RugFreeCoins Audit

Matador Token

Smart Contract Security Audit

June 07, 2021

Contents

Audit details

Audited project

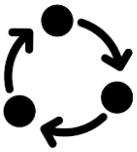
Matador Token (MTDR)

Deployer Address

0x3C969c2EF54E53fFe61d11c241B7273f7E3F1c76

Client contact

Matador Token Team



Blockchain

Binance smart chain

Project website

https://www.matadortoken.com/

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as at the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the disclaimer below – please make sure to read it in full.

DISCLAIMER: By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and Rugfreecoins and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (Rugfreecoins) owe no duty of care towards you or any other person, nor does Rugfreecoins make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and Rugfreecoins hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, Rugfreecoins hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against Rugfreecoins, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report. 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.

Background

Rugfreecoins was commissioned by Matador Token to perform an audit of the smart contract.

https://bscscan.com/address/0x994517e000aa3f117e7ad61b0e2336c76b4fd94a

The focus of this audit is to verify that the smart contract is secure, resilient and working according to the specifications.

The information in this report should be used to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

About the project

Matador Token is a rug-proof community-owned token built on the Binance Smart Chain that rewards holders with a distribution of 20% from every transaction made. Distributions are spread proportionally amongst all holders. The more tokens investors have, the more percentage of the distribution investors receive. In addition to burning a portion of tokens through each transaction. MTDR team will have a monthly randomized burn as circulating supply unlocks!

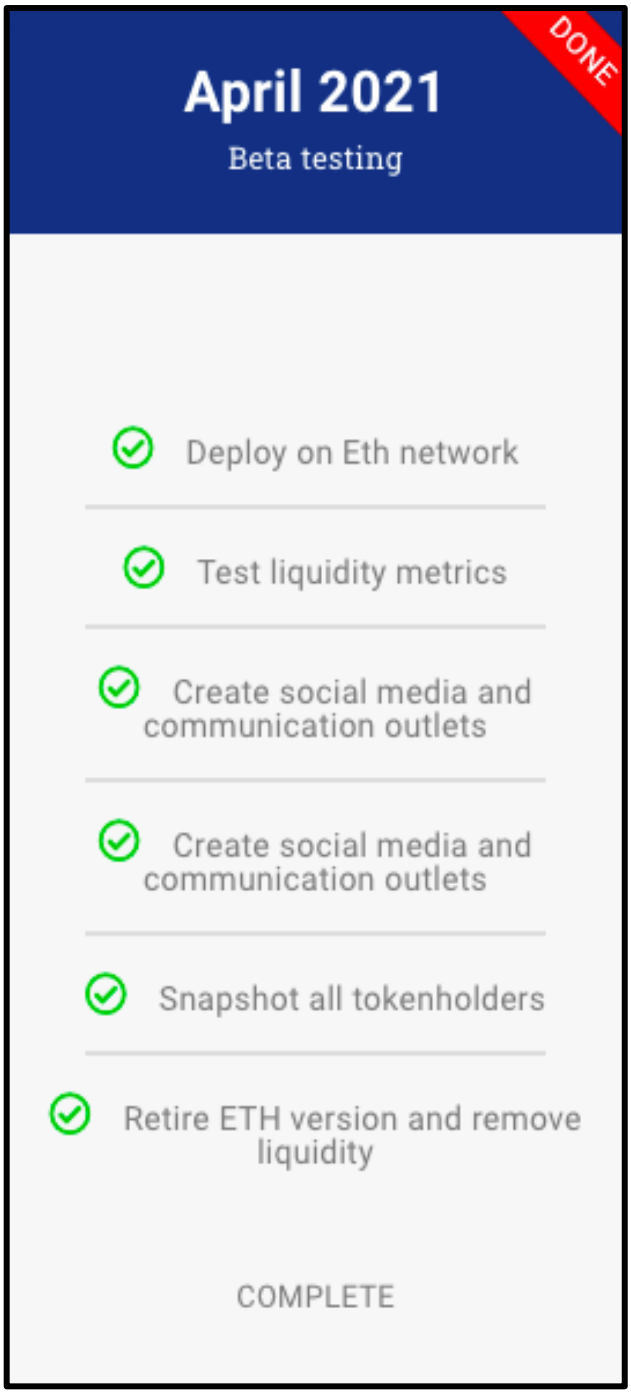
Matador Token applies a 30% fee on all transactions (buys/sells/transfers). Out of this, 20% is automatically distributed to the holders. It's unique to reward long-term holders, and while keeping market makers, whales, and bots out of the token from manipulating the price. The number of tokens in investors' wallets will forever increase as people transact, even though a premium amount was paid to enter the position. This way impairment losses are completely avoided, and investors get paid for just holding Matador.

