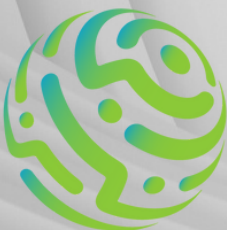




**AUDIT REPORT**

# **SecureWise**

**SMART CONTRACT AUDIT**



**META AIR**



<https://github.com/securewise>



<https://t.me/securewise>



<https://securewise.info/>



# Table of Contents

**03**

Disclaimer

**04**

Overview

**05**

Quick Result

**06**

Auditing  
Approach and  
Methodologies

**07**

Automated  
Analysis

**10**

Inheritance  
Graph

**11**

Contract  
Summary

**12**

Manual Review



# Disclaimer

SecureWise provides the smart contract audit of solidity. Audit and report are for informational purposes only and not, nor should be considered, as an endorsement to engage with, invest in, participate, provide an incentive, or disapprove, criticise, discourage, or purport to provide an opinion on any particular project or team.

This audit report doesn't provide any warranty or guarantee regarding the nature of the technology analysed. These reports, in no way, provide investment advice, nor should be used as investment advice of any sort. Investors must always do their own research and manage their risk.

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 SecureWise and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) SecureWise owe no duty of care towards you or any other person, nor does SecureWise 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 SecureWise 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, SecureWise hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against SecureWise, 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.

# Overview

**Token Name:** META AIR (**MAIR**)

**Methodology:** Automated Analysis, Manual Code Review

**Language:** Solidity

**Contract Address:** 0x1828e38E7271c44e4BeC6810dC5279223cCd14ff

**ContractLink:** <https://bscscan.com/address/0x1828e38E7271c44e4BeC6810dC5279223cCd14ff>

**Network:** Binance Smart Chain (BSC)

**Decimals:** 9

**Supply:** 1.000.000.000

**Website:** <https://metaaircoin.com/>

**Twitter:** <https://twitter.com/metaaircoin>

**Telegram:** <https://t.me/metaair>

**Report Date:** August 15, 2022

# Quick Result

SecureWise has applied the automated and manual analysis of Smart Contract and were reviewed for common contract vulnerabilities and centralized exploits

## Owner Privileges



The owner can pause trading



The owner can set the max tx amount "0"



The owner can set fees up to 100%



Auto liquidity is going to an externally owned account



The owner can exclude accounts from rewards



The owner can add an account to bot list



The owner can change transaction lock-time without limit



The owner can change max wallet token amount to "0"



The owner can exclude accounts from fees



The owner can change swap settings



Buyback BNB is going to an externally owned account



The owner can withdraw any token from the contract

**META AIR (MAIR)** has succesfully **PASSED** the smart contract audit with **HIGH** severity issue

# Auditing Approach and Methodologies

SecureWise has performed starting with analyzing the code, issues, code quality, and libraries. Reviewed line-by-line by our team. Finding any potential issue like race conditions, transaction-ordering dependence, timestamp dependence, and denial of service attacks.

## Methodology

- Understanding the size, scope and functionality of your project's source code
- Manual review of code, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
- Testing and automated analysis of the Smart Contract to determine proper logic has been followed throughout the whole process
- Deploying the code on testnet using multiple live test
- Analyzing a program to determine the specific input that causes different parts of a program to execute its functions.
- Checking whether all the libraries used in the code are on the latest version.

## Goals

Smart Contract System is secure, resilient and working according to the specifications and without any vulnerabilities.



## Risk Classification




































**High:** Exploits, vulnerabilities or errors that will certainly or probabilistically lead towards loss of funds, control, of the contract and its functions. Must be fixed as soon as possible.

**Medium:** Bugs or issues with that may be subject to exploit, though their impact is somewhat limited. Must be fixed as soon as possible.

**Low:** Effects are minimal in isolation and do not pose a significant danger to the project or its users. Issues under this classification are recommended to be fixed nonetheless.

