



# SAFUAUDIT

SMART CONTRACT AUDITING

# SHIBABUSD

## SMART CONTRACT AUDIT



March 12, 2022

# INTRODUCTION

---

<b>Client</b>	ShibaBUSD (SBUSD)
<b>Language</b>	Solidity
<b>Contract address</b>	0x53187D8A21260766e3Fbf99ACD69Ce657309f15e
<b>Decimals</b>	18
<b>Supply</b>	100,000,000
<b>Platform</b>	Binance Smart Chain
<b>Compiler</b>	v0.8.4+commit.c7e474f2
<b>Optimization</b>	Yes, with 200 runs
<b>Website</b>	<a href="http://shibabusd.com/">http://shibabusd.com/</a>
<b>Telegram</b>	<a href="https://t.me/ShibaBUSDOfficial">https://t.me/ShibaBUSDOfficial</a>
<b>Twitter</b>	<a href="https://twitter.com/ShibaBusd">https://twitter.com/ShibaBusd</a>

## Description

ShibaBUSD is the step forward in yield-generating contracts on the Binance Smart Chain (BSC); you receive rewards distributed in BUSD-Pegged rather than token reflections and the contract employs a static reward. A Token that gives 18% BUSD Rewards.

# TABLE OF CONTENTS

## 01 INTRODUCTION

---

Introduction	02
Approach	04
Risk classification	05

## 02 ABSTRACT

---

Abstract	06
----------	----

## 03 VULNERABILITIES TEST

---

Vulnerabilities Test	07
----------------------	----

## 04 MANUAL ANALYSIS

---

Manual analysis	09
Contract Inspection	10
Inheritance Tree	17
Important Snippets	18
Good Practices	19

## 05 WEBSITE

---

Website Audit	20
---------------	----

## 06 CONCLUSIONS

---

Disclaimer	21
Audit Results	22
SafuScore	23
Summary	24

# Approach

---



## Audit Details

Our comprehensive audit report provides a full overview of the audited system's architecture, smart contract codebase, and details on any vulnerabilities found within the system.

---



## Audit Goals

The audit goal is to ensure that the project is built to protect investors and users, preventing potentially catastrophic vulnerabilities after launch, that lead to scams and rugpulls.

---



## Code Quality

Our analysis includes both automatic tests and manual code analysis for the following aspects:

- Exploits
  - Back-doors
  - Vulnerability
  - Accuracy
  - Readability
- 



## Tools

- Remix IDE
- MythX, Myhrlil
- SWC Registry
- Open Zeppelin Code Analyzer
- Solidity Code Complier

# RISK CLASSIFICATION

---

## CRITICAL

---

Issues on this level are critical to the smart contract's performance/functionality and should be fixed before moving to a live environment.

## MEDIUM

---

Issues on this level could potentially bring problems and should eventually be fixed.

## MINOR

---

Issues on this level are minor details and warning that can remain unfixed but would be better fixed at some point in the future

