RugFreeCoins Audit

Bomber Babes Token

Smart Contract Security Audit

April 09, 2022

Contents

Audit details

Audited project
Bomber Babes Token

Contract Address
0x4B78bD1e38Fc5d2Dd43a0c9326A20E7f9099C95a

Client contact
Bomber Babes Team

Blockchain
Binance smart chain

Project website
https://bomberbabes.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 the Bomber Babes Team to perform an audit of the smart contract.

https://bscscan.com/token/0x4B78bD1e38Fc5d2Dd43a0c9326A20E7f9099C95a

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 improving the security posture of the smart contract by remediating the issues that were identified.

About the project

Bomber Babes Token is a token built on the Binance Smart Chain that is with an innovative investment use case the main purpose of which is to seek out constant revenue sources, **auto staking protocol backed by Defi 3.0 yield farming** on BSC. Bomber Babes will bring an unparallel, fixed APY of **%, the highest of its kind** onto the BSC blockchain while imposing profound ease, simplicity, and accessibility upon all Bomber Babes holders. Each transaction, purchase incurs an 11% fee, and sale incurs a 19% fee.

Features

- 4% of the buy and sale fees are directed to the vault which helps sustain and back the Staking Rewards provided by the Positive Rebase.

- The sustainability fee of 5% when buying and selling for marketing, and 3% when buying and selling for development, is what allows Bomber Babes Token to hold the aforementioned promise. Tokens will be swapped into BNB and will be sent to a marketing wallet. This way, Bomber Babes Token will have enough funds to promote the coin and spend for future development without selling tokens as the traditional way.

- The additional component included under the sustainability section is a liquidity fee of 2% when buying and selling , which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.

Roadmap


SEED-SALE | 100 BNB
GAME DEVELOPMENT
WEBSITE/WHITEPAPER RELEASE
BUILD AWARENESS
PRE-MARKETING CAMPAIGN
2022 / Q1
PRIVATE SALE | 100 BNB
PRE-MARKETING
NFT-DEVELOPMENT
AMA'S & PARTNERSHIPS
2022 / Q1
PRE-SALE | 200 SC / 400 HC
LAUNCH-MARKETING
STAKING DAPP LAUNCH
2022 / Q2
BETA GAME VERSION RELEASE
ETH MULTICHAIN LAUNCH
2022 / Q2
V1 GAME RELEASE
NFT AIRDROP
V1 NFT MARKET PLACE RELEASE
2022 / Q2
V2 GAME RELEASE
FIRST TOURNAMENT HOSTING
2022 / Q3

Tokenomics

14% fee when buying

- 5% of trade goes to marketing wallet in BNB
- 4% of trade goes to vault in tokens
- 3% trade goes to the development wallet
- 2% of trade goes to the liquidity pool.

16% fee when selling

- 5% of trade goes to marketing wallet in BNB
- 4% of trade goes to vault in tokens
- 3% trade goes to the development wallet
- 2% of trade goes to the liquidity pool.

18% fee when selling (increased sell fee by 4% first 72h launch)

- 9% of trade goes to marketing wallet in BNB
- 4% of trade goes to vault in tokens
- 3% trade goes to the development wallet
- 2% of trade goes to the liquidity pool.

Target market and the concept

Target market

- Anyone who's interested in the Crypto space with long-term investment plans.
- Anyone who's ready to earn a passive income by holding tokens.
- Anyone who's interested in trading tokens.
- Anyone who's ready in receiving automatic staking and compound rewards every 8 minutes.
- Anyone who's interested in receiving fixed interest of 2,033,199.56% per year.
- Anyone who's interested in taking part in the future plans of the Bomber Babes token.
- Anyone who's interested in making financial transactions with any other party using Bomber Babe as the currency.

.

Core concept

Reward mechanism

4% of when buying selling are stored in the Insurance fund which helps sustain and back the staking rewards provided by the positive rebase.

Bomber Babes fund is a separate wallet in the ecosystem. The Bomber Babes fund uses an algorithm that backs the Rebase Rewards and is supported by a portion of the buy and sell trading fees that accrue in the wallet.

