auditace.tech

AuditAce

FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

SBF

DATED : 5 MAY 23'

AUDIT SUMMARY

Project name – SBF

Date: 5 May, 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: The code has undergone a line-by-line review by the **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.

Token Information

Name : SBF CEO

Symbol : SBFCEO

Decimals: 18

Network: Binance smart chain

Token Type: BEP20

Token Address :
0xAbB8752cd864670c1ad80DEE559C98fFe531bfcf

Owner:
0x6a88aAD5547035BDE42020B18CDF4beD566a2e56**(at time of audit)**

Deployer:
0x087fC35808268055D82e6638E280dE517509f518

Token Information

Fees:

Buy Fees: 10%

Sell Fees: 10%

Transfer Fees: 0%

Fees Privilige: None

Ownership : Owned

Minting: None

Max Tx Amount/ Max Wallet Amount: Yes

Blacklist: No

Other Priviliges: Including or excluding from fees - changing swap threshold

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

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

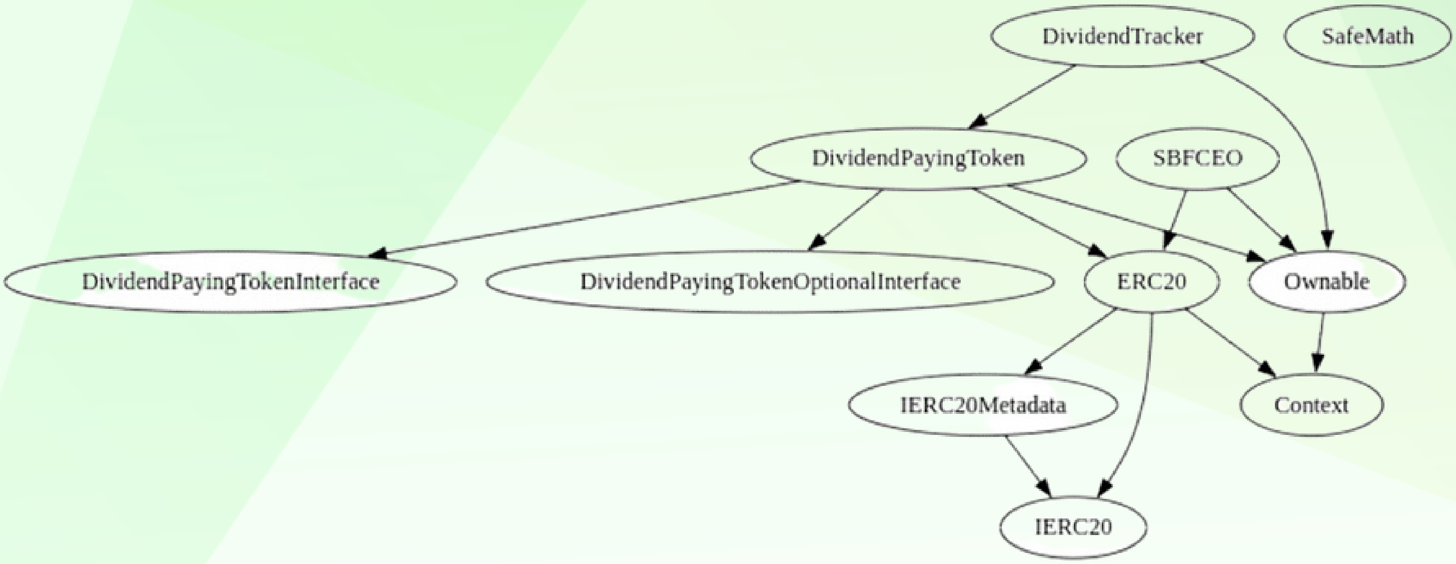
CLASSIFICATION OF RISK

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 not able to change current tax (10% for buy and sell)
- Owner is not ablet o set transfer taxes (0% forever)
- 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 mint new tokens
- Owner is not able to disable trades
- Trades are already enabled for public
- Token has BNB rewards with auto-distribution
- Fees are used for rewards (BNB) and marketing

