CERTIK

CertiK Audit Report For CoFiX

CertiK Audit Report For

CoFiX

Request Date: 2020-09-24

Revision Date: 2020-09-30

Platform Name: EVM

Contents

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 **CoFiX** (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 **CoFix** to discover issues and vulnerabilities in the source code of their **Smart Contract** 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

File	Vulnerabilities	Result
CoFiXVaultForLP.sol	4 Informational	Pass
CoFiXVaultForTrader.sol	5 Informational	Pass
CoFiXStakingRewards.sol	1 Informational	Pass

Smart Contract Audit
This report has been prepared as a product of the Smart Contract Audit request by CoFiX.
This audit was conducted to discover issues and vulnerabilities in the source code of CoFiX's smart contract.

TYPE	Smart Contract
SOURCE CODE	https://github.com/Computable-Finance/CoFiX/tree/certik-2nd-audit/contracts
PLATFORM	EVM
LANGUAGE	Solidity
REQUEST DATE	Sep 24, 2020
DELIVERY DATE	Sep 30, 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 **CoFiX** team to audit the design and implementation of their **smart contract**.

This audit report is only conducted to examine the **CoFiX** as 3 files detail information are listed below.

The audited source code link is:

- Source Code:
  https://github.com/Computable-Finance/CoFiX/tree/certik-2nd-audit/contracts
  commit 1b04ea8ffcdbd9a5867d6c0358d1610bcbbbf561
- Source Code SHA-256 Checksum
  1) **CoFiXVaultForLP.sol** hash
     e547d51d7e6df0d236af376e2206465057bd1ac77eec1de762e5fc82fa17723b
  2) **CoFiXVaultForTrader.sol** hash
     98f8599818ea9a135b7499c5f6b39e433e7382e3e3ada9b4b5c2e8bb308acb11
  3) **CoFiXStakingRewards.sol** hash
     0840a69facf5141f644a5d8d6fd368af7b9fbc6cb9c3398d6abc17876b5e702d

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 lackluster and **are something we advise to be enriched to aid in the legibility of the codebase as well as project**. To help aid our understanding of each contract's functionality we referred to in-line comments and naming conventions.

These were considered the specification, and when discrepancies arose with the actual code behaviour, we consulted with the **CoFiX** team or reported an issue.

Summary

The codebase of the project is a typical ERC implementation and the locking mechanism of the token is derived from an officially recognized library, specifically from OpenZeppelin.

Certain optimization steps that we pinpointed in the source code mostly referred to coding standards and inefficiencies.

Certain discrepancies between the expected specification and the implementation of it were identified and were relayed to the team, however they pose no type of vulnerability and concern an optional code path that was unaccounted for.

Recommendations

Overall, the codebase of the contracts should be refactored to assimilate the findings of this report, enforce linters and / or coding styles as well as correct any spelling errors and mistakes that appear throughout the code **to achieve a high standard of code quality and security**.

Findings

Exhibit 1

TITLE	TYPE	SEVERITY	LOCATION
Unlock solidity version	Optimization	Informational	CoFiXStakingRewards.sol

[INFORMATIONAL] Description:

All compiler versions utilized throughout the project used the "^" prefix specifier, denoting that a compiler at or above the version included after the specifier should be used to compile the contracts.

Recommendations:

Lock the compiler at a specific version rather than allow a range of compiler versions to be utilized to avoid compiler-specific bugs and be able to identify ones more easily.

Alleviation:

CoFiX resolved in commit 4d45edbf4ed5bfff4414ef4b85e8e6228dbffca3

CoFiXStakingRewards.sol hash

b61e32421b47cce32e644de6301cb94b1e23091bca69077a851c31d1d2067d0c

Exhibit 2

TITLE	TYPE	SEVERITY	LOCATION
State variables that could be declared constant	Optimization	Informational	CoFiXVaultForTrader.sol L35,L36,L37,L39,L41

[INFORMATIONAL] Description:

State variables like “sDecayRate” don't change in the contract, but they all cost gas .

Recommendations:

Add the constant attributes to state variables that never change thus limit gas costs.

Exhibit 3

TITLE	TYPE	SEVERITY	LOCATION
Add “emit event” in important functions	Optimization	Informational	CoFiXVaultForTrader.sol L159, L178 CoFiXVaultForLP.sol L111

[INFORMATIONAL] Description:

Add “emit event” to functions that involve tokens.

