RUGFREECOINS

Audinals Token

RugfreeCoins Verified on September 2nd, 2023

Overview

✅ No mint function found, the owner cannot mint tokens after initial deployment.

✅ The owner can’t set a max transaction limit.

✅ The owner can’t enable or disable trading.

❌ The owner must enable trade for the holders, if trading remains disabled, no one would be able to buy and sell.

✅ The owner can't change fees.

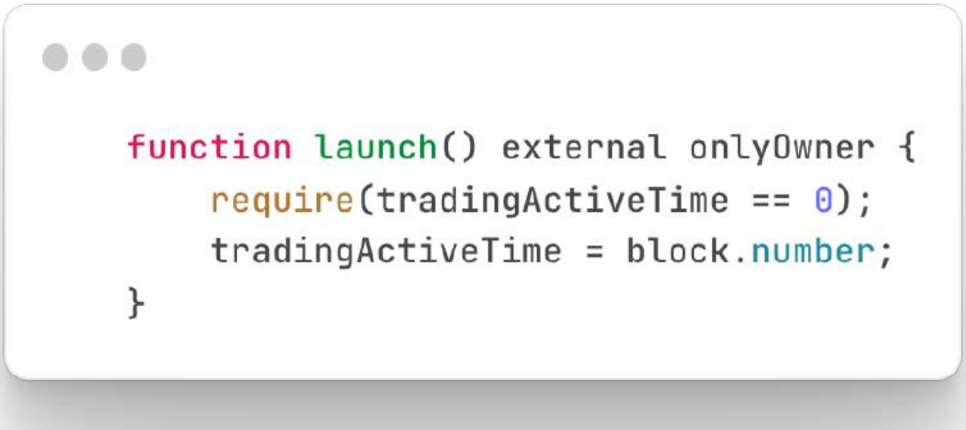
❌ The owner can blacklist wallets. By using the transferProtection function the owner can blacklist any wallets from selling.

✅ The owner can’t set a max wallet limit.

✅ The owner can’t claim the contract’s balance of its own token.

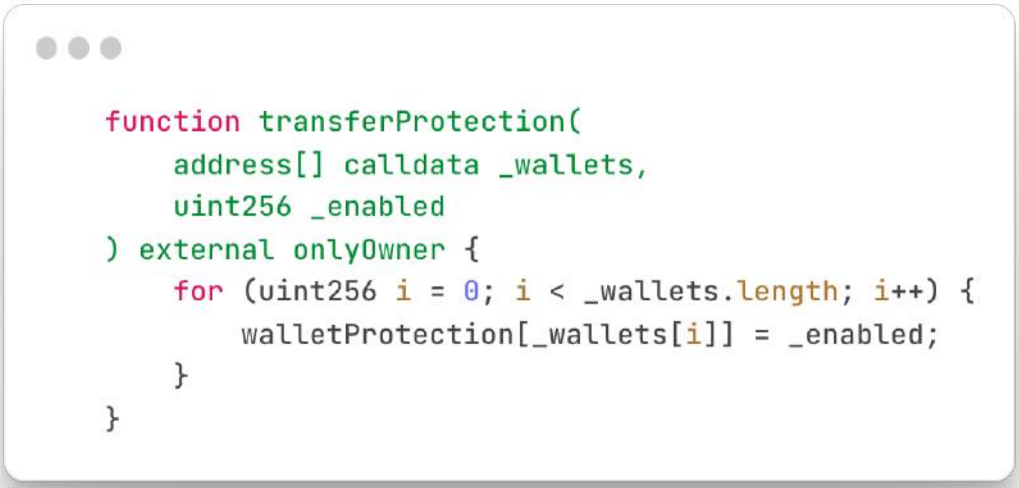
! High severity issues

The owner must enable trade for the holders, if trading remains disabled, no one would be able to buy and sell.



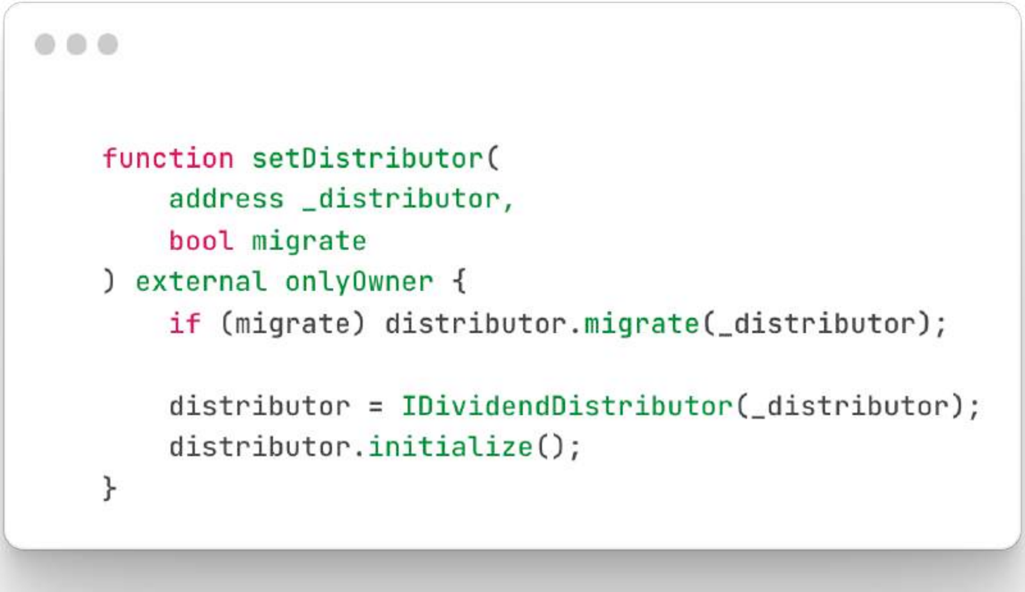
```
function launch() external onlyOwner {
    require(tradingActiveTime == 0);
    tradingActiveTime = block.number;
}
```

By using the transferProtection function owner can blacklist any wallets from selling



```
function transferProtection(
    address[] calldata _wallets,
    uint256 _enabled
) external onlyOwner {
    for (uint256 i = 0; i < _wallets.length; i++) {
        walletProtection[_wallets[i]] = _enabled;
    }
}
```

