auditace.tech

AuditAce
FROM INCEPTION TO SUCCESS

# Smart Contract Audit

FOR

# SHIKOKUCEO

DATED : 08 Mar 23'

# AUDIT SUMMARY

**Project name** – SHIKOKUCEO

**Date**: 08 March, 2023

**Scope of Audit-** Audit Ace was consulted to conduct the smart contract audit of the solidity source codes.

**Audit Status: Passed**

## Issues Found

| Status | Critical | High | Medium | Low | Suggestion |
|---|---|---|---|---|---|
| Open | 0 | 0 | 0 | 1 | 0 |
| Acknowledged | 0 | 0 | 0 | 0 | 0 |
| Resolved | 0 | 0 | 0 | 0 | 0 |

# USED TOOLS

## Tools:

**1- Manual Review:**
a line by line code review has been performed by audit ace team.

**2- BSC Test Network:**
all tests were done on BSC Test network, each test has its transaction has attached to it.

**3- Slither : Static Analysis**

**Testnet Link:** all tests were done using this contract, tests are done on BSC Testnet

https://testnet.bscscan.com/token/0xe8472dae5ff7a0db66bf891fa68656de958f87f0

# Token Information

**Token Name** : SHIKOKUCEO

**Token Symbol**: SUCEO

**Decimals:** 9

**Token Supply**: 420,000,000,000,000,000

**Token Address:**
0x828AbCca698a2988Fad35590fb266D5c85732fD5

**Checksum:**
3c5ca3afa9cb0ff79b203c27d40e3d06f08b9bc9

**Owner:**
0xf153B0a6FbB4a091320b3d8a6A7F14a5e4Ef3eB3
**(at time of writing the audit)**

# TOKEN OVERVIEW

**Fees:**

Buy Fees: 10%

Sell Fees: 10%

Transfer Fees: 10%

**Fees Privilige:** None

**Ownership** : Owned

**Minting:** No mint function

**Max Tx Amount/ Max Wallet Amount:** No

**Blacklist:** No

**Other Priviliges**: including and excluding from fees and rewards

# AUDIT METHODOLOGY

The auditing process will follow a routine as special considerations by Auditace:

- Review of the specifications, sources, and instructions provided to Auditace to make sure the contract logic meets the intentions of the client without exposing the user's funds to risk.
- Manual review of the entire codebase by our experts, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
- Specification comparison is the process of checking whether the code does what the specifications, sources, and instructions provided to Auditace describe.
- Test coverage analysis determines whether the test cases are covering the code and how much code isexercised when we run the test cases.
- Symbolic execution is analysing a program to determine what inputs cause each part of a program to execute.
- Reviewing the codebase to improve maintainability, security, and control based on the established industry and academic practices.

# VULNERABILITY CHECKLIST

- [x] Return values of low-level calls
- [x] Private modifier
- [x] Multiple Sends
- [x] Using Suicide
- [x] Gas Limitand Loops
- [x] Address hardcoded
- [x] Exception Disorder
- [x] Using inline assembly
- [x] Divide before multiply
- [x] Missing Zero Address Validation
- [x] Compiler version not fixed
- [x] **Gasless Send**
- [x] Using block.timestamp
- [x] Re-entrancy
- [x] Tautology or contradiction
- [x] Timestamp Dependence
- [x] Revert/require functions
- [x] Use of tx.origin
- [x] Integer overflow/underflow
- [x] Dangerous strict equalities
- [x] Using SHA3
- [x] Using throw

