RugFreeCoins Audit

Safemoonomics Token

Smart Contract Security Audit

January 3, 2022

Contents

Audit details

Audited project

Safemoonomics Token

Contract Address

0xeb206d5449563055d7d5b5933d891f777b319750

Client contact

Safemoonomics Token Team

Blockchain

Binance smart chain

Project website

https://www.safemoonomicstoken.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 Safemoonomics Token to perform an audit of the smart contract.

https://bscscan.com/token/0xeb206d5449563055d7d5b5933d891f777b319750

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

Safemoonomics Token is a token built on the Binance Smart Chain that investors have a chance to earn SafeMoon rewards. Each transaction, purchase incurs a 12% fee, and sell incur a 16% fee.

Features

- 5% of each transaction when buying and 7% when selling gets sent amongst all holders in Safemoon rewards. The holders will be eligible to receive Safemoon, everyone hour, and rewards are proportional to how many tokens each individual holds.

- The **sustainability fee of 6% when buying and selling for marketing** is what allows Safemoonomics to hold the aforementioned promise. Tokens will be swapped into BNBs and will be sent to a marketing wallet, which will be allocated for marketing. This way, Safemoonomics 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 1% when buying and 3% when selling,** which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity. This is a key element for decentralized exchanges like Pancakeswap.

Tokenomics

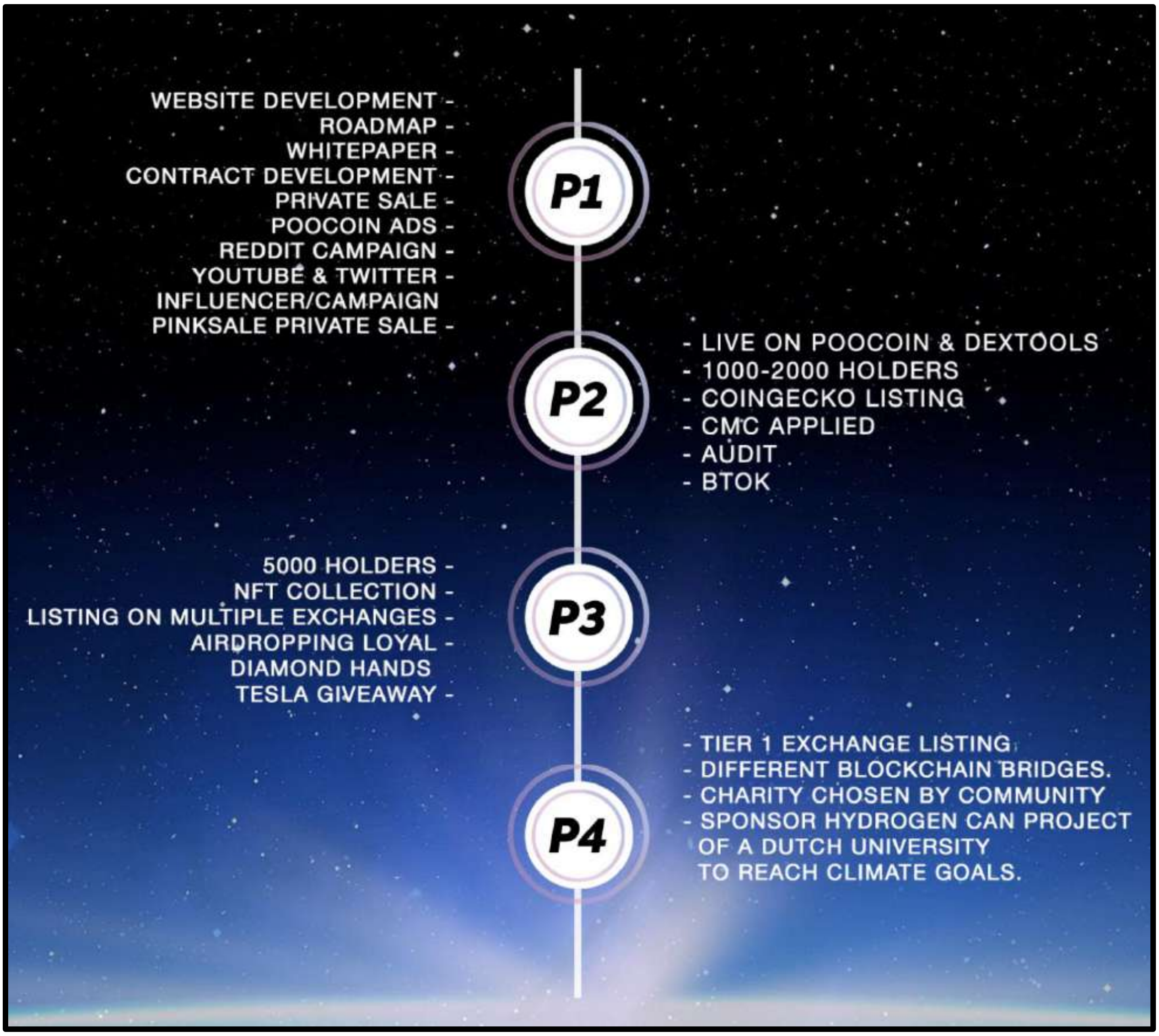
12% fee when buying

- 5% of trade goes to holders pockets in Safemoon.
- 6% of trade goes to the marketing wallet.
- 1% of trade goes to the liquidity pool.

