VIDMA

Teleport

SMART CONTRACT AUDIT

Project: Teleport Network
Date: August 31st, 2022

TABLE OF CONTENTS

Summary . . . 02

Scope of Work . . . 05

Workflow of the auditing process . . . 06

Structure and organization of the findings . . . 08

Manual Report . . . 10

Low | Unresolved

Gas optimization . . . 10

Test Results . . . 11

Tests written by Vidma auditors . . . 12

SUMMARY

Vidma is pleased to present this audit report outlining our assessment of code, smart contracts, and other important audit insights and suggestions for management, developers, and users.

The smart contract is an ERC20 token with initial balance assigned to the chosen address and without a possibility to mint/burn tokens.

During the audit process, the Vidma team found an informational issue. A detailed summary and the current state are displayed in the table below.

Severity of the issue	Total found	Resolved	Unresolved
Critical	0 issues	0 issues	0 issues
High	0 issues	0 issues	0 issues
Medium	0 issues	0 issues	0 issues
Low	1 issue	0 issues	1 issue
Informational	0 issues	0 issues	0 issues
Total	**1 issue**	**0 issues**	**1 issue**

After evaluating the findings in this report and the final state after fixes, the Vidma auditors can state that the contract is operational and secure. Under the given circumstances, we set the following risk level:

Our auditors are evaluating the initial commit given for the scope of the audit and the last commit with the fixes. This approach helps us adequately and sequentially evaluate the quality of the code. Code style, optimization of the contracts, the number of issues, and risk level of the issues are all taken into consideration. The Vidma team has developed a transparent scoring system presented below.

Severity of the issue	Resolved	Unresolved
Critical	1	10
High	0.8	7
Medium	0.5	5
Low	0.2	0.5
Informational	0	0.1

Please note that the points are deducted out of 100 for each and every issue on the list of findings (according to the current status of the issue). Issues marked as "not valid" are not subject to point deduction.

Based on the **overall result of the audit**, the Vidma audit team grants the following score:

In addition to manual check and static analysis, the auditing team has conducted a number of integrated autotests to ensure the given codebase has an adequate performance and security level.

The test results and the coverage can be found in the accompanying section of this audit report.

Please be aware that this audit does not certify the definitive reliability and security level of the contract. This document describes all vulnerabilities, typos, performance issues, and security issues found by the Vidma audit team. If the code is still under development, we highly recommend running one more audit once the code is finalized.

SCOPE OF WORK

Teleport

Teleport Network provides infrastructure and framework for cross-chain communication. It consists of a decentralized blockchain - Teleport Chain, a cross-chain protocol - XIBC, and developer SDKs for cross-chain dApps integration.

Within the scope of this audit, two independent auditors thoroughly investigated the given codebase and analyzed the overall security and performance of the smart contract.

The audit was conducted on Aug 31st, 2022. The outcome is disclosed in this document.

The scope of work for the given audit consists of the following contract:

- CoinToken.

The source code was taken from the following **source**:
0xf6158bdFE9e013673269b4D1ca468E8eFD77Ca3f#code

Initial commit submitted for the audit:
0xf6158bdFE9e013673269b4D1ca468E8eFD77Ca3f#code

WORKFLOW OF THE AUDITING PROCESS

Vidma audit team uses the most sophisticated and contemporary methods and well-developed techniques to ensure the contract is free of vulnerabilities and security risks. The overall workflow consists of the following phases:

Phase 1: The research phase

Research

After the Audit kick-off, our security team conducts research on the contract's logic and expected behavior of the audited contract.

Documentation reading

Vidma auditors do a deep dive into your tech documentation with the aim of discovering all the behavior patterns of your codebase and analyzing the potential audit and testing scenarios.

The outcome

At this point, the Vidma auditors are ready to kick off the process. We set the auditing strategies and methods and are prepared to conduct the first audit part.

Phase 2: Manual part of the audit

Manual check

During the manual phase of the audit, the Vidma team manually looks through the code in order to find any security issues, typos, or discrepancies with the logic of the contract. The initial commit as stated in the agreement is taken into consideration.

Static analysis check

Static analysis tools are used to find any other vulnerabilities in smart contracts that were missed after a manual check.

The outcome

An interim report with the list of issues.

Phase 3: Testing part of the audit

Integration tests

Within the testing part, Vidma auditors run integration tests using the Truffle or Hardhat testing framework. The test coverage and the test results are inserted in the accompanying section of this audit report.

The outcome

Second interim report with the list of new issues found during the testing part of the audit process.

STRUCTURE AND ORGANIZATION OF THE FINDINGS

For simplicity in reviewing the findings in this report, Vidma auditors classify the findings in accordance with the severity level of the issues. (from most critical to least critical).

All issues are marked as “Resolved” or “Unresolved”, depending on if they have been fixed by Teleport Network or not. The issues with “Not Valid” status are left on the list of findings but are not eligible for the score points deduction.

The latest commit with the fixes reviewed by the auditors is indicated in the “Scope of Work” section of the report.

The Vidma team always provides a detailed description of the issues and recommendations on how to fix them.

Classification of found issues is graded according to 6 levels of severity described below:

Critical

The issue affects the contract in such a way that funds may be lost or allocated incorrectly, or the issue could result in a significant loss.

Example: Underflow/overflow, precisions, locked funds.

High

The issue significantly affects the ability of the contract to compile or operate. These are potential security or operational issues.

