QuillAudits

Audit Report
August, 2021

CoinMama

Contents

Scope of Audit

The scope of this audit was to analyze and document the CoinMama Token smart contract codebase for quality, security, and correctness.

Checked Vulnerabilities

We have scanned the smart contract for commonly known and more specific vulnerabilities. Here are some of the commonly known vulnerabilities that we considered:

- Re-entrancy
- Timestamp Dependence
- Gas Limit and Loops
- DoS with Block Gas Limit
- Transaction-Ordering Dependence
- Use of tx.origin
- Exception disorder
- Gasless send
- Balance equality
- Byte array
- Transfer forwards all gas
- ERC20 API violation
- Malicious libraries
- Compiler version not fixed
- Redundant fallback function
- Send instead of transfer
- Style guide violation
- Unchecked external call
- Unchecked math
- Unsafe type inference
- Implicit visibility level

Techniques and Methods

Throughout the audit of smart contract, care was taken to ensure:

- The overall quality of code.
- Use of best practices.
- Code documentation and comments match logic and expected behaviour.
- Token distribution and calculations are as per the intended behaviour mentioned in the whitepaper.
- Implementation of ERC-20 token standards.
- Efficient use of gas.
- Code is safe from re-entrancy and other vulnerabilities.

The following techniques, methods and tools were used to review all the smart contracts.

Structural Analysis
In this step we have analyzed the design patterns and structure of smart contracts. A thorough check was done to ensure the smart contract is structured in a way that will not result in future problems.
SmartCheck.

Static Analysis
Static Analysis of Smart Contracts was done to identify contract vulnerabilities. In this step a series of automated tools are used to test security of smart contracts.

Code Review / Manual Analysis
Manual Analysis or review of code was done to identify new vulnerability or verify the vulnerabilities found during the static analysis. Contracts were completely manually analyzed, their logic was checked and compared with the one described in the whitepaper. Besides, the results of automated analysis were manually verified.

Gas Consumption
In this step we have checked the behaviour of smart contracts in production. Checks were done to know how much gas gets consumed and possibilities of optimization of code to reduce gas consumption.

Tools and Platforms used for Audit
Remix IDE, Truffle, Truffle Team, Ganache, Solhint, Mythril, Slither, SmartCheck.

Issue Categories

Every issue in this report has been assigned with a severity level. There are four levels of severity and each of them has been explained below.

High severity issues

A high severity issue or vulnerability means that your smart contract can be exploited. Issues on this level are critical to the smart contract's performance or functionality and we recommend these issues to be fixed before moving to a live environment.

Medium level severity issues

The issues marked as medium severity usually arise because of errors and deficiencies in the smart contract code. Issues on this level could potentially bring problems and they should still be fixed.

Low level severity issues

Low level severity issues can cause minor impact and or are just warnings that can remain unfixed for now. It would be better to fix these issues at some point in the future.

Informational

These are severity four issues which indicate an improvement request, a general question, a cosmetic or documentation error, or a request for information. There is low-to-no impact.

Number of issues per severity

Type	High	Medium	Low	Informational
Open	0	0	0	0
Acknowledged	0	0	0	4
Closed	0	1	4	3

Introduction

During the period of **August 7, 2021 to August 16, 2021** - QuillAudits Team performed a security audit for CoinMama Token smart contract.

The code for the audit was taken from following the official link: https://github.com/CoinMamaToken/CoinMamaToken/blob/main/CoinMama.sol

Note	Date	Commit hash
Version 1	11/08/2021	aec477aa949921a883faa694c690cc0ec26eacfd
Version 2	13/08/2021	c597a12d8b59497bb51c2780d865ae197484fc28
Version 3	16/08/2021	2d30af305f30b6cac1b0231766ff3ef534adf06f

Issues Found – Code Review / Manual Testing

High severity issues

No issues were found.

Medium severity issues

1. **Features mentioned have not been done**

   Description

   - The Anti-Dump (no one will be able to sell more than 1% of the supply within 24hrs, that's freezing of account when you reach that threshold)).
   - Initially, 10% of the supply should be burnt.
   - Different types of wallets settings with their token allocations are not there in code.

   Remediation
   A hardcoded limit(10000000 ether) has been used to check. Instead, the contract has to check for users not to sell more than 1% of the supply within 24hrs.
   We don’t get code for initial burn 10% of supply.