16% fee when selling

- 7% of trade goes to holders pockets in Safemoon.
- 6% of trade goes to the marketing wallet.
- 3% of trade goes to the liquidity pool.

Roadmap



P1

WEBSITE DEVELOPMENT -
ROADMAP -
WHITEPAPER -
CONTRACT DEVELOPMENT -
PRIVATE SALE -
POOCOIN ADS -
REDDIT CAMPAIGN -
YOUTUBE & TWITTER -
INFLUENCER/CAMPAIGN
PINKSALE PRIVATE SALE -

P2

- LIVE ON POOCOIN & DEXTOOLS
- 1000-2000 HOLDERS
- COINGECKO LISTING
- CMC APPLIED
- AUDIT
- BTOK

P3

5000 HOLDERS -
NFT COLLECTION -
LISTING ON MULTIPLE EXCHANGES -
AIRDROPPING LOYAL -
DIAMOND HANDS
TESLA GIVEAWAY -

P4

- TIER 1 EXCHANGE LISTING
- DIFFERENT BLOCKCHAIN BRIDGES.
- CHARITY CHOSEN BY COMMUNITY
- SPONSOR HYDROGEN CAN PROJECT OF A DUTCH UNIVERSITY TO REACH CLIMATE GOALS.

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 Safemoon by holding tokens.
- ❖ Anyone who's interested in trading tokens.
- ❖ Anyone who's interested in taking part in the weekly giveaways and rewards.
- ❖ Anyone who's interested in taking part with the future plans of the Safemoonomics token.
- ❖ Anyone who's interested in making financial transactions with any other party using Safemoonomics token as the currency.

Core concept

The Safemoon reward system

5% of each transaction when buying and 7% when selling gets sent amongst all holders in Safemoon rewards. The holders will be eligible to receive Safemoon, every one hour, and rewards are proportional to how many tokens each individual holds.

Sustainable mechanism

The **sustainability fee of 6% when buying and selling to the marketing wallet will be allocated for marketing, development, and team**. This is what allows Safemoonomics Token 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 a marketing and gaming pool wallet per transaction. This way, Safemoonomics Token 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 1% when buying and 3% when selling, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.

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	9/10
7	Impact on the business	9/10
8	Preparing for the future	8/10
Total Points		**8.25/10**

Contract details

Token contract details for 03rd January 2022

Contract name	SafeMoonomics
Contract address	0xeb206d5449563055d7d5b5933d891f777b319750
Token supply	1,000,000,000,000,000
Token ticker	SAFEMOONOMICS
Decimals	9
Token holders	3
Transaction count	4
Dividend distributor	c0x1e88d2cef98ad1b0f10c54f9717604bf0e6c96e3
Marketing wallet	0x9d79bc333e770d48f220658b72a7a1d9f0c3a4e2
Contract deployer address	0xD9EA912E0169388dfAe2fADfEfaaca85dC505066
Contract's current owner address	0xbb90e44f6225da9b621fdb32c47ac04316b1d7c5

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**			

L	factory	External ❗		NO❗
L	WETH	External ❗		NO❗
L	addLiquidity	External ❗	🛑	NO❗
L	addLiquidityETH	External ❗	💵	NO❗
L	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
L	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💵	NO❗
L	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
IDividendDistributor	**Interface**			
L	setDistributionCriteria	External ❗	🛑	NO❗
L	setShare	External ❗	🛑	NO❗
L	deposit	External ❗	💵	NO❗
L	process	External ❗	🛑	NO❗
L	purge	External ❗	🛑	NO❗
DividendDistributor	**Implementation**	**IDividendDistributor**		
L		Public ❗	🛑	NO❗

L	setDistributionCriteria	External ❗	🛑	onlyToken
L	purge	External ❗	🛑	onlyToken
L	setShare	External ❗	🛑	onlyToken
L	deposit	External ❗	💵	onlyToken
L	process	External ❗	🛑	onlyToken
L	shouldDistribute	Internal 🔒		
L	distributeDividend	Internal 🔒	🛑	
L	claimDividend	External ❗	🛑	NO❗
L	getUnpaidEarnings	Public ❗		NO❗
L	getCumulativeDividends	Internal 🔒		
L	addShareholder	Internal 🔒	🛑	
L	removeShareholder	Internal 🔒	🛑	
SMNToken	**Implementation**	**IBEP20, Auth**		
L		Public ❗	🛑	Auth
L		External ❗	💵	NO❗
L	totalSupply	External ❗		NO❗
L	decimals	External ❗		NO❗

