iosiro

Lyra Vesting Smart Contract Audit

Lyra Finance, 16 June 2021

hello@iosiro.com

Contents

1. **Introduction** 1
2. **Disclaimer** 2
3. **Methodology** 3
4. **Audit findings** 4
   IO-LYR-VES-001 transfer should be replaced with safeTransfer for non-Lyra/stLyra tokens. . 5
   IO-LYR-VES-002 setDelegate(...) passes an empty string for the ID parameter. 6

1. Introduction

iosiro was commissioned by Lyra Finance to perform a smart contract audit of Lyra's Vesting smart contracts. One auditor conducted the audit between 15 and 16 June 2021, using two audit days.

Overview

The Lyra Vesting Contract system provides token vesting functionality for the Lyra governance token. The system consists of a factory contract that creates individual vesting escrow instances, each responsible for managing the gradual release of tokens to a recipient over a specified vesting schedule. The Lyra token itself is a standard ERC-20 governance token.

The audit identified two informational findings. The first finding noted that the use of `transfer` instead of `safeTransfer` could cause issues if non-standard ERC-20 tokens were used with the contracts. The second finding highlighted that the `setDelegate` function passes an empty string for the delegation ID parameter.

	Critical	High	Medium	Low	Informational
Open	0	0	0	0	2
Resolved	0	0	0	0	0
Closed	0	0	0	0	0

Scope

The assessment focused on the source files listed below, with all other files considered out of scope. Any out-of-scope code interacting with the assessed code was presumed to operate correctly without introducing functional or security vulnerabilities.

- **Project name:** lyra-governance
- **Commit:** `5b2be65`
- **Files:** VestingEscrow.sol, VestingEscrowFactory.sol, Lyra.sol

2. Disclaimer

This report aims to provide an overview of the assessed smart contracts' risk exposure and a guide to improving their security posture by addressing identified issues. The audit, limited to specific source code at the time of review, sought to:

- Identify potential security flaws.
- Verify that the smart contracts' functionality aligns with their documentation.

Off-chain components, such as backend web application code, keeper functionality, and deployment scripts were explicitly not within the scope of this audit.

Given the unregulated nature and ease of cryptocurrency transfers, operations involving these assets face a high risk from cyber attacks. Maintaining the highest security level is crucial, necessitating a proactive and adaptive approach that accounts for the experimental and rapidly evolving nature of blockchain technology. To encourage secure code development, developers should:

- Integrate security throughout the development lifecycle.
- Employ defensive programming to mitigate the risks posed by unexpected events.
- Adhere to current best practices wherever possible.

3. Methodology

The audit was conducted using the techniques described below.

Code review	The source code was manually inspected to identify potential security flaws. Code review is a useful approach for detecting security flaws, discrepancies between the specification and implementation, design improvements, and high-risk areas of the system.
Dynamic analysis	The contracts were compiled, deployed, and tested in a test environment, both manually and through the test suite provided. Dynamic analysis was used to identify additional edge cases, confirm that the code was functional, and to validate the reported issues.
Automated analysis	Automated tooling was used to detect the presence of various types of security vulnerabilities. Static analysis results were reviewed manually, and any false positives were removed. Any true positive results are included in this report.

4. Audit findings

The table below provides an overview of the audit's findings, with detailed write-ups in the following sections.

ID	Issue	Risk	Status
IO-LYR-VES-001	`transfer` should be replaced with `safeTransfer` for non-Lyra/stLyra tokens	Informational	Closed
IO-LYR-VES-002	`setDelegate(...)` passes an empty string for the ID parameter	Informational	Closed

Each issue identified during the audit has been assigned a risk rating. The rating is determined based on the criteria outlined below.

Critical risk	The issue could result in the theft of funds from the contract or its users.
High risk	The issue could result in the loss of funds for the contract owner or its users.
Medium risk	The issue resulted in the code being dysfunctional or the specification being implemented incorrectly.
Low risk	A design or best practice issue that could affect the ordinary functioning of the contract.
Informational	An improvement related to best practice or a suboptimal design pattern.

In addition to a risk rating, each issue is assigned a status:

Open	The issue remained present in the code as of the final commit reviewed and may still pose a risk.
Resolved	The issue was identified during the audit and has since been satisfactorily addressed, removing the risk it posed.
Closed	The issue was identified during the audit and acknowledged by the developers as an acceptable risk without actioning any change.

`IO-LYR-VES-001`

`transfer` should be replaced with `safeTransfer` for non-Lyra/stLyra tokens

Informational	Closed	VestingEscrow.sol, VestingEscrowFactory.sol

The vesting escrow contracts use the ERC-20 `transfer` function to send tokens to recipients. While this works correctly for standard ERC-20 implementations such as the Lyra and stLyra tokens, some non-standard ERC-20 tokens do not return a boolean value from their `transfer` function. If such tokens were used with the vesting contracts, the call could revert unexpectedly or, in some cases, the return value could be incorrectly handled.

Recommendation

If the vesting contracts are intended to support tokens beyond Lyra and stLyra, the `transfer` calls should be replaced with `safeTransfer` from OpenZeppelin's `SafeERC20` library. The library checks the return value where present and does not require one where absent, providing a more robust approach to token transfers.

Client response

Lyra acknowledged the finding.

IO-LYR-VES-002

`setDelegate(...)` passes an empty string for the ID parameter

Informational	Closed	VestingEscrow.sol

The `setDelegate` function in the `VestingEscrow` contract passes an empty string (`""`) as the ID parameter when calling the delegation functionality. The delegation ID is typically used to distinguish between different types of delegations, allowing a contract to delegate different responsibilities to different addresses. By passing an empty string, all delegations made through the vesting escrow contract use the same default identifier, which could limit flexibility if multiple delegation types need to be supported in the future.

Recommendation

A meaningful ID should be passed to the `setDelegate` function, or the function should be updated to accept an ID parameter from the caller, allowing different delegation types to be distinguished.

Client response

Lyra acknowledged the finding.