The owner has the capability to modify the distributor address to any chosen address. In the event that the owner designates this action to an inactive contract, trading will cease. For instance, if a new contract does not possess the "setShare" function, trading will be halted.



```
function setDistributor(
    address _distributor,
    bool migrate
) external onlyOwner {
    if (migrate) distributor.migrate(_distributor);

    distributor = IDividendDistributor(_distributor);
    distributor.initialize();
}
```

Contents

Audit details

Audited project

Audinals Token

Contract Address

Main contract - 0x5F68F6e8Da909e3d1Ed3c1d28553563bADE5BB4a
Distributor - 0x4671746BBFb3Ac19168e5eE223266440765F7dFE

Client contact

Audinals Token Team

Blockchain

Ethereum

Project website

https://www.audinals.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 **Audinals Token Team** to perform an audit of the smart contract.

https://etherscan.io/token/0x5F68F6e8Da909e3d1Ed3c1d28553563bADE5BB4a

This audit focuses on verifying 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, and long-term sustainability, and as a guide to improving the smart contract's security posture by remediating the identified issues.

Tokenomics

▲ 15% tax when buying & selling in the 1st block

15% of trade goes to the reward tracker in ETH and will be converted to USDT and distributed among holders: Holders will have to manually claim rewards.

▲ 5% tax when buying & selling

5% of trade goes to the reward tracker in ETH and will be converted to USDT and distributed among holders: Holders will have to manually claim rewards.

Target market and the concept

- 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 interested in taking part in the Audinals Token ecosystem.
- Anyone who’s interested in taking part in the future plans of Audinals Token.
- Anyone who’s interested in making financial transactions with any other party using Audinals Token as the currency.

Potential to grow with score points

⚡ Project efficiency	8 / 10
🌟 Project uniqueness	8 / 10
📊 Information quality	8 / 10
👌 Service quality	8 / 10
💻 System quality	8 / 10
🌍 Impact on the community	8 / 10
💼 Impact on the business	9/ 10
🔮 Preparing for the future	8 / 10
🔒 Smart contract security	7/ 10
🛠️ Smart contract functionality assessment	9/ 10
🏆 **Total Score**	**8.1**/ **10**

Contract details

Token contract details for 2nd of September 2023

Contract name	**Audinals(AUDO)**
Contract address	**0x5F68F6e8Da909e3d1Ed3c1d28553563bADE5BB4a**
Token supply	**1,000,000,000**
Token ticker	**AUDO**
Decimals	**9**
Token holders	**896**
Transaction count	**2969**
Contract deployer address	**0x6e8F7D5cA67D888c5E0aD77011e603b93Bc1b299**
Contract's current owner address	**0x389346E15bd2D4CFB046E1C70911Dc1D9b9B639B**
Distributor	**0x389346E15bd2D4CFB046E1C70911Dc1D9b9B639B**

Contract code function details

№	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
		Self destruct function security	PASS
3	Business security & centralisation	Access control of owners	HIGH
		Business logics	PASS
		Business implementation	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
Context	**Implementation**			
L	_msgSender	Internal 🔒		
L	_msgData	Internal 🔒		
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 ❗
IERC20Metadata	**Interface**	**IERC20**		
L	name	External ❗		NO ❗
L	symbol	External ❗		NO ❗
L	decimals	External ❗		NO ❗
ERC20	**Implementation**	**Context, IERC20, IERC20 Metadata**		
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	_approve	Internal 🔒	🛑	
L	_initialTransfer	Internal 🔒	🛑	
Ownable	**Implementation**	**Context**		
L		Public ❗	🛑	NO ❗
L	owner	Public ❗		NO ❗
L	renounceOwnership	Public ❗	🛑	onlyOwner
L	transferOwnership	Public ❗	🛑	onlyOwner
USDT	**Interface**			
L	balanceOf	External ❗	🛑	NO ❗
L	transfer	External ❗	🛑	NO ❗
L	approve	External ❗	🛑	NO ❗
IDividend Distributor	**Interface**			
L	initialize	External ❗	🛑	NO ❗
L	setDistributionCriteria	External ❗	🛑	NO ❗

L	setShare	External ❗	🛑	NO ❗
L	deposit	External ❗	💵	NO ❗
L	claimDividend	External ❗	🛑	NO ❗
L	getUnpaidEarnings	External ❗		NO ❗
L	getPaidDividends	External ❗		NO ❗
L	getTotalPaid	External ❗		NO ❗
L	getClaimTime	External ❗		NO ❗
L	getLostRewards	External ❗		NO ❗
L	getTotalDividends	External ❗		NO ❗
L	getTotalDistributed	External ❗		NO ❗
L	getTotalSacrificed	External ❗		NO ❗
L	countShareholders	External ❗		NO ❗
L	migrate	External ❗	🛑	NO ❗
ILpPair	**Interface**			
L	sync	External ❗	🛑	NO ❗
IDexRouter	**Interface**			
L	factory	External ❗		NO ❗
L	WETH	External ❗		NO ❗
L	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑	NO ❗
L	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🛑	NO ❗
L	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💵	NO ❗
L	swapExactETHForTokens	External ❗	💵	NO ❗
L	swapETHForExactTokens	External ❗	💵	NO ❗
L	addLiquidityETH	External ❗	💵	NO ❗
L	getAmountsOut	External ❗		NO ❗

IDexFactory	**Interface**			
└	createPair	External ❗️	🛑	NO ❗️
Audinals	**Implementation**	**ERC20, Ownable**		
└		Public ❗️	🛑	ERC20
└		External ❗️	💵	NO ❗️
└	decimals	Public ❗️		NO ❗️
└	updateSwapTokens	External ❗️	🛑	onlyOwner
└	toggleSwap	External ❗️	🛑	onlyOwner
└	setPair	External ❗️	🛑	onlyOwner
└	getSellFees	Public ❗️		NO ❗️
└	getBuyFees	Public ❗️		NO ❗️
└	excludeFromFees	Public ❗️	🛑	onlyOwner
└	setDividendExempt	External ❗️	🛑	onlyOwner
└	_transfer	Internal 🔒	🛑	
└	swapTokensForEth	Private 🔐	🛑	
└	swapBack	Private 🔐	🛑	
└	withdrawTax	External ❗️	🛑	NO ❗️
└	updateSplit	External ❗️	🛑	onlyOwner
└	prepare	External ❗️	💵	onlyOwner
└	launch	External ❗️	🛑	onlyOwner
└	setDistributor	External ❗️	🛑	onlyOwner
└	setTaxCollector	External ❗️	🛑	onlyOwner
└	setDistributionCriteria	External ❗️	🛑	onlyOwner
└	manualDeposit	External ❗️	💵	NO ❗️
└	getPoolStatistics	External ❗️		NO ❗️
└	myStatistics	External ❗️		NO ❗️
└	checkClaimTime	External ❗️		NO ❗️
└	claim	External ❗️	🛑	NO ❗️

