



Smart Contract Audit

FOR

Degen Donalds

DATED : 8 APR 23'



AUDIT SUMMARY

Project name – Degen Donalds

Date: 8 April, 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 : Degen Donalds

Symbol : MDD

Decimals: 18

Netowrk: BSC

Token Type: BEP20

Owner:

0xe4477a54140f22D588Efa6C7C93A87826a4C887A
(at time of writing the audit)

Deployer:

0xe4477a54140f22D588Efa6C7C93A87826a4C887A

Token Address :

0xBba7924bd14168A09D5B9A69d0731Fc42A25dA7b



Token Information

Fees:

Buy Fees: up to 25%

Sell Fees: up to 25%

Transfer Fees: up to 25%

Fees Privilege: Owner

Ownership : Owned

Minting: None

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 Limitand Loops | ✓ Timestamp Dependence |
| ✓ Address hardcoded | ✓ Revert/require functions |
| ✓ Exception Disorder | ✓ Use of tx.origin |
| ✓ Using inline assembly | ✓ Integer overflow/underflow |
| ✓ Divide before multiply | ✓ Dangerous strict equalities |
| ✓ Missing Zero Address Validation | ✓ Using SHA3 |
| ✓ Compiler version not fixed | ✓ Using throw |
-



CLASSIFICATION OF RISK

Severity

Description

◆ Critical	These vulnerabilities could be exploited easily and can lead to asset loss, data loss, asset, or data manipulation. They should be fixed right away.
◆ High-Risk	A vulnerability that affects the desired outcome when using a contract, or provides the opportunity to use a contract in an unintended way.
◆ Medium-Risk	A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.
◆ Low-Risk	A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.
◆ Gas Optimization / Suggestion	A vulnerability that has an informational character but is not affecting any of the code.

