Beefy Audit Report

Version 1.0.0

Serial No. 2021101800012018

Presented by Fairyproof

October 18, 2021

FAIRYPROOF

01. Introduction

This document includes the results of the audit performed by the Fairyproof team on the Beefy Finance project.

Audit Start Time:

October 14, 2021

Audit End Time:

October 15, 2021

Audited Code's Github Repository:

https://github.com/beefyfinance/beefy-protocol

Audited Code's Github Commit Number When Audit Started:

a212107992880d7cff05c042435541e68df2e15c

Audited Source Files or Asset Pools' Onchain Addresses:

BIFI.sol: 0xCa3F508B8e4Dd382eE878A314789373D80A5190A (on BSC)

WBNB's staking pool: 0x565EB5e5B21F97AE9200D121e77d2760FFf106cb (on BSC)

ETH's staking pool: 0x50013DA75AC4224B5c49fe197bfaf68C16E19d6E (on BSC)

LINK's staking pool: 0x44bBCa30AEa5f2D226668c197053Db3FA9eC9771 (on BSC)

CREAM's staking pool: 0x38191ea2A077D86B3dF410379F8c0Bb130683116 (on BSC)

BSCswapLP (WBNB/BUSD)'s staking pool: 0x1263F0BFfE2D740Ea3279416D0e84943B66958eb (on BSC)

BSCswapLP (WBNB-SPARTA)'s staking pool: 0x2b8C4aD8053b7933CFde936F16aBc55BB5F694c6 (on BSC)

Audited Source Files:

The calculated SHA-256 values for the audited files when the audit was done are as follows:

```
IRewardDistributionRecipient.sol:
0xba3fe503687f33dddf44907756deb948632cf38c54ac95d0812ffb320c45a20c

LPTokenWrapper.sol:
0xbffc3610ec19bd8ab1148a77befd7e104e212e9bab6faca73b62c09379abd659

RewardPool.sol:
0x62797eab1ba0e3a932e736503cdd44b0b38506db836e0b1988b267693f99e362

TokenTimelock.sol:
```

```
0xa536e7de5c58feed2d5237bb9d228124a15473ee087476cfcf82a0f862148daa

YFI.sol:
0x166ff896c342aaf7c5d2a551d0b65422707e8b8d104908d28ce7cbf6809e2dd9

1_wBNB.sol:
0x37c7e845c34c37690f8182fc5613aef24626299d2d4aa1b92aee4f450cc771f1

2_ETH.sol:
0x9ece3c373cd03e580cf7a8159e95b499c72da4c3da931893a8e45610aa5707b2

3_LINK.sol:
0x7ea754f9075e2ecad27a573be07f8783b6f65763e1a5f80cf184a02bf80694e7

4_CREAM.sol:
0x9febc200ba331daaaadafe53cf2c8c4b345fae42c3660245cb7640fd8ffa42e4

5_BNB_BUSD.sol:
0x1e141a67c9d4e221bd47e9a661833483f7dd06df31dbabbb8ca19dbc4816e3f4

6_BNB_SPARTA.sol:
0x6cade3fe9d8f20eb5b255499d8a6d25d8bedf6775d045d70c97392eba656b46d

0x0dc06716bc1efdb5affd1308302fd45dbc31d09a.sol:
0xb6799f7b931a45f77630aef04f9dcfc21a4bc2acf2add90ef92b87235b4d1923

0x48091c485eb842c86805251a5d68993bf49c90a1.sol:
0xb6799f7b931a45f77630aef04f9dcfc21a4bc2acf2add90ef92b87235b4d1923

0x4dece2b4c14083250f5a68faa18372563d174fea.sol:
0xb6799f7b931a45f77630aef04f9dcfc21a4bc2acf2add90ef92b87235b4d1923

0xeDBf7a30453cf9A63Cca4353b38Ae60AD4E27Bac.sol:
0xb6799f7b931a45f77630aef04f9dcfc21a4bc2acf2add90ef92b87235b4d1923

BIFI.sol:
0x49efefdacd8880389516e4ad84f7ab6d8e65dfb8692d297572829200a621cd1e
```

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

```
contracts/
├── IRewardDistributionRecipient.sol
├── LPTokenWrapper.sol
├── RewardPool.sol
├── TokenTimelock.sol
└── YFI.sol

pools/
├── 1_wBNB.sol
```

FAIRYPROOF

```
├── 2_ETH.sol
├── 3_LINK.sol
├── 4_CREAM.sol
├── 5_BNB_BUSD.sol
└── 6_BNB_SPARTA.sol

timelocks/
├── 0x0dc06716bc1efdb5affd1308302fd45dbc31d09a.sol
├── 0x48091c485eb842c86805251a5d68993bf49c90a1.sol
├── 0x4dece2b4c14083250f5a68faa18372563d174fea.sol
└── 0xeDBf7a30453cf9A63Cca4353b38Ae60AD4E27Bac.sol

token/
└── BIFI.sol
```

The goal of this audit is to review Beefy Finance's solidity implementation for its token issurance and staking 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.

This audit only applies to the specified code, software or any materials supplied by the Beefy Finance team for specified versions. Whenever the code, software, materials, settings, enviroment etc is changed, the comments of this audit will no longer apply.

— 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 token issurance and staking functions should work:

https://www.beefy.finance/

This was considered the specification.

— Comments from Auditor

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

Additional notice: 0.

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 Beefy Finance

Beefy Finance is a multi-chain yield optimizer.

04. Major functions of audited code

The audited code mainly implements the following functions:

- Token Issuance

- Name: beefy.finance
- Symbol: BIFI
- Precisions: 18
- Max Supply: 80,000
- Flexible Max Supply: No
- Onchain Address: 0xCa3F508B8e4Dd382eE878A314789373D80A5190A (on BSC)

- Staking

Users can stake crypto assets to earn rewards in the BIFI token.

- Asset Locking and Withdrawal

After a crypto asset is locked in a specific address it can only be withdrawn after a defined locking period.

05. Admin rights

In this application the admin's previlleges have been revoked:

- The existing implementation grants previlleges in the token issuance function to the admin but the previlleges have been transferred to a zero address. The transaction that trasferred the right can be viewed at https://bscscan.com/tx/0x5e883d6da804fc8fe29bd74406774f3f1bc7e86d8750d140c52da3fa3f83f3f0. The minter's previlleges have been revoked as well. The transaction that revoked the previlleges can be viewed at https://bscscan.com/tx/0x672a2c33769750ab69cafc47d9d5e1f8d0decd6da2f32fc6f412fa39aa0c195b

- The admin had previlleges to set reward amouts. After the reward amounts were set the previlleges had been revoked. For instance the previllege to set reward amounts for the WBNB pool has been transferred to a zero address. The transaction can be viewed at: https://bscscan.com/tx/0x72e509ed8df7975db720bb4814fa78bafb62003ac41e86e2ba694d9074514539 . The address of `rewardDistribution` has been set to a zero address as well. The transacton of this setting can be viewed at: https://bscscan.com/tx/0x18f6477ef5078d0719cf7ce3dbbd500fdbda186c8b5f81dcfcf77555f866d474

FAIRYPROOF

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. List of issues by severity

A. Critical

- N/A

B. High

- N/A

C. Medium

- N/A

D. Low

- N/A

09. List of issues by source file

- N/A

10. Issue descriptions

- N/A

11. 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