RugFreeCoins Audit

Emperors DAO Token

Smart Contract Security Audit

April 04, 2022

Contents

Audit details

Audited project
Emperors DAO Token

Contract Address

0x63564124A204EB8dbFc8C75B26892E72DE1fa663

Client contact

Emperors DAO Team

Blockchain

Binance smart chain

Project website

www.EmperorsDAO.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 the Emperors DAO Team to perform an audit of the smart contract.

https://bscscan.com/token/0x63564124A204EB8dbFc8C75B26892E72DE1fa663

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

Emperors DAO 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. IEmperors DAO will bring an unparallel, fixed APY of **365,700%,** onto the BSC blockchain while imposing profound ease, simplicity, and accessibility upon all Inu Base holders. Each transaction, purchase incurs an 11% fee, and sale incurs a 13% fee.

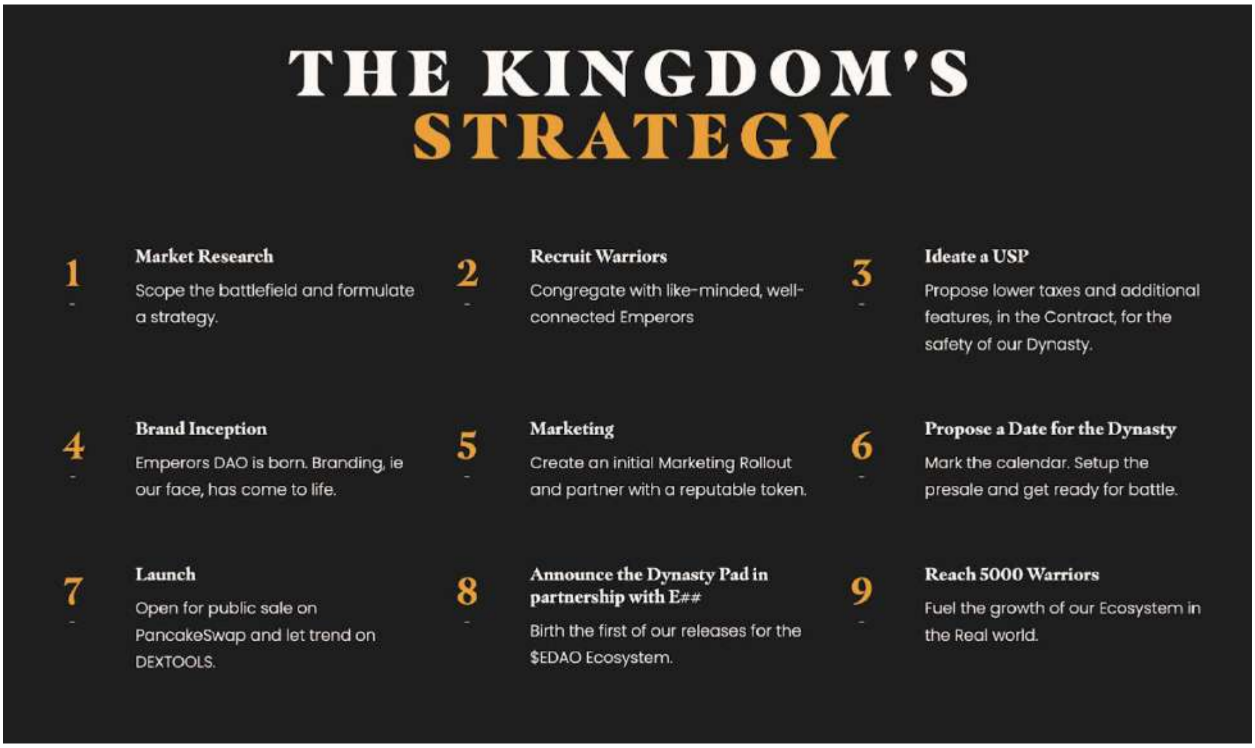
Features

- **2% of the buy and sale fees** are directed to the Dynasty funds which helps sustain and back the Staking Rewards provided by the Positive Rebase.

- The **sustainability fee of 4% when buying and 6% when selling for Emperor chest Treasury, which is allocated for marketing** is what allows Emperors DAO Token to hold the aforementioned promise. Tokens will be swapped into BNB and will be sent to a marketing wallet. This way, Emperors DAO 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.

- 2% of all Emperors DAO tokens traded are burnt in the Sacrifice wallet. The more that is traded, the more get put into the fire causing the fire pit to grow in size, larger and larger through self-fulfilling auto-compounding which in return acts to reduce the circulating supply of Emperors DAO and keep the Emperors DAO stable.