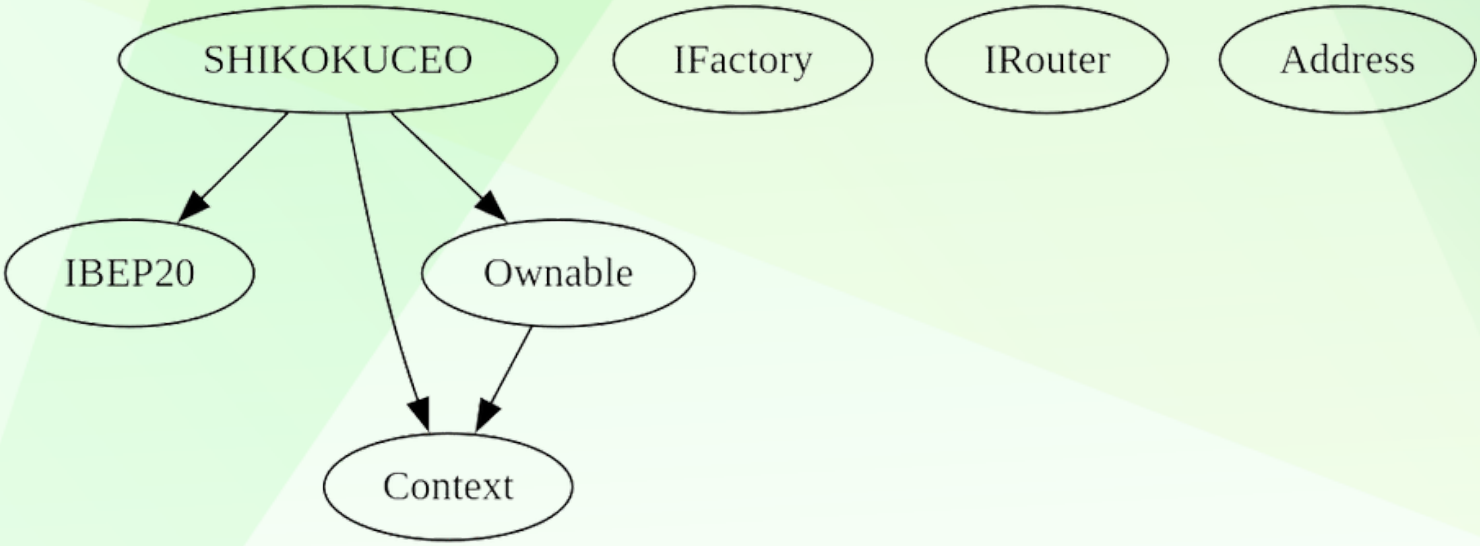
# CLASSIFICATION OF RISK

| Severity | Description |
| --- | --- |
| Critical | These vulnerabilities could be exploited easily and can lead to asset loss, data loss, asset, or data manipulation. They should be fixed right away. |
| High-Risk | A vulnerability that affects the desired outcome when using a contract, or provides the opportunity to use a contract in an unintended way. |
| Medium-Risk | A vulnerability that could affect the desired outcome of executing the contract in a specific scenario. |
| Low-Risk | A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective. |
| Gas Optimization /Suggestion | A vulnerability that has an informational character but is not affecting any of the code. |

# Findings

| Severity | Found |
| --- | --- |
| Critical | 0 |
| High-Risk | 0 |
| Medium-Risk | 0 |
| Low-Risk | 1 |
| Gas Optimization / Suggestions | 0 |

# INHERITANCE TREE



# POINTS TO NOTE

- Owner is not able to change fees (10% for buy/sell/transfers static)
- Owner is not able to set max buy/sell/transfer/hold amount
- Owner is not able to blacklist an arbitrary wallet
- Owner is not able to disable trades
- Owner is not able to mint new tokens

# CONTRACT ASSESMENT

| Contract | Type | Bases | | |
|:----------:|:-------------------:|:----------------:|:----------------:|:---------------:|
| └ | **Function Name** | **Visibility** | **Mutability** | **Modifiers** |
| | | | | |
| **IBEP20** | Interface | | | |
| └ | totalSupply | External ❗ | | NO❗ |
| └ | balanceOf | External ❗ | | NO❗ |
| └ | transfer | External ❗ | 🛑 | NO❗ |
| └ | allowance | External ❗ | | NO❗ |
| └ | approve | External ❗ | 🛑 | NO❗ |
| └ | transferFrom | External ❗ | 🛑 | NO❗ |
| | | | | |
| **Context** | Implementation | | | |
| └ | _msgSender | Internal 🔒 | | |
| └ | _msgData | Internal 🔒 | | |
| | | | | |
| **Ownable** | Implementation | Context | | |
| └ | <Constructor> | Public ❗ | 🛑 | NO❗ |
| └ | owner | Public ❗ | | NO❗ |
| └ | renounceOwnership | Public ❗ | 🛑 | onlyOwner |
| └ | _setOwner | Private 🔐 | 🛑 | |
| | | | | |
| **IFactory** | Interface | | | |
| └ | createPair | External ❗ | 🛑 | NO❗ |
| | | | | |
| **IRouter** | Interface | | | |
| └ | factory | External ❗ | | NO❗ |
| └ | WETH | External ❗ | | NO❗ |
| └ | addLiquidityETH | External ❗ | 💵 | NO❗ |
| └ | swapExactTokensForETHSupportingFeeOnTransferTokens | External ❗ | 🛑 | NO❗ |
| | | | | |
| **Address** | Library | | | |
| └ | sendValue | Internal 🔒 | 🛑 | |
| | | | | |
| **SHIKOKUCEO** | Implementation | Context, IBEP20, Ownable | | |
| └ | <Constructor> | Public ❗ | 🛑 | NO❗ |
| └ | name | Public ❗ | | NO❗ |
| └ | symbol | Public ❗ | | NO❗ |
| └ | decimals | Public ❗ | | NO❗ |
| └ | totalSupply | Public ❗ | | NO❗ |
| └ | balanceOf | Public ❗ | | NO❗ |
| └ | allowance | Public ❗ | | NO❗ |

