HACKEN

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

Customer: LTO Network
Date: October 15th, 2021

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

This document may contain confidential information about IT systems and the intellectual property of the Customer as well as information about potential vulnerabilities and methods of their exploitation.

The report containing confidential information can be used internally by the Customer, or it can be disclosed publicly after all vulnerabilities are fixed — upon a decision of the Customer.

Document

Name	Smart Contract Code Review and Security Analysis Report for LTO Network.
Approved by	Andrew Matiukhin | CTO Hacken OU
Type	ERC20 token; Token Sale
Platform	Ethereum / Solidity
Methods	Architecture Review, Functional Testing, Computer-Aided Verification, Manual Review
Repository	https://github.com/ltonetwork/lto-erc20-token
Commit	8e6c6619da6784080b9786791a3cc0f9cd0ffc7d
Technical Documentation	YES
JS tests	YES
Timeline	14 SEPTEMBER 2021 - 16 SEPTEMBER 2021
Changelog	16 SEPTEMBER 2021 - Initial Audit 24 SEPTEMBER 2021 - Second Review 15 OCTOBER 2021 - Third Review

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Table of contents

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Introduction

Hacken OÜ (Consultant) was contracted by LTO Network (Customer) to conduct a Smart Contract Code Review and Security Analysis. This report presents the findings of the security assessment of the Customer's smart contract and its code review conducted between September 14th, 2021 - September 16th, 2021.

Second review conducted on September 24th, 2021.

Third review conducted on October 15th, 2021.

Scope

The scope of the project is smart contracts in the repository:

Repository:
https://github.com/ltonetwork/lto-erc20-token
Commit:
8e6c6619da6784080b9786791a3cc0f9cd0ffc7d
Technical Documentation: Yes
JS tests: Yes
Contracts:
ERC20Swap.sol
LTOToken.sol
Migrations.sol
TestToken.sol

We have scanned this smart contract for commonly known and more specific vulnerabilities. Here are some of the commonly known vulnerabilities that are considered:

Category	Check Item
Code review	▪ Reentrancy ▪ Ownership Takeover ▪ Timestamp Dependence ▪ Gas Limit and Loops ▪ DoS with (Unexpected) Throw ▪ DoS with Block Gas Limit ▪ Transaction-Ordering Dependence ▪ Style guide violation ▪ Costly Loop ▪ ERC20 API violation ▪ Unchecked external call ▪ Unchecked math ▪ Unsafe type inference ▪ Implicit visibility level ▪ Deployment Consistency ▪ Repository Consistency ▪ Data Consistency

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Functional review	▪ Business Logics Review ▪ Functionality Checks ▪ Access Control & Authorization ▪ Escrow manipulation ▪ Token Supply manipulation ▪ Assets integrity ▪ User Balances manipulation ▪ Data Consistency manipulation ▪ Kill-Switch Mechanism ▪ Operation Trails & Event Generation

Executive Summary

According to the assessment, the Customer's smart contracts are well-secured.

Insecure	Poor secured	Secured	Well-secured

You are here 

Our team performed an analysis of code functionality, manual audit, and automated checks with Mythril and Slither. All issues found during automated analysis were manually reviewed, and important vulnerabilities are presented in the Audit overview section. All found issues can be found in the Audit overview section.

As a result of the audit, security engineers found **5** low severity issues.

After the second review security engineers found that all issues were fixed.

After the third review security engineers found that part of the contracts were removed (BalanceCopier.sol, ERC20PreMint.sol, FakeWallet.sol, LTOTokenSale.sol), but instead new contracts were added (ERC20Swap.sol, TestToken.sol). Despite that, no new issues were found.

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Severity Definitions

Risk Level	Description
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to assets loss or data manipulations.
High	High-level vulnerabilities are difficult to exploit; however, they also have a significant impact on smart contract execution, e.g., public access to crucial functions
Medium	Medium-level vulnerabilities are important to fix; however, they can't lead to assets loss or data manipulations.
Low	Low-level vulnerabilities are mostly related to outdated, unused, etc. code snippets that can't have a significant impact on execution

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Audit overview

Critical

No critical issues were found.

High

No high severity issues were found.

Medium

No medium severity issues were found.

Low

1. Two of eighteen tests are failing

   There are 18 tests for the project repository and 2 of them are failed, both with the same error: “Cannot read property 'status' of undefined”.

   Recommendation: Please fix failed tests and also ensure that your tests are cover at least 95% of branches.

   Fixed before the second review

2. Conformance to Solidity naming conventions

   Solidity defines a naming convention that should be followed. e.g. constants should be declared UPPER_CASE_WITH_UNDERSCORES.

   Recommendation: Follow the Solidity naming convention.

   Fixed before the second review

3. Excess require check

   While using “Safemath.sub” function which is already checking that subtrahend is less than minuend, so putting additional require statement for the same comparison is excess.

   Recommendation: Please remove excess require statement.

   Fixed before the second review

4. Implicit visibility declaration

   State variables that don’t have explicitly declared visibility is defaulted as the internal, which is not always understandable for reviewers and could cause misunderstanding.

   Recommendation: Please try always to declare visibility explicitly.

   Fixed before the second review

5. A public function that could be declared external

public functions that are never called by the contract should be declared **external** to save gas.

Recommendation: Use the external attribute for functions never called from the contract.

Fixed before the second review

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Conclusion

Smart contracts within the scope were manually reviewed and analyzed with static analysis tools.

The audit report contains all found security vulnerabilities and other issues in the reviewed code.

As a result of the audit, security engineers found **5** low severity issues.

After the second review security engineers found that all issues were fixed.

After the third review security engineers found that part of the contracts were removed (BalanceCopier.sol, ERC20PreMint.sol, FakeWallet.sol, LTOTokenSale.sol), but instead new contracts were added (ERC20Swap.sol, TestToken.sol). Despite that, no new issues were found.

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Disclaimers

Hacken Disclaimer

The smart contracts given for audit have been analyzed in accordance with the best industry practices at the date of this report, in relation to cybersecurity vulnerabilities and issues in smart contract source code, the details of which are disclosed in this report (Source Code); the Source Code compilation, deployment, and functionality (performing the intended functions).

The audit makes no statements or warranties on the security of the code. It also cannot be considered as a sufficient assessment regarding the utility and safety of the code, bug-free status, or any other statements of the contract. While we have done our best in conducting the analysis and producing this report, it is important to note that you should not rely on this report only – we recommend proceeding with several independent audits and a public bug bounty program to ensure the security of smart contracts.

Technical Disclaimer

Smart contracts are deployed and executed on a blockchain platform. The platform, its programming language, and other software related to the smart contract can have vulnerabilities that can lead to hacks. Thus, the audit can't guarantee the explicit security of the audited smart contracts.