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
└	_transferOwnership	Internal 🔒	🛑	
SafeMath	Library			
└	add	Internal 🔒		
└	sub	Internal 🔒		
└	sub	Internal 🔒		
└	mul	Internal 🔒		
└	div	Internal 🔒		
└	div	Internal 🔒		
└	mod	Internal 🔒		
└	mod	Internal 🔒		
SafeMathInt	Library			
└	mul	Internal 🔒		
└	div	Internal 🔒		
└	sub	Internal 🔒		
└	add	Internal 🔒		
└	abs	Internal 🔒		
└	toUint256Safe	Internal 🔒		
SafeMathUint	Library			
└	toInt256Safe	Internal 🔒		
IterableMapping	Library			
└	get	Internal 🔒		
└	getIndexOfKey	Internal 🔒		
└	getKeyAtIndex	Internal 🔒		
└	size	Internal 🔒		

CONTRACT ASSESMENT

```
| └ | set | Internal 🔒 | 🛑  | |
| └ | remove | Internal 🔒 | 🛑  | |
||||||
| **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 ❗️ |
| └ | mint | 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 ❗ |
| └ | removeLiquidityETHWithPermit | External ❗ | 🛑  |NO ❗ |
| └ | swapExactTokensForTokens | External ❗ | 🛑  |NO ❗ |
| └ | swapTokensForExactTokens | External ❗ | 🛑  |NO ❗ |
| └ | swapExactETHForTokens | External ❗ | 💵 |NO ❗ |
| └ | swapTokensForExactETH | External ❗ | 🛑  |NO ❗ |
| └ | swapExactTokensForETH | External ❗ | 🛑  |NO ❗ |
| └ | swapETHForExactTokens | External ❗ | 💵 |NO ❗ |
| └ | quote | External ❗ |   |NO ❗ |
| └ | getAmountOut | External ❗ |   |NO ❗ |
| └ | getAmountIn | External ❗ |   |NO ❗ |
| └ | getAmountsOut | External ❗ |   |NO ❗ |
| └ | getAmountsIn | External ❗ |   |NO ❗ |
||||||
| **IUniswapV2Router02** | Interface | IUniswapV2Router01 |||
| └ | removeLiquidityETHSupportingFeeOnTransferTokens | External ❗ | 🛑  |NO ❗ |
| └ | removeLiquidityETHWithPermitSupportingFeeOnTransferTokens | External ❗ | 🛑  |NO ❗ |
| └ | swapExactTokensForTokensSupportingFeeOnTransferTokens | External ❗ | 🛑  |NO ❗ |
| └ | swapExactETHForTokensSupportingFeeOnTransferTokens | External ❗ | 💵 |NO ❗ |
| └ | swapExactTokensForETHSupportingFeeOnTransferTokens | External ❗ | 🛑  |NO ❗ |
||||||
| **IERC20** | Interface |  |||
| └ | totalSupply | External ❗ |   |NO ❗ |
| └ | balanceOf | External ❗ |   |NO ❗ |
| └ | allowance | External ❗ |   |NO ❗ |
| └ | transfer | External ❗ | 🛑  |NO ❗ |
| └ | approve | External ❗ | 🛑  |NO ❗ |
| └ | transferFrom | External ❗ | 🛑  |NO ❗ |
||||||
| **Address** | Library |  |||
| └ | isContract | Internal 🔒 |   | |
| └ | sendValue | Internal 🔒 | 🛑  | |
| └ | functionCall | Internal 🔒 | 🛑  | |
```

