smart state

> Smart
Contract
Audit #

BES

Nov 29
2021

TABLE OF CONTENTS

METHODOLOGY

MAIN TESTS LIST:

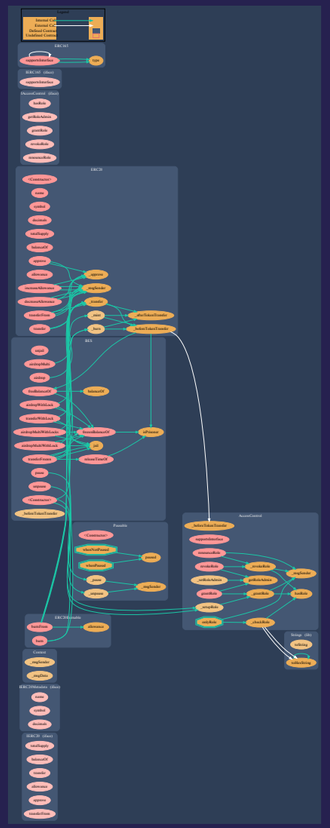
- Best code practices
- ERC20/BEP20 compliance (if applicable)
- Logical bugs
- General Denial Of Service(DOS)
- Locked ether
- Private data leaks
- Using components with known vulns
- Weak PRNG
- Unsed vars
- Uncheked call return method
- Code with no effects
- Pool Asset Security (backdoors in the 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

BES

CONTRACT METHODS ANALYSIS:

- jail(address account,uint256 until,uint256 amount)
  Jail method overrides until time. So, every time it is called until time is set again. So is user will have 1000 tokens which are going to be released in 2 weeks and admin for example calls method transferWithLock specifying our user and for example 500 tokens which should unlock in 1 month, user will now have 1500 tokens that will unlock in 1 month, instead of having 1000 tokens unlocking in 2 weeks and 500 tokens unlocking in 1 month. We find this logic incorrect and suggest that it requires refactoring in order to store each of locks individually.



Pic. 1.1
BES

- unjail(address account
  Function should be declared external
- isPrisoner(address account)
  Vulnerabilities not detected
- releaseTimeOf(address account)
  Vulnerabilities not detected
- frozenBalanceOf(address account)
  Vulnerabilities not detected
- freeBalanceOf(address account)
  Vulnerabilities not detected
- pause()
  Function should be declared external
- unpause()
  Function should be declared external
- airdropMultiWithLocks(address[] memory to,uint256[] memory amount,uint256[] memory until)
  Function should be declared external.
  Recommended to store .length of array in a variable instead of calling it every time in a loop, in order to optimize gas usage

- airdropMultiWithLock(address[] memory to,uint256[] memory amount,uint256 until)
  Function should be declared external.
  Recommended to store .length of array in a variable instead of calling it every time in a loop, in order to optimize gas usage

- airdropMulti(address[] memory to, uint256[] memory amount)
  Function should be declared external.
  Recommended to store .length of array in a variable instead of calling it every time in a loop, in order to optimize gas usage

- airdropWithLock(address[] memory to,uint256 amount,uint256 until)
  Function should be declared external.
  Recommended to store .length of array in a variable instead of calling it every time in a loop, in order to optimize gas usage

- airdrop(address[] memory to, uint256 amount)
  Function should be declared external.
  Recommended to store .length of array in a variable instead of calling it every time in a loop, in order to optimize gas usage

- transferFrozen(address recipient, uint256 amount)
  Function should be declared external

- transferWithLock(address recipient,uint256 amount,uint256 until)
  Function should be declared external

- beforeTokenTransfer(address from,address to,uint256 amount)
  Vulnerabilities not detected

VERIFICATION CHECK SUMS

Contract Name	Solc version	Optimisation	Bytecode hash (SHA 256)
BES	0.8.9	200	69d9993663a541aa5ab438fb58e2b97afa7c914c6550e4fcd39fc1e88aafa7d4

Get In Touch

info@smartstate.tech

smartstate.tech