   Status: Fixed
   This issue was reported in Version 1 and found fixed in Version 2.

Low level severity issues

2. **Expected Identifier but got reserved keyword**

Line	Code
881	address public immutable uniswapV2Pair;
1384	uint256 public immutable minimumTokenBalanceForDividends;

Description
Expected Identifier but used reserve keyword “immutable”.

Remediation
Line no: 881, 1384 Remove the word immutable.

Status: Fixed
This issue was reported in Version 1 and found fixed in Version 2.

3. Wrong data location return parameter used

Line	Code
8	function _msgData() internal view virtual returns (bytes calldata) { this; // silence state mutability warning without generating bytecode - see https://github.com/ethereum/solidity/issues/2691 return msg.data; }
1033	function excludeMultipleAccountsFromFees(address[] calldata accounts, bool excluded) public onlyOwner { for(uint256 i = 0; i < accounts.length; i++) { _isExcludedFromFees[accounts[i]] = excluded; } emit ExcludeMultipleAccountsFromFees(accounts, excluded); }

Description
The data location must be “storage” or “memory” for the Return parameter in the function, but “calldata” is used.

Remediation
Line no: 8, 1033 change “calldata” to “memory”.

Status: Fixed
This issue was reported in Version 1 and found fixed in Version 2.

4. Owner can drain funds

Line	Code
1137	function withdraw(uint256 weiAmount) external onlyOwner{ msg.sender.transfer(weiAmount); }

Description
Line no: 1138: Owner can withdraw all funds from the contract.

Remediation
The owner must handle the private key of the owner's wallet very securely because if the private key is compromised, then it will create problems.

Status: Fixed
This issue was reported in Version 1 and found fixed in Version 3.

5. Function input parameters lack of check

Line	Code
1280	function setLiquidityFee(uint256 value) external onlyOwner{ liquidityFee = value; totalFees = BNBRewardsFee.add(liquidityFee).add(marketingAndBuybackFee); }
1286	function setMarketingFee(uint256 value) external onlyOwner{ marketingAndBuybackFee = value; totalFees = BNBRewardsFee.add(liquidityFee).add(marketingAndBuybackFee); }
1295	function setBurnFee(uint256 value) external onlyOwner{ burnFee = value; }

Description
Variable validation is not performed in below functions:

- setLiquidityFee
- setMarketingFee
- setBurnFee
- updateDividendTracker
- updateUniswapV2Router
- updateClaimWait
- processDividendTracker
- Withdraw
- setoperationalallet

Remediation
There should be some limit for values as this affects the calculation and if the address passed then it should not be 0 address.

Status: Fixed
This issue was reported in Version 1 and found fixed in Version 2.

Informational

6. Solidity version

pragma solidity ^0.6.2;

Description
Using the latest solidity will prevent any compiler-level bugs.

Remediation
Please use 0.8.6, which is the latest version.

Status: Acknowledged by the Auditee

7. Use SPDX License Identifier

Description
SPDX license identifier not provided in source file. Before publishing, consider adding a comment containing "SPDX-License-Identifier: <SPDX-License>".

Remediation
Use "// SPDX-License-Identifier: MIT License" for non-open-source code. Please see https://spdx.org for more information.

Status: Fixed
This issue was reported in Version 1 and found fixed in Version 2.

8. Typing mistake in a variable name

Line	Code
883	address payable public operationalallet = 0x388654d492375EcfCC8dFA56BDc87c7a2dab3380;

Description
Typing mistake in variable name operationalallet.

Remediation
Line no: 883 change variable name from operationalallet to operationalWallet.

Status: Fixed
This issue was reported in Version 1 and found fixed in Version 2.

9. Empty function used

Line	Code
648	function _beforeTokenTransfer(address from, address to, uint256 amount) internal virtual {}

Description
_beforeTokenTransfer this function is empty and used many times but has no effect on code or anything.

Remediation
Line no: 648 Add code in this function or remove this empty function from other functions.