Roadmap



THE KINGDOM'S STRATEGY

1 - **Market Research**

Scope the battlefield and formulate a strategy.

2 - **Recruit Warriors**

Congregate with like-minded, well-connected Emperors

3 - **Ideate a USP**

Propose lower taxes and additional features, in the Contract, for the safety of our Dynasty.

4 - **Brand Inception**

Emperors DAO is born. Branding, ie our face, has come to life.

5 - **Marketing**

Create an initial Marketing Rollout and partner with a reputable token.

6 - **Propose a Date for the Dynasty**

Mark the calendar. Setup the presale and get ready for battle.

7 - **Launch**

Open for public sale on PancakeSwap and let trend on DEXTOOLS.

8 - **Announce the Dynasty Pad in partnership with E##**

Birth the first of our releases for the $EDAO Ecosystem.

9 - **Reach 5000 Warriors**

Fuel the growth of our Ecosystem in the Real world.

Tokenomics

11% fee when buying

- 2% of trade goes to Dynasty funds in tokens
- 4% of trade goes to the Emperor chest in BNB
- 2% trade goes to the sacrifice wallet (Burn)
- 2% of trade goes to the liquidity pool.

13% fee when selling

- 2% of trade goes to Dynasty funds in tokens
- 6% of trade goes to the Emperor chest in BNB
- 2% trade goes to the sacrifice wallet (Burn)
- 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 15 minutes.
- Anyone who's interested in receiving fixed interest of 365,700% per year.
- Anyone who's interested in taking part in the future plans of the Emperors DAO token.
- Anyone who's interested in making financial transactions with any other party using Emperors DAO as the currency.

Core concept

Reward mechanism

2% of when buying and selling are stored in the Dynasty Funds fund which helps sustain and back the staking rewards provided by the positive rebase.

Emperors DAO fund which is a separate wallet in the ecosystem. The Emperors DAO 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) which are distributed every 15 mins at a rate of 0.02368% are backed by the Emperors DAO parameter, thus ensuring a high and stable interest rate for Emperors DAO holders.

Sustainable mechanism

The sustainability fee of 4% when buying and 6% selling for Treasury that is allocated for marketing is what allows Emperors DAO 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 wallet. This way, Emperors DAO 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.

2% of Emperors DAO tokens from buying and selling traded are burnt in **Sacrifice**. The more that is traded the more gets put into the fire causing the fire pit to grow in size, larger and larger through self-fulfilling Auto-Compounding, reducing the circulating supply and keeping the Emperors DAO table.

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

Contract details

Token contract details for 4th April 2022

Contract name	Emperors DAO
Contract address	0x63564124A204EB8dbFc8C75B26892E72DE1fa663
Token supply	354,756.20425
Token ticker	$EDAO
Decimals	5
Token holders	3
Transaction count	5
Dunasty funds receiver	0x201cff062a1bdc833d8005d967af17275b87f0fb
Emperors chest receivers	0x4821bc8be45b98a6a177c28fc3292c1c45ceb0c9
Sacrifice	0xf4ff3306a7d14d1daf8e01be2c5c9065793570c6
Auto Liquidity Receiver	0xe37d55807c6bf67971628c750f9a218534ef8b5d
Contract deployer address	0xe37D55807c6bf67971628C750F9A218534EF8b5d
Contract’s current owner address	0x4821bc8be45b98a6a177c28fc3292c1c45ceb0c9

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

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**		
└	Function Name	Visibility	Mutability	Modifiers
SafeMathInt	**Library**			
└	mul	Internal 🔒		
└	div	Internal 🔒		
└	sub	Internal 🔒		
└	add	Internal 🔒		
└	abs	Internal 🔒		
SafeMath	**Library**			
└	add	Internal 🔒		
└	sub	Internal 🔒		
└	sub	Internal 🔒		
└	mul	Internal 🔒		
└	div	Internal 🔒		
└	div	Internal 🔒		
└	mod	Internal 🔒		

IERC20	**Interface**			
└	totalSupply	External ❗		NO❗
└	balanceOf	External ❗		NO❗
└	allowance	External ❗		NO❗
└	transfer	External ❗	🛑	NO❗
└	approve	External ❗	🛑	NO❗
└	transferFrom	External ❗	🛑	NO❗
IPancakeSwapPair	**Interface**			
└	name	External ❗		NO❗
└	symbol	External ❗		NO❗
└	decimals	External ❗		NO❗
└	totalSupply	External ❗		NO❗
└	balanceOf	External ❗		NO❗
└	allowance	External ❗		NO❗
└	approve	External ❗	🛑	NO❗
└	transfer	External ❗	🛑	NO❗
└	transferFrom	External ❗	🛑	NO❗
└	DOMAIN_SEPARATOR	External ❗		NO❗
└	PERMIT_TYPEHASH	External ❗		NO❗
└	nonces	External ❗		NO❗

