

EigenLayer Rewards v1 Audit Report

OpenBlock Labs

Contents

[1. Executive Summary](#)

[2. Fixes and Acknowledgments](#)

[3. Issues Identified](#)

[3.1 Negative Shares for Stakers Effected by Withdrawal Migration](#)

[3.2 Slashed Native ETH Restakers Leading to Negative Shares](#)

[3.3 Being Able to Emit QueueWithdrawal Multiple Times](#)

[3.4 PodSharesUpdated Event Timing](#)

[3.5 Time Str Inconsistency](#)

[3.6 Discrepancy Between Deposit Event Shares vs Deposit Amount](#)

[3.7 Snapshot Inconsistency Due to cutoff date \$\leq\$ usage](#)

[4. Retest](#)

[4.1. Scope of the Retest](#)

[4.2. Retest Results](#)

[5. Conclusion](#)

1. Executive Summary

OpenBlock Labs conducted a comprehensive two-phase audit of EigenLayer's Rewards v1 implementation between June 25th and July 20th, 2024. The audit focused on the robustness and correctness of the rewards calculation mechanisms and identified critical

issues affecting reward distribution accuracy. All identified issues were resolved during the audit period.

Audit Details

Audit Scope	Reward calculation logic in the Bronze, Silver, and Gold DBT models, along with the DAG implementation
Repository	https://github.com/Layr-Labs/eigenpipe
Date	June 25th, 2024 – July 20th, 2024
Audit Phases	Phase 1: Bronze and Silver Queries, Phase 2: Gold Queries

Risk Level Classification

Risk Level	Impact: High	Impact: Medium	Impact: Low
Likelihood: High	Critical	High	Medium
Likelihood: Medium	High	Medium	Low
Likelihood: Low	Medium	Low	Low

Action Required for risk levels

Critical: Must be fixed immediately, especially if already deployed to production.

High: Must be addressed before deployment, or as soon as possible if not yet deployed.

Medium: Should be fixed to improve functionality or reduce potential risks.

Low: Could be resolved if time permits, as it has minimal impact.

Findings Count

Risk Level	Amount
Critical	1
High	0
Medium	1
Low	5

Summary of Key Findings and Resolutions

Title	Risk Level	Status
Negative Shares for Stakers Effected by Withdrawal Migration	Critical	Fixed
Slashed Native ETH Restakers Leading to Negative Shares	Medium	Fixed
Being Able to Emit QueueWithdrawal Multiple Time	Low	Fixed
PodSharesUpdated Event Timing	Low	Acknowledged
Time Str Inconsistency	Low	Fixed
Discrepancy Between Deposit Event Shares vs Deposit Amount	Low	Acknowledged
Snapshot Inconsistency Due to cutoff date <= usage	Low	Fixed

2. Fixes and Acknowledgments

EigenLayer resolved all critical and medium-risk findings identified in the Rewards v1 audit. Low-risk findings were also addressed where possible, with some acknowledged as design choices to balance trade-offs between functionality and performance.

3. Issues Identified

3.1 Negative Shares for Stakers Effected by Withdrawal Migration

3.1.1 Risk Level: Critical

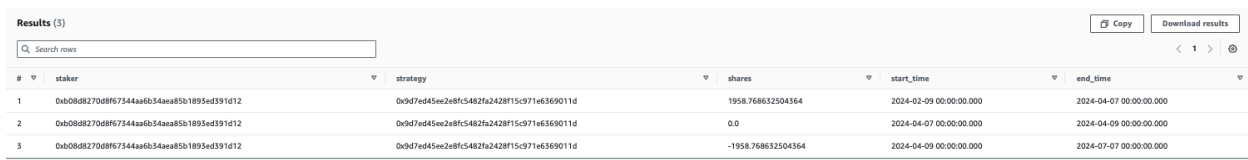
3.1.2 Status: [Fixed](#)

3.1.3 Root Cause

During the queued withdrawal migration on Apr 4th, the **WithdrawalQueued** event was emitted one extra time for restakers with queued withdrawal transactions. Due to this extra emitted event, some restakers have negative running sum shares in the mainnet dataset.

3.1.4 Impact

Misrepresentation of staker balances. An example of a user affected by this issue can be seen in the image below.

A screenshot of a data table interface. At the top, it says "Results (5)" and has a search bar. Below the search bar is a table with 7 columns: #, staker, strategy, shares, start_time, and end_time. The table contains 3 rows of data. The first row shows a staker with 1958.768632504364 shares. The second row shows a staker with 0.0 shares. The third row shows a staker with -1958.768632504364 shares. The table is bordered by a blue line.

#	staker	strategy	shares	start_time	end_time
1	0xb0808270d8f67344aa6b54aa85b1895ed391d12	0xd7ed45ee2e8fc5482fa2428f15c971e6369011d	1958.768632504364	2024-02-09 00:00:00.000	2024-04-07 00:00:00.000
2	0xb0808270d8f67344aa6b54aa85b1895ed391d12	0xd7ed45ee2e8fc5482fa2428f15c971e6369011d	0.0	2024-04-07 00:00:00.000	2024-04-09 00:00:00.000
3	0xb0808270d8f67344aa6b54aa85b1895ed391d12	0xd7ed45ee2e8fc5482fa2428f15c971e6369011d	-1958.768632504364	2024-04-09 00:00:00.000	2024-07-07 00:00:00.000

3.1.5 Recommended Fix

Filter out the **WithdrawalQueued** events emitted by migration transactions from the dataset.

