



RugFreeCoins Audit



ThunderBNB Token Audit
Smart Contract Security Audit
August 23, 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
Top token holders	11
Token distribution	12
Contract code function details	14
Contract description table	15
Security issue checking status	26
Owner privileges	27
Audit conclusion	34

Audit details



Audited project

ThunderBNB Token



Contract Address

0xb9654A42f0F5dCDEf5617DebF8bd048E33F180E7



Client contact

ThunderBNB Team



Blockchain

Binance smart chain



Project website

<https://thunderbnb.app/>

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 ThunderBNB to perform an audit of the smart contract.

<https://bscscan.com/token/0xb9654A42f0F5dCDEf5617DebF8bd048E33F180E7>

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

ThunderBNB is a token built on the Binance Smart Chain. Each transaction, purchase incur a 18% fee, and sales incur a 27% fee. The ThunderBNB Token launched on 16th August 2021 on PancakeSwap.

The first token of its type to provide its users with double reflection rewards, at a ratio of 80% given in a token which will provide immediate benefits to the user short term, and 20% provided in a token specifically built to provide profit for long term investors.

- ❖ **80% BNB distribution**
- ❖ **20% THOREUM distribution**

Features

- ❖ The **reward** system is what ThunderBNB's entire marketing strategy is based around, especially in the early stages of the life of the coin. The notion of 'passive income' can be leveraged in ThunderBNB's marketing approach and facilitate the next steps in the development and growth of the community. The fee of 10% from every trade will be divided to distribute 8% of BNB and 2% of THOREUM everyone hour among every holder proportional to how many tokens each individual holds.
- ❖ The **sustainability fee of 4% marketing fee when buying and selling** is what allows ThunderBNB to hold the aforementioned promise. Tokens will be swapped into BNBs and will be sent to a marketing wallet per transaction. This way, ThunderBNB will have enough funds to promote the coin and spend for future development without selling tokens as the traditional way.
- ❖ The 2% fee is used to buy back & burn ThunderCake, which is affiliated with ThunderBNB token.
- ❖ 2% auto liquidity pool and buy back & burn ThunderBNB. Based on volume & liquidity analysis, 2% of every transaction is transformed either into liquidity for Pancakeswap or into ThunderBNB automatic Buyback & burn fund. Once the system detects a sell, it will automatically buy back a certain value of tokens and burn. It's automatic and helps create either a higher price floor for stability or pump the price and rarify token.
- ❖ Anti-dump: Sales are restricted to less than 0.1% of the total supply. Buys have a 0.5% limitation. So only big buys and never see a big dump.
- ❖ Anti-bot Exploit: The Buyback and Burn will only cover the same sell value under a certain threshold. So massive bot-triggered micro-sells will not dry out the buyback fund.

Tokenomics

18% fee when buying

- ❖ 8% of trade goes to holders pockets in BNB.
- ❖ 2% of trade goes to holders pockets in THOREUM.
- ❖ 4% of trade will be allocated for marketing.
- ❖ 2% of trade will be allocated to buyback and burn ThunderCake.
- ❖ 2% of trade will be allocated for liquidity pool or buyback and burn ThunderBNB.

27% fee when selling

- ❖ 12% of trade goes to holders pockets in BNB.
- ❖ 3% of trade goes to holders pockets in THOREUM.
- ❖ 6% of trade will be allocated for marketing.
- ❖ 3% of trade will be allocated to buyback and burn ThunderCake.
- ❖ 3% of trade will be allocated for liquidity pool or buyback and burn ThunderBNB.

Roadmap

Phase 1

- ❖ Website Launch.
- ❖ 5000 Telegram Members.
- ❖ Presale Launch & Liquidity locked.
- ❖ Launch on PancakeSwap.
- ❖ \$5M Marketcap.
- ❖ Coingecko listing.
- ❖ Blockfolio Application.
- ❖ Influencer marketing push.
- ❖ Poocoin and other big website banner ads.

Phase 2

- ❖ +40k members in Telegram group.
- ❖ +10k followers in Twitter.
- ❖ 10,000 holders.
- ❖ Partnership with influencer Youtube, Twitter, & TikTok.
- ❖ ThunderBNB Dashboard (earnings tracking).
- ❖ CoinmarketCap listing.
- ❖ Massive Marketing.
- ❖ \$25M Market Cap.
- ❖ \$50M Market Cap.
- ❖ \$100M Market Cap.

