SOMISH

Blockchain Labs

Smart Contract Audit Report *GoodDollar*

July 4th, 2020

Amended on July 14th, 2020

Index

Summary

Audit Report prepared by Somish Blockchain Labs for GoodDollar Smart contracts.

Process and Delivery

Two (2) independent experts performed an unbiased and isolated audit of the code below. The debrief took place on July 4th, 2020, and the final results are presented here.

Audited Files

The following contract(s) were covered during the audit:

SimpleDAIStaking.sol
GoodReserveCDai.sol
GoodMarketMaker.sol
GoodFundManager.sol
FirstClaimPool.sol
UBIScheme.sol
ContributionCalculation.sol
BancorFormula.sol

Client Documents

The following files were shared along with the contracts by the client for the purpose of the audit.

- MVP GoodContracts Specs phase 1

Notes

Audit was performed on the commit 20b71ea06a6eb8a39976aba160177516ced906b4 of GoodContracts repo shared by the GoodDollar team.

Intended Behavior

An OpenUBI protocol that uses DEFI and token economics to fund a sustainable universal basic income based digital currency. Staker send DAI to the staking contract. Staking contract invests new funds in DEFI. Daily "Fund Manager" contract pulls capital gains from all staking contracts and deposit them to the reserve. Reserve is updated with new funds from the fund manager. Reserve Ratio is declined which allows mint more G$. The Reserve mints new G$ based on the amount received from the fund manager and the declined reserve ratio. New G$ from reserve in an equal value to the interest is sent back to “Fund Manager”. If a staker is interested he can withdraw his G$. Rest of the newly minted G$ tokens are sent to a UBI scheme contract. Active Claimers “claim” daily UBI from the UBI scheme contract. Claimants who have not claimed in the last 14 days can be marked as inactive users by any one, this process is called fishing. The fisherman who fishes inactive users gets the daily UBI of the fished user.

Issues Found

Critical

No critical issues found.

Major

1. Doesn’t handle receipt and transfer of COMP tokens

Compound protocol is now issuing COMP tokens for all participants in the network. Since the current implementation of GoodDollar staking invests DAI received into Compound, it is a valid recipient of COMP token. The current contracts do not handle receipt and transfer of COMP tokens.
Amended (July 14th 2020): A new function was added in GoodReserveCDai.sol and SimpleDAIStaking.sol contracts to recover any ERC20 tokens stuck. This allows Avtar to withdraw COMP tokens issued to Staking and Reserve contracts. The issue is no longer present in commit 14cfce49569637ff80d2391267e2afce76ac6647.

Minor

1. Staking Contract is not Upgradeable

As the system is dependent on an external protocol, which themselves are upgradable via governance, it is always good to have upgradable contracts to handle the upgrades. This will help the system be ready for scenarios like issuance of COMP token or any other features introduced/removed from the external protocol. This will also enable resolution of unforeseen bugs or any feature upgrades in G$ in future.

Usage of proxy technique for upgrade of contracts can be considered. The right to upgrade can be vested with governance.

2. Miscalculation of totalStaked while withdrawing stake

In SimpleDAIStaking.sol, in function withdrawStake(), the variable daiWithdraw is being subtracted from totalStaked. However, the amount being transferred to the user is minimum of (daiActual,daiWithdraw). On the looks of it, it may seem the same but technically if (daiActual < daiWithdraw) then this could result in issues.
Amended (July 14th 2020): As per the response of the GoodDollar team, redeeming in compound may result in a tiny fraction of precision error and can be neglected for the calculation of totalStaked.

3. Use SafeMath library for all calculations

SafeMath library was used for most of the calculations, we would recommend to use it for all the calculations. Following are the calculations in which SafeMath is not used:
GoodMarketMaker.sol: 132
SimpleDaiStaking.sol: 147, 167 169

Notes

1. Upgrade code to Solidity version > 0.6.0

Currently, the smart contract is written in Solidity version 0.5.4, a now known vulnerable version of Solidity. The contracts should be updated and their compiler version fixed
Recommendation
Refer to the Ethereum changelog (https://github.com/ethereum/solidity/blob/develop/Changelog.md) for the most up to date Solidity version. Contract elements may need to be updated in order to work with the latest Solidity version.

2. Duplicate checks for allowance

In stakeDAI() of SimpleDAIStaking.sol, DAI allowance is checked before transferring user funds, This check can be avoided as the allowance check is already present in transferFrom().

3. Global variable used for storing current day

In UBIScheme.sol, a global variable is used to store current day and function to update its value. Transaction gas can be reduced by removing the variable and using the formula(currentDay = (now.sub(periodStart)) / 1 days) directly instead.

4. Abstract the token from the contract

GoodReserveCDai contract tries to follow a style such that functions like buy(), sell(), mintInterestAndUBI() can accept tokens apart from cDAI. However, the naming and implementation of the contract is hardcoded for cDAI. It is a good idea to abstract the token from the contract.

5. **Unused function parameters**

   In calculateContribution() of ContributionCalculation.sol, arguments _marketMaker, _reserve. _contributer, _token were defined but not used anywhere in the function. Remove if the parameters are not intended to use.

6. **Code coverage**

   Test cases were written and 100% function, line coverage was achieved, we would recommend achieving 100% branch coverage in order to avoid any unforeseen issues..

Closing Summary

Upon audit of the GoodDollar smart contracts, it was observed that the contracts contain minor issues, along with several areas of notes.

We recommend that the major and minor issues should be resolved. Resolving the areas of notes is up to GoodDollar's discretion. The notes refer to improving the operations of the smart contract.

Disclaimer

Somish Blockchain Labs's audit is not a security warranty, investment advice, or an endorsement of GoodDollar's platform or its products. This audit does not provide a security or correctness guarantee of the audited smart contract.

The individual audit reports are anonymized and combined during a debrief process, in order to provide an unbiased delivery and protect the auditors of Somish from legal and financial liability.
Somish Solutions Limited