



Linear Token (LINA) Token Security Audit Report

**Are Your
Funds
Safe?**

Linear Token (LINA)
smart contract audit by
Callisto Network.



Linear Token



Callisto Security Dept.

#AuditedbyCLO

Linear Token (LINA) Token smart contract security audit, conducted by the Callisto Network Security Department during December 2021.

Linear Token (LINA) Token Security Audit Report

Are Your Funds Safe?

Summary

Linear Token (LINA) smart contract security audit report performed by Callisto Security Audit Department.

Platform

Binance Smart Chain.

1. In scope

- <https://bscscan.com/address/0x762539b45a1dcce3d36d080f74d1aed378>
- Token implementation:
<https://bscscan.com/address/0x2ae9952c1038da3dafeba026b96053f24c>

2. Findings

In total, **1 issues** were reported including:

- 0 high severity issues.
- 0 medium severity issue.
- 1 low severity issue.

In total, **1 notes** were reported, including:

- 0 notes.
- 1 owner privileges.

No critical security issues were found.

2.1 Known vulnerabilities of ERC-20 token

Severity: low.

Description:

Lack of transaction handling mechanism issue. **WARNING!** This is a very common issue, and it already caused millions of dollars in losses for lots of token users!

More details [here](#).

Recommendation

Add the following code to the `transfer(_to address, ...)` function:

```
require( _to != address(this) );
```

2.2 Upgradable token contract

Severity: owner privileges.

Description:

This token contract is upgradable. Owner has right to change contract implementation code to add any functions (which may be not secure).

3. Security practices

☒ **Open-source contract.**

☐ **The contract should pass a bug bounty after the completion of the security audit.**

☐ **Public testing.**

☐ **Automated anomaly detection systems.** – NOT IMPLEMENTED. A simple anomaly detection algorithm is recommended to be implemented to detect behavior that is atypical compared to normal for this contract. For instance, the contract must halt deposits in case a

large amount is being withdrawn in a short period of time until the owner or the community of the contract approves further operations.

☐ **Multisig owner account.**

☐ **Standard ERC20-related issues.** – NOT IMPLEMENTED. It is known that every contract can potentially receive an unintended ERC20-token deposit without the ability to reject it even if the contract is not intended to receive or hold tokens. As a result, it is recommended to implement a function that will allow extracting any arbitrary number of tokens from the contract.

☐ **Crosschain address collisions.** ETH, ETC, CLO, etc. It is possible that a transaction can be sent to the address of your contract at another chain (as a result of a user mistake or some software fault). It is recommended that you deploy a “mock contract” that would allow you to withdraw any tokens from that address or prevent any funds deposits. Note that you can reject transactions of native token deposited, but you can not reject the deposits of ERC20 tokens. You can use this source code as a mock contract: [extractor contract source code](#). The address of a new contract deployed using CREATE (0xf0) opcode is assigned following this scheme $\text{keccak256}(\text{rlp}([\text{sender}, \text{nonce}]))$. Therefore you need to use the same address that was originally used at the main chain to deploy the mock contract at a transaction with the **nonce** that matches that on the original chain. *Example: If you have deployed your main contract with address 0x010101 at your 2021th transaction then you need to increase your nonce of 0x010101 address to 2020 at the chain where your mock*

contract will be deployed. Then you can deploy your mock contract with your 2021th transaction, and it will receive the same address as your mainnet contract.

4. Conclusion

The audited smart contract can be deployed. Only low severity issues were found during the audit.

Users should note that the Contract Administrator (owner) can change the code at any time.

It is recommended to adhere to the security practices described in pt. 4 of this report to ensure the contract's operability and prevent any issues that are not directly related to the code of this smart contract.

Appendix

[Smart Contract Audits by Callisto Network.](#)

Miscellaneous

[Why Audit Smart Contracts?](#)

[Our Most Popular Audit Reports.](#)

Trust the Blockchain, Audit the Smart

Contracts.

Follow Callisto's Security Department on [Twitter](#) to get our latest news and updates!

By [tonton](#) | December 30, 2021 | [Smart Contract Audits](#) | Comments Off

Share This Story, Choose Your Platform!



Resources

[FAQ](#)
[Airdrop](#)
[Community](#)
[Guidelines](#)
[Media Kit](#)

Callisto Network

[Partners](#)
[GitHub](#)
[Our Careers](#)
[Contact us](#)

Get Your CLO Today

[Gate.io](#)
[1inch](#)
[SOY Finance](#)
[BitcoinMat](#)

Join the Callistonians



© All rights reserved 2023