3.2 Slashed Native ETH Restakers Leading to Negative Shares

3.2.1 Risk Level: Medium

3.2.2 Status: [Fixed](#)

3.2.3 Root Cause

Native ETH restakers with an effective balance of less than 32 ETH at the time of calling QueueWithdrawals function can temporarily have a negative running sum of shares in staker_shares dataset between the times they call verifyAndProcessWithdrawals and completeQueuedWithdrawal.

3.2.4 Impact

Misrepresentation of staker balances.

3.3 Being Able to Emit QueueWithdrawal Multiple Times

3.3.1 Risk Level: Low

3.3.2 Status: [Fixed](#)

3.3.3 Root Cause

Users could request multiple withdrawals without completing the initial one, leading to negative balances **staker_shares** table

3.3.4 Impact

Misrepresentation of staker balances.

3.3.5 Recommended Fix

This is due to a smart contract bug. Since M1 withdrawals are deprecated and only one user is affected by this issue, the affected transaction can be filtered out.

3.4 PodSharesUpdated Event Timing

3.4.1 Risk Level: Low

3.4.2 Status: Acknowledged as design choice

3.4.3 Root Cause

The **PodSharesUpdated** event is not triggered when the validator becomes active, instead it is emitted when the pod owner initiates a transaction, such as starting to delegate their stake on EigenLayer. If the native staker never delegates, they will not be tracked and if the Rewards for All are called, these users will be missed

3.4.4 Impact

Native ETH stakers who never delegated their stakes were not being tracked, causing them to miss out on rewards distribution.

3.5 Time Str Inconsistency

3.5.1 Risk Level: Low

3.5.2 Status: [Fixed](#)

3.5.3 Root Cause

In `eigenlabs_rewards_mvp.py:248` the format used for the str includes micro-seconds, but on line 253 the assumption is that there are not micro-seconds. This will crash if `dag_run` is ever None during the run of `get_gold_calculation_dates`.

3.5.4 Impact

Potential crashes in the rewards pipeline when encountering specific conditions (e.g., dag_run being None).

3.6 Discrepancy Between Deposit Event Shares vs Deposit Amount

3.6.1 Risk Level: Low

3.6.2 Status: Acknowledged as design choice

3.6.3 Root Cause

Deposits do not provide a 1:1 ratio of token amount to shares with different ratios for each token/transaction. While this may end up being the chosen functionality, this makes comparing stakes between users of different tokens inconsistent.

3.7 Snapshot Inconsistency Due to cutoff date \leq usage

3.7.1 Risk Level: Low

3.7.2 Status: [Fixed](#)

3.7.3 Root Cause

The [Bronze tables](#) use “ \leq TIMESTAMP '{{ var("cutoff_date") }}'” in their queries. In [Staker Share Windows](#) the block_time is CAST to DATE which maps it to 00:00 for the current DATE during the ROW_NUMBER() function

3.7.4 Impact

This will cause an inconsistency if an event occurs at exactly 00:00 on a given day.

Example:

- Deposit event is emitted on 7/14/2024 00:00:00

- This is included in the run at 7/16/2024 16:00:00 because it is \leq cutoff_date.
 - On this run, since that is the only event on 7/14, the modification to the state by that event will be chosen
- In the run at 7/17/2024 16:00:00 additional events on 7/14 will now be included
 - On this run the state on 7/14 could change if additional events are emitted for that staker/strategy combination on 7/14, leading the historic state to be different between two runs

This is not possible currently as the last block of the day on mainnet is currently at 23:59:59, so this would require a +1 sec shift in blocks to occur for some reason. However it also shouldn't change anything then to use a strict $<$.

4. Retest

4.1. Scope of the Retest

The retest phase validated the fixes applied to issues identified during the audit. It also included testing the consistency of rewards calculations across all tiers (Bronze, Silver, Gold).

4.2. Retest Results

Negative Shares for Stakers Effected by Withdrawal Migration: Fix validated. All stakers now have consistent share calculations. The fix can be found here: [PR](#)

Slashed Native ETH Restakers Leading to Negative Shares: Resolved edge case scenarios. The dataset no longer exhibits negative shares. The fix can be found here: [PR](#)

Being Able to Emit QueueWithdrawal Multiple Times: Retest confirmed that multiple withdrawals are now filtered out from the dataset. The fix can be found here: [PR](#)

Time Str Inconsistency: Fix validated. All date strings are now consistently formatted. The fix can be found here: [PR](#)

Snapshot Inconsistency Due to cutoff date \leq usage: The usage of \leq has been replaced with $<$, and the fix has been validated. The fix can be found here: [PR](#)

5. Conclusion

The fixes and adjustments made have enhanced the robustness of the Rewards v1 implementation, ensuring accurate and reliable reward distributions. All critical, medium-risk, and low-risk issues have been resolved, with design choices documented for certain low-risk findings.