Phase 3

- ❖ ThunderBNBSwap.
- ❖ ThunderBNB Merch Shop.
- ❖ Marketing: ThunderBNB goes to mainstream.
- ❖ Listing on CEX.
- ❖ \$500M MarketCap.
- ❖ Charity donation.
- ❖ Research donation.
- ❖ More to be announced

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 or THOREUM by holding tokens.
- ❖ Anyone who's interested in trading tokens.
- ❖ All BNB or THOREUM and fans out there.
- ❖ Anyone who's interested in taking part with the future plans of the project.
- ❖ Anyone who's interested in making financial transactions with any other party using ThunderBNB as the currency.

Core concept

The reward system

10% of each transaction when buying and 15% when selling gets converted to BNB and THOREUM in the power of 80/20 and is split amongst all holders. The rewards are sent to holders that have at least 200,000 ThunderBNB tokens, holders will be eligible to receive tokens everyone hour and rewards are proportional to how many tokens each individual holds.

BNB - 80% for now and mid-term:

BNB is now about \$400 but expected to be \$800 at the end of this year and potential to \$2,000 (x5) in mid-term because it is the dominant decentralized exchange on the BSC with billions of volume going on every day.

8% of transaction tax in BNB make sure that ThunderBNB holders benefit immediately from the big rewards coming from every transaction, and this can be equal to 24% when BNB price x3 in the near future.

THOREUM - 20% for long-term future:

Thorem is expected to be the next Safemoon 2.0 with many innovative features that is the first in the crypto world, such as Thor Thunder to buy back & burn with 3700+ BNB fund & increasing ~100 BNB every day, or Thunder Boost to earn staking rewards while still getting reflection rewards, etc. With a hyper deflationary nature, Thorem price is expected to be x100 in this year and next.

2% of transaction tax for a token like Thorem is like a long-term investment and it can be equal to 200% in the long-term future.

Sustainable mechanism

The **fee of 4% marketing when buying and 6% when selling** is what allows ThunderBNB to promote the token and use funds to further development of the platform. Tokens will be swapped into BNB and will be sent to a marketing wallet per transaction. This way, ThunderBNB 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 **liquidity fee** of 3%, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.

The 2% buy back & burn ThunderCake is to promote the affiliated token of ThunderBNB, which will enable ThunderBNB to be more popular and be strong in terms of marketing as ThunderCake success will directly affect the ThunderBNB and vice versa.

2% auto liquidity pool and buy back & burn ThunderBNB

Based on volume & liquidity analysis, 2% of every transaction is transformed either into liquidity for Pancakeswap or into ThunderBNB automatic Buyback & burn fund. Liquidity pool is the redistribution mechanism that ensures the trading pool always has sufficient liquidity. Once the system detects a sell, it will automatically buy back a certain value of tokens and burn. It's automatic and helps create either a higher price floor for stability or pump the price and rarify token.

Potential to grow with score points

1.	Project efficiency	9/10
2.	Project uniqueness	9/10
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	8/10
Total Points		8.75/10

Contract details

Token contract details for 23rd August 2021

Contract name	ThunderBNB Token
Contract address	0xb9654A42f0F5dCDEf5617DebF8bd048E33F180E7
Token supply	100,000,000,000
Token ticker	THUNDERBNB
Decimals	18
Token holders	4,845
Transaction count	36,816
Top 100% holders dominance	71.47%
Marketing wallet	0x8800f4755d5ec881132126b2f2010fe8b09ce880
Team wallet	0x9783d2770a5970aa982c7018784a16e69a7897d1
ThunderCake burn address	0xce4a4a15fccd532ead67be3ecf7e6122c61d06bb
Contract deployer address	0x8cA23deA974781c94D961950230FB1Ffdb61bB60
Contract's current owner address	0x8ca23dea974781c94d961950230fb1ffdb61bb60

