



# Smart Contract Audit

FOR

## Shiba Saga 2.0

DATED : 24 August 2023'



# AUDIT SUMMARY

---

**Project name –** Shiba Saga 2.0

**Date:** 24 August 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	0	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 conducted on the BSC Test network, and each test has a corresponding transaction attached to it. These tests can be found in the "Functional Tests" section of the report.

### 3- Slither :

The code has undergone static analysis using Slither.

### Testnet version:

The tests were performed using the contract deployed on the BSC Testnet, which can be found at the following address:

<https://testnet.bscscan.com/token/0xbB4F97e15ba9F818eD9ac09650F01b2E4337dE03>

---



# Token Information

---

**Token Address :**

**0x6F81c9bf9eCe26d75E8A12d93aAf930988339659**

**Name: Shiba Saga 2.0**

**Symbol: SHIA20**

**Decimals: 9**

**Netowrk: BSC**

**Token Type: BEP20**

**Owner: 0xCD06Ff23cc7819822734cAbcFcC18431bC2229E9**  
**(at time of writing the audit)**

**Deployer:**

**0xCD06Ff23cc7819822734cAbcFcC18431bC2229E9**

**Token Supply: 10,000,000,000**

**Checksum:**

**fa17c9f5736affaf74cb2b9343cf49e4888f335b**

**Testnet version:**

**<https://testnet.bscscan.com/token/0xbB4F97e15ba9F818eD9ac09650F01b2E4337dE03>**

---



# TOKEN OVERVIEW

---

**buy fee: 0-6%**

**Sell fee: 0-6%**

**transfer fee: 0%**

---

**Fee Privilege: Owner**

---

**Ownership: Owned**

---

**Minting: None**

---

**Max Tx: No**

---

**Blacklist: No**

---

**Other Privileges:**

- Initial distribution of the tokens
  - Modifying fees
- 
-



# 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 is exercised 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

---

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

0

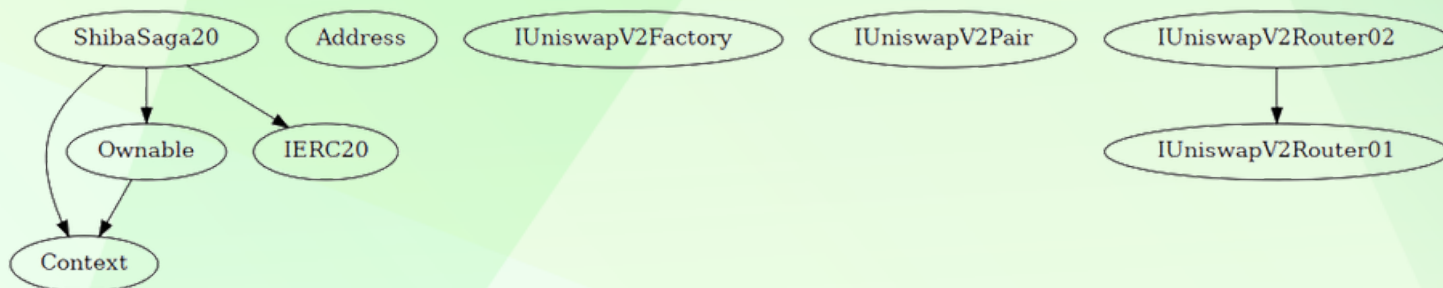
### ◆ Gas Optimization / Suggestions

0



# INHERITANCE TREE

---



# POINTS TO NOTE

---

- Owner is able to change fees in range 0-6% for buy and sells (0% transfer tax)
  - Owner is not able to blacklist an arbitrary address.
  - Owner is not able to disable trades
  - Owner is not able to limit buy/sell/transfer/wallet amounts
  - Owner is not able to mint new tokens
-