└	symbol	External ❗		NO❗
└	name	External ❗		NO❗
└	getOwner	External ❗		NO❗
└	balanceOf	Public ❗		NO❗
└	allowance	External ❗		NO❗
└	approve	Public ❗	🛑	NO❗
└	approveMax	External ❗	🛑	NO❗
└	transfer	External ❗	🛑	NO❗
└	transferFrom	External ❗	🛑	NO❗
└	_transferFrom	Internal 🔒	🛑	
└	_basicTransfer	Internal 🔒	🛑	
└	shouldTakeFee	Internal 🔒		
└	takeFee	Internal 🔒	🛑	
└	shouldSwapBack	Internal 🔒		
└	clearStuckBalance	External ❗	🛑	onlyOwner
└	updateBuylFees	Public ❗	🛑	onlyOwner
└	updateSellFees	Public ❗	🛑	onlyOwner
└	updateSwapPercentage	Public ❗	🛑	onlyOwner

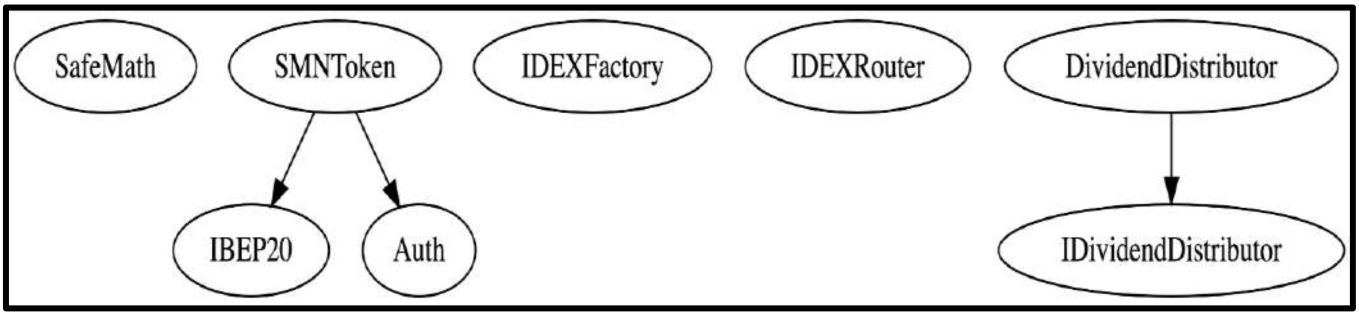
└	tradingStatus	Public ❗	🛑	onlyOwner
└	whitelistPreSale	Public ❗	🛑	onlyOwner
└	cooldownEnabled	Public ❗	🛑	onlyOwner
└	purgeBeforeSwitch	Public ❗	🛑	onlyOwner
└	switchToken	Public ❗	🛑	onlyOwner
└	___claimRewards	Public ❗	🛑	NO❗
└	claimProcess	Public ❗	🛑	NO❗
└	swapBack	Internal 🔒	🛑	swapping
└	swapAndLiquify	Private 🔐	🛑	
└	swapTokensForEth	Private 🔐	🛑	
└	addLiquidity	Private 🔐	🛑	
└	setIsDividendExempt	External ❗	🛑	authorized
└	setIsFeeExempt	External ❗	🛑	authorized
└	setIsTimelockExempt	External ❗	🛑	authorized
└	setFeeReceivers	External ❗	🛑	authorized
└	setSwapBackSettings	External ❗	🛑	authorized
└	setDistributionCriteria	External ❗	🛑	authorized
└	setDistributorSettings	External ❗	🛑	authorized

L	getCirculatingSupply	Public ❗		NO❗
L	getLiquidityBacking	Public ❗		NO❗
L	isOverLiquified	Public ❗		NO❗
L	claimTokens	External ❗	🛑	onlyOwner

Legend

Symbol	Meaning
🛑	**Function can modify state**
💵	**Function is payable**

Inheritance Hierarchy



Security issue checking status

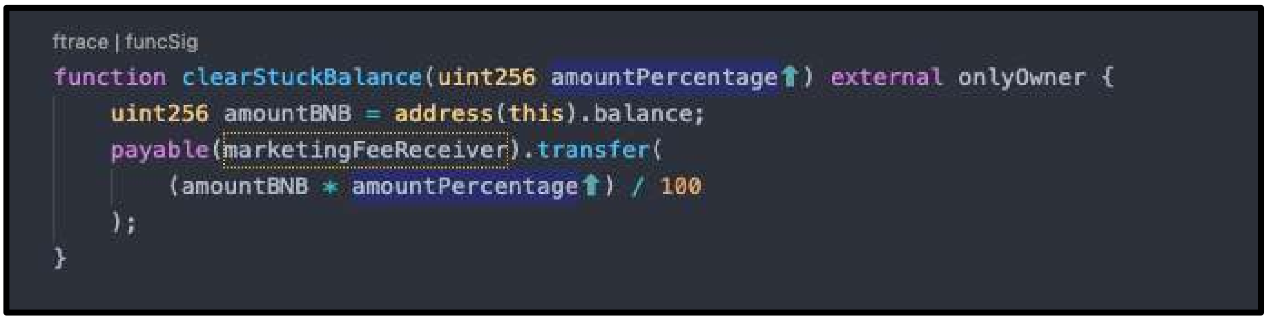
- **High severity issues**
  No high severity issues found.

