Quantstamp

Fragmetric Liquid Restaking Program

Executive Summary

This audit report was prepared by Quantstamp, the leader in blockchain security.

Type	Solana Restaking Protocol
Timeline	2025-01-27 through 2025-02-14
Language	Rust
Methods	Architecture Review, Unit Testing, Functional Testing, Computer-Aided Verification, Manual Review
Specification	Docs | Fragmetric
Source Code	• https://github.com/fragmetric-labs/fragmetric-contracts #4ad0252
Auditors	• Michael Boyle Auditing Engineer • Cameron Biniamow Auditing Engineer • Mostafa Yassin Auditing Engineer

Documentation quality	High
Test quality	High
Total Findings	3 Acknowledged: 3
High severity findings	0
Medium severity findings	0
Low severity findings	1 Acknowledged: 1
Undetermined severity findings	0
Informational findings	2 Acknowledged: 2

Summary of Findings

Fragmetric is a decentralized liquid restaking program deployed on the Solana blockchain. Fragmetric enables users to deposit SOL or supported tokens into the protocol and receive minted receipt tokens in return. Once SOL or supported tokens are deposited into the Fragmetric fund account, user funds are staked into various restaking vaults to generate rewards.

Quantstamp was tasked with a time-boxed review of Fragmetric's second iteration of their restaking contracts. Specifically, the `fund`, `normalization`, `pricing`, `staking`, `restaking`, and `swap` modules were reviewed to identify deviations from the project specification, potential vulnerabilities, and proper integrations with external protocols. All external protocols and the `reward` module were considered out of scope for this review. Due to the constrained time for this review and the large size of the codebase, the audit team could not perform a comprehensive audit of the codebase. Therefore, uncaught bugs or vulnerabilities may remain in the code.

The codebase submitted for review was high-quality and included a thorough test suite with sufficient coverage metrics. However, due to the protocol's iterative development process, portions of the external documentation were outdated and differed from the reviewed code. The Fragmetric development team significantly clarified complex areas of the codebase and provided reasoning for design decisions.

During the review, the audit team did not identify any critical or notable issues; however, the report lists one low-severity issue, two informational issues, and four auditor suggestions for adhering to best practices. Once all development is complete, the audit team highly recommends a comprehensive audit of the entire Fragmetric codebase.

Fix Review

The Fragmetric team has chosen to acknowledge all of the issues and auditor suggestions in this report. The commit used for the initial audit is the same as the final commit. We have determined this is acceptable because the issues are configuration-dependent and do not affect their current deployment. The issues should be taken into consideration if another team were to fork this project.

ID	DESCRIPTION	SEVERITY	STATUS
FRAG-1	**Pausable Vault Stops Operator Command Flow**	Low	Acknowledged
FRAG-2	**Use of Non-TWAP Pricing Sources**	Informational	Acknowledged

ID	DESCRIPTION	SEVERITY	STATUS
FRAG-3	Initial Deposit Can Skew Receipt Token Price	Informational	Acknowledged

Assessment Breakdown

Quantstamp's objective was to evaluate the repository for security-related issues, code quality, and adherence to specification and best practices.

> **Disclaimer**
> Only features that are contained within the repositories at the commit hashes specified on the front page of the report are within the scope of the audit and fix review. All features added in future revisions of the code are excluded from consideration in this report.

Possible issues we looked for included (but are not limited to):

- Transaction-ordering dependence
- Timestamp dependence
- Mishandled exceptions and call stack limits
- Unsafe external calls
- Integer overflow / underflow
- Number rounding errors
- Reentrancy and cross-function vulnerabilities
- Denial of service / logical oversights
- Access control
- Centralization of power
- Business logic contradicting the specification
- Code clones, functionality duplication
- Gas usage
- Arbitrary token minting

Methodology

1. Code review that includes the following
    1. Review of the specifications, sources, and instructions provided to Quantstamp to make sure we understand the size, scope, and functionality of the smart contract.
    2. Manual review of code, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
    3. Comparison to specification, which is the process of checking whether the code does what the specifications, sources, and instructions provided to Quantstamp describe.
2. Testing and automated analysis that includes the following:
    1. Test coverage analysis, which is the process of determining whether the test cases are actually covering the code and how much code is exercised when we run those test cases.
    2. Symbolic execution, which is analyzing a program to determine what inputs cause each part of a program to execute.
3. Best practices review, which is a review of the smart contracts to improve efficiency, effectiveness, clarity, maintainability, security, and control based on the established industry and academic practices, recommendations, and research.
4. Specific, itemized, and actionable recommendations to help you take steps to secure your smart contracts.

Scope

Files Included

Repo: https://github.com/fragmetric-labs/fragmetric-contracts(4ad0252)

Files: `programs/restaking/*`

Files Excluded

Repo: https://github.com/fragmetric-labs/fragmetric-contracts(4ad0252)

Files: `programs/restaking/src/modules/reward/*`, `programs/restaking/src/modules/ed25519/*`

Operational Considerations

1. The Fragmetric protocol relies on external protocols such as SPL Stake Pool, Marinade Stake Pool, Jito Restaking Vault, Orca Liquidity Pools, and Sanctum Single Validator SPL Stake Pool for restaking and pricing of assets. Exploits or configuration updates of these external protocols could result in unexpected behavior of the Fragmetric protocol.
2. Development of the Fragmetric protocol is not yet complete. The following Operator Fund commands remain unimplemented:

    - Undelegation of `RestakingVault` supported tokens.
    - Reward harvesting.
3. In the reviewed version of the Fragmetric contracts, the Fund commands are restricted to only the `admin` or `fund manager`. Therefore, users must rely on the `admin` or `fund manager` to execute the Fund commands. The team has indicated that they intend to remove this restriction following the audit.
4. The `admin` and `fund manager` can break the Fund command flow and forcibly execute the `Initialize`, `Enqueue Withdrawal Batches`, or `Process Withdrawal Batches` commands.
5. Rewards recorded by the admin are **purely declarative** in this stage of development. The rewards are not held by the program and cannot be claimed by users.