# STATIC ANALYSIS

```
Variable ShibaSage20._getValues(uint256).rTransferAmount (contracts/Token.sol#868) is too similar to ShibaSage20._transferFromExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1200)
Variable ShibaSage20._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1221) is too similar to ShibaSage20._transferToExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1177)
Variable ShibaSage20._getRValues(uint256,uint256,uint256,uint256,uint256).rTransferAmount (contracts/Token.sol#907) is too similar to ShibaSage20._transferFromExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1200)
Variable ShibaSage20._getValues(uint256).rTransferAmount (contracts/Token.sol#868) is too similar to ShibaSage20._getTValues(uint256).tTransferAmount (contracts/Token.sol#892)
Variable ShibaSage20._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1221) is too similar to ShibaSage20._getValues(uint256).tTransferAmount (contracts/Token.sol#863)
Variable ShibaSage20._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1175) is too similar to ShibaSage20._transferFromExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1200)
Variable ShibaSage20._getValues(uint256).rTransferAmount (contracts/Token.sol#868) is too similar to ShibaSage20._transferToExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1177)
Variable ShibaSage20._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1175) is too similar to ShibaSage20._getTValues(uint256).tTransferAmount (contracts/Token.sol#892)
Variable ShibaSage20._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1221) is too similar to ShibaSage20._transferStandard(address,address,uint256).tTransferAmount (contracts/Token.sol#1155)
Variable ShibaSage20._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1221) is too similar to ShibaSage20._transferBothExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1223)
Variable ShibaSage20.reflectionFromToken(uint256,bool).rTransferAmount (contracts/Token.sol#765) is too similar to ShibaSage20._transferFromExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1200)
Variable ShibaSage20._getValues(uint256).rTransferAmount (contracts/Token.sol#868) is too similar to ShibaSage20._getValues(uint256).tTransferAmount (contracts/Token.sol#863)
Variable ShibaSage20._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1175) is too similar to ShibaSage20._transferToExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1177)
Variable ShibaSage20._getValues(uint256).rTransferAmount (contracts/Token.sol#868) is too similar to ShibaSage20._transferStandard(address,address,uint256).tTransferAmount (contracts/Token.sol#1155)
Variable ShibaSage20._getValues(uint256).rTransferAmount (contracts/Token.sol#868) is too similar to ShibaSage20._transferBothExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1223)
Variable ShibaSage20._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1175) is too similar to ShibaSage20._getValues(uint256).tTransferAmount (contracts/Token.sol#863)
Variable ShibaSage20._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1221) is too similar to ShibaSage20._transferFromExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1200)
Variable ShibaSage20._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1221) is too similar to ShibaSage20._getTValues(uint256).tTransferAmount (contracts/Token.sol#892)
Variable ShibaSage20._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1175) is too similar to ShibaSage20._transferStandard(address,address,uint256).tTransferAmount (contracts/Token.sol#1155)
Variable ShibaSage20._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1175) is too similar to ShibaSage20._transferBothExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1223)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar
INFO:Detectors:
Loop condition `0` {
  ui' (contracts/Token.sol#919) should use cached array length instead of referencing 'length' member of the storage array.
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#cache-array-length
INFO:Detectors:
ShibaSage20.DEAD (contracts/Token.sol#568) should be constant
ShibaSage20._decimals (contracts/Token.sol#541) should be constant
ShibaSage20._name (contracts/Token.sol#539) should be constant
ShibaSage20._symbol (contracts/Token.sol#540) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
INFO:Detectors:
ShibaSage20.DEV (contracts/Token.sol#569) should be immutable
ShibaSage20._tTotal (contracts/Token.sol#544) should be immutable
ShibaSage20.mk (contracts/Token.sol#566) should be immutable
ShibaSage20.totalBuyFees (contracts/Token.sol#561-562) should be immutable
ShibaSage20.totalSellFees (contracts/Token.sol#563-564) should be immutable
ShibaSage20.uniswapV2Pair (contracts/Token.sol#572) should be immutable
ShibaSage20.uniswapV2Router (contracts/Token.sol#571) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
INFO:Slither:/contracts/Token.sol analyzed (9 contracts with 88 detectors), 107 result(s) found
```