└	permit	External ❗	🛑	NO❗
└	MINIMUM_LIQUIDITY	External ❗		NO❗
└	factory	External ❗		NO❗
└	token0	External ❗		NO❗
└	token1	External ❗		NO❗
└	getReserves	External ❗		NO❗
└	price0CumulativeLast	External ❗		NO❗
└	price1CumulativeLast	External ❗		NO❗
└	kLast	External ❗		NO❗
└	mint	External ❗	🛑	NO❗
└	burn	External ❗	🛑	NO❗
└	swap	External ❗	🛑	NO❗
└	skim	External ❗	🛑	NO❗
└	sync	External ❗	🛑	NO❗
└	initialize	External ❗	🛑	NO❗
IPancakeSwap Router	**Interface**			
└	factory	External ❗		NO❗
└	WETH	External ❗		NO❗
└	addLiquidity	External ❗	🛑	NO❗
└	addLiquidityETH	External ❗	💵	NO❗

└	removeLiquidity	External ❗	🛑	NO❗
└	removeLiquidityETH	External ❗	🛑	NO❗
└	removeLiquidityWithPermit	External ❗	🛑	NO❗
└	removeLiquidityETHWithPermit	External ❗	🛑	NO❗
└	swapExactTokensForTokens	External ❗	🛑	NO❗
└	swapTokensForExactTokens	External ❗	🛑	NO❗
└	swapExactETHForTokens	External ❗	💵	NO❗
└	swapTokensForExactETH	External ❗	🛑	NO❗
└	swapExactTokensForETH	External ❗	🛑	NO❗
└	swapETHForExactTokens	External ❗	💵	NO❗
└	quote	External ❗		NO❗
└	getAmountOut	External ❗		NO❗
└	getAmountIn	External ❗		NO❗
└	getAmountsOut	External ❗		NO❗
└	getAmountsIn	External ❗		NO❗
└	removeLiquidityETHSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
└	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
└	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
└	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💵	NO❗
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑	NO❗

IPancakeSwap Factory	**Interface**			
└	feeTo	External ❗		NO❗
└	feeToSetter	External ❗		NO❗
└	getPair	External ❗		NO❗
└	allPairs	External ❗		NO❗
└	allPairsLength	External ❗		NO❗
└	createPair	External ❗	🛑	NO❗
└	setFeeTo	External ❗	🛑	NO❗
└	setFeeToSetter	External ❗	🛑	NO❗
Ownable	**Implementation**			
└		Public ❗	🛑	NO❗
└	owner	Public ❗		NO❗
└	isOwner	Public ❗		NO❗
└	renounceOwnership	Public ❗	🛑	onlyOwner
└	transferOwnership	Public ❗	🛑	onlyOwner
└	_transferOwnership	Internal 🔒	🛑	
ERC20Detailed	**Implementation**	**IERC20**		
└		Public ❗	🛑	NO❗
└	name	Public ❗		NO❗

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

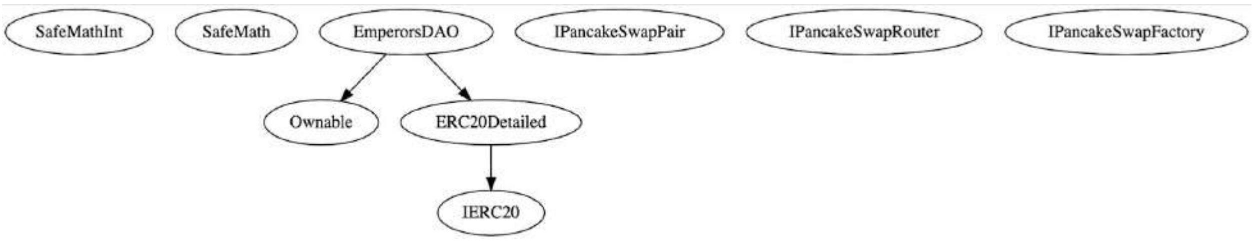
└	setAutoAddLiquidity	External ❗	🛑	onlyOwner
└	allowance	External ❗		NO❗
└	decreaseAllowance	External ❗	🛑	NO❗
└	increaseAllowance	External ❗	🛑	NO❗
└	approve	External ❗	🛑	NO❗
└	checkFeeExempt	External ❗		NO❗
└	getCirculatingSupply	Public ❗		NO❗
└	getCurrentDay	Public ❗		NO❗
└	getUserHoldLimit	Public ❗		NO❗
└	getUserSellLimit	Public ❗		NO❗
└	isNotInSwap	External ❗		NO❗
└	manualSync	External ❗	🛑	NO❗
└	setFeeReceivers	External ❗	🛑	onlyOwner
└	getLiquidityBacking	Public ❗		NO❗
└	setWhitelistMultiple	External ❗	🛑	onlyOwner
└	setBotBlacklist	External ❗	🛑	onlyOwner
└	setPairAddress	Public ❗	🛑	onlyOwner
└	setLP	External ❗	🛑	onlyOwner
└	setExcludeFromLimit	Public ❗	🛑	onlyOwner
└	setLimit	Public ❗	🛑	onlyOwner
└	totalSupply	External ❗		NO❗