Recommendations:

Add “emit event” in the following functions. “distributeReward”, “clearPendingRewardOfCNode”.

Alleviation:

CoFiX resolved in commit e37ce453099be4f4fb89cc8d20bfbff858893a2f

CoFiXVaultForTrader.sol hash

968d282619deaf96396b228906c20ea8d6e713ad40e450cb994435a1c5199854

CoFiXVaultForLP.sol hash

4ba5ca20bf6326c37debbe4df50fe69ab7b6e5acfad7fdb6b642475d192928b0

Exhibit 4

TITLE	TYPE	SEVERITY	LOCATION
Compares to a boolean constant	Optimization	Informational	CoFiXVaultForTrader.sol L67,L74,L160 CoFiXVaultForLP.sol L73,L81,L112

[Informational] Description:

Boolean constants can be used directly and do not need to be compared to true or false.

Recommendations:

Remove the equality to the boolean constant.

Alleviation:

CoFiX resolved in commit 6b52deb9c1edefe4b319e58d474e118846afe71d

CoFiXVaultForTrader.sol hash

968d282619deaf96396b228906c20ea8d6e713ad40e450cb994435a1c5199854

CoFiXVaultForLP.sol hash

4ba5ca20bf6326c37debbe4df50fe69ab7b6e5acfad7fdb6b642475d192928b0

Exhibit 5

TITLE	TYPE	SEVERITY	LOCATION
Simplifying Existing Code	Optimization	Informational	CoFiXVaultForTrader.sol L61, L66, L73 CoFiXVaultForLP.sol L51, L56, L61, L67, L72, L80

[INFORMATIONAL] Description:

The same verification statement appears in many functions.

Recommendations:

Summarize the same verification conditions to “modify” function.

Alleviation:

CoFiX resolved in commit a4d658e87969780e716d5c65f7ee416fa08dbe62

CoFiXVaultForTrader.sol hash

968d282619deaf96396b228906c20ea8d6e713ad40e450cb994435a1c5199854

CoFiXVaultForLP.sol hash

4ba5ca20bf6326c37debbe4df50fe69ab7b6e5acfad7fdb6b642475d192928b0

Exhibit 6

TITLE	TYPE	SEVERITY	LOCATION
Confirm calculation formula	Question	Discussion	CoFiXVaultForTrader.sol L144

[DISCUSSION] Description:

Function “actualMiningAmountAndDensity” corresponds to which calculation formula in the white paper?

Exhibit 7

TITLE	TYPE	SEVERITY	LOCATION
Greater-Than Comparison with Zero	Mathematical Operations	Informational	CoFiXVaultForTrader.sol L159 param pair and rewardTo CoFiXVaultForLP.sol L71 param pool, L111 param to, L115 variable air

[INFORMATIONAL] Description:

When comparing variables of unsigned type, it's more efficient gas-wise, while taking into account that any value other than zero is indeed valid.

Recommendations:

Change the condition to check inequality with zero, as it is more efficient regarding unsigned integer variables.

Alleviation:

CoFiX resolved in commit 40c2d6cedc4b0fbaadfc8de3433c2616662632b6

CoFiXVaultForTrader.sol hash

968d282619deaf96396b228906c20ea8d6e713ad40e450cb994435a1c5199854

CoFiXVaultForLP.sol hash

4ba5ca20bf6326c37debbe4df50fe69ab7b6e5acfad7fdb6b642475d192928b0

Exhibit 8

TITLE	TYPE	SEVERITY	LOCATION
Question about the function “currentCoFiRate”	Question	Discussion	CoFiXVaultForTrader.sol L90

[DISCUSSION] Description:

Why is the value of White Paper 7.1 inconsistent with the function “currentCoFiRate” model?

Exhibit 9

TITLE	TYPE	SEVERITY	LOCATION
Usage of function “stakeForOther”	Question	Discussion	CoFiXStakingRewards.sol L109

[DISCUSSION] Description:

What’s the purpose of the function stakeForOther?

Exhibit 10

TITLE	TYPE	SEVERITY	LOCATION
Usage of function “addReward”	Question	Discussion	CoFiXStakingRewards.sol L168

[DISCUSSION] Description:

What’s the purpose of the function addReward? Who will call this function?