



RugFreeCoins Audit



Brave Doge Token Audit
Smart Contract Security Audit
August 25, 2021

Contents

Audit details	1
Disclaimer	2
Background	3
About the project	4
Target market and the concept	7
Potential to grow with score points	9
Total Points	9
Contract details	10
Token distribution	11
Contract code function details	12
Contract description table	13
Security issue checking status	17
Owner privileges	19
Audit conclusion	21

Audit details



Audited project

Brave Doge Token



Contract Address

0x07510cd60d7e0ccf5e23f70c8dc4fdb1c150b54



Client contact

Brave Doge Token Team



Blockchain

Binance smart chain



Project website

<https://bravedoge.io/>

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 Brave Doge to perform an audit of the smart contract.

<https://bscscan.com/address/0x07510cd60d7e0ccf5e23f70c8dc4fdfb1c150b54>

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, project feasibility, long term sustainability and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

About the project

Brave Doge is a token built on the Binance Smart Chain, and a 100% decentralized community token, investors have full control of their investments. Each transaction, purchase and sales incur a 12% fee.

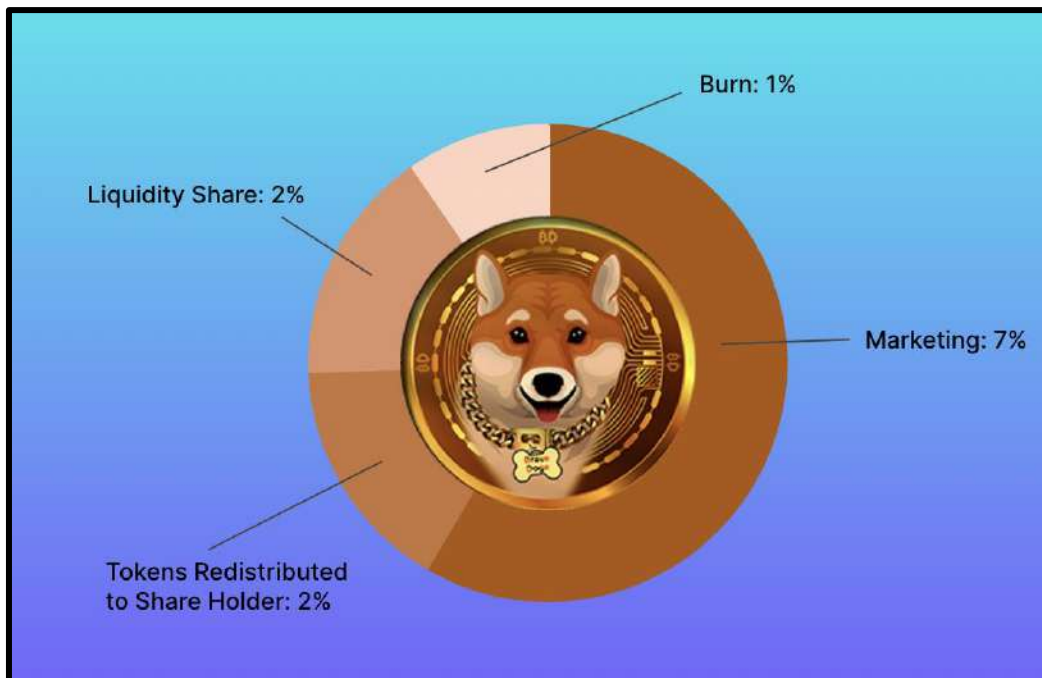
Features

- ❖ The **automatic Brave Doge token rewards of 2% when buying and selling** is what investors will be benefited in the long term, where tokens will be distributed among every holder proportional to how many tokens each individual holds.
- ❖ The **sustainability fee of 7% marketing** is what allows Brave Doge to hold the aforementioned promise. Tokens will be swapped into BNBs and will be sent to a marketing wallet per transaction. This way, Brave Doge will have enough funds to promote the coin and spend for future development without selling tokens as the traditional way.
- ❖ The liquidity fee of 2%, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity. This is a key element for decentralized exchanges like Pancakeswap.
- ❖ The fee of 1% from each trade will be sent to a burn wallet, which will enable to keep the market price and the marketcap in balance.