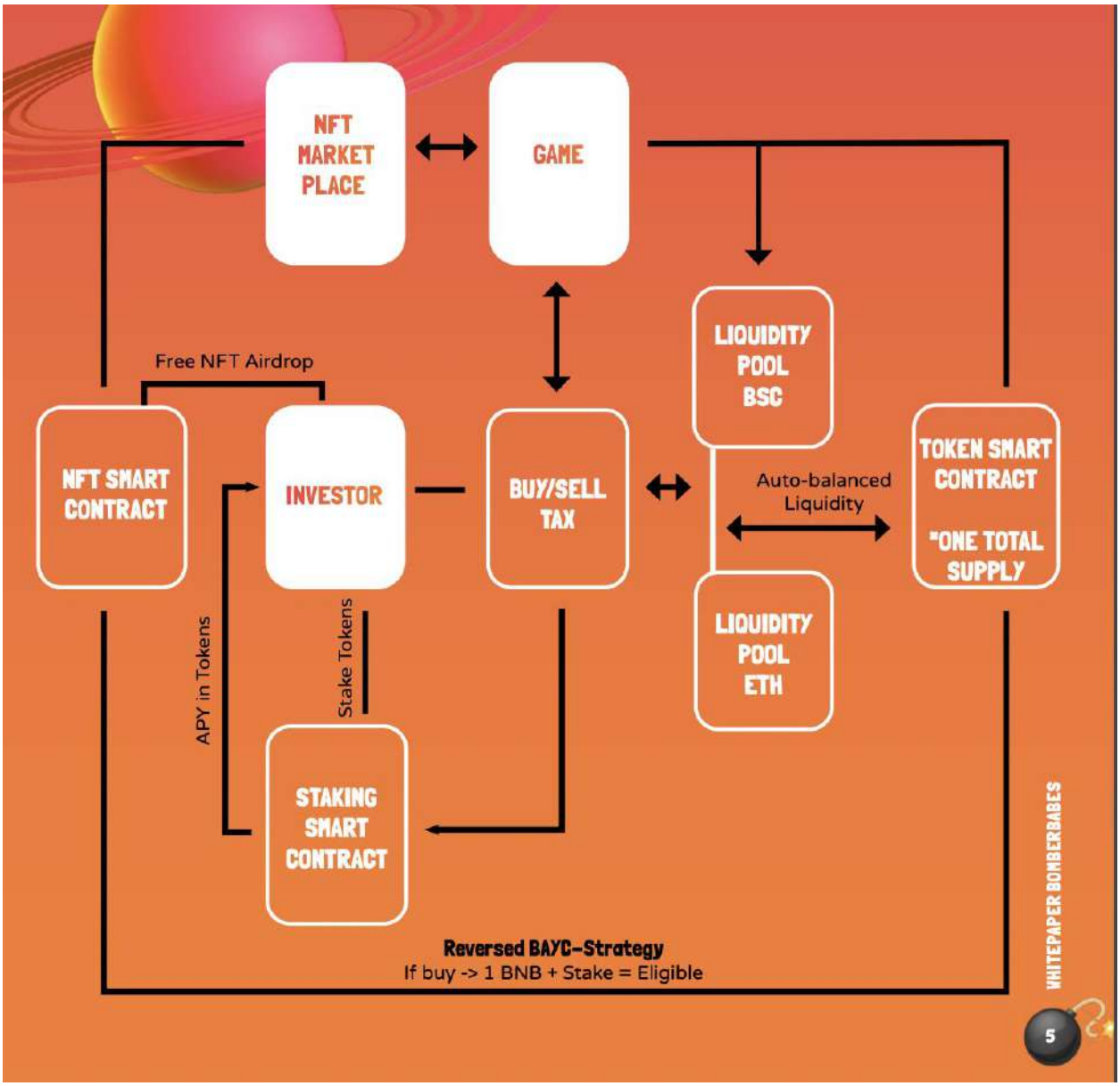
In simple terms, the staking rewards (rebase rewards) are distributed every 60 mins backed by the Bomber Babes parameter, thus ensuring a high and stable interest rate for bomber Babes holders.