# CONTRACT ASSESMENT

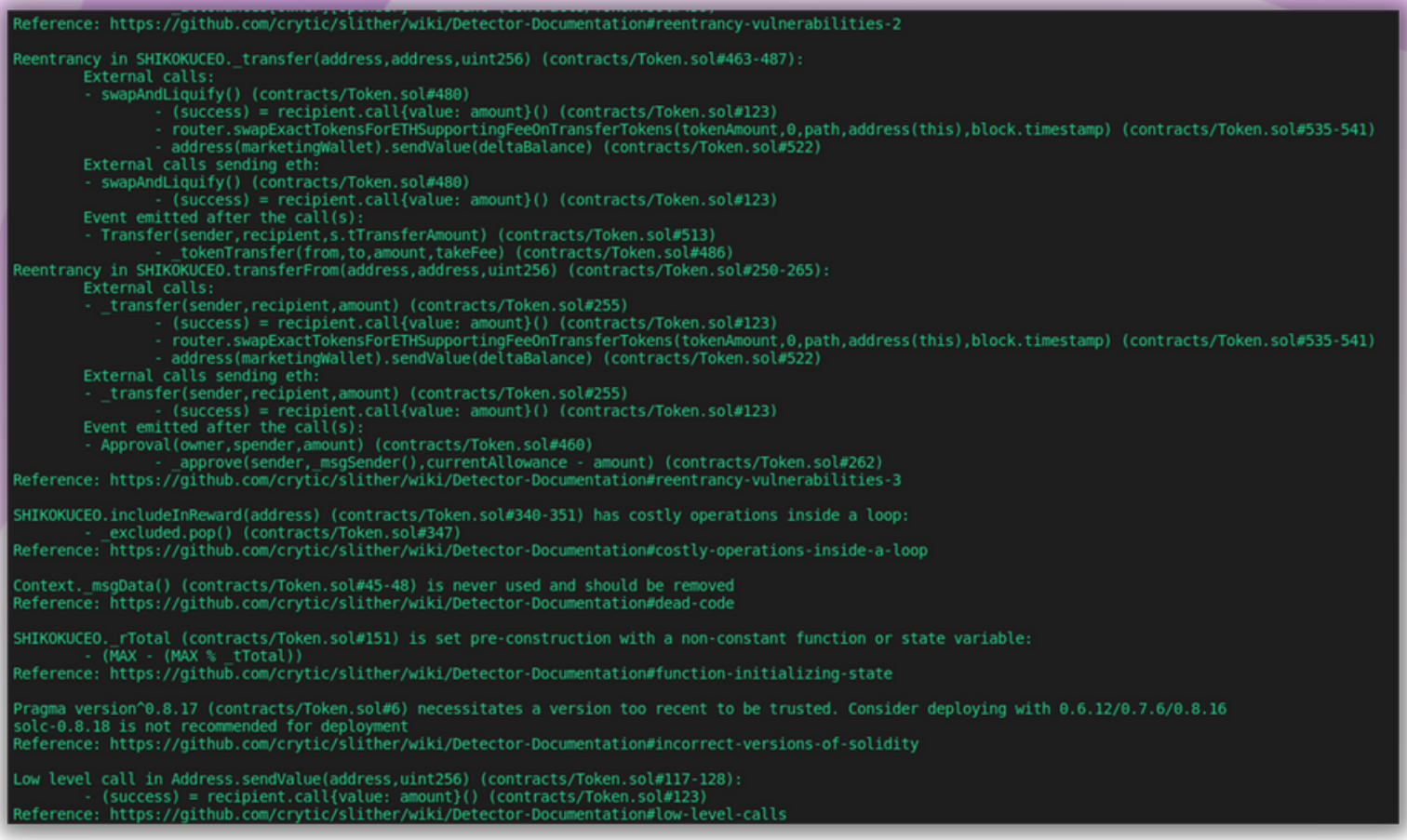
| └ | approve | Public ❗ | 🛑 | NO❗ |
| --- | --- | --- | --- | --- |
| └ | transferFrom | Public ❗ | 🛑 | NO❗ |
| └ | increaseAllowance | Public ❗ | 🛑 | NO❗ |
| └ | decreaseAllowance | Public ❗ | 🛑 | NO❗ |
| └ | transfer | Public ❗ | 🛑 | NO❗ |
| └ | isExcludedFromReward | Public ❗ | | NO❗ |
| └ | reflectionFromToken | Public ❗ | | NO❗ |
| └ | tokenFromReflection | Public ❗ | | NO❗ |
| └ | excludeFromReward | Public ❗ | 🛑 | onlyOwner |
| └ | includeInReward | External ❗ | 🛑 | onlyOwner |
| └ | excludeFromFee | Public ❗ | 🛑 | onlyOwner |
| └ | includeInFee | Public ❗ | 🛑 | onlyOwner |
| └ | isExcludedFromFee | Public ❗ | | NO❗ |
| └ | _reflectRfi | Private 🔐 | 🛑 | |
| └ | _takeMarketing | Private 🔐 | 🛑 | |
| └ | _getValues | Private 🔐 | | |
| └ | _getTValues | Private 🔐 | | |
| └ | _getRValues | Private 🔐 | | |
| └ | _getRate | Private 🔐 | | |
| └ | _getCurrentSupply | Private 🔐 | | |
| └ | _approve | Private 🔐 | 🛑 | |
| └ | _transfer | Private 🔐 | 🛑 | |
| └ | _tokenTransfer | Private 🔐 | 🛑 | |
| └ | swapAndLiquify | Private 🔐 | 🛑 | lockTheSwap |
| └ | swapTokensForBNB | Private 🔐 | 🛑 | |
| └ | bulkExcludeFee | External ❗ | 🛑 | onlyOwner |
| └ | <Receive Ether> | External ❗ | 💵 | NO❗ |

