RugFreeCoins Audit

Captain BNB Coin Token

Smart Contract Security Audit

August 19, 2021

Contents

Audit details

Audited project

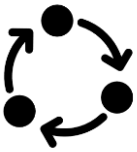
Captain BNB Token

Contract Address

0xc9043e9b90c0f6cde0164d85aef05fc8ef63d603

Client contact

Captain BNB Team



Blockchain

Binance smart chain

Project website

https://captainbnb.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 Captain BNB to perform an audit of the smart contract.

https://bscscan.com/token/0xc9043e9b90c0f6cde0164d85aef05fc8ef63d603

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

CaptainBNB is a token based on the Binance Smart Chain, which combines superhero memes, reflections, and games. Each transaction, purchase and sales incur a 14% fee.

CaptainBNB is 100% community driven and the holders will be given the opportunity to vote on a range of options for future marketing, game development as well as other community decisions. Two games are planned to be developed to support the ecosystem, one is a lottery game and the other one is an “earn to play” game.

Features:

- 7% of each transaction gets converted to BNBs, and is split amongst all holders. The holders will be eligible to receive tokens every one hour and rewards are proportional to how many tokens each individual holds.

- The **sustainability fee of 4% marketing and controlled buyback** is what allows CaptainBNB to hold the aforementioned promise. Tokens will be swapped into BNBs and will be sent to a marketing wallet per transaction. This way, CaptainBNB will have enough funds to promote the coin and spend for future development without selling tokens as the traditional way.

  Controlled buyback, the same funds will be used to save from massive dips in order to keep the token market price stable.

- **The liquidity fee of 3%**, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.

Tokenomics

14% tax when buying and selling

- 7% of trade goes to holders pockets in BNB.
- 4% of trade goes to the marketing wallet.
- 3% of trade goes to the liquidity pool.

Roadmap


ROADMAP
PHASE 1
• Ads Campaign & Marketing Push
• Website Launch with Dashboard
• 5,000 Telegram members
• First Audit before Launch
• 2 Play-to-Earn Game
PHASE 2
• Successful Launch
• CMS & Coinsniper Trending
• 10,000 Telegram members
• Sticker Development
• Listing on Coingecko
PHASE 3
• Influencers Marketing Push
• 20,000 Telegram members
• Dextools Trending
• Website Redesign
• Listing on Coinmarketcap
PHASE 4
• Huge Marketing Campaign
• Game Ver2 Update
• 50,000 Telegram members
• Second Audit
• Partnerships
PHASE 5
• Listing on CEX
• Billboard Ads
• NFT Collection
• CaptainBNB SWAP
• Staking
PHASE 6
• Global Marketing Push
• Third Audit
• Huge Donation
• More to be Announced
7

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 from BNB by holding tokens.
- Anyone who's interested in trading tokens.
- Anyone who's interested in taking part with CaptainBNB future plans.
- Anyone who's interested in playing games and earning.
- Anyone who's interested in taking part in making decisions in terms of marketing and development of the CaptainBNB ecosystem being a token holder.
- Anyone who's interested in making financial transactions with any other party CaptainBNB using as the currency.

Core concept

The BNB reward system

7% of each transaction gets converted to BNBs and is split amongst all holders. Holders will be eligible to receive tokens everyone hour and rewards are proportional to how many tokens each individual holds.

Sustainable mechanism

The **fee of 4% marketing** is what allows CaptainBNB 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, CaptainBNB 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 same funds will be consumed for buyback tokens manually, it is to save from massive dips in order to keep the token market price stable.

The liquidity fee of 3%, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.

CaptainBNB will launch two games, which are kingdom defense and Flapcat. Players can have fun and earn rewards at the same time. Also, the team is planning to structure a multi game universe game associated with the CaptainBNB brand.