└	balanceOf	External ❗		NO❗
└	isContract	Internal 🔒		
└	minZero	Private 🔐		
└		External ❗	💵	NO❗

Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

Inheritance Hierarchy



Security issue checking status

- **High severity issues**

No medium severity issues found.

- **Medium severity issues**

No medium severity issues found

- **Low severity issues**

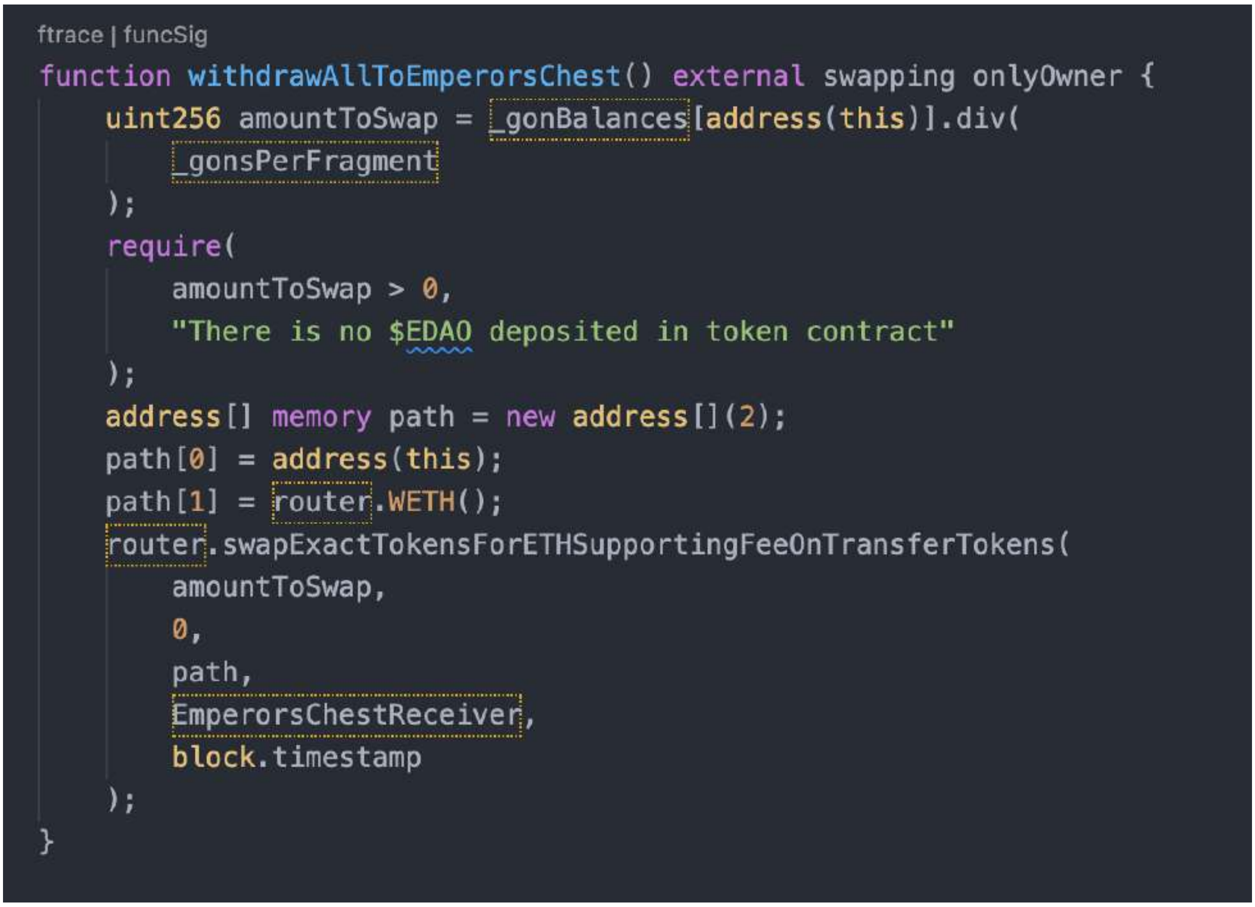
No low severity issues found

- **Centralization risk**

No Centralization issues found

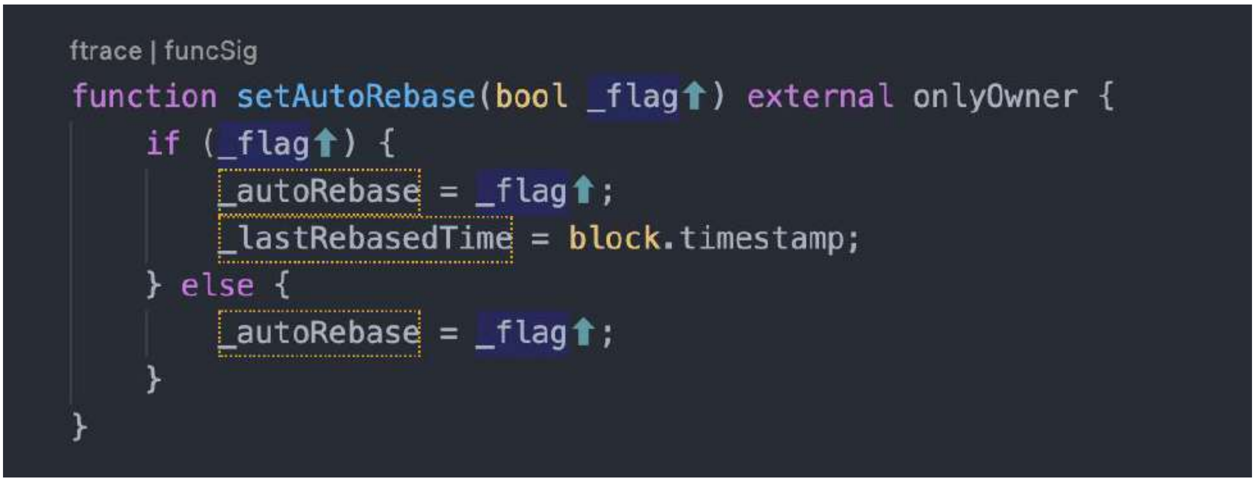
Owner privileges

❖ The owner can withdraw tokens in contract by swapping them into BNB



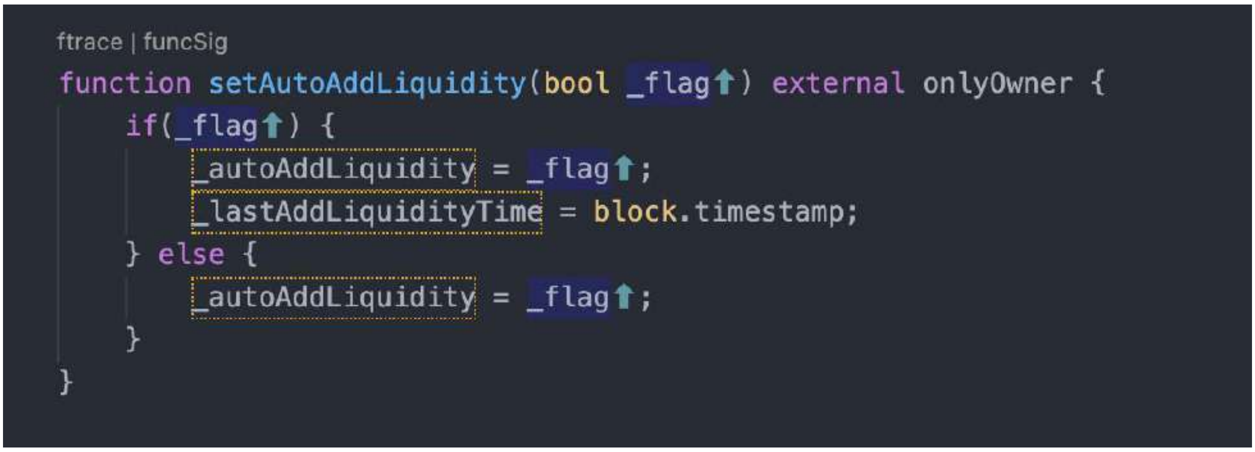
```