L	airdropToWallets	External ❗	🛑	onlyOwner
L	transferProtection	External ❗	🛑	onlyOwner
L	_beforeTokenTransfer	Internal 🔒		
Context	**Implementation**			
L	_msgSender	Internal 🔒		
L	_msgData	Internal 🔒		
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 ❗
IERC20Metadata	**Interface**	**IERC20**		
L	name	External ❗		NO ❗
L	symbol	External ❗		NO ❗
L	decimals	External ❗		NO ❗
ERC20	**Implementation**	**Context, IERC20, IERC20 Metadata**		
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	_approve	Internal 🔒	🛑	
L	_initialTransfer	Internal 🔒	🛑	
Ownable	**Implementation**	**Context**		
L		Public ❗	🛑	NO ❗
L	owner	Public ❗		NO ❗
L	renounceOwnership	Public ❗	🛑	onlyOwner
L	transferOwnership	Public ❗	🛑	onlyOwner
USDT	**Interface**			
L	balanceOf	External ❗	🛑	NO ❗
L	transfer	External ❗	🛑	NO ❗
L	approve	External ❗	🛑	NO ❗
IDividend Distributor	**Interface**			
L	initialize	External ❗	🛑	NO ❗
L	setDistributionCriteria	External ❗	🛑	NO ❗
L	setShare	External ❗	🛑	NO ❗
L	deposit	External ❗	💵	NO ❗
L	claimDividend	External ❗	🛑	NO ❗
L	getUnpaidEarnings	External ❗		NO ❗
L	getPaidDividends	External ❗		NO ❗

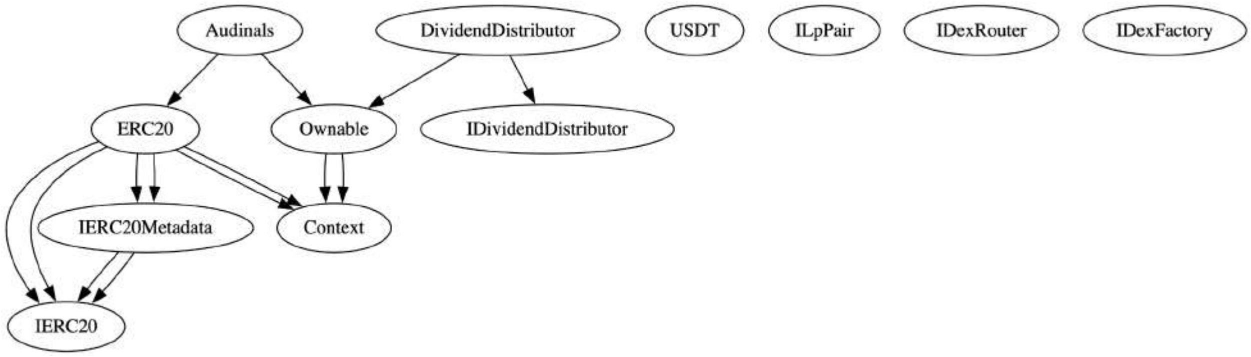
L	getTotalPaid	External ❗		NO ❗
L	getClaimTime	External ❗		NO ❗
L	getLostRewards	External ❗		NO ❗
L	getTotalDividends	External ❗		NO ❗
L	getTotalDistributed	External ❗		NO ❗
L	getTotalSacrificed	External ❗		NO ❗
L	countShareholders	External ❗		NO ❗
L	migrate	External ❗	🛑	NO ❗
Dividend Distributor	**Implementation**	**IDividend Distributor, Ownable**		
L	getTotalDividends	External ❗		NO ❗
L	getTotalDistributed	External ❗		NO ❗
L	getTotalSacrificed	External ❗		NO ❗
L		Public ❗	🛑	NO ❗
L	initialize	External ❗	🛑	NO ❗
L	setToken	External ❗	🛑	onlyOwner
L	setDistributionCriteria	External ❗	🛑	onlyToken
L	setBonusPeriod	External ❗	🛑	onlyOwner
L	setShare	External ❗	🛑	onlyToken
L	deposit	External ❗	💵	NO ❗
L	reinjectSacrificed	External ❗	🛑	onlyOwner
L	repurposeSacrificed	External ❗	🛑	onlyOwner
L	extractLostETH	External ❗	🛑	onlyOwner
L	extractExcessTokens	External ❗	🛑	onlyOwner
L	setSlippage	External ❗	🛑	onlyOwner
L	migrate	External ❗	🛑	onlyToken
L	shouldDistribute	Internal 🔒		
L	getClaimTime	External ❗		onlyToken

└	distributeDividend	Internal 🔒	🛑	
└	claimDividend	External ❗️	🛑	onlyToken
└	buyback	External ❗️	🛑	NO ❗️
└	claimBonus	External ❗️	🛑	NO ❗️
└	getUnpaidEarnings	Public ❗️		NO ❗️
└	getPaidDividends	External ❗️		onlyToken
└	getTotalPaid	External ❗️		onlyToken
└	getLostRewards	External ❗️		onlyToken
└	getCumulativeDividends	Internal 🔒		
└	countShareholders	Public ❗️		NO ❗️
└	addShareholder	Internal 🔒	🛑	
└	removeShareholder	Internal 🔒	🛑	
└		External ❗️	💵	NO ❗️
ILpPair	**Interface**			
└	sync	External ❗️	🛑	NO ❗️
IDexRouter	**Interface**			
└	factory	External ❗️		NO ❗️
└	WETH	External ❗️		NO ❗️
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗️	🛑	NO ❗️
└	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗️	🛑	NO ❗️
└	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗️	💵	NO ❗️
└	swapExactETHForTokens	External ❗️	💵	NO ❗️
└	swapETHForExactTokens	External ❗️	💵	NO ❗️
└	addLiquidityETH	External ❗️	💵	NO ❗️
└	getAmountsOut	External ❗️		NO ❗️