Sustainable mechanism

The sustainability fee of 5% when buying and selling for marketing, and 3% when buying and selling for development is what allows Bomber Babes to promote the token and use funds to further the development of the platform. Tokens will be swapped into BNB and will be sent to the marketing wallet and development. This way, Bomber Babes 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 2% when buying and selling, is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.

The Flow


NFT MARKET PLACE
GAME
LIQUIDITY POOL BSC
Free NFT Airdrop
NFT SMART CONTRACT
INVESTOR
BUY/SELL TAX
Auto-balanced Liquidity
TOKEN SMART CONTRACT
*ONE TOTAL SUPPLY
LIQUIDITY POOL ETH
APY in Tokens
Stake Tokens
STAKING SMART CONTRACT
Reversed BAYC-Strategy
If buy -> 1 BNB + Stake = Eligible
WHITEPAPER BOMBERBABES
5

ABOUT THE GAME

The Game REVIVAL of the year!

The original Bomberman is one of the most played retro games when it comes to the early 90's.

That's why we set our goal to bring back the old times and make it even more creative and enjoyable! Play BOMBERBABES and earn BNB – bet on your win and grab the pot!

Additionally we are drawing the fanciest NFT Collection Defi has ever seen - stylish, building collectors value and unique hooks.

- Play to earn BNB
- Staking
- NFT's with UseCase
- NFT marketplace
- Available on BSC & ETH
- One supply, same price

COMMUNITY

Bomberbabes P2E

Multiplayer Arcade Game 1v1 / 2v2 / 4 player / Rumble Multiplayer

EARN BNB

Power Ups

Power-ups will be introduced one by one to the game in the later stages.

FUN & INTERACTIVITY

Traps

Traps will be introduced one by one to the game in the later stages.

STAY ALERT

NFT MARKETPLACE

Buy the game asset to add unique abilities to the gaming environment.

SELL YOUR NFT

Potential to grow with score points

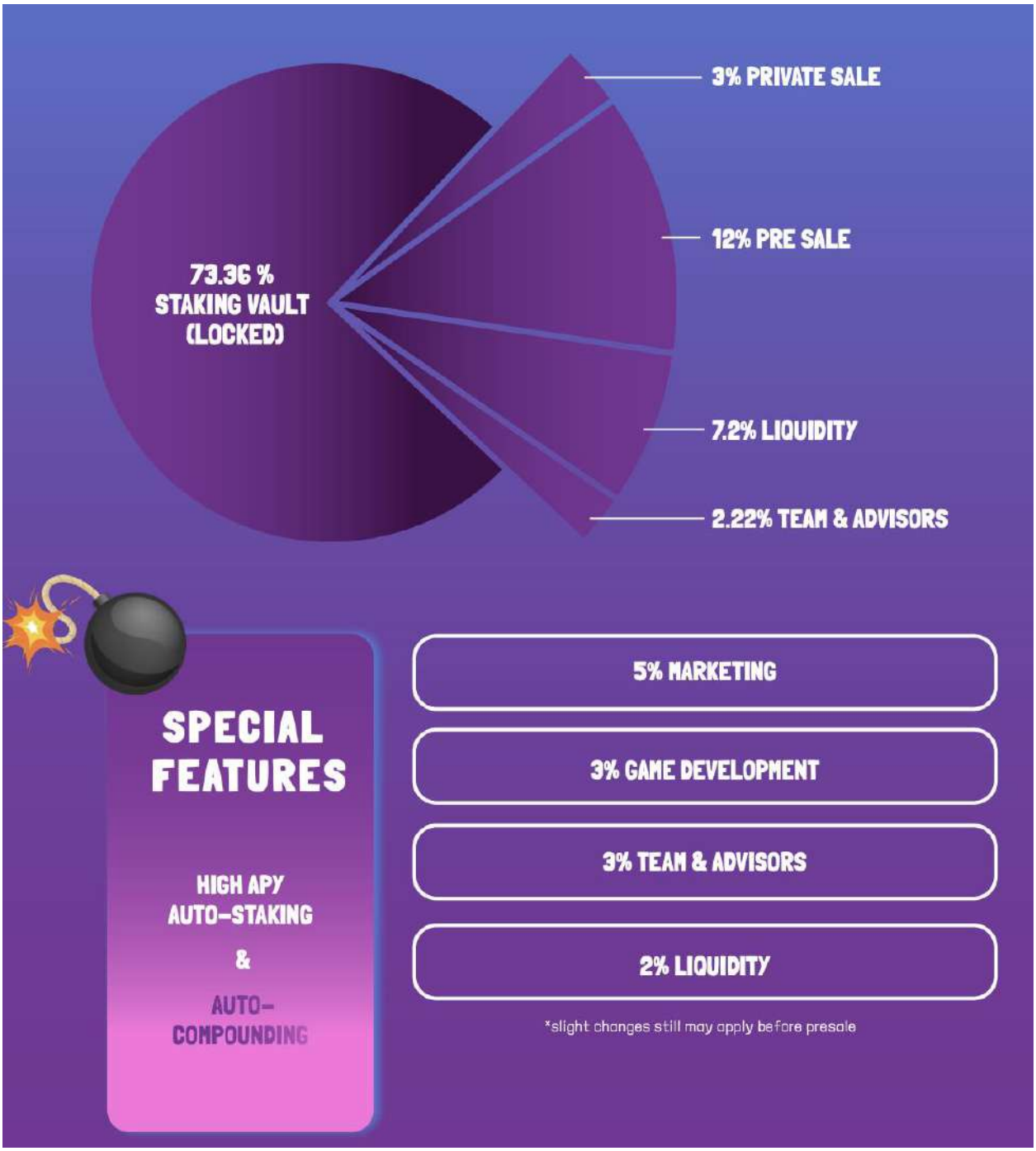
1.	Project efficiency	10/10
2.	Project uniqueness	9/10
3	Information quality	10/10
4	Service quality	9/10
5	System quality	10/10
6	Impact on the community	10/10
7	Impact on the business	10/10
8	Preparing for the future	10/10
Total Points		**9.75/10**