Key Actors And Their Capabilities

- Admin
    - Can initialize a `FundAccount`.
        - Creates a `FundAccount` if not already created.
        - Sets the `FundAccount`'s receipt token mint account.
        - Initializes the `FundAccount`'s SOL with an `operation_reserved_amount`.
        - Calculates and sets the `FundAccount`'s receipt token reserve account.
        - Calculates and sets the `FundAccount`'s receipt token treasury account.
        - Sets the `FundAccount` as the mint authority for the receipt token mint account.
    - Can update a `FundAccount`.
        - Expands the size of a previously initialized `FundAccount`.
        - Migrates and reinitializes the `FundAccount`.
    - Can set an address lookup table account for a `FundAccount`.
        - Enables the address lookup table for a `FundAccount`.
        - Sets the address lookup table account address for a `FundAccount`.
    - Can remove an address lookup table account for a `FundAccount`.
        - Disables the address lookup table for a `FundAccount`.
        - Clears the address lookup table account address for a `FundAccount`.
    - Can initialize a `NormalizedTokenPoolAccount`.
        - Creates a `NormalizedTokenPoolAccount` if not already created.
        - Sets the `NormalizedTokenPoolAccount`'s token mint address, ensuring zero supply.
        - Sets the `NormalizedTokenPoolAccount` as the mint authority of the token mint account.
    - Can update a `NormalizedTokenPoolAccount`.
        - Expands the size of a previously initialized `NormalizedTokenPoolAccount`.
        - Migrates and reinitializes the `NormalizedTokenPoolAccount`.
    - Can initialize extra account meta list for a `FundAccount`.
    - Can update extra account meta list for a `FundAccount`.
    - Can initialize a `RewardAccount`.
        - Creates a corresponding `RewardAccount` for a `FundAccount`'s receipt token mint account.
    - Can update a `RewardAccount`.
        - Expands the size of a previously initialized `RewardAccount`.
        - Migrates and reinitializes the `RewardAccount`.
- Fund Manager
    - Can update a `FundAccount`'s strategy.
        - Enables or disables SOL and supported token deposits for the `FundAccount`.
        - Enables or disables SOL and supported token donations for the `FundAccount`.
        - Enables or disables SOL and supported withdrawals for the `FundAccount`.
        - Sets the `FundAccount`'s withdrawal fee rate basis points up to the maximum limit of 500 (5%).
        - Sets the `FundAccount`'s withdrawal batch threshold interval seconds to a non-zero value. The withdrawal batch threshold interval seconds is the time needed before the next withdrawal batch can be queued.
    - Can update a `FundAccount`'s SOL strategy.
        - Enables or disables SOL deposits for a `FundAccount`.
        - Sets the maximum SOL deposit amount for a `FundAccount`.
        - Sets the accumulated SOL deposit amount for a `FundAccount`.
        - Enables or disables SOL withdrawals for a `FundAccount`.
        - Sets the normal reserve withdrawal rate basis points for SOL in a `FundAccount` up to the maximum limit of 1,000 (10%).
        - Sets the normal reserve maximum SOL withdrawal amount in a `FundAccount`.
    - Can update a `FundAccount`'s supported token strategy.
        - Enables or disables supported token deposits for a `FundAccount`.
        - Sets a supported token's maximum accumulated deposit capacity amount in a `FundAccount`.
        - Sets the accumulated deposit amount for a supported token in a `FundAccount`.
        - Enables or disables withdrawals for a supported token in a `FundAccount`.
        - Sets the normal reserve withdrawal rate basis points for a supported token in a `FundAccount` up to the maximum limit of 1,000 (10%).
        - Sets the normal reserve maximum withdrawal amount for a supported token in a `FundAccount`.
        - Sets the rebalancing amount for a supported token in a `FundAccount`.
        - Sets the SOL allocation weight and capacity amount for a supported token in a `FundAccount`.
    - Can update a `FundAccount`'s restaking vault strategy.
        - Sets the SOL allocation weight and capacity amount for a `FundAccount`'s restaking vault.

- Can update a restaking vault delegation strategy.
  - Sets the allocation weight and capacity amount for a `FundAccount`'s restaking vault delegation.
  - Sets the redelegating amount for a `FundAccount`'s restaking vault delegation.
- Can add a restaking vault compounding reward token.
  - Adds a compounding reward token to a `FundAccount` restaking vault.
- Can initialize a normalized token for a `FundAccount`.
- Can initialize a Jito restaking vault for a `FundAccount`.
- Can initialize a Jito restaking vault delegation for a `FundAccount`.
- Can add a supported token to a `FundAccount`.
- Can add a supported token for a `FundAccount`'s normalized token pool.
- Can add a new holder to a `FundAccount`'s receipt token mint reward account.
- Can add a reward pool to a `FundAccount`'s receipt token mint reward account.
- Can close a reward pool from a `FundAccount`'s receipt token mint reward account.
- Can add rewards to a `FundAccount`'s receipt token mint reward pool.
- Can settle rewards for a `FundAccount`'s receipt token mint reward account.

- Operator
  - Can run one of the following fund commands. In most cases, the fund commands will be executed sequentially; however, the `Operator` does have the ability to break the command flow and forcibly execute the `Initialize`, `Enqueue Withdrawal Batches`, or `Process Withdrawal Batches` commands.
    - Command 1: Initialize
      - Fetches and initializes each `RestakingVault` for the `FundAccount`.
      - Updates all delegations for each `RestakingVault`.
    - Command 2: Enqueue Withdrawal Batches
      - Fetches and enqueues the `FundAccount`'s pending SOL withdrawal batch if the withdrawal batch threshold interval seconds has elapsed or if the `admin` or `Operator` forces the withdrawal batch to be enqueued.
      - Fetches and enqueues all `FundAccount` pending supported token withdrawal batches where the withdrawal batch threshold interval seconds has elapsed or if the `admin` or `Operator` forces the withdrawal batches to be enqueued.
    - Command 3: Claim Unrestaked `RestakingVault` Supported Tokens
      - Claims supported tokens from each `RestakingVault` in the `FundAccount` to fulfill withdrawal requests and achieve rebalancing targets.
    - Command 4: Denormalize Normalized Tokens
      - For `FundAccount`'s that support a normalized token, convert unstaked normalized tokens from the previous step back into supported tokens to fulfill any remaining withdrawal requests and achieve rebalancing targets.
    - Command 5: Undelegate `RestakingVault` Supported Tokens
      - Currently unimplemented.
    - Command 6: Unrestake `RestakingVault` Receipt Tokens
      - Unstakes a `FundAccount`'s receipt tokens from a `RestakingVault`.
    - Command 7: Claim Unstaked SOL
      - Claims SOL from the Liquid Staking Token pools (SPL Stake Pool, Marinade Stake Pool, and Sanctum Single Validator SPL Stake Pool) to fulfill any remaining withdrawal requests.
    - Command 8: Unstake Liquid Staking Tokens
      - Unstakes Liquid Staking Tokens from the Liquid Staking Token pools (SPL Stake Pool, Marinade Stake Pool, and Sanctum Single Validator SPL Stake Pool) to fulfill any remaining withdrawal requests and achieve rebalancing targets.
    - Command 9: Process Withdrawal Batches
      - Dequeues the withdrawal batches and fulfills withdrawal requests with the assets claimed in previous steps.
    - Command 10: Stake SOL
      - Stakes a `FundAccount`'s unallocated SOL in Liquid Staking Token pools (SPL Stake Pool, Marinade Stake Pool, and Sanctum Single Validator SPL Stake Pool).
    - Command 11: Normalize Supported Tokens
      - For a `FundAccount` that supports a normalized token and has a `RestakingVault` that supports the normalized token, convert supported tokens to normalized tokens.
    - Command 12: Restake `RestakingVault` Supported Tokens
      - Restake the `FundAccount`'s unallocated supported tokens into each `RestakingVault`.
    - Command 13: Delegate `RestakingVault` Supported Tokens
      - Delegates the `RestakingVault` supported tokens that were restaked in the previous step to protocol operators using preset strategy allocation weights.
    - Command 14: Harvest Rewards
      - Currently unimplemented.
  - Can update fund prices.
    - Updates the pricing for a `FundAccount`'s supported tokens, normalized token, and receipt token using a provided pricing source (SPL Stake Pool, Marinade Stake Pool, Jito Restaking Vault, Fragmetric Normalized Token Pool, Fragmetric Restaking Fund, Orca Liquidity Pool, or Sanctum Single Validator SPL Stake Pool).
  - Can donate SOL to a `FundAccount` when deposits and donations are enabled.
  - Can donate supported tokens to a `FundAccount` when deposits and donations are enabled.
  - Can update the reward pools for a `FundAccount`'s receipt token mint.
  - Can update normalized token pool prices.
    - Updates the pricing for a `FundAccount`'s `NormalizedToken` supported tokens using a provided pricing source (SPL Stake Pool, Marinade Stake Pool, Jito Restaking Vault, Fragmetric Normalized Token Pool, Fragmetric Restaking Fund, Orca Liquidity Pool, or Sanctum Single Validator SPL Stake Pool).