Top token holders

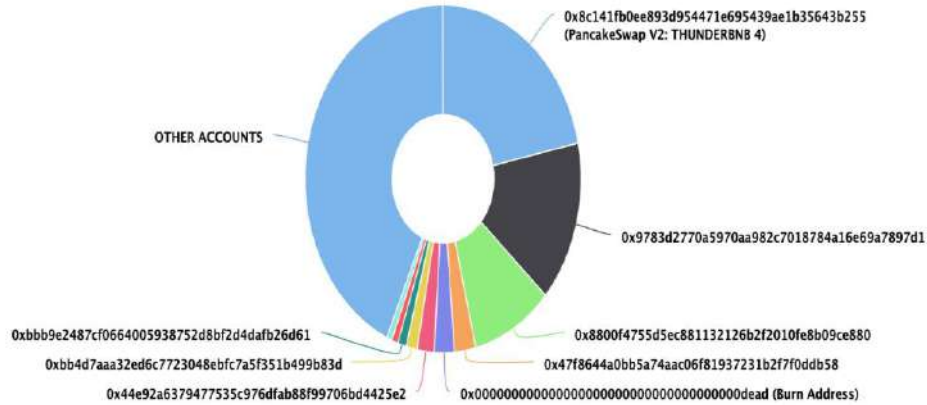
Top 10 Token Holders

The top 10 holders collectively own 56.69% (56,693,767,283.44 Tokens) of ThunderBNB Token

Token Total Supply: 100,000,000,000.00 Token | Total Token Holders: 4,847

ThunderBNB Token Top 10 Token Holders

Source: BscScan.com



(A total of 56,693,767,283.44 tokens held by the top 10 accounts from the total supply of 100,000,000,000.00 token)

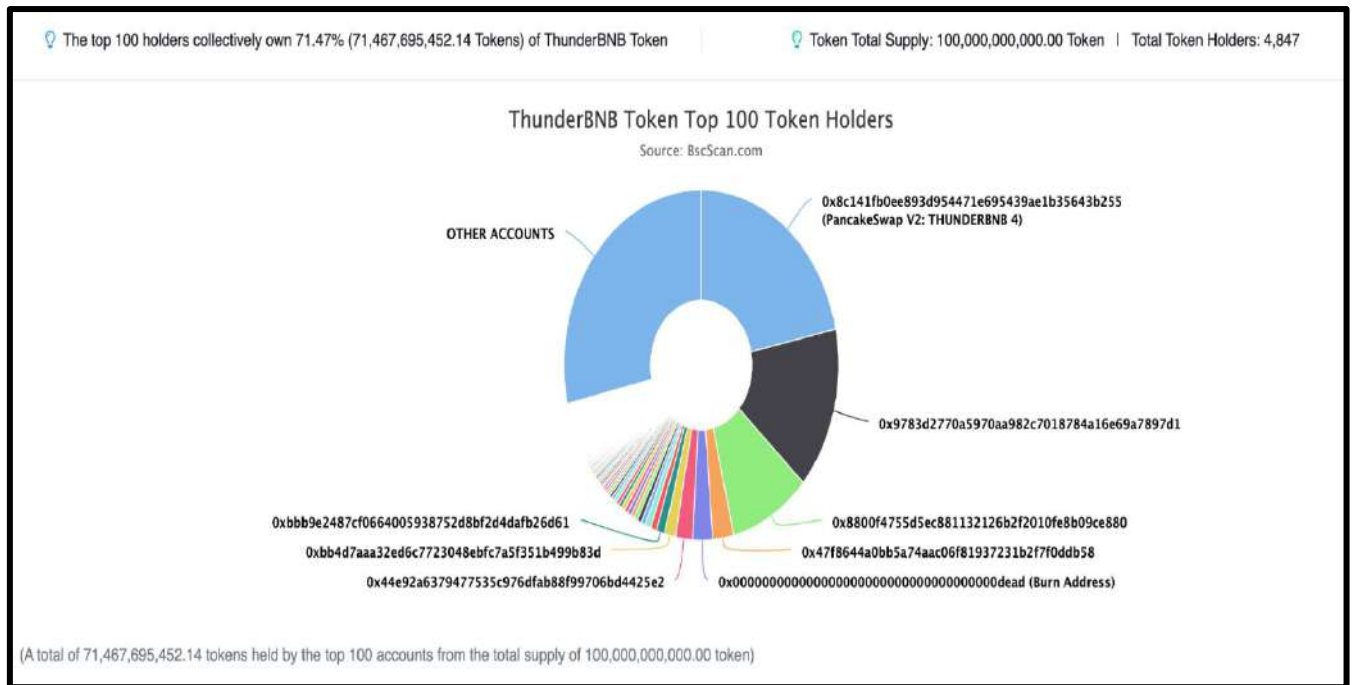
Rank	Address	Quantity (Token)	Percentage
1	PancakeSwap V2: THUNDERBNB 4	21,704,617,202.404208533938574399	21.7046%
2	0x9783d2770a5970aa982c7018784a16e69a7897d1	15,000,000,000	15.0000%
3	0x8800f4755d5ec881132126b2f2010fe8b09ce880	9,499,610,000	9.4996%
4	0x47f8644a0bb5a74aac06f81937231b2f7f0ddb58	2,481,319,418.60530948625187812	2.4813%
5	Burn Address	2,330,702,134.965791347011370576	2.3307%
6	0x44e92a6379477535c976dfab88f99706bd4425e2	2,000,000,000	2.0000%
7	0xbbb4d7aaa32ed6c7723048ebfc7a5f351b499b83d	1,305,356,058.043706304847425962	1.3054%
8	0xbbb9e2487cf0664005938752d8bf2d4dafb26d61	1,000,000,000	1.0000%
9	0xca4bec9ffe32990a7ac7145d994b73106b913586	725,788,518.582998418958195555	0.7258%
10	0xe49adc4454e7487aada4c8ce68be9833ad642689	646,373,950.838737350237999563	0.6464%