Status: Acknowledged by the Auditee

10. Visibility of constructor ignored

Line	Code
24	constructor () public { address msgSender = _msgSender(); _owner = msgSender; emit OwnershipTransferred(address(0), msgSender); }
398	constructor(string memory name_, string memory symbol_) public { _name = name_; _symbol = symbol_; }
734	constructor(string memory _name, string memory _symbol) public ERC20(_name, _symbol) { }

Description

The visibility of the constructor is ignored.

Remediation

Line no: 24,398,734 remove word public from constructor.

Status: Acknowledged by the Auditee

11. Make variable constant

Line	Code
898	uint256 public maxSoldAmount = 10000000 ether;
848	address public deadWallet = 0x000000000000000000000000000000000000dEaD;

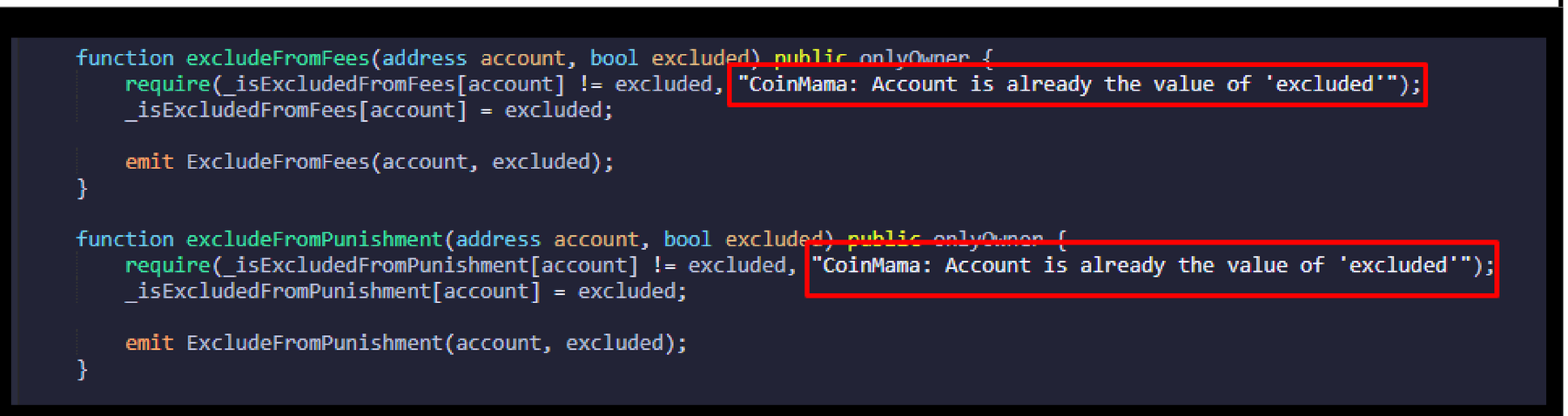
Description

This variables' value will be unchanged. So, please make them constant.
It will save some gas.

Remediation
Declare those variables as constant. Just put a constant keyword.

Status: Fixed
This issue was reported in Version 1 and found fixed in Version 2.

12. Ambiguous Error message

Line	Code
1006	

```
function excludeFromFees(address account, bool excluded) public onlyOwner {
    require(_isExcludedFromFees[account] != excluded, "CoinMama: Account is already the value of 'excluded'");
    _isExcludedFromFees[account] = excluded;

    emit ExcludeFromFees(account, excluded);
}

function excludeFromPunishment(address account, bool excluded) public onlyOwner {
    require(_isExcludedFromPunishment[account] != excluded, "CoinMama: Account is already the value of 'excluded'");
    _isExcludedFromPunishment[account] = excluded;

    emit ExcludeFromPunishment(account, excluded);
}
```

Description
The mentioned error messages are ambiguous.

Remediation
Please use a meaningful error message.