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

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



# CONTRACT ASSESMENT

```
|Contract |      Type      |Bases |      |      |
|:-----:|:-----:|:-----:|:-----:|:-----:|
|  ⌞      | **Function Name** | **Visibility** | **Mutability** | **Modifiers** |
|||||
| **Context** | Implementation | |||
|  ⌞ | _msgSender | Internal 🔒 | | |
|  ⌞ | _msgData | Internal 🔒 | | |
|||||
| **Ownable** | Implementation | Context |||
|  ⌞ | <Constructor> | Public ! | ● | NO ! |
|  ⌞ | owner | Public ! | | NO ! |
|  ⌞ | renounceOwnership | Public ! | ● | onlyOwner |
|  ⌞ | transferOwnership | Public ! | ● | onlyOwner |
|||||
| **IERC20** | Interface | |||
|  ⌞ | totalSupply | External ! | | NO ! |
|  ⌞ | balanceOf | External ! | | NO ! |
|  ⌞ | transfer | External ! | ● | NO ! |
|  ⌞ | allowance | External ! | | NO ! |
|  ⌞ | approve | External ! | ● | NO ! |
|  ⌞ | transferFrom | External ! | ● | NO ! |
|||||
| **Address** | Library | |||
|  ⌞ | isContract | Internal 🔒 | | |
|  ⌞ | sendValue | Internal 🔒 | ● | |
|  ⌞ | functionCall | Internal 🔒 | ● | |
|  ⌞ | functionCall | Internal 🔒 | ● | |
|  ⌞ | functionCallWithValue | Internal 🔒 | ● | |
|  ⌞ | functionCallWithValue | Internal 🔒 | ● | |
|  ⌞ | _functionCallWithValue | Private 🔒🔑 | ● | |
|||||
```



# CONTRACT ASSESMENT

---

```
| **IUniswapV2Factory** | Interface | |||
|  ↳ | feeTo | External ! | | NO ! |
|  ↳ | feeToSetter | External ! | | NO ! |
|  ↳ | getPair | External ! | | NO ! |
|  ↳ | allPairs | External ! | | NO ! |
|  ↳ | allPairsLength | External ! | | NO ! |
|  ↳ | createPair | External ! | ● | NO ! |
|  ↳ | setFeeTo | External ! | ● | NO ! |
|  ↳ | setFeeToSetter | External ! | ● | NO ! |
|||||
| **IUniswapV2Pair** | Interface | |||
|  ↳ | name | External ! | | NO ! |
|  ↳ | symbol | External ! | | NO ! |
|  ↳ | decimals | External ! | | NO ! |
|  ↳ | totalSupply | External ! | | NO ! |
|  ↳ | balanceOf | External ! | | NO ! |
|  ↳ | allowance | External ! | | NO ! |
|  ↳ | approve | External ! | ● | NO ! |
|  ↳ | transfer | External ! | ● | NO ! |
|  ↳ | transferFrom | External ! | ● | NO ! |
|  ↳ | DOMAIN_SEPARATOR | External ! | | NO ! |
|  ↳ | PERMIT_TYPEHASH | External ! | | NO ! |
|  ↳ | nonces | External ! | | NO ! |
|  ↳ | permit | External ! | ● | NO ! |
|  ↳ | MINIMUM_LIQUIDITY | External ! | | NO ! |
|  ↳ | factory | External ! | | NO ! |
|  ↳ | token0 | External ! | | NO ! |
|  ↳ | token1 | External ! | | NO ! |
|  ↳ | getReserves | External ! | | NO ! |
|  ↳ | price0CumulativeLast | External ! | | NO ! |
|  ↳ | price1CumulativeLast | External ! | | NO ! |
|  ↳ | kLast | External ! | | NO ! |
|  ↳ | burn | External ! | ● | NO ! |
|  ↳ | swap | External ! | ● | NO ! |
|  ↳ | skim | External ! | ● | NO ! |
|  ↳ | sync | External ! | ● | NO ! |
|  ↳ | initialize | External ! | ● | NO ! |
```

---



# CONTRACT ASSESMENT

---

|||||

```
| **IUniswapV2Router01** | Interface | |||  
|  └ | factory | External ! | |NO ! |  
|  └ | WETH | External ! | |NO ! |  
|  └ | addLiquidity | External ! | ● |NO ! |  
|  └ | addLiquidityETH | External ! | 📦 |NO ! |  
|  └ | removeLiquidity | External ! | ● |NO ! |  
|  └ | removeLiquidityETH | External ! | ● |NO ! |  
|  └ | removeLiquidityWithPermit | External ! | ● |NO ! |
```

|||||

```
| **IUniswapV2Router01** | Interface | |||  
|  └ | factory | External ! | |NO ! |  
|  └ | WETH | External ! | |NO ! |  
|  └ | addLiquidity | External ! | ● |NO ! |  
|  └ | addLiquidityETH | External ! | 📦 |NO ! |  
|  └ | removeLiquidity | External ! | ● |NO ! |  
|  └ | removeLiquidityETH | External ! | ● |NO ! |  
|  └ | removeLiquidityWithPermit | External ! | ● |NO ! |
```