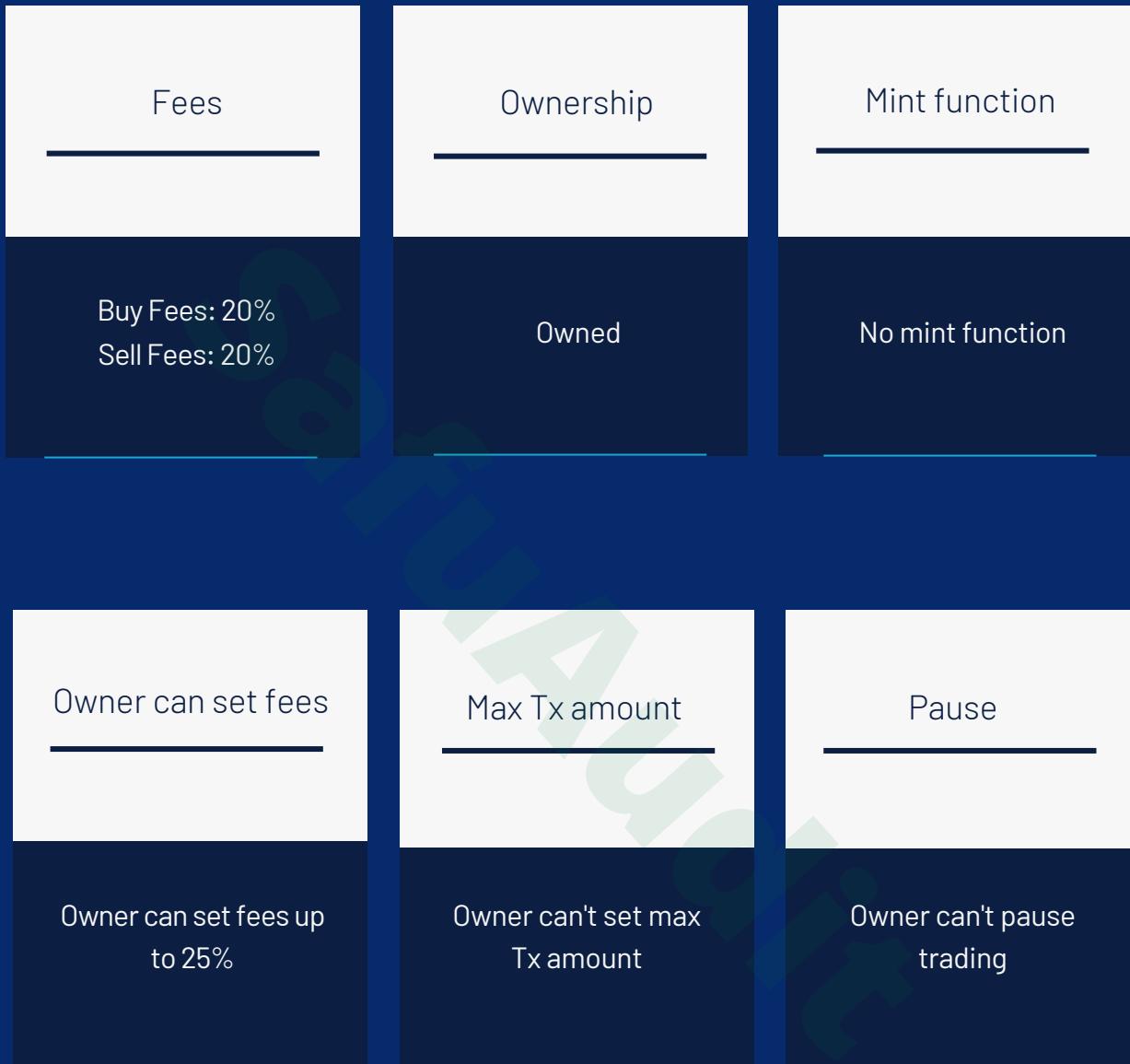
## INFORMATIONAL

---

Information level is to offer suggestions for improvement of efficacy or security for features with a risk free factor.

# ABSTRACT

---



# Vulnerabilities Test

SWC ID	Description	
<b>SWC-100</b>	Function Default Visibility	<b>Passed</b>
<b>SWC-101</b>	Integer Overflow and Underflow	<b>Passed</b>
<b>SWC-102</b>	Outdated Compiler Version	<b>Passed</b>
<b>SWC-103</b>	FloatingPragma	<b>Passed</b>
<b>SWC-104</b>	Unchecked Call Return Value	<b>Passed</b>
<b>SWC-105</b>	Unprotected Ether Withdrawal	<b>Passed</b>
<b>SWC-106</b>	Unprotected SELF-DESTRUCT Instruction	<b>Passed</b>
<b>SWC-107</b>	Re-entrancy	<b>Passed</b>
<b>SWC-108</b>	State Variable Default Visibility	<b>Passed</b>
<b>SWC-109</b>	Uninitialized Storage Pointer	<b>Passed</b>
<b>SWC-110</b>	Assert Violation	<b>Passed</b>
<b>SWC-111</b>	Use of Deprecated Solidity Functions	<b>Passed</b>
<b>SWC-112</b>	Delegate Call to Untrusted Callee	<b>Passed</b>
<b>SWC-113</b>	DoS with Failed Call	<b>Passed</b>
<b>SWC-114</b>	Transaction Order Dependence	<b>Passed</b>
<b>SWC-115</b>	Authorization through tx.origin	<b>Minor</b>

<b>SWC-116</b>	Block values as a proxy for time	<b>Passed</b>
<b>SWC-117</b>	Signature Malleability	<b>Passed</b>
<b>SWC-118</b>	Incorrect Constructor Name	<b>Passed</b>
<b>SWC-119</b>	Shadowing State Variables	<b>Passed</b>
<b>SWC-120</b>	Weak Sources of Randomness from Chain Attributes	<b>Passed</b>
<b>SWC-121</b>	Missing Protection against Signature Replay Attacks	<b>Passed</b>
<b>SWC-122</b>	Lack of Proper Signature Verification	<b>Passed</b>
<b>SWC-123</b>	Requirement Violation	<b>Passed</b>
<b>SWC-124</b>	Write to Arbitrary Storage Location	<b>Passed</b>
<b>SWC-125</b>	Incorrect Inheritance Order	<b>Passed</b>
<b>SWC-126</b>	Insufficient Gas Griefing	<b>Passed</b>
<b>SWC-127</b>	Arbitrary Jump with Function Type Variable	<b>Passed</b>
<b>SWC-128</b>	DoS With Block Gas Limit	<b>Passed</b>
<b>SWC-129</b>	Typographical Error	<b>Passed</b>
<b>SWC-130</b>	Right-To-Left-Override control character (U+202E)	<b>Passed</b>
<b>SWC-131</b>	Presence of unused variables	<b>Passed</b>
<b>SWC-132</b>	Unexpected Ether balance	<b>Passed</b>
<b>SWC-133</b>	Hash Collisions With Multiple Variable Length Arguments	<b>Passed</b>
<b>SWC-134</b>	Message call with the hardcoded gas amount	<b>Passed</b>
<b>SWC-135</b>	Code With No Effects (Irrelevant/Dead Code)	<b>Passed</b>
<b>SWC-136</b>	Unencrypted Private Data On-Chain	<b>Passed</b>

# MANUAL ANALYSIS

The contract is verified to check if functions do and work as they should and malicious code is not inserted.