Status: Acknowledged by the Auditee

Functional test

Function Names	Testing results
receive	Passed
updateDividendTracker	Passed
updateUniswapV2Router	Passed
excludeFromFees	Ambiguous Error message
excludeFromPunishment	Ambiguous Error message
excludeMultipleAccountsFromFees	Passed
_msgData	Passed
setTrigger	Passed
setAutomatedMarketMakerPair	Passed
_setAutomatedMarketMakerPair	Passed
updateGasForProcessing	Passed
updateClaimWait	Passed
getClaimWait	Passed
getTotalDividendsDistributed	Passed
isExcludedFromFees	Passed
withdrawableDividendOf	Passed
dividendTokenBalanceOf	Passed
getAccountDividendsInfo	Passed
getAccountDividendsInfoAtIndex	Passed
processDividendTracker	Passed

Function Names	Testing results
claim	Passed
getLastProcessedIndex	Passed
setSwapTokensAtAmt	Passed
withdraw	Passed
getNumberOfDividendTokenHolders	Passed
getUserPunishmentEndTime	Passed
_transfer	Passed
setBNBRewardsfee	Passed
swapAndSendToFee	Passed
excludeFromDividends	Passed
setLiquidityFee	Passed
setMarketingFee	Passed
setoperationalallet	Passed
setBurnFee	Passed
swapAndLiquify	Passed
swapTokensForEth	Passed
addLiquidity	Passed
swapAndSendDividends	Passed
name	Passed
symbol	Passed
decimals	Passed

Function Names	Testing results
totalSupply	Passed
balanceOf	Passed
transfer	Passed
allowance	Passed
approve	Passed
transferFrom	Passed
increaseAllowance	Passed
decreaseAllowance	Passed
_transfer	Passed
_mint	Passed
_burn	Passed
_approve	Passed
_beforeTokenTransfer	Empty function used
owner	Passed
onlyOwner	Passed
renounceOwnership	Passed
previousOwner	Passed
getUnlockTime	Passed
transferOwnership	Passed