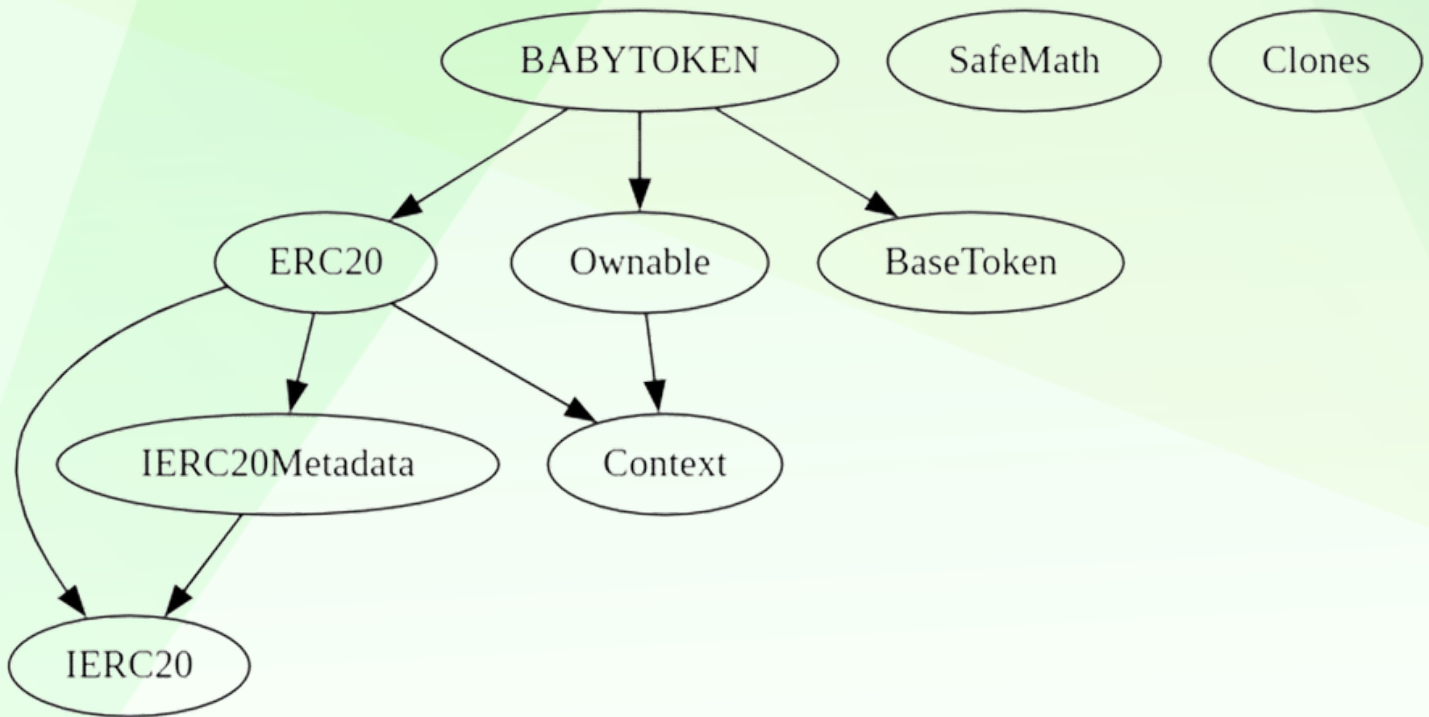
Findings

Severity

Found

◆ Critical	0
◆ High-Risk	0
◆ Medium-Risk	0
◆ Low-Risk	0
◆ Gas Optimization / Suggestions	0

INHERITANCE TREE



POINTS TO NOTE

- Owner is able to change 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 ASSESMENT

Contract	Type	Bases			
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**	
IERC20 Interface					
└	totalSupply	External	!		NO !
└	balanceOf	External	!		NO !
└	transfer	External	!		NO !
└	allowance	External	!		NO !
└	approve	External	!		NO !
└	transferFrom	External	!		NO !
IERC20Metadata Interface IERC20					
└	name	External	!		NO !
└	symbol	External	!		NO !
└	decimals	External	!		NO !
Context Implementation					
└	_msgSender	Internal	🔒		
└	_msgData	Internal	🔒		
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	🔒		
└	_afterTokenTransfer	Internal	🔒		
Ownable Implementation Context					
└	<Constructor>	Public	!		NO !

CONTRACT ASSESMENT

```

└─ owner | Public ! | |NO ! |
└─ renounceOwnership | Public ! | ● | onlyOwner |
└─ transferOwnership | Public ! | ● | onlyOwner |
└─ _setOwner | Private 🔒 | ● | |
|||||
**SafeMath** | Library | |||
└─ tryAdd | Internal 🔒 | | |
└─ trySub | Internal 🔒 | | |
└─ tryMul | Internal 🔒 | | |
└─ tryDiv | Internal 🔒 | | |
└─ tryMod | Internal 🔒 | | |
└─ add | Internal 🔒 | | |
└─ sub | Internal 🔒 | | |
└─ mul | Internal 🔒 | | |
└─ div | Internal 🔒 | | |
└─ mod | Internal 🔒 | | |
└─ sub | Internal 🔒 | | |
└─ div | Internal 🔒 | | |
└─ mod | Internal 🔒 | | |
|||||
**Clones** | Library | |||
└─ clone | Internal 🔒 | ● | |
└─ cloneDeterministic | Internal 🔒 | ● | |
└─ predictDeterministicAddress | Internal 🔒 | | |
└─ predictDeterministicAddress | Internal 🔒 | | |
|||||
**Address** | Library | |||
└─ isContract | Internal 🔒 | | |
└─ sendValue | Internal 🔒 | ● | |
└─ functionCall | Internal 🔒 | ● | |
└─ functionCall | Internal 🔒 | ● | |
└─ functionCallWithValue | Internal 🔒 | ● | |
└─ functionCallWithValue | Internal 🔒 | ● | |
└─ functionStaticCall | Internal 🔒 | | |
└─ functionStaticCall | Internal 🔒 | | |
└─ functionDelegateCall | Internal 🔒 | ● | |
└─ functionDelegateCall | Internal 🔒 | ● | |
└─ verifyCallResult | Internal 🔒 | | |
|||||
**IUniswapV2Factory** | Interface | |||
└─ feeTo | External ! | |NO ! |
└─ feeToSetter | External ! | |NO ! |

```

CONTRACT ASSESMENT

```

└─ getPair | External ! | NO ! |
└─ allPairs | External ! | NO ! |
└─ allPairsLength | External ! | NO ! |
└─ createPair | External ! | ● NO ! |
└─ setFeeTo | External ! | ● NO ! |
└─ setFeeToSetter | External ! | ● NO ! |

```

```

|||||

```

```

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

```

```

|||||

```

```

**IERC20Upgradeable** | Interface | |||
└─ totalSupply | External ! | NO ! |
└─ balanceOf | External ! | NO ! |
└─ transfer | External ! | ● NO ! |
└─ allowance | External ! | NO ! |
└─ approve | External ! | ● NO ! |
└─ transferFrom | External ! | ● NO ! |

