Smart Contract Security Audit V1

Aomen City Token

19/2/2022

https://saferico.com/

business@saferico.com
https://t.me/SFI_ANN

Table of Contents

Background

The purpose of the audit was to achieve the following:

- Ensure that the smart contract functions as intended.
- Identify potential security issues with the smart contract.

The information in this report should be used to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

Project Information

- **Platform**: Binance Smart Chain
- **Contract Address**: 0x681BDc66dD0C1380b67654b3601f20578Ceb4dEd
- **Code Source:**

https://bscscan.com/address/0x681BDc66dD0C1380b67654b3601f20578Ceb4dEd#code

Token Information

- Name: $AMC
- Total Supply: 1,000,000,000
- Holders:
- Total transactions:

Contracts address deployed to test net (BSC)

Aomen City smart contract on testnet.bsc by the auditor to test every function (BSC Test Net)

https://testnet.bscscan.com/address/0xb529c49b5dc8ec6d49a17f4ecafc6a4d59cad195

Executive Summary

According to our assessment, the customer`s solidity smart contract is **Secured**.

Well Secured	
Secured	✓
Poor Secured	
Insecure	

Automated checks are with remix IDE. All issues were performed by the team, which included the analysis of code functionality, manual audit found during automated analysis were manually reviewed and applicable vulnerabilities are presented in the audit overview section. The general overview is presented in the Project Information section and all issues found are located in the audit overview section.

Team found 0 critical, 0 high, 0 medium, 2 low, 0 very low-level issues and 3 notes in all solidity files of the contract

The files:

AomenCity.sol

File and Function Level Report

File in Scope:

Contract Name	SHA 256 hash	Contract Address
AomenCity.sol	**0c7a1d16a7c2caec5e20b69f0db0408cd0e3a7df6305a56362dd0fb9f9467253**	0x681BDc66dD0C1380b67654b3601f20578Ceb4dEd

- Contract: AomenCity
- Inherit: Context, IERC20, Ownable
- Observation: All passed including security check
- Test Report: passed
- Score: passed
- Conclusion: passed

Function	Test Result	Type / Return Type	Score
name	✓	Read / public	**Passed**
symbol	✓	Read / public	**Passed**
decimals	✓	Read / public	**Passed**
totalSupply	✓	Read / public	**Passed**
allowance	✓	Read / public	**Passed**
balanceOf	✓	Read / public	**Passed**
Owner	✓	Read / public	**Passed**
reflectionFromToken	✓	Read / public	**Passed**
numCheckpoints	✓	Read / public	**Passed**
nonces	✓	Read / public	**Passed**
tokenFromReflection	✓	Read / public	**Passed**
_liquidityFee	✓	Read / public	**Passed**

_burnFeeTotal	✓	Read / public	**Passed**
_taxFeeTotal	✓	Read / public	**Passed**
_treasuryFee2	✓	Read / public	**Passed**
_treasuryFee	✓	Read / public	**Passed**
_treasury2FeeTotal	✓	Read / public	**Passed**
_treasuryFeeTotal	✓	Read / public	**Passed**
_burnFee	✓	Read / public	**Passed**
DOMAIN_TYPEHASH	✓	Read / public	**Passed**
isBlackListed	✓	Read / public	**Passed**
getCurrentVotes	✓	Read / public	**Passed**
getPriorVotes	✓	Read / public	**Passed**
isExcluded	✓	Read / public	**Passed**
liquidityAddress	✓	Read / public	**Passed**
DELEGATION_TYPEHASH	✓	Read / public	**Passed**
delegates	✓	Read / public	**Passed**
checkpoints	✓	Read / public	**Passed**
BurnAddress	✓	Read / public	**Passed**
antiDump	✓	Read / public	**Passed**
blacklister	✓	Read / public	**Passed**
_taxFee	✓	Read / public	**Passed**
_liquidityFeeTotal	✓	Read / public	**Passed**
treasury2Address	✓	Read / public	**Passed**
treasuryAddress	✓	Read / public	**Passed**
approve	✓	Write / public	**Passed**
transferFrom	✓	Write / public	**Passed**
transfer	✓	Write / public	**Passed**
updateBlacklister	✓	Write / public	**Passed**
excludeFromFee	✓	Write / public	**Passed**
excludeAccount	✓	Write / public	**Passed**

includeInFee	✓	Write / public	**Passed**
renounceOwnership	✓	Write / public	**Passed**
transferOwnership	✓	Write / public	**Passed**
_burn	✓	Write / public	**Passed**
unBlacklist	✓	Write / public	**Passed**
decreaseAllowance	✓	Write / public	**Passed**
TurnOffFees	✓	Write / public	**Passed**
setTreasury2Fee	✓	Write / public	**Passed**
blackList	✓	Write / public	**Passed**
setTreasury1Fee	✓	Write / public	**Passed**
increaseAllowance	✓	Write / public	**Passed**
settreasury1Address	✓	Write / public	**Passed**
setReflectionFee	✓	Write / public	**Passed**
setLiquidityFee	✓	Write / public	**Passed**
setLiquidityAddress	✓	Write / public	**Passed**
setBurnPercent	✓	Write / public	**Passed**
includeAccount	✓	Write / public	**Passed**
delegateBySig	✓	Write / public	**Passed**
delegate	✓	Write / public	**Passed**