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

- **Low severity issues**
  No low severity issues found.

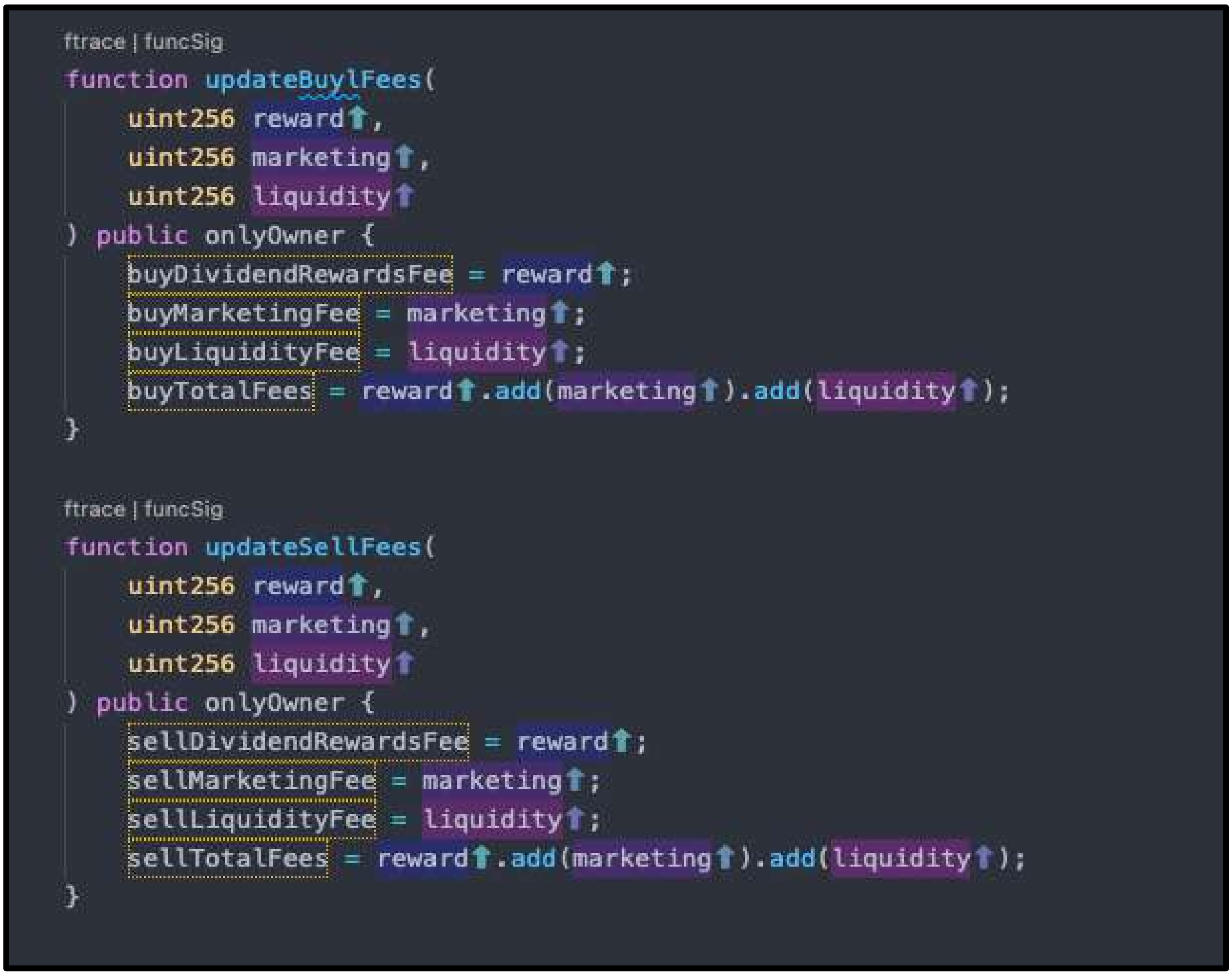
Owner privileges

❖ The owner can get contract bnb balance to marketing wallet.