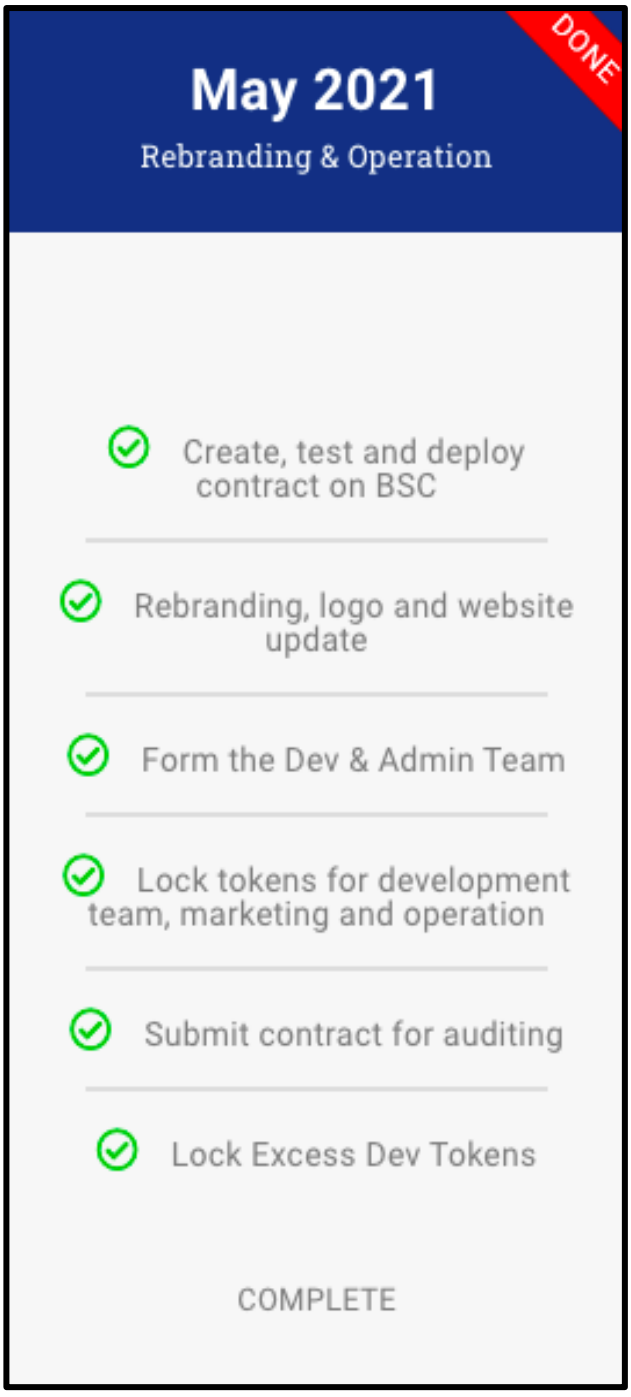
Additionally, out of the 30% Tax, Matador takes 5% of each transaction and automatically adds it to the liquidity pool. The liquidity pool will increase in perpetuity, which in turn leads to a higher price floor for Matador. This is what the team is trying to accomplish: Peace of mind in your long-term investment.

Tokenomics

- 20% goes into HODLer pockets.
- 5% fee is split 50/50, half of which is sold by the contract into BNB, while the other half of the Matador tokens are paired automatically with BNB and added as a liquidity pair on Pancake Swap.
- 5%-coin burn.

Roadmap


DONE
April 2021
Beta testing
Deploy on Eth network
Test liquidity metrics
Create social media and communication outlets
Create social media and communication outlets
Snapshot all tokenholders
Retire ETH version and remove liquidity
COMPLETE


DONE
May 2021
Rebranding & Operation
Create, test and deploy contract on BSC
Rebranding, logo and website update
Form the Dev & Admin Team
Lock tokens for development team, marketing and operation
Submit contract for auditing
Lock Excess Dev Tokens
COMPLETE

QTR3 2021 - 2022

Push on listings & Grow the team

- Coinmarketcap listing
- Coingecko listing
- Apply to multiple exchanges
- Grow our admin team
- Hire a blockchain consultant for continued growth
- Onboard a marketing agency

PENDING

QTR3 2021 - 2022

Push on listings & Grow the team

- Coinmarketcap listing
- Coingecko listing
- Apply to multiple exchanges
- Grow our admin team
- Hire a blockchain consultant for continued growth
- Onboard a marketing agency

PENDING

Target market and the concept

Target market

- Anyone who’s interested in Crypto space with long term investment plans.
- Anyone who’s interested in earning a passive income (20% interest for your investment per each transaction in the Matador ecosystem.)

Core concept

The most unique and important part is Matador token applies a 30% fee on all transactions (buys/sells/transfers). The Out of this 30%, the majority of tokens of 20% is automatically distributed among all the holders. The 30% tax concept is to attract and encourage investors to hold for a long time and make them believe in the project. This way, all the market manipulators, whales, and bots will stay out of the project since they can't make any quick profits.

Additionally, out of the 30% Tax, Matador takes 5% of each transaction and automatically adds it to the liquidity pool. The liquidity pool will increase in perpetuity, which in turn leads to a higher price floor for Matador. The other 5% is automatically getting burnt for every transaction, which that the whole supply is getting reduced every day. Hence, the price of Matador is going up day by day.

Matador Team

1. **Developer: DAN**

   Telegram: @Bookingcrypto

   Matador Wallet: 0xCc55A1B3cFaCA50C403bAb8aFB650599CFdC1a38

2. **Co-Founder/Admin – Operations: Business manager and financier: T.M.**

   Telegram: @Tamatador

   Matador Wallet: 0xB03FFa89A45bA70d31167D9a9090C61435462e67

3. **Sales manager - MIKE M.**

   Telegram: @matador_mike

   Matador Wallet: 0x279a56Fc7f3b3393a2Ea62DfF7210E7650978485

4. **Social media marketer: SCOTT G.**

   Telegram: @ghostmatador

   Matador Wallet: 0x790B817d800B4047786C6ED1166207Fe4Fc04f93

5. **Admin – Marketing: Nic M.**

   Marketing designer and promotions.