Issues Checking Status

No.	Issue Description	Checking Status
1	Compiler warnings.	**Passed**
2	Race conditions and Reentrancy. Cross-function race conditions.	**Passed**
3	Possible delays in data delivery.	**Passed**
4	Oracle calls.	**Passed**
5	Design Logic.	**Passed**
6	Timestamp dependence.	**Passed**
7	Integer Overflow and Underflow.	**Passed**
8	DoS with Revert.	**Passed**
9	DoS with block gas limit.	**Passed with notes**
10	Methods execution permissions.	**Passed**
11	Economy model. If application logic is based on an incorrect economic model, the application would not function correctly and participants would incur financial losses. This type of issue is most often found in bonus rewards systems, Staking and Farming contracts, Vault and Vesting contracts, etc.	**Passed**
12	The impact of the exchange rate on the logic.	**Passed**
13	Private user data leaks.	**Passed**
14	Malicious Event log.	**Passed**
15	Scoping and Declarations.	**Passed**
16	Uninitialized storage pointers.	**Passed**
17	Arithmetic accuracy.	**Passed**

Severity Definitions

Risk Level	Description
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to tokens loss etc.
High	High-level vulnerabilities are difficult to exploit; however, they also have significant impact on smart contract execution, e.g. public access to crucial functions
Medium	Medium-level vulnerabilities are important to fix; however, they can’t lead to tokens lose
Low	Low-level vulnerabilities are mostly related to outdated, unused etc. code snippets, that can’t have significant impact on execution
Note	Lowest-level vulnerabilities, code style violations and info statements can’t affect smart contract execution and can be ignored.

Audit Findings

Critical:

No critical severity vulnerabilities were found.

High:

No High severity vulnerabilities were found

Medium:

No Medium severity vulnerabilities were found.

Low:

#Use of block.timestamp for comparisons

Description
The value of block.timestamp can be manipulated by the miner.
And conditions with strict equality is difficult to achieve - block.timestamp

Remediation
Avoid use of block.timestamp

Status: Acknowledged

#Owner privileges (In the period when the owner isn’t renounced)

Description
Owner can change Fees or make it = zero.
Owner can add any address to Blacklist.
Owner can include / exclude any address from Fees or Reward.

```
function ExcludedFromFee(address account, bool) public onlyOwner {
        isExcludedFromFee[account] = true;
    }

    function IncludeFromFee(address account, bool) public onlyOwner {
        isExcludedFromFee[account] = false;
    }

    function setReflectionFee(uint256 fee) public onlyOwner {
        _taxFee = fee;
    }
```

```
   function setLiquidityFee(uint256 fee) public onlyOwner {
       _liquidityFee = fee;
   }

   function setTreasury1Fee(uint256 fee) public onlyOwner {
       _treasuryFee = fee;
   }

   function setTreasury2Fee(uint256 fee) public onlyOwner {
       _treasury2Fee = fee;
   }

       function setBurnPercent(uint256 fee) public onlyOwner {
       _BurnFee = fee;
   }

   function settreasury1Address(address _Address) public onlyOwner {
       require(_Address != treasuryAddress);

       treasuryAddress = _Address;
   }

   function setLiquidityAddress(address _Address) public onlyOwner {
       require(_Address != liquidityAddress);

       liquidityAddress = _Address;
   }
function TurnOffFees() external onlyOwner {
       _BurnFee = 0;
       _taxFee = 0;
       _liquidityFee = 0;
       _treasury2Fee = 0;
       _treasuryFee = 0;
   }
```

Remediation

Make these functions internal in next version or the team should announce the investors before change the fees and give them time if they want to use the old fees.

P.S: This issue is common to the majority of rewards smart contracts.

Status: Acknowledged.

Very Low:

No Very Low severity vulnerabilities were found.

Notes:

#Unnecessary use of SafeMath

Description
Solidity version 0.8 was released with SafeMath checks inbuilt, we can avoid using an explicit safe math library.

Remediation
Remove SafeMath Library to save gas fees.

Status: Acknowledged

#Naming Conventions

Description
The contract follows a consistent naming convention where we are private variables with leading"_" and public variables without it. But we have missed to comply to the condition for certain variable names "__burn" which is public.

Remediation
Remove "_" from external variable names and add it to private variable names.

Status: Acknowledged

# Constant calculations in the contract

Description
recalculated initialization will save 2847 units of gas in deployment

```
uint256 internal _tokenTotal = 1000000000 *10**18;
```

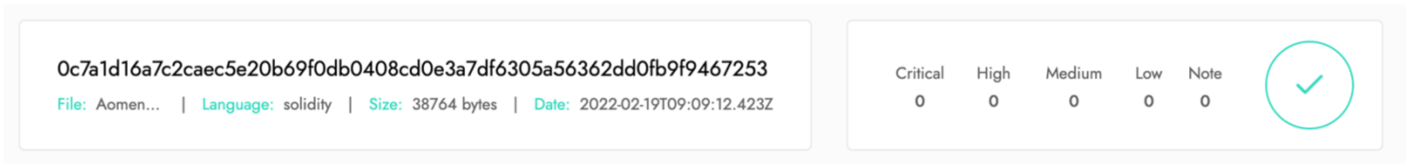
Recommendation
Replace the initialization as

```
uint256 internal _tokenTotal = 1000000000000000000000000000;
```