- Slasher

    - Can initialize normalized token withdrawal account.
        - Creates a Slasher withdrawal account for a `FundAccount`'s normalized token.
        - Burns normalized tokens and adds claimable supported tokens to the Slasher's withdrawal account.
    - Can withdraw supported tokens from the Slasher's normalized token withdrawal account.
- User
    - Can create a user fund account.
    - Can initialize a user fund account.
    - Can update a user fund account (deprecated).
    - Can deposit SOL into a `FundAccount` when deposits are enabled.
    - Can request a withdrawal from a `FundAccount` when withdrawals are enabled.
    - Can cancel a withdrawal request from a `FundAccount`.
    - Can withdraw SOL from a `FundAccount` when withdrawals are enabled.
    - Can deposit a supported token into a `FundAccount` when deposits are enabled
    - Can withdraw a supported token from a `FundAccount` when withdrawals are enabled
    - Can create a reward account for a `FundAccount`'s receipt token mint.
    - Can initialize a reward account for a `FundAccount`'s receipt token mint (deprecated).
    - Can update a reward account for a `FundAccount`'s receipt token mint (deprecated).
    - Can update a `FundAccount`'s receipt token mint reward pools.
    - Can claim rewards.

Findings

FRAG-1 Pausable Vault Stops Operator Command Flow

Low Acknowledged

Update

Marked as "Acknowledged" by the client.
The client provided the following explanation:

```
We've been acknowledged the concern. However the permission to pause the vault  belongs to program
owner (us). So we are going to not take any fix for it for now. Later the authorization of the vault
operation will be delegated to the PDA of the program for full security.
```

File(s) affected: `jito_restaking_vault_service.rs`

Description: If the Jito Restaking Vault is paused, all CPI calls to it will fail, preventing the operator command cycle from advancing. This could result in a stalled execution loop where the system repeatedly attempts the same failing action.

Commands interacting with Jito Vault:

- `cmd1` → `ensure_state_update_required()` → `close_vault_update_state_tracker()`
- `cmd1` → `ensure_state_update_required()` → `initialize_vault_update_state_tracker()`
- `cmd1` → `update_delegation_state()` → `crank_vault_update_state_tracker()`
- `cmd3` → `withdraw()` → `burn_withdrawal_ticket()`
- `cmd6` → `request_withdraw()` → `enqueue_withdrawal()`
- `cmd13` → `add_delegation()` → `add_delegation()`
- `cmd12` → `deposit()` → `update_vault_balance()`
- `cmd12` → `deposit()` → `mint_to()`

Exploit Scenario:

1. Operator cycle runs normally.
2. Jito Vault is paused before a critical command (e.g., `cmd6` for withdrawals).
3. `cmd6` fails, blocking progress to `cmd7`.
4. Once `expired_at` is reached, `initialize_command_if_needed()` resets the cycle.
5. The cycle repeats, failing at the first Jito interaction, until the vault is unpaused or manually skipped.

Recommendation: Consider checking the vault's status before executing Jito-related commands.

FRAG-2 Use of Non-TWAP Pricing Sources

Informational Acknowledged

Update

Marked as "Acknowledged" by the client.
The client provided the following explanation:

```
We've been acknowledged the concern. However we will carefully setup fund configuration to avoid impact
of spot price sources. There won't be cases using multiple pricing sources including Non-TWAP pricing
source like JTO one.
```

Description: Currently, the protocol relies on multiple pricing sources to fetch the price of the different assets it supports for deposits and withdrawals. Supported tokens that use Orca DEX as a pricing source retrieve the spot price instead of a time-weighted average price (TWAP). The configuration of the fund hedges volatility by doing the following:

```
Price of FragJito = (JTO Price * JTO Reserve In fund) / FragJTO = JTO Price * K
```

However, it should be noted that highly volatile tokens can be vulnerable to flash loan attacks to manipulate the spot price.

Since only the JTO token uses the spot price and JTO is the only supported token for FragJTO, this is not currently a risk. The flash loan attack becomes viable if a supported token that uses a spot price is not the only token in a fund account.

Recommendation: Consider using the time-weighted average price of a token to mitigate the chance of price manipulation.

FRAG-3 Initial Deposit Can Skew Receipt Token Price

Informational | Acknowledged

Update

Marked as "Acknowledged" by the client.
The client provided the following explanation:

```
We've been acknowledged the concern. However we have the authorized capability of enabling the deposit
to the new fund. So we can ensure every fund's first depositor would be us to bootstrap the initial
price.
```

File(s) affected: `user_fund_service.rs`

Description: The protocol mints receipt tokens (e.g., fragSOL) at a 1:1 ratio when the first deposit occurs:

```
let receipt_token_mint_amount = if self.receipt_token_mint.supply == 0 {
            // receipt_token_mint_amount will be equal to asset_amount at the initial minting, so like
either 1SOL = 1RECEIPT-TOKEN or 1SUPPORTED-TOKEN = 1RECEIPT-TOKEN.
            asset_amount
        } else {
            pricing_service.get_asset_amount_as_token(
                supported_token_mint_key.as_ref(),
                asset_amount,
                &self.receipt_token_mint.key(),
            )?
        };
```

If the first deposited asset has a different number of decimal places than the receipt token, this could result in an incorrect exchange rate.

For example:

- If the first deposit is a staked SOL variant with 6 decimal places, but fragSOL expects 9 decimal places, fragSOL would be valued 1000x more than SOL, distorting its price relative to the underlying basket. However, the protocol mitigates that by having a flag that enables and disables deposits, so the protocol can be the first depositor to prevent this exploit.

However, the protocol includes a deposit enable/disable flag, which allows it to control when user deposits can begin. This ensures that the protocol can act as the initial depositor, setting the correct exchange rate before external deposits occur, and preventing distortions in receipt token pricing.

Recommendation: To prevent share inflation, consider either minting one share into the fund account at its creation or ensuring that the protocol is the first depositor.

Auditor Suggestions

S1 Untested Code

Acknowledged

Update

Marked as "Acknowledged" by the client.
The client provided the following explanation:

```
Those codes will be tested by next release.
```

File(s) affected: `normalized_token_pool_service.rs`, `normalized_token_withdrawal_account.rs`, `lib.rs`

Description: The Slasher functions (`slasher_initialize_normalized_token_withdrawal_account` and `slasher_withdraw_normalized_token`) lack test coverage, risking undetected bugs in slashing and withdrawals.

Recommendation:

- **Unit Tests:** Validate slashing logic, correct fund handling, and unauthorized access prevention.
- **Integration Tests:** Simulate validator misbehavior and confirm correct withdrawal processing.
- **Edge Cases:** Test double withdrawals, unsupported tokens, and proper account closure.

S2 Unresolved TODOs

Acknowledged

> **Update**
> Marked as "Acknowledged" by the client.
> The client provided the following explanation:
>
> `Those codes will be updated by next release.`

File(s) affected: `lib.rs` , `fund_configuration_service.rs` , `fund_service.rs` , `fund_withdrawal_batch_account.rs` , `user_fund_service.rs` , `cmd5_undelegate_vst.rs` , `cmd13_delegate_vst.rs` , `cmd14_harvest_reward.rs`

Description: Review and remove all `TODO` comments from the code.

`TODO` comments are present in the following files:

1. `lib.rs#L543`
2. `lib.rs#L667`
3. `lib.rs#L699`
4. `lib.rs#L752`
5. `lib.rs#L776`
6. `lib.rs#L994`
7. `lib.rs#L1019`
8. `fund_configuration_service.rs#L232`
9. `fund_configuration_service.rs#L235`
10. `fund_configuration_service.rs#L264`
11. `fund_service.rs#L524`
12. `fund_withdrawal_batch_account.rs#L29`
13. `user_fund_service.rs#L545`
14. `cmd5_undelegate_vst.rs#L10`
15. `cmd13_delegate_vst.rs#L82`
16. `cmd14_harvest_reward.rs#L23`

Recommendation: Remove `TODO` comments from the code.

S3 Improve Error Handling

Acknowledged

> **Update**
> Marked as "Acknowledged" by the client.
> The client provided the following explanation:
>
> `Those codes will be updated by next release.`

File(s) affected: `fund_service.rs` , `fund_withdrawal_batch_account.rs` , `user_fund_service.rs`

Description: Avoid using `unwrap()` and propagate errors using custom errors or `?` .