# Automated Analysis

Symbol	Meaning
	Function can modify state
	Function is payable




























Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
Address	Library			
L	sendValue	Internal 		
METAAIR	Implementation	ERC20, Ownable		
L		Public 		ERC20
L		External 		NO 
L	updateDividendTracker	Public 		onlyOwner
L	processDividendTracker	External 		NO 
L	claim	External 		NO 
L	rescueBEP20Tokens	External 		onlyOwner
L	forceSend	External 		NO 
L	updateRouter	External 		onlyOwner
L	excludeFromFees	Public 		onlyOwner
L	excludeMultipleAccountsFromFees	Public 		onlyOwner
L	excludeFromDividends	External 		onlyOwner
L	setMarketingWallet	External 		onlyOwner
L	setBuybackWallet	External 		onlyOwner
L	setOperationsWallet	External 		onlyOwner

# Automated Analysis

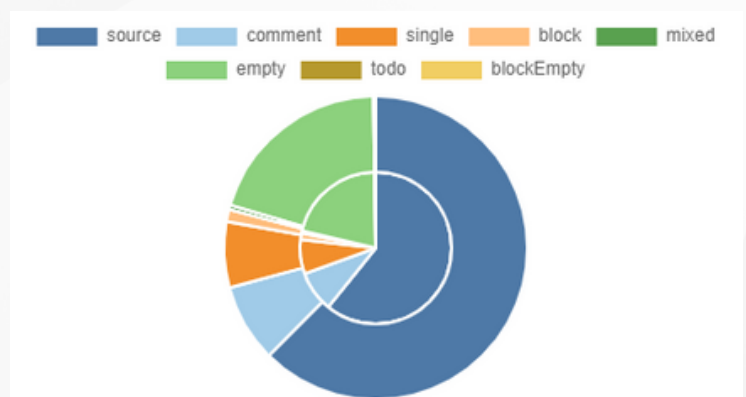
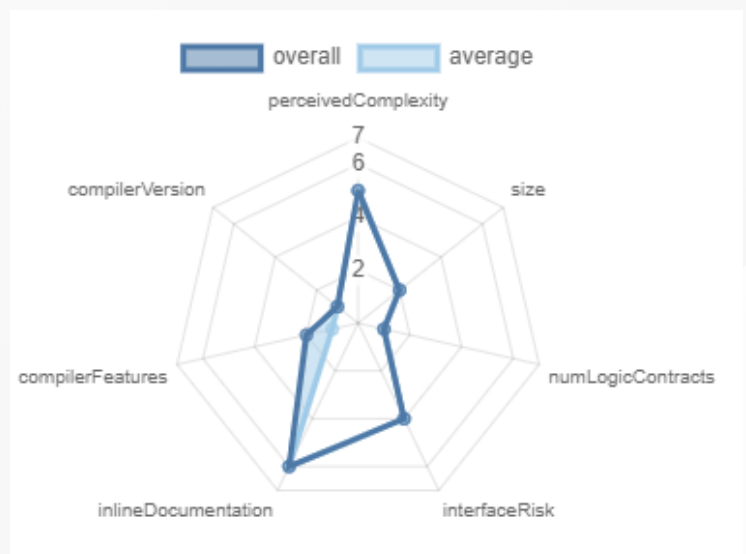
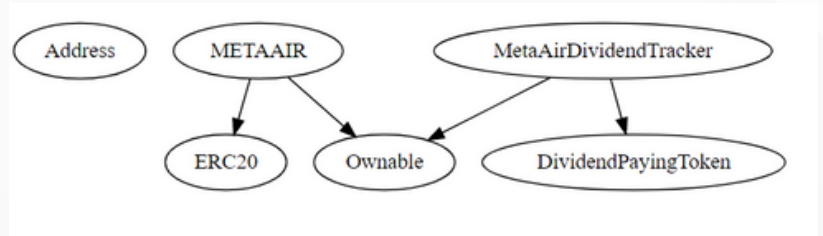
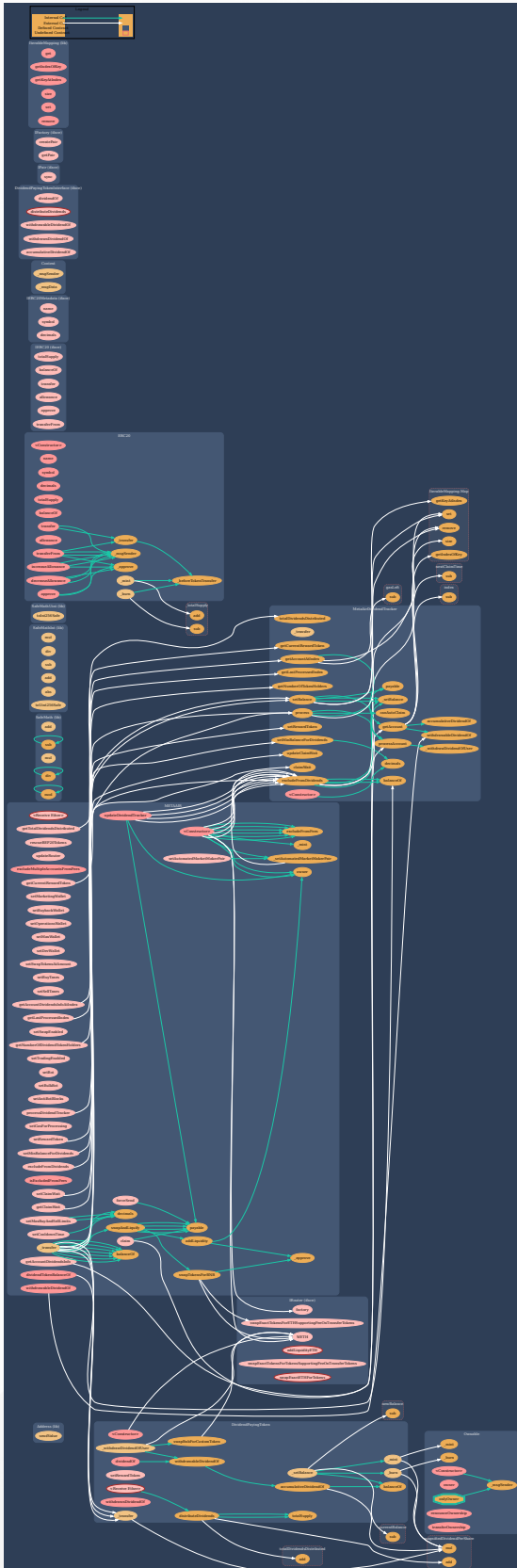
L	setMaxWallet	External		onlyOwner
L	setDevWallet	External		onlyOwner
L	setSwapTokensAtAmount	External		onlyOwner
