



Smart Contract Audit

FOR
Jack Inu
DATED : 12 May 23'



AUDIT SUMMARY

Project name - Jack Inu

Date: 12 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	1
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/0x1fb93a0B846A5fCe52691a7Af0f8165d9478dd2A>



Token Information

Token Name: Jack Inu

Token Symbol: JACK

Decimals: 18

Token Supply: 10,000,000,000

Token Address:

0x303d371f80059C99cCcdd4DfF3A5b45726C6BF9c

Checksum:

42abad87538bce881eb83c4cf83893748c052f6

Owner:

0x09793a14dadE565fCDb4f0EBF3787601e86Bf97f

Deployer:

0x09793a14dadE565fCDb4f0EBF3787601e86Bf97f



TOKEN OVERVIEW

Fees:

Buy Fees: up to 25%

Sell Fees: up to 25%

Transfer Fees: up to 25%

Fees Privilege: Owner

Ownership: Owned

Minting: No mint function

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

Other Privileges: changing fee - including and excluding form fee - changing distribution settings (min tokens to be eligible, cool down 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



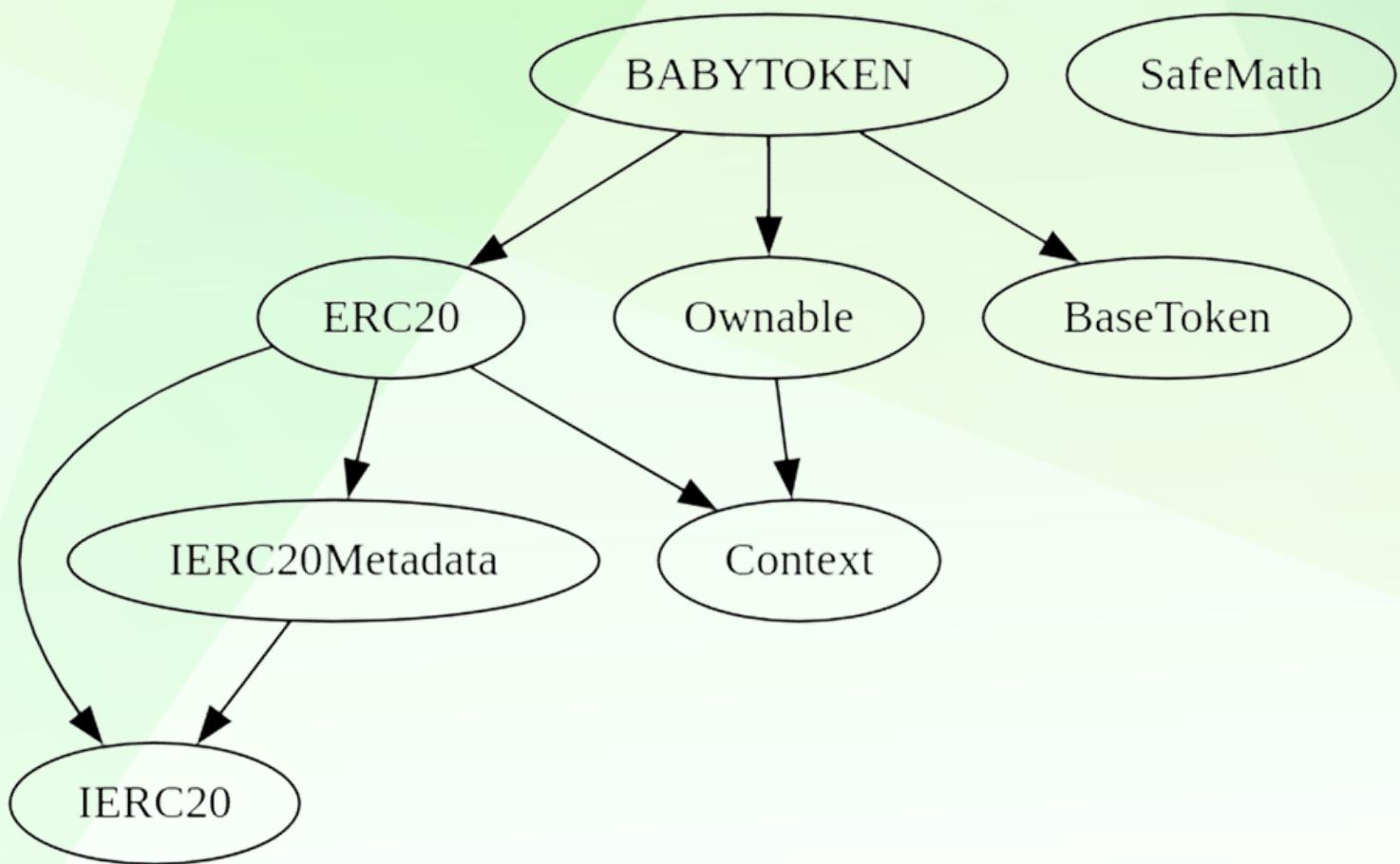
CLASSIFICATION OF RISK

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	0
◆ Gas Optimization / Suggestions	1

INHERITANCE TREE





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



CONTRACT ASSESSMENT

Contract	Type	Bases			
	L	**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 					
	L	<Constructor> Public !	● NO !		



