Giveth GIVeconomy Contracts Code Review

Oleksii Matiiasevych
May, 2022

REVIEWER IS PROVIDING THIS CODE REVIEW AS PART OF A QUALITY ASSURANCE PROCESS ON A “BEST EFFORTS”, “AS IS” AND “AS AVAILABLE” BASIS AT THIS PARTICULAR POINT AND TIME, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

WITHOUT LIMITING THE FOREGOING, REVIEWER EXPRESSLY DISCLAIM ANY AND
ALL WARRANTIES AND CONDITIONS OF MERCHANTABILITY, TITLE, ACCURACY AND COMPLETENESS, OR FUNCTIONING OF THIS CODE.

BY ACCESSING THIS DOCUMENT YOU ACKNOWLEDGE, ACCEPT AND AGREE TO THE FOREGOING. THESE MATERIALS ARE NOT AN OFFER OF SECURITIES FOR SALE. SECURITIES MAY NOT BE OFFERED OR SOLD IN THE UNITED STATES IN ABSENCE OF REGISTRATION WITH THE U.S. SECURITIES AND EXCHANGE COMMISSION OR AN EXEMPTION FROM REGISTRATION UNDER

THE U.S. SECURITIES ACT OF 1933, AS AMENDED.

1. **Introduction.** Giveth requested to perform a code review of the contracts implementing the GIVeconomy ecosystem. The contracts in question can be identified by the following git commit hash:

   ```
   31c9971a8369d01752ff52a72b5e454a8f12447b
   ```

   There are 10 contracts in scope. They are located under the **Distributors**, **TokenDistro** and **Tokens** folders.

2. **Disclaimer.** The code review makes no statements or warranties about utility of the code, safety of the code, suitability of the business model, regulatory regime for the business model, or any other statements about fitness of the contracts for any specific purpose, or their bugfree status.

3. **Executive Summary.** There are 0 critical, 0 major, 4 minor, 5 informational issues identified in the initial version of the contracts. There are **no** known compiler bugs for the specified compiler version (0.8.6), that might affect the contracts' logic.

4. **Critical Bugs and Vulnerabilities.** No critical bugs or vulnerabilities were found.

5. **Initial Line By Line Review.**

5.1. **GardenUnipoolTokenDistributor**, line 203. Minor, the **notifyRewardAmount()** function doesn't verify that there are enough unallocated tokens in the **TokenDistro**.

5.2. **UnipoolTokenDistributor**, line 81. Note, the _PERMIT_SIGNATURE constant could use **IUni(0x0).permit.selector** instead of raw bytes.

5.3. **UnipoolTokenDistributor**, line 84. Note, the _PERMIT_SIGNATURE_BRIDGE constant could use **IBridge(0x0).permit.selector** instead of raw bytes.

5.4. **UnipoolTokenDistributor**, line 171. Minor, the **notifyRewardAmount()** function doesn't verify that there are enough unallocated tokens in the **TokenDistro**.

5.5. **TokenDistro**, line 34. Optimization, all the state variables in the contract except for **balances/startTime/cliffTime** could be made immutable.

5.6. **TokenDistro**, line 117. Note, in the **setStartTime()** error sig should be **TokenDistro::setStartTime** instead of **TokenDistro::assign**.

5.7. **TokenDistro**, line 121. Note, in the **setStartTime()** error sig should be **TokenDistro::setStartTime** instead of **TokenDistro::assign**.

5.8. **BridgeToken**, line 585. Minor, the **safeTransfer()** function is broken. It will pass if the **IERC20.transfer()** function returns false. In the current contract the result of **safeTransfer()** is irrelevant, that is why it is not exploitable. The function should not be used in any future contracts though, and has to be fixed.

5.9. **TokenERC677**, line 585: Minor, the **safeTransfer()** function is broken. It will pass if the **IERC20.transfer()** function returns false. In the current contract the result of **safeTransfer()** is irrelevant, that is why it is not exploitable. The function should not be used in any future contracts though, and has to be fixed.

Oleksii Matiiasevych