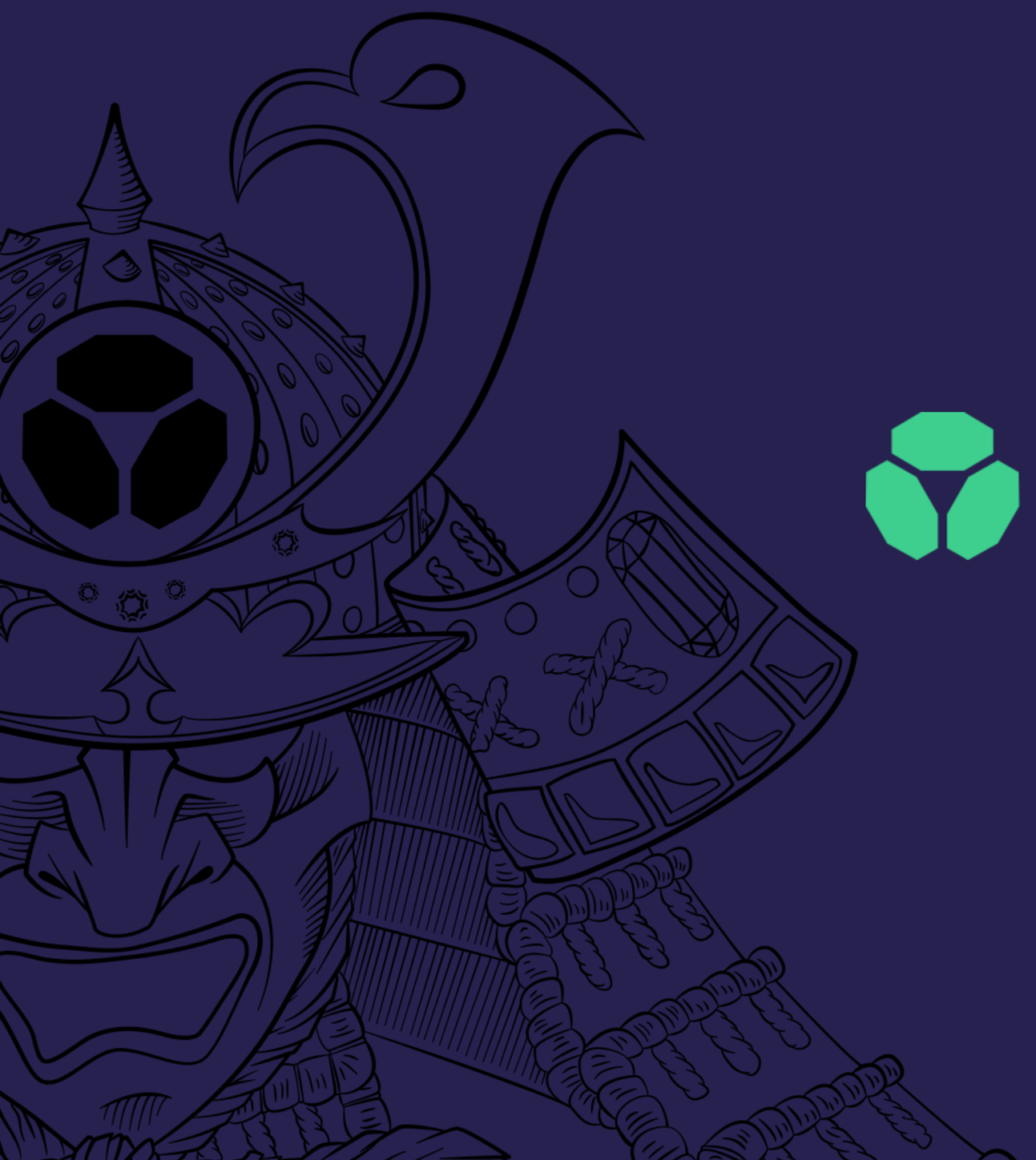


smart state

> Smart
Contract
Audit #

safle

Dec 20
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

SAFLETOKEN.SOL

CHECK SUMMARY:

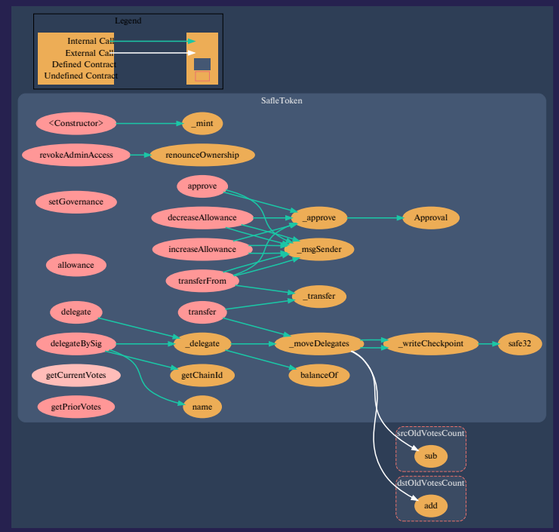
Since solidity 0.8.0 SafeMath librabry is redundant and can be removed for gas savings.