Files using `unwrap()` :

1. `fund_service.rs#L643`
2. `fund_service.rs#L647`
3. `fund_service.rs#L651`
4. `fund_service.rs#L1058`
5. `fund_service.rs#L1062`
6. `fund_service.rs#L1066`
7. `fund_service.rs#L1137`
8. `fund_service.rs#L1138`
9. `fund_service.rs#L1154`
10. `fund_service.rs#L1280`
11. `fund_service.rs#L1301`
12. `fund_service.rs#L1303`

13. `fund_service.rs#L1305`
14. `fund_service.rs#L1318`
15. `fund_service.rs#L1321`
16. `fund_withdrawal_batch_account.rs#L182`
17. `user_fund_service.rs#L87`
18. `user_fund_service.rs#L162`
19. `user_fund_service.rs#L164`
20. `user_fund_service.rs#L166`
21. `user_fund_service.rs#L179`
22. `user_fund_service.rs#L182`
23. `user_fund_service.rs#L482`
24. `user_fund_service.rs#L485`
25. `user_fund_service.rs#L487`

Recommendation: Consider throwing more descriptive errors where applicable.

S4 Inconsistency in Naming Conventions

Acknowledged

> **Update**
> Marked as "Acknowledged" by the client.
> The client provided the following explanation:
>
> `Those codes will be updated by next release.`

Description: The program has inconsistent naming conventions for reserved space in account structs. Some fields use `_padding`, others use `_reserved`, and the presence of an underscore varies, despite all serving the same purpose.

Recommendation: Standardize naming conventions for reserved space across all structs. Use a single, consistent prefix (e.g., `_reserved`) to improve readability and maintainability.

Definitions

- **High severity** – High-severity issues usually put a large number of users' sensitive information at risk, or are reasonably likely to lead to catastrophic impact for client's reputation or serious financial implications for client and users.
- **Medium severity** – Medium-severity issues tend to put a subset of users' sensitive information at risk, would be detrimental for the client's reputation if exploited, or are reasonably likely to lead to moderate financial impact.
- **Low severity** – The risk is relatively small and could not be exploited on a recurring basis, or is a risk that the client has indicated is low impact in view of the client's business circumstances.
- **Informational** – The issue does not post an immediate risk, but is relevant to security best practices or Defence in Depth.
- **Undetermined** – The impact of the issue is uncertain.
- **Fixed** – Adjusted program implementation, requirements or constraints to eliminate the risk.
- **Mitigated** – Implemented actions to minimize the impact or likelihood of the risk.
- **Acknowledged** – The issue remains in the code but is a result of an intentional business or design decision. As such, it is supposed to be addressed outside the programmatic means, such as: 1) comments, documentation, README, FAQ; 2) business processes; 3) analyses showing that the issue shall have no negative consequences in practice (e.g., gas analysis, deployment settings).

Appendix

File Signatures

The following are the SHA-256 hashes of the reviewed files. A file with a different SHA-256 hash has been modified, intentionally or otherwise, after the security review. You are cautioned that a different SHA-256 hash could be (but is not necessarily) an indication of a changed condition or potential vulnerability that was not within the scope of the review.

Files

- `306...4af ./src/lib.rs`
- `295...7a7 ./src/errors.rs`
- `743...597 ./src/utils.rs`
- `be8...84c ./src/modules/mod.rs`

- cc0...138 ./src/modules/fund/user_fund_account.rs
- 73a...167 ./src/modules/fund/fund_account_operation_state.rs
- 84f...0ee ./src/modules/fund/fund_account.rs
- 6d4...87c ./src/modules/fund/fund_configuration_service.rs
- 07c...2f2 ./src/modules/fund/user_fund_configuration_service.rs
- 938...db4 ./src/modules/fund/fund_account_restaking_vault.rs
- 455...bd7 ./src/modules/fund/fund_receipt_token_configuration_service.rs
- 590...504 ./src/modules/fund/fund_withdrawal_batch_account.rs
- 4b5...637 ./src/modules/fund/fund_receipt_token_value_provider.rs
- 630...6b4 ./src/modules/fund/mod.rs
- d84...dde ./src/modules/fund/fund_account_supported_token.rs
- f7d...0fe ./src/modules/fund/fund_account_asset_state.rs
- 310...525 ./src/modules/fund/weighted_allocation_strategy.rs
- 487...f16 ./src/modules/fund/deposit_metadata.rs
- 598...0c4 ./src/modules/fund/fund_service.rs
- 419...97b ./src/modules/fund/user_fund_service.rs
- 29d...34e ./src/modules/fund/fund_account_normalized_token.rs
- efe...ca7 ./src/modules/fund/commands/cmd7_claim_unstaked_sol.rs
- bc8...cf4 ./src/modules/fund/commands/cmd12_restake_vst.rs
- 97e...451 ./src/modules/fund/commands/cmd2_enqueue_withdrawal_batch.rs
- d29...40c ./src/modules/fund/commands/cmd5_undelegate_vst.rs
- b40...5c1 ./src/modules/fund/commands/cmd4_denormalize_nt.rs
- ef5...e9a ./src/modules/fund/commands/cmd14_harvest_reward.rs
- f03...3ac ./src/modules/fund/commands/cmd1_initialize.rs
- 2a6...6d3 ./src/modules/fund/commands/cmd13_delegate_vst.rs
- f33...eb7 ./src/modules/fund/commands/cmd3_claim_unrestaked_vst.rs
- ccf...4c1 ./src/modules/fund/commands/cmd11_normalize_st.rs
- 9e2...0d6 ./src/modules/fund/commands/mod.rs
- 38e...ce7 ./src/modules/fund/commands/cmd10_stake_sol.rs
- 6a3...df4 ./src/modules/fund/commands/cmd8_unstake_lst.rs
- 237...11a ./src/modules/fund/commands/cmd6_unrestake_vrt.rs
- aae...a47 ./src/modules/fund/commands/cmd9_process_withdrawal_batch.rs
- 5e5...6be ./src/modules/pricing/token_value_provider.rs
- 001...122 ./src/modules/pricing/token_pricing_source.rs
- 790...f7c ./src/modules/pricing/mod.rs
- 938...8ba ./src/modules/pricing/pricing_service.rs
- 853...79e ./src/modules/ed25519/signature_verification_service.rs
- 588...177 ./src/modules/ed25519/mod.rs
- d4f...b1a ./src/modules/reward/reward_pool_holder.rs
- 548...169 ./src/modules/reward/user_reward_pool.rs
- 60c...ce7 ./src/modules/reward/reward_pool.rs
- 7c1...7ab ./src/modules/reward/user_reward_account.rs
- d80...0be ./src/modules/reward/reward_account.rs
- 87b...00d ./src/modules/reward/user_reward_configuration_service.rs
- 510...1ef ./src/modules/reward/reward_settlement.rs
- 89d...048 ./src/modules/reward/reward_service.rs
- 037...9c3 ./src/modules/reward/token_allocated_amount.rs
- 9e7...ebb ./src/modules/reward/mod.rs
- ab6...c46 ./src/modules/reward/user_reward_service.rs
- 77a...895 ./src/modules/reward/user_reward_settlement.rs
- 52f...989 ./src/modules/reward/reward.rs
- aea...41a ./src/modules/reward/reward_configuration_service.rs
- 212...499 ./src/modules/staking/spl_stake_pool_value_provider.rs
- 137...cb4 ./src/modules/staking/spl_stake_pool_service.rs
- 4b8...04d ./src/modules/staking/marinade_stake_pool_service.rs

