CERTIK

CertiK Audit Report for Frontier

Contents

Contents 1

Disclaimer 2

About CertiK 2

Executive Summary 3

Testing Summary 4

Review Notes 5

- Introduction 5
- Documentation 6
- Summary 6
- Recommendations 6

Findings 7

Exhibit 1 7

Exhibit 2 8

Exhibit 3 9

Exhibit 4 10

Exhibit 5 11

Disclaimer

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Verification Services Agreement between CertiK and Frontier (the "Company"), or the scope of services/verification, and terms and conditions provided to the Company in connection with the verification (collectively, the "Agreement"). This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes without CertiK's prior written consent.

About CertiK

CertiK is a technology-led blockchain security company founded by Computer Science professors from Yale University and Columbia University built to prove the security and correctness of smart contracts and blockchain protocols.

CertiK, in partnership with grants from IBM and the Ethereum Foundation, CertiK's mission of every audit is to apply different approaches and detection methods, ranging from manual, static, and dynamic analysis, to ensure that projects are checked against known attacks and potential vulnerabilities. CertiK leverages a team of seasoned engineers and security auditors to apply testing methodologies and assessments to each project, in turn creating a more secure and robust software system.

CertiK has served more than 100 clients with high quality auditing and consulting services, ranging from stablecoins such as Binance's BGBP and Paxos Gold to decentralized oracles such as Band Protocol and Tellor. CertiK customizes its engineering tool kits, while applying cutting-edge research on smart contracts, for each client on its project to offer a high quality deliverable. For more information: https://certik.io.

Executive Summary

This report has been prepared for **Frontier** to discover issues and vulnerabilities in the source code of their **ERC-20 & Vesting Smart Contracts** as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Dynamic Analysis, Static Analysis, and Manual Review techniques.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

Testing Summary

SECURITY LEVEL

Smart Contract Audit

This report has been prepared as a product of the Smart Contract Audit request by Frontier.

This audit was conducted to discover issues and vulnerabilities in the source code of Frontier's Smart Contracts.

TYPE	Smart Contract
SOURCE CODE	https://github.com/frontierwallet/front-sc
PLATFORM	EVM
LANGUAGE	Solidity
REQUEST DATE	Aug 28, 2020
DELIVERY DATE	Sept 11, 2020
METHODS	A comprehensive examination has been performed using Dynamic Analysis, Static Analysis, and Manual Review.

Review Notes

Introduction

CertiK team was contracted by the Frontier team to audit the design and implementation of their ERC-20 & Vesting smart contracts and its compliance with the EIPs it is meant to implement.

The audited source code link is:

- The audited source code link: https://github.com/frontierwallet/front-sc/tree/98043aa939b846768023357ba8c3d9159639a781

The remediated source code link:

- https://github.com/frontierwallet/front-sc/tree/627fb08ca94e8c1f8b70ae96ce9c05339ae822cb

The goal of this audit was to review the Solidity implementation for its business model, study potential security vulnerabilities, its general design and architecture, and uncover bugs that could compromise the software in production.

The findings of the initial audit have been conveyed to the team behind the contract implementations and the source code is expected to be re-evaluated before another round of auditing has been carried out.

Documentation

The sources of truth regarding the operation of the contracts in scope were extensive. To help aid our understanding of each contract's functionality we also referred to in-line comments and naming conventions.

Summary

The codebase of the project is a typical EIP20 implementation, along with a vesting mechanism.

Although **certain optimization steps** that we pinpointed in the source code mostly referred to coding standards and inefficiencies, **the flaws** that were identified **should be remediated as soon as possible to ensure the security of the contracts.**

The codebase of the project strictly adheres to the standards and interfaces imposed by the OpenZeppelin open-source libraries and as such its typical ERC-20 functions **can be deemed to be secure**.

Recommendations

Overall, the codebase of the contracts should be refactored to assimilate the findings of this report **to achieve a high standard of code quality and security**.

Findings

Exhibit 1

TITLE	TYPE	SEVERITY	LOCATION
Unused Imported Contract	Optimization	Informational	FrontierToken: L3

[INFORMATIONAL] Description:

The imported contract ERC20Pausable remains unused.

Recommendations:

We advise the team to remove unnecessary code.

Alleviations:

The team opted to consider our references and remove the unused `import` statement.

Exhibit 2

TITLE	TYPE	SEVERITY	LOCATION
Ambiguous Variable Naming	Coding Style	Informational	FrontierTokenVesting: L45

[INFORMATIONAL] Description:

The variable day counts minutes, despite what its name implies.

Recommendations:

We advise the team to properly name the constant variables.

Alleviations:

The team opted to consider our references and changed the name of the `constant` variable.

Exhibit 3

TITLE	TYPE	SEVERITY	LOCATION
Redundant Utilization of SafeMath	Optimization	Informational	FrontierTokenVesting: L144

[INFORMATIONAL] Description:

The variable vestingId should never overflow.

Recommendations:

We advise the team to use simple Math operations instead of the SafeMath library.

Alleviations:

No alleviations.

Exhibit 4

TITLE	TYPE	SEVERITY	LOCATION
Conversion to Inequality Comparison	Optimization	Informational	FrontierTokenVesting: L144

[INFORMATIONAL] Description:

When comparing numbers, it is a good practice to check for the edge case. An example would be when comparing unsigned integers that are not equal to zero, instead of checking all the values greater than zero.

Recommendations:

We advise the team to change the `require` statement to the following:

require(_amount != 0, "error message");

Alleviations:

The team opted to consider our references and changed the codebase as described.

Exhibit 5

TITLE	TYPE	SEVERITY	LOCATION
Ambiguous Documentation	Documentation	Informational	Documentation Page

[INFORMATIONAL] Description:

The sum of the number of tokens presented for each month in the `Supply Shock (#)` tab, taking into account the `Initial Circulating Supply` , adds to the `Total Token Supply`. Yet, when adding the amount of the percentages of the `Total Token Supply` for each month in the `Supply Shock (%)` tab, plus the `Initial Circulating Supply`, there is a shortage of 40000 tokens.

Example:

The increase of tokens for month 10 appears to be `881250` , yet the percentage appears to be only `0.88%` of the `Total Supply` (100000000 tokens).

Recommendations:

We advise the team to add more precise percentages in the respective tab, so the "appearance" of the token shortage disappears.

Alleviations:

The team opted to consider our references and changed the documentation as described.