Contract details

Token contract details for 9th April 2022

Contract name	Bomber Babes
Contract address	0x4B78bD1e38Fc5d2Dd43a0c9326A20E7f9099C95a
Token supply	10,000,000
Token ticker	$BABES
Decimals	5
Token holders	1
Transaction count	2
Auto liquidity receiver	0xa6b390212ff29a27fcb00507cc5fc4cd9d55164e
Development Receiver	0xd3cf4215687cceed6783212ae8d70057d9f2df5d
Marketing Receiver	0xc0b9cf5a5da5076e374f8c4c5fa9929ea169e555
Vault	0x134885861e2917e8c15ee3a30e18050ea8f029fb
Contract deployer address	0xc5343208819F9Ef830e471E8Cf3d79cb994B7590
Contract's current owner address	0xe86791647155e489c0867580263e0c526951879d

Token Distribution



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	High Centralization Risk
		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

The 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 their visibility and mutability.

Contract	Type	Bases		
L	Function Name	Visibility	Mutability	Modifiers
SafeMathInt	**Library**			
L	mul	Internal 🔒		
L	div	Internal 🔒		
L	sub	Internal 🔒		
L	add	Internal 🔒		
L	abs	Internal 🔒		
SafeMath	**Library**			
L	add	Internal 🔒		
L	sub	Internal 🔒		
L	sub	Internal 🔒		
L	mul	Internal 🔒		
L	div	Internal 🔒		
L	div	Internal 🔒		
L	mod	Internal 🔒		

IERC20	Interface			
L	totalSupply	External ❗		NO❗
L	balanceOf	External ❗		NO❗
L	allowance	External ❗		NO❗
L	transfer	External ❗	🛑	NO❗
L	approve	External ❗	🛑	NO❗
L	transferFrom	External ❗	🛑	NO❗
IPancakeSwap Pair	**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❗
IPancakeSwap Router	**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❗
L	removeLiquidityETHSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
L	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
L	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
L	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💵	NO❗
L	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑	NO❗