IDexFactory	Interface			
L	createPair	External ❗	🛑	NO ❗

Legend

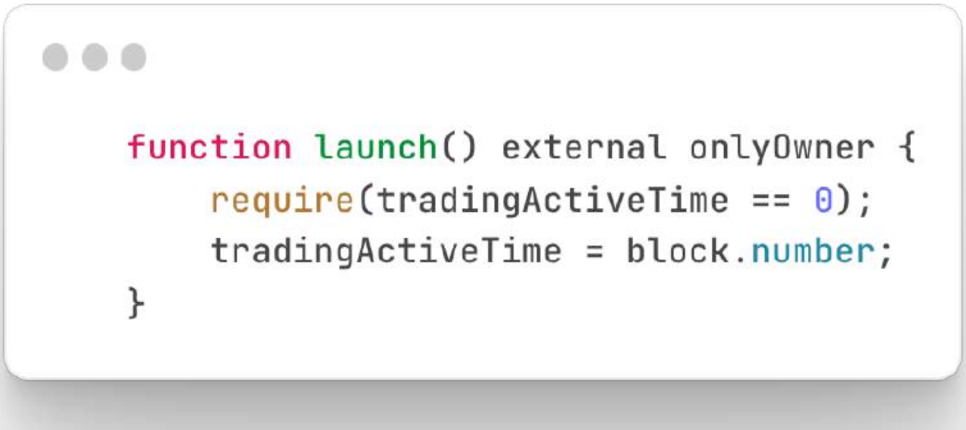
Symbol	Meaning
🛑	Function can modify state
💵	Function is payable



Security issue checking status

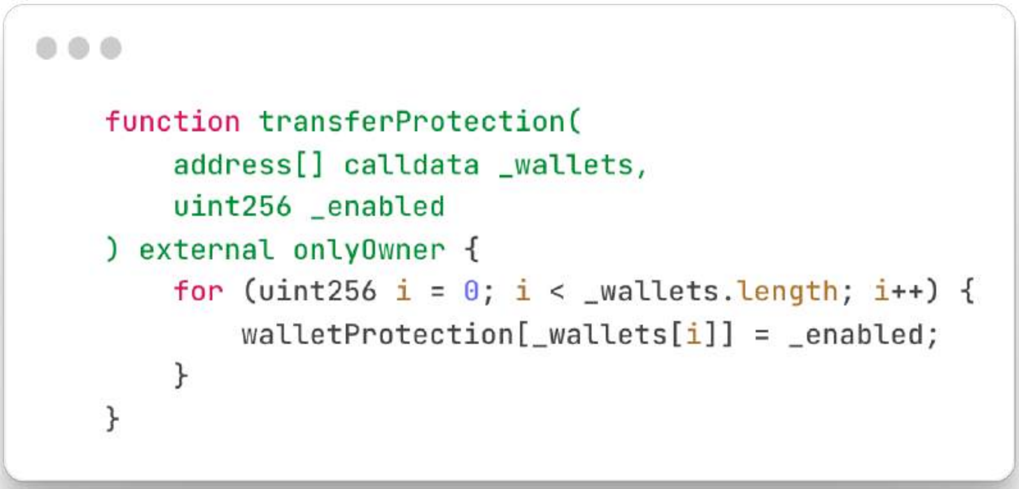
- ❖ High severity issues

The owner must enable trade for the holders, if trading remains disabled, no one would be able to buy and sell.



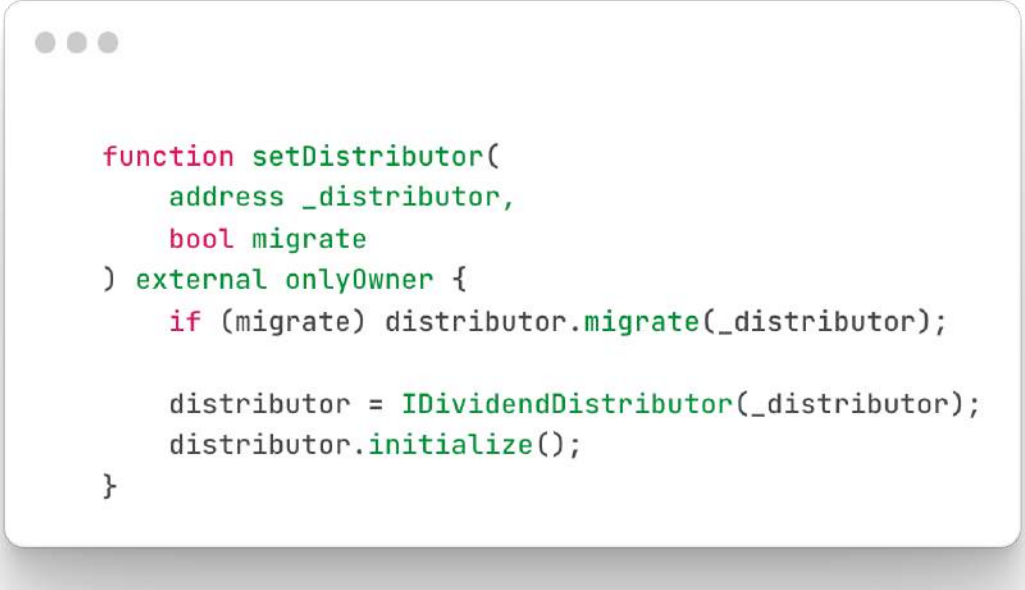
```
function launch() external onlyOwner {
    require(tradingActiveTime == 0);
    tradingActiveTime = block.number;
}
```

By using the transferProtection function owner can blacklist any wallets from selling



```
function transferProtection(
    address[] calldata _wallets,
    uint256 _enabled
) external onlyOwner {
    for (uint256 i = 0; i < _wallets.length; i++) {
        walletProtection[_wallets[i]] = _enabled;
    }
}
```