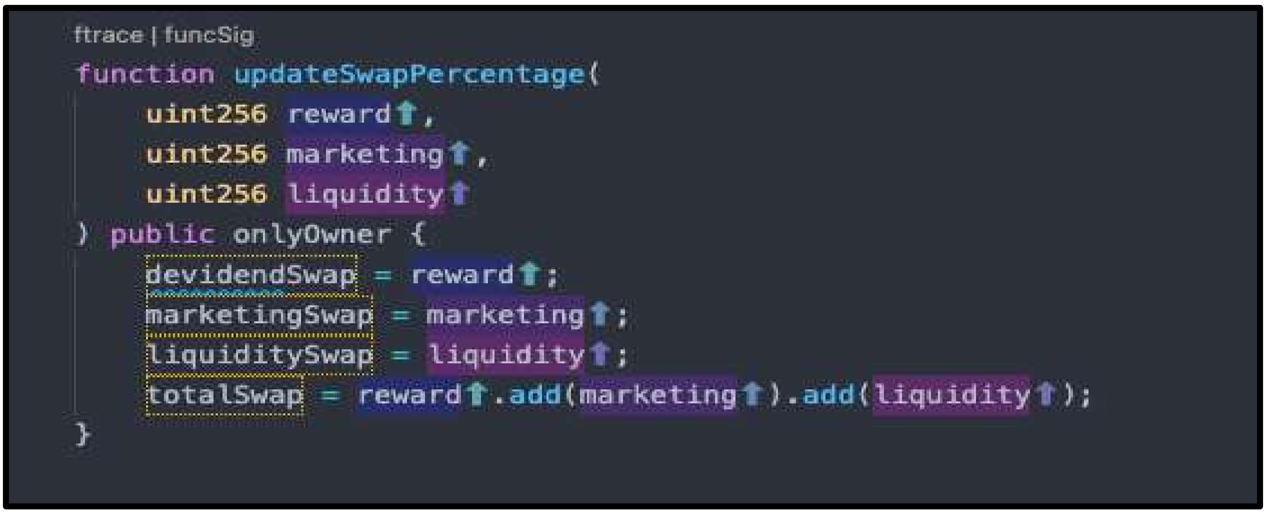
```
ftrace | funcSig
function clearStuckBalance(uint256 amountPercentage) external onlyOwner {
    uint256 amountBNB = address(this).balance;
    payable(marketingFeeReceiver).transfer(
        (amountBNB * amountPercentage) / 100
    );
}
```

❖ The owner can update all buy and sell fees.



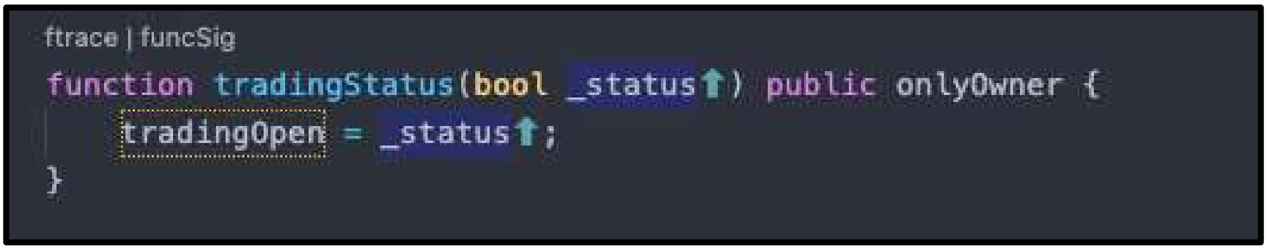
```
ftrace | funcSig
function updateBuylFees(
    uint256 reward,
    uint256 marketing,
    uint256 liquidity
) public onlyOwner {
    buyDividendRewardsFee = reward;
    buyMarketingFee = marketing;
    buyLiquidityFee = liquidity;
    buyTotalFees = reward.add(marketing).add(liquidity);
}

ftrace | funcSig
function updateSellFees(
    uint256 reward,
    uint256 marketing,
    uint256 liquidity
) public onlyOwner {
    sellDividendRewardsFee = reward;
    sellMarketingFee = marketing;
    sellLiquidityFee = liquidity;
    sellTotalFees = reward.add(marketing).add(liquidity);
}
```

- The owner can update swap percentages.



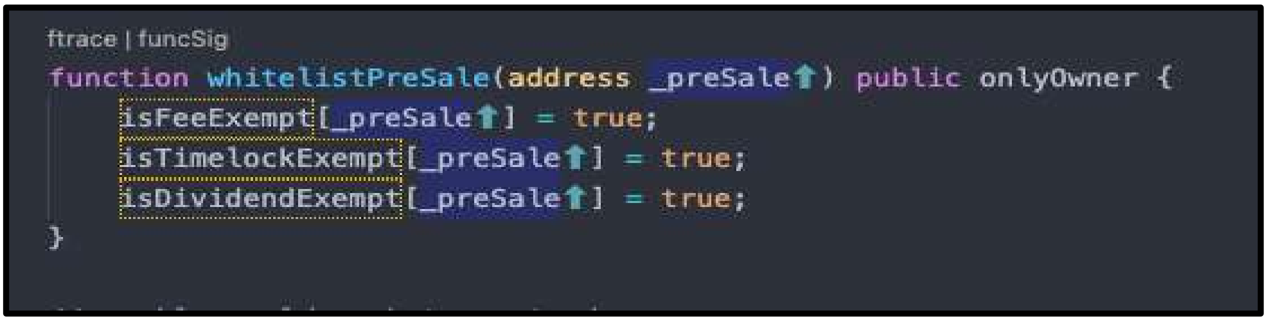
```
ftrace | funcSig
function updateSwapPercentage(
    uint256 reward,
    uint256 marketing,
    uint256 liquidity
) public onlyOwner {
    devidendSwap = reward;
    marketingSwap = marketing;
    liquiditySwap = liquidity;
    totalSwap = reward.add(marketing).add(liquidity);
}
```