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

# TOKEN DISTRIBUTION

It should be noted that the owner currently holds 100% of the total supply. However, information about the distribution of these tokens is not available, and it is recommended that investors exercise caution when considering this aspect.

# STATIC ANALYSIS



```
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

Reentrancy in SHIKOKUCEO._transfer(address,address,uint256) (contracts/Token.sol#463-487):
        External calls:
        - swapAndLiquify() (contracts/Token.sol#480)
                - (success) = recipient.call{value: amount}() (contracts/Token.sol#123)
                - router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (contracts/Token.sol#535-541)
                - address(marketingWallet).sendValue(deltaBalance) (contracts/Token.sol#522)
        External calls sending eth:
        - swapAndLiquify() (contracts/Token.sol#480)
                - (success) = recipient.call{value: amount}() (contracts/Token.sol#123)
        Event emitted after the call(s):
        - Transfer(sender,recipient,s.tTransferAmount) (contracts/Token.sol#513)
                - _tokenTransfer(from,to,amount,takeFee) (contracts/Token.sol#486)
Reentrancy in SHIKOKUCEO.transferFrom(address,address,uint256) (contracts/Token.sol#250-265):
        External calls:
        - _transfer(sender,recipient,amount) (contracts/Token.sol#255)
                - (success) = recipient.call{value: amount}() (contracts/Token.sol#123)
                - router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (contracts/Token.sol#535-541)
                - address(marketingWallet).sendValue(deltaBalance) (contracts/Token.sol#522)
        External calls sending eth:
        - _transfer(sender,recipient,amount) (contracts/Token.sol#255)
                - (success) = recipient.call{value: amount}() (contracts/Token.sol#123)
        Event emitted after the call(s):
        - Approval(owner,spender,amount) (contracts/Token.sol#460)
                - _approve(sender,_msgSender(),currentAllowance - amount) (contracts/Token.sol#262)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

SHIKOKUCEO.includeInReward(address) (contracts/Token.sol#340-351) has costly operations inside a loop:
        - _excluded.pop() (contracts/Token.sol#347)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#costly-operations-inside-a-loop

Context._msgData() (contracts/Token.sol#45-48) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

SHIKOKUCEO._rTotal (contracts/Token.sol#151) is set pre-construction with a non-constant function or state variable:
        - (MAX - (MAX % _tTotal))
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#function-initializing-state

Pragma version^0.8.17 (contracts/Token.sol#6) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.16
solc-0.8.18 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (contracts/Token.sol#117-128):
        - (success) = recipient.call{value: amount}() (contracts/Token.sol#123)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
```