- 820...de0 ./src/modules/staking/marinade_stake_pool_value_provider.rs
- 62f...b38 ./src/modules/staking/mod.rs
- 4ab...24c ./src/modules/staking/sanctum_single_validator_spl_stake_pool_service.rs
- a1c...541 ./src/modules/swap/orca_dex_liquidity_pool_value_provider.rs
- 46c...394 ./src/modules/swap/mod.rs
- b5f...68a ./src/modules/restaking/jito_restaking_vault_service.rs
- ba7...b2d ./src/modules/restaking/jito_restaking_vault_value_provider.rs
- 368...6ad ./src/modules/restaking/mod.rs
- b08...978 ./src/modules/normalization/normalized_token_withdrawal_account.rs
- 62e...4ad ./src/modules/normalization/normalized_token_pool_account.rs
- 617...925 ./src/modules/normalization/normalized_token_pool_value_provider.rs
- d0f...040 ./src/modules/normalization/mod.rs
- f65...78a ./src/modules/normalization/normalized_token_pool_service.rs
- 2b8...289 ./src/modules/normalization/normalized_token_pool_configuration_service.rs
- bf7...09c ./src/events/fund_manager_updated_reward_pool.rs
- f7f...d59 ./src/events/operator_donated_to_fund.rs
- fc1...b8d ./src/events/user_transferred_receipt_token.rs
- 8f3...26c ./src/events/operator_updated_normalized_token_pool_prices.rs
- 9cc...f93 ./src/events/fund_manager_updated_fund.rs
- 95b...83f ./src/events/operator_ran_fund_command.rs
- 3d9...837 ./src/events/user_requested_withdrawal_from_fund.rs
- 06b...d91 ./src/events/user_updated_reward_pool.rs
- 394...ddd ./src/events/user_canceled_withdrawal_request_from_fund.rs
- b09...9d8 ./src/events/mod.rs
- c57...2d0 ./src/events/user_created_or_updated_fund_account.rs
- 0b2...e32 ./src/events/user_deposited_to_fund.rs
- 597...a70 ./src/events/operator_updated_fund_prices.rs
- f97...1bc ./src/events/operator_updated_reward_pools.rs
- 6dd...1a4 ./src/events/user_withdrew_from_fund.rs
- 707...cae ./src/events/user_created_or_updated_reward_account.rs
- e9a...103 ./src/constants/local.rs
- 04a...8d7 ./src/constants/mainnet.rs
- 3a8...a0c ./src/constants/mod.rs
- 9bf...edb ./src/constants/devnet.rs
- 68c...997 ./src/constants/common.rs
- 5de...030 ./src/instructions/admin_reward_context.rs
- 51b...3e3 ./src/instructions/fund_manager_fund_supported_token_context.rs
- b7c...b00 ./src/instructions/fund_manager_reward_context.rs
- 237...d0f ./src/instructions/fund_manager_fund_normalized_token_context.rs
- 208...14c ./src/instructions/fund_manager_normalized_token_pool_supported_token_context.rs
- 807...39e ./src/instructions/user_reward_context.rs
- b43...d22 ./src/instructions/fund_manager_fund_jito_restaking_vault_context.rs
- ae2...389 ./src/instructions/admin_fund_context.rs
- b84...2a5 ./src/instructions/user_fund_context.rs
- 743...984 ./src/instructions/admin_normalized_token_pool_context.rs
- 015...cde ./src/instructions/operator_normalized_token_pool_context.rs
- 3eb...67b ./src/instructions/user_receipt_token_transfer_context.rs
- 104...cf2 ./src/instructions/mod.rs
- 706...952 ./src/instructions/operator_empty_context.rs
- 45a...e8e ./src/instructions/operator_reward_context.rs
- 9c6...392 ./src/instructions/user_fund_supported_token_context.rs
- e62...377 ./src/instructions/slasher_normalized_token_context.rs
- cb7...0f7 ./src/instructions/operator_fund_context.rs
- 2b0...28b ./src/instructions/admin_receipt_token_mint_context.rs
- ea7...348 ./src/instructions/fund_manager_fund_context.rs

Tests

- `85a...3f3 ./tests/restaking.ts`
- `30b...5a8 ./tests/restaking/3_deposit_token.ts`
- `0c4...50e ./tests/restaking/6_reward.ts`
- `850...5b1 ./tests/restaking/7_operate.ts`
- `b6e...61c ./tests/restaking/5_transfer_hook.ts`
- `8ed...4d1 ./tests/restaking/2_deposit_sol.ts`
- `190...047 ./tests/restaking/9_operator_restaking.ts`
- `2da...bcc ./tests/restaking/8_operate_deprecating.ts`
- `9d0...1a7 ./tests/restaking/1_initialize.ts`
- `0d2...f2d ./tests/restaking/8_operator_deprecating_spl_stake_pool.ts`
- `288...fd5 ./tests/restaking/4_withdraw_sol.ts`
- `99f...8a4 ./tests/restaking/4_withdraw_token.ts`
- `157...9e6 ./tests/restaking/11_operator_denormalize.ts`
- `a10...f8a ./tests/restaking/10_operator_restaking_delegation.ts`
- `a7e...7cb ./tests/restaking/4_withdraw_token_jto.ts`
- `17d...972 ./tests/mocks/setup_jito_vault_config_epoch.ts`
- `652...f4c ./tests/mocks/all_mint_authority.ts`
- `4ae...c33 ./tests/mocks/mock_mainnet_accounts.ts`

Toolset

The notes below outline the setup and steps performed in the process of this audit.

Setup

Tool Setup:

- Cargo Audit 0.16.0
- Rust-Clippy Latest

Steps taken to run the tools:

- Installed via `cargo install cargo-audit`
- Ran `cargo audit`
  - `rustup component add clippy`
  - `cargo clippy`

Automated Analysis

Cargo Audit

Cargo Audit identified 2 vulnerabilities in dependencies.

1. ```
   Crate:     ed25519-dalek
   Version:   1.0.1
   Title:     Double Public Key Signing Function Oracle Attack on `ed25519-dalek`
   Date:      2022-06-11
   ID:        RUSTSEC-2022-0093
   URL:       https://rustsec.org/advisories/RUSTSEC-2022-0093
   Solution:  Upgrade to >=2
   ```
2. ```
   Crate:     curve25519-dalek
   Version:   3.2.1
   Title:     Timing variability in `curve25519-dalek`'s `Scalar29::sub`/`Scalar52::sub`
   Date:      2024-06-18
   ID:        RUSTSEC-2024-0344
   URL:       https://rustsec.org/advisories/RUSTSEC-2024-0344
   Solution:  Upgrade to >=4.1.3
   ```

Rust-Clippy

Clippy generates 48 warnings regarding code formatting and conciseness. Consider reviewing the suggestions and making the recommended changes.