Token distribution

Tokens are distributed as follows:



Top 100 Token Holders




















































































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	informational
		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 		
Ownable	Implementation	Context		
L		Public 		NO 
L	owner	Public 		NO 
L	renounceOwnership	Public 		onlyOwner
L	transferOwnership	Public 		onlyOwner
L	lock	Public 		onlyOwner
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 
ERC20	Implementation	Context, IERC20		
L		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	_setupDecimals	Internal 		
L	_beforeTokenTransfer	Internal 		
IDividendPayingToken		Interface		
L	dividendOf	External 		NO 
L	withdrawDividend	External 		NO 
IDividendPayingTokenOptional		Interface		
L	withdrawableDividendOf	External 		NO 
L	withdrawnDividendOf	External 		NO 
L	accumulativeDividendOf	External 		NO 
DividendPayingToken		Implementation	ERC20, IDividendPayingToken,	

























































		IDividendPaying TokenOptional		
L		Public		ERC20
L		External		NO
L	distributeDividends	Public		NO
L	withdrawDividend	Public		NO
L	setOnlyCaller	External		NO
L	setDividendTokenAddress	External		NO
L	setMinTokenBeforeSendDividend	External		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		
IUniswapV2Factory	Interface			
L	feeTo	External		NO
L	feeToSetter	External		NO
L	getPair	External		NO
L	allPairs	External		NO
L	allPairsLength	External		NO
L	createPair	External		NO
L	setFeeTo	External		NO
L	setFeeToSetter	External		NO























































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¶













































IUniswapV2Router01	Interface			
L	factory	External ¶		NO¶
L	WETH	External ¶		NO¶
L	addLiquidity	External ¶		NO¶
L	addLiquidityETH	External ¶		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 ¶		NO¶
L	swapTokensForExactETH	External ¶		NO¶
L	swapExactTokensForETH	External ¶		NO¶
L	swapETHForExactTokens	External ¶		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	swapExactETHFor TokensSupporting FeeOnTransferTo kens	External ⚠		NO⚠
L	swapExactTokens ForETHSupporting FeeOnTransferTo kens	External ⚠		NO⚠
IterableMapping	Library			
L	get	Public ⚠		NO⚠
L	getIndexOfKey	Public ⚠		NO⚠
L	getKeyAtIndex	Public ⚠		NO⚠
L	size	Public ⚠		NO⚠
L	set	Public ⚠		NO⚠
L	remove	Public ⚠		NO⚠
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 🔒		

