ConverterV2 Audit Report

Version 1.0.0

Serial No. 2021070900022025

Presented by Fairyproof

July 9, 2021

01. Introduction

This document includes the results of the audit performed by the Fairyproof team on the ConverterV2 project, at the request of the Converter team.

Audited Code's Github Repository:

N/A

Audited Code's Github Commit Number:

N/A

Audited Source Files' HECO Onchain Address:

OperatableMsg.sol:
https://hecoinfo.com/address/0x667C1C347aD1b895A846221f571C720523036Fa5

AutoStakingStrategyFactory.sol:
https://hecoinfo.com/address/0x8EBf83141b6308cd297e184dcF205173d8b72D24

AutoStakingVaultFactory.sol:
https://hecoinfo.com/address/0x3c3D70806752E56aa906B520Eff672EFC5A4c125

BaseJumpRateModelV2.sol:

https://hecoinfo.com/address/0xc5C0E5630634845E2FaCd7204c597C485E3C9dF4
BankConfig.sol:
https://hecoinfo.com/address/0x71f2e38CfcbB5273fC1F15C539085e6644DAE534

StrategyLiquidate.sol:

https://hecoinfo.com/address/0xa737231b37424CbAcAB8bb633C565ed06f0790C3

Bank.sol:

https://hecoinfo.com/address/0x6Baf18123c8eD23F7CEcFe878B339134006bCbf4

SwapGoblin.sol:

https://hecoinfo.com/address/0x9B7FA6919E5c0B669cDe39cEAB4bD8AFBc543a57

SwapStrategyAddTwoSidesOptimal.sol:

https://hecoinfo.com/address/0x2A69aaB2b59af6ffb6fA80f120B99eAe949d3f09

SwapStrategyWithdrawMinimizeTrading.sol:

https://hecoinfo.com/address/0x190CaEFF9Dc74bD0Ce41A59AFb6739E259E2782d

Audited Source Files:

The calculated SHA-256 values for the initial files are as follows:

```
AutoRewardStrategy.sol:
0xfc86d00d025af35999c1b25d79c6c75fdce38435ba18d62139e7e339f0263ad8
```

AutoStakingStrategyFactory.sol:
0x097cc1ab7b86dc6a89ff17d75c5b7b234a4b44fa6dbe2d87ab7978abf98409ee

AutoStakingVault.sol:
0x064be81824e33613bc34af4a106588143170cf4bb7f9d76a8d660096c1afe932

AutoStakingVaultFactory.sol:
0x8681dbb3594cfaee59bf3ae130db5800340286d033712450a3f57fe1312012a9

Bank.sol:
0xb19f04ea8eb718e796a69d314507800a4f188cda19c93f158beccfcd3cf82a58

BankConfig.sol:
0x529d0a05463454a821663121c03c544add8cf41e34cf0343ad09e4dc3d2e85a4

BaseJumpRateModelV2.sol:
0x787c73ff0746ec6e8188ddc4be13c62dfb814e32aacf69c7af02034494829500

CToken.sol:
0xc61553e5f33522767b64f33fb02a872154b51850 2ddf04ba1623a8bf2a408a78

CTokenFactory.sol:
0x1f4ad05fd4420f7cbbab10e135541a22f89f41c75fcfdc85eee3dc9704a5ec1c

CheckOper.sol:
0x06288b5ba91c551d92c300af67e00e6dd72f31d2e4fcc49bdda2736934f26071

Goblin.sol:
0x2f11d3a0014f370cb5efe11050cdd0b5d8d00e739727cd02bdb7615c8f3ba79d

Migrations.sol:
0xa41cc0e1d6fb5483850fbd1f461edb50695ec0f136a0887938d98090eba2dcc6

Operatable.sol:
0x2445acf83bfcbf7fc371718d6f4fc188343e6acc7e6cbeb5164b65e0effdfcb1

OperatableMsg.sol:
0xf525755bdba22d26dbe5da066be21de81c00ee62ecd495e910dce8283b647d85

