



Smart Contract Audit

FOR
Skills 100
DATED : 08 Mar 23'



AUDIT SUMMARY

Project name - Skills 100

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/0x443a0Eca78bb506149AE747aA7659735aaD7E94E#readContract>



Token Information

Token Name : Skills 100

Token Symbol: S100

Decimals: 18

Token Supply: 100,000

Token Address:

0x09C0e2bC53A2bB2da18CDDe4b190FE585aF68b75

Checksum:

1f2fc18bc0c9d665cb1505a567a63130ac1fb09f

Owner:

0x3cb718c917749338f362b99dCC170Fe741057D3F

(at time of writing the audit)



TOKEN OVERVIEW

Fees:

Buy Fees: 7% Currently

Sell Fees: 7% Currently

Transfer Fees: 7% Currently

Fees Privilege: Owner

Ownership : Owned

Minting: No mint function

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

Other Privileges: including and excluding form fee - changing distribution settings (min tokens to be eligible, cooldown between claims etc)



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 Limit and 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



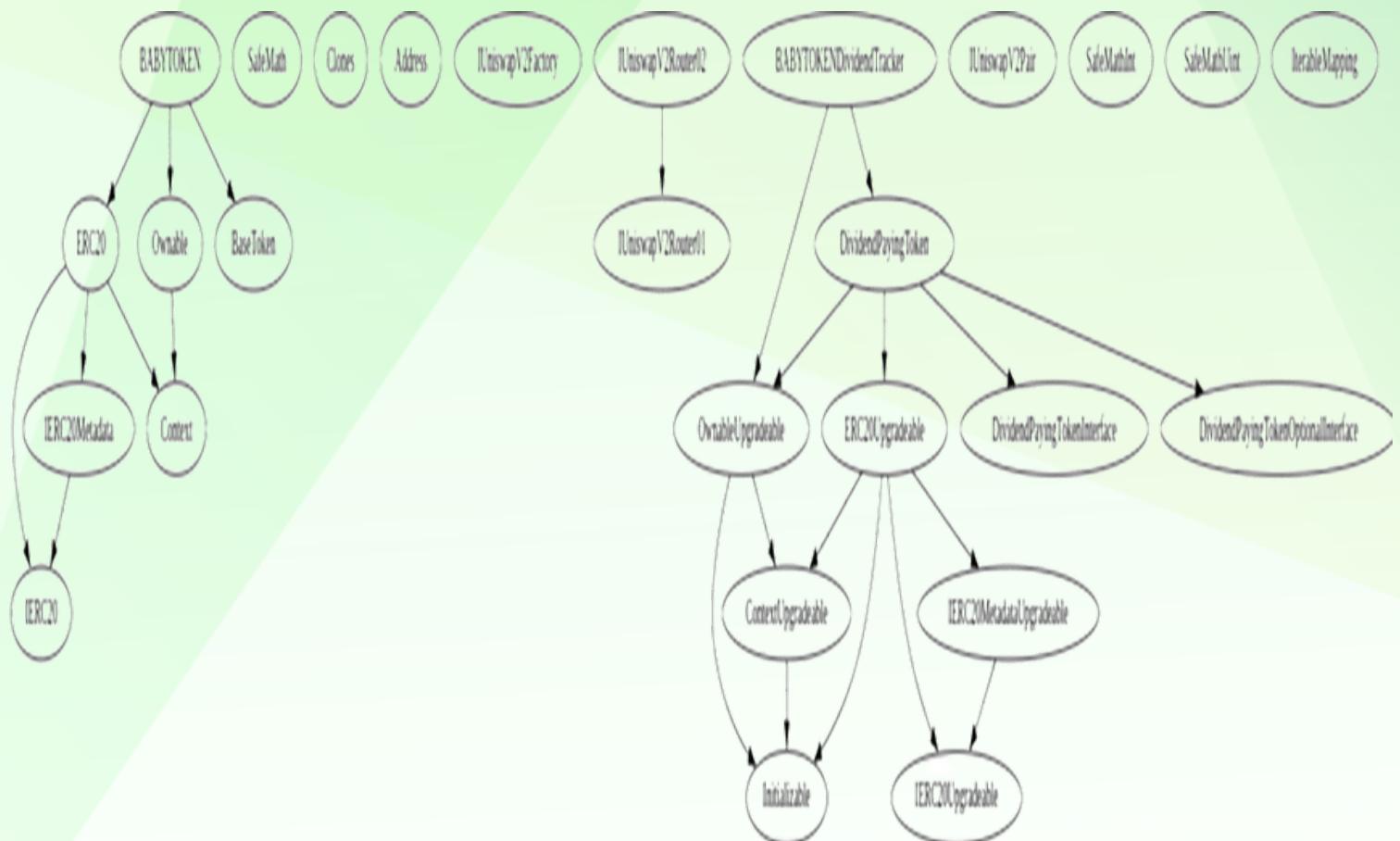
CLASSIFICATION OF RISK

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	1
◆ Gas Optimization / Suggestions	0

INHERITANCE TREE





OVERVIEW

This token has three different types of fees, which are used for marketing, maintaining liquidity, and providing rewards to holders. The fees collected are converted to a stablecoin called **BUSD** and sent to a marketing wallet. Another part of the fee is used to maintain liquidity and is sent to a dividend tracker to provide **BUSD** rewards to holders. Rewards are distributed to holders every time a buy, sell or transfer occurs, and can also be manually claimed from the BscScan website.



