HOGT Audit Report

Version 1.0.1

Serial No. 2021051600022021

Presented by Fairyproof

May 16, 2021

FAIRYPROOF

01. Introduction

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

Project Token's Name:

HOGT and YHOGT

Project Token's HECO Onchain Address:

HOGT: https://hecoinfo.com/address/0xFC33785f44c1069010b5De466eE7657C27aB8A0E

Audited Code's Github Repository:

N/A

Audited Code's Github Commit Number:

N/A

Audited Contract Files' HECO Onchain Addresses:

HOGToken.sol: https://hecoinfo.com/address/0xFC33785f44c1069010b5De466eE7657C27aB8A0E

Gontroller.sol : https://hecoinfo.com/address/0xA05B1dE168c9618950771c4fA9f14AdFE07f645D

Audited Contract Files:

The calculated SHA-256 values for 38 files audited are as follows:

```
DaoTokenStaked.sol      :
0x5040f9b9e06c547a1af4d8cf66b3e25c4fa4a0d60eb193824c287c51a1e0c7f6
Epoch.sol               :
0x4318bc162cf52415d177fe23604483acacf06dcc736ea12de8944f04f2ee1f6a
HOGToken.sol            :
0x3d9a9d17daae661ceb1af1a12b3da38c00ce9d8cf02af1b735d6a761ef10a2f5
HogtDaoLimitStaked.sol  :
0x872fc25030072d6a6021bad42149e11a88df8129f61e8d5e5c959ae2b3a5f51e
HogtLpStaked.sol        :
0xe787cc6633a3230fd5396c3910f09fe1c66eae9c893a0602fabb46d6c84ec1dd
HogtUpperLimitStaked.sol:
0x43c948bec7ef593946558a3a70b27587cc144f7ef7714e5395ba864b4085c9cd
Multicall.sol           :
0x9daacf91f4b43e9d9679428e5b4bff00b49a8817cf0c2f43086af84ef6121fa0
Release.sol             :
0x48675bf2261e9f6966922a401a562a14a76995998051b4e046486c7e586a6783
Reservoir.sol           :
0x00600af89502460918b47bca525540142e7e5fe871db3dba1fe5499eb250f615
```

```
YHogtStaked.sol         :
0x65a6d692a29c4c26467b09c367226b756a0696a52be29864add31259571c3543
Dao15HOGT.sol           :
0x4f0c92cb1525d2b129f0964f74a77e45853ac950dc31975eaf0f671047f9d61e
Dao30HOGT.sol           :
0xb2f95b4e9e9391d80dac632cac9b0c89c8366e5e79f1c2238fde60612d864e62
Dao60HOGT.sol           :
0x07000d1dc9d3de26326eabd2b8ec01131e6d1003a78e3486c021d88c38fa1993
Dao7HOGT.sol            :
0xee26e3b22aa28086c336a66ba8a83574303b0962cd52460cdf5c07ecd3892c78
gTokenStaked.sol        :
0x5b38d1377edaa31bfefd229dfe13a142901c2be8ed4cb2e890c51835aa4aef12
Gontroller.sol          :
0x9b475706f61a282f8b2d870eac6cde17d2e4aa68d64f4ab8934355a1004a025f
StrategyFilda.sol       :
0xf7ca00913e6c1a870822fdc16df1e65a3ff0db53edded62a9527e5fbd342ef8d
gHT.sol                 :
0x31966811582ac804a95f13baf01ad633bfc982b9fdc899c0b4300fc1172d251b
gVault.sol              :
0x3159fdc0f61677596066456364c57ee1bee871ed610121fff47b5d5e2a47d7f2
IController.sol         :
0xd2c39b0d2c1b5c7a4260e18c5baa144d1a7d8c69bf4815f6496b2c859b973228
IConverter.sol          :
0x79910e72e2bfdfb49529354e79e014548d1d6c34dde1eeafbe19b9b90d7b4bdd
IFarmPool.sol           :
0x17b31e090b48bef6b203a9b6d464ca103e497dd03b3789ce0bbba5f422ac09b8
IHogt.sol               :
0x575d73d838075f5ff10bf51e06bc2ec2668b755fad0d3c3b82f1cd46f56e0da1
ILendingStrategy.sol    :
0x65c755015e1fe1909d4284e2fac52b986b71f00c455a04ec4fb52bb3d83d340a
IMdexStrategy.sol       :
0x69969aa1995e010fcbc42340afd136f9aa4a5aca6aec2b1d41853623c06ff884
IRouter.sol             :
0x133dbd21c9bd0fc03e16e4f181e9adcccd39efbdda94e26801dbb97e5b550818
IWETH.sol               :
0x7ae646fda5bd03fe59352915ac09df03d564a5e11300001252d016fbbc257210
IYogt.sol               :
0xc8e5feae49f4a04ca597217ef1871ea122ccc8f51d89c704c315bd7ad8626d9e
lpStrat.sol             :
0x556896c83a9841e20a43f38d1a41b58a56c7f98487e7f0d93c1a1311dfa1d581
lpVault.sol             :
0x5995c6090426a2a7b899e231eb259dd24c18e80978909a35dd932603ca59e91e
DaoFrozenMap.sol        :
0x399ff80cbf00d70ef5b7a2b7056fd7b1e0d8302e385d0875023fef7f5a84597c
ERC20.sol               :
0xe12eff8f58ed512ec093ee8b3e1e064e86745e5fecc5a8c92e0f86da3847c680
ERC20Detailed.sol       :
0x5b64668259a96cca4c0a8d78bea655f223b6ae319377d12a3d0fdc21216f22c4
```

```
EnumerableMap.sol         :
0x88e3098e9761bdab0dac437a41a5f0bc88caf5e2ac4869a1dfac085c4362d661
Ownable.sol               :
0x7f5bfca189fb6d0b69ca577d85231a917451cc19660e33d78e459228dae32c82
Pausable.sol              :
0x4e0ab12c50ff8b6f7df710632e36c4df3b2097eb17f4e69b612b7f9450c3d803
ReentrancyGuard.sol       :
0x76e4bd14995be8032a60a31ed676f4f594f0216d20766f22eae3490f21b22f3a
yHOGToken.sol             :
0x746579f93408108b5cea9bcd93678a92adfb5c3c6c36fb22d06c28dc57a3cfec
```

