event Entry Name 13
[QA-02] Rewards Interval Length Should Be a Divisor of PoX Reward Cycle Length 14
[QA-03] Outdated Tracking Data Default Reserve Contract 15
[QA-04] Document the Correct Internal Contract Version 16
[QA-05] Outdated Behavioral Comments in the DAO Core Contracts 17
[QA-06] Migration of Third-Party Contracts May Permanently Block stSTXBTC Tokens 18
[QA-07] Deactivated Positions Particularities 19
[QA-08] Rewards Contract Optimization Opportunity 21
[QA-09] Potential Confusion with Commission Traits 22

[QA-03] Outdated Tracking Data Default Reserve Contract

Description

When retrieving an unsupported position using the `ststxbtc-tracking-data-v2::get-supported-positions` call, it incorrectly defaults to the previous version of the tracking data contract.

```
(define-read-only (get-supported-positions (position principal))
  (default-to
    {
      active: false,
      total: u0,
      reserve: .ststxbtc-tracking-data,
      deactivated-cumm-reward: u0
    }
    (map-get? supported-positions position)
```

The reserve value is not utilized until the position is activated, and at that point, a new reserve is employed.

Changing the default reserve contract to use the v2 version does not affect protocol functionality but enhances codebase uniformity.

Recommendation

In the `get-supported-positions` functions of the `ststxbtc-tracking-data-v2` contract, update the default reserve from `•ststxbtc-tracking-data` to the v2 variant.

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Stacking DAO 4
5. Risk Classification 5
   5.1. Impact 5
   5.2. Likelihood 5
   5.3. Action required for severity levels 5
6. Security Assessment Summary 6
7. Executive Summary 7
8. Summary of Findings 8
8.1. Critical Findings 9
[C-01] Tracking Contract Vulnerable to Draining Due to Double-Counted Rewards 9
8.2. Low Findings 11
[L-01] Differentiate Allowed Staking Contracts in Rewards 11
[L-02] Reserve Contract for Supported Positions Lacks Sanity Checks 12
8.3. QA Findings 13
[QA-01] Incorrect Event Entry Name 13
[QA-02] Rewards Interval Length Should Be a Divisor of PoX Reward Cycle Length 14
[QA-03] Outdated Tracking Data Default Reserve Contract 15
[QA-04] Document the Correct Internal Contract Version 16
[QA-05] Outdated Behavioral Comments in the DAO Core Contracts 17
[QA-06] Migration of Third-Party Contracts May Permanently Block stSTXBTC Tokens 18
[QA-07] Deactivated Positions Particularities 19
[QA-08] Rewards Contract Optimization Opportunity 21
[QA-09] Potential Confusion with Commission Traits 22

[QA-04] Document the Correct Internal Contract Version

Description

In the newly added batch of contracts, some internal versions do not match the actual contract versions:

- The `ststxbtc-token-v2` is missing an internal version entirely; it should be set to v2.
- `stacking-dao-core-v5` is incorrectly noted as version 3. It should be updated to version 5.
- `direct-helpers-v4` is listed as version 1; it should be changed to version 4.
- `ststxbtc-tracking-data-v2` is recorded as version 1; it should be updated to version 2.
- `stacking-dao-core-btc-v2` is noted as version 1; it should be changed to version 2.
- `rewards-v5` is incorrectly listed as version 4; it should be updated to version 5.
- `swap-ststx-ststxbtc-v2` is recorded as version 1; it should be changed to version 2.

Recommendation

In all mentioned cases, update to the correct version.

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Stacking DAO 4
5. Risk Classification 5
   5.1. Impact 5
   5.2. Likelihood 5
   5.3. Action required for severity levels 5
6. Security Assessment Summary 6
7. Executive Summary 7
8. Summary of Findings 8
8.1. Critical Findings 9
[C-01] Tracking Contract Vulnerable to Draining Due to Double-Counted Rewards 9
8.2. Low Findings 11
[L-01] Differentiate Allowed Staking Contracts in Rewards 11
[L-02] Reserve Contract for Supported Positions Lacks Sanity Checks 12
8.3. QA Findings 13
[QA-01] Incorrect Event Entry Name 13
[QA-02] Rewards Interval Length Should Be a Divisor of PoX Reward Cycle Length 14
[QA-03] Outdated Tracking Data Default Reserve Contract 15
[QA-04] Document the Correct Internal Contract Version 16
[QA-05] Outdated Behavioral Comments in the DAO Core Contracts 17
[QA-06] Migration of Third-Party Contracts May Permanently Block stSTXBTC Tokens 18
[QA-07] Deactivated Positions Particularities 19
[QA-08] Rewards Contract Optimization Opportunity 21
[QA-09] Potential Confusion with Commission Traits 22