IPancakeSwap Factory	**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❗
Ownable	**Implementation**			
L		Public ❗	🛑	NO❗
L	owner	Public ❗		NO❗
L	isOwner	Public ❗		NO❗
L	renounceOwnership	Public ❗	🛑	onlyOwner
L	transferOwnership	Public ❗	🛑	onlyOwner
L	_transferOwnership	Internal 🔒	🛑	
ERC20Detailed	**Implementation**	**IERC20**		
L		Public ❗	🛑	NO❗
L	name	Public ❗		NO❗

L	symbol	Public ❗		NO❗
L	decimals	Public ❗		NO❗
BomberBabes	**Implementation**	**ERC20Detailed, Ownable**		
L		Public ❗	🛑	ERC20Detailed Ownable
L	rebase	Internal 🔒	🛑	
L	transfer	External ❗	🛑	validRecipient
L	transferFrom	External ❗	🛑	validRecipient
L	_basicTransfer	Internal 🔒	🛑	
L	_transferFrom	Internal 🔒	🛑	
L	takeFee	Internal 🔒	🛑	
L	addLiquidity	Internal 🔒	🛑	swapping
L	swapBack	Internal 🔒	🛑	swapping
L	withdrawAllToMarketing	External ❗	🛑	swapping onlyOwner
L	shouldTakeFee	Internal 🔒		
L	shouldRebase	Internal 🔒		
L	shouldAddLiquidity	Internal 🔒		
L	shouldSwapBack	Internal 🔒		
L	setAutoRebase	External ❗	🛑	onlyOwner

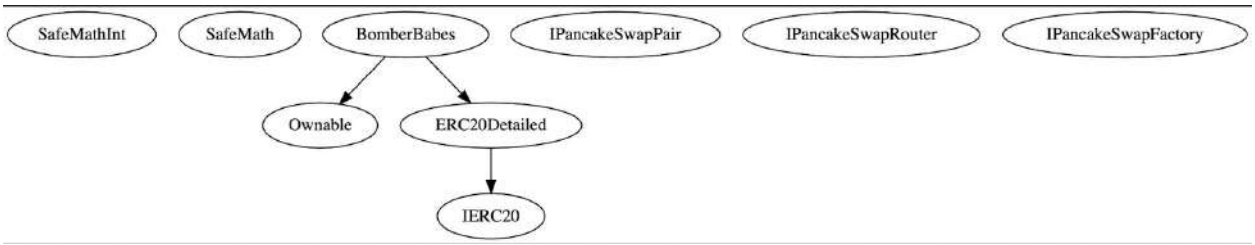
L	setAutoAddLiquidity	External ❗	🛑	onlyOwner
L	allowance	External ❗		NO❗
L	decreaseAllowance	External ❗	🛑	NO❗
L	increaseAllowance	External ❗	🛑	NO❗
L	approve	External ❗	🛑	NO❗
L	checkFeeExempt	External ❗		NO❗
L	getCirculatingSupply	Public ❗		NO❗
L	isNotInSwap	External ❗		NO❗
L	manualSync	External ❗	🛑	NO❗
L	setFeeReceivers	External ❗	🛑	onlyOwner
L	getLiquidityBacking	Public ❗		NO❗
L	setFeeStructure	External ❗	🛑	onlyOwner
L	setMaxSupply	External ❗	🛑	onlyOwner
L	setRebaseRate	External ❗	🛑	onlyOwner
L	setWhitelist	External ❗	🛑	onlyOwner
L	setBotBlacklist	External ❗	🛑	onlyOwner
L	setPairAddress	Public ❗	🛑	onlyOwner
L	setLP	External ❗	🛑	onlyOwner
L	totalSupply	External ❗		NO❗
L	balanceOf	External ❗		NO❗
L	isContract	Internal 🔒		

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

- **Centralization risk**

High severity centralization issues

❖ The owner can change all fees without maximum limit

```
ftrace | funcSig
function setFeeStructure(
    uint256 _liquidityFee,
    uint256 _marketingFee,
    uint256 _developmentFee,
    uint256 _sellFee,
    uint256 _vaultFee
) external onlyOwner {
    liquidityFee = _liquidityFee;
    marketingFee = _marketingFee;
    developmentFee = _developmentFee;
    sellFee = _sellFee;
    vaultFee = _vaultFee;
    totalFee = liquidityFee.add(marketingFee).add(developmentFee).add(
        vaultFee
    );
}
```