Status: Acknowledged

Automatic Testing

1- Check for security

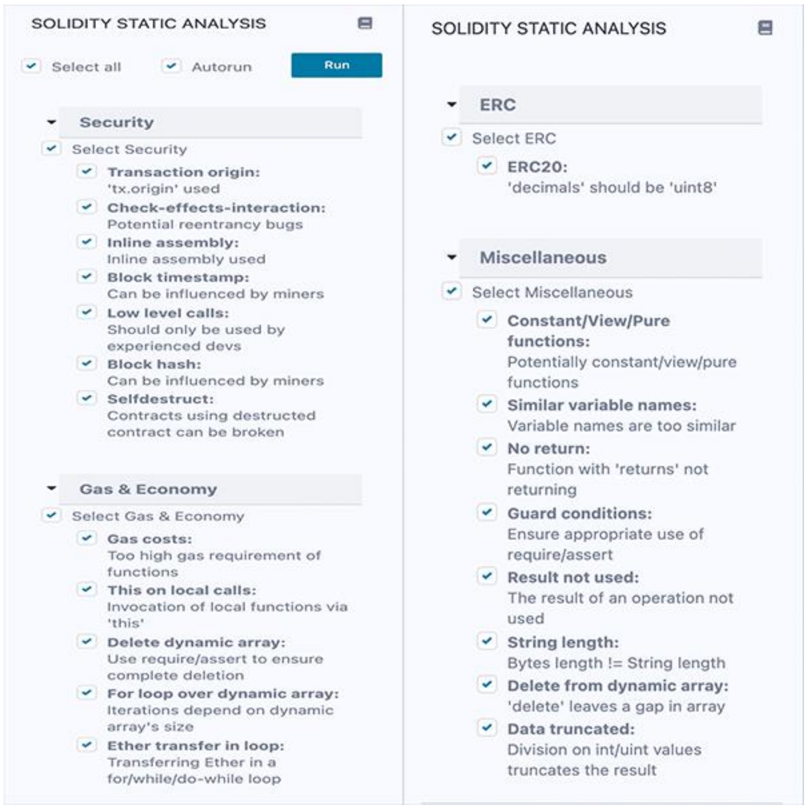


0c7a1d16a7c2caec5e20b69f0db0408cd0e3a7df6305a56362dd0fb9f9467253

File: Aomen... | Language: solidity | Size: 38764 bytes | Date: 2022-02-19T09:09:12.423Z

Critical	High	Medium	Low	Note
0	0	0	0	0

2- SOLIDITY STATIC ANALYSIS



SOLIDITY STATIC ANALYSIS

✓ Select all ✓ Autorun Run

Security

✓ Select Security

- ✓ **Transaction origin:** 'tx.origin' used
- ✓ **Check-effects-interaction:** Potential reentrancy bugs
- ✓ **Inline assembly:** Inline assembly used
- ✓ **Block timestamp:** Can be influenced by miners
- ✓ **Low level calls:** Should only be used by experienced devs
- ✓ **Block hash:** Can be influenced by miners
- ✓ **Selfdestruct:** Contracts using destructed contract can be broken

Gas & Economy

✓ Select Gas & Economy

- ✓ **Gas costs:** Too high gas requirement of functions
- ✓ **This on local calls:** Invocation of local functions via 'this'
- ✓ **Delete dynamic array:** Use require/assert to ensure complete deletion
- ✓ **For loop over dynamic array:** Iterations depend on dynamic array's size
- ✓ **Ether transfer in loop:** Transferring Ether in a for/while/do-while loop

SOLIDITY STATIC ANALYSIS

ERC

✓ Select ERC

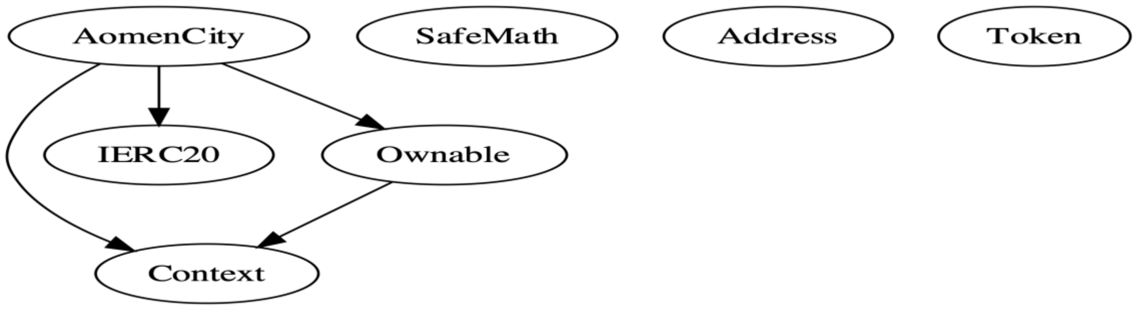
- ✓ **ERC20:** 'decimals' should be 'uint8'

