Aura Finance Migration Review

By Hephyrius.eth

Table of contents

1. Overview

The Aura Finance platform has undergone a series of changes and fixes, and as a result, a migration was necessary to ensure the smooth and secure operation of the platform. In this document, we will review the changes that were made to the Aura smart contracts, outline the migration flow, discuss the outcome of the migration, and evaluate the execution. A summary of changes that occurred during the migration can be found here.

2. Issues

2.1 Aura Mint Manipulation

The Aura token rewards are generated based on the number of BAL tokens received by the Booster contract when distributing rewards, which are provided to the protocol by Balancer's smart contracts. However, the Booster contract did not properly consider situations where BAL tokens were intentionally sent to it from unexpected sources, which could have resulted in more Aura tokens being minted than intended. specifically, the `_earmarkRewards` and `earmarkFees` functions utilised the balance of BAL in the booster after claiming any pending rewards, without accounting for prior balances.

Additionally, the reward contracts for the protocol included a `donate` function that allowed BAL tokens to be donated to pool participants. This function also resulted in the minting of additional Aura tokens based on the amount of BAL donated. This function was not properly guarded, which may have contributed to potential manipulation of Aura token emissions.

2.2 Governance Griefing

It was reported that there were a few different situations in which individuals with privileged rights (e.g. governance) could potentially abuse certain aspects of the code to take tokens from the Aura Finance protocol. However, it is believed that the likelihood of

these scenarios occurring was low due to the requirement for privileged rights. In other words, these scenarios would require a person with specific privileges to carry out the abuse, which made it less likely to happen.

3. Fixes

3.1 Aura Mint Manipulation

The team has successfully addressed both of the issues highlighted in 2.1

3.1.1 Inaccurate Accounting

A fix was implemented to accurately calculate the BAL balance of the `Booster` account when earmarking rewards. Instead of simply checking the current BAL balance of the `Booster`, the fix calculates the exact amount of BAL that was claimed from the gauge. This prevents external accounts from manipulating the number of reported rewards by transferring BAL directly to the `Booster`. Any existing BAL held by the `Booster` is transferred to the treasury address as the `Booster` should not have any BAL left in its balance, as it is transferred immediately to the rewards contract once claimed from the gauge.

3.1.2 Donate

The `donate` function has been removed from the `BaseRewardPool` contract. This means that additional rewards cannot be donated via the pool anymore.

3.2 Governance

The team has successfully addressed the issues highlighted in 2.2 and added additional security measures to harden smart contracts.

3.2.1 setVote

The `Booster` contract contained a function called `setVote` that allowed the voting delegate to mark any vote hash as either valid or invalid on the VoterProxy. This function could potentially be manipulated by governance, so the contributors decided to block valid votes as a measure to ensure the integrity of the voting process. The `setVote` function allowed the voting delegate to have significant control over the

outcome of votes on the VoterProxy, and by blocking valid votes, the contributors aimed to prevent any potential manipulation or interference with the voting process.

3.2.2 Reentrancy Guard

To prevent manipulation of the `Booster` functions, re-entrancy guards were added. Specifically the Open Zeppelin re-entrancy guard. The following functions now contain a `nonReentrant` modifier to guard against re-entrancy:

- Deposit
- Distribute L2 fees
- Earmark fees
- Earmark rewards
- Set fees
- Set fees info
- Shut down pool
- Withdraw

4. New Features

The aura team has taken the booster migration as an opportunity to add new functionality that will aid in the evolution and longevity of the protocol. This section reviews the changes that have been introduced.

4.1 Increase `Max Fees`

To update the `MaxFees` value, the constant value in the contract was changed. This constant is used in the `setFees()` function of the `Booster` and helps ensure that the total fees collected by the protocol do not exceed 40%. This represents a 15% absolute increase in the maximum fee compared to the previous maximum fee of 25% before the migration. The `setFees()` function and the `MaxFees` constant work together to establish limits on the fees that the protocol can collect, helping to ensure that the fees remain reasonable and fair for users.

4.2 Additional `Transfer` Events

In order to improve compatibility with off-chain providers, such as on block explorers, a `Transfer` event was added to the `BaseRewardPool`. This event is emitted every time a `deposit` or `withdraw` takes place. The event allows for better tracking and accounting of balances in the reward contracts.