Owner privileges

- The owner can swap contract tokens to bnb and send them to marketing wallet

```
ftrace | funcSig
function withdrawAllToMarketing() external swapping onlyOwner {
    uint256 amountToSwap = _gonBalances[address(this)].div(
        _gonsPerFragment
    );
    require(
        amountToSwap > 0,
        "There is no token deposited in token contract"
    );
    address[] memory path = new address[](2);
    path[0] = address(this);
    path[1] = router.WETH();
    router.swapExactTokensForETHSupportingFeeOnTransferTokens(
        amountToSwap,
        0,
        path,
        marketingReceiver,
        block.timestamp
    );
}
```

- The owner can enable/disable auto rebase

```
ftrace | funcSig
function setAutoRebase(bool _flag⬆) external onlyOwner {
    if (_flag⬆) {
        _autoRebase = _flag⬆;
        _lastRebasedTime = block.timestamp;
    } else {
        _autoRebase = _flag⬆;
    }
}
```

❖ The owner can enable/disable liquidity adding

```
ftrace | funcSig
function setAutoAddLiquidity(bool _flag) external onlyOwner {
    if (_flag) {
        _autoAddLiquidity = _flag;
        _lastAddLiquidityTime = block.timestamp;
    } else {
        _autoAddLiquidity = _flag;
    }
}
```

❖ The owner can change all fee receiver wallets

```
ftrace | funcSig
function setFeeReceivers(
    address _autoLiquidityReceiver,
    address _marketingReceiver,
    address _developmentReceiver,
    address _vault
) external onlyOwner {
    autoLiquidityReceiver = _autoLiquidityReceiver;
    marketingReceiver = _marketingReceiver;
    developmentReceiver = _developmentReceiver;
    vault = _vault;
}
```

- The owner can change all fees

```
ftrace | funcSig
function setFeeStructure(
    uint256 _liquidityFee,
    uint256 _marketingFee,
    uint256 _developmentFee,
    uint256 _sellFee,
    uint256 _vaultFee
) external onlyOwner {
    liquidityFee = _liquidityFee;
    marketingFee = _marketingFee;
    developmentFee = _developmentFee;
    sellFee = _sellFee;
    vaultFee = _vaultFee;
    totalFee = liquidityFee.add(marketingFee).add(developmentFee).add(
        vaultFee
    );
}
```

- The owner can change max supply

```
ftrace | funcSig
function setMaxSupply(uint256 supply) external onlyOwner {
    maxSupply = supply;
}
```

- The owner can change rebase rate (this will change APY)

```
ftrace | funcSig
function setRebaseRate(uint256 rate) external onlyOwner {
    rebaseRate = rate;
}
```

❖ The owner can exclude wallets from fee (once exclude, cannot include again)

```
ftrace | funcSig
function setWhitelist(address _addr) external onlyOwner {
    _isFeeExempt[_addr] = true;
}
```

❖ The owner can block/unblock contracts

```
ftrace | funcSig
function setBotBlacklist(address _botAddress, bool _flag)
    external
    onlyOwner
{
    require(
        isContract(_botAddress),
        "only contract address, not allowed exteranlly owned account"
    );
    blacklist[_botAddress] = _flag;
}
```

❖ The owner can change pair address and pair contract

```
ftrace | funcSig
function setPairAddress(address _pairAddress) public onlyOwner {
    pairAddress = _pairAddress;
}

ftrace | funcSig
function setLP(address _address) external onlyOwner {
    pairContract = IPancakeSwapPair(_address);
}
```

Audit conclusion

RugFreeCoins team has performed in-depth testings, line by line manual code review, and automated audit of the smart contract. The smart contract was analyzed mainly for common smart contract vulnerabilities, exploits, manipulations, and hacks. According to the smart contract audit.

Smart contract functional Status: **PASSED**

Number of risk issues: **1**

Solidity code functional issue level: **PASSED**

Number of owner privileges: **10**

Centralization risk correlated to the active owner: **HIGH**

Smart contract active ownership: **YES**