Miscellaneous

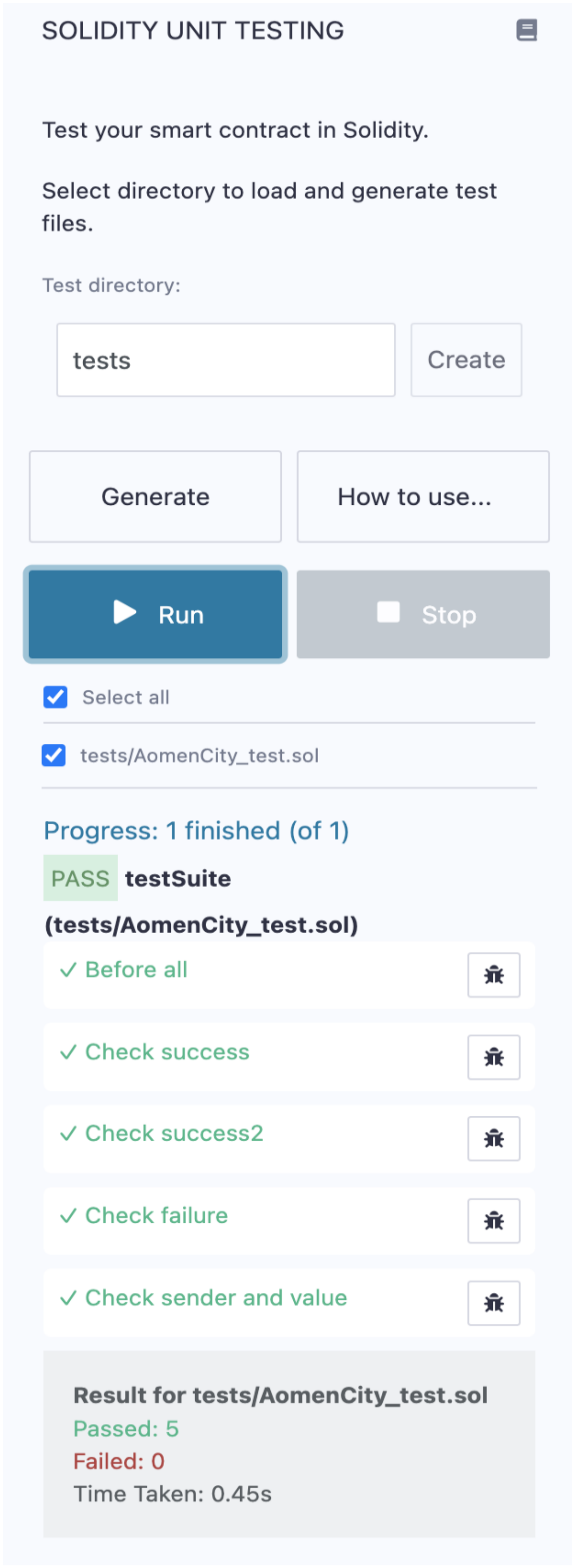
✓ Select Miscellaneous

- ✓ **Constant/View/Pure functions:** Potentially constant/view/pure functions
- ✓ **Similar variable names:** Variable names are too similar
- ✓ **No return:** Function with 'returns' not returning
- ✓ **Guard conditions:** Ensure appropriate use of require/assert
- ✓ **Result not used:** The result of an operation not used
- ✓ **String length:** Bytes length != String length
- ✓ **Delete from dynamic array:** 'delete' leaves a gap in array
- ✓ **Data truncated:** Division on int/uint values truncates the result

3- Inheritance graph



4- SOLIDITY UNIT TESTING


SOLIDITY UNIT TESTING

Test your smart contract in Solidity.

Select directory to load and generate test files.

Test directory: tests

tests/AomenCity_test.sol

Progress: 1 finished (of 1)

PASS testSuite (tests/AomenCity_test.sol)