CONTRACT ASSESMENT

```
| └ | functionCall | Internal 🔒 | 🛑 | |
| └ | functionCallWithValue | Internal 🔒 | 🛑 | |
| └ | functionCallWithValue | Internal 🔒 | 🛑 | |
| └ | _functionCallWithValue | Private 🔐 | 🛑 | |
||||||
| **IERC20Metadata** | Interface | IERC20 |||
| └ | name | External ❗ | |NO ❗ |
| └ | symbol | External ❗ | |NO ❗ |
| └ | decimals | External ❗ | |NO ❗ |
||||||
| **ERC20** | Implementation | Context, IERC20, IERC20Metadata |||
| └ | <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 ❗ |
| └ | _transfer | Internal 🔒 | 🛑 | |
| └ | _mint | Internal 🔒 | 🛑 | |
| └ | _burn | Internal 🔒 | 🛑 | |
| └ | _approve | Internal 🔒 | 🛑 | |
| └ | _beforeTokenTransfer | Internal 🔒 | 🛑 | |
||||||
| **DividendPayingTokenInterface** | Interface | |||
| └ | dividendOf | External ❗ | |NO ❗ |
| └ | withdrawDividend | External ❗ | 🛑 |NO ❗ |
||||||
| **DividendPayingTokenOptionalInterface** | Interface | |||
| └ | withdrawableDividendOf | External ❗ | |NO ❗ |
| └ | withdrawnDividendOf | External ❗ | |NO ❗ |
| └ | accumulativeDividendOf | External ❗ | |NO ❗ |
||||||
| **DividendPayingToken** | Implementation | ERC20, Ownable, DividendPayingTokenInterface,
DividendPayingTokenOptionalInterface |||
| └ | <Constructor> | Public ❗ | 🛑 | ERC20 |
| └ | distributeDividends | Public ❗️ | 🛑  | onlyOwner |
| └ | withdrawDividend | Public ❗️ | 🛑  |NO❗️ |
| └ | _withdrawDividendOfUser | Internal 🔒 | 🛑  | |
| └ | dividendOf | Public ❗️ |   |NO❗️ |
| └ | withdrawableDividendOf | Public ❗️ |   |NO❗️ |
| └ | withdrawnDividendOf | Public ❗️ |   |NO❗️ |
| └ | accumulativeDividendOf | Public ❗️ |   |NO❗️ |
| └ | _transfer | Internal 🔒 | 🛑  | |
| └ | _mint | Internal 🔒 | 🛑  | |
| └ | _burn | Internal 🔒 | 🛑  | |
| └ | _setBalance | Internal 🔒 | 🛑  | |
||||||
| **DividendTracker** | Implementation | Ownable, DividendPayingToken |||
| └ | <Constructor> | Public ❗️ | 🛑  | DividendPayingToken |
| └ | <Receive Ether> | External ❗️ | 💵 |NO❗️ |
| └ | _transfer | Internal 🔒 |   | |
| └ | withdrawDividend | Public ❗️ |   |NO❗️ |
| └ | updateMinimumTokenBalanceForDividends | External ❗️ | 🛑  | onlyOwner |
| └ | excludeFromDividends | External ❗️ | 🛑  | onlyOwner |
| └ | updateClaimWait | External ❗️ | 🛑  | onlyOwner |
| └ | setLastProcessedIndex | External ❗️ | 🛑  | onlyOwner |
| └ | getLastProcessedIndex | External ❗️ |   |NO❗️ |
| └ | getNumberOfTokenHolders | External ❗️ |   |NO❗️ |
| └ | getAccount | Public ❗️ |   |NO❗️ |
| └ | getAccountAtIndex | Public ❗️ |   |NO❗️ |
| └ | canAutoClaim | Private 🔐 |   | |
| └ | setBalance | External ❗️ | 🛑  | onlyOwner |
| └ | process | Public ❗️ | 🛑  |NO❗️ |
| └ | processAccount | Public ❗️ | 🛑  | onlyOwner |
||||||
| **SBFCEO** | Implementation | ERC20, Ownable |||
| └ | <Constructor> | Public ❗️ | 🛑  | ERC20 |
| └ | <Receive Ether> | External ❗️ | 💵 |NO❗️ |
| └ | claimStuckTokens | External ❗️ | 🛑  | onlyOwner |
| └ | excludeFromFees | External ❗️ | 🛑  | onlyOwner |
| └ | isExcludedFromFees | Public ❗️ |   |NO❗️ |
| └ | changeMarketingWallet | External ❗️ | 🛑  | onlyOwner |
| └ | _transfer | Internal 🔒 | 🛑  | |
| └ | swapAndSendDividends | Private 🔐 | 🛑  | |
| └ | setSwapEnabled | External ❗️ | 🛑  | onlyOwner |
```

