Candy Finance
Smart Contracts Audit

SMCAuditors

https://smcauditors.com / info@smcauditors.com

7th of March 2021

This document is the audit of Candy Finance smart contracts performed by SMCAuditors.

1. Executive Summary

This report was written to provide a security audit for the Candy Finance Binance Smart Chain smart contracts. SMCAuditors conducted the audit focusing on whether Candy Finance smart contracts are designed and implemented in accordance with publicly released information and whether it has any security vulnerabilities. The contracts as stated are listed below with their links.

- CandyMasterFarmer.sol
  - https://github.com/candyfinance/candyfinance-contracts/blob/main/CandyMasterFarmer.sol
  - https://bscscan.com/address/0x1095a7AB736910E4364bbb29782b103AFB02CaAc
- CandyToken.sol
  - https://github.com/candyfinance/candyfinance-contracts/blob/main/CandyToken.sol

    - https://bscscan.com/address/0x0885198BbC7D33c20CfF807C0701F3A74d6858b5
- Timelock.sol
    - https://github.com/candyfinance/candyfinance-contracts/blob/main/Timelock.sol
    - https://bscscan.com/address/0x575b8F15CB2E0A2841E045fCf62931B8C70D4093

Candy Finance team requested rigorous review of their smart contracts. We have run extensive static analysis of the codebase as well as standard security assessment utilising industry approved tools. There are no high level issues with the currently deployed contracts. Our medium and low level findings are available in the next section.

2. Audit Findings

CandyMasterFarmer.sol

Medium Level Findings

1. *withdraw* function has code repetitions and too many if/else statements. It is prone to bugs and can be refactored.

Low Level Findings

1. There is no function to remove a pool. New pool addition has to be carefully reviewed considering this issue.
2. *_startBlock* and *_halvingAfterBlock* values are not being checked for correct values. *_startBlock* value can be checked that it's bigger than the current block. *_halvingAfterBlock* value can be checked that it is bigger than the *_startBlock* value. Constructor has to be carefully initiated considering these values.

CandyToken.sol

Low Level Findings

1. *delegateBySig* function does not check if the signer is the delegator. This may result in delegating another address's voting power instead of the delegator if (nonce, expiry) is not the one that was signed. You can add one more parameter in delegateBySig() function that gets the delegator's address and verifies if the recovered address is the delegator.

Timelock.sol

Medium Level Findings

1. Candy Token contract is properly administered by the CandyMasterFarmer contract that is authorized to mint new CNDY tokens per block. The CandyMasterFarmer contract is administered by the Timelock contract and this administration is also appropriate as the Timelock contract is indeed authorized to configure various aspects of CandyMasterFarmer, including the addition of new pools, the share adjustment of each existing pool (if necessary), and the setting of the upcoming migrator contract. However Timelock is not governed by an on-chain governance module. With a proper community-based on-chain governance, its admin chain should be depicted.

3. Conclusion

In this audit, we thoroughly analyzed the Candy Finance smart contracts. Overall, the smart contracts were well written and common security standards were used by the team. Our identified issues are promptly confirmed, taken into consideration and resolved accordingly.

4. Disclaimer

This report is not advice on investment, nor does it guarantee adequacy of a business model and/or a bug-free code. This report should be used only to discuss known technical problems. It will be necessary to resolve addressed issues and conduct thorough tests to ensure the safety of the smart contract.