SafeMathInt	Library			
L	mul	Internal 		
L	div	Internal 		
L	sub	Internal 		
L	add	Internal 		
L	toUint256Safe	Internal 		
SafeMathUint	Library			
L	toInt256Safe	Internal 		
ThunderBnb	Implementation	ERC20, Ownable		
L		Public 		ERC20
L		External 		NO 
L	prepareForPartnerOrExchangeListing	External 		onlyOwner
L	setMaxBuyTransaction	External 		onlyOwner
L	setMaxSellTransaction	External 		onlyOwner
L	updateA80DividendToken	External 		onlyOwner
L	updateB20DividendToken	External 		onlyOwner
L	updateMinA80BeforeSendDividend	External 		onlyOwner
L	updateMinB20BeforeSendDividend	External 		onlyOwner
L	getMinA80BeforeSendDividend	External 		NO 
L	getMinB20BeforeSendDividend	External 		NO 
L	setSendA80InTx	External 		onlyOwner
L	setSendB20InTx	External 		onlyOwner
L	setA80DividendPriority	External 		onlyOwner
L	updateBurnThunderCakeAddress	External 		onlyOwner
L	updateTeamWallet	External 		onlyOwner
L	updateMarketingWallet	External 		onlyOwner

L	setToBurnThunderCakeFee	External 		onlyOwner
L	setMaxWalletToken	External 		onlyOwner
L	setSwapTokensAtAmount	External 		onlyOwner
L	setSellTransactionMultiplier	External 		onlyOwner
L	afterPreSale	External 		onlyOwner
L	setTradingIsEnabled	External 		onlyOwner
L	setBuyBackMode	External 		onlyOwner
L	setMinimumBalanceRequired	Public 		onlyOwner
L	setMinimumSellOrderAmount	Public 		onlyOwner
L	setBuyBackUpperLimit	External 		onlyOwner
L	_updateTotalFee	Internal 		
L	setBuyBackAndLiquifyEnabled	External 		onlyOwner
L	setA80DividendEnabled	External 		onlyOwner
L	setB20DividendEnabled	External 		onlyOwner
L	setMarketingEnabled	External 		onlyOwner
L	updateA80DividendTracker	External 		onlyOwner
L	updateB20DividendTracker	External 		onlyOwner
L	updateA80DividendRewardFee	External 		onlyOwner
L	updateB20DividendRewardFee	External 		onlyOwner
L	updateMarketingFee	External 		onlyOwner
L	updateBuyBackAndLiquidityFee	External 		onlyOwner
L	updateUniswapV2Router	External 		onlyOwner
L	excludeFromFees	Public 		onlyOwner
L	excludeFromDividend	Public 		onlyOwner
L	setAutomatedMarketMakerPair	Public 		onlyOwner
L	_setAutomatedMarketMakerPair	Private 		onlyOwner
L	updateGasForProcessing	External 		onlyOwner
L	updateMinimumBalanceForDividends	External 		onlyOwner

L	updateClaimWait	External 		onlyOwner
L	getA80ClaimWait	External 		NO 
L	getB20ClaimWait	External 		NO 
L	getTotalA80DividendsDistributed	External 		NO 
L	getTotalB20DividendsDistributed	External 		NO 
L	getIsExcludedFromFees	Public 		NO 
L	withdrawableA80DividendOf	External 		NO 
L	withdrawableB20DividendOf	External 		NO 
L	a80DividendTokenBalanceOf	External 		NO 
L	b20DividendTokenBalanceOf	External 		NO 
L	getAccountA80DividendsInfo	External 		NO 
L	getAccountB20DividendsInfo	External 		NO 
L	getAccountA80DividendsInfoAtIndex	External 		NO 
L	getAccountB20DividendsInfoAtIndex	External 		NO 
L	processDividendTracker	External 		onlyOwner
L	rand	Internal 		
L	claim	External 		NO 
L	getLastA80DividendProcessedIndex	External 		NO 
L	getLastB20DividendProcessedIndex	External 		NO 
L	getNumberOfA80DividendTokenHolders	External 		NO 
L	getNumberOfB20DividendTokenHolders	External 		NO 
L	_transfer	Internal 		
L	swapAndLiquify	Private 		
L	addLiquidity	Private 		
L	buyBackThunderCakeAndBurn	Private 		
L	buyBackAndBurn	Private 		
L	manualBuyBackAndBurn	Public 		onlyOwner