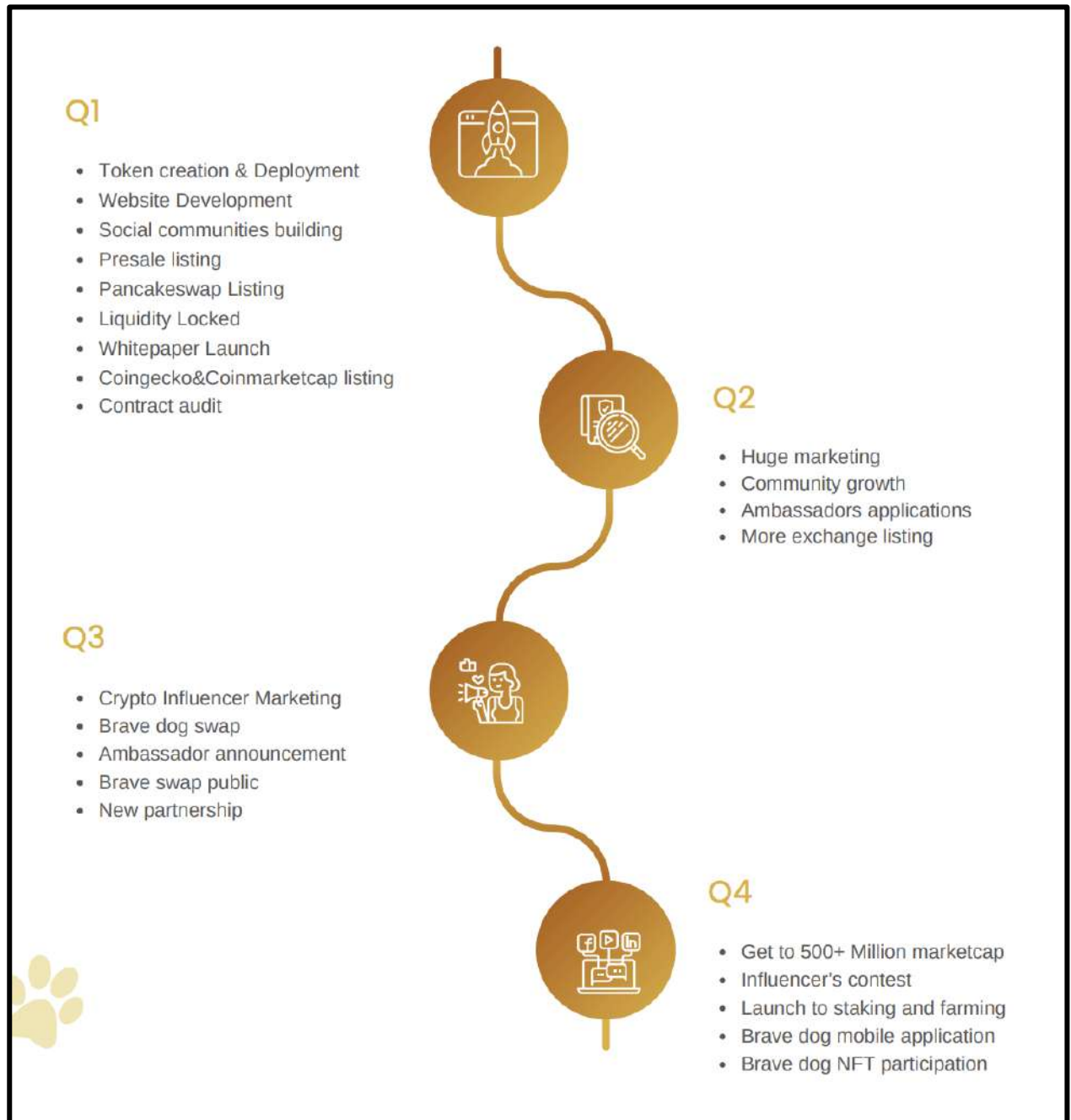
Tokenomics

12% tax fee when buying and selling

- ❖ 2% of trade goes to holders pockets in tokens.
- ❖ 2% of trade goes to the liquidity pool.
- ❖ 7% of trade goes to marketing wallet in BNB.
- ❖ 1% of trade goes to the burn wallet.



Roadmap



Target market and the concept

Target market

- ❖ Anyone who's interested in Crypto space with long term investment plans.
- ❖ Anyone who's ready to earn a passive income in BNB by holding tokens.
- ❖ Anyone who's interested in trading tokens.
- ❖ Anyone who's interested in taking part with Brave Doge future plans.
- ❖ Anyone who's interested in making financial transactions with any other party using Brave Doge as the currency.