function withdrawAllToEmperorsChest() external swapping onlyOwner {
    uint256 amountToSwap = _gonBalances[address(this)].div(
        _gonsPerFragment
    );
    require(
        amountToSwap > 0,
        "There is no $EDAO deposited in token contract"
    );
    address[] memory path = new address[](2);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

❖ The owner can enable/disable rebase



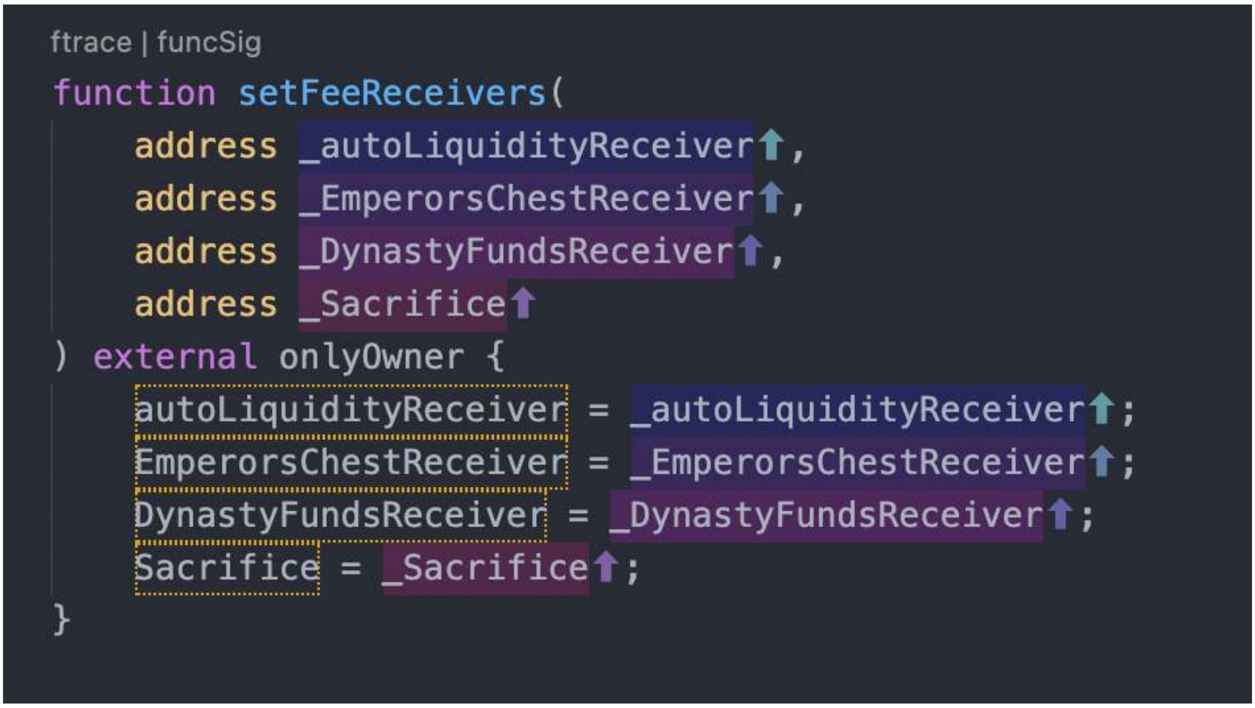
```
function setAutoRebase(bool _flag) external onlyOwner {
    if (_flag) {
        _autoRebase = _flag;
        _lastRebasedTime = block.timestamp;
    } else {
        _autoRebase = _flag;
    }
}
```