CONTRACT ASSESSMENT

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 !
L feeToSetter External ! NO !
L getPair External ! NO !

CONTRACT ASSESSMENT

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 ! [S [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 ! [S [NO !
L swapTokensForExactETH External ! [● [NO !
L swapExactTokensForETH External ! [● [NO !
L swapETHForExactTokens External ! [S [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 ! [S [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 !



CONTRACT ASSESSMENT

```
|| **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 ! | ● | onlyOwner |
| L | transferOwnership | Public ! | ● | onlyOwner |
| L | _setOwner | Private 🔒 | ● |
```

CONTRACT ASSESSMENT

```
| **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 🔒 | |||
||| |
| **SafeMathUint** | Library | |||
| L | toInt256Safe | Internal 🔒 | |||
||| |
| **IterableMapping** | Library | |||
| L | get | Internal 🔒 | |||
| L | getIndexOfKey | Internal 🔒 | |||
```



CONTRACT ASSESSMENT

```
| 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 |  
| 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 |
```

CONTRACT ASSESSMENT

```
| 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 🔒 | ● | |
| L | swapAndSendToFee | Private 🔒 | ● | |
| L | swapAndLiquify | Private 🔒 | ● | |
| L | swapTokensForEth | Private 🔒 | ● | |
| L | swapTokensForCake | Private 🔒 | ● | |
| L | addLiquidity | Private 🔒 | ● | |
| L | swapAndSendDividends | Private 🔒 | ● | |
```