	Tested	Result
<b>Transfer</b>	Yes	<b>Passed</b>
<b>Total Supply</b>	Yes	<b>Passed</b>
<b>Buy Back</b>	Yes	<b>N/A</b>
<b>Burn</b>	Yes	<b>N/A</b>
<b>Mint</b>	Yes	<b>N/A</b>
<b>Rebase</b>	Yes	<b>N/A</b>
<b>Pause</b>	Yes	<b>N/A</b>
<b>Blacklist</b>	Yes	<b>N/A</b>
<b>Lock</b>	Yes	<b>N/A</b>
<b>Max Transaction</b>	Yes	<b>N/A</b>
<b>Transfer Ownership</b>	Yes	<b>Passed</b>
<b>Renounce Ownership</b>	Yes	<b>Passed</b>

MANUAL AUDIT

# CONTRACT INSPECTION



**IERC20**   Interface
L   totalSupply   External     NO
L   balanceOf   External     NO
L   transfer   External     <span style="color:red;">●</span>   NO
L   allowance   External     NO
L   approve   External     <span style="color:red;">●</span>   NO
L   transferFrom   External     <span style="color:red;">●</span>   NO
**IERC20Metadata**   Interface   IERC20
L   name   External     NO
L   symbol   External     NO
L   decimals   External     NO
**Context**   Implementation
L   _msgSender   Internal   <span style="color:blue;">🔒</span>
L   _msgData   Internal   <span style="color:blue;">🔒</span>
**ERC20**   Implementation   Context, IERC20, IERC20Metadata
L   <Constructor>   Public     <span style="color:red;">●</span>   NO
L   name   Public     NO
L   symbol   Public     NO
L   decimals   Public     NO
L   totalSupply   Public     NO
L   balanceOf   Public     NO
L   transfer   Public     <span style="color:red;">●</span>   NO
L   allowance   Public     NO
L   approve   Public     <span style="color:red;">●</span>   NO
L   transferFrom   Public     <span style="color:red;">●</span>   NO
L   increaseAllowance   Public     <span style="color:red;">●</span>   NO
L   decreaseAllowance   Public     <span style="color:red;">●</span>   NO
L   _transfer   Internal   <span style="color:blue;">🔒</span>     <span style="color:red;">●</span>
L   _mint   Internal   <span style="color:blue;">🔒</span>     <span style="color:red;">●</span>
L   _burn   Internal   <span style="color:blue;">🔒</span>     <span style="color:red;">●</span>
L   _approve   Internal   <span style="color:blue;">🔒</span>     <span style="color:red;">●</span>
L   _beforeTokenTransfer   Internal   <span style="color:blue;">🔒</span>     <span style="color:red;">●</span>
L   _afterTokenTransfer   Internal   <span style="color:blue;">🔒</span>     <span style="color:red;">●</span>

```
| **Ownable** | Implementation | Context |||
| L | <Constructor> | Public ! | ● | NO! |
| L | owner | Public ! | | NO! |
| L | renounceOwnership | Public ! | ● | onlyOwner |
| L | transferOwnership | Public ! | ● | onlyOwner |
| L | _setOwner | Private 🔒 | ● | |
||||||

| **SafeMath** | Library | || |
| L | tryAdd | Internal 🔒 | | |
| L | trySub | Internal 🔒 | | |
| L | tryMul | Internal 🔒 | | |
| L | tryDiv | Internal 🔒 | | |
| L | tryMod | Internal 🔒 | | |
| L | add | Internal 🔒 | | |
| L | sub | Internal 🔒 | | |
| L | mul | Internal 🔒 | | |
| L | div | Internal 🔒 | | |
| L | mod | Internal 🔒 | | |
| L | sub | Internal 🔒 | | |
| L | div | Internal 🔒 | | |
| L | mod | Internal 🔒 | | |
||||||

| **Clones** | Library | || |
| L | clone | Internal 🔒 | ● | |
| L | cloneDeterministic | Internal 🔒 | ● | |
| L | predictDeterministicAddress | Internal 🔒 | | |
| L | predictDeterministicAddress | Internal 🔒 | | |
||||||

| **IUniswapV2Factory** | Interface | || |
| L | feeTo | External ! | | NO! |
| L | feeToSetter | External ! | | NO! |
| L | getPair | External ! | | NO! |
| L | allPairs | External ! | | NO! |
| L | allPairsLength | External ! | | NO! |
| L | createPair | External ! | ● | NO! |
| L | setFeeTo | External ! | ● | NO! |
| L | setFeeToSetter | External ! | ● | NO! |
```

```
| **IUniswapV2Router01** | Interface | ||| | |
| L | factory | External | | NO | |  
| L | WETH | External | | NO | |  
| L | addLiquidity | External | |  | NO | |  
| L | addLiquidityETH | External | |  | NO | |  
| L | removeLiquidity | External | |  | NO | |  
| L | removeLiquidityETH | External | |  | NO | |  
| L | removeLiquidityWithPermit | External | |  | NO | |  
| L | removeLiquidityETHWithPermit | External | |  | NO | |  
| L | swapExactTokensForTokens | External | |  | NO | |  
| L | swapTokensForExactTokens | External | |  | NO | |  
| L | swapExactETHForTokens | External | |  | NO | |  
| L | swapTokensForExactETH | External | |  | NO | |  