Potential to grow with score points

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

Contract details

Token contract details for 20th August 2021

Contract name	Captain BNB Token
Contract address	0xc9043e9b90c0f6cde0164d85aef05fc8ef63d603
Token supply	2,000,000,000,000
Token ticker	CAPTBNB
Decimals	9
Token holders	1
Transaction count	1
Marketing address	0xf23bfd788d496800d837a3099121baf2a2ad3825
Auto liquidity receiver	0xf23bfd788d496800d837a3099121baf2a2ad3825
Top 100% holders dominance	99.82%
Distributor	0x9be575cb7a9949c4ed0004f6e56cbb3deffa7bf8
Contract deployer address	0xf23bfd788d496800d837a3099121baF2a2Ad3825
Contract's current owner address	0xf23bfd788d496800d837a3099121baf2a2ad3825

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**		
└	Function Name	Visibility	Mutability	Modifiers
SafeMath	**Library**			
└	add	Internal 🔒		
└	sub	Internal 🔒		
└	sub	Internal 🔒		
└	mul	Internal 🔒		
└	div	Internal 🔒		
└	div	Internal 🔒		
IBEP20	**Interface**			
└	totalSupply	External 📱		NO📱
└	decimals	External 📱		NO📱
└	symbol	External 📱		NO📱
└	name	External 📱		NO📱
└	getOwner	External 📱		NO📱
└	balanceOf	External 📱		NO📱
└	transfer	External 📱	🛑	NO📱
└	allowance	External 📱		NO📱
└	approve	External 📱	🛑	NO📱
└	transferFrom	External 📱	🛑	NO📱

Auth	Implementation			
└		Public ❗	🔘	NO❗
└	authorize	Public ❗	🔘	onlyOwner
└	unauthorize	Public ❗	🔘	onlyOwner
└	isOwner	Public ❗		NO❗
└	isAuthorized	Public ❗		NO❗
└	transferOwnership	Public ❗	🔘	onlyOwner
IDEXFactory	**Interface**			
└	createPair	External ❗	🔘	NO❗
IDEXRouter	**Interface**			
└	factory	External ❗		NO❗
└	WETH	External ❗		NO❗
└	addLiquidity	External ❗	🔘	NO❗
└	addLiquidityETH	External ❗	💷	NO❗
└	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🔘	NO❗
└	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💷	NO❗
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🔘	NO❗
IDividendDistributor	**Interface**			
└	setDistributionCriteria	External ❗	🔘	NO❗
└	setShare	External ❗	🔘	NO❗
└	deposit	External ❗	💷	NO❗
└	process	External ❗	🔘	NO❗

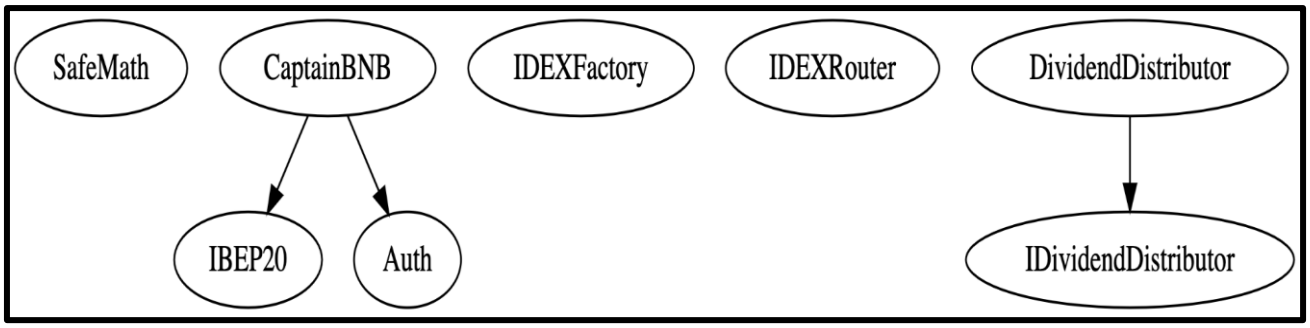
DividendDistributor	Implementation	IDividendDistributor		
└		Public ❗	🛑	NO❗
└	setDistributionCriteria	External ❗	🛑	onlyToken
└	setShare	External ❗	🛑	onlyToken
└	deposit	External ❗	💵	onlyToken
└	process	External ❗	🛑	onlyToken
└	shouldDistribute	Internal 🔒		
└	distributeDividend	Internal 🔒	🛑	
└	claimDividend	External ❗	🛑	NO❗
└	getUnpaidEarnings	Public ❗		NO❗
└	getCumulativeDividends	Internal 🔒		
└	addShareholder	Internal 🔒	🛑	
└	removeShareholder	Internal 🔒	🛑	
└	totalDividendsDistributed	Public ❗		NO❗
└	getAccount	Public ❗		NO❗
CaptainBNB	**Implementation**	**IBEP20, Auth**		
└		Public ❗	🛑	Auth
└		External ❗	💵	NO❗
└	totalSupply	External ❗		NO❗
└	decimals	External ❗		NO❗
└	symbol	External ❗		NO❗
└	name	External ❗		NO❗
└	getOwner	External ❗		NO❗
└	balanceOf	Public ❗		NO❗
└	allowance	External ❗		NO❗
└	approve	Public ❗	🛑	NO❗
└	transfer	External ❗	🛑	NO❗