[QA-05] Outdated Behavioral Comments in the DAO Core Contracts

Description

The `stacking-dao-core-v5` and `stacking-dao-core-btc-v2` contracts have undergone modifications in their withdrawal logic:

- Previously, tokens were retained in the contract until they were fully withdrawn.
- Now, tokens are burned immediately upon initiating a withdrawal.
- This change prevents DAO contracts from holding LSTs and inadvertently receiving rewards.

Although these changes have been implemented, the internal documentation of the functions still refers to the old behavior of holding tokens and burning them only upon withdrawal finalization.

The comment issues for both contracts are as follows:

- `init-withdraw:` : The outdated comment states, “tokens are transferred to this contract, and are burned on the actual withdrawal” whereas tokens are now burned within this contract.
- `withdraw:` : The outdated comment reads, “The NFT and <stSTX/stSTXbtc> tokens will be burned and the user will receive STX tokens,” even though the tokens have already been burned during `init-withdraw` .

Additionally, there is a typo in the `stacking-dao-core-v5: :deposit` event print, where the entry name `stxstx-amount` is mistakenly used instead of `ststx-amount` .

Recommendation

Update the outdated or incorrect wording as mentioned.

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Stacking DAO 4
5. Risk Classification 5
   5.1. Impact 5
   5.2. Likelihood 5
   5.3. Action required for severity levels 5
6. Security Assessment Summary 6
7. Executive Summary 7
8. Summary of Findings 8
8.1. Critical Findings 9
[C-01] Tracking Contract Vulnerable to Draining Due to Double-Counted Rewards 9
8.2. Low Findings 11
[L-01] Differentiate Allowed Staking Contracts in Rewards 11
[L-02] Reserve Contract for Supported Positions Lacks Sanity Checks 12
8.3. QA Findings 13
[QA-01] Incorrect Event Entry Name 13
[QA-02] Rewards Interval Length Should Be a Divisor of PoX Reward Cycle Length 14
[QA-03] Outdated Tracking Data Default Reserve Contract 15
[QA-04] Document the Correct Internal Contract Version 16
[QA-05] Outdated Behavioral Comments in the DAO Core Contracts 17
[QA-06] Migration of Third-Party Contracts May Permanently Block stSTXBTC Tokens 18
[QA-07] Deactivated Positions Particularities 19
[QA-08] Rewards Contract Optimization Opportunity 21
[QA-09] Potential Confusion with Commission Traits 22

[QA-06] Migration of Third-Party Contracts May Permanently Block stSTXBTC Tokens

Description

The `ststxbtc-migration-v1::migrate-ststxbtc` contract can be executed by the protocol team using any arbitrary principal addresses, including those belonging to contract principals.

Migrating contract principals without first confirming that these contracts have the capability to transfer the v2 version of the stSTXBTC token may result in tokens and yield being permanently blocked.

Recommendation

Thoroughly inspect each principal migrated by the team to ensure that tokens can be transferred and that no integration issues arise with the associated team.

For instance, while the currently supported Zest Position Reserve allows the transfer of any type of tokens, the StackingDAO must still verify with Zest to ensure there are no integration issues within the Zest ecosystem.

Since this verification is conducted off-chain, acknowledge this issue for the purpose of this report.