| L | swapExactTokensForETH | External | |  | NO | |  
| L | swapETHForExactTokens | External | |  | NO | |  
| L | quote | External | | NO | |  
| L | getAmountOut | External | | NO | |  
| L | getAmountIn | External | | NO | |  
| L | getAmountsOut | External | | NO | |  
| L | getAmountsIn | External | | NO | |  
||||||
```

```
| **IUniswapV2Router02** | Interface | IUniswapV2Router01 ||| | |
| L | removeLiquidityETHSupportingFeeOnTransferTokens | External | |  | NO | |  
| L | removeLiquidityETHWithPermitSupportingFeeOnTransferTokens | External | |  | NO | |  
| L | swapExactTokensForTokensSupportingFeeOnTransferTokens | External | |  | NO | |  
| L | swapExactETHForTokensSupportingFeeOnTransferTokens | External | |  | NO | |  
| L | swapExactTokensForETHSupportingFeeOnTransferTokens | External | |  | NO | |  
||||||
```

```
| **IERC20Upgradeable** | Interface | ||| | |
| L | totalSupply | External | | NO | |  
| L | balanceOf | External | | NO | |  
| L | transfer | External | |  | NO | |  
| L | allowance | External | | NO | |  
| L | approve | External | |  | NO | |  
| L | transferFrom | External | |  | NO | |
```

```
| **IERC20MetadataUpgradeable** | Interface | IERC20Upgradeable ||| |
| L | name | External | | NO | |
| L | symbol | External | | NO | |
| L | decimals | External | | NO | |
|||||
| **Initializable** | Implementation | ||
|||||
| **ContextUpgradeable** | Implementation | Initializable ||
| L | __Context_init | Internal 🔒 | 🔞 | initializer |
| L | __Context_init_unchained | Internal 🔒 | 🔞 | initializer |
| L | _msgSender | Internal 🔒 | |
| L | _msgData | Internal 🔒 | |
|||||
| **ERC20Upgradeable** | Implementation | Initializable, ContextUpgradeable, IERC20Upgradeable,
IERC20MetadataUpgradeable ||
| L | __ERC20_init | Internal 🔒 | 🔞 | initializer | |
| L | __ERC20_init_unchained | Internal 🔒 | 🔞 | initializer |
| L | name | Public | | NO | |
| L | symbol | Public | | NO | |
| L | decimals | Public | | NO | |
| L | totalSupply | Public | | NO | |
| L | balanceOf | Public | | NO | |
| L | transfer | Public | | NO | |
| L | allowance | Public | | NO | |
| L | approve | Public | | NO | |
| L | transferFrom | Public | | NO | |
| L | increaseAllowance | Public | | NO | |
| L | decreaseAllowance | Public | | NO | |
| L | _transfer | Internal 🔒 | 🔞 | |
| L | _mint | Internal 🔒 | 🔞 | |
| L | _burn | Internal 🔒 | 🔞 | |
| L | _approve | Internal 🔒 | 🔞 | |
| L | _beforeTokenTransfer | Internal 🔒 | 🔞 | |
| L | _afterTokenTransfer | Internal 🔒 | 🔞 | |
|||||
| **OwnableUpgradeable** | Implementation | Initializable, ContextUpgradeable ||
| L | __Ownable_init | Internal 🔒 | 🔞 | initializer |
| L | __Ownable_init_unchained | Internal 🔒 | 🔞 | initializer |
| L | owner | Public | | NO | |
| L | renounceOwnership | Public | | NO | onlyOwner |
| L | transferOwnership | Public | | NO | onlyOwner |
| L | _setOwner | Private 🔒 | 🔞 |
```

```
| **IUniswapV2Pair** | Interface | ||| | |
| L | name | External | | NO | |
| L | symbol | External | | NO | |
| L | decimals | External | | NO | |
| L | totalSupply | External | | NO | |
| L | balanceOf | External | | NO | |
| L | allowance | External | | NO | |
| L | approve | External | | O | NO | |
| L | transfer | External | | O | NO | |
| L | transferFrom | External | | O | NO | |
| L | DOMAIN_SEPARATOR | External | | NO | |
| L | PERMIT_TYPEHASH | External | | NO | |
| L | nonces | External | | NO | |
| L | permit | External | | O | NO | |
| L | MINIMUM_LIQUIDITY | External | | NO | |
| L | factory | External | | NO | |
| L | token0 | External | | NO | |
| L | token1 | External | | NO | |
| L | getReserves | External | | NO | |
| L | price0CumulativeLast | External | | NO | |
| L | price1CumulativeLast | External | | NO | |
| L | kLast | External | | NO | |
| L | mint | External | | O | NO | |
| L | burn | External | | O | NO | |
| L | swap | External | | O | NO | |
| L | skim | External | | O | NO | |
| L | sync | External | | O | NO | |
| L | initialize | External | | O | NO | |
||||||
```

```
| **SafeMathInt** | Library | || |
| L | mul | Internal 🔒 | | |
| L | div | Internal 🔒 | | |
| L | sub | Internal 🔒 | | |
| L | add | Internal 🔒 | | |
| L | abs | Internal 🔒 | | |
| L | toUint256Safe | Internal 🔒 | | |
```