L	setBuyTaxes	External		onlyOwner
L	setSellTaxes	External		onlyOwner
L	setRewardToken	External		onlyOwner
L	setMaxBuyAndSellLimits	External		onlyOwner
L	setSwapEnabled	External		onlyOwner
L	setCooldownTime	External		onlyOwner
L	setTradingEnabled	External		onlyOwner
L	setBot	External		onlyOwner
L	setBulkBot	External		onlyOwner
L	setAntiBotBlocks	External		onlyOwner
L	setAutomatedMarketMakerPair	External		onlyOwner
L	setMinBalanceForDividends	External		onlyOwner
L	_setAutomatedMarketMakerPair	Private		
L	setGasForProcessing	External		onlyOwner
L	setClaimWait	External		onlyOwner
L	getClaimWait	External		NO
L	getTotalDividendsDistributed	External		NO
L	isExcludedFromFees	Public		NO
L	withdrawableDividendOf	Public		NO
L	getCurrentRewardToken	External		NO
L	dividendTokenBalanceOf	Public		NO
L	getAccountDividendsInfo	External		NO
L	getAccountDividendsInfoAtIndex	External		NO
L	getLastProcessedIndex	External		NO
L	getNumberOfDividendTokenHolders	External		NO
L	_transfer	Internal		
L	swapAndLiquify	Private		
L	swapTokensForBNB	Private		
L	addLiquidity	Private		