The owner has the capability to modify the distributor address to any chosen address. In the event that the owner designates this action to an inactive contract, trading will cease. For instance, if a new contract does not possess the "setShare" function, trading will be halted.



```
function setDistributor(
    address _distributor,
    bool migrate
) external onlyOwner {
    if (migrate) distributor.migrate(_distributor);

    distributor = IDividendDistributor(_distributor);
    distributor.initialize();
}
```

❖ Medium severity issues

No medium severity issues found

❖ Low severity issues

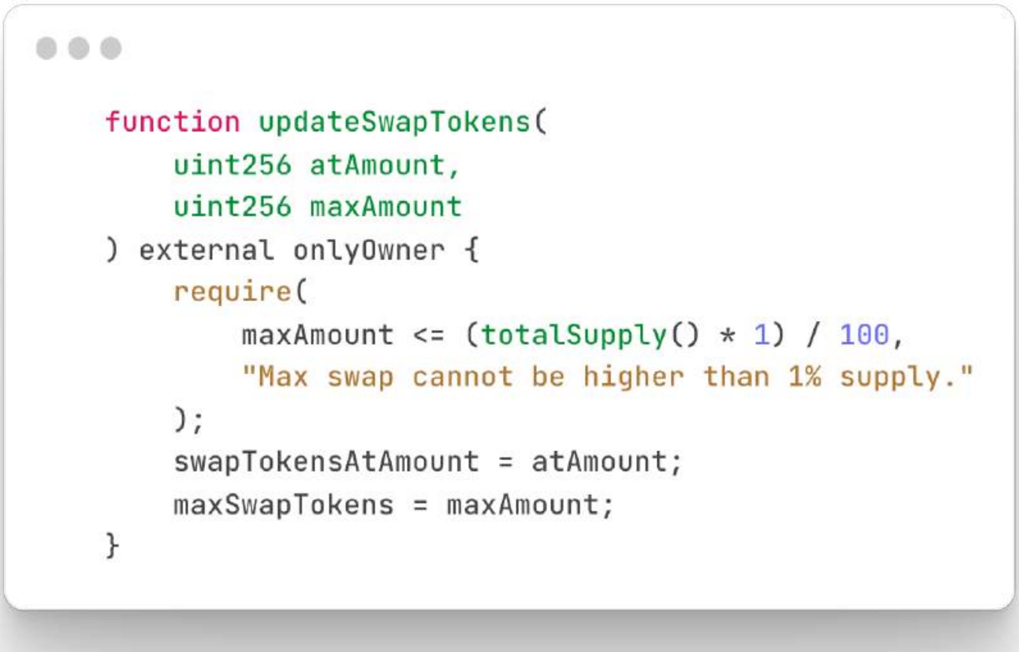
No low-severity issues found

❖ Centralization issues

No Centralization issues found

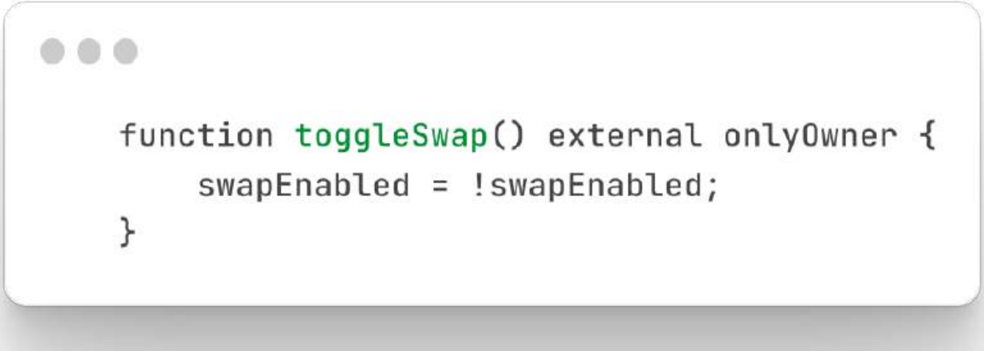
Owner privileges

- Owner can change swap point and maximum swapping token amount maximum up to 1%



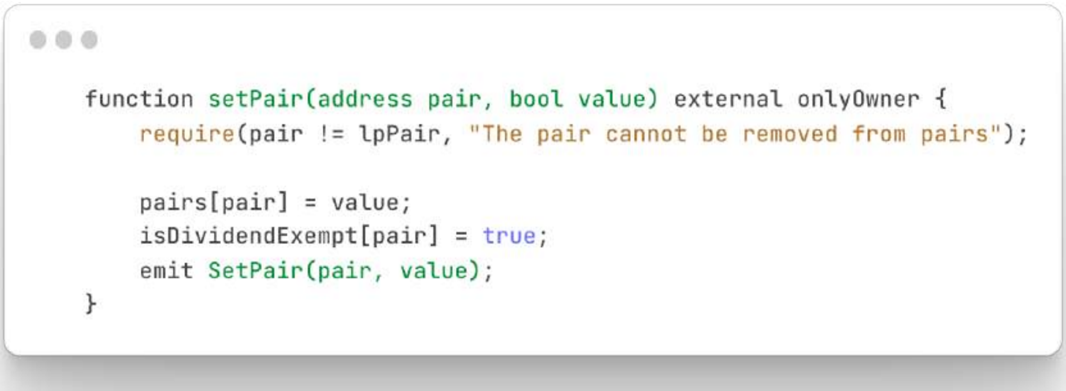
```
function updateSwapTokens(
    uint256 atAmount,
    uint256 maxAmount
) external onlyOwner {
    require(
        maxAmount <= (totalSupply() * 1) / 100,
        "Max swap cannot be higher than 1% supply."
    );
    swapTokensAtAmount = atAmount;
    maxSwapTokens = maxAmount;
}
```