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Stacking DAO 4
5. Risk Classification 5
   5.1. Impact 5
   5.2. Likelihood 5
   5.3. Action required for severity levels 5
6. Security Assessment Summary 6
7. Executive Summary 7
8. Summary of Findings 8
8.1. Critical Findings 9
[C-01] Tracking Contract Vulnerable to Draining Due to Double-Counted Rewards 9
8.2. Low Findings 11
[L-01] Differentiate Allowed Staking Contracts in Rewards 11
[L-02] Reserve Contract for Supported Positions Lacks Sanity Checks 12
8.3. QA Findings 13
[QA-01] Incorrect Event Entry Name 13
[QA-02] Rewards Interval Length Should Be a Divisor of PoX Reward Cycle Length 14
[QA-03] Outdated Tracking Data Default Reserve Contract 15
[QA-04] Document the Correct Internal Contract Version 16
[QA-05] Outdated Behavioral Comments in the DAO Core Contracts 17
[QA-06] Migration of Third-Party Contracts May Permanently Block stSTXBTC Tokens 18
[QA-07] Deactivated Positions Particularities 19
[QA-08] Rewards Contract Optimization Opportunity 21
[QA-09] Potential Confusion with Commission Traits 22

[QA-07] Deactivated Positions Particularities

Description

Within the tracking logic, once a position is marked as supported, it becomes eligible for rewards and can be refreshed using the `ststxbtc-tracking-v2::refresh-position` call.

The team has the ability to both activate and deactivate a position through `ststxbtc-tracking-v2::set-supported-positions` function.

There are several important aspects to consider regarding deactivated positions:

1. Once a position is deactivated, any `stSTXBTC` held in that position contract will not generate rewards.
2. If a position is deactivated, any holders who have not refreshed their positions will lose out on rewards accumulated up to that point.
3. A position that has been activated and then deactivated should not be reactivated, although it can be under unusual conditions.

This restriction is enforced by the following check:

```
;; Cannot activate position if it was already active previously
(asserts! (is-eq (get total supported-position) u0) (err ERR_POSITION_USED))
```

Here, the `total` amount refers to the total tracked balance of the specific position token.

However, this check can be circumvented if, before a position is deactivated, all position holders coordinate to transfer their external token balances and then call `refresh-position`, effectively reducing the total `amount` to 0.

This scenario would require an extraordinary level of coordination among position holders, and even if executed, StackingDAO would need to decide to re-support the position, which is unlikely to be justified.

1. Supported positions that hold balances of tokens from other supported positions do not earn rewards.

For instance, if tokens from a supported position, such as ZEST, are held in the position contract of another supported position, like ALEX, the ALEX position contract will not receive any yield from holding ZEST. This is a side effect of intentionally omitting rewards from position contracts.

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Stacking DAO 4
5. Risk Classification 5
   - 5.1. Impact 5
   - 5.2. Likelihood 5
   - 5.3. Action required for severity levels 5
6. Security Assessment Summary 6
7. Executive Summary 7
8. Summary of Findings 8
8.1. Critical Findings 9
- [C-01] Tracking Contract Vulnerable to Draining Due to Double-Counted Rewards 9

8.2. Low Findings 11
- [L-01] Differentiate Allowed Staking Contracts in Rewards 11
- [L-02] Reserve Contract for Supported Positions Lacks Sanity Checks 12

8.3. QA Findings 13
- [QA-01] Incorrect Event Entry Name 13
- [QA-02] Rewards Interval Length Should Be a Divisor of PoX Reward Cycle Length 14
- [QA-03] Outdated Tracking Data Default Reserve Contract 15
- [QA-04] Document the Correct Internal Contract Version 16
- [QA-05] Outdated Behavioral Comments in the DAO Core Contracts 17
- [QA-06] Migration of Third-Party Contracts May Permanently Block stSTXBTC Tokens 18
- [QA-07] Deactivated Positions Particularities 19
- [QA-08] Rewards Contract Optimization Opportunity 21
- [QA-09] Potential Confusion with Commission Traits 22

Recommendation

For all the points mentioned, it is crucial to clearly document this behavior for users to understand. The effort required to address any of these particularities far exceeds any potential benefits.

For scenario `(2)`, consider creating a helper function, `refresh-position-many`, to allow the protocol or users to bulk sync positions before any deactivation occurs.