✓ Before all

✓ Check success

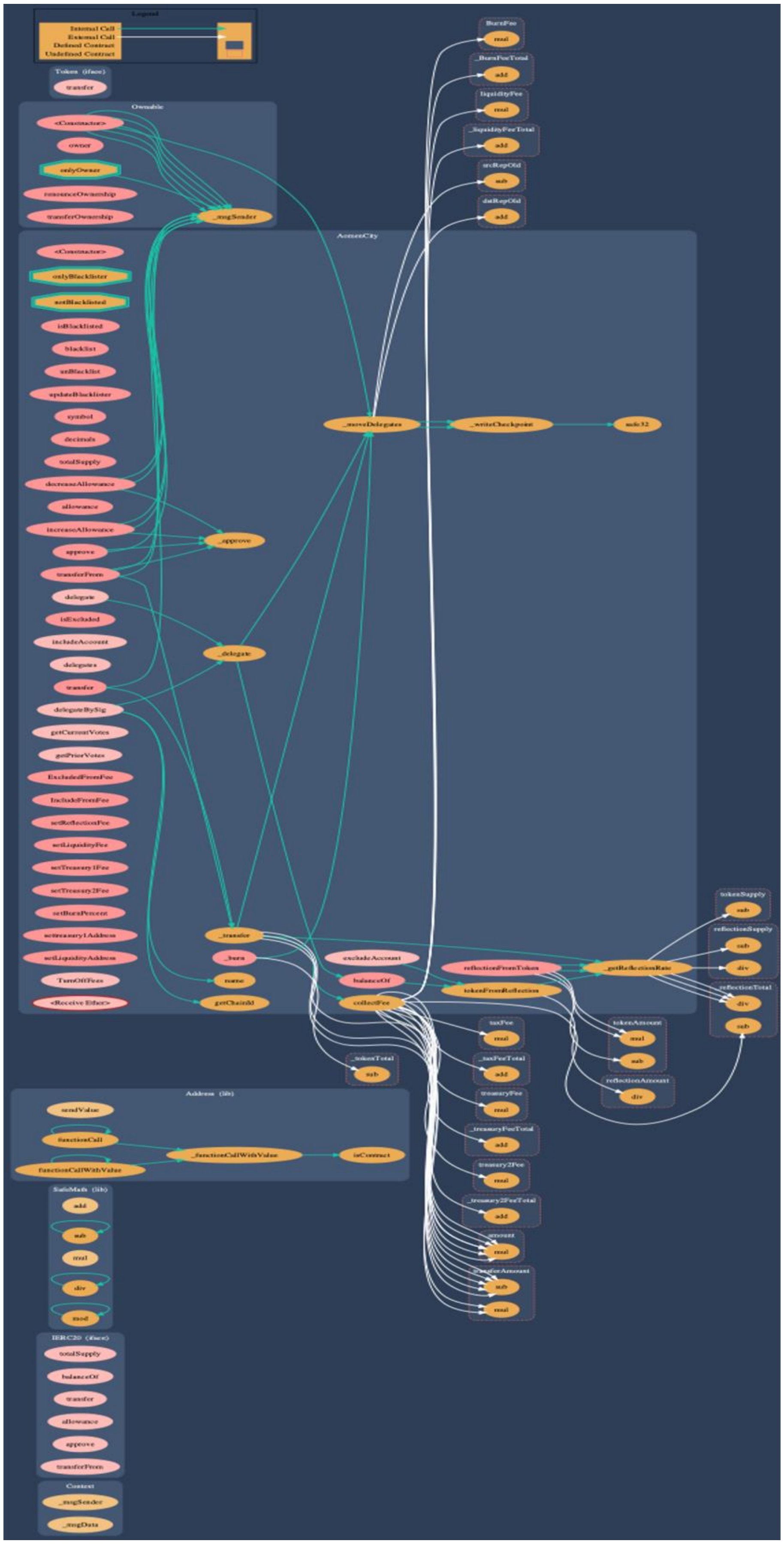
✓ Check success2

✓ Check failure

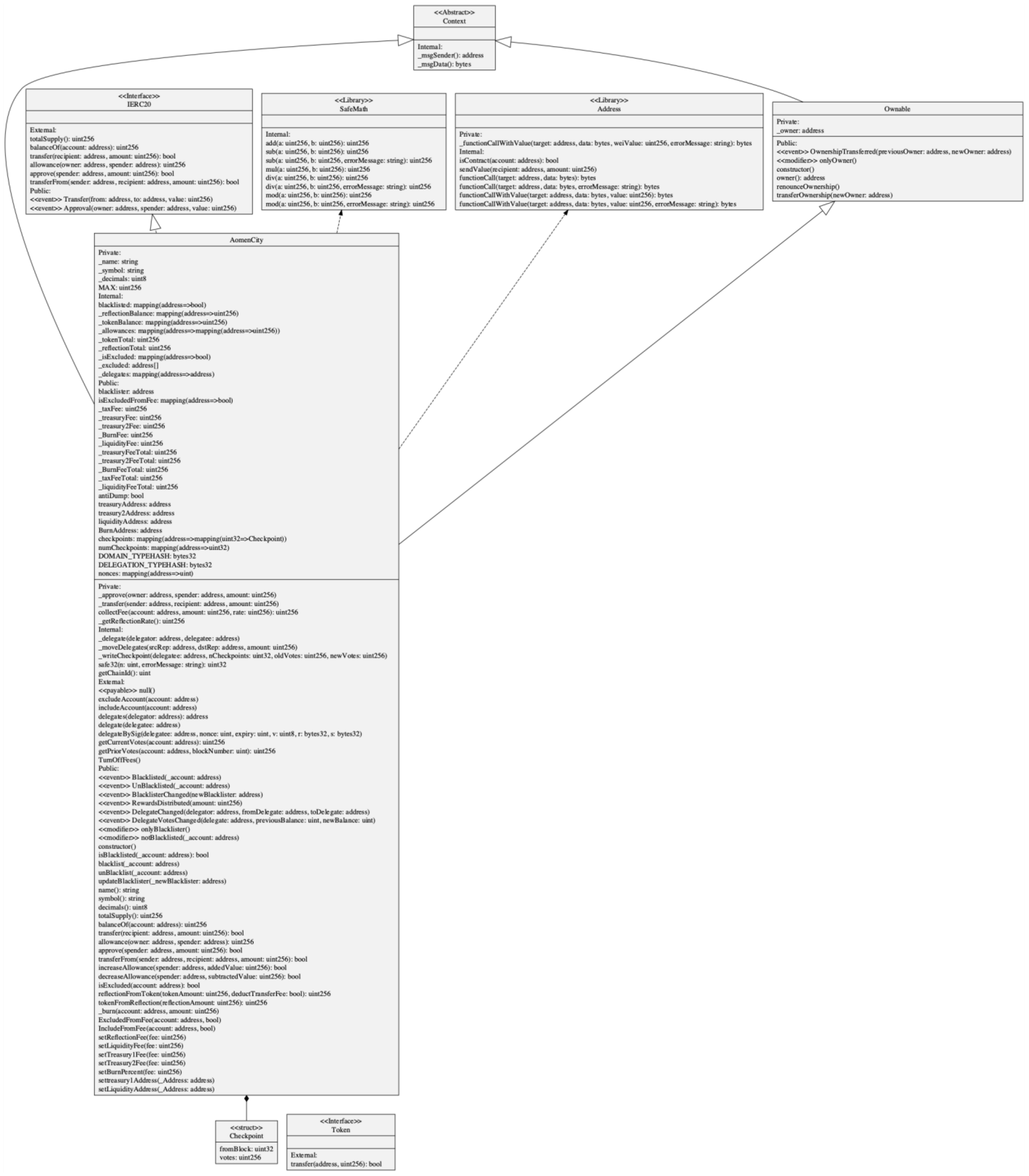
✓ Check sender and value

Result for tests/AomenCity_test.sol
Passed: 5
Failed: 0
Time Taken: 0.45s

5- Call graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
Tokens (iface)
Ownable
AcmenCity
Address (lib)
SafeMath (lib)
IERC20 (iface)
Context
_moveDelegates
_writeCheckpoint
safe32
_approve
_delegate
_transfer
_burn
collectFee
reflectionFromToken
tokenFromReflection
_getReflectionRate

Unified Modeling Language (UML)