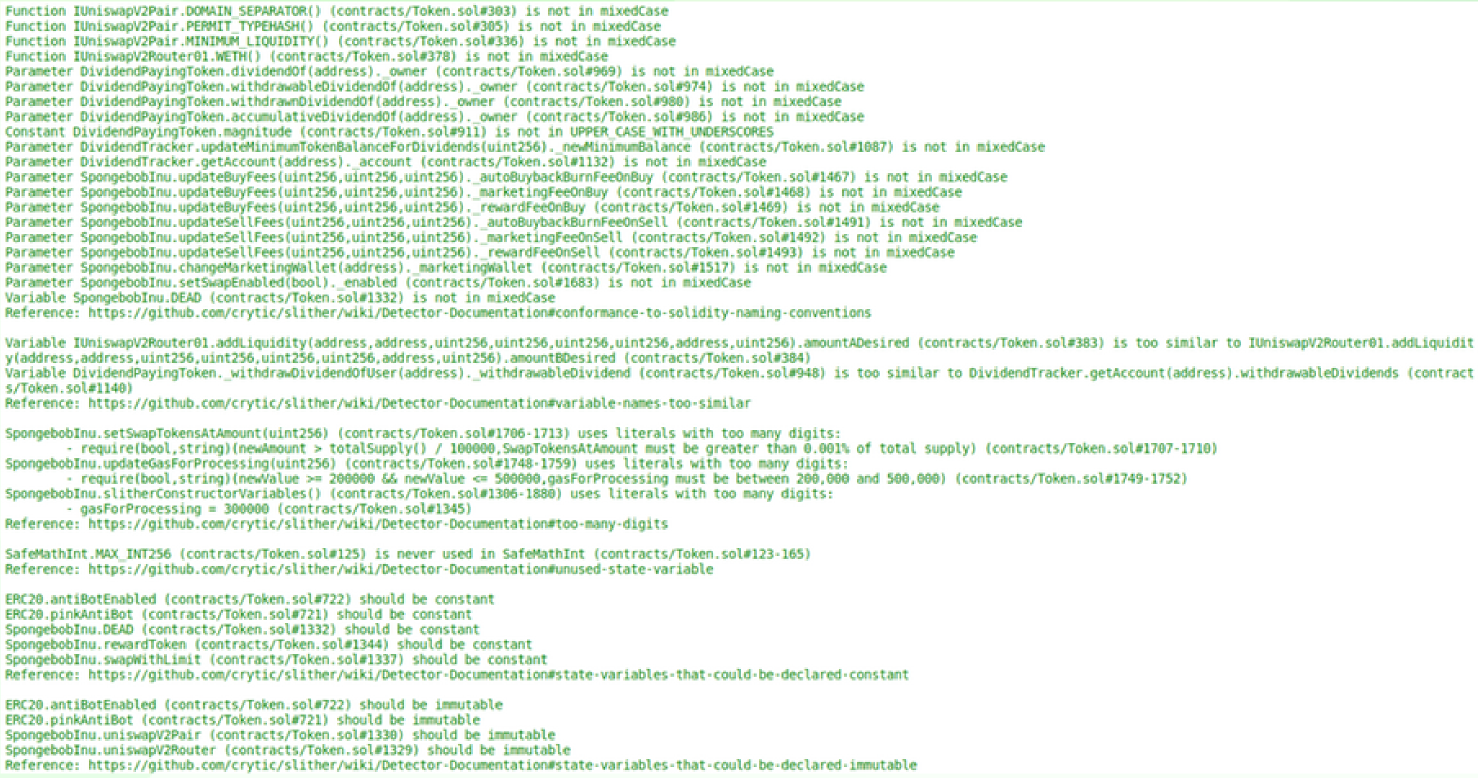
CONTRACT ASSESMENT

```
| └ | setSwapTokensAtAmount | External ❗ | 🛑 | onlyOwner |
| └ | updateGasForProcessing | Public ❗ | 🛑 | onlyOwner |
| └ | updateMinimumBalanceForDividends | External ❗ | 🛑 | onlyOwner |
| └ | updateClaimWait | External ❗ | 🛑 | onlyOwner |
| └ | getClaimWait | External ❗ | |NO ❗ |
| └ | getTotalDividendsDistributed | External ❗ | |NO ❗ |
| └ | withdrawableDividendOf | Public ❗ | |NO ❗ |
| └ | dividendTokenBalanceOf | Public ❗ | |NO ❗ |
| └ | totalRewardsEarned | Public ❗ | |NO ❗ |
| └ | excludeFromDividends | External ❗ | 🛑 | onlyOwner |
| └ | isExcludedFromDividends | Public ❗ | |NO ❗ |
| └ | getAccountDividendsInfo | External ❗ | |NO ❗ |
| └ | getAccountDividendsInfoAtIndex | External ❗ | |NO ❗ |
| └ | processDividendTracker | External ❗ | 🛑 |NO ❗ |
| └ | claim | External ❗ | 🛑 |NO ❗ |
| └ | claimAddress | External ❗ | 🛑 | onlyOwner |
| └ | getLastProcessedIndex | External ❗ | |NO ❗ |
| └ | setLastProcessedIndex | External ❗ | 🛑 | onlyOwner |
| └ | getNumberOfDividendTokenHolders | External ❗ | |NO ❗ |
```