- The owner can enable/disable trading.



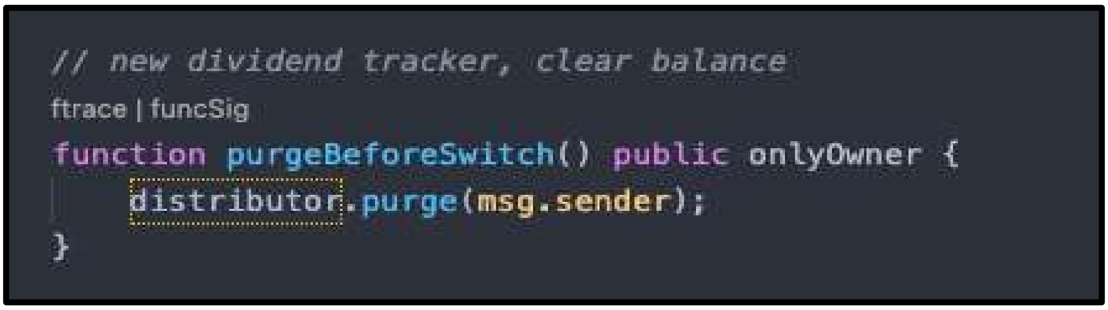
```
ftrace | funcSig
function tradingStatus(bool _status) public onlyOwner {
    tradingOpen = _status;
}
```

- The owner can whitelist pre-sale address.



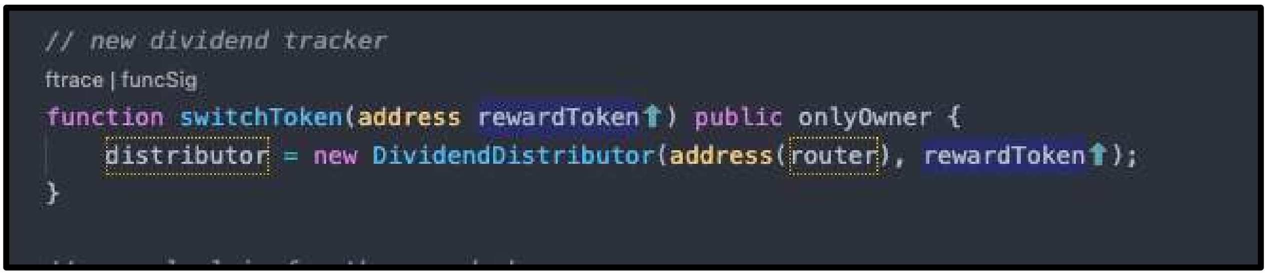
```
ftrace | funcSig
function whitelistPreSale(address _preSale) public onlyOwner {
    isFeeExempt[_preSale] = true;
    isTimelockExempt[_preSale] = true;
    isDividendExempt[_preSale] = true;
}
```

- The owner can get tokens in dividend tracker to the owner wallet.