<<Abstract>>
Context
Internal:
_msgSender(): address
_msgData(): bytes
<<Interface>>
IERC20
External:
totalSupply(): uint256
balanceOf(account: address): uint256
transfer(recipient: address, amount: uint256): bool
allowance(owner: address, spender: address): uint256
approve(spender: address, amount: uint256): bool
transferFrom(sender: address, recipient: address, amount: uint256): bool
Public:
<<event>> Transfer(from: address, to: address, value: uint256)
<<event>> Approval(owner: address, spender: address, value: uint256)
<<Library>>
SafeMath
Internal:
add(a: uint256, b: uint256): uint256
sub(a: uint256, b: uint256): uint256
sub(a: uint256, b: uint256, errorMessage: string): uint256
mul(a: uint256, b: uint256): uint256
div(a: uint256, b: uint256): uint256
div(a: uint256, b: uint256, errorMessage: string): uint256
mod(a: uint256, b: uint256): uint256
mod(a: uint256, b: uint256, errorMessage: string): uint256
<<Library>>
Address
Private:
_functionCallWithValue(target: address, data: bytes, weiValue: uint256, errorMessage: string): bytes
Internal:
isContract(account: address): bool
sendValue(recipient: address, amount: uint256)
functionCall(target: address, data: bytes): bytes
functionCall(target: address, data: bytes, errorMessage: string): bytes
functionCallWithValue(target: address, data: bytes, value: uint256): bytes
functionCallWithValue(target: address, data: bytes, value: uint256, errorMessage: string): bytes
Ownable
Private:
_owner: address
Public:
<<event>> OwnershipTransferred(previousOwner: address, newOwner: address)
<<modifier>> onlyOwner()
constructor()
owner(): address
renounceOwnership()
transferOwnership(newOwner: address)
AomenCity
Private:
_name: string
_symbol: string
_decimals: uint8
MAX: uint256
Internal:
blacklisted: mapping(address=>bool)
_reflectionBalance: mapping(address=>uint256)
_tokenBalance: mapping(address=>uint256)
_allowances: mapping(address=>mapping(address=>uint256))
_tokenTotal: uint256
_reflectionTotal: uint256
_isExcluded: mapping(address=>bool)
_excluded: address[]
_delegates: mapping(address=>address)
Public:
blacklister: address
isExcludedFromFee: mapping(address=>bool)
_taxFee: uint256
_treasuryFee: uint256
_treasury2Fee: uint256
_BurnFee: uint256
_liquidityFee: uint256
_treasuryFeeTotal: uint256
_treasury2FeeTotal: uint256
_BurnFeeTotal: uint256
_taxFeeTotal: uint256
_liquidityFeeTotal: uint256
antiDump: bool
treasuryAddress: address
treasury2Address: address
liquidityAddress: address
BurnAddress: address
checkpoints: mapping(address=>mapping(uint32=>Checkpoint))
numCheckpoints: mapping(address=>uint32)
DOMAIN_TYPEHASH: bytes32
DELEGATION_TYPEHASH: bytes32
nonces: mapping(address=>uint)
Private:
_approve(owner: address, spender: address, amount: uint256)
_transfer(sender: address, recipient: address, amount: uint256)
collectFee(account: address, amount: uint256, rate: uint256): uint256
_getReflectionRate(): uint256
Internal:
_delegate(delegator: address, delegatee: address)
_moveDelegates(srcRep: address, dstRep: address, amount: uint256)
_writeCheckpoint(delegatee: address, nCheckpoints: uint32, oldVotes: uint256, newVotes: uint256)
safe32(n: uint, errorMessage: string): uint32
getChainId(): uint
External:
<<payable>> null()
excludeAccount(account: address)
includeAccount(account: address)
delegates(delegator: address): address
delegate(delegatee: address)
delegateBySig(delegatee: address, nonce: uint, expiry: uint, v: uint8, r: bytes32, s: bytes32)
getCurrentVotes(account: address): uint256
getPriorVotes(account: address, blockNumber: uint): uint256
TurnOffFees()
Public:
<<event>> Blacklisted(_account: address)
<<event>> UnBlacklisted(_account: address)
<<event>> BlacklisterChanged(newBlacklister: address)
<<event>> RewardsDistributed(amount: uint256)
<<event>> DelegateChanged(delegator: address, fromDelegate: address, toDelegate: address)
<<event>> DelegateVotesChanged(delegate: address, previousBalance: uint, newBalance: uint)
<<modifier>> onlyBlacklister()
<<modifier>> notBlacklisted(_account: address)
constructor()
isBlacklisted(_account: address): bool
blacklist(_account: address)
unBlacklist(_account: address)
updateBlacklister(_newBlacklister: address)
name(): string
symbol(): string
decimals(): uint8
totalSupply(): uint256
balanceOf(account: address): uint256
transfer(recipient: address, amount: uint256): bool
allowance(owner: address, spender: address): uint256
approve(spender: address, amount: uint256): bool
transferFrom(sender: address, recipient: address, amount: uint256): bool
increaseAllowance(spender: address, addedValue: uint256): bool
decreaseAllowance(spender: address, subtractedValue: uint256): bool
isExcluded(account: address): bool
reflectionFromToken(tokenAmount: uint256, deductTransferFee: bool): uint256
tokenFromReflection(reflectionAmount: uint256): uint256
_burn(account: address, amount: uint256)
ExcludedFromFee(account: address, bool)
IncludeFromFee(account: address, bool)
setReflectionFee(fee: uint256)
setLiquidityFee(fee: uint256)
setTreasury1Fee(fee: uint256)
setTreasury2Fee(fee: uint256)
setBurnPercent(fee: uint256)
settreasury1Address(_Address: address)
setLiquidityAddress(_Address: address)
<<struct>>
Checkpoint
fromBlock: uint32
votes: uint256
<<Interface>>
Token
External:
transfer(address, uint256): bool