Legend

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

STATIC ANALYSIS



```
Function IUniswapV2Pair.DOMAIN_SEPARATOR() (contracts/Token.sol#303) is not in mixedCase
Function IUniswapV2Pair.PERMIT_TYPEHASH() (contracts/Token.sol#305) is not in mixedCase
Function IUniswapV2Pair.MINIMUM_LIQUIDITY() (contracts/Token.sol#336) is not in mixedCase
Function IUniswapV2Router01.WETH() (contracts/Token.sol#378) is not in mixedCase
Parameter DividendPayingToken.dividendOf(address)._owner (contracts/Token.sol#969) is not in mixedCase
Parameter DividendPayingToken.withdrawableDividendOf(address)._owner (contracts/Token.sol#974) is not in mixedCase
Parameter DividendPayingToken.withdrawnDividendOf(address)._owner (contracts/Token.sol#980) is not in mixedCase
Parameter DividendPayingToken.accumulativeDividendOf(address)._owner (contracts/Token.sol#986) is not in mixedCase
Constant DividendPayingToken.magnitude (contracts/Token.sol#911) is not in UPPER_CASE_WITH_UNDERSCORES
Parameter DividendTracker.updateMinimumTokenBalanceForDividends(uint256)._newMinimumBalance (contracts/Token.sol#1087) is not in mixedCase
Parameter DividendTracker.getAccount(address)._account (contracts/Token.sol#1132) is not in mixedCase
Parameter SpongebobInu.updateBuyFees(uint256,uint256,uint256)._autoBuybackBurnFeeOnBuy (contracts/Token.sol#1467) is not in mixedCase
Parameter SpongebobInu.updateBuyFees(uint256,uint256,uint256)._marketingFeeOnBuy (contracts/Token.sol#1468) is not in mixedCase
Parameter SpongebobInu.updateBuyFees(uint256,uint256,uint256)._rewardFeeOnBuy (contracts/Token.sol#1469) is not in mixedCase
Parameter SpongebobInu.updateSellFees(uint256,uint256,uint256)._autoBuybackBurnFeeOnSell (contracts/Token.sol#1491) is not in mixedCase
Parameter SpongebobInu.updateSellFees(uint256,uint256,uint256)._marketingFeeOnSell (contracts/Token.sol#1492) is not in mixedCase
Parameter SpongebobInu.updateSellFees(uint256,uint256,uint256)._rewardFeeOnSell (contracts/Token.sol#1493) is not in mixedCase
Parameter SpongebobInu.changeMarketingWallet(address)._marketingWallet (contracts/Token.sol#1517) is not in mixedCase
Parameter SpongebobInu.setSwapEnabled(bool)._enabled (contracts/Token.sol#1683) is not in mixedCase
Variable SpongebobInu.DEAD (contracts/Token.sol#1332) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Variable IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountADesired (contracts/Token.sol#383) is too similar to IUniswapV2Router01.addLiquidit
y(address,address,uint256,uint256,uint256,uint256,address,uint256).amountBDesired (contracts/Token.sol#384)
Variable DividendPayingToken._withdrawDividendOfUser(address)._withdrawableDividend (contracts/Token.sol#948) is too similar to DividendTracker.getAccount(address).withdrawableDividends (contract
s/Token.sol#1140)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar

