smart state

new generation of
smart contract audit

DAO Maker

DAO Maker

Jul 30

2022

TABLE OF CONTENTS

METHODOLOGY

MAIN TESTS LIST:

- Best code practices
- ERC20/BEP20 compliance (if applicable)
- FA2 compliance (if applicable)
- Logical bugs
- General Denial Of Service(DOS)
- Locked ether
- Private data leaks
- Using components with known vulns
- Weak PRNG
- Unused vars
- Unchecked call return method
- Code with no effects
- Pool Asset Security (backdoors in the underlying ERC-20)
- Function visibility
- Use of deprecated functions
- Authorization issues
- Re-entrancy
- Arithmetic Over/Under Flows
- Hidden Malicious Code
- External Contract Referencing
- Short Address/ Parameter Attack
- Race Conditions / Front Running
- Uninitialized Storage Pointers
- Floating Points and Precision
- Signatures Replay

STRUCTURE OF CONTRACT

DAOFARM.SOL

CONTRACT METHODS ANALYSIS:

- update()
  Vulnerabilities not detected

TOKEN FLOW

- stake(uint amount)
  Vulnerabilities not detected

Tokens in, public

- requestUnstake(bool withoutClaim)
  Vulnerabilities not detected

TOKEN FLOW

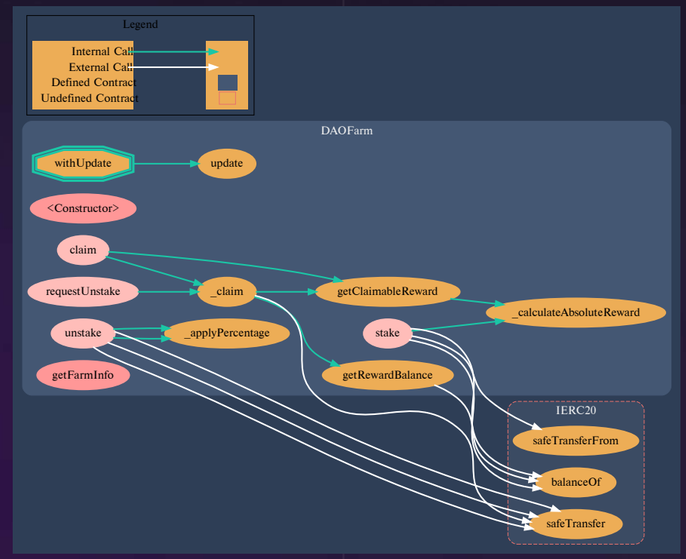
- unstake()
  Vulnerabilities not detected

Tokens out, public

TOKEN FLOW

- claim()
  Vulnerabilities not detected

Tokens out, public



Pic. 1.1
DAOFarm.sol

- _claim(address userAddress)
  Vulnerabilities not detected

- getClaimableReward(address userAddress)
  Vulnerabilities not detected

- getRewardBalance()
  Vulnerabilities not detected

- getFarmInfo(address userAddress)
  Vulnerabilities not detected

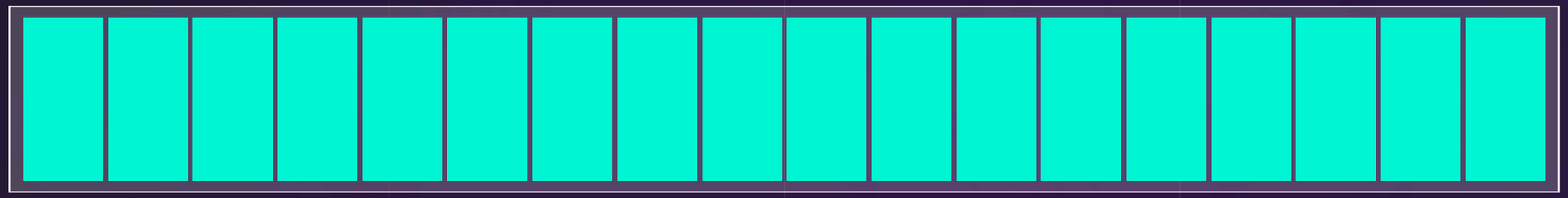
- _calculateAbsoluteReward(uint shares)
  Vulnerabilities not detected

- _applyPercentage(uint value, uint percentage)
  Vulnerabilities not detected

VERIFICATION CHECK SUMS

Contract Name	Solc version	Optimization	Bytecode hash (SHA 256)
DAOFarm.sol	0.8.14	200	5f902f8c0a11cc46f49c594ae6c4c7b5dced6aa91fc0c9a73ca2fdb5702ffbc2

PROJECT EVALUATION



10/10

smart state

GET IN TOUCH

info@smartstate.tech
smartstate.tech

View this report on smartstate.tech