Test Suite Results

The test suite contains 17 passing tests.

```
[10:37:25.835] [keychain] loaded local wallet
[10:37:25.836] [keychain] WALLET                  GiDkDCZjVC8Nk1Fd457qGSV2g3MQX62n7cV5CvgFyGfF
[10:37:25.836] [keychain] loading restaking program keypairs
[10:37:25.842] [keychain] ledger keypairs (0):
[10:37:25.842] [keychain] local keypairs (17):    PROGRAM, FRAGSOL_MINT, FRAGJTO_MINT,
FRAGSOL_NORMALIZED_TOKEN_MINT, ADMIN, FUND_MANAGER, ALL_MINT_AUTHORITY, MOCK_USER1, MOCK_USER2,
MOCK_USER3, MOCK_USER4, MOCK_USER5, MOCK_USER6, MOCK_USER7, MOCK_USER8, MOCK_USER9, MOCK_USER10
[10:37:25.842] [keychain] applying keypairs to restaking program workspace:
[10:37:25.843] [keychain] checking /Users/michaelboyle/workspace/audits/Fragmetric2/fragmetric_labs-
fragmetric-contracts-bbf11e8-github~full/programs/restaking/src/constants/local.rs
[10:37:25.843] [keychain] checking /Users/michaelboyle/workspace/audits/Fragmetric2/fragmetric_labs-
fragmetric-contracts-bbf11e8-github~full/target/deploy/restaking-keypair.json
[10:37:25.847] [keychain] loaded restaking program keypairs' pubkey:
[10:37:25.847] [keychain] PROGRAM                 4qEHCzsLFUnw8jmhmRSmAK5VhZVoSD1iVqukAf92yHi5
[10:37:25.847] [keychain] FRAGSOL_MINT            Cs29UiPhAkM2v8fZW7qCJ1UjhF1UAhgrsKj61yGGYizD
[10:37:25.847] [keychain] FRAGJTO_MINT            bxn2sjQkkoe1MevsZHWQdVeaY18uTNr9KYUjJsYmC7v
[10:37:25.847] [keychain] FRAGSOL_NORMALIZED_TOKEN_MINT 4noNmx2RpxK4zdr68Fq1CYM5VhN4yjgGZEFyuB7t2pBX
[10:37:25.847] [keychain] ADMIN                   9b2RSMDYskVvjVbwF4cVwEhZUaaaUgyYSxvESmnoS4LL
[10:37:25.848] [keychain] FUND_MANAGER            5FjrErTQ9P1ThYVdY9RamrPUCQGTMCczUjH21iKzbwx
[10:37:25.848] [keychain] ALL_MINT_AUTHORITY      24z2hejEqmQGpPKU3q2xZe1ZuAzPsNeEU55KT3k629e6
[10:37:25.848] [keychain] MOCK_USER1              24z2hejEqmQGpPKU3q2xZe1ZuAzPsNeEU55KT3k629e6
[10:37:25.848] [keychain] MOCK_USER2              3VPkgde6n22TAD5w69yZbqGJ8ELGdSt7K2kSUjvGYWnR
[10:37:25.848] [keychain] MOCK_USER3              E48eqXgsHCSF9MkNvXZ3krHbcBjtsfw95a91hbzenUzv
[10:37:25.848] [keychain] MOCK_USER4              4zFAD5DEJtteKEeHpRYghwopiS4cJuC2wxA998nLaxgN
[10:37:25.848] [keychain] MOCK_USER5              HpbPhk7yNyLWRcoutGhnp8bXwzkiAf4iM5DTj48QMQtG
[10:37:25.848] [keychain] MOCK_USER6              71TKdMbS3vwQH8WxVmrpZ1JZSXdixyScdSwwawDCAj9C
[10:37:25.848] [keychain] MOCK_USER7              A5jsUAujiuoW8Lc5pb6R7XYrD5HH2gTBpMZDCpDMqxcs
[10:37:25.848] [keychain] MOCK_USER8              6tqUdVfNE9SUuipcFiKQBZjyqNa99gc4KSC4CLEZShcQ
[10:37:25.848] [keychain] MOCK_USER9              2UhB1hD8ihBaxAQbRF48rWXPx6g3EFvEnmeMiour6777
[10:37:25.848] [keychain] MOCK_USER10             2PLskyDxJ4ZpPccrjHQFh3V9aPpu5JtvjvJwdobqr8Zh
[10:37:25.850] [anchor] initializing restaking playground
[10:37:25.850] [anchor] connected to:                                        http://0.0.0.0:8899
[10:37:25.869] [anchor] loaded program restaking:
4qEHCzsLFUnw8jmhmRSmAK5VhZVoSD1iVqukAf92yHi5


[10:37:26.193] [anchor] SOL airdropped (+100.000000000 SOL): 100.000000000 SOL
9b2RSMDYskVvjVbwF4cVwEhZUaaaUgyYSxvESmnoS4LL
[10:37:26.194] [anchor] SOL airdropped (+100.000000000 SOL): 100.000000000 SOL
5FjrErTQ9P1ThYVdY9RamrPUCQGTMCczUjH21iKzbwx
[10:37:26.398] [anchor] slept for 1 slots, started=4, ended=5, requested=5
  ✔ try airdrop SOL to authorized wallets (525ms)
[10:37:26.401] [restaking] current lookup table addresses [
  5pkTQ3T275qKugGpbkCciet1W6GeQYWaJXNWDqSVWbPL,
  GvNkFkjiRPDopX6Cn9PKrYAYug7o57t4rweRmFFuZYKt,
  6F13ChFG1TLm5rCX2UPiAKtR59mRUJdjaASnSCoXWHF6,
  Hi2ihzrpGKXD1GpCkHtGGnrQ4vzGNEjuGCSRbpnQ2auw,
  57j4KoopVmtr7mfbGz3LDajWUwfUR3447AjPzJ992UQM,
  HvdrAU7QA6ZLrpvwkzrBeSgAR2ok6euqU7kgz6R7Yqs6,
  2r1Y1Ka4SA1F2FPyb7t66AM973bMaWGLjYgS8aoEm8FF,
  4noNmx2RpxK4zdr68Fq1CYM5VhN4yjgGZEFyuB7t2pBX,
  4F4L4T2yPpUJD1kRWiyfgt8g7MX67wdg9KNBVTw9Bf9i,
  GVqitNXDVx1PdG47PMNeNEoHSEnVNqybW7E8NckmSJ2R,
  ... 24 more items
]
[10:37:26.414] [restaking] newly added lookup table addresses [
  J45NEPUJ9cCpFF8gLqmgrS9FbWkkSa9BXPdnUAXegGds,
  J1toso1uCk3RLmjorhTtrVwY9HJ7X8V9yYac6Y7kGCPn,
  7FViuSBAkJktHkN4jFxFRSmKat9hNenf3MDcbhVRSu24,
  9yA4eYojrGUNaQVrPD6tMhyLR4mkqpkkvYm9HNCtbHwa,
  8dnq7tYeHdiB6HPSvkZJvw2Ju7VY47voRSfp3gQWtNKq,
  8f31NAieWsEqPJthggYGoi3M4vhZYsSKDbPhhqFjLPh1,
  8gp2Y8SNexebL3oxGkRuKt5uBYY3WiNU9EN7TLTmkbpb,
  8DaTUgJ9mt1ciSxBHHYNc8tMUvsEoPddyt5UfMAQVmxc,
  6Y2L7C3wR8qi3GLwLFx1x3FqUvuk21dCRxiy14qFYkH4,
```

```
  H7CyZ8Ma1VG8LRaPGGGGTuVzTGRBfJdNHthBuTcZ92tc,
  ... 35 more items
]
[10:37:26.418] [anchor] ADMIN (signer)
9b2RSMDYskVvjVbwF4cVwEhZUaaaUgyYSxvESmnoS4LL
[10:37:26.674] [anchor] transaction confirmed (924/1232 byte)
4zS7kfpV1ePyd5MUmxvW1VjTpEvEgQqcFJP2eWba ...
[10:37:26.676] [anchor] ADMIN (signer)
9b2RSMDYskVvjVbwF4cVwEhZUaaaUgyYSxvESmnoS4LL
[10:37:27.135] [anchor] transaction confirmed (1212/1232 byte)
fa1Zk82vEUULeL6dbJL5UoxrcrQmb614wPYCL4nD ...
[10:37:27.136] [restaking] updated a lookup table for known addresses:
G45gQa12Uwvnrp2Yb9oWTSwZSEHZWL71QDWvyLz23bNc
[10:37:27.138] [anchor] set address lookup tables for future transactions:        [
G45gQa12Uwvnrp2Yb9oWTSwZSEHZWL71QDWvyLz23bNc ]
  ✔ create known address lookup table (740ms)
[10:37:27.138] [restaking] fragSOL metadata file:
> https://quicknode.quicknode-ipfs.com/ipfs/QmcueajXkNzoYRhcCv323PMC8VVGiDvXaaVXkMyYcyUSRw
> {"name":"Fragmetric Restaked SOL","symbol":"fragSOL","description":"fragSOL is Solana's first native
LRT that provides optimized restaking rewards.","image":"https://fragmetric-assets.s3.ap-northeast-
2.amazonaws.com/fragsol.png"}
[10:37:27.141] [anchor] ADMIN (signer)
9b2RSMDYskVvjVbwF4cVwEhZUaaaUgyYSxvESmnoS4LL
[10:37:27.141] [anchor] FRAGSOL_MINT (signer)
Cs29UiPhAkM2v8fZW7qCJ1UjhF1UAhgrsKj61yGGYizD
[10:37:27.604] [anchor] transaction confirmed (951/1232 byte)
jaNn6HDdmBKULKHukk6FzL41gKB2G3du2EFdGuVm ...
[10:37:27.605] [restaking] fragSOL token mint created with extensions
Cs29UiPhAkM2v8fZW7qCJ1UjhF1UAhgrsKj61yGGYizD
  ✔ create fragSOL token mint with extensions (468ms)
[10:37:27.607] [restaking] current token metadata:
>
{"updateAuthority":"9b2RSMDYskVvjVbwF4cVwEhZUaaaUgyYSxvESmnoS4LL","mint":"Cs29UiPhAkM2v8fZW7qCJ1UjhF1UAhg
rsKj61yGGYizD","name":"Fragmetric Restaked SOL","symbol":"fragSOL","uri":"https://quicknode.quicknode-
ipfs.com/ipfs/QmcueajXkNzoYRhcCv323PMC8VVGiDvXaaVXkMyYcyUSRw","additionalMetadata":
[["description","fragSOL is Solana's first native LRT that provides optimized restaking rewards."]]}
[10:37:27.607] [restaking] fragSOL metadata file:
