RAPID LABS

SMART CONTRACT SECURITY AUDIT

Final report **Plan: Simple**

Revert Finance

July 2022

RL@RAPIDLABS.FINANCE RAPIDLABS.FINANCE

TABLE OF CONTENTS

1. Introduction 3

2. Analyzed Contracts 3

3. Audit Process 3

3.1 Auto-analysis 3

3.2 Expert audit 3

4. Known issues checked 4

5. Issue Classification 7

6. Issues 7

6.1 High risk issues 7

6.2 Medium risk issues 7

6.3 Low risk issues 7

7. Conclusion 9

8. Disclaimer 10

INTRODUCTION

The report has been prepared for Revert Finance. Revert Compoundor protocol allows for the automation through awarding executors (compoundors) a small fee to compensate for their gas costs, and a simple mechanism that incentivizes the compounding of positions as close to optimal as possible. The code is available in the Github repository. The code was checked in the cee8623 commit.

Name	Revert Finance
Audit date	2022-07-25 - 2022-07-29
Language	Solidity
Platform	Ethereum

ANALYZED CONTRACTS

Name	Address
ICompoundor	https://github.com/revert-finance/compoundor/blob/cee8623433aa8567090e9a64fc4a4c42ed9b24d2/contracts/ICompoundor.sol
Compoundor	https://github.com/revert-finance/compoundor/blob/cee8623433aa8567090e9a64fc4a4c42ed9b24d2/contracts/Compoundor.sol

AUDIT PROCESS

Our audit structure consists of two stages:

Auto-analysis

- Our automated tools allow us to scan smart contract code and find potential issues
- We hand pick and verify all the issues found by the tools

Expert audit

- Manual analysis of potential issues and vulnerabilities
- Contract code is reviewed thoroughly

KNOWN ISSUES CHECKED

Title	Result
Unencrypted Private Data On-Chain	✓ passed
Code With No Effects	✓ passed
Message call with hardcoded gas amount	✓ passed
Typographical Error	✓ passed
DoS With Block Gas Limit	✓ passed
Presence of unused variables	✓ passed
Incorrect Inheritance Order	✓ passed
Requirement Violation	✓ passed
Weak Sources of Randomness from Chain Attributes	✓ passed

Shadowing State Variables	✓ passed
Incorrect Constructor Name	✓ passed
Block values as a proxy for time	✓ passed
Authorization through tx.origin	✓ passed
DoS with Failed Call	✓ passed
Delegatecall to Untrusted Callee	✓ passed
Use of Deprecated Solidity Functions	✓ passed
Assert Violation	✓ passed
State Variable Default Visibility	✓ passed
Reentrancy	✓ passed
Unprotected SELFDESTRUCT Instruction	✓ passed
Unprotected Ether Withdrawal	✓ passed
Unchecked Call Return Value	✓ passed
Floating Pragma	✕ not passed
Outdated Compiler Version	✓ passed

Integer Overflow and Underflow	✓ passed
Function Default Visibility	✓ passed

ISSUE CLASSIFICATION

High risk	Issues leading to assets theft, locking or any other loss of assets or leading to contract malfunctioning.
Medium risk	Issues that can trigger a contract failure of malfunctioning.
Low risk	Issues that do now affect contract functionality. For example, unoptimised gas usage, outdated or unused code, code style violations, etc.

ISSUES

High risk issues

No issues were found

Medium risk issues

No issues were found

Low risk issues

1. Unused import (ICompoundor)

The IERC20Metadata import is not used in this contract.

2. Constructor lacks validation of input parameters (Compoundor)

The contract constructor does not check the addresses _weth, _factory, _nonfungiblePositionManager and _swapRouter against a null address.

3. Unused variable (Compoundor)

The weth variable is not used in the contract.

4. Floating Pragma (Compoundor)

Contracts should be deployed with the same compiler version and flags that they have been tested with thoroughly. Locking the pragma helps to ensure that contracts do not accidentally get deployed using, for example, an outdated compiler version that might introduce bugs that affect the contract system negatively.

CONCLUSION

Revert Finance ICompoundor, Compoundor contracts were audited. 4 low risk issues were found.

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 Services Agreement, or the scope of services, and terms and conditions provided to the Company in connection with 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 RapidLabs prior written consent.

This report is not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any "product" or "asset" created by any team or project that contracts RapidLabs to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

RAPID LABS

YOUR PROJECT SECURED

RL@RAPIDLABS.FINANCE

RAPIDLABS.FINANCE