6. **Admin – Marketing: Jason**

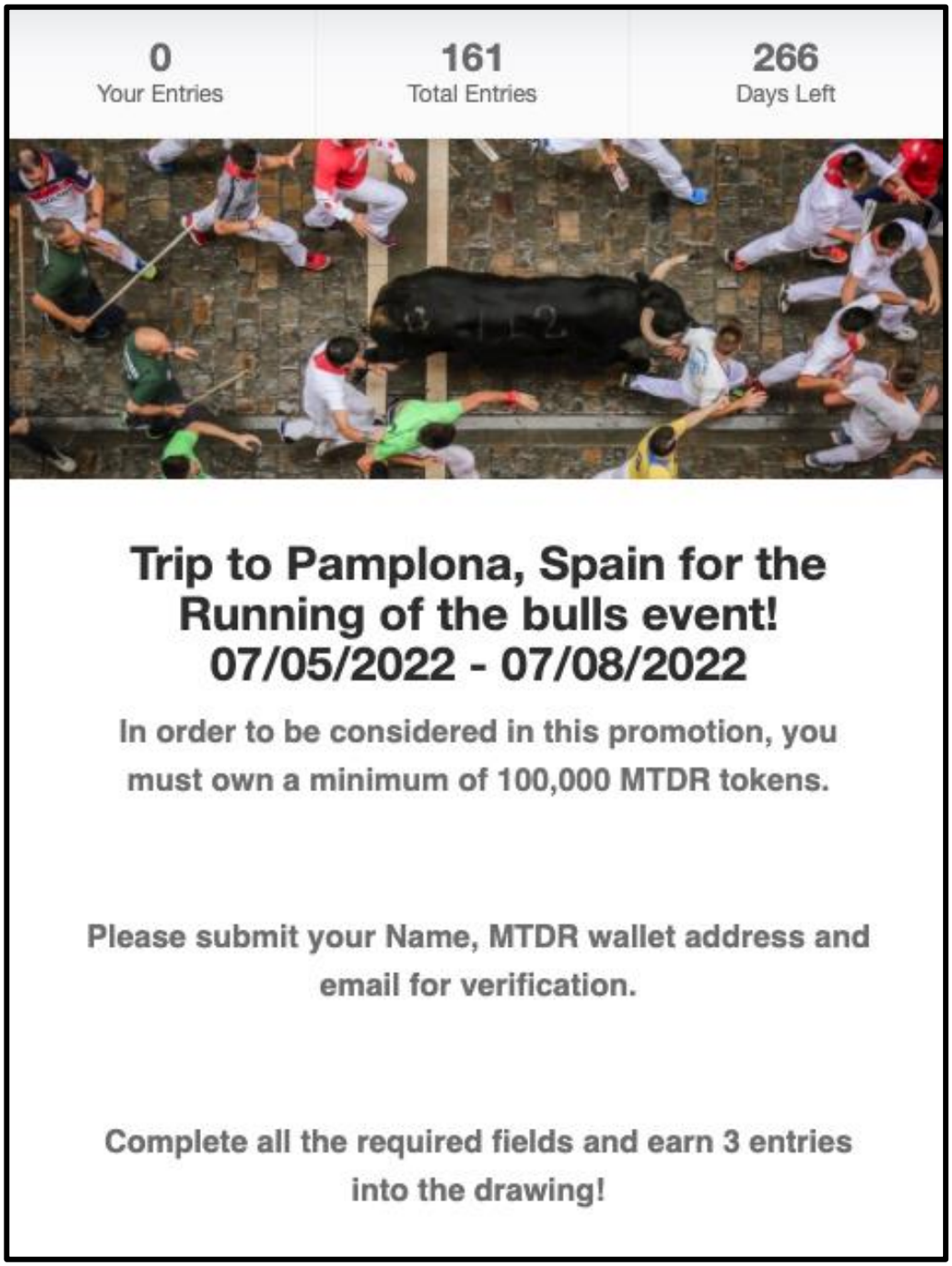
   A marketing genius.

   Telegram: @Jay_Webs

Reward competitions

Reward competitions for the long-term holders.

Competition as of 07th June 2021



Potential to grow with score points

1.	Project efficiency	7/10
2.	Project uniqueness	8/10
3	Information quality	8/10
4	Service quality	7/10
5	System quality	6/10
6	Impact on the community	8/10
7	Impact on the business	7/10
8	Preparing for the future	7/10
Total Points		**7.25/10**

Contract details

Token contract details for 07.06.2021

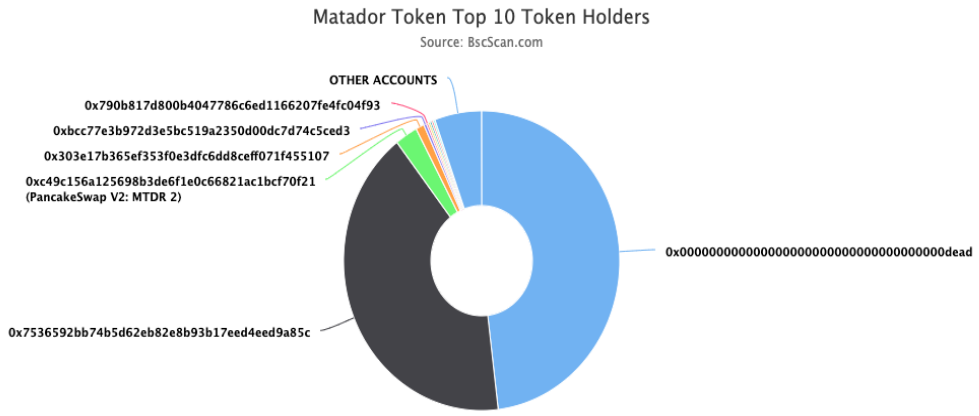
Contract name	Matador Token
Contract address	0x994517e000aa3f117e7ad61b0e2336c76b4fd94a
Token supply	1,000,000,000,000,000 MTDR
Token ticker	MTDR
Decimals	18
Token holders	2431 addresses
Transaction count	7,025
Top 100% holders dominance	97.37%
Contract deployer address	0x3C969c2EF54E53fFe61d11c241B7273f7E3F1c76
Contract's current owner address	0xcc55a1b3cfaca50c403bab8afb650599cfdc1a38