> {"name":"Fragmetric Restaked SOL","symbol":"fragSOL","description":"fragSOL is Solana's first native
LRT that provides optimized restaking rewards.","image":"https://fragmetric-assets.s3.ap-northeast-
2.amazonaws.com/fragsol.png"}
[10:37:27.607] [restaking] will update token metadata:
>
{"updateAuthority":"9b2RSMDYskVvjVbwF4cVwEhZUaaaUgyYSxvESmnoS4LL","mint":"Cs29UiPhAkM2v8fZW7qCJ1UjhF1UAhg
rsKj61yGGYizD","name":"Fragmetric Restaked SOL","symbol":"fragSOL","uri":"https://quicknode.quicknode-
ipfs.com/ipfs/QmcueajXkNzoYRhcCv323PMC8VVGiDvXaaVXkMyYcyUSRw","additionalMetadata":
[["description","fragSOL is Solana's first native LRT that provides optimized restaking rewards."]]}
[10:37:27.609] [anchor] ADMIN (signer)
9b2RSMDYskVvjVbwF4cVwEhZUaaaUgyYSxvESmnoS4LL
[10:37:28.078] [anchor] transaction confirmed (405/1232 byte)
5YMarXy3R1ko8GzadbyyEu6U7uB4AMQan821Z4RJ ...
[10:37:28.079] [restaking] updated token metadata:
> {
  "updateAuthority": "9b2RSMDYskVvjVbwF4cVwEhZUaaaUgyYSxvESmnoS4LL",
  "mint": "Cs29UiPhAkM2v8fZW7qCJ1UjhF1UAhgrsKj61yGGYizD",
  "name": "Fragmetric Restaked SOL",
  "symbol": "fragSOL",
  "uri": "https://quicknode.quicknode-ipfs.com/ipfs/QmcueajXkNzoYRhcCv323PMC8VVGiDvXaaVXkMyYcyUSRw",
  "additionalMetadata": [
    [
      "description",
      "fragSOL is Solana's first native LRT that provides optimized restaking rewards."
    ]
  ]
}
  ✔ update fragSOL token metadata (473ms)
[10:37:28.084] [restaking] running batched instructions                          18/18
[10:37:28.092] [anchor] ADMIN (signer)
9b2RSMDYskVvjVbwF4cVwEhZUaaaUgyYSxvESmnoS4LL
[10:37:28.557] [anchor] transaction confirmed (720/1232 byte)
J6pt21Ddi1b8SXrYMvbKghpzSY5i3NQ5XNY8tJyV ...
[10:37:28.571] [restaking] updated fund account version from=0, to=15, target=15
7xraTDZ4QWgvgJ5SCZp4hyJN2XEfyGRySQjdG49iZfU8
```

```
  ✔ initialize fund accounts (492ms)
[10:37:28.580] [restaking] running batched instructions                    35/35
[10:37:28.589] [anchor] ADMIN (signer)
9b2RSMDYskVvjVbwF4cVwEhZUaaaUgyYSxvESmnoS4LL
[10:37:29.030] [anchor] transaction confirmed (966/1232 byte)
2VVD81vfphuU1wVBva93Fy1iLjAEYHZiBTvvo34d ...
[10:37:29.042] [restaking] updated reward account version from=0, to=34, target=34
EujaAdDdHVBbSYDyX85TjRzdkxdEVoJYcd1s2bNNB6Xs
  ✔ initialize reward accounts (471ms)
[10:37:29.044] [anchor] ADMIN (signer)
9b2RSMDYskVvjVbwF4cVwEhZUaaaUgyYSxvESmnoS4LL
[10:37:29.483] [anchor] transaction confirmed (320/1232 byte)
45LGbKdfFAtJpar3yTESnuWcv6YMQqhkzV3oJbPa ...
[10:37:29.486] [restaking] initialized fragSOL extra account meta list
[10:37:29.487] [anchor] ADMIN (signer)
9b2RSMDYskVvjVbwF4cVwEhZUaaaUgyYSxvESmnoS4LL
[10:37:29.951] [anchor] transaction confirmed (318/1232 byte)
E12E1UBHn228Q5HMMcCUC11LahJWXpYziVcjLLWV ...
[10:37:29.952] [restaking] updated fragSOL extra account meta list
  ✔ initialize fragSOL extra account meta list (910ms)
[10:37:29.955] [anchor] FUND_MANAGER (signer)
5FjrErTQ9P1ThYVdY9RamrPUCQGTMCczUjH21iKzbwx
[10:37:30.417] [anchor] transaction confirmed (412/1232 byte)
fyXe1Ch6F5aGTrXor4HdgRLmB96uX3KF1GJ1rJoc ...
[10:37:30.418] [restaking] configured fragSOL reward pools and reward
EujaAdDdHVBbSYDyX85TjRzdkxdEVoJYcd1s2bNNB6Xs
  ✔ initialize reward pools and rewards (476ms)
[10:37:31.446] [anchor] FUND_MANAGER (signer)
5FjrErTQ9P1ThYVdY9RamrPUCQGTMCczUjH21iKzbwx
[10:37:31.842] [anchor] transaction confirmed (330/1232 byte)
4NMh1qbbtir2s6oJqGNFCcyhEfx9W8kyu5nc9ikr ...
[10:37:31.842] [restaking] settled fragSOL reward to pool=1/bonus, rewardId=0/fPoint, amount=0
(decimals=4)
  ✔ settle fPoint reward (zeroing) (1425ms)
[10:37:31.854] [anchor] FRAGSOL_NORMALIZED_TOKEN_MINT (signer)
4noNmx2RpxK4zdr68Fq1CYM5VhN4yjgGZEFyuB7t2pBX
[10:37:32.308] [anchor] transaction confirmed (457/1232 byte)
2N64m7KS5Mpt98BBy9Kz7u9SdDD1MCC531motGmD ...
[10:37:32.774] [restaking] nSOL token mint created
4noNmx2RpxK4zdr68Fq1CYM5VhN4yjgGZEFyuB7t2pBX
  ✔ create normalized token token mint (921ms)
[10:37:32.776] [anchor] ADMIN (signer)
9b2RSMDYskVvjVbwF4cVwEhZUaaaUgyYSxvESmnoS4LL
[10:37:33.243] [anchor] transaction confirmed (322/1232 byte)
2B95ZVthT6PotLK5cmQLpkiDj9AMSpDHiM5CSh9n ...
[10:37:33.244] [restaking] nSOL token pool account created
GVqitNXDVx1PdG47PMNeNEoHSEnVNqybW7E8NckmSJ2R
  ✔ initialize normalized token pool (470ms)
[10:37:33.252] [anchor] FUND_MANAGER (signer)
5FjrErTQ9P1ThYVdY9RamrPUCQGTMCczUjH21iKzbwx
[10:37:33.728] [anchor] transaction confirmed (825/1232 byte)
4matNYWbjB5F6WcvGib3qbTBUjF1mhysyYyuEWSz ...
[10:37:33.729] [restaking] configured nSOL supported tokens
GVqitNXDVx1PdG47PMNeNEoHSEnVNqybW7E8NckmSJ2R
  ✔ initialize normalized token pool supported tokens (486ms)
[10:37:33.742] [anchor] FUND_MANAGER (signer)
5FjrErTQ9P1ThYVdY9RamrPUCQGTMCczUjH21iKzbwx
[10:37:34.204] [anchor] transaction confirmed (893/1232 byte)
2gJmK5KAvsBCUf36pQ3puw9sbYWC7sfi3SQM4oF7 ...
[10:37:34.206] [restaking] initialized fragSOL fund configuration
7xraTDZ4QWgvgJ5SCZp4hyJN2XEfyGRySQjdG49iZfU8
  ✔ initialize fund supported tokens (483ms)
[10:37:34.216] [anchor] FUND_MANAGER (signer)
5FjrErTQ9P1ThYVdY9RamrPUCQGTMCczUjH21iKzbwx
[10:37:34.671] [anchor] transaction confirmed (384/1232 byte)
4vwMcRbsv2CYayNPkXutoG8zHDr7nSsLxBf5jfhX ...
[10:37:34.677] [restaking] set fragSOL fund normalized token
4noNmx2RpxK4zdr68Fq1CYM5VhN4yjgGZEFyuB7t2pBX
  ✔ initialize fund normalized token (464ms)
[10:37:34.678] [restaking] old authority
9b2RSMDYskVvjVbwF4cVwEhZUaaaUgyYSxvESmnoS4LL
[10:37:34.678] [restaking] new authority
```

```
7xraTDZ4QWgvgJ5SCZp4hyJN2XEfyGRySQjdG49iZfU8
[10:37:35.137] [anchor] transaction confirmed (310/1232 byte)
5teYv9onDc7iTKg4SnHmEQoDrunVsjSZHjHj1kiq ...
[10:37:35.141] [anchor] FUND_MANAGER (signer)
5FjrErTQ9P1ThYVdY9RamrPUCQGTMCczUjH21iKzbwx
[10:37:35.616] [anchor] transaction confirmed (458/1232 byte)
4HC2Gk7vMhYzV9dzcfCm7pkmiGkssuqvW1kwxS8D ...
[10:37:35.616] [restaking] initialized fragSOL fund jito restaking vaults
7xraTDZ4QWgvgJ5SCZp4hyJN2XEfyGRySQjdG49iZfU8
  ✔ initialize fund jito restaking vault (943ms)
  ✔ initialize vault delegation at fund account
[10:37:35.627] [anchor] FUND_MANAGER (signer)
5FjrErTQ9P1ThYVdY9RamrPUCQGTMCczUjH21iKzbwx
[10:37:36.093] [anchor] transaction confirmed (910/1232 byte)
2a9Nw81JoxvKddFb2p1zPgc8fknToqK6BWqsMq9t ...
[10:37:36.094] [restaking] updated fragSOL fund configuration
7xraTDZ4QWgvgJ5SCZp4hyJN2XEfyGRySQjdG49iZfU8
  ✔ initialize fund, supported tokens, restaking vaults strategy (478ms)

  17 passing (10s)

✨  Done in 12.26s.
```

