16th November 2020

SMART CONTRACT AUDIT REPORT

-

version v2.0

Smart Contract Security Audit and General Analysis

HAECHI AUDIT

Table of Contents

0 Issues (0 Critical, 0 Major, 0 Minor) Found

About HAECHI AUDIT

HAECHI AUDIT is a global leading smart contract security audit and development firm operated by HAECHI LABS. HAECHI AUDIT consists of professionals with years of experience in blockchain R&D and provides the most reliable smart contract security audit and development services.

So far, based on the HAECHI AUDIT's security audit report, our clients have been successfully listed on the global cryptocurrency exchanges such as Huobi, Upbit, OKEX, and others.

Our notable portfolios include SK Telecom, Ground X by Kakao, and Carry Protocol while HAECHI AUDIT has conducted security audits for the world's top projects and enterprises.

Trusted by the industry leaders, we have been incubated by Samsung Electronics and awarded the Ethereum Foundation Grants and Ethereum Community Fund.

Contact : audit@haechi.io
Website : audit.haechi.io

01. Introduction

This report was written to provide a security audit for the 1Inch.exchange smart contract. HAECHI AUDIT conducted the audit focusing on whether 1Inch.exchange smart contract is designed and implemented in accordance with publicly released information and whether it has any security vulnerabilities.

The issues found are classified as CRITICAL, MAJOR, MINOR or TIPS according to their severity.

CRITICAL	Critical issues are security vulnerabilities that MUST be addressed in order to prevent widespread and massive damage.
MAJOR	Major issues contain security vulnerabilities or have faulty implementation issues and need to be fixed.
MINOR	Minor issues are some potential risks that require some degree of modification.
TIPS	Tips could help improve the code's usability and efficiency

HAECHI AUDIT advises addressing all the issues found in this report.

02. Summary

The code used for the audit can be found at GitHub (https://github.com/CryptoManiacsZone/1inch-contract/). The last commit for the code audited is at "092204000611366626d596a0933439e5f6520ec40".

Issues

HAECHI AUDIT has 0 Critical Issues, 0 Major Issues, and 0 Minor Issue; also, we included 2 Tip category that would improve the usability and/or efficiency of the code.

Severity	Issue	Status
TIPS	RevertReasonParser cannot decode empty Error(string) type error message	(Found - v1.0) (Resolved - v2.0)
TIPS	Add possibility to parse unknown error in RevertReasonParser for clarity	(Found - v1.0) (Resolved - v2.0)
Notice	OneInchExchange cannot be unpaused	(Found - v1.0) (Acknowledged - v2.0)

03. Overview

Contracts Subject to Audit

- OneInchExchange.sol
- RevertReasonParser.sol
- UniERC20.sol
- GasDiscountCalculator.sol
- OneInchFlags.sol

Roles

The 1Inch.exchange Smart contract has the following authorizations:

- **Owner**

The features accessible by each level of authorization is as follows:

Role	Functions

Owner	• OneInchExchange
	  ○ rescueFunds()
	  ○ pause()
	• Ownable
	  ○ renounceOwnership()
	  ○ transferOwnership()

Notice

- **OneInchExchange cannot be unpaused [Acknowledged - v2.0]**

  OneInchExchange is inheriting a Pausable contract which enables to check if current status is paused or not. And the only function that is affected by current paused state is the OneInchExchange#swap(). But in OneInchExchange, only pause() function that makes a contract to pause is available and the external function to unpause the contract does not exist. Which makes the “paused” state an actual “closed” state which is irreversible. Although this could lead to contract failure, it can be simply fixed by redeploying the contract.

Update

[v2.0] - 1Inch.exchange team has responded that they are aware of this and this function will be renamed to shutdown() in further updates

04. Issues Found

TIPS : RevertReasonParser cannot decode empty Error(string) type error message (Found - v1.0) (Resolved - v2.0) TIPS

Currently, RevertReasonParser only parses errors with a message since it assumes that length of error message is not zero. If the error message is null, total length of error reason will be 68 not 100 which is not parsable with the current statement.

Recommendation

Enable parsing for data length is larger or equal to 68 not 100.

Update

[v2.0] - 1Inch.exchange team already fixed this issue on a deployed contract by reducing data.length requirement to 68. And also they have addressed the issue of solidity that does not return proper length on empty error[1].

[1] https://github.com/ethereum/solidity/issues/10170

TIPS: Add possibility to parse unknown error in RevertReasonParser for clarity (Found - v1.0)(Resolved - v2.0) TIPS

Currently, only revert with Error(string) type message can be parsed with RevertReasonParser, and error without Error(string) signature will be hard to decode.

Recommendation

Prefix a string "UNKNOWN ERROR" or something similar to differentiate the unknown error from Error(string) type error.

Also, Panic(uint256) type error messages will be introduced on solidity 0.8[2] it can be adopted in RevertReasonParser to be future proof.

Update

[v2.0] - 1Inch.exchange team already fixed this issue on a deployed contract by parsing Panic(uint256) type error and returning Unknown() when it cannot be parsed. And also they are planning to improve the Unknown() error message by hexifying the unknown error message .

[2] https://solidity.ethereum.org/2020/10/28/solidity-0.8.x-preview/

05. Disclaimer

This report is not an advice on investment, nor does it guarantee adequacy of a business model and/or a bug-free code. This report should be used only to discuss known technical problems. The code may include problems on Ethereum that are not included in this report. It will be necessary to resolve addressed issues and conduct thorough tests to ensure the safety of the smart contract.