Functions signature

```
Sighash   |   Function Signature
========================
16279055  =>  isContract(address)
39509351  =>  increaseAllowance(address,uint256)
119df25f  =>  _msgSender()
8b49d47e  =>  _msgData()
18160ddd  =>  totalSupply()
70a08231  =>  balanceOf(address)
a9059cbb  =>  transfer(address,uint256)
dd62ed3e  =>  allowance(address,address)
095ea7b3  =>  approve(address,uint256)
23b872dd  =>  transferFrom(address,address,uint256)
771602f7  =>  add(uint256,uint256)
b67d77c5  =>  sub(uint256,uint256)
e31bdc0a  =>  sub(uint256,uint256,string)
c8a4ac9c  =>  mul(uint256,uint256)
a391c15b  =>  div(uint256,uint256)
b745d336  =>  div(uint256,uint256,string)
f43f523a  =>  mod(uint256,uint256)
71af23e8  =>  mod(uint256,uint256,string)
24a084df  =>  sendValue(address,uint256)
a0b5ffb0  =>  functionCall(address,bytes)
241b5886  =>  functionCall(address,bytes,string)
2a011594  =>  functionCallWithValue(address,bytes,uint256)
d525ab8a  =>  functionCallWithValue(address,bytes,uint256,string)
36455e42  =>  _functionCallWithValue(address,bytes,uint256,string)
8da5cb5b  =>  owner()
715018a6  =>  renounceOwnership()
f2fde38b  =>  transferOwnership(address)
fe575a87  =>  isBlacklisted(address)
f9f92be4  =>  blacklist(address)
1a895266  =>  unBlacklist(address)
ad38bf22  =>  updateBlacklister(address)
06fdde03  =>  name()
95d89b41  =>  symbol()
313ce567  =>  decimals()
a457c2d7  =>  decreaseAllowance(address,uint256)
cba0e996  =>  isExcluded(address)
4549b039  =>  reflectionFromToken(uint256,bool)
2d838119  =>  tokenFromReflection(uint256)
f2cc0c18  =>  excludeAccount(address)
f84354f1  =>  includeAccount(address)
104e81ff  =>  _approve(address,address,uint256)
30e0789e  =>  _transfer(address,address,uint256)
6161eb18  =>  _burn(address,uint256)
9a2178cc  =>  collectFee(address,uint256,uint256)
1417d2a8  =>  _getReflectionRate()
587cde1e  =>  delegates(address)
5c19a95c  =>  delegate(address)
c3cda520  =>  delegateBySig(address,uint256,uint256,uint8,bytes32,bytes32)
b4b5ea57  =>  getCurrentVotes(address)
782d6fe1  =>  getPriorVotes(address,uint256)
a28a42b3  =>  _delegate(address,address)
955f9fd8  =>  _moveDelegates(address,address,uint256)
ee59e77f  =>  _writeCheckpoint(address,uint32,uint256,uint256)
869d1f83  =>  safe32(uint256,string)
```

```
3408e470  =>  getChainId()
2d43abd8  =>  ExcludedFromFee(address,bool)
8112287d  =>  IncludeFromFee(address,bool)
e547be69  =>  setReflectionFee(uint256)
357bf15c  =>  setLiquidityFee(uint256)
0bbd2411  =>  setTreasury1Fee(uint256)
6cb0832d  =>  setTreasury2Fee(uint256)
bb1570da  =>  setBurnPercent(uint256)
421f4e2f  =>  settreasury1Address(address)
525fa81f  =>  setLiquidityAddress(address)
87b0f24c  =>  TurnOffFees()
```