POINTS TO NOTE

- Owner is able to change buy/sell/transfer fees but sum of fees can not exceed 25%
- Contract is recognized as proxy in bscscan, however this is because dividend distributor is minimal proxy contract which is not upgradeable currently
- 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



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.

CONTRACT ASSESSMENT

Contract	Type	Bases			
	Function Name	**Visibility**	**Mutability**	**Modifiers**	
	IERC20 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 !	
	IERC20Metadata Interface IERC20				
L	name	External !	NO !		
L	symbol	External !	NO !		
L	decimals	External !	NO !		
	Context Implementation				
L	_msgSender	Internal 			
L	_msgData	Internal 			
	ERC20 Implementation Context, IERC20, IERC20Metadata				
L	<Constructor>	Public !		NO !	
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			
	Ownable Implementation Context				



CONTRACT ASSESSMENT

```
| 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 🔒 | |||
|||||||
| **Address** | Library | ||
| L | isContract | Internal 🔒 | |||
| L | sendValue | Internal 🔒 | ○ | ||
| L | functionCall | Internal 🔒 | ○ | ||
| L | functionCall | Internal 🔒 | ○ | ||
| L | functionCallWithValue | Internal 🔒 | ○ | ||
| L | functionCallWithValue | Internal 🔒 | ○ | ||
| L | functionStaticCall | Internal 🔒 | |||
| L | functionStaticCall | Internal 🔒 | |||
| L | functionDelegateCall | Internal 🔒 | ○ | ||
| L | functionDelegateCall | Internal 🔒 | ○ | ||
| L | verifyCallResult | Internal 🔒 | |||
|||||||
| **IUniswapV2Factory** | Interface | ||
| L | feeTo | External ! | | NO! |
```

CONTRACT ASSESSMENT

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!

CONTRACT ASSESSMENT

```
| 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! |
```

CONTRACT ASSESSMENT

```
| L | renounceOwnership | Public ! | ○ | onlyOwner | |
| L | transferOwnership | Public ! | ○ | 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 ! | ○ | NO! |
| L | transfer | External ! | ○ | NO! |
| L | transferFrom | External ! | ○ | NO! |
| L | DOMAIN_SEPARATOR | External ! | | NO! |
| L | PERMIT_TYPEHASH | External ! | | NO! |
| L | nonces | External ! | | NO! |
| L | permit | External ! | ○ | 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 ! | ○ | NO! |
| L | burn | External ! | ○ | NO! |
| L | swap | External ! | ○ | NO! |
| L | skim | External ! | ○ | NO! |
| L | sync | External ! | ○ | NO! |
| L | initialize | External ! | ○ | NO! |
|||||||
| **SafeMathInt** | Library | ||
| L | mul | Internal 🔒 | | |
| L | div | Internal 🔒 | | |
| L | sub | Internal 🔒 | | |
| L | add | Internal 🔒 | | |
| L | abs | Internal 🔒 | | |
| L | toUint256Safe | Internal 🔒 | | |
|||||||
```



CONTRACT ASSESSMENT

```
| **SafeMathUint** | Library | ||| |
| L | toInt256Safe | Internal 🔒 | |||
|||||||
| **IterableMapping** | Library | |||
| L | get | Internal 🔒 | |||
| L | getIndexOfKey | Internal 🔒 | |||
| L | getKeyAtIndex | Internal 🔒 | |||
| L | size | Internal 🔒 | |||
| L | set | Internal 🔒 | 🔴 | |
| L | remove | Internal 🔒 | 🔴 | |
|||||||
| **DividendPayingTokenInterface** | Interface | |||
| L | dividendOf | External 🔴 | | NO! | |
| L | withdrawDividend | External 🔴 | 🔴 | NO! | |
|||||||
| **DividendPayingTokenOptionalInterface** | Interface | |||
| L | withdrawableDividendOf | External 🔴 | | NO! | |
| L | withdrawnDividendOf | External 🔴 | | NO! | |
| L | accumulativeDividendOf | External 🔴 | | NO! | |
|||||||
| **DividendPayingToken** | Implementation | ERC20Upgradeable, OwnableUpgradeable,
DividendPayingTokenInterface, DividendPayingTokenOptionalInterface | ||
| L | __DividendPayingToken_init | Internal 🔒 | 🔴 | initializer | |
| L | distributeCAKEDividends | Public 🔴 | 🔴 | onlyOwner | |
| L | withdrawDividend | Public 🔴 | 🔴 | NO! | |
| L | _withdrawDividendOfUser | Internal 🔒 | 🔴 | |
| L | dividendOf | Public 🔴 | | NO! | |
| L | withdrawableDividendOf | Public 🔴 | | NO! | |
| L | withdrawnDividendOf | Public 🔴 | | NO! | |
| L | accumulativeDividendOf | Public 🔴 | | NO! | |
| L | _transfer | Internal 🔒 | 🔴 | |
| L | _mint | Internal 🔒 | 🔴 | |
| L | _burn | Internal 🔒 | 🔴 | |
| L | _setBalance | Internal 🔒 | 🔴 | |
|||||||
| **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 | |
```