SpongebobInu.setSwapTokensAtAmount(uint256) (contracts/Token.sol#1706-1713) uses literals with too many digits:
        - require(bool,string)(newAmount > totalSupply() / 100000,SwapTokensAtAmount must be greater than 0.001% of total supply) (contracts/Token.sol#1707-1710)
SpongebobInu.updateGasForProcessing(uint256) (contracts/Token.sol#1748-1759) uses literals with too many digits:
        - require(bool,string)(newValue >= 200000 && newValue <= 500000,gasForProcessing must be between 200,000 and 500,000) (contracts/Token.sol#1749-1752)
SpongebobInu.slitherConstructorVariables() (contracts/Token.sol#1306-1880) uses literals with too many digits:
        - gasForProcessing = 300000 (contracts/Token.sol#1345)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

SafeMathInt.MAX_INT256 (contracts/Token.sol#125) is never used in SafeMathInt (contracts/Token.sol#123-165)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

ERC20.antiBotEnabled (contracts/Token.sol#722) should be constant
ERC20.pinkAntiBot (contracts/Token.sol#721) should be constant
SpongebobInu.DEAD (contracts/Token.sol#1332) should be constant
SpongebobInu.rewardToken (contracts/Token.sol#1344) should be constant
SpongebobInu.swapWithLimit (contracts/Token.sol#1337) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

ERC20.antiBotEnabled (contracts/Token.sol#722) should be immutable
ERC20.pinkAntiBot (contracts/Token.sol#721) should be immutable
SpongebobInu.uniswapV2Pair (contracts/Token.sol#1330) should be immutable
SpongebobInu.uniswapV2Router (contracts/Token.sol#1329) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
```

FUNCTIONAL TESTING

1- Adding liquidity (passed):
https://testnet.bscscan.com/tx/0x7d17dc1b304b328094ca401ab7552afce505d486409ba9b2d1914e693c1a2956

2- Buying when excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0x5c01fa9829716c62d56b66b9a84a40816f31962e41c9e5f21af1944772b669d6

3- Selling when excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0x980e79d1b97cd5e191a32011d6c17714009b0f32e590f27b34a5c39b54800b82

4- Transferring when excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0x0664673a4cea3fda66c8e06a1f09d978d7a7ef770e8e61d5e1b3b145e78c7046

5- Buying when not excluded from fees (10% tax) (passed):
https://testnet.bscscan.com/tx/0x7a461d946faa13000c48fa97b2743cc9e2758100b7303c21f8801e8f4ccfe06f

6- Selling when not excluded from fees (10% tax) (passed):
https://testnet.bscscan.com/tx/0x17000905025ccd8e14ff28200a9f0ab091f534ce8316013d76a7ded76c05cbbe

7- Transferring when not excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0x172c27f46b16caf02ecc3757b0626649a7e183c8ddb7358c5ca7b9950bffd504

FUNCTIONAL TESTING

8- Internal swap **(passed):**
Marketing wallet received BNB
https://testnet.bscscan.com/address/0xf21d472cf93acc209087a7d260404246742dcda0#internaltx

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