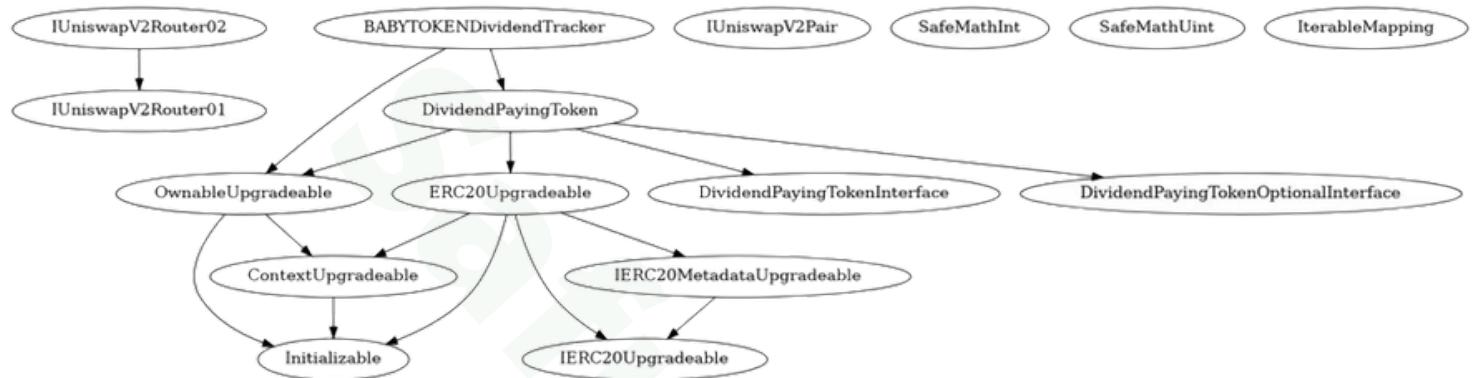
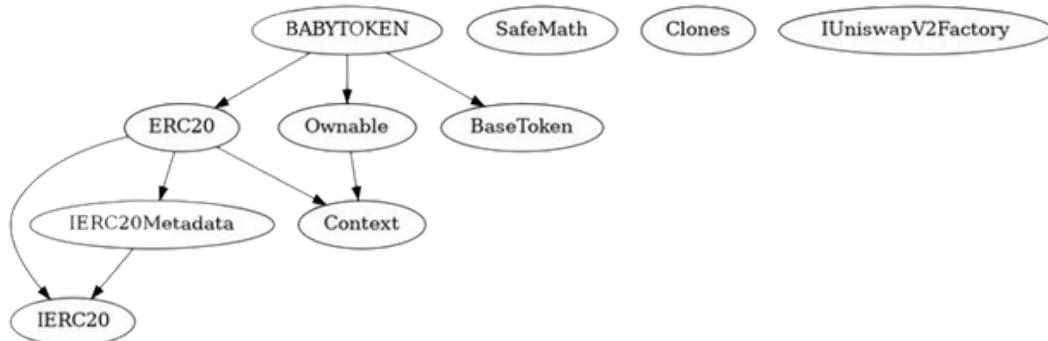
```
| **SafeMathUint** | Library | ||| | |
| L | toInt256Safe | Internal 🔒 | |||
|||||
| **IterableMapping** | Library | |||
| L | get | Public | | NO! | |
| L | getIndexOfKey | Public | | NO! | |
| L | getKeyAtIndex | Public | | NO! | |
| L | size | Public | | NO! | |
| L | set | Public | | 🔴 | NO! | |
| L | remove | Public | | 🔴 | NO! | |
|||||
| **DividendPayingTokenInterface** | Interface | |||
| L | dividendOf | External | | NO! | |
| L | withdrawDividend | External | | 🔴 | NO! | |
|||||
| **DividendPayingTokenOptionalInterface** | Interface | |||
| L | withdrawableDividendOf | External | | NO! | |
| L | withdrawnDividendOf | External | | NO! | |
| L | accumulativeDividendOf | External | | NO! | |
| **BABYTOKENDividendTracker** | Implementation | OwnableUpgradeable, DividendPayingToken ||
| L | initialize | External | | 🔴 | initializer | |
| L | _transfer | Internal 🔒 | | |
| L | withdrawDividend | Public | | NO! | |
| L | excludeFromDividends | External | | 🔴 | onlyOwner | |
| L | isExcludedFromDividends | Public | | NO! | |
| L | updateClaimWait | External | | 🔴 | onlyOwner | |
| L | updateMinimumTokenBalanceForDividends | External | | 🔴 | onlyOwner | |
| L | getLastProcessedIndex | External | | NO! | |
| L | getNumberOfTokenHolders | External | | NO! | |
| L | getAccount | Public | | NO! | |
| L | getAccountAtIndex | Public | | NO! | |
| L | canAutoClaim | Private 🎉 | | |
| L | setBalance | External | | 🔴 | onlyOwner | |
| L | process | Public | | 🔴 | NO! | |
| L | processAccount | Public | | 🔴 | onlyOwner |
```