CONTRACT ASSESSMENT

```
| 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 |
|||||||
| **BaseToken** | Implementation | ||
|||||||
| **BABYTOKEN** | Implementation | ERC20, Ownable, BaseToken |||
| L | <Constructor> | Public ! | 📁 | ERC20 |
| L | <Receive Ether> | External ! | 📁 | NO! |
| L | setSwapTokensAtAmount | External ! | ○ | onlyOwner |
| L | excludeFromFees | External ! | ○ | onlyOwner |
| L | excludeMultipleAccountsFromFees | External ! | ○ | onlyOwner |
| L | setMarketingWallet | External ! | ○ | onlyOwner |
| L | setTokenRewardsFee | External ! | ○ | onlyOwner |
| L | setLiquiditFee | External ! | ○ | onlyOwner |
| L | setMarketingFee | External ! | ○ | 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 🔒 | ○ | |
```



CONTRACT ASSESSMENT

		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	



STATIC ANALYSIS

**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/0x38a1917661da7d7e321ad422799e23e3910cf330b6b8e1c2aee941bf63b9615>

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

<https://testnet.bscscan.com/tx/0x7a11a88f56898ede5c755a818cb78025651bf441add628edf45f17e75d9f6598>

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

<https://testnet.bscscan.com/tx/0x2e3e2d75be52402cd9199fbccf452bf0d511a4e69f2a8fb66df6348efde6778b>

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

<https://testnet.bscscan.com/tx/0xb26c4f1d9cd73ea4907080ec860c5ad7b36d9835d79bd5eb3125cd6bdfda8c83>



FUNCTIONAL TESTING

5- Buying when not excluded from fees (7% tax) (passed):

<https://testnet.bscscan.com/tx/0xd1680de78c32da5d156474eff6fab11c6152132e194c1c333acca1222e9c4581>

6- Selling when not excluded from fees (7% tax) (passed):

<https://testnet.bscscan.com/tx/0x9245bf64d382d5f17e740338247eba549805847253c22b2259d99eea7c289194>

7- Transferring when not excluded from fees (7% tax) (passed):

<https://testnet.bscscan.com/tx/0x54070a16b9c7b56f447c511c5079827d448e326cc37052b33f42a08c761fccef>

8-Internal swap (passed):

marketing wallet received ETH

<https://testnet.bscscan.com/tx/0x9245bf64d382d5f17e740338247eba549805847253c22b2259d99eea7c289194>

9- Reflections (passed):

as can be seen in this transaction:

<https://testnet.bscscan.com/tx/0x9245bf64d382d5f17e740338247eba549805847253c22b2259d99eea7c289194>

distributor is sending BUSD to holders (auto distribution)

10- Auto Liquidity (passed):

<https://testnet.bscscan.com/token/0x34e07586acbcd1046bb54102361c41873167156?>

a=0x00dead



MANUAL TESTING

Low Risk Issue

Issue: some ERC20 tokens may not return a boolean after transfer success

Type : **Logical**

Function: _withdrawDividendOfUser

Line: 2370 - 2380

Severity: **Low**

Overview: Some ERC20 contracts may not support returning a boolean (true) if the transfer was successful

```
function _withdrawDividendOfUser(
    address payable user)
internal returns (uint256) {
    uint256 _withdrawableDividend = withdrawableDividendOf(user);
    if (_withdrawableDividend > 0) {
        withdrawnDividends[user] = withdrawnDividends[user].add(
            _withdrawableDividend
        );
        emit DividendWithdrawn(user, _withdrawableDividend);
        bool success = IERC20(rewardToken).transfer(
            user,
            _withdrawableDividend
        );
        if (!success) {
            withdrawnDividends[user] = withdrawnDividends[user].sub(
                _withdrawableDividend
            );
            return 0;
        }
    }
}
```

Recommendations

use SafeERC20 to handle token transfers



Social Media Overview

**Here are the Social Media Accounts of
Skills 100**



https://t.me/s100_token



https://twitter.com/S100_token



<https://skills100.net/>



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://twitter.com/auditace_



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