Strategy.sol:
0xe68de13cba9da71ab725b7204649125f88681e78ccb64143f511937aeb72650c

StrategyLiquidate.sol:
0xf9127959cf2b5fddac914bcf2339c300 49edfc3e8013c2e83ac83382882a2eeb

SwapGoblin.sol:
0xe4b005ac4d06e9deaf74bd69441d9da02654f0bd4464628ba489d1dfe54d3b41

SwapMining.sol:
0xfa2ff582864c18768174deb7785db85b3a3fc868ebbddba3db3fcb67b4de89f0

SwapStrategyAddTwoSidesOptimal.sol:
0x769e9c9815d1a270af4e11f111f6452ac92f471f695f7042b1f0e5012ca7da88

SwapStrategyWithdrawMinimizeTrading.sol:
0x52aee72db91e5c31cf1099961dc6f968b8f09062b9a733fe40c4322eeaddf490

IBank.sol:
0x1b8b8d6c88eb118571e6af72b551dff9d3397dbf783ab10121bb43fb206c7692

IBankConfig.sol:
0xa978642769eb82918b7da1d0de02f0869db173816570cabae14d717d5fe14d3

IMasterChef.sol:
0xb3dedb66e6b6e0d5958fbe36ae069cbb9eca90f29a7e9b9d2f42663d82e42181

IOperContract.sol:
0x87ed721c0cf4d9ec34a7131116b16fdb5420f3be9dcb2ef82eec99ee1553b9b0

IStakingRewards.sol:
0xc37597e8cbc7f3bb52a6bfc2d6c50c360946f39d223246012b062f4bf267ad02

ISwapMining.sol:
0xb4a111cef44212d4246f3c3f35827f7f253ec4057291c04b4c0a31039011c2cb

IWETH.sol:
0x8697c46038ba08490df8dad50368eae4585d98a7376657f6f89aab80caaf8a81

IWHT.sol:
0xc67bd870881d33770152cb4da7b1b5530c7894429aa556951bff6cd3cdc34e66

InterestModel.sol:
0xe200c2b177569ce9f6a5c381dd60881619386e1f8fe994ee6fb203eb38cfcba5

IRewards.sol:
0xb13e0fc2906b3b60f8d65bb4652f3e54fd9a6faa1592abb0e05bb2272d3a6c98

IERC20.sol:
0x0673f53effab31117f1876ea0cc5b5ff24778c719f41f2f75f5a8fd2dbb2bd41

IUniswapV2Callee.sol:
0x33c2a13cd06a2dd56341f39375c8cd9ef2008b4d2a15486bd59e54137509a2d3

IUniswapV2ERC20.sol:
0x7bc1216bd1027bbc982feb8627ec0ad68fb05ad58d4c16ff96a5c7714c865c8d

IUniswapV2Factory.sol:
0x20f0e8aef66d15b13f2e0a7d327fe16af76376281abed22ec0be1dfdf85f9954

IUniswapV2Pair.sol:
0x942b0ecd9f18fa84dd6e80b49f4b1597c7d4e392538c52eb41e1e86eb86e18f5

IUniswapV2Router02.sol:
0x780e3d0f29594df54b1cf10796a0bf247bbbb5d8d1e39475624a7b2d3f50129c

IController.sol:
0xfce2bf8d2b6868d91bee26b8b2e72a102a31e4649ae3a8346194aa2173340971

IStrategy.sol:
0xe1e1bbf94e4f34a621388a1b0ab08c6e27f9e7473ef1530f600a8a5fd1c1c21f

IVault.sol:
0xfb5def0c34364fc54c878022bfe85e14a68a3ece064b76fae1e9d24e403803f7

TransferHelper.sol:

```
0x159794c3c320ae7593f33089ac99f626668cc54d070c936ea93da7867db24f57

UniMath.sol:
0x7cb391764a5846a706d0ed01c4ed1c06a9aa50f7b0ae7f76228c106b5e15876c
```