Automated Testing

Slither

```
INFO:Detectors:
CoinMama.withdraw(uint256) (CoinMama.sol#1137-1139) sends eth to arbitrary user
        Dangerous calls:
        - msg.sender.transfer(weiAmount) (CoinMama.sol#1138)
CoinMama.addLiquidity(uint256,uint256) (CoinMama.sol#1343-1358) sends eth to arbitrary user
        Dangerous calls:
        - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,deadWallet,block.timestamp) (CoinMama.sol#1349-
1356)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#functions-that-send-ether-to-arbitrary-destinations
INFO:Detectors:
Reentrancy in CoinMama._transfer(address,address,uint256) (CoinMama.sol#1150-1259):
        External calls:
        - swapAndSendToFee(operationalallet,marketingTokens) (CoinMama.sol#1210)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (C
oinMama.sol#1333-1339)
        External calls sending eth:
        - swapAndSendToFee(operationalallet,marketingTokens) (CoinMama.sol#1210)
                - wallet.transfer(newBalance) (CoinMama.sol#1273)
        State variables written after the call(s):
        - lastSwap = block.timestamp (CoinMama.sol#1211)
Reentrancy in CoinMama._transfer(address,address,uint256) (CoinMama.sol#1150-1259):
        External calls:
        - swapAndSendToFee(operationalallet,marketingTokens) (CoinMama.sol#1210)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (C
oinMama.sol#1333-1339)
        - swapAndLiquify(swapTokens) (CoinMama.sol#1215)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,deadWallet,block.timestamp) (CoinMama.s
ol#1349-1356)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (C
oinMama.sol#1333-1339)
        - swapAndSendDividends(sellTokens) (CoinMama.sol#1218)
                - (success) = address(dividendTracker).call{value: dividends}() (CoinMama.sol#1363)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (C
oinMama.sol#1333-1339)
        External calls sending eth:
        - swapAndSendToFee(operationalallet,marketingTokens) (CoinMama.sol#1210)
                - wallet.transfer(newBalance) (CoinMama.sol#1273)
        - swapAndLiquify(swapTokens) (CoinMama.sol#1215)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,deadWallet,block.timestamp) (CoinMama.s
ol#1349-1356)
        - swapAndSendDividends(sellTokens) (CoinMama.sol#1218)
                - (success) = address(dividendTracker).call{value: dividends}() (CoinMama.sol#1363)
        State variables written after the call(s):
        - super._transfer(from,address(this),fees) (CoinMama.sol#1240)
                - _balances[sender] = _balances[sender].sub(amount,ERC20: transfer amount exceeds balance) (CoinMama.sol#564)
                - _balances[recipient] = _balances[recipient].add(amount) (CoinMama.sol#565)
        - super._transfer(from,deadWallet,burnAmt) (CoinMama.sol#1241)
                - _balances[sender] = _balances[sender].sub(amount,ERC20: transfer amount exceeds balance) (CoinMama.sol#564)
                - _balances[recipient] = _balances[recipient].add(amount) (CoinMama.sol#565)
        - super._transfer(from,to,amount) (CoinMama.sol#1244)
                - _balances[sender] = _balances[sender].sub(amount,ERC20: transfer amount exceeds balance) (CoinMama.sol#564)
                - _balances[recipient] = _balances[recipient].add(amount) (CoinMama.sol#565)
        - swapping = false (CoinMama.sol#1220)
Reentrancy in DividendPayingToken._withdrawDividendOfUser(address) (CoinMama.sol#777-793):
        External calls:
        - (success) = user.call{gas: 3000,value: _withdrawableDividend}() (CoinMama.sol#782)
        State variables written after the call(s):
        - withdrawnDividends[user] = withdrawnDividends[user].sub(_withdrawableDividend) (CoinMama.sol#785)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities
INFO:Detectors:
Reentrancy in CoinMama.updateDividendTracker(address) (CoinMama.sol#995-1011):
        External calls:
        - newDividendTracker.excludeFromDividends(address(newDividendTracker)) (CoinMama.sol#1002)
        - newDividendTracker.excludeFromDividends(address(this)) (CoinMama.sol#1003)
        - newDividendTracker.excludeFromDividends(owner()) (CoinMama.sol#1004)
        - newDividendTracker.excludeFromDividends(address(0)) (CoinMama.sol#1005)
        - newDividendTracker.excludeFromDividends(address(uniswapV2Router)) (CoinMama.sol#1006)
        State variables written after the call(s):
        - dividendTracker = newDividendTracker (CoinMama.sol#1010)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1
INFO:Detectors:
CoinMama._transfer(address,address,uint256).claims (CoinMama.sol#1252) is a local variable never initialized
CoinMama._transfer(address,address,uint256).lastProcessedIndex (CoinMama.sol#1252) is a local variable never initialized
```

```
CoinMama._transfer(address,address,uint256).claims (CoinMama.sol#1252) is a local variable never initialized
CoinMama._transfer(address,address,uint256).lastProcessedIndex (CoinMama.sol#1252) is a local variable never initialized
CoinMama._transfer(address,address,uint256).iterations (CoinMama.sol#1252) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables
INFO:Detectors:
CoinMama.claim() (CoinMama.sol#1125-1127) ignores return value by dividendTracker.processAccount(msg.sender,false) (CoinMama.sol#1126)
CoinMama._transfer(address,address,uint256) (CoinMama.sol#1150-1259) ignores return value by dividendTracker.process(gas) (CoinMama.sol#1
252-1257)
CoinMama.addLiquidity(uint256,uint256) (CoinMama.sol#1343-1358) ignores return value by uniswapV2Router.addLiquidityETH{value: ethAmount}