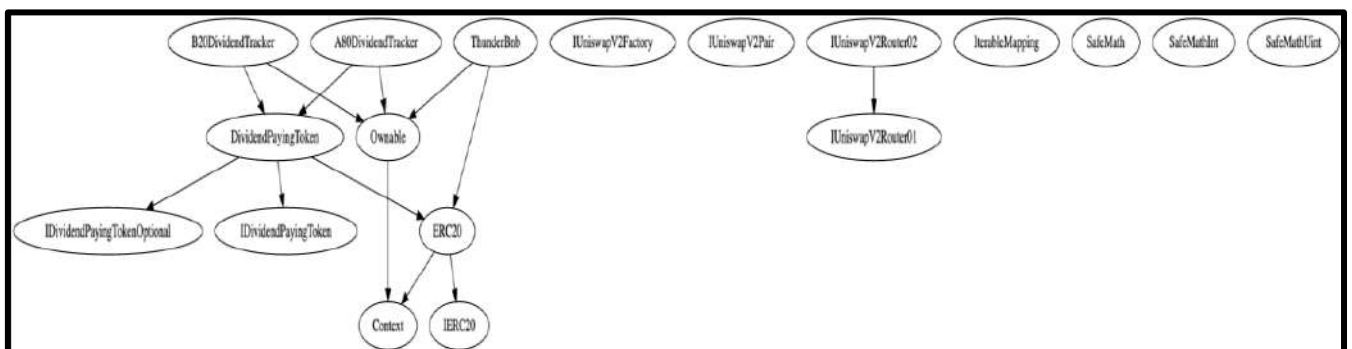
L	swapTokensForB NB	Private 		
L	swapTokensForDi videndToken	Private 		
L	swapAndSendA80 Dividends	Private 		
L	swapAndSendB20 Dividends	Private 		
L	transferToWallet	Private 		
A80DividendTracker	Implementation	DividendPaying Token, Ownable		
L		Public 		DividendPa yingToken
L	_transfer	Internal 		
L	withdrawDividend	Public 		NO 
L	setDividendToken Address	External 		onlyOwner
L	updateMinimumTo kenBalanceForDivi dends	External 		onlyOwner
L	excludeFromDivid ends	External 		onlyOwner
L	updateClaimWait	External 		onlyOwner
L	getLastProcessedI ndex	External 		NO 
L	getNumberOfToke nHolders	External 		NO 
L	getAccount	Public 		NO 
L	getAccountAtIndex	Public 		NO 
L	canAutoClaim	Private 		
L	setBalance	External 		onlyOwner
L	process	Public 		NO 
L	processAccount	Public 		onlyOwner
B20DividendTracker	Implementation	DividendPaying Token, Ownable		
L		Public 		DividendPa yingToken
L	_transfer	Internal 		
L	withdrawDividend	Public 		NO 

L	setDividendTokenAddress	External 卩	⬢	onlyOwner
L	updateMinimumTokenBalanceForDividends	External 卩	⬢	onlyOwner
L	excludeFromDividends	External 卩	⬢	onlyOwner
L	updateClaimWait	External 卩	⬢	onlyOwner
L	getLastProcessedIndex	External 卩		NO卩
L	getNumberOfTokenHolders	External 卩		NO卩
L	getAccount	Public 卩		NO卩
L	getAccountAtIndex	Public 卩		NO卩
L	canAutoClaim	Private 卩		
L	setBalance	External 卩	⬢	onlyOwner
L	process	Public 卩	⬢	NO卩
L	processAccount	Public 卩	⬢	onlyOwner

Legend

Symbol	Meaning
⬢	Function can modify state
卩	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.