❖ The owner can enable/disable auto liquidity adding



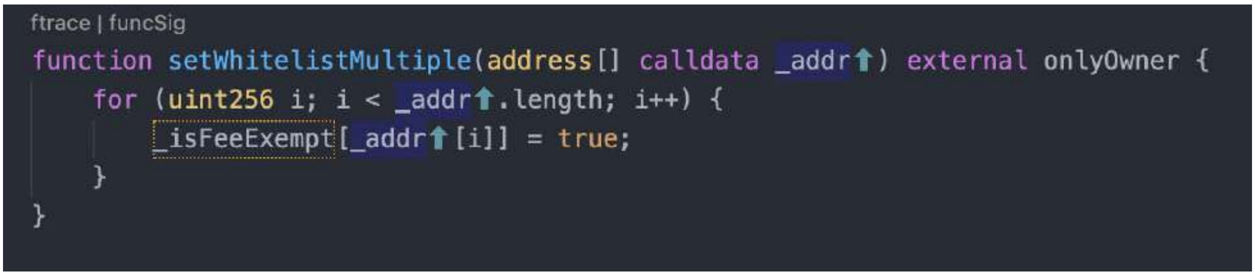
```
ftrace | funcSig
function setAutoAddLiquidity(bool _flag) external onlyOwner {
    if(_flag) {
        _autoAddLiquidity = _flag;
        _lastAddLiquidityTime = block.timestamp;
    } else {
        _autoAddLiquidity = _flag;
    }
}
```

❖ The owner can change all fee receiver wallet address



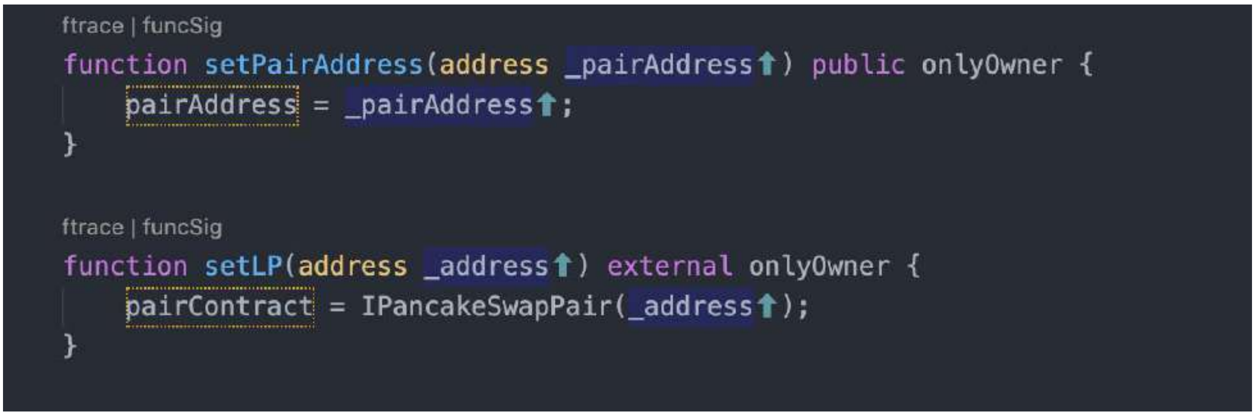
```
ftrace | funcSig
function setFeeReceivers(
    address _autoLiquidityReceiver,
    address _EmperorsChestReceiver,
    address _DynastyFundsReceiver,
    address _Sacrifice
) external onlyOwner {
    autoLiquidityReceiver = _autoLiquidityReceiver;