The source files audited include all the files with the extension "sol" as follows:

```
contracts/
├── AutoRewardStrategy.sol
├── AutoStakingStrategyFactory.sol
├── AutoStakingVault.sol
├── AutoStakingVaultFactory.sol
├── Bank.sol
├── BankConfig.sol
├── BaseJumpRateModelV2.sol
├── CToken.sol
├── CTokenFactory.sol
├── CheckOper.sol
├── Goblin.sol
├── Migrations.sol
├── Operatable.sol
├── OperatableMsg.sol
├── Strategy.sol
├── StrategyLiquidate.sol
├── SwapGoblin.sol
├── SwapMining.sol
├── SwapStrategyAddTwoSidesOptimal.sol
├── SwapStrategyWithdrawMinimizeTrading.sol
├── interfaces
|   ├── IBank.sol
|   ├── IBankConfig.sol
|   ├── IMasterChef.sol
|   ├── IOperContract.sol
|   ├── IStakingRewards.sol
|   ├── ISwapMining.sol
|   ├── IWETH.sol
|   ├── IWHT.sol
|   ├── InterestModel.sol
|   ├── depoist
|   |   └── IRewards.sol
|   ├── uniswap
|   |   ├── IERC20.sol
|   |   ├── IUniswapV2Callee.sol
|   |   ├── IUniswapV2ERC20.sol
|   |   ├── IUniswapV2Factory.sol
|   |   ├── IUniswapV2Pair.sol
|   |   └── IUniswapV2Router02.sol
|   └── yearn
|       ├── IController.sol
|       ├── IStrategy.sol
|       └── IVault.sol
└── lib
    ├── TransferHelper.sol
    └── UniMath.sol
```

The goal of this audit is to review ConverterV2's solidity implementation for its leveraged mining function, study potential security vulnerabilities, its general design and architecture, and uncover bugs that could compromise the software in production.

We make observations on specific areas of the code that present concrete problems, as well as general observations that traverse the entire codebase horizontally, which could improve its quality as a whole.

— Disclaimer

Note that as of the date of publishing, the contents of this report reflect the current understanding of known security patterns and state of the art regarding system security. You agree that your access and/or use, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your sole risk.

The review does not extend to the compiler layer, or any other areas beyond the programming language, or other programming aspects that could present security risks. If the audited source files are smart contract files, risks or issues introduced by using data feeds from offchain sources are not extended by this review either.

Given the size of the project, the findings detailed here are not to be considered exhaustive, and further testing and audit is recommended after the issues covered are fixed.

To the fullest extent permitted by law, we disclaim all warranties, expressed or implied, in connection with this report, its content, and the related services and products and your use thereof, including, without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement.

We do not warrant, endorse, guarantee, or assume responsibility for any product or service advertised or offered by a third party through the product, any open source or third-party software, code, libraries, materials, or information linked to, called by, referenced by or accessible through the report, its content, and the related services and products, any hyperlinked websites, any websites or mobile applications appearing on any advertising, and we will not be a party to or in any way be responsible for monitoring any transaction between you and any third-party providers of products or services.

FOR AVOIDANCE OF DOUBT, THE REPORT, ITS CONTENT, ACCESS, AND/OR USAGE THEREOF, INCLUDING ANY ASSOCIATED SERVICES OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, INVESTMENT, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

— Methodology

The above files' code was studied in detail in order to acquire a clear impression of how the its specifications were implemented. The codebase was then subject to deep analysis and scrutiny, resulting in a series of observations. The problems and their potential solutions are discussed in this document and, whenever possible, we identify common sources for such problems and comment on them as well.

The Fairyproof auditing process follows a routine series of steps:

1. Code review that includes the following
   i. Review of the specifications, sources, and instructions provided to Fairyproof to make sure

we understand the size, scope, and functionality of the project's source code.
ii. Manual review of code, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
iii. Comparison to specification, which is the process of checking whether the code does what the specifications, sources, and instructions provided to Fairyproof describe.

2. Testing and automated analysis that includes the following:
i. Test coverage analysis, which is the process of determining whether the test cases are actually covering the code and how much code is exercised when we run the test cases.
ii. Symbolic execution, which is analyzing a program to determine what inputs cause each part of a program to execute.
3. Best practices review, which is a review of the source code to improve maintainability, security, and control based on the established industry and academic practices, recommendations, and research.