- Owner can enable/disable swapping



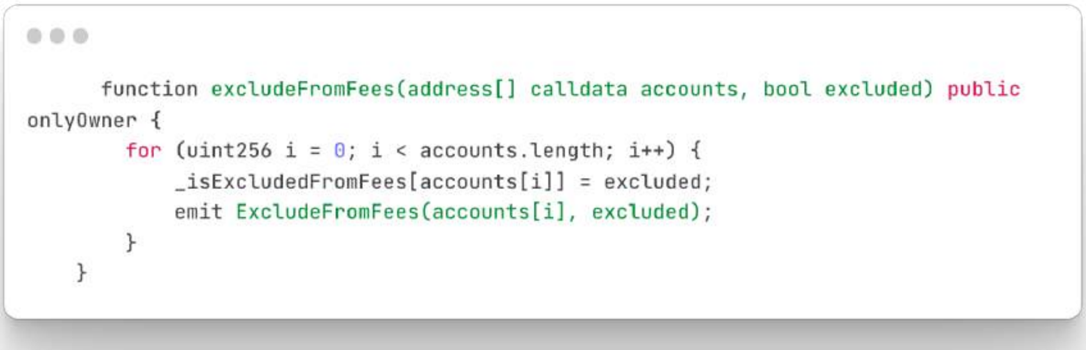
```
function toggleSwap() external onlyOwner {
    swapEnabled = !swapEnabled;
}
```

- Owner can add remove new pairs



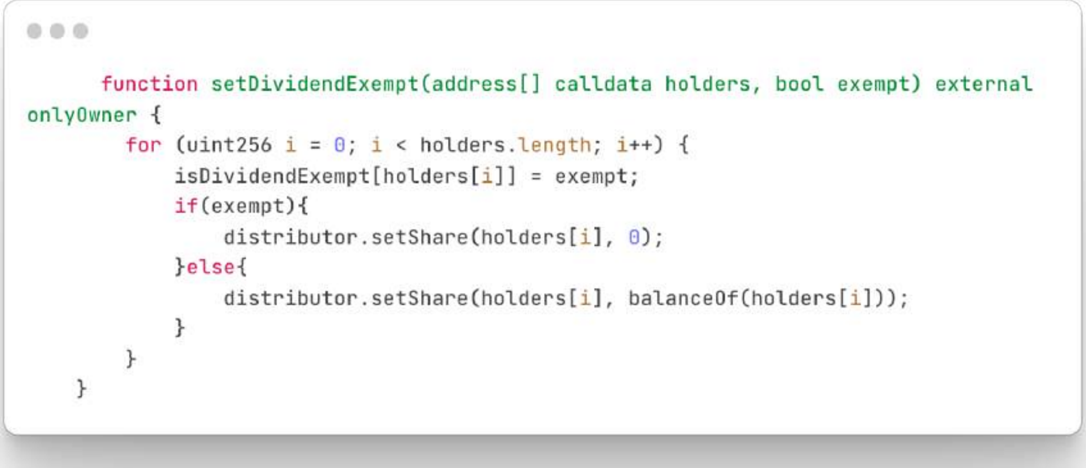
```
function setPair(address pair, bool value) external onlyOwner {
    require(pair != lpPair, "The pair cannot be removed from pairs");

    pairs[pair] = value;
    isDividendExempt[pair] = true;
    emit SetPair(pair, value);
}
```

- Owner can include/exclude wallets from fees



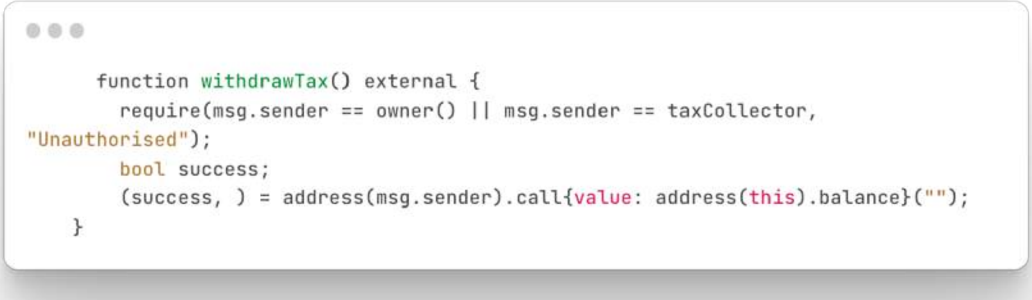
```
      function excludeFromFees(address[] calldata accounts, bool excluded) public
onlyOwner {
        for (uint256 i = 0; i < accounts.length; i++) {
            _isExcludedFromFees[accounts[i]] = excluded;
            emit ExcludeFromFees(accounts[i], excluded);
        }
    }
```

- Owner can include/exclude wallets from rewards



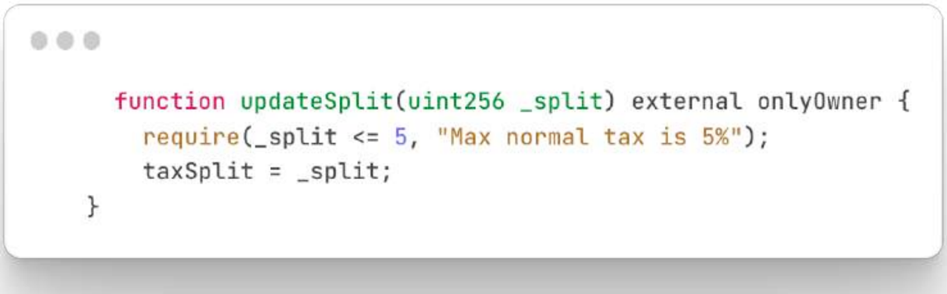
```
      function setDividendExempt(address[] calldata holders, bool exempt) external
onlyOwner {
        for (uint256 i = 0; i < holders.length; i++) {
            isDividendExempt[holders[i]] = exempt;
            if(exempt){
                distributor.setShare(holders[i], 0);
            }else{
                distributor.setShare(holders[i], balanceOf(holders[i]));
            }
        }
    }
```

