31th MARCH 2021

Stacker Ventures

SMART CONTRACT AUDIT REPORT

-

version v2.0

Smart Contract Security Audit and General Analysis

HAECHI AUDIT

Table of Contents

2 Issues (1 Critical, 0 Major, 1 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 StackerVC smart contract. HAECHI AUDIT conducted the audit focusing on whether StackerVC 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/jack0x-tech/StackerVC_VentureFund001/tree/fund-2
    - Commit Hash : b0192cf354537c1b6a7dc3257066c086624e60fb

Update

[v2.0] - new commit(d6cd3ea1b4257e742e3abd48aa6f6df3800fa11f) has fixed 1 Critical, 1 Minor issues and applied 2 TIPS.

Issues

HAECHI AUDIT has 1 Critical Issues, 0 Major Issues, and 1 Minor Issue; also, we included 3 Tips that would improve the usability and/or efficiency of the code.

Severity	Issue	Status
CRITICAL	1inch swap() function will enable transfer	(Found - v1.0) (Resolved - v2.0)
MINOR	does not check if internal transaction succeeded	(Found - v1.0) (Resolved - v2.0)
TIPS	Wrong error message prefix	(Found - v1.0) (Resolved - v2.0)
TIPS	use type(uint256).max instead of declaring constant variables	(Found - v1.0) (Resolved - v2.0)
TIPS	use contract.function.selector instead of declaring constant variables	(Found - v1.0)
Notice	governance can transfer the underlying asset of FarmTreasuryV1	(Found - v1.0)

03. Overview

Contracts Subject to Audit

- FarmBossV1.sol
- FarmBossV1_TEST.sol
- FarmBossV1_USDC.sol
- FarmBossV1_WBTC.sol
- FarmBossV1_WETH.sol
- FarmTokenV1.sol
- FarmTreasuryV1.sol
- FarmTreasuryV1_ETH.sol

Notice

- **governance can transfer the underlying asset of FarmTreasuryV1**
  FarmTreasuryV1 contract has a rescue() function to withdraw tokens owned by the contract. Although this function won't make any trouble in most cases, if governance withdraws the underlying asset of the contract, it will result in losses to depositors.

04. Issues Found

CRITICAL : 1inch swap() function will enable transfer (Found - v.1.0) (Resolved - v2.0)

CRITICAL

Problem Statement

FarmBoss contracts limit functions that can be called by farmers. To be safe from farmers withdrawing assets, _checkContractAndFn() does not allow transfer/approve function's signature. However, all FarmBoss contract whitelists 1inch exchange's swap function and approved the maximum amount to 1inch exchange. Since swap function supports sending swap destination token to non-msg.sender address using SwapDescription.dstReceiver, it will open the vulnerability that enables farmers transfering the asset.

Recommendation

Do not allow swap functions to be called by farmers.

Update

The StackerVC team has fixed the issue by removing 1inch functionality.

MINOR : does not check if internal transaction succeeded (Found - v1.0) (Resolved - v2.0)

MINOR

Problem Statement

FarmBoss#govExecute/farmerExecute function calls _executes function but does not check if the contract call was successful. This may not open any vulnerabilities but it may result in confusing results.

Recommendation

revert when _execute() function returns false on first return value.

Update

The StackerVC team has confirmed that they know of this issue and this is intended behavior.

TIPS : Wrong error message prefix (Found - v1.0) (Fixed - v2.0)

TIPS

FarmBoss#rescue() function has require message prefixed with “FARMTREASURY”

TIPS : use type(uint256).max instead of declaring constant variables (Found - v.1.0) (Fixed - v2.0)

TIPS

Instead of declaring MAX_UINT256 constant variable, use type(uint256).max[1] which is supported in solidity natively.

TIPS : use contract.function.selector instead of declaring constant variables (Found - v.1.0)

TIPS

Instead of declaring a function selectors as a bytes4 variable, use contract.function.selector[2] to increase clarity and to avoid stack too deep error.

[1] https://docs.soliditylang.org/en/v0.6.12/types.html#integers
[2] https://docs.soliditylang.org/en/v0.6.12/types.html#function-types

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.