\*\*BABYTOKEN\*\* | Implementation | ERC20, Ownable, BaseToken |||

- | L | <Constructor> | Public ! |  | ERC20 |
- | L | <Receive Ether> | External ! |  | NO! |
- | L | setSwapTokensAtAmount | External ! |  | onlyOwner |
- | L | updateDividendTracker | Public ! |  | onlyOwner |
- | L | updateUniswapV2Router | Public ! |  | onlyOwner |
- | L | excludeFromFees | Public ! |  | onlyOwner |
- | L | excludeMultipleAccountsFromFees | Public ! |  | onlyOwner |
- | L | setMarketingWallet | External ! |  | onlyOwner |
- | L | setTokenRewardsFee | External ! |  | onlyOwner |
- | L | setLiquidityFee | External ! |  | onlyOwner |
- | L | setMarketingFee | External ! |  | onlyOwner |
- | L | setAutomatedMarketMakerPair | Public ! |  | onlyOwner |
- | L | \_setAutomatedMarketMakerPair | Private  |  |
- | L | updateGasForProcessing | Public ! |  | onlyOwner |
- | L | updateClaimWait | External ! |  | onlyOwner |
- | L | getClaimWait | External ! | NO! |
- | L | updateMinimumTokenBalanceForDividends | External ! |  | onlyOwner |
- | L | getMinimumTokenBalanceForDividends | External ! | NO! |
- | L | getTotalDividendsDistributed | External ! | NO! |
- | L | isExcludedFromFees | Public ! | NO! |
- | L | withdrawableDividendOf | Public ! | NO! |
- | L | dividendTokenBalanceOf | Public ! | NO! |
- | L | excludeFromDividends | External ! |  | onlyOwner |
- | L | isExcludedFromDividends | Public ! | NO! |
- | L | getAccountDividendsInfo | External ! | NO! |
- | L | getAccountDividendsInfoAtIndex | External ! | NO! |
- | L | processDividendTracker | External ! |  | NO! |
- | L | claim | External ! |  | NO! |
- | L | getLastProcessedIndex | External ! | NO! |
- | L | getNumberOfDividendTokenHolders | External ! | NO! |
- | L | \_transfer | Internal  |  |
- | L | swapAndSendToFee | Private  |  |
- | L | swapAndLiquify | Private  |  |
- | L | swapTokensForEth | Private  |  |
- | L | swapTokensForCake | Private  |  |
- | L | addLiquidity | Private  |  |
- | L | swapAndSendDividends | Private  |  |

Symbol	Meaning
	Function can modify state
	Function is payable
	Private function
	Internal function
NO!	Function has no modifier

# INHERITANCE TREE



Inheritance is a feature of the object-oriented programming language. It is a way of extending the functionality of a program, used to separate the code, reduces the dependency, and increases the re-usability of the existing code. Solidity supports inheritance between smart contracts, where multiple contracts can be inherited into a single contract.

# Important Snippets



## Exclude multiple accounts from fees

```
function excludeMultipleAccountsFromFees(
    address[] calldata accounts,
    bool excluded
) public onlyOwner {
    for (uint256 i = 0; i < accounts.length; i++) {
        _isExcludedFromFees[accounts[i]] = excluded;
    }

    emit ExcludeMultipleAccountsFromFees(accounts, excluded);
}
```

## Exclude from dividends

```
function excludeFromDividends(address account) external onlyOwner {
    dividendTracker.excludeFromDividends(account);
}
```

## Owner can set fees up to 25%

```
function setTokenRewardsFee(uint256 value) external onlyOwner {
    tokenRewardsFee = value;
    totalFees = tokenRewardsFee.add(liquidityFee).add(marketingFee);
    require(totalFees <= 25, "Total fee is over 25%");
}

function setLiquidityFee(uint256 value) external onlyOwner {
    liquidityFee = value;
    totalFees = tokenRewardsFee.add(liquidityFee).add(marketingFee);
    require(totalFees <= 25, "Total fee is over 25%");
}

function setMarketingFee(uint256 value) external onlyOwner {
    marketingFee = value;
    totalFees = tokenRewardsFee.add(liquidityFee).add(marketingFee);
    require(totalFees <= 25, "Total fee is over 25%");
}
```

