Converter Audit Report

Version 1.0.0

Serial No. 2021031600022019

Presented by Fairyproof

March 16, 2021

01. Introduction

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

The audited code can be found in the public Converter Github repository, and the version used for this report is commit

48aaf97f9ece5e684ba7b8a5a873bdda14e8c008

The audited contract files' onchain address: N/A (not deployed yet at the time of writing)

The contract files audited include all the files with the extension "sol" under the `contracts` directory and its sub-directories but exclude the files under the `mock` directory. The audited files are as follows:

```
contracts/
├── ContractWhiteList.sol
├── Migrations.sol
├── Operatable.sol
├── controllers
│   └── Controller.sol
├── fee
│   └── FeeBurn.sol
├── interfaces
│   ├── IMasterChef.sol
│   ├── compound
│   │   ├── CTokenInterfaces.sol
│   │   ├── ComptrollerInterface.sol
│   │   ├── ICEther.sol
│   │   └── InterestRateModel.sol
│   ├── depoist
│   │   └── IRewards.sol
│   ├── mdex
│   │   ├── IMDexRouter.sol
│   │   ├── IMdexPair.sol
│   │   └── ISwapMining.sol
│   ├── weth
│   │   └── WETH9.sol
│   └── yearn
│       ├── IController.sol
│       ├── IStrategy.sol
│       └── IVault.sol
├── lib
│   ├── Const.sol
│   └── TransferHelper.sol
├── strategies
│   ├── AbstractLPStrategy.sol
│   ├── AbstractStakeLPStrategy.sol
│   ├── AbstractStakeSingleStrategy.sol
```

```
|   ├── CompleteCToken.sol
|   ├── CompoundInteractor.sol
|   ├── CompoundStrategy.sol
|   ├── MdexBoardroomLPStrategy.sol
|   ├── MdexBoardroomSingleStrategy.sol
|   ├── ProfitNotifier.sol
|   ├── StrategyLP.sol
|   └── StrategySingle.sol
├── token
|   ├── DAOPool.sol
|   ├── IRewardPool.sol
|   ├── LPTokenWrapper.sol
|   ├── Reservoir.sol
|   ├── RewardPool.sol
|   └── RewardToken.sol
└── vaults
    ├── ERCVault.sol
    └── Vault.sol
```

