Three Sigma Labs

Code Audit

LAYER3

Layer3 The Layer3 ERC-20 token

Disclaimer

Code Audit

Layer3 The Layer3 ERC-20 token

Disclaimer

The ensuing audit offers no assertions or assurances about the code's security. It cannot be deemed an adequate judgment of the contract's correctness on its own. The authors of this audit present it solely as an informational exercise, reporting the thorough research involved in the secure development of the intended contracts, and make no material claims or guarantees regarding the contract's post-deployment operation. The authors of this report disclaim all liability for all kinds of potential consequences of the contract's deployment or use. Due to the possibility of human error occurring during the code's manual review process, we advise the client team to commission several independent audits in addition to a public bug bounty program.

Table of Contents

Code Audit

Layer3 The Layer3 ERC-20 token

Table of Contents

Summary

Code Audit

Layer3 The Layer3 ERC-20 token

Summary

Three Sigma Labs audited Layer3 in a 1 day engagement. The audit was conducted on 06-07-2024.

Protocol Description

The Layer3 token is a governance and utility token that will power the future of the Layer3 ecosystem.

Scope

Code Audit

Layer3 The Layer3 ERC-20 token

Scope

Layer3.sol

Assumptions

External dependencies are considered safe.

Methodology

Code Audit

Layer3 The Layer3 ERC-20 token

Methodology

To begin, we reasoned meticulously about the contract's business logic, checking security-critical features to ensure that there were no gaps in the business logic and/or inconsistencies between the aforementioned logic and the implementation. Second, we thoroughly examined the code for known security flaws and attack vectors. Finally, we discussed the most catastrophic situations with the team and reasoned backwards to ensure they are not reachable in any unintentional form.

Taxonomy

In this audit we report our findings using as a guideline Immunefi's vulnerability taxonomy, which can be found at immunefi.com/severity-updated/. The final classification takes into account the severity, according to the previous link, and likelihood of the exploit. The following table summarizes the general expected classification according to severity and likelihood; however, each issue will be evaluated on a case-by-case basis and may not strictly follow it.

Severity / Likelihood	LOW	MEDIUM	HIGH
NONE	None		
LOW	Low		
MEDIUM	Low	Medium	Medium
HIGH	Medium	High	High
CRITICAL	High	Critical	Critical

Project Dashboard

Code Audit

Layer3 The Layer3 ERC-20 token

Project Dashboard

Application Summary

Name	Layer3
Commit	f1cc66fd8da366b043832546841149O9f19583a8
Language	Solidity
Platform	Ethereum

Engagement Summary

Timeline	06-07-2024
№ of Auditors	1
Review Time	1 day

Vulnerability Summary

No vulnerabilities were found.

Code Maturity Evaluation

Code Audit

Layer3 The Layer3 ERC-20 token

Code Maturity Evaluation

Code Maturity Evaluation Guidelines

Category	Evaluation
Access Controls	The use of robust access controls to handle identification and authorization and to ensure safe interactions with the system.
Arithmetic	The proper use of mathematical operations and semantics.
Centralization	The presence of a decentralized governance structure for mitigating insider threats and managing risks posed by contract upgrades
Code Stability	The extent to which the code was altered during the audit.
Upgradeability	The presence of parameterizations of the system that allow modifications after deployment.
Function Composition	The functions are generally small and have clear purposes.
Front-Running	The system's resistance to front-running attacks.
Monitoring	All operations that change the state of the system emit events, making it simple to monitor the state of the system. These events need to be correctly emitted.
Specification	The presence of comprehensive and readable codebase documentation.
Testing and Verification	The presence of robust testing procedures (e.g., unit tests, integration tests, and verification methods) and sufficient test coverage.

Code Maturity Evaluation Results

Category	Evaluation
Access Controls	Satisfactory. The codebase has a strong access control mechanism.
Arithmetic	Satisfactory. The codebase uses Solidity version >0.8.0 as well as takes the correct measures in rounding the results of arithmetic operations.
Centralization	Satisfactory. The owner does not have significant control over the funds.
Code Stability	Satisfactory. No new functionality was being added throughout the audit.
Upgradeability	Satisfactory. The protocol is upgradeable.
Function Composition	Moderate. Functionality was well split into functions, but the codebase has significant code duplication.
Front-Running	Satisfactory. No significant frontrunning opportunities were found.
Monitoring	Satisfactory. Events are correctly emitted for the most significant state changes.
Specification	Satisfactory. The code matched the design specifications.
Testing and Verification	Moderate. Unit tests were present for most functionality but fuzzing and invariant tests could be performed.