Legend



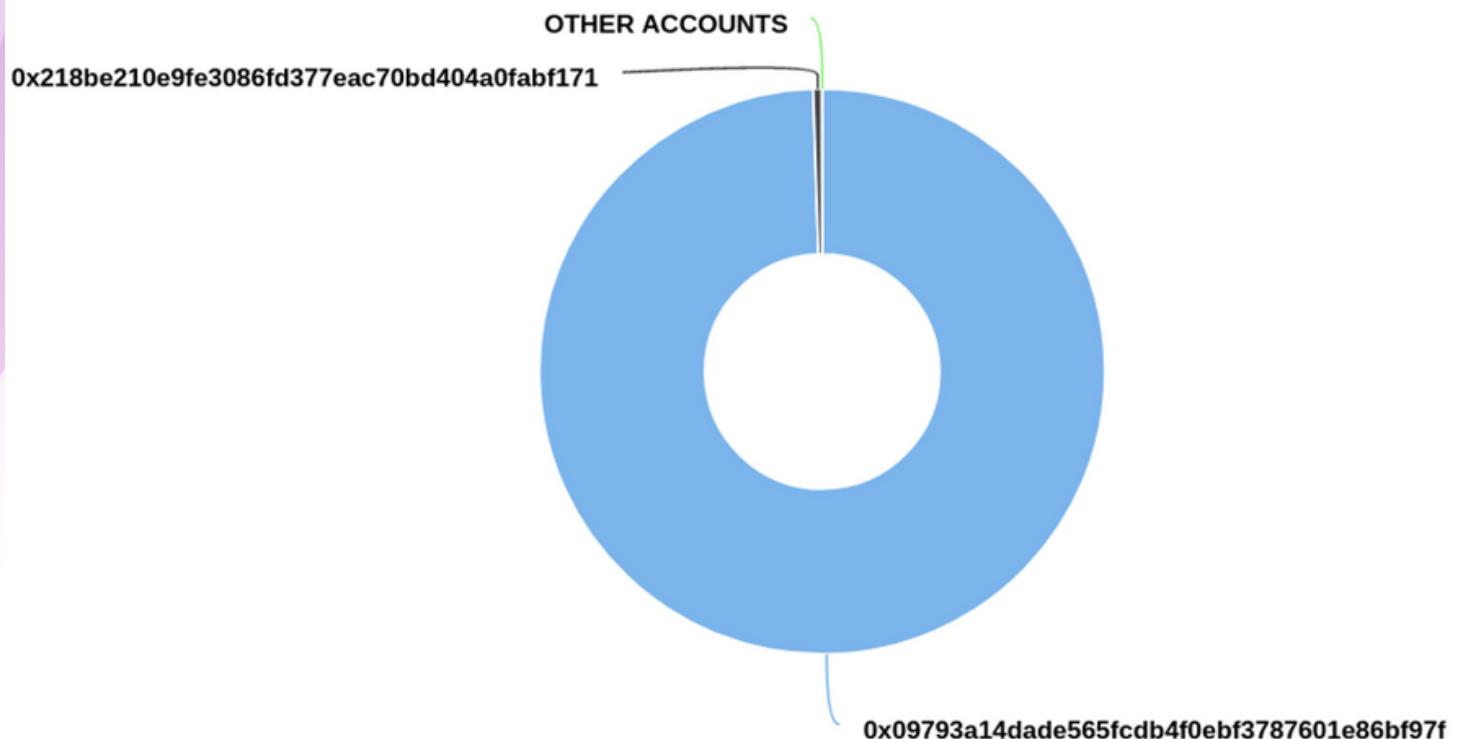
CONTRACT ASSESSMENT

Symbol	Meaning
●	Function can modify state
\\$	Function is payable

HOLDERS AT TIME OF AUDIT

Jack Inu Top 100 Token Holders

Source: BscScan.com





STATIC ANALYSIS

```
Function OwnableUpgradeable._Ownable_init() (contracts/Token.sol#2087-2090) is not in mixedCase
Function OwnableUpgradeable._Ownable_init_unchained() (contracts/Token.sol#2092-2094) is not in mixedCase
Variable OwnableUpgradeable._gap (contracts/Token.sol#2140) is not in mixedCase
Function IUniswapV2Pair.DOMAIN_SEPARATOR() (contracts/Token.sol#2176) is not in mixedCase
Function IUniswapV2Pair.PERMIT_TYPEHASH() (contracts/Token.sol#2178) is not in mixedCase
Function IUniswapV2Pair.MINIMUM_LIQUIDITY() (contracts/Token.sol#2209) is not in mixedCase
Function DividendPayingToken.DividendPayingToken_init(address,string,string) (contracts/Token.sol#2507-2515) is not in mixedCase
Parameter DividendPayingToken._DividendPayingToken_init(address,string,string).rewardToken (contracts/Token.sol#2508) is not in mixedCase
Parameter DividendPayingToken._DividendPayingToken_init(address,string,string)._name (contracts/Token.sol#2509) is not in mixedCase
Parameter DividendPayingToken._DividendPayingToken_init(address,string,string)._symbol (contracts/Token.sol#2510) is not in mixedCase
Parameter DividendPayingToken.dividendOf(address).owner (contracts/Token.sol#2568) is not in mixedCase
Parameter DividendPayingToken.withdrawableDividendOf(address).owner (contracts/Token.sol#2576) is not in mixedCase
Parameter DividendPayingToken.withdrawDividendOf(address).owner (contracts/Token.sol#2585) is not in mixedCase
Parameter DividendPayingToken.accumulateDividendOf(address).owner (contracts/Token.sol#2596) is not in mixedCase
Constant DividendPayingToken.magnitude (contracts/Token.sol#2487) is not in UPPER_CASE_WITH_UNDERSCORES
Parameter BABYTOKENDividendTracker.getAccount(address).account (contracts/Token.sol#2757) is not in mixedCase
Variable BABYTOKEN._marketingWalletAddress (contracts/Token.sol#2982) 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,address,uint256).amountADesired (contracts/Token.sol#1261) is too similar to IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,address,uint256).amountBDesired (contracts/Token.sol#1262)
Variable DividendPayingToken._DividendPayingToken_init(address,string,string).rewardToken (contracts/Token.sol#2508) is too similar to BABYTOKENDividendTracker.initialize(address,uint256).rewardToken (contracts/Token.sol#2699)
Variable DividendPayingToken.withdrawDividendOfUser(address).withdrawableDividends (contracts/Token.sol#2765)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar

Clones.clone(address) (contracts/Token.sol#880-895) uses literals with too many digits:
- mstore(uint256,uint256)(ptr_clone.asm_0,0x3d602d88600a3d3981f363d3d73d3d7300000000000000000000000000000000) (contracts/Token.sol#883-886)
Clones.clone(address) (contracts/Token.sol#880-895) uses literals with too many digits:
- mstore(uint256,uint256)(ptr_clone.asm_0+0x28,0x5af43d82803e903d91602b57fd5bf3800000000000000000000000000000000000000000) (contracts/Token.sol#888-891)
Clones.cloneDeterministic(address,bytes32) (contracts/Token.sol#904-922) uses literals with too many digits:
- mstore(uint256,uint256)(ptr_cloneDeterministic.asm_0,0x3d602d80600a3d3981f3363d3d73d3d3d363d73000000000000000000000000000000000) (contracts/Token.sol#910-913)
Clones.cloneDeterministic(address,bytes32) (contracts/Token.sol#904-922) uses literals with too many digits:
- mstore(uint256,uint256)(ptr_cloneDeterministic.asm_0+0x28,0x5af43d82803e903d91602b57fd5bf3f0000000000000000000000000000000000000000000) (contracts/Token.sol#915-918)
Clones.predictDeterministicAddress(address,bytes32,address) (contracts/Token.sol#927-948) uses literals with too many digits:
- mstore(uint256,uint256)(ptr_predictDeterministicAddress.asm_0,0x3d602d80600a3d3981f3363d3d73d3d363d730000000000000000000000000000000) (contracts/Token.sol#934-937)
Clones.predictDeterministicAddress(address,bytes32,address) (contracts/Token.sol#927-948) uses literals with too many digits:
- mstore(uint256,uint256)(ptr_predictDeterministicAddress.asm_0+0x28,0x5af43d82803e903d91602b57fd5bf3f00000000000000000000000000000000) (contracts/Token.sol#939-942)
BABYTOKEN.constructor(string,string,uint256,address[4],uint256[3],uint256) (contracts/Token.sol#3020-3083) uses literals with too many digits:
- gasForProcessing = 300000 (contracts/Token.sol#3047)
BABYTOKEN.updateGasForProcessing(uint256) (contracts/Token.sol#3156-3167) uses literals with too many digits:
- require(bool,string)(newValue >= 200000 && newValue <= 500000,BABYTOKEN: gasForProcessing must be between 200,000 and 500,000) (contracts/Token.sol#3157-3160)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

SafeMathInt.MAX_INT256 (contracts/Token.sol#2256) is never used in SafeMathInt (contracts/Token.sol#2254-2311)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

BABYTOKEN.dividendTracker (contracts/Token.sol#2971) should be immutable
BABYTOKEN.rewardToken (contracts/Token.sol#2973) should be immutable
BABYTOKEN.uniswapV2Pair (contracts/Token.sol#2967) should be immutable
BABYTOKEN.uniswapV2Router (contracts/Token.sol#2966) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
```