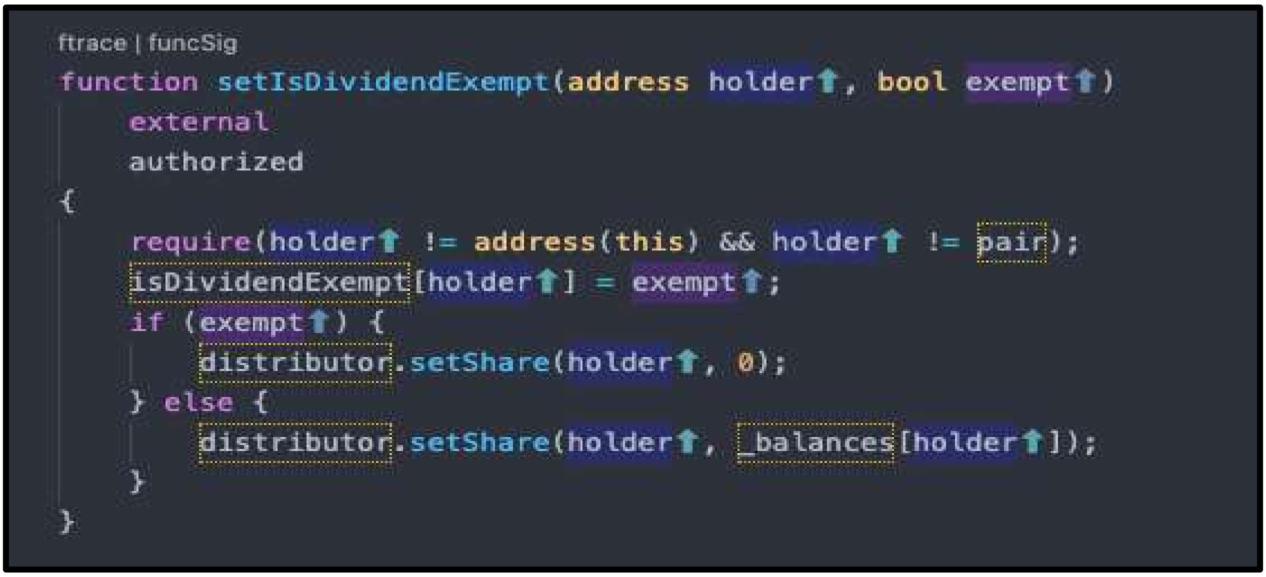
```
// new dividend tracker, clear balance
ftrace | funcSig
function purgeBeforeSwitch() public onlyOwner {
    distributor.purge(msg.sender);
}
```

- The owner can change the reward token address.



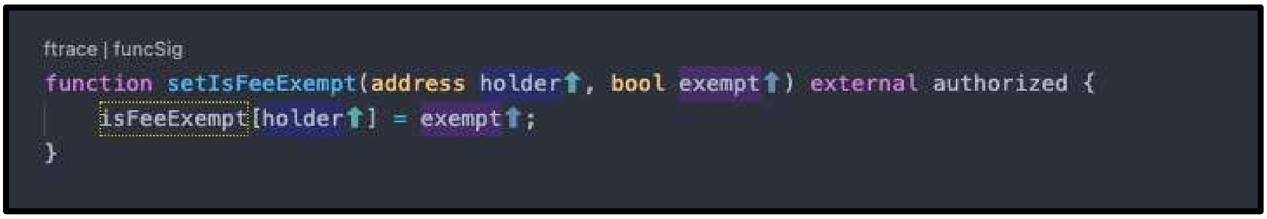
```
// new dividend tracker
ftrace | funcSig
function switchToken(address rewardToken) public onlyOwner {
    distributor = new DividendDistributor(address(router), rewardToken);
}
```

- The owner can exclude wallets from dividend.