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Stacking DAO 4
5. Risk Classification 5
   5.1. Impact 5
   5.2. Likelihood 5
   5.3. Action required for severity levels 5
6. Security Assessment Summary 6
7. Executive Summary 7
8. Summary of Findings 8
8.1. Critical Findings 9
[C-01] Tracking Contract Vulnerable to Draining Due to Double-Counted Rewards 9
8.2. Low Findings 11
[L-01] Differentiate Allowed Staking Contracts in Rewards 11
[L-02] Reserve Contract for Supported Positions Lacks Sanity Checks 12
8.3. QA Findings 13
[QA-01] Incorrect Event Entry Name 13
[QA-02] Rewards Interval Length Should Be a Divisor of PoX Reward Cycle Length 14
[QA-03] Outdated Tracking Data Default Reserve Contract 15
[QA-04] Document the Correct Internal Contract Version 16
[QA-05] Outdated Behavioral Comments in the DAO Core Contracts 17
[QA-06] Migration of Third-Party Contracts May Permanently Block stSTXBTC Tokens 18
[QA-07] Deactivated Positions Particularities 19
[QA-08] Rewards Contract Optimization Opportunity 21
[QA-09] Potential Confusion with Commission Traits 22

[QA-08] Rewards Contract Optimization Opportunity

Description

In the `rewards-v5` contract, certain implementation details can be modified to reduce execution costs and enhance contract readability.

In the `add-rewards-sbtc` and `add-rewards` functions, the final `print` command retrieves the current PoX cycle using `cycle: (get-pox-cycle)` instead of utilizing the already declared `(rewards-cycle (get-pox-cycle))`. It is recommended to use `rewards-cycle` for the `cycle` print entry.

Within the `process-rewards` function, multiple `contract-of` calls are duplicated. For instance, `(contract-of commission-ststx-contract)` is called twice. It should be stored in a `let` variable and reused. The same approach should be applied to the `reserve` and `commission-ststxbtc-contract` contracts.

Additionally, in `process-rewards`, the values `(get protocol-stx rewards-info)` and `(get commission-stx rewards-info)` are each retrieved four times. These should be placed in `let` variables for reuse.

Recommendation

Implement the suggested changes.

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Stacking DAO 4
5. Risk Classification 5
   5.1. Impact 5
   5.2. Likelihood 5
   5.3. Action required for severity levels 5
6. Security Assessment Summary 6
7. Executive Summary 7
8. Summary of Findings 8
8.1. Critical Findings 9
[C-01] Tracking Contract Vulnerable to Draining Due to Double-Counted Rewards 9
8.2. Low Findings 11
[L-01] Differentiate Allowed Staking Contracts in Rewards 11
[L-02] Reserve Contract for Supported Positions Lacks Sanity Checks 12
8.3. QA Findings 13
[QA-01] Incorrect Event Entry Name 13
[QA-02] Rewards Interval Length Should Be a Divisor of PoX Reward Cycle Length 14
[QA-03] Outdated Tracking Data Default Reserve Contract 15
[QA-04] Document the Correct Internal Contract Version 16
[QA-05] Outdated Behavioral Comments in the DAO Core Contracts 17
[QA-06] Migration of Third-Party Contracts May Permanently Block stSTXBTC Tokens 18
[QA-07] Deactivated Positions Particularities 19
[QA-08] Rewards Contract Optimization Opportunity 21
[QA-09] Potential Confusion with Commission Traits 22

[QA-09] Potential Confusion with Commission Traits

Description

Within the StackingDAO codebase, there are instances where users need to pass a commission trait with specific permissions.

Some of these instances require distinguishing between the stSTX commission contract (`.commission-v2`) and the stSTXBTC commission contract (`.commission-btc-v1`).

In the `stacking-dao-core-v5` and `stacking-dao-core-btc-v2` contracts, when performing operations such as `deposit` or `withdraw`, users might mistakenly pass the stSTXBTC commission instead of the stSTX commission contract.

This error will result in an ambiguous `u1` revert (indicating insufficient tokens for transfer) when attempting to transfer sBTC from the DAO contract to the commission contract.

Recommendation

As was done in the `rewards-v5` contract, implement specific differentiation for commission contracts in the `stacking-dao-core-v5` and `stacking-dao-core-btc-v2` contracts.