**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/0x50567847829484820dcd2c55b9b29789a49da05b68113685f50f9b260c6352d1>

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

<https://testnet.bscscan.com/tx/0x7520d75487109c3cfa1cdf28fb385acc0c47e9e2e86ee5ac1adbc0299fe35ba8>

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

<https://testnet.bscscan.com/tx/0xe9926a9c3a6ab79fec5f6489cd9adaaf8aeae98e0d9f5a8e6ac4e28e057d46e2>

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

<https://testnet.bscscan.com/tx/0x2d6262235042a17bec181ec9ef5cdad5bc16910bf84552b6545ad3ab51102612>

5- Buying when not excluded from fees (up to 25% tax) (**passed**):

<https://testnet.bscscan.com/tx/0xde536cd3a8e9c63d19a141a247b7519e79ce26cb1014e20542a254be65becef>

6- Selling when not excluded from fees (up to 25% tax) (**passed**):

<https://testnet.bscscan.com/tx/0x1327d499cf31326a9b7a095c255a95b4543747646de8ee4ba605263e3d4f494a>



FUNCTIONAL TESTING

7- Transferring when not excluded from fees (up to 25% tax) (passed):

<https://testnet.bscscan.com/tx/0xe123a720ffcf65a2d9d8debcabcf0e2bec7f29dc1199c6cd6c3d16c968a7f66>

8- Internal swap (passed):

marketing wallet received BUSD

<https://testnet.bscscan.com/address/0xa2da001d772453f7a1d520148663462ebcbd79b4#tokentxns>

9- Reflections (passed):

<https://testnet.bscscan.com/tx/0x1327d499cf31326a9b7a095c255a95b4543747646de8ee4ba605263e3d4f494a>

8- Auto Liquidity (passed):

<https://testnet.bscscan.com/tx/0x1327d499cf31326a9b7a095c255a95b4543747646de8ee4ba605263e3d4f494a>



MANUAL TESTING

Informational – Use of external token

Status: Not Resolved

Overview:

Contract is using 0x92eD61FB8955Cc4e392781cB8b7cD04AADc43D0c (Oggy Inu) as the reward token. This contract is beyond scope of this audit and we assume that this token has enough liquidity and doesn't have major issues that could disable the trades and affect reward system of JACK token.



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>