❖ Informational

- The owner can enable and disable trading any time.

```
ftrace | funcSig
function setTradingIsEnabled(bool _enabled↑) external onlyOwner {
    tradingIsEnabled = _enabled↑;
}
```

- The owner can increase the sell fee without any limitation.

```
ftrace | funcSig
function setSellTransactionMultiplier(uint256 _multiplier↑)
    external
    onlyOwner
{
    sellFeeIncreaseFactor = _multiplier↑;
}
```

Owner privileges

- ❖ The owner can change max buy and sell transaction amount.

```
ftrace | funcSig
function setMaxBuyTransaction(uint256 _maxTxn↑) external onlyOwner {
    maxBuyTransactionAmount = _maxTxn↑ * (10**18);
}

ftrace | funcSig
function setMaxSellTransaction(uint256 _maxTxn↑) external onlyOwner {
    maxSellTransactionAmount = _maxTxn↑ * (10**18);
}
```

- ❖ The owner can change dividend tokens.

```
ftrace | funcSig
function updateA80DividendToken(address _newContract↑) external onlyOwner {
    a80DividendToken = _newContract↑;
    a80DividendTracker.setDividendTokenAddress(_newContract↑);
}

ftrace | funcSig
function updateB20DividendToken(address _newContract↑) external onlyOwner {
    b20DividendToken = _newContract↑;
    b20DividendTracker.setDividendTokenAddress(_newContract↑);
}
```

- ❖ The owner can change the minimum dividend amount that can be sent.

```
function updateMinA80BeforeSendDividend(uint256 _newAmount↑)
    external
    onlyOwner
{
    a80DividendTracker.setMinTokenBeforeSendDividend(_newAmount↑);
}

ftrace | funcSig
function updateMinB20BeforeSendDividend(uint256 _newAmount↑)
    external
    onlyOwner
{
    b20DividendTracker.setMinTokenBeforeSendDividend(_newAmount↑);
}
```

- ❖ The owner can enable/disable dividend sending.

```
ftrace | funcSig
function setSendA80InTx(bool _newStatus↑) external onlyOwner {
    sendA80InTx = _newStatus↑;
}

ftrace | funcSig
function setSendB20InTx(bool _newStatus↑) external onlyOwner {
    sendB20InTx = _newStatus↑;
}
```

- ❖ The owner can change max BNB sending percentage as dividends.

```
ftrace | funcSig
function setA80DividendPriority(uint256 _newAmount↑) external onlyOwner {
    require(_newAmount↑ >= 0 && _newAmount↑ <= 100, "Error amount");
    a80DividendPriority = _newAmount↑;
}
```

- ❖ The owner can change the Thundercake address.

```
ftrace | funcSig
function updateBurnThunderCakeAddress(address _newAddress↑)
    external
    onlyOwner
{
    toBurnThunderCakeAddress = _newAddress↑;
}
```

- ❖ The owner can change the Team wallet address.

```
ftrace | funcSig
function updateTeamWallet(address _newWallet↑) external onlyOwner {
    excludeFromFees(_newWallet↑, true);
    teamWallet = _newWallet↑;
}
```

- ❖ The owner can change the marketing wallet address.

```
ftrace | funcSig
function updateMarketingWallet(address _newWallet↑) external onlyOwner {
    excludeFromFees(_newWallet↑, true);
    marketingWallet = _newWallet↑;
}
```

- ❖ The owner can change Thundercake burn fee maximum up to 25%.

```
ftrace | funcSig
function setToBurnThunderCakeFee(uint256 newFee↑) external onlyOwner {
    require(newFee↑ <= MAX_FEE_RATE, "wrong");
    toBurnThunderCakeFee = newFee↑;
    _updateTotalFee();
}
```

- ❖ The owner can change max wallet token amount (the maximum token amount a wallet can hold).

```
ftrace | funcSig
function setMaxWalletToken(uint256 _maxToken↑) external onlyOwner {
    maxWalletToken = _maxToken↑ * (10**18);
}
```

- ❖ The owner can change the swap token amount.

```
ftrace | funcSig
function setSwapTokensAtAmount(uint256 _swapAmount↑) external onlyOwner {
    swapTokensAtAmount = _swapAmount↑ * (10**18);
}
```

- ❖ The owner can enable and disable trading.

```
ftrace | funcSig
function setTradingIsEnabled(bool _enabled↑) external onlyOwner {
    tradingIsEnabled = _enabled↑;
}
```


- ❖ The owner can change the minimum required BNB balance for buy back.

```
ftrace | funcSig
function setMinimumBalanceRequired(uint256 _newAmount↑) public onlyOwner {
    require(_newAmount↑ >= 0, "newAmount error");
    minimumBalanceRequired = _newAmount↑;
}
```