# Automated Analysis

MetaAirDividendTracker	Implementation	Ownable, DividendPayingToken		
L		Public 		DividendPayingToken
L	_transfer	Internal 		
L	setMinBalanceForDividends	External 		onlyOwner
L	excludeFromDividends	External 		onlyOwner
L	updateClaimWait	External 		onlyOwner
L	getLastProcessedIndex	External 		NO 
L	getNumberOfTokenHolders	External 		NO 
L	getCurrentRewardToken	External 		NO 
L	getAccount	Public 		NO 
L	getAccountAtIndex	Public 		NO 
L	canAutoClaim	Private 		
L	setBalance	Public 		onlyOwner
L	process	Public 		NO 
L	processAccount	Public 		onlyOwner

# Inheritance Graph



# Contract Summary

Logic Contracts	Interfaces	Lines	nLines	nSLOC	Comment Lines	Complex. Score	Capabilities
3	————	760	724	490	70	440	
3	————	760	724	490	70	440	

## Components

Contracts	Libraries	Interfaces	Abstract
2	1	0	0

## Exposed Functions

This section lists functions that are explicitly declared public or payable. Please note that getter methods for public stateVars are not included.

Public	Payable
51	1

External	Internal	Private	Pure	View
40	46	5	1	16

## StateVariables

Total	Public
33	29

## Capabilities

Solidity Versions observed	Experimental Features	Can Receive Funds	Uses Assembly	Has Destroyable Contracts
^0.8.6		yes	————	————

Transfers ETH	Low-Level Calls	DelegateCall	Uses Hash Functions	ECRRecover	New/Create/Create2
yes	————	————	————	————	yes → NewContract:MetaAirDividendTracker

TryCatch	Σ Unchecked
yes	————

# Manual Review

## The owner can pause trading

```
255     function setTradingEnabled(bool _enabled) external onlyOwner{
256         tradingEnabled = _enabled;
257         if(startTradingBlock == 0 && _enabled == true) startTradingBlock = block.number;
258     }
```

### Recommendation

Privileged roles can be granted the power to pause() the contract in case of an external attack. Remove pause trading function.

## The owner can set the max tx amount "0"

```
239     function setMaxBuyAndSellLimits(uint256 maxBuy, uint256 maxSell) external onlyOwner{
240         maxBuyAmount = maxBuy * 10**decimals();
241         maxSellAmount = maxSell * 10**decimals();
242     }
```

### Recommendation

These functions should be provided arbitrary limits, e.g., put a **require** check that allows minimum /maximum limit etc. if **set 0** these cause pause the trading.

## The owner can set fees up to 100%

```
227     function setBuyTaxes(uint256 _rewards, uint256 _marketing, uint256 _liquidity, uint256 _operations, uint256 _buyback, uint256 _dev) external onlyOwner{
228         buyTaxes = Taxes(_rewards, _marketing, _liquidity, _operations, _buyback, _dev);
229     }
230
231     function setSellTaxes(uint256 _rewards, uint256 _marketing, uint256 _liquidity, uint256 _operations, uint256 _buyback, uint256 _dev) external onlyOwner{
232         sellTaxes = Taxes(_rewards, _marketing, _liquidity, _operations, _buyback, _dev);
233     }
```

### Recommendation

These functions should be provided arbitrary limits, e.g., put a **require** check that allows maximum limit etc.

# Manual Review

## Auto liquidity is going to an externally owned account

```
524  function addLiquidity(uint256 tokenAmount, uint256 ethAmount) private {  
525  
526      // approve token transfer to cover all possible scenarios  
527      _approve(address(this), address(router), tokenAmount);  
528  
529      // add the liquidity  
530  router.addLiquidityETH(value: ethAmount)(  
531      address(this),  
532      tokenAmount,  
533      0, // slippage is unavoidable  
534      0, // slippage is unavoidable  
535      owner(),  
536      block.timestamp  
537  );  
538  
539  }  
540  
541  }
```

### Recommendation

Authorizing privileged roles to externally-owned-account (EOA) is dangerous. Send LP tokens to dead address or unreachable address.