The Fairyproof team calculated an sha256 value for each of the audited files and these sha256 values are as follows:

```
contracts/ContractWhiteList.sol:
0xd590abcefc4efd3c558d48cc9a59038402cd6118e95c7a59702475cf8efc9b20
contracts/Migrations.sol:
0xa41cc0e1d6fb5483850fbd1f461edb50695ec0f136a0887938d98090eba2dcc6
contracts/Operatable.sol:
0x2445acf83bfcbf7fc371718d6f4fc188343e6acc7e6cbeb5164b65e0effdfcb1
contracts/controllers/Controller.sol:
0x9d61678837a58b5757153b716415a6ba82087f6e922eda027d07e5882bf2063a
contracts/fee/FeeBurn.sol:
0x501c307361fe3bc69309ee78c444bfda23f1b5609d6f1e396f3164eb4bb43e83
contracts/interfaces/IMasterChef.sol:
0x1d29fcba73b3c4c20cbce7635b6815b374daa328e4c37c3c0e36c83be1634fef
contracts/interfaces/compound/CTokenInterfaces.sol:
0x2e26fbe0992f41707939bc4f0b245346923d15e1c41cf3bd65e9b487fbacb691
contracts/interfaces/compound/ComptrollerInterface.sol:
0x8ceef04a5fe79417a803ba226cf5253fb83e296eaf1a2bc43d35e5e895853450
contracts/interfaces/compound/ICEther.sol:
0x5b4f801da8f8466cba8d71dac2190dd84375fe2b3394e3412adc577c9be92379
contracts/interfaces/compound/InterestRateModel.sol:
0xa64897c70a270a50ec1b5a7a62a2f7fc7f6193d662f3b490de7e8c27ce480d68
contracts/interfaces/depoist/IRewards.sol:
0x30acefe274793d71d689863c57ce172509ac90d6eefaca8e03f973ccfa06c481
contracts/interfaces/mdex/IMDexRouter.sol:
0x0a20a045f1a841980dbf1054035e771ee1eef76ab73d0362b775a4bac5707b34
contracts/interfaces/mdex/IMdexPair.sol:
0x086c3ffb10a417f4a38f2a7ba6930e1d344876c8fc71a0023c79dc9b27ff71c0
contracts/interfaces/mdex/ISwapMining.sol:
0xb4a111cef44212d4246f3c3f35827f7f253ec4057291c04b4c0a31039011c2cb
contracts/interfaces/weth/WETH9.sol:
0x0ffd6878a9f74f2fef8692cf6ac1453943ff3f544d5ebdb46254576ec229bc02
contracts/interfaces/yearn/IController.sol:
0xfce2bf8d2b6868d91bee26b8b2e72a102a31e4649ae3a8346194aa2173340971
contracts/interfaces/yearn/IStrategy.sol:
0xb294cb11ebbdb9a2d7c9e617a36180b0d815d97df7bdc188c70f780bd67dfd90
```

```
contracts/interfaces/yearn/IVault.sol:
0x649e569687d7fc8ceda0b651e14252e36c0e3fca92ebccd7f306a7d468df8dd3
contracts/lib/Const.sol:
0x8335df1bdf3932668dca38a1a05dab5b1b046dd20225cf1c96990bf4f3ee9f6e
contracts/lib/TransferHelper.sol:
0x46d36ec1722d8d27b470ef15dc25272619081de855708f27cc4d19665036a48e
contracts/strategies/AbstractLPStrategy.sol:
0x5bb224cd8eeb603b9ae83ce68701b70cd968419a27f597fdca2231d8b28eca59
contracts/strategies/AbstractStakeLPStrategy.sol:
0x7bdacd8f843f51852d55c42f623e91eab000116be0ce093fd7d5102c19ce9d4d
contracts/strategies/AbstractStakeSingleStrategy.sol:
0x2300d1d94ee600797b6723334879ce85cb4833abad3335efaa45ab22bcae1f52
contracts/strategies/CompleteCToken.sol:
0x7c86a71d9499da9e5396b7ff1b4a73dd61d9c4f7fb6d247d9fe760a2e5cc4e5a
contracts/strategies/CompoundInteractor.sol:
0x635b37dabfada439b9a5371438d8ed0bee3b9336c449333bcc3b3bc3e9724565
contracts/strategies/CompoundStrategy.sol:
0x79fa040244b6b7dd85a26933ba9b881d9978a192d225aeaccc2d481e505e03c8
contracts/strategies/MdexBoardroomLPStrategy.sol:
0x2c833adef9289d5c02ee13b194d7dbf26dd5d30a8a461aabcb657bc4c0d35125
contracts/strategies/MdexBoardroomSingleStrategy.sol:
0xf01835b8f33ca723ce665bb9df64ad4e4447795b75fcfa90a1c005603ac3f80d
contracts/strategies/ProfitNotifier.sol:
0x635cef7acf27d999ac078c9711b6fd0dbe2be1b5087ef5bf534891b73dfa459e
contracts/strategies/StrategyLP.sol:
0xfeb8d7813839cabaaf68d8b0a22157cc8cb28e99ef8c821cbfbd86e9df3890a6
contracts/strategies/StrategySingle.sol:
0xe5056bc3d573d3cb88862261fb16529c953c900ae5f0d5196c8f461c4211a946
contracts/token/DAOPool.sol:
0xae231e3acc351bc86a0fdba1a0ab5ec1a378cae78a85c588b8f3972cfc3bab0a
contracts/token/IRewardPool.sol:
0x26381ca268595a6e7ba8488b9d38a0b33363ebb14637fe278ae46ef053f69608
contracts/token/LPTokenWrapper.sol:
0x420974cc2117ad231d3faf7195a9e53ba14eddbe30ce58166c0978419c6b725c
contracts/token/Reservoir.sol:
0xcce867002a7bef889881a2be6ebbd9789cb04825989c7504803191b73b8432b5
contracts/token/RewardPool.sol:
0x1514734eb1f6314153e92660cfda0af8e1ce0e7d3a67fe961f1afa0d7537e636
contracts/token/RewardToken.sol:
0x40a0417d875b1faf46683da979f3e055e39065b7ee68f49df482662a2da754f9
contracts/vaults/ERCVault.sol:
0xb6e8798c3bc84988106c7475fa2d3231010e05465205ba9d3a341d30ecff4652
contracts/vaults/Vault.sol:
0x030102177e1a0157d612b39364b63238672ed2d1981e93d3250bd3bdbb9f4f2c
```