└	transferFrom	External	●	NO
└	_transferFrom	Internal	●	
└	_basicTransfer	Internal	●	
└	shouldTakeFee	Internal		
└	takeFee	Internal	●	
└	shouldSwapBack	Internal		
└	swapBack	Internal	●	swapping
└	setBuyFees	Public	●	authorized
└	setSellFees	Public	●	authorized
└	clearStuckBalance	External	●	onlyOwner
└	setTxLimit	External	●	authorized
└	setIsDividendExempt	External	●	authorized
└	setIsFeeExempt	External	●	authorized
└	setIsTxLimitExempt	External	●	authorized
└	setFeeReceivers	External	●	onlyOwner
└	setSwapBackSettings	External	●	authorized
└	setDistributionCriteria	External	●	authorized
└	setAntiBotSettings	External	●	authorized
└	setDistributorSettings	External	●	authorized
└	airdropFixed	External	●	authorized
└	updateLaunchTime	External	●	onlyOwner
└	getAccountDividendsInfo	External		NO
└	getTotalDividendsDistributed	External		NO
└	claim	Public	●	NO
└	claimProcess	Public	●	NO
└	setIsBlacklisted	External	●	authorized
└	getCirculatingSupply	Public		NO
└	getLiquidityBacking	Public		NO
└	isOverLiquified	Public		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

Fixed issues

No maximum limit to update launch time - Informed and fixed

There is no maximum time limit in updateLaunchTime function so owners can disable trading by updating the launch time to a big value.

Owner privileges

Ownership privileges owner had

- The owner can change buying and selling fees.

```
ftrace | funcSig
function setBuyFees(
    uint256 _reflectionFee,
    uint256 _marketingFee,
    uint256 _liquidityFee
) public authorized {
    buyFees = FeeSet({
        reflectionFee: _reflectionFee,
        marketingFee: _marketingFee,
        liquidityFee: _liquidityFee,
        totalFee: _reflectionFee + _marketingFee + _liquidityFee
    });
    require(buyFees.totalFee < feeDenominator / 4);
}

ftrace | funcSig
function setSellFees(
    uint256 _reflectionFee,
    uint256 _marketingFee,
    uint256 _liquidityFee
) public authorized {
    sellFees = FeeSet({
        reflectionFee: _reflectionFee,
        marketingFee: _marketingFee,
        liquidityFee: _liquidityFee,
        totalFee: _reflectionFee + _marketingFee + _liquidityFee
    });
    require(sellFees.totalFee < feeDenominator / 4);
}
```