- Owner can get contract ETH balance



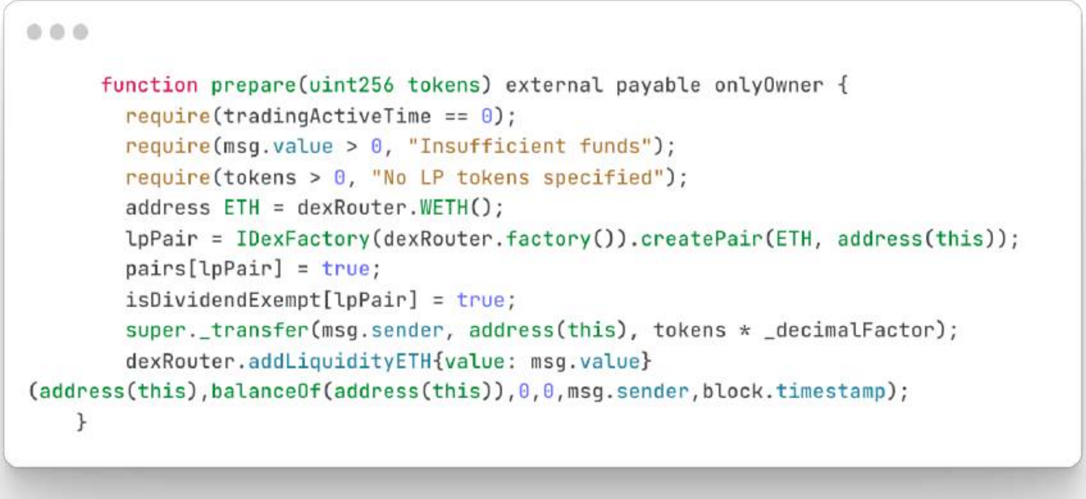
```
      function withdrawTax() external {
        require(msg.sender == owner() || msg.sender == taxCollector,
"Unauthorised");
        bool success;
        (success, ) = address(msg.sender).call{value: address(this).balance}("");
    }
```

- Owner can split swapping eth maximum up-to 5%



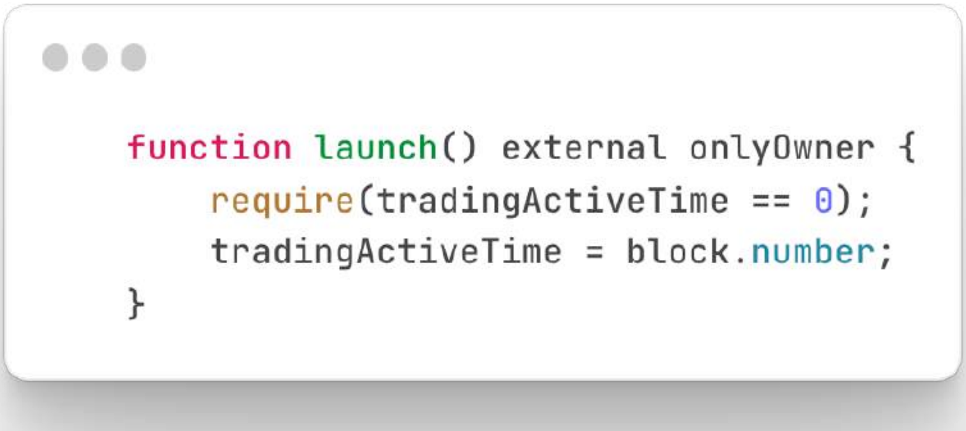
```
    function updateSplit(uint256 _split) external onlyOwner {
      require(_split <= 5, "Max normal tax is 5%");
      taxSplit = _split;
  }
```

- Owner can add Liquidity using prepare function



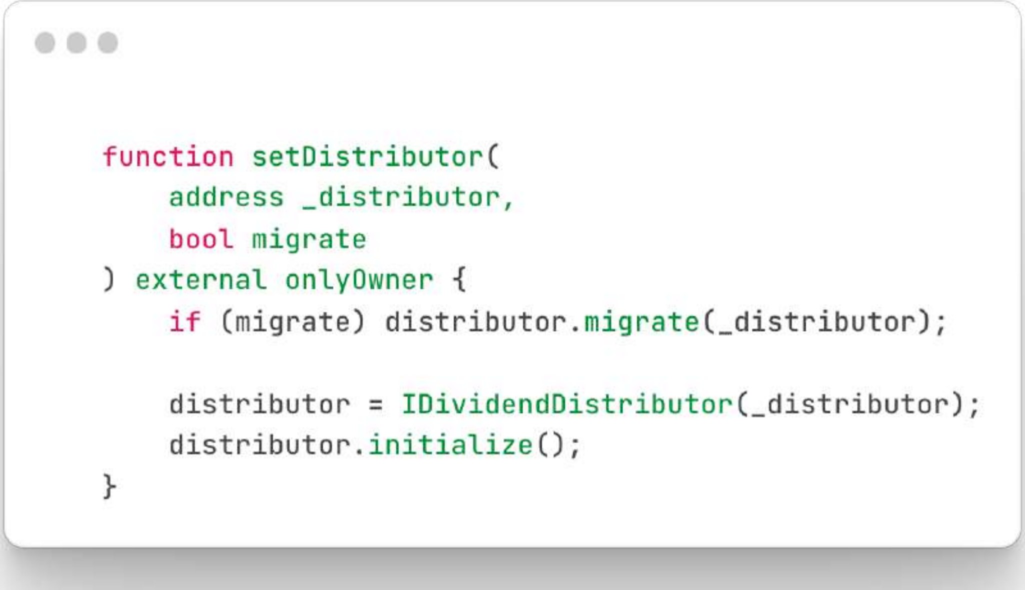
```
      function prepare(uint256 tokens) external payable onlyOwner {
        require(tradingActiveTime == 0);
        require(msg.value > 0, "Insufficient funds");
        require(tokens > 0, "No LP tokens specified");
        address ETH = dexRouter.WETH();
        lpPair = IDexFactory(dexRouter.factory()).createPair(ETH, address(this));
        pairs[lpPair] = true;
        isDividendExempt[lpPair] = true;
        super._transfer(msg.sender, address(this), tokens * _decimalFactor);
        dexRouter.addLiquidityETH{value: msg.value}
(address(this),balanceOf(address(this)),0,0,msg.sender,block.timestamp);
    }
```