Top token holders

Top 10 Token Holders

The top 10 holders collectively own 94.50% (944,956,586,571,732.00 Tokens) of Matador Token

Token Total Supply: 1,000,000,000,000,000.00 Token | Total Token Holders: 2,417



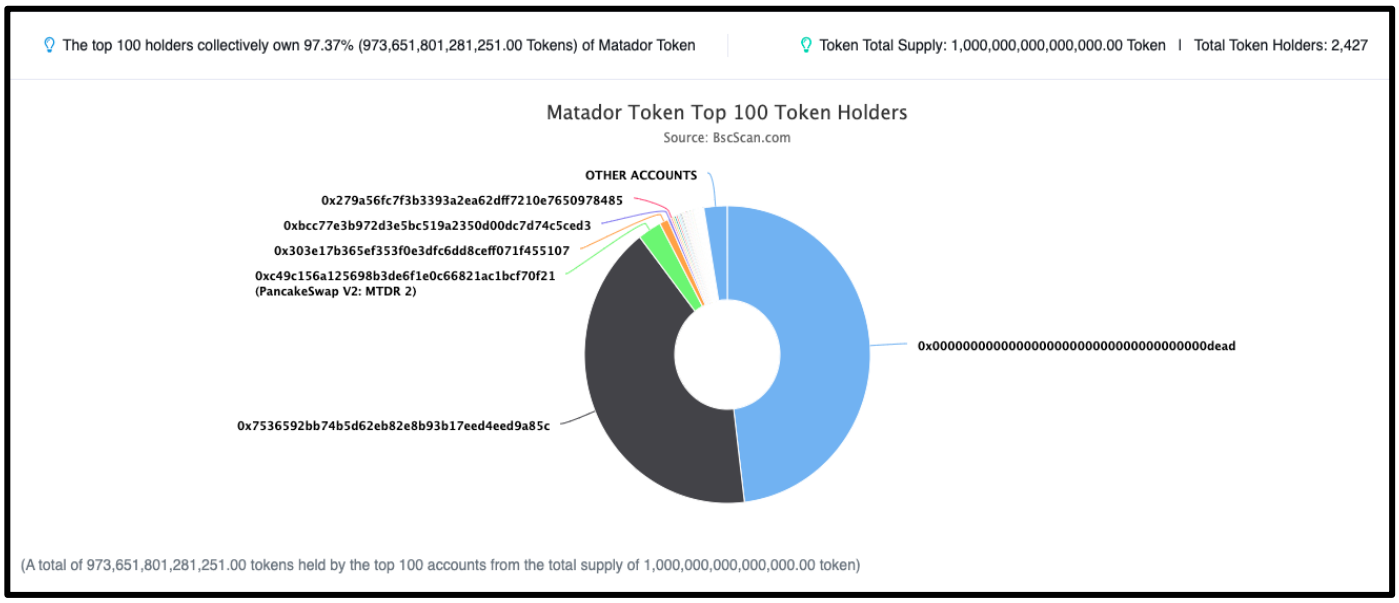
(A total of 944,956,586,571,732.00 tokens held by the top 10 accounts from the total supply of 1,000,000,000,000,000.00 token)

(A total of 945,549,847,495,503.00 tokens held by the top 10 accounts from the total supply of 1,000,000,000,000,000.00 token)