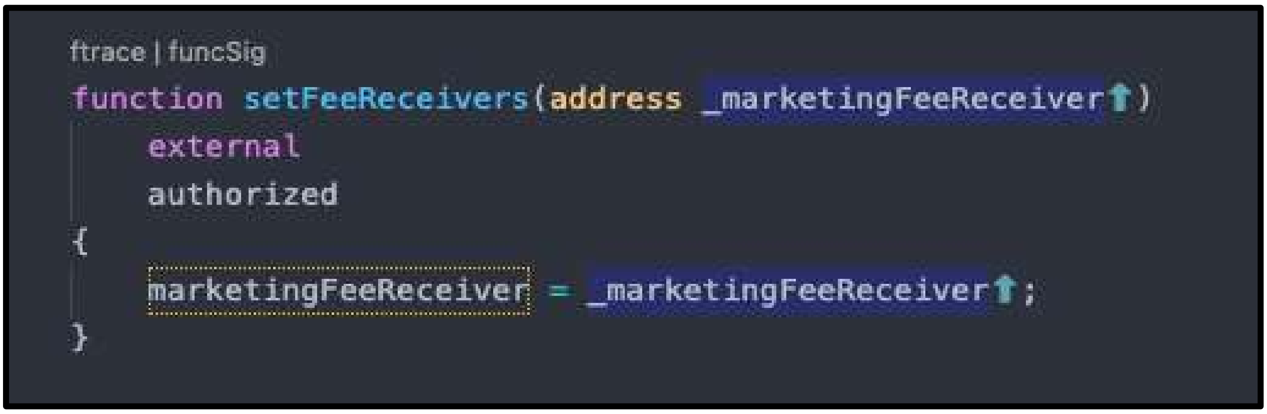
```
ftrace | funcSig
function setIsDividendExempt(address holder, bool exempt)
    external
    authorized
{
    require(holder != address(this) && holder != pair);
    isDividendExempt[holder] = exempt;
    if (exempt) {
        distributor.setShare(holder, 0);
    } else {
        distributor.setShare(holder,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

- The owner can exclude wallets from fees.



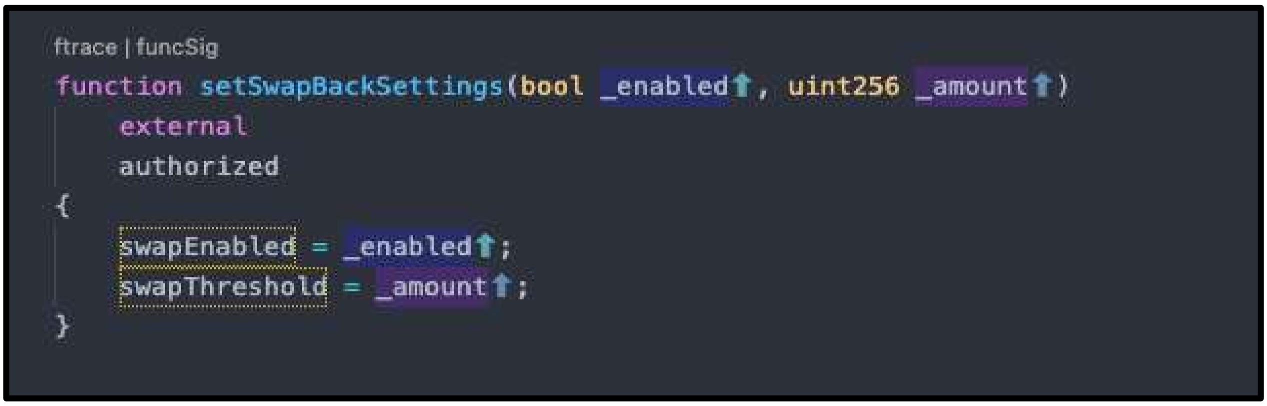
```
ftrace | funcSig
function setIsFeeExempt(address holder, bool exempt) external authorized {
    isFeeExempt[holder] = exempt;
}
```

- The owner can change the marketing wallet.



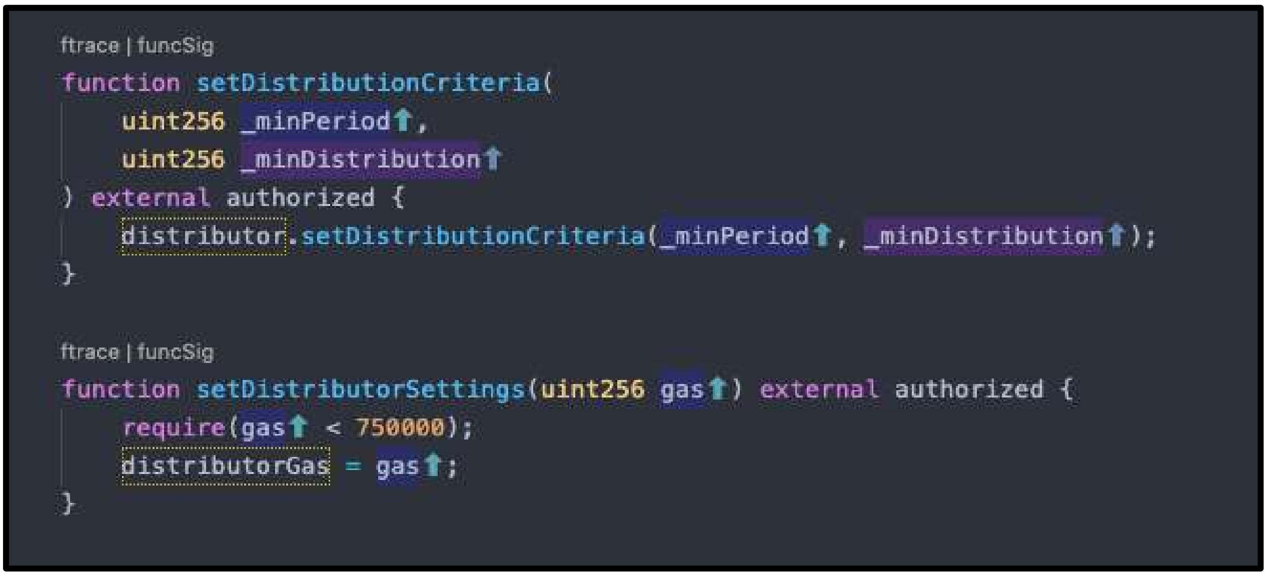
```
ftrace | funcSig
function setFeeReceivers(address _marketingFeeReceiver)
    external
    authorized
{
    marketingFeeReceiver = _marketingFeeReceiver;
}
```

- The owner can enable/disable swap.



```
ftrace | funcSig
function setSwapBackSettings(bool _enabled, uint256 _amount)
    external
    authorized
{
    swapEnabled = _enabled;
    swapThreshold = _amount;
}
```

- The owner can change distribution settings.



```
ftrace | funcSig
function setDistributionCriteria(
    uint256 _minPeriod,
    uint256 _minDistribution
) external authorized {
    distributor.setDistributionCriteria(_minPeriod, _minDistribution);
}

ftrace | funcSig
function setDistributorSettings(uint256 gas) external authorized {
    require(gas < 750000);
    distributorGas = gas;
}
```

Audit conclusion

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