



verichains

SECURITY AUDIT OF

**LINEAR STAKING POOL V2 SMART
CONTRACTS**



Public Report

Jul 01, 2024

Verichains Lab

info@verichains.io

<https://www.verichains.io>

Driving Technology > Forward

ABBREVIATIONS

Name	Description
Ethereum	An open source platform based on blockchain technology to create and distribute smart contracts and decentralized applications.
Ether (ETH)	A cryptocurrency whose blockchain is generated by the Ethereum platform. Ether is used for payment of transactions and computing services in the Ethereum network.
Smart contract	A computer protocol intended to digitally facilitate, verify or enforce the negotiation or performance of a contract.
Solidity	A contract-oriented, high-level language for implementing smart contracts for the Ethereum platform.
Solc	A compiler for Solidity.
ERC20	ERC20 (BEP20 in Binance Smart Chain or xRP20 in other chains) tokens are blockchain-based assets that have value and can be sent and received. The primary difference with the primary coin is that instead of running on their own blockchain, ERC20 tokens are issued on a network that supports smart contracts such as Ethereum or Binance Smart Chain.



EXECUTIVE SUMMARY

This Security Audit Report was prepared by Verichains Lab on Jul 01, 2024. We would like to thank the DOS Labs for trusting Verichains Lab in auditing smart contracts. Delivering high-quality audits is always our top priority.

This audit focused on identifying security flaws in code and the design of the Linear Staking Pool V2 Smart Contracts. The scope of the audit is limited to the source code files provided to Verichains. Verichains Lab completed the assessment using manual, static, and dynamic analysis techniques.

During the audit process, the audit team identified no vulnerable issues in the smart contracts code.



TABLE OF CONTENTS

1. MANAGEMENT SUMMARY	5
1.1. About Linear Staking Pool V2 Smart Contracts	5
1.2. Audit scope.....	5
1.3. Audit methodology	5
1.4. Disclaimer	6
1.5. Acceptance Minute.....	6
2. AUDIT RESULT	7
2.1. Overview	7
2.1.1. LinearStakingPoolProxy contract	7
2.1.2. LinearStakingPool contract	7
2.2. Findings.....	7
3. VERSION HISTORY	8

1. MANAGEMENT SUMMARY

1.1. About Linear Staking Pool V2 Smart Contracts

Linear Staking Pool is a contract designed to incentivize users to stake DOS Labs tokens in exchange for rewards.

1.2. Audit scope

This audit focused on identifying security flaws in code and the design of the Linear Staking Pool V2 Smart Contracts. It was conducted on commit [094f7ca843c6ade769f79eca99361af8b3bd9d7c](https://github.com/DOSLabs/Smart-Contracts/commit/094f7ca843c6ade769f79eca99361af8b3bd9d7c) from git repository link: <https://github.com/DOSLabs/Smart-Contracts/>.

There are 2 files in our audit scope:

- auditsStakingV2/LinearStakingPool.sol
- auditsStakingV2/LinearStakingPoolProxy.sol

1.3. Audit methodology

Our security audit process for smart contract includes two steps:

- Smart contract codes are scanned/tested for commonly known and more specific vulnerabilities using public and RK87, our in-house smart contract security analysis tool.
- Manual audit of the codes for security issues. The contracts are manually analyzed to look for any potential problems.

Following is the list of commonly known vulnerabilities that were considered during the audit of the smart contract:

- Integer Overflow and Underflow
- Timestamp Dependence
- Race Conditions
- Transaction-Ordering Dependence
- DoS with (Unexpected) revert
- DoS with Block Gas Limit
- Gas Usage, Gas Limit and Loops
- Redundant fallback function
- Unsafe type Inference
- Reentrancy
- Explicit visibility of functions state variables (external, internal, private and public)
- Logic Flaws

For vulnerabilities, we categorize the findings into categories as listed in table below, depending on their severity level:

SEVERITY LEVEL	DESCRIPTION
CRITICAL	A vulnerability that can disrupt the contract functioning; creates a critical risk to the contract; required to be fixed immediately.
HIGH	A vulnerability that could affect the desired outcome of executing the contract with high impact; needs to be fixed with high priority.
MEDIUM	A vulnerability that could affect the desired outcome of executing the contract with medium impact in a specific scenario; needs to be fixed.
LOW	An issue that does not have a significant impact, can be considered as less important.

Table 1. Severity levels

1.4. Disclaimer

DOS Labs acknowledges that the security services provided by Verichains, are conducted to the best of their professional abilities but cannot guarantee 100% coverage of all security vulnerabilities. DOS Labs understands and accepts that despite rigorous auditing, certain vulnerabilities may remain undetected. Therefore, DOS Labs agrees that Verichains shall not be held responsible or liable, and shall not be charged for any hacking incidents that occur due to security vulnerabilities not identified during the audit process.

1.5. Acceptance Minute

This final report served by Verichains to the DOS Labs will be considered an Acceptance Minute. Within 7 days, if no any further responses or reports is received from the DOS Labs, the final report will be considered fully accepted by the DOS Labs without the signature.

2. AUDIT RESULT

2.1. Overview

The Linear Staking Pool V2 Smart Contracts was written in `Solidity` language, with the required version to be `^0.8.20`.

2.1.1. LinearStakingPoolProxy contract

A proxy contract, inheriting the `TransparentUpgradeableProxy` contract from the OpenZeppelin library, facilitates easy upgrades to the logic contract. By default, the deployer assumes ownership of the contract, granting them the authority to modify the logic contract by invoking the `upgradeTo` function at any time. This mechanism enables logic contract upgrades without necessitating a change in the proxy contract's address.

2.1.2. LinearStakingPool contract

The core contract incentivizes users to stake DOS Labs tokens for rewards that increase with the amount staked and staking duration. Users can withdraw staked tokens anytime, but early reward claims incur penalties. The specific reward and penalty parameters depend on the chosen staking pool.

The contract owner, initially the caller of the `initialize` function, can add new pools and modify certain pool parameters. However, the penalty rate is fixed and set only when a new pool is added.

2.2. Findings

During the audit process, the audit team identified no vulnerable issues in the contract code.

3. VERSION HISTORY

Version	Date	Status/Change	Created by
1.0	Jul 01, 2024	Public Report	Verichains Lab

Table 2. Report versions history