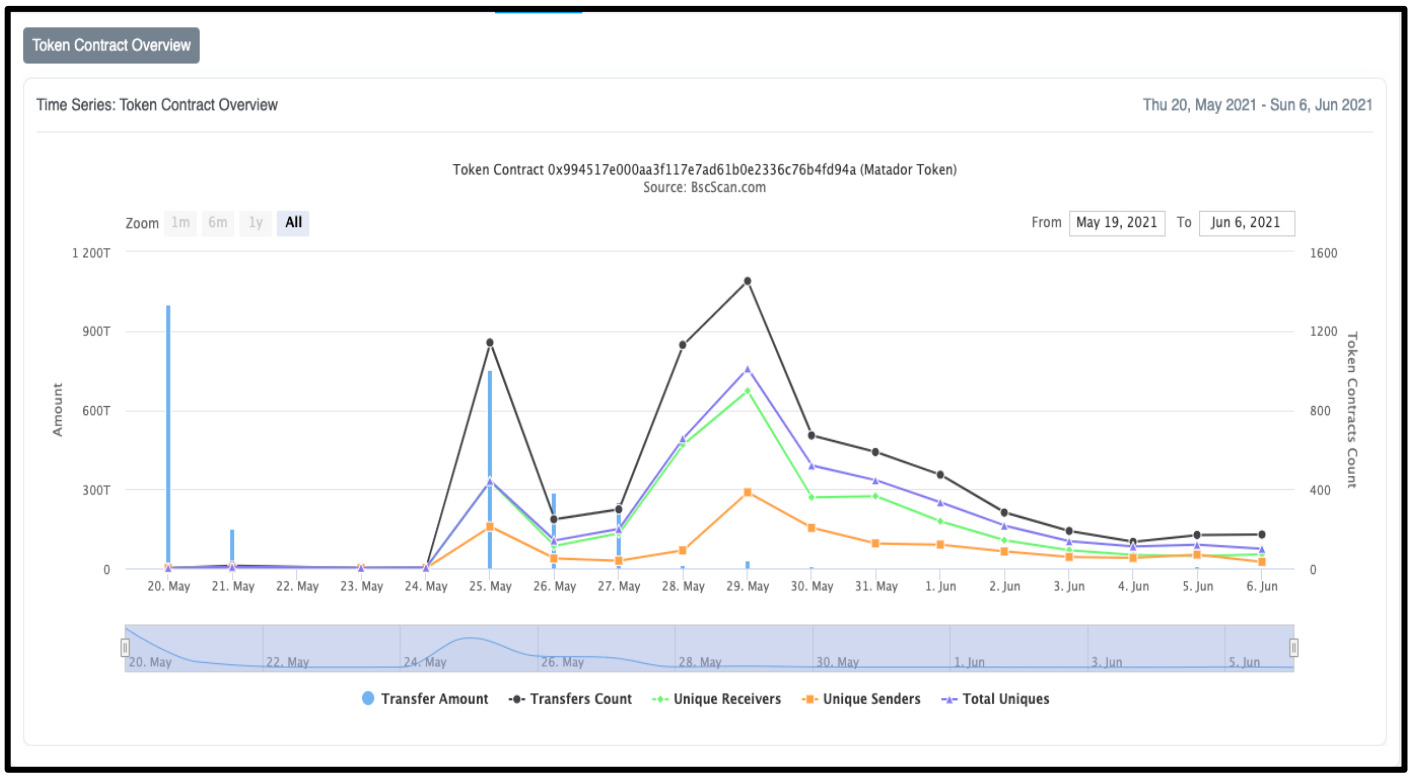
Rank	Address	Quantity (Token)	Percentage
1	0x0000000000000000000000000000000000000dead	480,856,222,293,553.076427040978378323	48.0856%
2	0x7536592bb74b5d62eb82e8b93b17eed4eed9a85c	414,148,005,302,913.746206489284999857	41.4148%
3	PancakeSwap V2: MTDR 2	26,808,287,782,214.823599656654394296	2.6808%
4	0x303e17b365ef353f0e3dfc6dd8ceff071f455107	9,906,281,690,004.719961802977394325	0.9906%
5	0xbcc77e3b972d3e5bc519a2350d00dc7d74c5ced3	3,205,066,350,242.510680383437067348	0.3205%
6	0x279a56fc7f3b3393a2ea62dff7210e7650978485	2,209,159,197,501.887017462177056603	0.2209%
7	0x790b817d800b4047786c6ed1166207fe4fc04f93	2,207,756,093,368.886997078623426408	0.2208%
8	0xb03ffa89a45ba70d31167d9a9090c61435462e67	2,203,592,503,514.709845065149862042	0.2204%
9	0x5beba542432c6dd9a30b9f95a5fd51307c30a7f5	2,018,752,921,299.363578050698348398	0.2019%
10	0xc7029e939075f48fa2d5953381660c7d01570171	1,986,723,360,888.369644566172292858	0.1987%

Token distribution

Top 100 Token Holders



Contract interaction details



Contract code function details

No	Category	Item	Result
1	Coding conventions	BRC20 Token standards	pass
		compile errors	pass
		Compiler version security	pass
		visibility specifiers	pass
		Gas consumption	low issue
		SafeMath features	pass
		Fallback usage	pass
		tx.origin usage	pass
		deprecated items	pass
		Redundant code	pass
		Overriding variables	pass
2	Function call audit	Authorization of function call	pass
		Low level function (call/delegate call) security	pass
		Returned value security	pass
		Selfdestruct function security	pass
3	Business security	Access control of owners	pass
		Business logics	pass
		Business implementations	pass
4	Integer overflow/underflow		pass
5	Reentrancy		pass
6	Exceptional reachable state		pass
7	Transaction ordering dependence		pass
8	Block properties dependence		pass
9	Pseudo random number generator (PRNG)		pass
10	DoS (Denial of Service)		pass
11	Token vesting implementation		pass
12	Fake deposit		pass
13	Event security		pass

Below table represents the summary of the contracts and methods in the token contract. We scanned the whole contract and listed down all the Interfaces, functions and implementations with its visibility and mutability.