**Result => A static analysis of contract's source code has been performed using slither,**

**No major issues were found in the output**

# FUNCTIONAL TESTING

**Router (PCS V2):**
0xD99D1c33F9fC3444f8101754aBC46c52416550D1

All the functionalities have been tested, no issues were found

**1- Adding liquidity (passed):**
https://testnet.bscscan.com/tx/0xd2183cb1416d7478c523ca038f45762e79ff4cad266f0e06135e884b8300cc13

**2- Buying when excluded (0% tax) (passed):**
https://testnet.bscscan.com/tx/0x972b9cf1edc6debee2b0b12fb2da50e766e425f76da9461cae09241a52233fed

**3- Selling when excluded (0% tax) (passed):**
https://testnet.bscscan.com/tx/0x9b751a1e09eb44de0a93350354d2c77259975a531ea638b07fded59296d6f7ec

**4- Transferring when excluded (0% tax) (passed):**
https://testnet.bscscan.com/tx/0xd0814ae62338bcb6c0f99f9168748f154839a6a6c4a5a934dc1ccde5c76ce077

# FUNCTIONAL TESTING

**5- Buying when not excluded from fees (10% tax) (passed):**
https://testnet.bscscan.com/tx/0xd7adf8923c0fce977e351d81f80e9d53b0dad7e9d5dda3f0bcd0016895d453ce

**6- Selling when not excluded from fees (10% tax) (passed):**
https://testnet.bscscan.com/tx/0xe122550cceb5894e3bac024682bfbb9ec0f46007a2bde1404e39c1f5c6fa0fbd

**7- Transferring when not excluded from fees (10% tax) (passed):**
https://testnet.bscscan.com/tx/0x41996c1e9d4f51d2334366561355 6ef631a6da35270148b9b73f1b637ac53fc3

**8-Internal swap (passed):**
**marketing wallet received ETH**
https://testnet.bscscan.com/address/0xd8999a98dc6e17dc7713ecc2c6b6f55ed7f965a7

# MANUAL TESTING

## Low Risk Issue

---

IIssue: **no transferOwnership Function**
Type : Logical
Function: ---
Line: 42-69
Severity: Low

**Overview:** the contract does not have a **transferOwnership** function, which means that the ownership of the contract cannot be transferred to another address. This can be a problem if the original owner loses control of their private keys or if they are no longer able to manage the contract for any reason

### Recommendations

To address the issue identified in this audit, we recommend the following:

1.Implement a **TransferOwnership** Function The contract should be updated to include a transferOwnership function that allows the current owner to transfer ownership of the contract to another address. This function should include proper access control to prevent unauthorized transfers of ownership.

# DISCLAIMER

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment. Team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed. The Auditace team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Auditace receive a payment to manipulate those results or change the awarding badge that we will be adding in our website. Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token. The Auditace team disclaims any liability for the resulting losses.

# ABOUT AUDITACE

We specializes in providing thorough and reliable audits for Web3 projects. With a team of experienced professionals, we use cutting-edge technology and rigorous methodologies to evaluate the security and integrity of blockchain systems. We are committed to helping our clients ensure the safety and transparency of their digital assets and transactions.


**https://auditace.tech/**

**https://t.me/Audit_Ace**

**https://twitter.com/auditace_**

**https://github.com/Audit-Ace**