```



CONTRACT ASSESMENT

```
||||| |
| **IERC20MetadataUpgradeable** | Interface | IERC20Upgradeable |||
|  | name | External ! | | NO ! |
|  | symbol | External ! | | NO ! |
|  | decimals | External ! | | NO ! |
|||||
| **Initializable** | Implementation | |||
|||||
| **ContextUpgradeable** | Implementation | Initializable |||
|  | __Context_init | Internal 🔒 | ● | initializer |
|  | __Context_init_unchained | Internal 🔒 | ● | initializer |
|  | _msgSender | Internal 🔒 | | |
|  | _msgData | Internal 🔒 | | |
|||||
| **ERC20Upgradeable** | Implementation | Initializable, ContextUpgradeable, IERC20Upgradeable,
IERC20MetadataUpgradeable |||
|  | __ERC20_init | Internal 🔒 | ● | initializer |
|  | __ERC20_init_unchained | Internal 🔒 | ● | initializer |
|  | 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 🔒 | ● | |
|  | _afterTokenTransfer | Internal 🔒 | ● | |
|||||
| **OwnableUpgradeable** | Implementation | Initializable, ContextUpgradeable |||
|  | __Ownable_init | Internal 🔒 | ● | initializer |
|  | __Ownable_init_unchained | Internal 🔒 | ● | initializer |
|  | owner | Public ! | | NO ! |
|  | renounceOwnership | Public ! | ● | onlyOwner |
|  | transferOwnership | Public ! | ● | onlyOwner |
```



CONTRACT ASSESMENT

```
|  | _setOwner | Private | 🔒 | ● | |
|||||
| **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 ! |
|||||
| **SafeMathInt** | Library | |||
|  | mul | Internal | 🔒 | | |
|  | div | Internal | 🔒 | | |
|  | sub | Internal | 🔒 | | |
|  | add | Internal | 🔒 | | |
|  | abs | Internal | 🔒 | | |
|  | toUint256Safe | Internal | 🔒 | | |
|||||
| **SafeMathUint** | Library | |||
|  | toInt256Safe | Internal | 🔒 | | |
|||||
```




CONTRACT ASSESMENT

```
**IterableMapping** | Library | |||
└─ get | Internal 🔒 | ||
└─ getIndexOfKey | Internal 🔒 | ||
└─ getKeyAtIndex | Internal 🔒 | ||
└─ size | Internal 🔒 | ||
└─ set | Internal 🔒 | ● |
└─ remove | Internal 🔒 | ● |
|||||
**DividendPayingTokenInterface** | Interface | |||
└─ dividendOf | External ! | |NO ! |
└─ withdrawDividend | External ! | ● |NO ! |
|||||
**DividendPayingTokenOptionalInterface** | Interface | |||
└─ withdrawableDividendOf | External ! | |NO ! |
└─ withdrawnDividendOf | External ! | |NO ! |
└─ accumulativeDividendOf | External ! | |NO ! |
|||||
**DividendPayingToken** | Implementation | ERC20Upgradeable, OwnableUpgradeable,
DividendPayingTokenInterface, DividendPayingTokenOptionalInterface |||
└─ __DividendPayingToken_init | Internal 🔒 | ● | initializer |
└─ distributeCAKEDividends | 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 🔒 | ● |
|||||
**BABYTOKENDividendTracker** | Implementation | OwnableUpgradeable, DividendPayingToken |||
└─ initialize | External ! | ● | initializer |
└─ _transfer | Internal 🔒 | ||
└─ withdrawDividend | Public ! | |NO ! |
└─ excludeFromDividends | External ! | ● | onlyOwner |
└─ isExcludedFromDividends | Public ! | |NO ! |
└─ updateClaimWait | External ! | ● | onlyOwner |
└─ updateMinimumTokenBalanceForDividends | External ! | ● | onlyOwner |
└─ getLastProcessedIndex | External ! | |NO ! |
└─ getNumberOfTokenHolders | External ! | |NO ! |
```