## The owner can exclude accounts from rewards

```
190  /// @dev "true" to exclude, "false" to include  
191  function excludeFromDividends(address account, bool value) external onlyOwner{  
192      dividendTracker.excludeFromDividends(account, value);  
193  }
```

### Recommendation

Authorizing privileged roles to exclude accounts from rewards. These cause can affect decentralization.

# Manual Review

## The owner can add an account to bot list

```
261  /// @param bot the bot address
262  /// @param value "true" to blacklist, "false" to unblacklist
263  function setBot(address bot, bool value) external onlyOwner{
264      require(!_isBot[bot] != value);
265      _isBot[bot] = value;
266  }
267
268  function setBulkBot(address[] memory bots, bool value) external onlyOwner{
269      for(uint256 i; i<bots.length; i++){
270          _isBot[bots[i]] = value;
271      }
272  }
273
274  function setAntiBotBlocks(uint256 numberOfBlocks) external onlyOwner{
275      antiBotBlocks = numberOfBlocks;
276  }
```

### Recommendation

Authorizing privileged roles to add an account to bot list and pause trade for bots. These cause can affect decentralization.

## The owner can change max wallet token amount to "0"

```
213  function setMaxWallet(uint256 amount) external onlyOwner{
214      maxWalletBalance = amount * 10**9;
215  }
```

### Recommendation

Authorizing privileged roles to change max wallet amount without limit and these cause by semi pause trading. shouldn't be 0. Put a **require** check that allows minimum reasonable limit etc.

# Manual Review

## The owner can change transaction lock-time without limit

```
250  function setCooldownTime(uint256 timeInSeconds, uint256 balance) external onlyOwner{
251      coolDownTime = timeInSeconds;
252      coolDownBalance = balance * 10**decimals();
253  }
```

### Recommendation

Authorizing privileged roles to change transaction lock-time without limit and these cause by semi pause trading. Put a **require** check that allows maximum limit etc.

## The owner can exclude accounts from fees

```
176  function excludeFromFees(address account, bool excluded) public onlyOwner {
177      require(!_isExcludedFromFees[account] != excluded, "MetaAir: Account is already the value of 'excluded'");
178      _isExcludedFromFees[account] = excluded;
179
180      emit ExcludeFromFees(account, excluded);
181  }
182
183  function excludeMultipleAccountsFromFees(address[] calldata accounts, bool excluded) public onlyOwner {
184      for(uint256 i = 0; i < accounts.length; i++) {
185          _isExcludedFromFees[accounts[i]] = excluded;
186      }
187      emit ExcludeMultipleAccountsFromFees(accounts, excluded);
188  }
```

### Recommendation

Authorizing privileged roles to exclude accounts from fees. These cause can affect decentralization.



# Manual Review

## The owner can change swap settings

```
244    /// @notice Enable or disable internal swaps
245    /// @dev Set "true" to enable internal swaps for liquidity, marketing and dividends
246    function setSwapEnabled(bool _enabled) external onlyOwner{
247        swapEnabled = _enabled;
248    }
```

### Recommendation

Authorizing privileged roles to enable or disable the swap.

## Buyback BNB is going to an externally owned account

```
487    // Send BNB to buyback
488    uint256 buybackAmt = unitBalance * 2 * sellTaxes.buyback;
489    if(buybackAmt > 0){
490        payable(buybackWallet).sendValue(buybackAmt);
491    }
```

### Recommendation

Authorizing privileged roles to externally-owned-account (EOA) is dangerous. Buybacks should be automated and should go to an unreachable address. (Dead address etc.)

## The owner can withdraw any token from the contract

```
154    /// @notice Withdraw tokens sent by mistake.
155    /// @param tokenAddress The address of the token to withdraw
156    function rescueBEP20Tokens(address tokenAddress) external onlyOwner{
157        IERC20(tokenAddress).transfer(msg.sender, IERC20(tokenAddress).balanceOf(address(this)));
158    }
```

### Recommendation




Authorizing privileged roles to withdraw any token from the contract. Include native token too. Put a require check that allows **token != address(this)**



## AUDIT REPORT

# SecureWise

## SMART CONTRACT AUDIT

 <https://github.com/securewise>  
 <https://t.me/securewise>  
 <https://securewise.info/>

