HashBrigdeOracle Audit Report

Version 1.0.0

Serial No. 2021021000022012

Presented by Fairyproof

Feburary 10, 2021

FAIRYPROOF

01. Introduction

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

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

ad3741645b4ccbb643f62cc9f34377e4f6b3aacd

The goal of this audit is to review hashbridgeoracle's solidity implementation for an oracle, 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 document reflect the current understanding of known security patterns and state of the art regarding smart contract security.

And the solidity implementation was audited based on Ethereum's running environment. Whether or not this implementation can run on other blockchains or would encounter issues running on other blockchains is not covered by this audit.

Risks or issues introduced by this implementation interacting with contracts from other projects are not covered by this audit.

Risks or issues introduced by using data feeds from offchain sources are not covered by this audit.

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.

— Methodology

Hashbridgeoracle's codebase 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.

— Structure of the document

This report contains a list of issues and comments on all the contract files under the directory https://github.com/hb-oracle/hboracle/ and its sub-directories. 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 hashbridgeoracle system should work:

https://github.com/hb-oracle/hboracle/

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

— Comments from Auditor

No vulnerabilities with critical, high or medium severities were found in the hashbridgeoracle's codebase. All the vulnerabilities with low severity were acknowledged by the team, and the team doesn't think they will trigger issues or risks and may make changes in future upgrades.

The hashbridgeoracle's codebase passed the audit performed by the Fairyproof team.

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. 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.

04. List of issues by severity

A. Critical

- N/A

B. High

- N/A

C. Medium

- N/A

D. Low

- FluxAggregator.sol

Shadowed Declaration

- AggregatorProxy.sol

Shadowed Declaration

Inappropriate Function Name

- interfaces/HashBridgeOracleTokenInterface.sol

Inappropriate Parameter Names

- HBOToken Directory

Obsolete Usage

05. List of issues by contract file

- FluxAggregator.sol

Shadowed Declaration: Low

- AggregatorProxy.sol

Shadowed Declaration: Low

Inappropriate Function Name: Low

- interfaces/HashBridgeOracleTokenInterface.sol

Inappropriate Parameter Names: Low

- HBOToken Directory

Obsolete Usage: Low

06. Issue descriptions and recommendations by contract file

- FluxAggregator.sol

Shadowed Declaration: Low

Source:

Line 671： in the statement `RoundDetails storage details = details[_queriedRoundId];` the locally declared variable `details` shadows the global variable `details` declared in the statement `mapping(uint32 => RoundDetails) internal details` in line 102. This causes reader confusions.

Recommendation:

Consider renaming the variable `details` declared in line 671 to `detail` and making changes in lines 677 and 680 accordingly.

Update: Acknowledged by the hashbridgeoracle team. The team doesn't think this will cause potential issues and therefore prefers to keep it for now, and may make a change in a future upgrade.

- AggregatorProxy.sol

Shadowed Declaration: Low

Source:

Lines 382 and 394: the variable `phaseId` declared in the statement `uint16 phaseId = uint16(_roundId >> PHASE_OFFSET);` in line 382 and the variable `phaseId` defined as a function parameter in line 394 shadows the function `phaseId` defined in line 313.

Recommendation:

Consider renaming `phaseId` that appears in lines 382, 385, 394, 407 and 411 to `_phaseId`.

Update: Acknowledged by the hashbridgeoracle team. The team doesn't think this will cause potential issues and therefore prefers to keep it for now, and may make a change in a future upgrade.

Inappropriate Function Name: Low

Source:

Line 373: the function `addPhase` implies by its name, that a "phase" will be added and some states will be updated. However it is defined as a pure function and its behavior is to compose a `roundId`. This function is not named in a way that describes its behavior. This causes reader confusions.

Recommendation:

Consider renaming the function `addPhase` to `encodeRoundId`, defining it as `internal` and making changes in lines 134, 407 and 411 accordingly.

Update: Acknowledged by the hashbridgeoracle team. The team doesn't think this will cause potential issues and therefore prefers to keep it for now, and may make a change in a future upgrade.

- interfaces/HashBridgeOracleTokenInterface.sol

Inappropriate Parameter Names: Low

Source:

Line 18: the parameter `addedValue` defined in the function `decreaseApproval` in line 18 and the parameter `subtractedValue` defined in the function `increaseApproval` in line 22 don't match their functions' behaviors respectively. This causes reader confusions.

Recommendation:

Consider swapping the two parameters.

Update: Acknowledged by the hashbridgeoracle team. The team doesn't think this will cause potential issues and therefore prefers to keep it for now, and may make a change in a future upgrade.

- HBOToken Directory

Obsolete Usage: Low

Source: the contract files contained in this directory use older versions of Solidity compiler(as old as 0.4.24) and therefore have lots of obsolete Solidity usages.

Recommendation:

Consider rewriting the code by using a Solidity compiler with version 0.6.0 or above and replacing the obsolete usages with new usages that match the selected compiler version.

Update: Acknowledged by the hashbridgeoracle team. The team doesn't think this will cause potential issues and therefore prefers to keep it for now, and may make a change in a future upgrade.