# GOOD PRACTICES ✓

---

- The owner cannot stop or pause the smart contract
- The owner cannot mint new tokens after deployment
- The owner cannot set the fees over 25%
- The owner cannot set max Tx
- The smart contract utilizes "SafeMath" to prevent overflows

```
library SafeMath {
    function tryAdd(uint256 a, uint256 b) internal pure returns (bool, uint256) {
        unchecked {
            uint256 c = a + b;
            if (c < a) return (false, 0);
            return (true, c);
        }
    }

    function trySub(uint256 a, uint256 b) internal pure returns (bool, uint256) {
        unchecked {
            if (b > a) return (false, 0);
            return (true, a - b);
        }
    }

    function tryMul(uint256 a, uint256 b) internal pure returns (bool, uint256) {
        unchecked {
            // Gas optimization: this is cheaper than requiring 'a' not being zero, but
            // benefit is lost if 'b' is also tested.
            // See: https://github.com/OpenZeppelin/openzeppelin-contracts/pull/522
            if (a == 0) return (true, 0);
            uint256 c = a * b;
            if (c / a != b) return (false, 0);
            return (true, c);
        }
    }

    function tryDiv(uint256 a, uint256 b) internal pure returns (bool, uint256) {
        unchecked {
            if (b == 0) return (false, 0);
            return (true, a / b);
        }
    }

    function tryMod(uint256 a, uint256 b) internal pure returns (bool, uint256) {
        unchecked {
            if (b == 0) return (false, 0);
            return (true, a % b);
        }
    }
}
```

# WEBSITE



<b>Website</b>	<a href="http://shibabusd.com/">http://shibabusd.com/</a>
<b>Domain Registry</b>	<a href="http://www.openprovider.com">http://www.openprovider.com</a>
<b>Domain Expiry Date</b>	2023-02-11
<b>Response Code</b>	200
<b>SSL Checker and HTTPS Test</b>	High
<b>Deprecated HTML tags</b>	Passed
<b>Robots.txt</b>	Passed
<b>Sitemap Test</b>	Passed
<b>SEO Friendly URL</b>	Passed
<b>Responsive Test</b>	Low
<b>JS Error Test</b>	Passed
<b>Console Errors Test</b>	Passed
<b>Site Loading Speed Test</b>	2.98 seconds - Passed
<b>HTTP2 Test</b>	Medium
<b>Safe Browsing Test</b>	Passed

# DISCLAIMER

---

SafuAudit.com is not a financial institution and the information provided on this website does not constitute investment advice, financial advice, trading advice or any other sort of advice. You should not treat any of the website's content as such. Investing in crypto assets carries a high level of risk and does not hold guarantees for not sustaining financial loss due to their volatility.

## Accuracy of Information

SafuAudit will strive to ensure accuracy of information listed on this website although it will not hold any responsibility for any missing or wrong information. SafuAudit provides all information as is. You understand that you are using any and all information available here at your own risk. Any use or reliance on our content and services is solely at your own risk and discretion.

The purpose of the audit is to analyse the on-chain smart contract source code, and to provide basic overview of the project.

While we have used all the information available to us for this straightforward investigation, you should not rely on this report only – we recommend proceeding with several independent audits. Be aware that smart contracts deployed on a blockchain aren't secured enough against external vulnerability, or a hack. Be aware that active smart contract owner privileges constitute an elevated impact to smart contract's safety and security. Therefore, SafuAudit does not guarantee the explicit security of the audited smart contract. 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.

# AUDIT RESULTS

---

## CRITICAL

---

No critical severity issues have been found.

## MEDIUM

---

No medium severity issues have been found.