- The owner can transfer contract BNB balance to the marketing address.

```
ftrace | funcSig
function clearStuckBalance(uint256 amountPercentage) external onlyOwner {
    uint256 amountBNB = address(this).balance;
    payable(marketingFeeReceiver).transfer(
        (amountBNB * amountPercentage) / 100
    );
}
```

❖ The owner can transfer contract BNB balance to the marketing address.

```
ftrace | funcSig
function clearStuckBalance(uint256 amountPercentage) external onlyOwner {
    uint256 amountBNB = address(this).balance;
    payable(marketingFeeReceiver).transfer(
        (amountBNB * amountPercentage) / 100
    );
}
```

❖ The owner can change max transaction limit.

```
ftrace | funcSig
function setTxLimit(uint256 amount) external authorized {
    require(amount >= _totalSupply / 2000);
    _maxTxAmount = amount;
}
```

❖ The owner can exclude wallets from dividends.

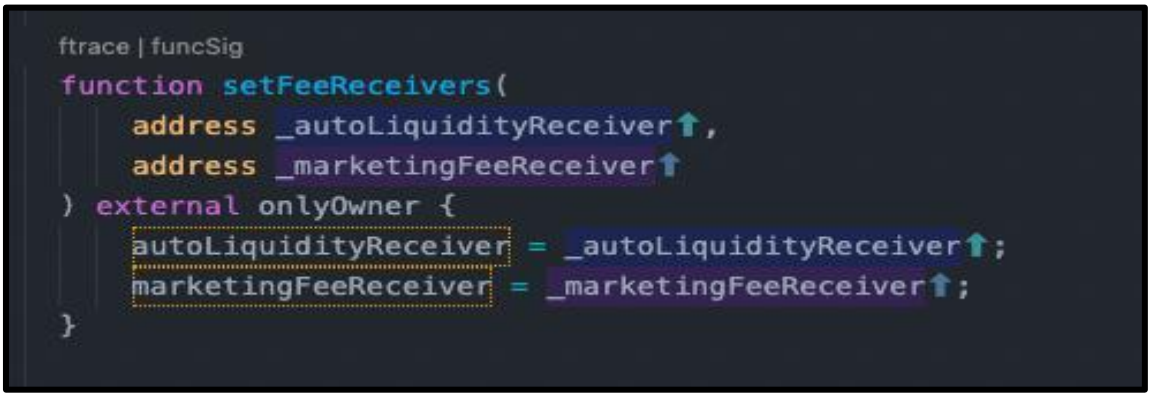
```
ftrace | funcSig
function setIsDividendExempt(address holder, bool exempt)
    external
    authorized
{
    require(holder != address(this) && holder != pair);
    isDividendExempt[holder] = exempt;
    if (exempt) {
        distributor.setShare(holder, 0);
    } else {
        distributor.setShare(holder, _balances[holder]);
    }
}
```

❖ The owner can exclude wallets from the fee.

```
ftrace | funcSig
function setIsFeeExempt(address holder, bool exempt) external authorized {
    isFeeExempt[holder] = exempt;
}
```

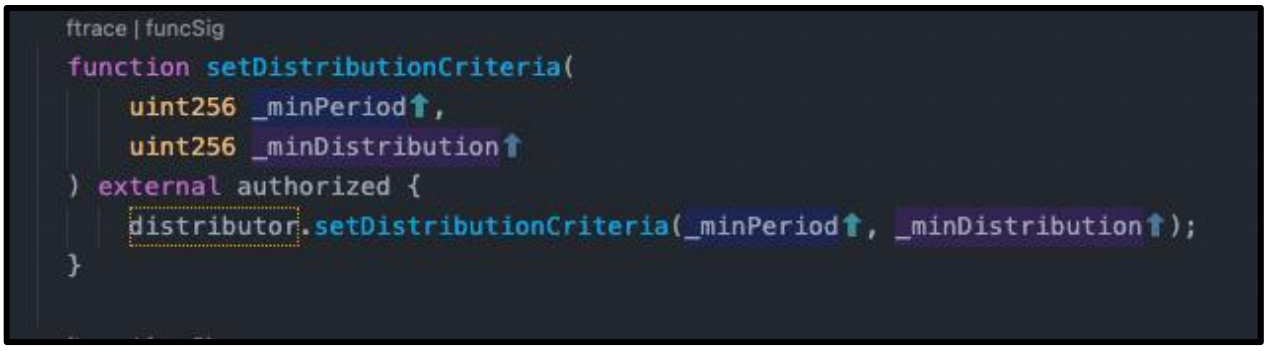
❖ The owner can exclude wallets from the max transaction limit.