Example: Compilation errors, pausing/unpausing of some functionality, a random value, recursion, the logic that can use all gas from block (too many iterations in the loop), no limitations for locking period, cooldown, arithmetic errors which can cause underflow, etc.

Medium

The issue slightly impacts the contract's ability to operate by slightly hindering its intended behavior.

Example: Absence of emergency withdrawal of funds, using assert for parameter sanitization.

Low

The issue doesn't contain operational or security risks, but are more related to optimization of the codebase.

Example: Unused variables, inappropriate function visibility (public instead of external), useless importing of SCs, misuse or disuse of constant and immutable, absent indexing of parameters in events, absent events to track important state changes, absence of getters for important variables, usage of string as a key instead of a hash, etc.

Informational

Are classified as every point that increases onboarding time and code reading, as well as the issues which have no impact on the contract's ability to operate.

Example: Code style, NatSpec, typos, license, refactoring, naming convention (or unclear naming), layout order, functions order, lack of any type of documentation.

MANUAL REPORT

Gas optimization

Low | Unresolved

In contract CoinToken.sol used an old version of OpenZeppelin contract ERC20.sol. As solidity 0.8.x brings safe math operations by default new version of ERC20 from OpenZeppelin provide gas improvements for general functions with help of unchecked math in cases where overflow/underflow is reachless.

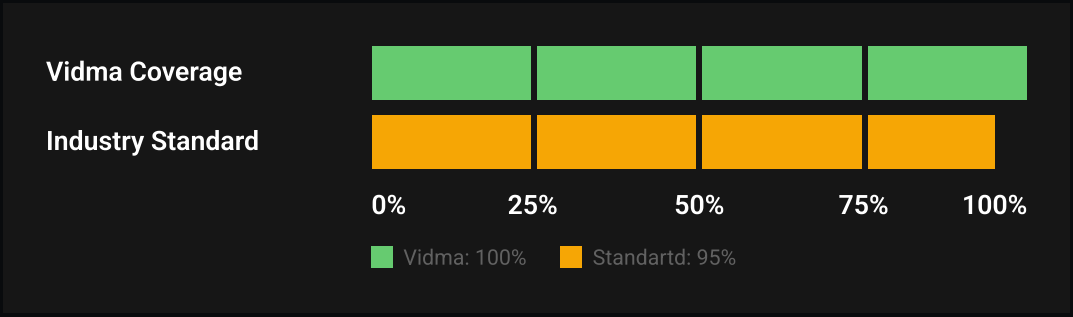
Recommendation:

If redeployment of the contract is desired consider the possibility to use a newer version of OpenZeppelin. Also, all additional logic can be implemented directly on CoinToken.sol without the need to change the standard methods of ERC20.sol.

TEST RESULTS

To verify the security of contract and the performance, a number of integration tests were carried out using the Truffle testing framework.

In this section, we provide tests written by Vidma auditors.



It is important to note that Vidma auditors do not modify, edit or add tests to the existing tests provided in the Teleport Network repository. We write totally separate tests with code coverage of a minimum of 95% to meet the industry standards.

Tests written by Vidma auditors

Test Coverage

File	% Stmts	% Branch	% Funcs	% Lines
contracts\	100.00	100.00	100.00	100.00
CoinToken.sol	100.00	100.00	100.00	100.00
All Files	**100.00**	**100.00**	**100.00**	**100.00**

Test Results

```
Contract: CoinToken
  Teleport Test Cases
    Teleport Token Deploy Test Cases
      ✓ should deploy with correct name
      ✓ should deploy with correct symbol
      ✓ should deploy with correct decimals
      ✓ should deploy with correct initial total supply
      ✓ should deploy with correct token owner balance
    Teleport Token Deploy Test Cases
      ✓ should transfer tokens correctly (67ms)
      ✓ shouldn't transfer tokens to the zero address
      ✓ shouldn't transfer tokens from the zero address
      ✓ shouldn't transfer tokens if transfer amount exceed balance
      ✓ should approve correctly
      ✓ shouldn't approve to the zero address
      ✓ shouldn't approve from the zero address
      ✓ should increase allowance correctly
      ✓ shouldn't increase allowance for zero address
      ✓ shouldn't increase allowance from zero address
      ✓ should decrease allowance correctly (41ms)
      ✓ shouldn't decrease allowance for zero address
      ✓ shouldn't decrease allowance from zero address
```

```
    ✓ shouldn't decrease allowance below zero
    ✓ should transfer tokens from address correctly (66ms)
    ✓ shouldn't transfer tokens from address if amount exceed
      allowance
    ✓ shouldn't transfer tokens from address to the zero address
    ✓ shouldn't transfer tokens from zero address

  23 passing (907ms)
```

We are delighted to have a chance to work with the Teleport Network team and contribute to your company's success by reviewing and certifying the security of your smart contracts.

The statements made in this document should be interpreted neither as investment or legal advice, nor should its authors be held accountable for decisions made based on this document.

Vidma is a security audit company helping crypto companies ensure their code and products operate safely and as intended, enabling founders to sleep soundly at night. We specialize in auditing DeFi protocols, layer one protocols, and marketplace solutions. Our team consists of experienced and internationally trained specialists. Our company is based in Ukraine, known for its strong engineering, cryptography, and cybersecurity culture.

Website: **vidma.io**
Email: **security@vidma.io**

VIDMA