CONTRACT ASSESMENT

```

└─| getAccount | Public ! | |NO ! |
└─| getAccountAtIndex | Public ! | |NO ! |
└─| canAutoClaim | Private 🔒 | |
└─| setBalance | External ! | ● | onlyOwner |
└─| process | Public ! | ● |NO ! |
└─| processAccount | Public ! | ● | onlyOwner |
|||||
**BaseToken** | Implementation | |||
|||||
**BABYTOKEN** | Implementation | ERC20, Ownable, BaseToken |||
└─| <Constructor> | Public ! | 💰 | ERC20 |
└─| <Receive Ether> | External ! | 💰 |NO ! |
└─| setSwapTokensAtAmount | External ! | ● | onlyOwner |
└─| excludeFromFees | External ! | ● | onlyOwner |
└─| excludeMultipleAccountsFromFees | External ! | ● | onlyOwner |
└─| setMarketingWallet | External ! | ● | onlyOwner |
└─| setTokenRewardsFee | External ! | ● | onlyOwner |
└─| setLiquiditFee | External ! | ● | onlyOwner |
└─| setMarketingFee | External ! | ● | onlyOwner |
└─| _setAutomatedMarketMakerPair | Private 🔒 | ● |
└─| updateGasForProcessing | Public ! | ● | onlyOwner |
└─| updateClaimWait | External ! | ● | onlyOwner |
└─| getClaimWait | External ! | |NO ! |
└─| updateMinimumTokenBalanceForDividends | External ! | ● | onlyOwner |
└─| getMinimumTokenBalanceForDividends | External ! | |NO ! |
└─| getTotalDividendsDistributed | External ! | |NO ! |
└─| isExcludedFromFees | Public ! | |NO ! |
└─| withdrawableDividendOf | Public ! | |NO ! |
└─| dividendTokenBalanceOf | Public ! | |NO ! |
└─| excludeFromDividends | External ! | ● | onlyOwner |
└─| isExcludedFromDividends | Public ! | |NO ! |
└─| getAccountDividendsInfo | External ! | |NO ! |
└─| getAccountDividendsInfoAtIndex | External ! | |NO ! |
└─| processDividendTracker | External ! | ● |NO ! |
└─| claim | External ! | ● |NO ! |
└─| getLastProcessedIndex | External ! | |NO ! |
└─| getNumberOfDividendTokenHolders | External ! | |NO ! |
└─| _transfer | Internal 🔒 | ● |
└─| swapAndSendToFee | Private 🔒 | ● |
└─| swapAndLiquify | Private 🔒 | ● |
└─| swapTokensForEth | Private 🔒 | ● |

```



CONTRACT ASSESMENT

	└		swapTokensForCake		Private	🔒		●		
	└		addLiquidity		Private	🔒		●		
	└		swapAndSendDividends		Private	🔒		●		

Legend

	Symbol		Meaning	
	:-----:		-----	
	●		Function can modify state	
	💰		Function is payable	

[illegible]

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

No issues found



FUNCTIONAL TESTING

Router (PCS V2):

0xD99D1c33F9fC3444f8101754aBC46c52416550D1

1- Adding liquidity (passed):

<https://testnet.bscscan.com/tx/0xa746bfff20a26713775feed3e14ec85d7d974c36c122e901ad27de026d2677d7>

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

<https://testnet.bscscan.com/tx/0x8b5860cb7d205231a4c9c6bb6f6db1ca4f2a0a223710539d7c52140bdce9993b>

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

<https://testnet.bscscan.com/tx/0x4623e9d9d4a5901895a024efedf66d65473303177f8b1a7eb609d1b7d700ce22>

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

<https://testnet.bscscan.com/tx/0x1a2b32dae563093f8cafadb970ed38e9e9aba334a216a55b948bcf9020942e3>

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

<https://testnet.bscscan.com/tx/0x1dba829783bae865780db3252be787f4090d553d13084d7ad5116904bc27c923>

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

<https://testnet.bscscan.com/tx/0x613b1182bb9d3540069f084e4efac4802c6389c1c0288c6c91ef45518bf344ef>

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

<https://testnet.bscscan.com/tx/0x29477f76f9190d05d1e20720fa14bee4495db35cbcd36ceffb89296dbaf9f61>

FUNCTIONAL TESTING

8- Internal swap (passed):

marketing wallet received BUSD

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

9- Reflections (passed):

as seen in this transaction:

<https://testnet.bscscan.com/address/0xe7af7dc357ec9a5cbc5af5c17b7c3a59d721d8c7#tokentxns>

distributor is sending BUSD to holders (auto distribution)

10- Auto Liquidity (passed):

<https://testnet.bscscan.com/token/0xe7af7dc357ec9a5cbc5af5c17b7c3a59d721d8c7?>

```
a=0x0000000000000000000000000000000000000000000000000
```

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>