4.3 L2 Pool Rewards

AURA intends to launch instances of the protocol on other chains that have a Balancer deployment. In order to distribute rewards to LPs on these chains, a method is required that allows the minting of AURA based on fees accrued cross-chain.

To address this, a new function was added to the updated `Booster` contract. This feature allows a delegate to exchange the claimed rewards collected from L2 pools for an amount of AURA rewards that these LPs should be eligible for. There is a hard cap of `70,000` BAL for each `EPOCH` period. An epoch is considered to be 1 week.

4.4 Reward Multipliers

To give Aura more control over the distribution of AURA rewards assigned to each pool, a new concept of multipliers was introduced. Previously, the number of AURA rewards was directly tied to the number of rewards collected by the `Booster`. Essentially, AURA emissions were indirectly controlled by Balancer.

With the introduction of this new concept, a multiplier can be set, which allows for the adjustment of the number of AURA rewards minted per BAL accrued per pool. This means that the number of AURA rewards can now be increased or decreased for each pool as desired. The range of multipliers is 0% for no emissions, 200% for double emissions and a default of 100% for unaltered emissions rates.

4.5 Migrator Contract

The migrator contract is responsible for easing the migration process by allowing users to withdraw their staked balance from the old `Booster` and deposit it into the new `Booster`. It enables users to migrate staked balances for multiple pools simultaneously, making it a convenient and efficient way to move balances between the two

`Booster` contracts. In order to use this contract, the user simply needs to grant it allowances to access their staked balance and initiate the migration process.

5. Migration

5.1 Overview

The migration process involved several stages, which were tested on forked instances of the main network to assess:

1. The successful implementation of fixes and changes
2. The impact of the migration on existing smart contracts
3. The overall success of the migration

Before the migration process, a thorough test was conducted on the newly deployed contracts to verify the expected outcome. The set of test covered a wide range of protocol aspects and ensured that not only the fixes and new features were working as intended, but also that the rest of the protocol's functionalities remained unaffected.

The team did an excellent job in implementing these tests, which helped ensure the smooth and successful execution of the migration. As a result of the thorough testing and evaluation conducted before the migration, the team was able to confidently proceed with the migration process

5.2 New Addresses

The new contracts were deployed to the Ethereum Mainnet to the following addresses;

- AuraClaimZap `0x2E307704EfaE244c4aae6B63B601ee8DA69E92A9`
- Booster `0xA57b8d98dAE62B26Ec3bcC4a365338157060B234`
- CvxCrvRewards `0x00A7BA8Ae7bca0B10A32Ea1f8e2a1Da980c6CAd2`
- PoolMigrator `0x12addE99768a82871EAaecFbDB065b12C56F0578`

5.3 Migration Steps

The migration process involved several steps, most of which were carried out using the AURA SAFE multi-sig, which was the only address with the necessary permissions to

perform certain actions. The steps taken during the migration included:

- Deploying new contracts
- Shutting down existing pools
- Shutting down the previous `PoolManager` and `BoosterOwner` contracts
- Setting AURA as the operator of the new `Booster` instance
- Setting new peripheral contracts
- Adding all pools back to the new `Booster` instance

These steps were necessary to ensure a smooth and successful migration from the old `Booster` to the new one. By following this process, the team was able to update the `Booster` contract and related contracts in a controlled and orderly manner, minimizing any disruption to the protocol.

6. Conclusion

The Aura Finance platform underwent a necessary migration in order to address several issues with the smart contracts and ensure the smooth and secure operation of the platform. The Aura team demonstrated exceptional professionalism and diligence throughout this process, carefully analysing all reported vulnerabilities and developing comprehensive solutions to address them while also taking proactive measures to mitigate any potential related issues. The quality of the fixes and the overall migration process was consistently high and in line with the high standards set by the project since its inception.

To ensure the smooth execution of the migration, multiple verification steps were taken, including testing on forked instances of the mainnet and involving external reviewers to provide additional oversight. The leadership and attention to detail of **0xMaha** and **Fry**, who took on full responsibility for the migration and successfully guided it to completion, were instrumental in ensuring its success.

Overall, the Aura team should be commended for their thorough and professional approach to the migration process. Their efforts were instrumental in ensuring the smooth and successful execution of the migration and deserve recognition.