- ❖ The owner can change the minimum sell token amount to buy back.

```
ftrace | funcSig
function setMinimumSellOrderAmount(uint256 _newAmount↑) public onlyOwner {
    require(_newAmount↑ > 0, "newAmount error");
    minimumSellOrderAmount = _newAmount↑;
}
```

- ❖ The owner can change the buy back upper limit.

```
ftrace | funcSig
function setBuyBackUpperLimit(uint256 buyBackLimit↑) external onlyOwner {
    require(buyBackLimit↑ > 0, "buyBackLimit error");
    buyBackUpperLimit = buyBackLimit↑;
}
```

- ❖ The owner can change all fees maximum up to 25%.

```
ftrace | funcSig
function updateA80DividendRewardFee(uint8 newFee↑) external onlyOwner {
    require(newFee↑ <= MAX_FEE_RATE, "wrong");
    a80DividendRewardsFee = newFee↑;
    _updateTotalFee();
}

ftrace | funcSig
function updateB20DividendRewardFee(uint8 newFee↑) external onlyOwner {
    require(newFee↑ <= MAX_FEE_RATE, "wrong");
    b20DividendRewardsFee = newFee↑;
    _updateTotalFee();
}

ftrace | funcSig
function updateMarketingFee(uint8 newFee↑) external onlyOwner {
    require(newFee↑ <= MAX_FEE_RATE, "wrong");
    marketingFee = newFee↑;
    _updateTotalFee();
}

ftrace | funcSig
function updateBuyBackAndLiquidityFee(uint8 newFee↑) external onlyOwner {
    require(newFee↑ <= MAX_FEE_RATE, "wrong");
    buyBackAndLiquidityFee = newFee↑;
    _updateTotalFee();
}
```

- ❖ The owner can change v2 router address.

```
ftrace | funcSig
function updateUniswapV2Router(address newAddress↑) external onlyOwner {
    uniswapV2Router = IUniswapV2Router02(newAddress↑);
}
```

- ❖ The owner can exclude wallet from fees and dividends.

```
ftrace | funcSig
function excludeFromFees(address account↑, bool excluded↑) public onlyOwner {
    require(isExcludedFromFees[account↑] != excluded↑, "Already excluded");
    isExcludedFromFees[account↑] = excluded↑;

    emit ExcludeFromFees(account↑, excluded↑);
}

ftrace | funcSig
function excludeFromDividend(address account↑) public onlyOwner {
    a80DividendTracker.excludeFromDividends(address(account↑));
    b20DividendTracker.excludeFromDividends(address(account↑));
}
```

- ❖ The owner can change the claim wait.

```
ftrace | funcSig
function updateClaimWait(uint256 claimWait↑) external onlyOwner {
    a80DividendTracker.updateClaimWait(claimWait↑);
    b20DividendTracker.updateClaimWait(claimWait↑);
}
```

- ❖ The owner can manually process dividends.

```
ftrace | funcSig
function processDividendTracker(uint256 gas↑) external onlyOwner {
    (
        uint256 aIterations,
        uint256 aClaims,
        uint256 aLastProcessedIndex
    ) = a80DividendTracker.process(gas↑);
    emit ProcessedA80DividendTracker(
        aIterations,
        aClaims,
        aLastProcessedIndex,
        false,
        gas↑,
        tx.origin
    );

    (
        uint256 bIterations,
        uint256 bClaims,
        uint256 bLastProcessedIndex
    ) = b20DividendTracker.process(gas↑);
    emit ProcessedB20DividendTracker(
        bIterations,
        bClaims,
        bLastProcessedIndex,
        false,
        gas↑,
        tx.origin
    );
}
```

- ❖ The owner can manually trigger the buy back function.

```
ftrace | funcSig
function manualBuyBackAndBurn(uint256 _amount↑) public onlyOwner {
    uint256 balance = address(this).balance;
    require(buyBackAndLiquifyEnabled, "not enabled");
    require(
        balance >= minimumBalanceRequired.add(_amount↑),
        "amount is too big"
    );

    if (!swapping) {
        buyBackAndBurn(_amount↑);
    }
}
```

Audit conclusion

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