The goal of this audit is to review converter's solidity implementation for its token issurance, liquidity mining and aggregator service functions, 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 smart contract 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. 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 smart contracts.
   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 smart contracts 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 contract 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 converter system should work:

https://converter.finance

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, or low severities were found in the above contract files.

Two vulnerabilities with medium-severity were found in the above contract files.

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 smart contract systems.

03. Introduction to Converter

Converter is an aggregator service for DeFi investors, using automation to maximize profits from yield farming.

04. Major functions of audited code

The audited contract files implement the project's token issurance, liquidity mining and aggregator service functions.

The total supply of the tokens is fixed. The contract files don't have a function to mint additional tokens.

Users can stake specific crypto tokens in Converter's liquidity pools to get the project's token as a reward.

Attention: Converter uses automation strategies to invest users' staked crypto tokens in third party applications to maximize profits. When the third party applications are exploited users' staked tokens may suffer losses.

05. Key points in audit

During the audit, we worked closely with the Converter team and helped the team：

- fix some bugs in its application logic,
- add functions to withdraw staked assets for emergent cases,
- remove some functions that could be exploited, and
- refine some code writing

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 contract files the Fairyproof team focused on the possible issues and risks related to the following functions or areas.

- Contract Migration/Upgrade

Some of the the contract files can be migrated or upgraded and this may cause potential risks.

We found a vulnerability with medium-severity. For more details please refer to section 11 "Issue descriptions and recommendations by contract file".

- Access Control

Centralized access control prevails in the audited contract files and this centralized access control is accessed by an external account address. This may cause potential risks.

We found a vulnerability with medium-severity. For more details please refer to section 11 "Issue descriptions and recommendations by contract file".

- Miscellaneous

The Fairyproof team 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

- Vault.sol

Contract Migration/Upgrade Risk

- General Contract Files

Centralized Access Control

D. Low

- N/A

10. List of issues by contract file

- Vault.sol

Contract Migration/Upgrade Risk: Medium

- General Contract Files

Centralized Access Control: Medium

11. Issue descriptions and recommendations by contract file

- Vault.sol

Contract Migration/Upgrade Risk: Medium

Source and Description:

This contracts that are used as automation strategies can be migrated or upgraded. When mistakes are made in contract migration or upgrade, or the upgraded new contract has vulnerabilities the locked assets may suffer losses.

Recommendation:

Consider migrating or upgrading contract files with extreme caution or completely disabling this feature. When upgrading an existing contract to a new one, an audit must be done for the new contract before migration happens.

Update: acknowledged by the Converter team. The team ensures that migration or upgrade will happen only when new contract files are audited with extreme caution, and migration or upgrade will be performed as little as possible.

- General Contract Files

Centralized Access Control: Medium

Source and Description:

Centralized access control prevails in the audited contract files and this centralized access control is accessed by an external account address. This may cause potential risks.

Recommendation:

Consider transferring the centralized access control to a DAO or a multi-sig wallet.

Update: acknowledged by the Converter team. The team plans to tranfer the access control in a future upgrade.

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.

- Fixing Compiler Warnings

Consider fixing all the compiler warnings.

Update: acknowledged by the Converter team. The team plans to make changes in a future upgrade.