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

```
/
├── DaoTokenStaked.sol
├── Epoch.sol
├── HOGToken.sol
├── HogtDaoLimitStaked.sol
├── HogtLpStaked.sol
├── HogtUpperLimitStaked.sol
├── Multicall.sol
├── Release.sol
├── Reservoir.sol
├── YHogtStaked.sol
├── dao
|   ├── Dao15HOGT.sol
|   ├── Dao30HOGT.sol
|   ├── Dao60HOGT.sol
|   └── Dao7HOGT.sol
├── gTokenStaked.sol
├── gVault
|   ├── Gontroller.sol
|   ├── StrategyFilda.sol
|   ├── gHT.sol
|   └── gVault.sol
├── interfaces
|   ├── IController.sol
|   ├── IConverter.sol
|   ├── IFarmPool.sol
|   ├── IHogt.sol
|   ├── ILendingStrategy.sol
|   ├── IMdexStrategy.sol
|   ├── IRouter.sol
|   ├── IWETH.sol
|   └── IYogt.sol
├── lpVault
|   ├── lpStrat.sol
|   └── lpVault.sol
├── utils
```

```
|    ├── DaoFrozenMap.sol
|    ├── ERC20.sol
|    ├── ERC20Detailed.sol
|    ├── EnumerableMap.sol
|    ├── Ownable.sol
|    ├── Pausable.sol
|    └── ReentrancyGuard.sol
└── yHOGToken.sol
```

The goal of this audit is to review HOGT's solidity implementation for its token issurance, yield aggregator and liquidity mining 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 HOGT system should work:

N/A

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

— Comments from Auditor

No vulnerabilities with critical, high, medium or low-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 HOGT

HOGT is a HECO based one-stand DeFi platform that provides users with crypto lending, trading and investment services.

04. Major functions of audited code

The audited code implements the following functions:

- Issurance of the HOGT governance token:
  - it is issued for rewarding token stakers
  - a mint function can be used to mint tokens, and the max supply is 500 million
- Issurance of the YHOGT reward token:
  - it is issued for rewarding referers
  - a mint function can be used to mint tokens, and the max supply is 400 million
- Obtaining shares in DAO by staking
  - users can stake a pre-defined token to obtain shares in DAO
  - if a user's share reaches a thredhold he/she can participate in staking and get staking rewards
- Aggregator service
  - a user can deposit a pre-defined token in the aggregator vault, receive a gToken as a certificate and get returns from the vault's investment

    - a user can stake gToken and get rewards in the HOGT token
    - the HOGT platform will use its aggregator service to invest users' deposited tokens in third-party applications to get profits
    - 70% of the total profits earned by the platform will be automatically re-invested and the remaining 30% will be used to buy back and burn the HOGT token
- Staking
    - types of staking
        - staking of an HOGT LP token
        - permissionless staking of the HOGT token
        - permissioned staking of the HOGT token for a user who has a share in DAO, which reaches a threshold
        - staking of the YHOGT token to receive rewards in the HOGT token
        - staking of the gToken to receive rewards in the HOGT token
    - admin can modify reward settings in staking
- Recruit mechanism
    - Both people who recruit others and people who are recruited can participate in staking to get rewards in the YHOGT token. There is only one layer in this recruit mechanism
    - A staking operation and a withdrawal operation will generate the same amount of reward in the YHOGT token, among which 70% is distributed to the recruiters and 30% is distributed to those recruited

05. Key points in audit

During the audit, the audit team worked closely with the HOGT team, helped:

- refine the implementation to burn tokens,
- remove some admin's inappropriate access control,
- remove some redundant code,
- refine the implementation to set minimum values

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.

- Token Issurance

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

We didn't find issues or risks in this function or area at the time of writing.

- Asset Security

We checked whether or not the assets in staking, DAO and investment strategies were secure.

We didn't find issues or risks in this function or area 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 contract 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.

- Removing Redundant Parameters

In line 34 in the `lpStrat.sol` contract file, the `deposit` function has an unused input parameter `_user`.

Consider removing this parameter.

Update: this parameter is defined for handling compatibility issues.