|||||

```
| **ShibaSaga20** | Implementation | Context, IERC20, Ownable |||  
|  └ | <Constructor> | Public ! | ● |NO ! |  
|  └ | 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 ! |  
|  └ | isExcludedFromReward | Public ! | |NO ! |  
|  └ | totalReflectionDistributed | Public ! | |NO ! |  
|  └ | deliver | Public ! | ● |NO ! |
```

---



# CONTRACT ASSESMENT

---

	reflectionFromToken	Public !		NO !
	tokenFromReflection	Public !		NO !
	excludeFromReward	Public !		onlyOwner
	includeInReward	External !		onlyOwner
	<Receive Ether>	External !		NO !
	clearStuckTokens	External !		NO !
	updateFeeBuy	Public !		onlyOwner
	updateFeeSell	Public !		onlyOwner
	\_reflectFee	Private		
	\_getValues	Private		
	\_getTValues	Private		
	\_getRValues	Private		
	\_getRate	Private		
	\_getCurrentSupply	Private		
	\_takeLiquidity	Private		
	\_takeMarketing	Private		
	calculateTaxFee	Private		
	calculateLiquidityFee	Private		
	calculateMarketingFee	Private		
	removeAllFee	Private		
	setBuyFee	Private		
	setSellFee	Private		
	isExcludedFromFee	Public !		NO !
	\_approve	Private		
	\_transfer	Private		
	swapAndLiquify	Private		
	swapAndSendMarketing	Private		
	setSwapTokensAtAmount	External !		onlyOwner
	setSwapEnabled	External !		onlyOwner
	\_tokenTransfer	Private		
	\_transferStandard	Private		
	\_transferToExcluded	Private		
	\_transferFromExcluded	Private		
	\_transferBothExcluded	Private		
	excludeFromFees	External !		onlyOwner
	isContract	Internal		

---




# CONTRACT ASSESMENT


---

## ### Legend

|Symbol | Meaning|

|:------:|------|

|  | Function can modify state |

|  | Function is payable |





# FUNCTIONAL TESTING

---

1- Adding liquidity (**passed**):

<https://testnet.bscscan.com/tx/0xdea65f380a164dcc29d8237ad26b1ad0461a534b82d5e935127fc736e68e9b6d>

2- Buying when excluded from fees (0% tax) (**passed**):

<https://testnet.bscscan.com/tx/0x458a092fb2ad0045eb347fece98ac37a41493e53243de4fe3ea5134d027c7839>

3- Selling when excluded from fees (0% tax) (**passed**):

<https://testnet.bscscan.com/tx/0xf60edcc47af12eb14afdf4e6ee18bdaa7ee45cf86c89826330088071edddf5d7>

4- Transferring when excluded from fees (0% tax) (**passed**):

<https://testnet.bscscan.com/tx/0xa51000d0d7d43d72796673af41925ee2505c35a4f89288127e8802e7b0ba80ef>

5- Buying from a regular wallet (0-6% tax) (**passed**):

<https://testnet.bscscan.com/tx/0x3b27438a287e3d25fe051f77b59f957f3a3ed01d054b3a4daa4cbccb8a70342e>

6- Selling from a regular wallet (0-6% tax) (**passed**):

<https://testnet.bscscan.com/tx/0xe91b6f9d1a4a44beeee6d5b13f98e5c8dfa6c1bd0ed2a13152bab78e49bd1c1b>

7- Transferring from a regular wallet (0% tax) (**passed**):

<https://testnet.bscscan.com/tx/0x8b8110e1342a74c72c9bc8d79bcfd6e2130d3191b62ba2cb527dd6887da56ba2>

7- Internal swap (marketing and development wallets received BNB)(**passed**):

<https://testnet.bscscan.com/tx/0xe91b6f9d1a4a44beeee6d5b13f98e5c8dfa6c1bd0ed2a13152bab78e49bd1c1b>

---



# 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 specialize 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://t.me/Audit_Ace)**



**[https://twitter.com/auditace\\_](https://twitter.com/auditace_)**



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

---