Core concept

The reward system

2% of each transaction when buying and selling gets sent amongst all holders in tokens. The holders will be eligible to receive tokens, whenever a transaction occurs, and rewards are proportional to how many tokens each individual holds.

Sustainable mechanism

The liquidity fee of 2%, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.

The **fee of 7% marketing** is what allows Brave Doge to promote the token and use funds to further development of the platform. Tokens will be swapped into BNBs and will be sent to a marketing wallet per transaction. This way, Brave Doge will have access to the funds without selling tokens as the traditional way, which will enable them to consume funds without hurting the project.

The **fee of 1% burn** that benefits and rewards those who invest long-term. This feature slowly reduces supply making Brave Doge price more and more valuable.

Brave Doge – Future plan

- ❖ Influencer's contest
- ❖ Launch to staking and farming
- ❖ Brave dog mobile application
- ❖ Brave dog NFT participation

Brave Doge swap

BraveDoge Swap is the official decentralized exchange (DEX). It gives holders the ability to swap any BEP20 token for another one. It is powered by Pancake Swap, the safest and most trusted DEX in the world. Furthermore, it acts as the foundation from which we will build additional features and functionality exclusive to BraveDoge holders.



Potential to grow with score points

1.	Project efficiency	9/10
2.	Project uniqueness	7/10
3	Information quality	9/10
4	Service quality	9/10
5	System quality	8/10
6	Impact on the community	9/10
7	Impact on the business	9/10
8	Preparing for the future	9/10
Total Points		8.63/10

