Smart Contract Audit Report

Float Capital

Yield Manager Upgrade

18th of January 2022

Contents

Disclaimer

As of the date of publication, the information provided in this report reflects the presently held understanding of the auditor's knowledge of security patterns as they relate to the client's contract(s), assuming that blockchain technologies, in particular, will continue to undergo frequent and ongoing development and therefore introduce unknown technical risks and flaws. The scope of the audit presented here is limited to the issues identified in the preliminary section and discussed in more detail in subsequent sections. The audit report does not address or provide opinions on any security aspects of the Solidity compiler, the tools used in the development of the contracts or the blockchain technologies themselves, or any issues not specifically addressed in this audit report.

The audit report makes no statements or warranties about the utility of the code, safety of the code, suitability of the business model, investment advice, endorsement of the platform or its products, the legal framework for the business model, or any other statements about the suitability of the contracts for a particular purpose, or their bug-free status.

To the full extent permissible by applicable law, the auditors disclaim all warranties, express or implied. The information in this report is provided "as is" without warranty, representation, or guarantee of any kind, including the accuracy of the information provided. The auditors hereby disclaim, and each client or user of this audit report hereby waives, releases and holds all auditors harmless from, any and all liability, damage, expense, or harm (actual, threatened, or claimed) from such use.

1. Preface

The team of **Float Capital** contracted byterocket to conduct a smart contract audit of a contract upgrade added to their smart contracts, enabling them to integrate more than just one yield provider such as Aave or Compound. Float Capital is a *"peer-to-peer, yield-enhanced, floating synthetic asset exposure mechanism"*. They describe themselves as *"the easiest and safest way for users to buy synthetic assets. Users do not need to worry about over-collateralization, or suddenly getting liquidated"*.

Their smart contracts are being updated to include the feature of a more advanced yield manager, which aims to generalize the smart contract as well as add an additional one for Compound. Additionally, the Aave variant now contains an optimization, where interactions with the Aave contracts are only done when a certain capital threshold has been reached.

The functionality is contained in the **YieldManagerAave.sol**, **YieldManagerAave Basic.sol** and **YieldManagerCompound.sol** files. During this audit, we only focused on changes that have been made because of this new feature.

The team of byterocket reviewed and audited the above smart contracts in the course of this audit. We started on the 16th and finished on the 18thth of January 2022.

The audit included the following services:

- *Manual Multi-Pass Code Review*
- *In-Depth Protocol Analysis*
- *Automated Code Review*
- *Formal Report*

byterocket gained access to the code via their public GitHub repository. We based the audit on the monorepo-dev branch's state from January 12th, 2022 (*commit hash e597030da22e02138a39854c5d122d37b02c4594*).

2. Manual Code Review

We conducted a manual multi-pass code review of the smart contracts mentioned in section (1). Three different people went through the smart contract independently and compared their results in multiple concluding discussions.

These contracts are written according to the latest standards used within the Ethereum community and the Solidity community's best practices. The naming of variables is very logical and understandable, which results in the contract being useful to understand. The code is very well documented and up to the latest standards.

Due to the very structured and open process that is being used for contract development at Float Capital, it was very easy for us to gain insights in certain motivations and ideas behind the changes and implementations. Additionally, the team is working at the forefront of novel testing methods and techniques for smart contracts, which also made this a very pleasant endeavor for us.

On the code level, we **found no bugs or flaws**. A further check with multiple automated reviewing tools (*MythX*, *Slither*, *Manticore*, *and different fuzzing tools*) **did not find any additional bugs**.

3. Protocol/Logic Review

Part of our audits are also analyses of the protocol and its logic. A team of three auditors went through the implementation and documentation of the implemented feature change.

We went through all of the provided documentation, tests, and contracts in a very detailed manner. The general description of the feature and the protocol itself is very well made, it's very easy to understand how each function is supposed to work and what it implements.

We were **not able to discover any problems** in the protocol implemented in the smart contract.

3.1 Feature implementation

The feature implementation contains changes to generalize the previous implementation of the Aave variant of the Yield Manager contract. Additionally, there have been optimizations to only interact with the Aave contracts, once a certain threshold has been reached like for example two new variables show in **YieldManagerCompound.sol**.

```
/// @dev The maximum amount of payment token this contract will allow
function maxPaymentTokenNotInYieldManagerThreshold() internal pure
         virtual returns (uint256)
{
  return 0;
}

/// @dev The desired minimum amount of payment token this contract will
    target
function minPaymentTokenNotInYieldManagerTarget() internal pure virtual
         returns (uint256)
{
  return 0;
}
```

The Compound variant implements exactly the same logic, only the Aave calls have been replaced with the corresponding calls to the Compound cToken contract.

Throughout the implementation of this feature, we have **not found any logical errors**. Additionally, we have not found any changes, that may have introduced any flaws or bugs.

4. Summary

During our code review (*which was done manually and automated*), we **found no bugs or flaws**. Our automated systems and review tools also **did not find any additional ones**.

The protocol review and analysis did neither uncover any game-theoretical nature problems nor any other functions prone to abuse. We have **not found any logical errors** in the implementation. Additionally, we have not found any changes, that may have introduced any flaws or bugs due to this upgrade.

In general, we are **delighted** with the overall quality of the code and its documentation. Additionally, there are extensive tests and even a custom testing framework, covering all of the functionality of the system.