LINA Token Contract Audit Report

Preamble

This audit report was undertaken by **BlockchainLabs.nz** for the purpose of providing feedback to **Lina Network**.

It has subsequently been shared publicly without any express or implied warranty.

Solidity contracts were sourced from the Ethereum mainnet at 0xc05d14442a510de4d3d71a3d316585aa0ce32b50 - We would encourage all community members and token holders to make their own assessment of the contracts.

Scope

The following contracts were subject for static, dynamic and functional analyses:

- TokenERC20.sol

Focus areas

The audit report focuses on the following key areas, although this list is not exhaustive.

Correctness

- No correctness defects uncovered during static analysis?
- No implemented contract violations uncovered during execution?
- No other generic incorrect behaviour detected during execution?
- Adherence to adopted standards such as ERC20?

Testability

- Test coverage across all functions and events?
- Test cases for both expected behaviour and failure modes?
- Settings for easy testing of a range of parameters?
- No reliance on nested callback functions or console logs?
- Avoidance of test scenarios calling other test scenarios?

Security

- No presence of known security weaknesses?

- No funds at risk of malicious attempts to withdraw/transfer?
- No funds at risk of control fraud?
- Prevention of Integer Overflow or Underflow?

Best Practice

- Explicit labeling for the visibility of functions and state variables?
- Proper management of gas limits and nested execution?
- Latest version of the Solidity compiler?

Analysis Reports

- Functional Analysis
- Dynamic Analysis
- Gas Consumption
- Test Coverage

Issues

Severity Description

Minor	A defect that does not have a material impact on the contract execution and is likely to be subjective.
Moderate	A defect that could impact the desired outcome of the contract execution in a specific scenario.
Major	A defect that impacts the desired outcome of the contract execution or introduces a weakness that may be exploited.
Critical	A defect that presents a significant security vulnerability or failure of the contract across a range of scenarios.

Minor

- **burn() function doesn’t return any value** - `Best practice`

  burn() function doesn’t return any bool value as it announced in the function definition … View on GitHub

- **Total Supply unnecessary displayed twice** - `Best practice`

  It is not necessary to display totalSupply twice in the contract … View on GitHub

Moderate

- None found

Major

- None found

Critical

- None found

Conclusion

Overall we have not identified any potential vulnerabilities. This contract has a low level risk of LINA being hacked or stolen from the inspected contracts.

Disclaimer

Our team uses our current understanding of the best practises for Solidity and Smart Contracts. Development in Solidity and for Blockchain is an emerging area of software engineering which still has a lot of room to grow, hence our current understanding of best practices may not find all of the issues in this code and design.

We have not analysed any of the assembly code generated by the Solidity compiler. We have not verified the deployment process and configurations of the contracts. We have only analysed the code outlined in the scope. We have not verified any of the claims made by any of the organisations behind this code.

Security audits do not warrant bug-free code. We encourage all users interacting with smart contract code to continue to analyse and inform themselves of any risks before interacting with any smart contracts.