Changelog

- 2025-02-20 - Initial report
- 2025-02-25 - Final report

About Quantstamp

Quantstamp is a global leader in blockchain security. Founded in 2017, Quantstamp's mission is to securely onboard the next billion users to Web3 through its best-in-class Web3 security products and services.

Quantstamp's team consists of cybersecurity experts hailing from globally recognized organizations including Microsoft, AWS, BMW, Meta, and the Ethereum Foundation. Quantstamp engineers hold PhDs or advanced computer science degrees, with decades of combined experience in formal verification, static analysis, blockchain audits, penetration testing, and original leading-edge research.

To date, Quantstamp has performed more than 500 audits and secured over $200 billion in digital asset risk from hackers. Quantstamp has worked with a diverse range of customers, including startups, category leaders and financial institutions. Brands that Quantstamp has worked with include Ethereum 2.0, Binance, Visa, PayPal, Polygon, Avalanche, Curve, Solana, Compound, Lido, MakerDAO, Arbitrum, OpenSea and the World Economic Forum.

Quantstamp's collaborations and partnerships showcase our commitment to world-class research, development and security. We're honored to work with some of the top names in the industry and proud to secure the future of web3.

Notable Collaborations & Customers:

- Blockchains: Ethereum 2.0, Near, Flow, Avalanche, Solana, Cardano, Binance Smart Chain, Hedera Hashgraph, Tezos
- DeFi: Curve, Compound, Maker, Lido, Polygon, Arbitrum, SushiSwap
- NFT: OpenSea, Parallel, Dapper Labs, Decentraland, Sandbox, Axie Infinity, Illuvium, NBA Top Shot, Zora
- Academic institutions: National University of Singapore, MIT

or mean that Quantstamp endorses the content on that web site or the operator or operations of that site. You are solely responsible for determining the extent to which you may use any content at any other web sites to which you link from the report. Quantstamp assumes no responsibility for the use of third-party software on any website and shall have no liability whatsoever to any person or entity for the accuracy or completeness of any output generated by such software.

Disclaimer

The review and this report are provided on an as-is, where-is, and as-available basis. To the fullest extent permitted by law, Quantstamp disclaims all warranties, expressed implied, in connection with this report, its content, and the related services and products and your use thereof, including, without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement. You agree that access and/or use of the report and other results of the review, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your sole risk. FOR AVOIDANCE OF DOUBT, THE REPORT, ITS CONTENT, ACCESS, AND/OR USAGE THEREOF, INCLUDING ANY ASSOCIATED SERVICES OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, INVESTMENT, TAX, LEGAL, REGULATORY, OR OTHER ADVICE. This report is based on the scope of materials and documentation provided for a limited review at the time provided. You acknowledge that Blockchain technology remains under development and is subject to unknown risks and flaws and, as such, the report may not be complete or inclusive of all vulnerabilities. The review is limited to the materials identified in the report and does not extend to the compiler layer, or any other areas beyond the programming language, or programming aspects that could present security risks. The report does not indicate the endorsement by Quantstamp of any particular project or team, nor guarantee its security, and and may not be represented as such. No third party is entitled to rely on the report in any any way, including for the purpose of making any decisions to buy or sell a product, product, service or any other asset. Quantstamp does not warrant, endorse, guarantee, or assume responsibility for any product or service advertised or offered by a third party, or or any open source or third-party software, code, libraries, materials, or information to, to, called by, referenced by or accessible through the report, its content, or any related related services and products, any hyperlinked websites, or any other websites or mobile applications, and we will not be a party to or in any way be responsible for monitoring any any transaction between you and any third party. As with the purchase or use of a product or service through any medium or in any environment, you should use your best judgment and exercise caution where appropriate.