## MINOR

---

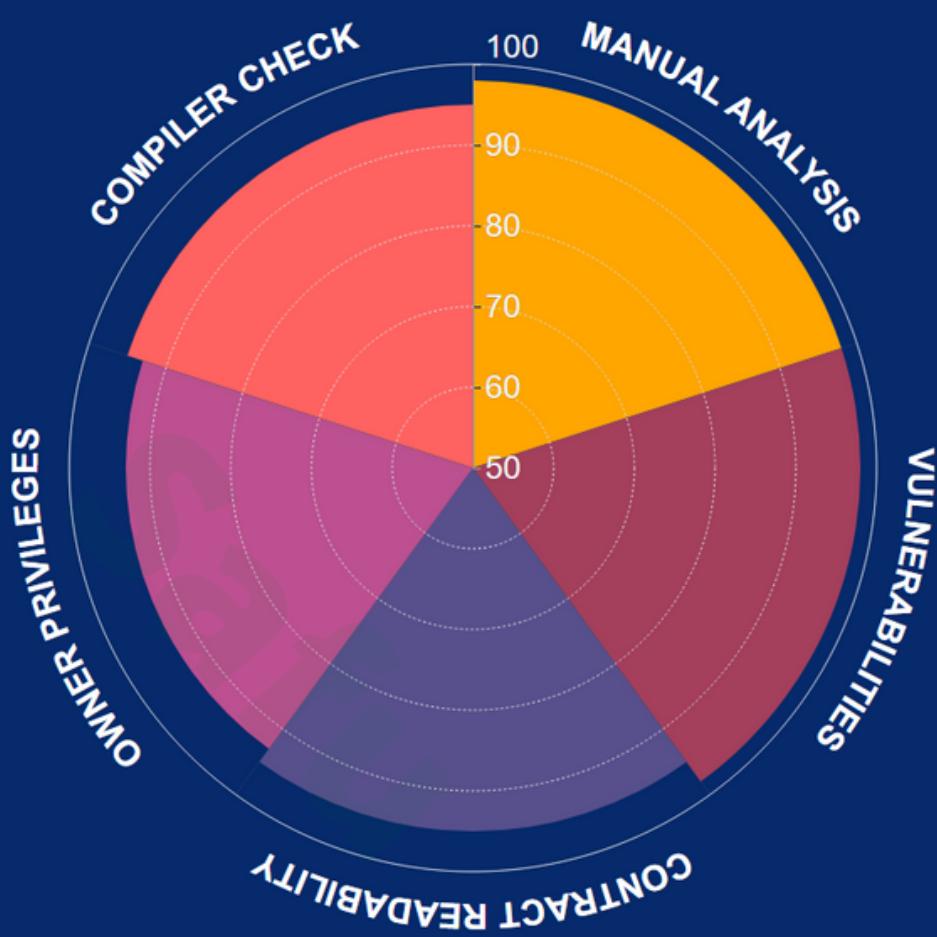
- Using "tx.origin" as a security control can lead to authorization bypass vulnerabilities. Consider using "msg.sender" instead.

## INFORMATIONAL

---

The standard audit model does not offer suggestions and consulting for improvements of efficacy.

# SAFUSCORE



Manual Analysis



Vulnerabilities



Contract Readability



Owner Privileges

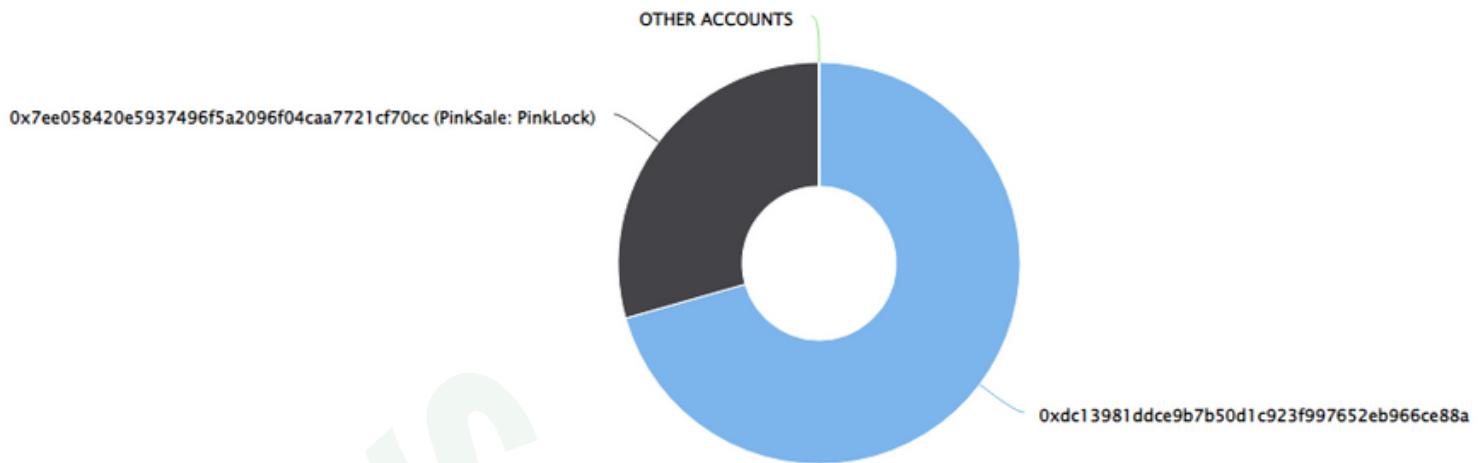


Compiler Check

**Final Score: 95.8**

# SUMMARY

## Top 10 holders



Rank	Address	Quantity (Token)	Percentage
1	0xdc13981ddce9b7b50d1c923f997652eb966ce88a	70,596,000	70.5960%
2	0x7ee058420e5937496f5a2096f04caa7721cf70cc (PinkSale: PinkLock)	29,404,000	29.4040%

## Conclusion

Project ShibaBUSD does not contain any severe issues or risk characteristics.

SafuAudit has tested the security based on manual and automated tests.

Please note that we don't offer any warranties for business model.





**SafuAudit.com**