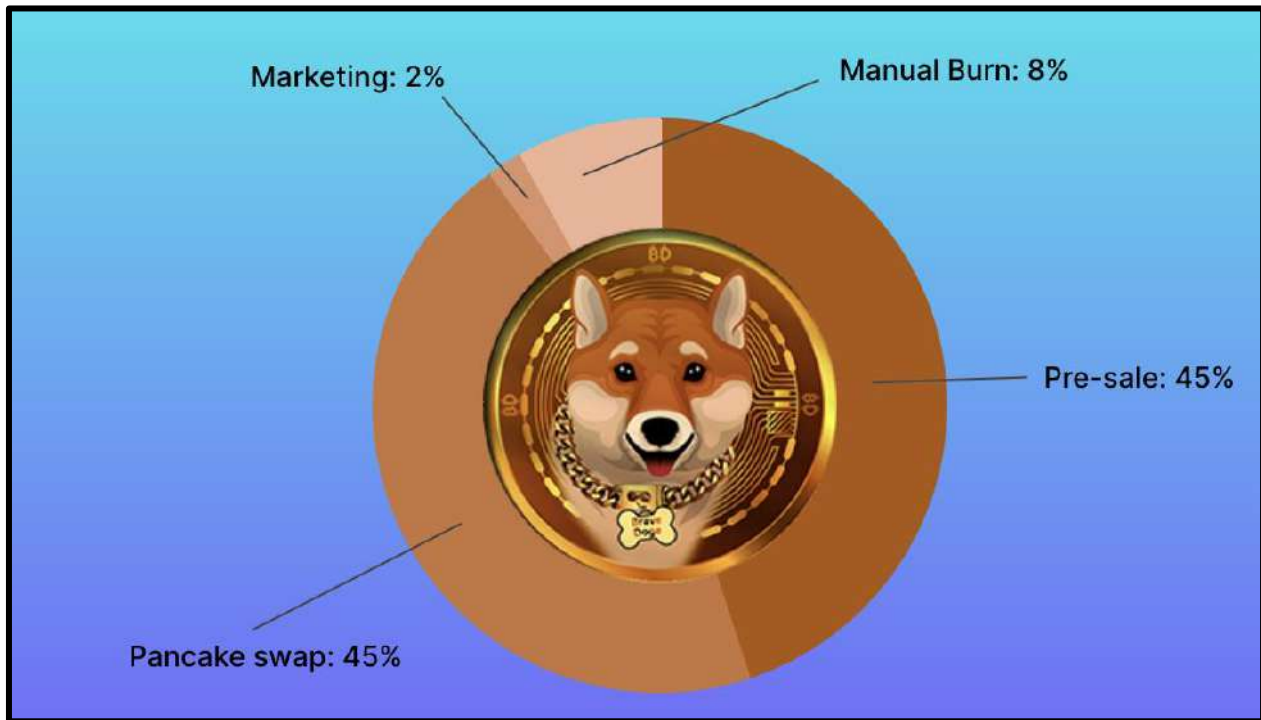
Contract details

Token contract details for 25th August 2021

Contract name	Brave Doge Token
Contract address	0x07510cd60d7e0ccf5e23f70c8dc4fdb1c150b54
Token supply	100,000,000,000,000
Token ticker	Brave
Decimals	9
Token holders	1
Transaction count	0
Marketing address	0x7d2F39032e0a2582239e57b6BD77358FF7D5f6b0
Contract deployer address	0x1Df925D52a7dade6411Cf669fa796F1e62bBc509
Contract's current owner address	0x6ed54c59f74cf8b03634f64dfa41583c00996bfc

Token distribution

Tokens are distributed as follows:





































































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	pass
		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

Contract description table

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		
L	Function Name	Visibility	Mutability	Modifiers
Context	Implementation			
L	_msgSender	Internal 		
L	_msgData	Internal 		
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 
SafeMath	Library			
L	add	Internal 		
L	sub	Internal 		
L	sub	Internal 		
L	mul	Internal 		
L	div	Internal 		
L	div	Internal 		
L	mod	Internal 		
L	mod	Internal 		
L	ceil	Internal 		

Address	Library			
L	isContract	Internal 		
L	sendValue	Internal 		
L	functionCall	Internal 		
L	functionCall	Internal 		
L	functionCallWithValue	Internal 		
L	functionCallWithValue	Internal 		
L	_functionCallWithValue	Private 		
Ownable	Implementation	Context		
L		Internal 		
L	owner	Public 		NO 
L	renounceOwnership	Public 		onlyOwner
L	transferOwnership	Public 		onlyOwner
IUniswapV2Factory	Interface			
L	createPair	External 		NO 
IUniswapV2Pair	Interface			
L	sync	External 		NO 
IUniswapV2Router01	Interface			
L	factory	External 		NO 
L	WETH	External 		NO 
L	addLiquidity	External 		NO 
L	addLiquidityETH	External 		NO 
IUniswapV2Router02	Interface	IUniswapV2Router01		
L	removeLiquidityETHSupportingFeeOnTransferTokens	External 		NO 

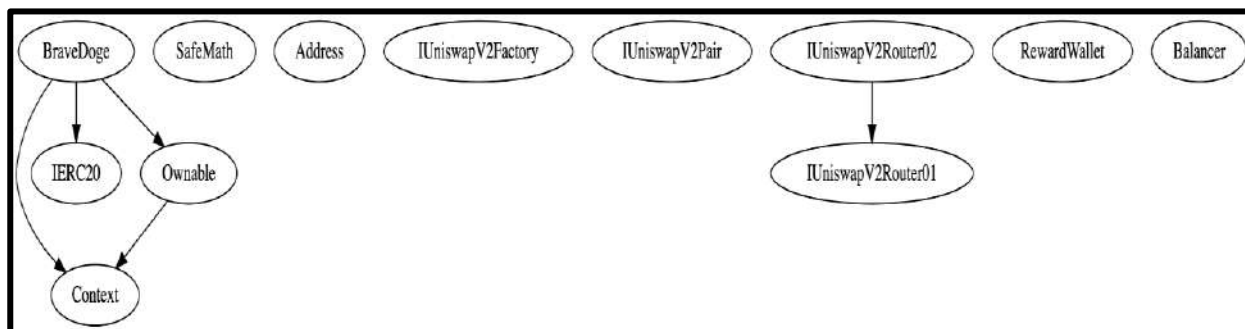
L	swapExactTokensForETHS upportingFeeOnTransferTo kens	External ⚡		NO⚡
L	swapExactTokensForToken sSupportingFeeOnTransfer Tokens	External ⚡		NO⚡
L	swapExactETHForTokensS upportingFeeOnTransferTo kens	External ⚡		NO⚡
RewardWallet	Implementation			
L		Public ⚡		NO⚡
Balancer	Implementation			
L		Public ⚡		NO⚡
BraveDoge	Implementation	Context, IERC20, Ownable		
L		Public ⚡		NO⚡
L	name	Public ⚡		NO⚡
L	symbol	Public ⚡		NO⚡
L	decimals	Public ⚡		NO⚡
L	totalSupply	Public ⚡		NO⚡
L	find2Percent	Internal 🔒		
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	isExcluded	Public ⚡		NO⚡
L	reflectionFromToken	Public ⚡		NO⚡
L	tokenFromReflection	Public ⚡		NO⚡