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❗
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 🔒		
Context	**Implementation**			
└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		
Address	**Library**			
└	isContract	Internal 🔒		
└	sendValue	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	

└	functionCallWithValue	Internal 🔒	🛑	
└	functionStaticCall	Internal 🔒		
└	functionStaticCall	Internal 🔒		
└	functionDelegateCall	Internal 🔒	🛑	
└	functionDelegateCall	Internal 🔒	🛑	
└	_verifyCallResult	Private 🔐		
Ownable	**Implementation**	**Context**		
└		Public ❗	🛑	NO❗
└	owner	Public ❗		NO❗
└	renounceOwnership	Public ❗	🛑	onlyOwner
└	transferOwnership	Public ❗	🛑	onlyOwner
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❗

IUniswapV2Router01	Interface			
└	factory	External ❗		NO❗
└	WETH	External ❗		NO❗
└	addLiquidity	External ❗	🔘	NO❗
└	addLiquidityETH	External ❗	💵	NO❗
└	removeLiquidity	External ❗	🔘	NO❗
└	removeLiquidity ETH	External ❗	🔘	NO❗
└	removeLiquidity WithPermit	External ❗	🔘	NO❗
└	removeLiquidity ETHWithPermit	External ❗	🔘	NO❗
└	swapExactToke nsForTokens	External ❗	🔘	NO❗
└	swapTokensFor ExactTokens	External ❗	🔘	NO❗
└	swapExactETHF orTokens	External ❗	💵	NO❗
└	swapTokensFor ExactETH	External ❗	🔘	NO❗
└	swapExactToke nsForETH	External ❗	🔘	NO❗
└	swapETHForExa ctTokens	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❗

MTDR	Implementation	Context, IERC20, Ownable		
└		Public ❗	🔘	NO❗
└	name	External ❗		NO❗
└	symbol	External ❗		NO❗
└	decimals	External ❗		NO❗
└	totalSupply	External ❗		NO❗
└	balanceOf	Public ❗		NO❗
└	transfer	External ❗	🔘	NO❗
└	allowance	External ❗		NO❗
└	approve	External ❗	🔘	NO❗
└	transferFrom	External ❗	🔘	NO❗
└	increaseAllowance	External ❗	🔘	NO❗
└	decreaseAllowance	External ❗	🔘	NO❗
└	isExcludedFromReward	External ❗		NO❗
└	totalFees	External ❗		NO❗
└	deliver	External ❗	🔘	NO❗

└	reflectionFromToken	External ❗		NO❗
└	tokenFromReflection	Public ❗		NO❗
└	excludeFromReward	External ❗	🔘	onlyOwner
└	includeInReward	External ❗	🔘	onlyOwner
└	_transferBothExcluded	Private 🔐	🔘	
└	excludeFromFee	External ❗	🔘	onlyOwner
└	includeInFee	External ❗	🔘	onlyOwner
└	setTaxFeePercent	External ❗	🔘	onlyOwner
└	setBurnFeePercent	External ❗	🔘	onlyOwner
└	setLiquidityFeePercent	External ❗	🔘	onlyOwner
└	setMaxTxPercent	External ❗	🔘	onlyOwner
└	setSwapAndLiquifyEnabled	External ❗	🔘	onlyOwner
└		External ❗	💵	NO❗
└	_reflectFee	Private 🔐	🔘	
└	_getValues	Private 🔐		

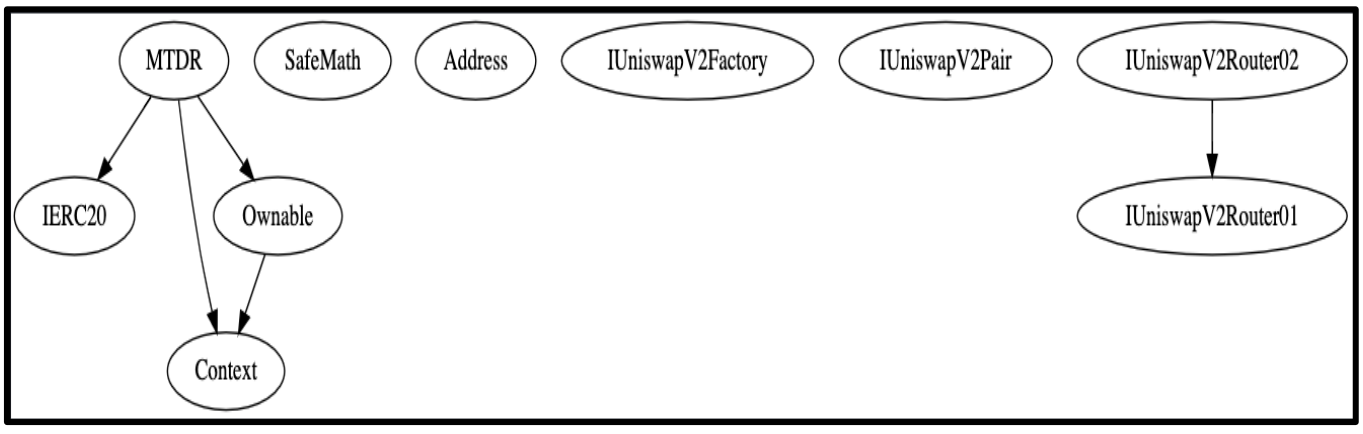
└	_getTValues	Private 🔐		
└	_getRValues	Private 🔐		
└	_getRate	Private 🔐		
└	_getCurrentSupply	Private 🔐		
└	_takeLiquidity	Private 🔐	🛑	
└	_takeBurn	Private 🔐	🛑	
└	calculateTaxFee	Private 🔐		
└	calculateBurnFee	Private 🔐		
└	calculateLiquidityFee	Private 🔐		
└	removeAllFee	Private 🔐	🛑	
└	restoreAllFee	Private 🔐	🛑	
└	isExcludedFromFee	External ❗		NO❗
└	_approve	Private 🔐	🛑	
└	_transfer	Private 🔐	🛑	
└	swapAndLiquify	Private 🔐	🛑	lockTheSwap
└	swapTokensForEth	Private 🔐	🛑	

└	addLiquidity	Private		
└	_tokenTransfer	Private		
└	_transferStandard	Private		
└	_transferToExcluded	Private		
└	_transferFromExcluded	Private		