- ❖ Owner can enable trading, once enabled can not disable it again



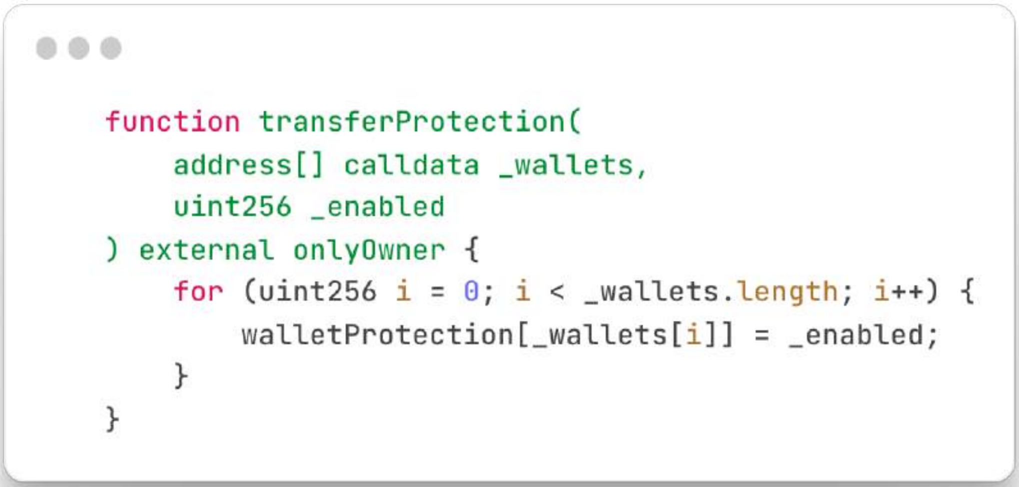
```
function launch() external onlyOwner {
    require(tradingActiveTime == 0);
    tradingActiveTime = block.number;
}
```

- ❖ Owner can change the distributor address



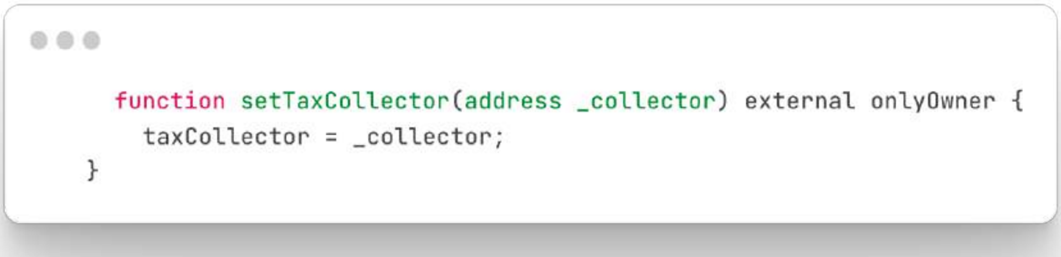
```
function setDistributor(
    address _distributor,
    bool migrate
) external onlyOwner {
    if (migrate) distributor.migrate(_distributor);

    distributor = IDividendDistributor(_distributor);
    distributor.initialize();
}
```

- Owner can block/unblock waller from transferring tokens



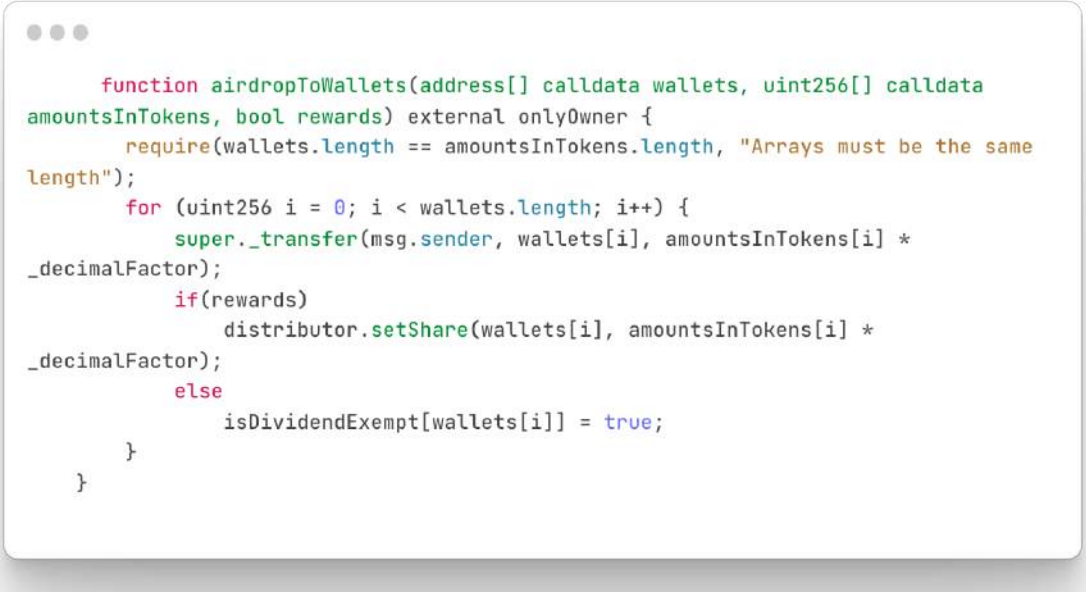
```
function transferProtection(
    address[] calldata _wallets,
    uint256 _enabled
) external onlyOwner {
    for (uint256 i = 0; i < _wallets.length; i++) {
        walletProtection[_wallets[i]] = _enabled;
    }
}
```

- Owner can change tax collector address (tax collector can withdraw eth from the contract)



```
    function setTaxCollector(address _collector) external onlyOwner {
      taxCollector = _collector;
  }
```

❖ Owner can airdrop tokens



```
      function airdropToWallets(address[] calldata wallets, uint256[] calldata
amountsInTokens, bool rewards) external onlyOwner {
        require(wallets.length == amountsInTokens.length, "Arrays must be the same
length");
        for (uint256 i = 0; i < wallets.length; i++) {
            super._transfer(msg.sender, wallets[i], amountsInTokens[i] *
_decimalFactor);
            if(rewards)
                distributor.setShare(wallets[i], amountsInTokens[i] *
_decimalFactor);
            else
                isDividendExempt[wallets[i]] = true;
        }
    }
```

Audit conclusion

RugFreeCoins team has performed in-depth testing, 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:	PASS
Smart contract Security Status:	HIGH ISSUES
Number of risk issues:	3
Solidity code functional issue level:	PASS
Number of owner privileges:	11
Centralization risk correlated to the active owner:	HIGH
Smart contract active ownership:	ACTIVE