L	excludeAccount	External !		onlyOwner
L	includeAccount	External !		onlyOwner
L	_approve	Private		
L	_transfer	Private		
L	collectFee	Private		
L	_getReflectionRate	Private		
L	swapAndLiquify	Private		lockTheSwap
L	swapTokensForEth	Private		
L	addLiquidity	Private		
L	setPair	External !		onlyOwner
L	setTaxless	External !		onlyOwner
L	setSwapAndLiquifyEnabled	External !		onlyOwner
L	setFeeActive	External !		onlyOwner
L	setTaxFee	External !		onlyOwner
L	setBurnFee	External !		onlyOwner
L	setLiquidityFee	External !		onlyOwner
L	setMarketingFee	External !		onlyOwner
L	setMaxTxAmount	External !		onlyOwner
L	setMinTokensBeforeSwap	External !		onlyOwner
L		External !		NO!

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

- No low severity issues found.

❖ Resolved issues

- Misleading variable name - Informed & fixed

```
uint256 tokensToBurn = find2Percent(transferAmount);
uint256 tokensToTransfer = transferAmount.sub(tokensToBurn);

//transfer reflection
_reflectionBalance[sender↑] = _reflectionBalance[sender↑].sub(
    amount↑.mul(rate)
);

//Owner fee
_reflectionBalance[MarketingAddress] = _reflectionBalance[
    MarketingAddress
].add(tokensToBurn.mul(rate));
```

- **Wrong reflection from token calculations - Informed & fixed**

```
tokenAmount .sub(tokenAmount.mul(_taxFee).div(10**(_feeDecimal + 2)))
.mul(_getReflectionRate());
```

- The function includeAccount() 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. - **Informed & fixed**
- The function _getReflectionRate() 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. - **Informed & fixed**
- The contract is sending tokens to the marketing wallet instead of BNB, which is not a feasible option since the project team will have to sell tokens to provide funds for the project and that will directly affect the project's price. - **Informed & fixed**

Owner privileges

- ❖ The owner can transfer and renounce the ownership.

```
ftrace | funcSig
function renounceOwnership() public virtual onlyOwner {
    emit OwnershipTransferred(_owner, address(0));
    _owner = address(0);
}

/**
 * @dev Transfers ownership of the contract to a new account (`newOwner`).
 * Can only be called by the current owner.
 */
ftrace | funcSig
function transferOwnership(address newOwner) public virtual onlyOwner {
    require(
        newOwner != address(0),
        "Ownable: new owner is the zero address"
    );
    emit OwnershipTransferred(_owner, newOwner);
    _owner = newOwner;
}
```

- ❖ The owner can change the v2 router address.

```
ftrace | funcSig
function setPair(address pair) external onlyOwner {
    uniswapV2Pair = pair;
}
```

- ❖ The owner can exclude the wallet from taxes.

```
ftrace | funcSig
function setTaxless(address account, bool value) external onlyOwner {
    isTaxless[account] = value;
}
```

- ❖ The owner can enable/disable swap and liquidity adding.

```
ftrace | funcSig
function setSwapAndLiquifyEnabled(bool enabled↑) external onlyOwner {
    swapAndLiquifyEnabled = enabled↑;
    SwapAndLiquifyEnabledUpdated(enabled↑);
}
```

- ❖ The owner can activate and deactivate fees.

```
ftrace | funcSig
function setFeeActive(bool value↑) external onlyOwner {
    isFeeActive = value↑;
}
```

- ❖ The owner can change all fees.

```
ftrace | funcSig
function setTaxFee(uint256 fee↑) external onlyOwner {
    taxFee = fee↑;
}

ftrace | funcSig
function setBurnFee(uint256 fee↑) external onlyOwner {
    burnFee = fee↑;
}

ftrace | funcSig
function setLiquidityFee(uint256 fee↑) external onlyOwner {
    liquidityFee = fee↑;
}

ftrace | funcSig
function setMarketingFee(uint256 amount↑) external onlyOwner {
    MarketingFee = amount↑;
}
```

- ❖ The owner can Change max transaction amount.

```
ftrace | funcSig
function setMaxTxAmount(uint256 amount↑) external onlyOwner {
    maxTxAmount = amount↑;
}
```

Audit conclusion

While conducting the audit of the Brave Doge Token smart contract, it was observed that there is nothing alarming with the code.