Legend

Symbol	Meaning
	Function can modify state
	Function is payable

Inheritance Hierarchy



Security issue checking status

- **High severity issues**
  No high severity issues found

- **Medium severity issues**
  No medium severity issues found

- **Low severity issues**

1. Out of gas

Issue:

- The function includeInReward() uses the loop to find and remove addresses from the _excluded list. Function will be aborted with OUT_OF_GAS exception if there will be a long excluded addresses list.

```
function includeInReward(address account) external onlyOwner() {
    require(_isExcluded[account], "Account is already included");
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (_excluded[i] == account) {
            _excluded[i] = _excluded[_excluded.length - 1];
            _tOwned[account] = 0;
            _isExcluded[account] = false;
            _excluded.pop();
            break;
        }
    }
}
```

- The function _getCurrentSupply also uses the loop for evaluating total supply. It also could be aborted with OUT_OF_GAS exception if there will be a long excluded addresses list.

```
}
function _getCurrentSupply() private view returns(uint256, uint256) {
    uint256 rSupply = _rTotal;
    uint256 tSupply = _tTotal;
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (_rOwned[_excluded[i]] > rSupply || _tOwned[_excluded[i]] > tSupply) return (_rTotal, _tTotal);
        rSupply = rSupply.sub(_rOwned[_excluded[i]]);
        tSupply = tSupply.sub(_tOwned[_excluded[i]]);
    }
    if (rSupply < _rTotal.div(_tTotal)) return (_rTotal, _tTotal);
    return (rSupply, tSupply);
}
```

Recommendation:
Check that the excluded array length is not too big.

Owner privileges

(In the period when the owner is not renounced)

- Owner can exclude from the fee

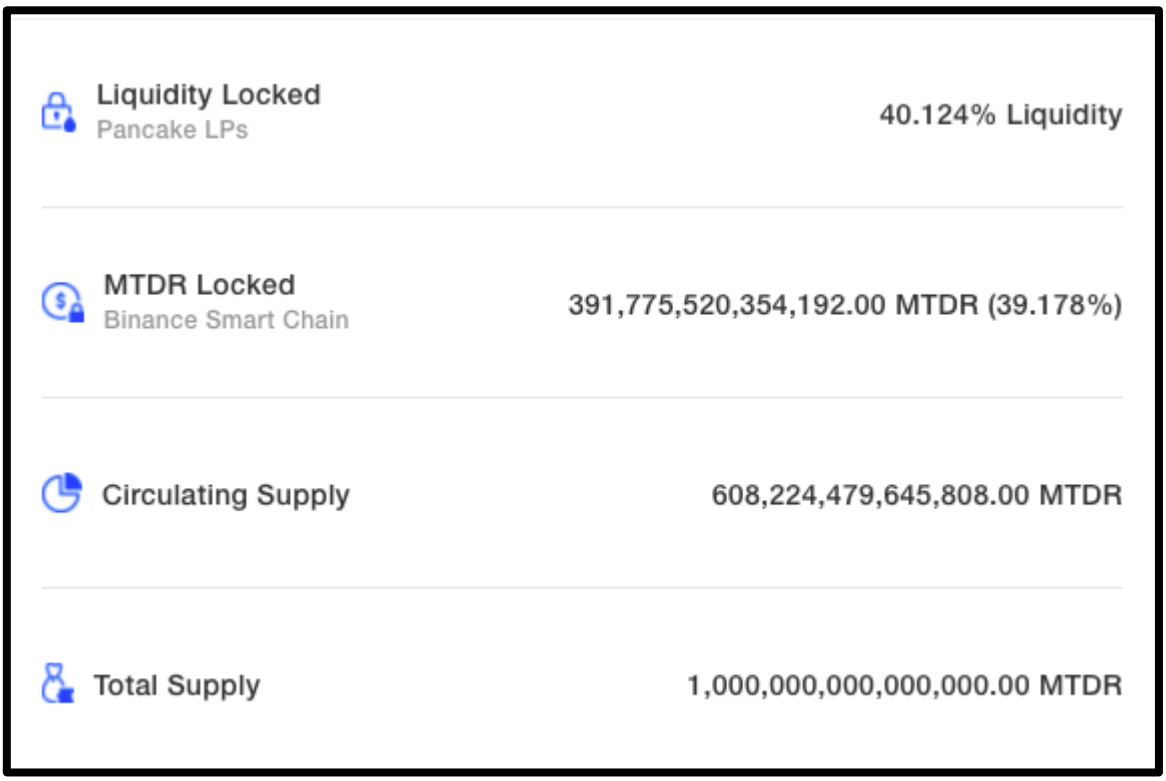
```
ftrace | funcSig
function excludeFromFee(address account) public onlyOwner {
    _isExcludedFromFee[account] = true;
}

ftrace | funcSig
```

