smart state

Web3 security easier than ever

DAO Maker
Smart contract audit report

Oct 06 2023

Table of contents

Methodology

- 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

Disclaimer

This smart contract is designed to claim refunds.

Refund claim mechanism:

When a user asks for a refund:
- send him back his stable coins (refundToken) that are on the SC
- all the user's tokens are removed from SC and sent to the wallet (refundReceiver)
- at the end of the window period it is necessary to claim refundToken which haven't been refunded.

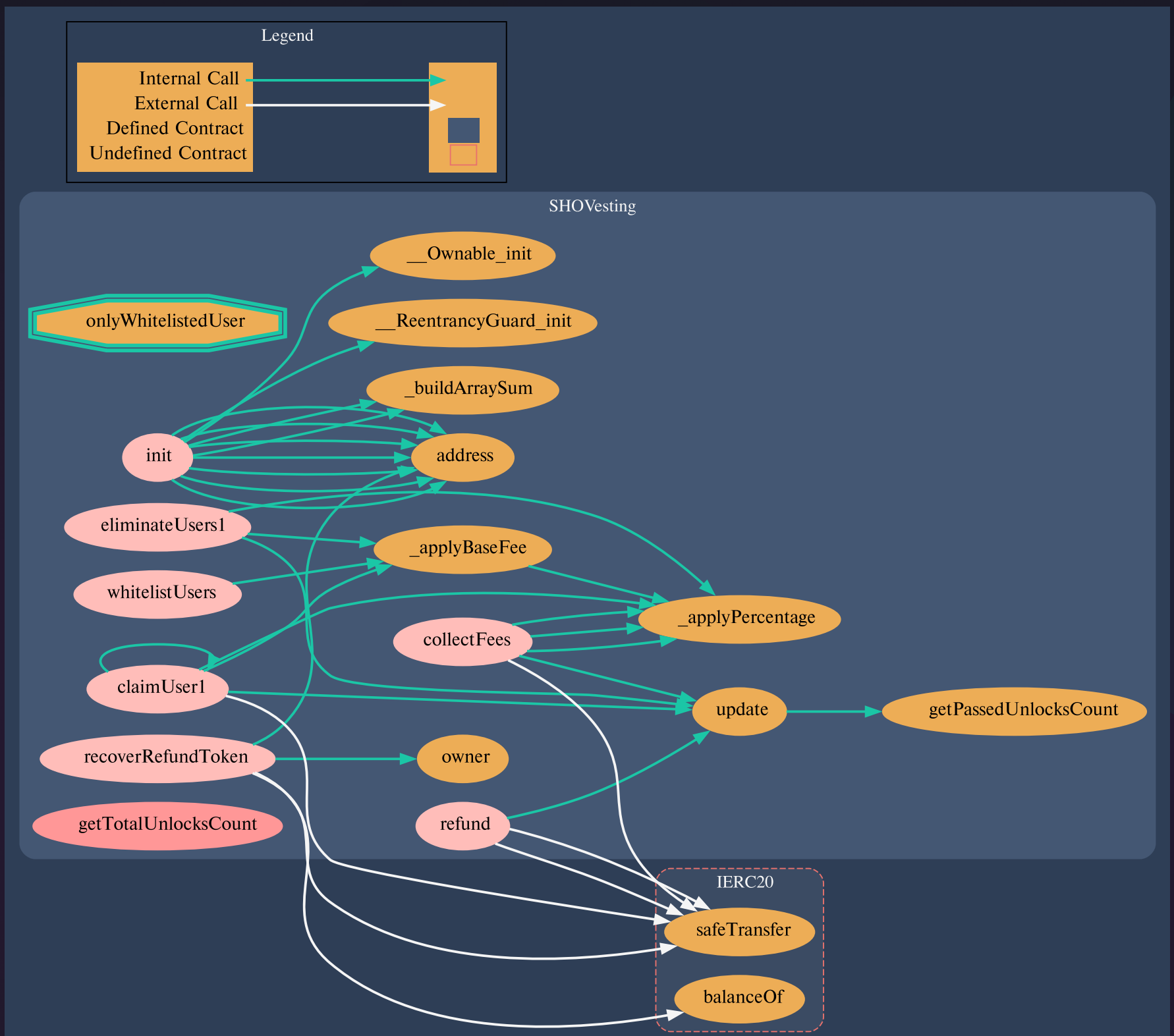
If a user has already claimed a refund once, he can't claim a refund again.

Vulnerabilities found by type

Info	0
Warning	0
Warning	0
Total	0

Structure of contract:

SHOVesting.sol



pic.1.1 SHOVesting.sol

SHOVesting.sol contract methods analysis:

init(InitParameters calldata params)	
Vulnerabilities not detected	

recoverRefundToken()	
Vulnerabilities not detected	
TOKEN FLOW	Tokens out, owner or refundReceiver

whitelistUsers(address[] calldata userAddresses, uint120[] calldata allocations, uint120[] calldata refundableAmounts, bool last)	
Vulnerabilities not detected	

claimUser1(address userAddress)	
Vulnerabilities not detected	
TOKEN FLOW	Tokens out, whitelisted users

claimUser1()	
Vulnerabilities not detected	

refund()	
Vulnerabilities not detected	
TOKEN FLOW	Tokens out, refundReceiver and user

eliminateUsers1(address[] calldata userAddresses)	
Vulnerabilities not detected	

SHOVesting.sol contract methods analysis:

collectFees()	
Vulnerabilities not detected	
TOKEN FLOW	Tokens out, public

update()
Vulnerabilities not detected

getPassedUnlocksCount()
Vulnerabilities not detected

getTotalUnlocksCount()
Vulnerabilities not detected

_applyPercentage(uint120 value, uint32 percentage)
Vulnerabilities not detected

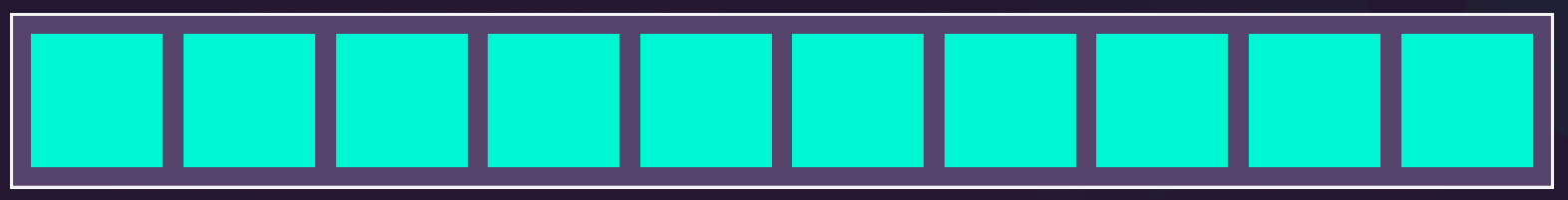
_applyBaseFee(uint120 value)
Vulnerabilities not detected

_buildArraySum(uint32[] memory diffArray)
Vulnerabilities not detected

Verification check sums

Contract name	Bytecode hash(SHA 256)	
SHOVesting.sol	38e0b9de817f645c4bec37c0d4a3e58baecccb040f5718dc069a72c7385a0bed	0.8.4

Project evaluation



10/10

Get in touch 👋

@smartstatetech

@smartstate

@SmartStateAudit

@smartstatetech

@smartstate.tech

View this report on Smartstate.tech

info@smartstate.tech

smartstate.tech