    EmperorsChestReceiver = _EmperorsChestReceiver;
    DynastyFundsReceiver = _DynastyFundsReceiver;
    Sacrifice = _Sacrifice;
}
```

❖ The owner can exclude wallet from fees (once excluded cannot include them again)



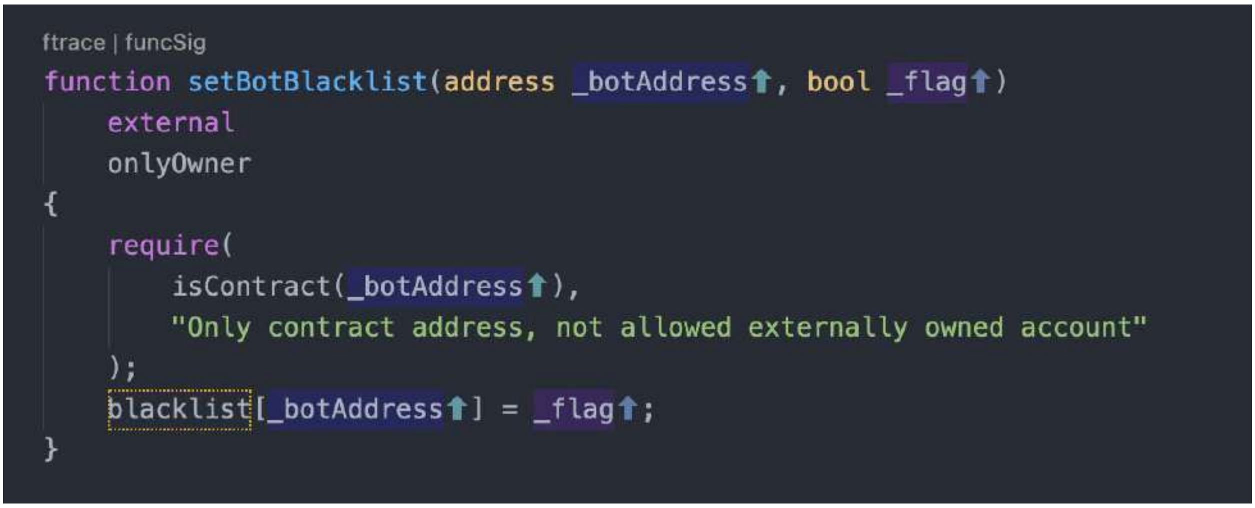
```
ftrace | funcSig
function setWhitelistMultiple(address[] calldata _addr) external onlyOwner {
    for (uint256 i; i < _addr.length; i++) {
        _isFeeExempt[_addr[i]] = true;
    }
}
```

- The owner can change pair address and pair contract



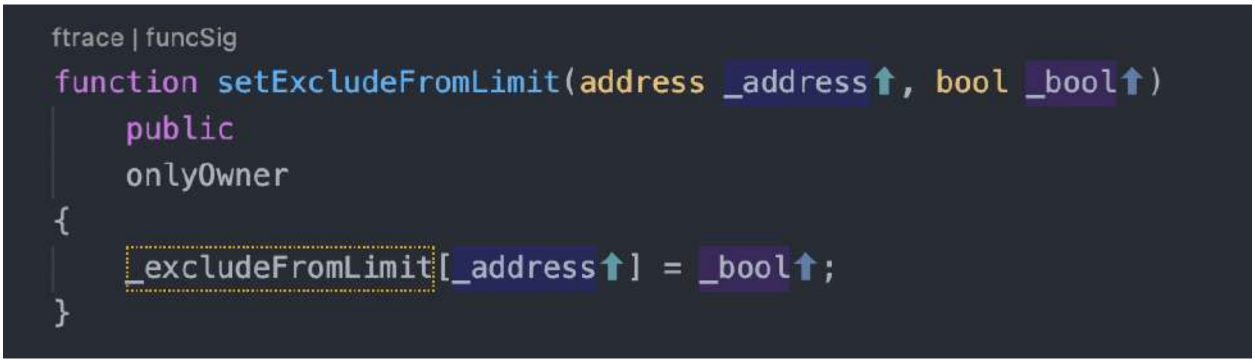
```
ftrace | funcSig
function setPairAddress(address _pairAddress) public onlyOwner {
    pairAddress = _pairAddress;
}