CONTRACT METHODS ANALYSIS:

- evokeAdminAccess()
  Function can be declared external

- setGovernance(address _governance)
  Function can be declared external. Function should emit an event

- transfer(address recepient, uint256 amount) override public returns (bool)
  Vulnerabilities not detected



Pic. 1.1
SafleToken.sol

WARNING

- transferFrom(address src, address dst, uint amount) override public returns (bool)
  Function should call _moveDelegates

- approve(address spender, uint256 amount) public virtual override returns (bool)
  Vulnerabilities not detected

- _approve(address owner,address spender,uint256 amount)
  Vulnerabilities not detected

- allowance(address owner, address spender) public view virtual override returns (uint256)
  Vulnerabilities not detected

- increaseAllowance(address spender, uint256 addedValue) public virtual override returns (bool)
  Vulnerabilities not detected

- decreaseAllowance(address spender, uint256 subtractedValue) public virtual override returns (bool)
  Vulnerabilities not detected

- delegate(address delegatee)
  Function can be declared external

- delegateBySig(address delegatee, uint256 nonce, uint256 expiry, uint8 v, bytes32 r, bytes32 s)
  Function can be declared external

- getCurrentVotes(address account) external view returns (uint256)
  Vulnerabilities not detected
- getPriorVotes(address account, uint256 blockNumber) public view returns (uint256)
  Function can be declared external
- _delegate(address delegator, address delegatee)
  Vulnerabilities not detected
- _moveDelegates(address source, address destination, uint256 amount)
  Vulnerabilities not detected
- _writeCheckpoint(address delegatee, uint256 nCheckpoints, uint256 oldVotes, uint256 newVotes)
  Vulnerabilities not detected
- getChainId() internal view returns (uint)
  Vulnerabilities not detected
- safe32(uint n, string memory errorMessage) internal pure returns (uint32)
  Vulnerabilities not detected

STRUCTURE OF CONTRACT

VESTING.SOL

CHECK SUMMARY:

Since solidity 0.8.0 SafeMath librabry is

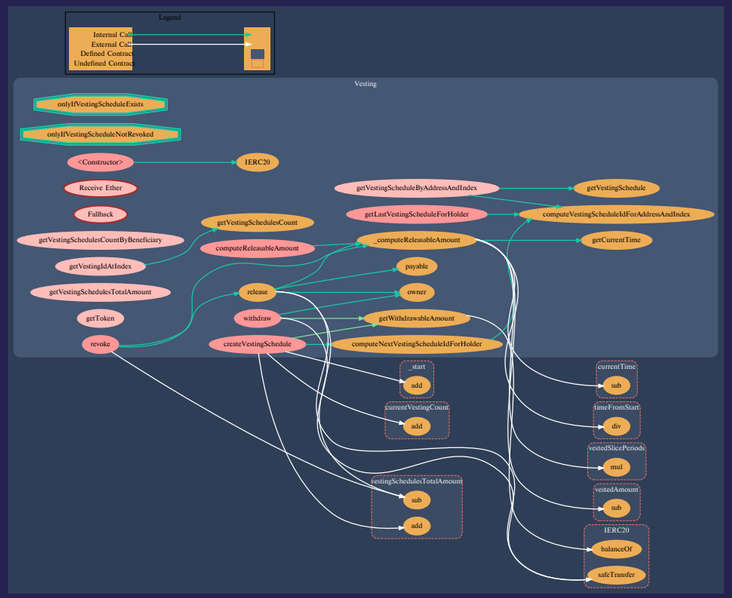
redundant and can be removed for gas savings.

CONTRACT METHODS ANALYSIS:

- getVestingSchedulesCountByBeneficiary(address _beneficiary)
  Vulnerabilities not detected

- getVestingIdAtIndex(uint256 index)
  Vulnerabilities not detected

- getVestingScheduleByAddressAndIndex(address holder, uint256 index)
  Vulnerabilities not detected



Pic. 1.2
Vesting.sol

- getVestingSchedulesTotalAmount()
  Vulnerabilities not detected

- getToken()
  Vulnerabilities not detected

- createVestingSchedule(address _beneficiary,uint256 _start,uint256 _cliff,uint256 _duration,uint256 _slicePeriodSeconds,bool _revocable,uint256 _amount)
  Function can be declared external. Function should emit an event

PAYABLE

- revoke(bytes32 vestingScheduleId)
  Function can be declared external. Function should emit an event

Tokens out, only owner

PAYABLE

- withdraw(uint256 amount)
  Function can be declared external. Function should emit an event

Tokens out, only owner

PAYABLE

- release(bytes32 vestingScheduleId,uint256 amount)
  Function can be declared external. Function should emit an event. Converting beneficiary address to pyable is redundant and can be removed.

Tokens out, public

- getVestingSchedulesCount()
  Vulnerabilities not detected

- computeReleasableAmount(bytes32 vestingScheduleId)
  Vulnerabilities not detected

- getVestingSchedule(bytes32 vestingScheduleId)
  Vulnerabilities not detected

- getWithdrawableAmount()
  Vulnerabilities not detected

- computeNextVestingScheduleIdForHolder(address holder)
  Vulnerabilities not detected

- getLastVestingScheduleForHolder(address holder)
  Vulnerabilities not detected

- computeVestingScheduleIdForAddressAndIndex(address holder, uint256 index)
  Vulnerabilities not detected

- _computeReleasableAmount(VestingSchedule memory vestingSchedule)
  Vulnerabilities not detected

- getCurrentTime()
  Vulnerabilities not detected

VERIFICATION CHECK SUMS

Contract Name	Solc version	Optimisation	Bytecode hash (SHA 256)
SafleToken	0.8.0	200	2750bc6788a398d4b32915d2d11588af1574950d30c9f59e5b56663e41f39b50
Vesting	0.8.0	200	f75d8f32f1417300fe14e262a252ae702bf113c5669b845a10c41cf4aaebeca8

Get In Touch

info@smartstate.tech

smartstate.tech