❖ The owner can change marketing and liquidity fee receiver.

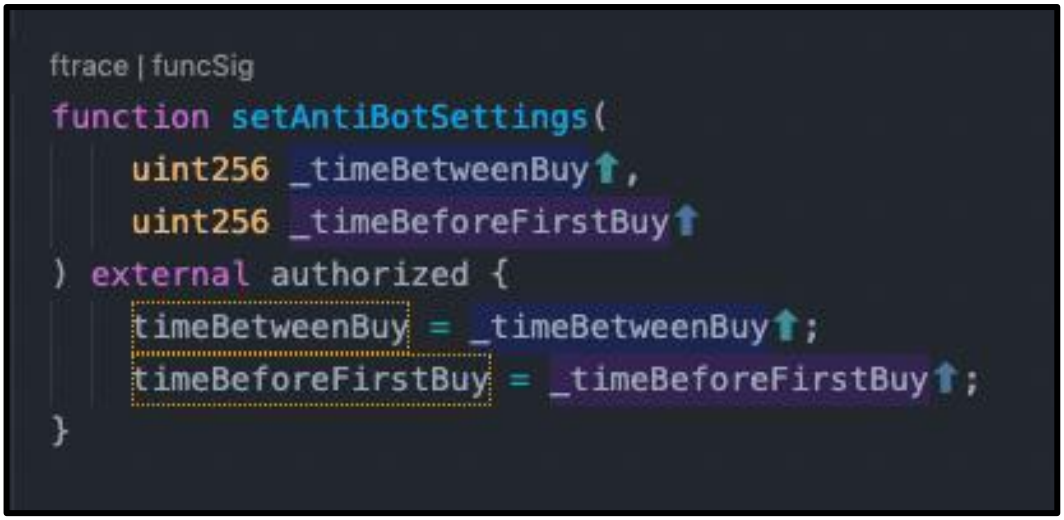


❖ The owner can enable/disable swap back and change swap back threshold.

❖ The owner can change the minimum distribution time and amount.

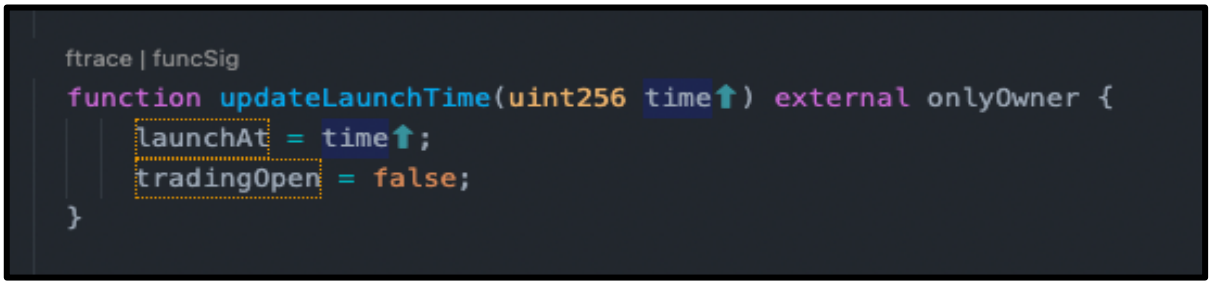


❖ The owner can change the anti bot setting.



❖ The owner can change the distribution gas fee up to 1000000.

❖ The owner can change launch time.



Audit conclusion

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