Automatic general report

```
Files Description Table

|  File Name  |  SHA-1 Hash  |
|-------------|--------------|
| /Users/macbook/Desktop/smart contracts/AomenCity.sol |
cd93db56b572cad3ce50a0883b49bd1b31e2a2fd |

 Contracts Description Table

|  Contract  |         Type        |       Bases      |                  |
|
|:----------:|:-------------------:|:----------------:|:----------------:|:--------
-------:|
|     └      |  **Function Name**  |  **Visibility**  |  **Mutability**  |
**Modifiers**  |
||||||
| **Context** | Implementation |  |||
| └ | _msgSender | Internal 🔒 |   | |
| └ | _msgData | Internal 🔒 |   | |
||||||
| **IERC20** | Interface |  |||
| └ | totalSupply | External ❗️ |   |NO❗️ |
| └ | balanceOf | External ❗️ |   |NO❗️ |
| └ | transfer | External ❗️ | 🛑  |NO❗️ |
| └ | allowance | External ❗️ |   |NO❗️ |
| └ | approve | External ❗️ | 🛑  |NO❗️ |
| └ | transferFrom | External ❗️ | 🛑  |NO❗️ |
||||||
| **SafeMath** | Library |  |||
| └ | add | Internal 🔒 |   | |
| └ | sub | Internal 🔒 |   | |
| └ | sub | Internal 🔒 |   | |
| └ | mul | Internal 🔒 |   | |
| └ | div | Internal 🔒 |   | |
| └ | div | Internal 🔒 |   | |
| └ | mod | Internal 🔒 |   | |
| └ | mod | Internal 🔒 |   | |
||||||
| **Address** | Library |  |||
| └ | isContract | Internal 🔒 |   | |
| └ | sendValue | Internal 🔒 | 🛑  | |
| └ | functionCall | Internal 🔒 | 🛑  | |
| └ | functionCall | Internal 🔒 | 🛑  | |
| └ | functionCallWithValue | Internal 🔒 | 🛑  | |
| └ | functionCallWithValue | Internal 🔒 | 🛑  | |
| └ | _functionCallWithValue | Private 🔐 | 🛑  | |
||||||
| **Ownable** | Implementation | Context |||
| └ | <Constructor> | Public ❗️ | 🛑  |NO❗️ |
| └ | owner | Public ❗️ |   |NO❗️ |
| └ | renounceOwnership | Public ❗️ | 🛑  | onlyOwner |
| └ | transferOwnership | Public ❗️ | 🛑  | onlyOwner |
||||||
| **Token** | Interface |  |||
| └ | transfer | External ❗️ | 🛑  |NO❗️ |
||||||
| **AomenCity** | Implementation | Context, IERC20, Ownable |||
| └ | <Constructor> | Public ❗️ | 🛑  |NO❗️ |
| └ | isBlacklisted | Public ❗️ |   |NO❗️ |
| └ | blacklist | Public ❗️ | 🛑  | onlyBlacklister |
| └ | unBlacklist | Public ❗️ | 🛑  | onlyBlacklister |
| └ | updateBlacklister | Public ❗️ | 🛑  | onlyOwner |
| └ | name | Public ❗️ |   |NO❗️ |
| └ | symbol | Public ❗️ |   |NO❗️ |
| └ | decimals | Public ❗️ |   |NO❗️ |
| └ | totalSupply | Public ❗️ |   |NO❗️ |
| └ | balanceOf | Public ❗️ |   |NO❗️ |
| └ | transfer | Public ❗️ | 🛑  |NO❗️ |
| └ | allowance | Public ❗️ |   |NO❗️ |
| └ | approve | Public ❗️ | 🛑  |NO❗️ |
| └ | transferFrom | Public ❗️ | 🛑  |NO❗️ |
| └ | increaseAllowance | Public ❗️ | 🛑  |NO❗️ |
| └ | decreaseAllowance | Public ❗️ | 🛑  |NO❗️ |
| └ | isExcluded | Public ❗️ |   |NO❗️ |
| └ | reflectionFromToken | Public ❗️ |   |NO❗️ |
| └ | tokenFromReflection | Public ❗️ |   |NO❗️ |
| └ | excludeAccount | External ❗️ | 🛑  | onlyOwner |
| └ | includeAccount | External ❗️ | 🛑  | onlyOwner |
| └ | _approve | Private 🔐 | 🛑  | |
| └ | _transfer | Private 🔐 | 🛑  | |
| └ | _burn | Public ❗️ | 🛑  | onlyOwner |
| └ | collectFee | Private 🔐 | 🛑  | |
| └ | _getReflectionRate | Private 🔐 |   | |
| └ | delegates | External ❗️ |   |NO❗️ |
| └ | delegate | External ❗️ | 🛑  |NO❗️ |
| └ | delegateBySig | External ❗️ | 🛑  |NO❗️ |
| └ | getCurrentVotes | External ❗️ |   |NO❗️ |
| └ | getPriorVotes | External ❗️ |   |NO❗️ |
| └ | _delegate | Internal 🔒 | 🛑  | |
| └ | _moveDelegates | Internal 🔒 | 🛑  | |
| └ | _writeCheckpoint | Internal 🔒 | 🛑  | |
| └ | safe32 | Internal 🔒 |   | |
| └ | getChainId | Internal 🔒 |   | |
| └ | ExcludedFromFee | Public ❗️ | 🛑  | onlyOwner |
| └ | IncludeFromFee | Public ❗️ | 🛑  | onlyOwner |
| └ | setReflectionFee | Public ❗️ | 🛑  | onlyOwner |
| └ | setLiquidityFee | Public ❗️ | 🛑  | onlyOwner |
| └ | setTreasury1Fee | Public ❗️ | 🛑  | onlyOwner |
| └ | setTreasury2Fee | Public ❗️ | 🛑  | onlyOwner |
| └ | setBurnPercent | Public ❗️ | 🛑  | onlyOwner |
| └ | settreasury1Address | Public ❗️ | 🛑  | onlyOwner |
| └ | setLiquidityAddress | Public ❗️ | 🛑  | onlyOwner |
| └ | TurnOffFees | External ❗️ | 🛑  | onlyOwner |
| └ | <Receive Ether> | External ❗️ | 💵 |NO❗️ |

 Legend

|  Symbol  |  Meaning  |
|:--------:|-----------|
|    🛑    | Function can modify state |
|    💵    | Function is payable |
```

Conclusion

The contracts are written systematically. Team found no critical issues. So, it is good to go for production and no need for redeploy the contract.

Since possible test cases can be unlimited and developer level documentation (code flow diagram with function level description) not provided, for such an extensive smart contract protocol, we provide no such guarantee of future outcomes. We have used all the latest static tools and manual observations to cover maximum possible test cases to scan Everything.

Security state of the reviewed contract is “secured”.

✓ No mint function.
✓ No volatile code.
✓ Not many high severity issues were found.

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as of the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against the team on the basis of what it says or doesn’t say, or how team produced it, and it is important for you to conduct your own independent investigations before making any decisions. team go into more detail on this in the below disclaimer below – please make sure to read it in full.

By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and Saferico and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (Saferico s) owe no duty of care towards you or any other person, nor does Saferico make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and Saferico hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, Saferico hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against Saferico, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report. The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.