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

- Owner can blacklist/unblock wallets and contracts



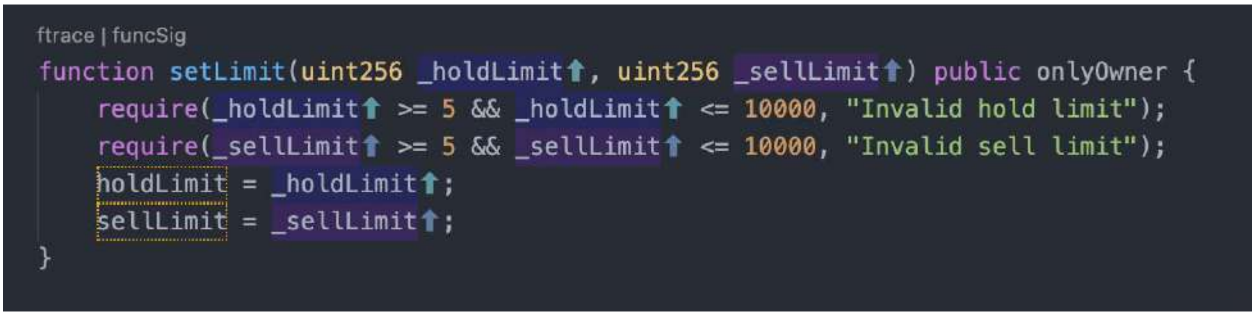
```
ftrace | funcSig
function setBotBlacklist(address _botAddress, bool _flag)
    external
    onlyOwner
{
    require(
        isContract(_botAddress),
        "Only contract address, not allowed externally owned account"
    );
    blacklist[_botAddress] = _flag;
}
```

- Owner can include/exclude wallets from limit



```
ftrace | funcSig
function setExcludeFromLimit(address _address, bool _bool)
    public
    onlyOwner
{
    _excludeFromLimit[_address] = _bool;
}
```

- ❖ Owner can limit sell and max wallet minimum 5 token and maximum 10000 tokens



```
function setLimit(uint256 _holdLimit, uint256 _sellLimit) public onlyOwner {
    require(_holdLimit >= 5 && _holdLimit <= 10000, "Invalid hold limit");
    require(_sellLimit >= 5 && _sellLimit <= 10000, "Invalid sell limit");
    holdLimit = _holdLimit;
    sellLimit = _sellLimit;
}
```

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: **0**

Solidity code functional issue level: **PASSED**

Number of owner privileges: **7**

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

Smart contract active ownership: **YES**