— Structure of the document

This report contains a list of issues and comments on all the above source files. Each issue is assigned a severity level based on the potential impact of the issue and recommendations to fix it, if applicable. For ease of navigation, an index by topic and another by severity are both provided at the beginning of the report.

— Documentation

For this audit, we used the following sources of truth about how the ConverterV2 system should work:

http://converter.finance

whitepaper

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

— Comments from Auditee

No vulnerabilities with critical, high, medium or low-severity were found in the above source code.

02. About Fairyproof

Fairyproof is a leading technology firm in the blockchain industry, providing consulting and security audits for organizations. Fairyproof has developed industry security standards for designing and deploying blockchain applications.

03. Introduction to ConverterV2

Converter.Finance allocates users' deposited assets to the mining pools with max returns and can use borrowed assets leverage users' returns.

Note: this audit only covered V2 version's contract files.

04. Major functions of audited code

The audited code implements the following functions:

- leveraged mining
- users deposit a single token A and the token will automatically be converted to an LP
- an LP can be deposited into V1's aggregator vaults to do liquidity mining
- rewards earned from liquidity mining in V1's aggregator vaults can be reinvested in V1's vaults
- rewarded CON tokens obtained in V1's vaults can be reinvested

Attention: when users put a large amount of assets into a mining pool in a single transaction, it may cause huge slippages, therefore causing huge volatility in users' investment returns.

05. Key points in audit

During the audit Fairyproof worked closely with the Converter team and reviewed possible vulnerabilities in leveraging and liquidity mining.

06. Coverage of issues

The issues that the Fairyproof team covered when conducting the audit include but are not limited to the following ones:

- Re-entrancy Attack
- DDos Attack
- Integer Overflow
- Function Visibility
- Logic Vulnerability
- Uninitialized Storage Pointer
- Arithmetic Precision
- Tx.origin
- Shadow Variable
- Design Vulnerability
- Token Issurance
- Asset Security
- Access Control

07. Severity level reference

Every issue in this report was assigned a severity level from the following:

Critical severity issues need to be fixed as soon as possible.

High severity issues will probably bring problems and should be fixed.

Medium severity issues could potentially bring problems and should eventually be fixed.

Low severity issues are minor details and warnings that can remain unfixed but would be better fixed at some point in the future.

08. Major areas that need attention

Based on the provided souce code the Fairyproof team focused on the possible issues and risks related to the following functions or areas.

- Integer Overflow/Underflow

We checked all the code sections, which have arithmetic operations and might introduce integer overflow or underflow if no safe libraries are used. All of them use safe libraries.

We didn't find issues or risks in these functions or areas at the time of writing.

- Setting of Transaction Fees

We checked whether or not the transaction fees were set properly.

We didn't find issues or risks in these functions or areas at the time of writing.

- Staking and Reward

We checked whether or not the reward for staking was calculated correctly and whether or not users could withdraw their rewards.

We didn't find issues or risks in these functions or areas at the time of writing.

- Access Control

We checked each of the functions that can modify a state, especially those functions that can only be accessed by "owner".

We didn't find issues or risks in these functions or areas at the time of writing.

- Token Issurance

We checked whether or not the contract files can mint tokens at will.

We didn't find issues or risks in these functions or areas at the time of writing.

- State Update

We checked some key state variables which should only be set at initialization.

We didn't find issues or risks in these functions or areas at the time of writing.

- Asset Security

We checked whether or not all the functions that transfer assets are safely hanlded.

We didn't find issues or risks in these functions or areas at the time of writing.

- Miscellaneous

We didn't find issues or risks in other functions or areas at the time of writing.

09. List of issues by severity

A. Critical

- N/A

B. High

- N/A

C. Medium

- N/A

D. Low

- N/A

10. List of issues by source file

- N/A

11. Issue descriptions

- N/A

12. Recommendations to enhance the overall security

We list some recommendations in this section. They are not mandatory but will enhance the overall security of the system if they are adopted.

- N/A