- Owner can transfer the ownership

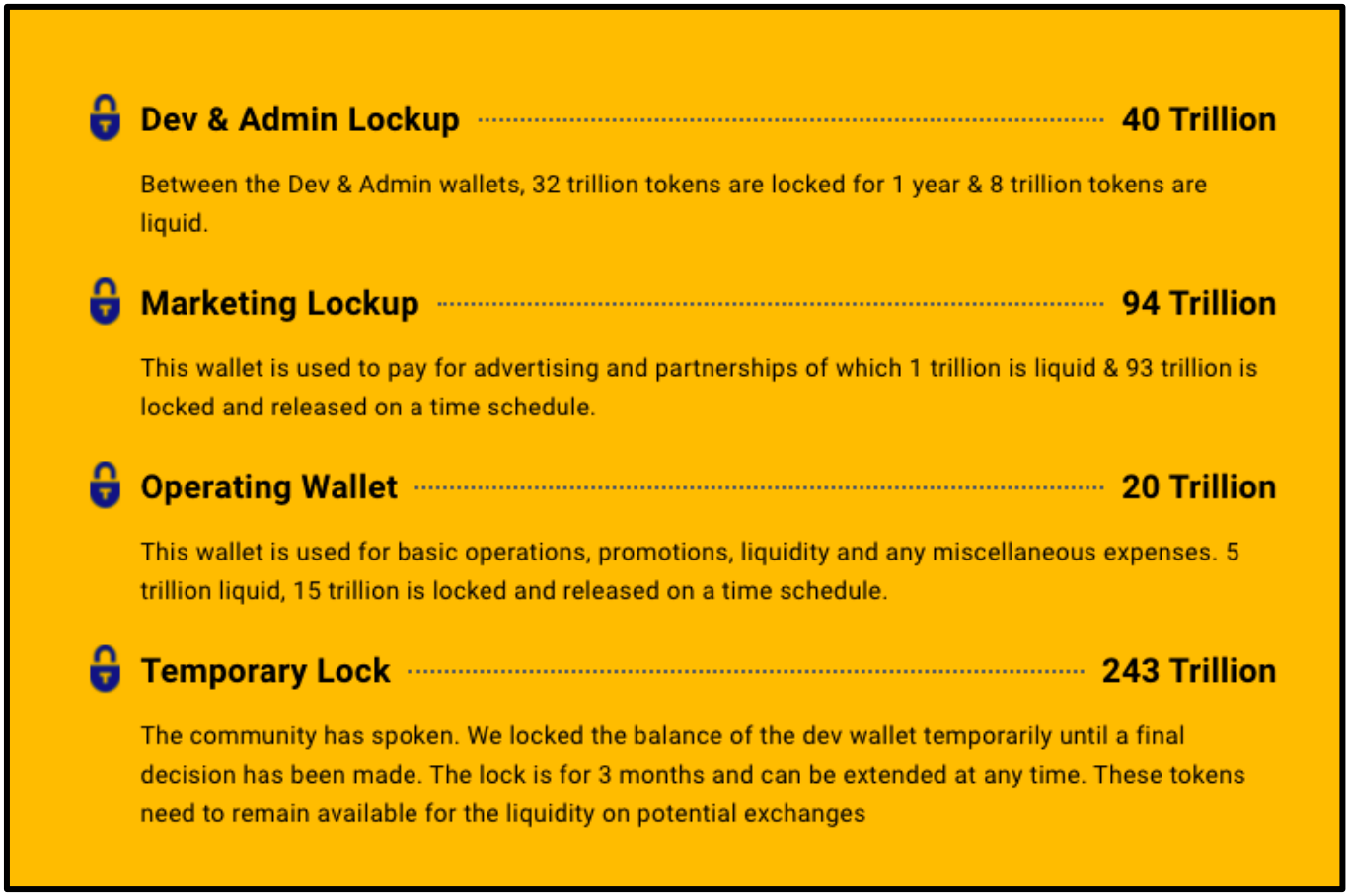
```
abstract contract Ownable is Context {
    address private _owner;
    event OwnershipTransferred(address indexed previousOwner, address indexed newOwner);
    constructor () {
        _owner = 0xCc55A1B3cFaCA50C403bAb8aFB650599CFdC1a38;
        emit OwnershipTransferred(address(0), _owner);
    }
    function owner() public view virtual returns (address) {
        return _owner;
    }
    modifier onlyOwner() {
        require(owner() == _msgSender(), "Ownable: caller is not the owner");
        _;
    }
    function renounceOwnership() public virtual onlyOwner {
        emit OwnershipTransferred(_owner, address(0));
        _owner = address(0);
    }
    function transferOwnership(address newOwner) public virtual onlyOwner {
        require(newOwner != address(0), "Ownable: new owner is the zero address");
        emit OwnershipTransferred(_owner, newOwner);
        _owner = newOwner;
    }
}
```

Audit conclusion

While conducting the audit of the Matador Token smart contract, it was observed that there is nothing alarming with the code and the contract contains only has low severity issues.



Breakdown of token lockup



Dev & admin team lockup details

- **Dan's wallet lockup**

https://bscscan.com/tx/0x947066d5088421d1a8f95d3b028bf439edd235aaf765da5abe2917c21feb6bdb

- **T. M'S wallet lockup**

https://bscscan.com/tx/0x947066d5088421d1a8f95d3b028bf439edd235aaf765da5abe2917c21feb6bdb

- **Mike's wallet lockup**

https://bscscan.com/tx/0x947066d5088421d1a8f95d3b028bf439edd235aaf765da5abe2917c21feb6bdb

- **Scott's wallet lockup**

https://bscscan.com/tx/0x947066d5088421d1a8f95d3b028bf439edd235aaf765da5abe2917c21feb6bdb

- **Marketing wallet**

https://bscscan.com/tx/0x947066d5088421d1a8f95d3b028bf439edd235aaf765da5abe2917c21feb6bdb

- **Operations wallet**

https://bscscan.com/tx/0x947066d5088421d1a8f95d3b028bf439edd235aaf765da5abe2917c21feb6bdb