(address(this),tokenAmount,0,0,deadWallet,block.timestamp) (CoinMama.sol#1349-1356)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return
INFO:Detectors:
DividendPayingToken.constructor(string,string)._name (CoinMama.sol#734) shadows:
        - ERC20._name (CoinMama.sol#386) (state variable)
DividendPayingToken.constructor(string,string)._symbol (CoinMama.sol#734) shadows:
        - ERC20._symbol (CoinMama.sol#387) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing
INFO:Detectors:
CoinMama.setoperationalallet(address).newwallet (CoinMama.sol#1291) lacks a zero-check on :
                - operationalallet = newwallet (CoinMama.sol#1292)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
INFO:Detectors:
Variable 'CoinMama._transfer(address,address,uint256).claims (CoinMama.sol#1252)' in CoinMama._transfer(address,address,uint256) (CoinMam
a.sol#1150-1259) potentially used before declaration: ProcessedDividendTracker(iterations,claims,lastProcessedIndex,true,gas,tx.origin) (
CoinMama.sol#1253)
Variable 'CoinMama._transfer(address,address,uint256).lastProcessedIndex (CoinMama.sol#1252)' in CoinMama._transfer(address,address,uint2
56) (CoinMama.sol#1150-1259) potentially used before declaration: ProcessedDividendTracker(iterations,claims,lastProcessedIndex,true,gas,
tx.origin) (CoinMama.sol#1253)
Variable 'CoinMama._transfer(address,address,uint256).iterations (CoinMama.sol#1252)' in CoinMama._transfer(address,address,uint256) (Coi
nMama.sol#1150-1259) potentially used before declaration: ProcessedDividendTracker(iterations,claims,lastProcessedIndex,true,gas,tx.origi
n) (CoinMama.sol#1253)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables
INFO:Detectors:
Reentrancy in CoinMama._transfer(address,address,uint256) (CoinMama.sol#1150-1259):
        External calls:
        - swapAndSendToFee(operationalallet,marketingTokens) (CoinMama.sol#1210)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (C
oinMama.sol#1333-1339)
```

```
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (C
oinMama.sol#1333-1339)
        - swapAndLiquify(swapTokens) (CoinMama.sol#1215)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,deadWallet,block.timestamp) (CoinMama.s
ol#1349-1356)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (C
oinMama.sol#1333-1339)
        External calls sending eth:
        - swapAndSendToFee(operationalallet,marketingTokens) (CoinMama.sol#1210)
                - wallet.transfer(newBalance) (CoinMama.sol#1273)
        - swapAndLiquify(swapTokens) (CoinMama.sol#1215)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,deadWallet,block.timestamp) (CoinMama.s
ol#1349-1356)
        State variables written after the call(s):
        - swapAndLiquify(swapTokens) (CoinMama.sol#1215)
                - _allowances[owner][spender] = amount (CoinMama.sol#630)
Reentrancy in CoinMama._transfer(address,address,uint256) (CoinMama.sol#1150-1259):
        External calls:
        - swapAndSendToFee(operationalallet,marketingTokens) (CoinMama.sol#1210)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (C
oinMama.sol#1333-1339)
        - swapAndLiquify(swapTokens) (CoinMama.sol#1215)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,deadWallet,block.timestamp) (CoinMama.s
ol#1349-1356)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (C
oinMama.sol#1333-1339)
        - swapAndSendDividends(sellTokens) (CoinMama.sol#1218)
                - (success) = address(dividendTracker).call{value: dividends}() (CoinMama.sol#1363)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (C
oinMama.sol#1333-1339)
        External calls sending eth:
        - swapAndSendToFee(operationalallet,marketingTokens) (CoinMama.sol#1210)
                - wallet.transfer(newBalance) (CoinMama.sol#1273)
        - swapAndLiquify(swapTokens) (CoinMama.sol#1215)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,deadWallet,block.timestamp) (CoinMama.s
ol#1349-1356)
        - swapAndSendDividends(sellTokens) (CoinMama.sol#1218)
                - (success) = address(dividendTracker).call{value: dividends}() (CoinMama.sol#1363)
```

```
ol#1349-1356)
        - swapAndSendDividends(sellTokens) (CoinMama.sol#1218)
                - (success) = address(dividendTracker).call{value: dividends}() (CoinMama.sol#1363)
        State variables written after the call(s):
        - swapAndSendDividends(sellTokens) (CoinMama.sol#1218)
                - _allowances[owner][spender] = amount (CoinMama.sol#630)
Reentrancy in CoinMama.constructor() (CoinMama.sol#951-989):
        External calls:
        - _uniswapV2Pair = IUniswapV2Factory(_uniswapV2Router.factory()).createPair(address(this),_uniswapV2Router.WETH()) (CoinMama.sol#
958-959)
        State variables written after the call(s):
        - uniswapV2Pair = _uniswapV2Pair (CoinMama.sol#962)
        - uniswapV2Router = _uniswapV2Router (CoinMama.sol#961)
Reentrancy in CoinMama.constructor() (CoinMama.sol#951-989):
        External calls:
        - _uniswapV2Pair = IUniswapV2Factory(_uniswapV2Router.factory()).createPair(address(this),_uniswapV2Router.WETH()) (CoinMama.sol#
958-959)
        - _setAutomatedMarketMakerPair(_uniswapV2Pair,true) (CoinMama.sol#964)
                - dividendTracker.excludeFromDividends(pair) (CoinMama.sol#1056)
        - dividendTracker.excludeFromDividends(address(dividendTracker)) (CoinMama.sol#967)
        - dividendTracker.excludeFromDividends(address(this)) (CoinMama.sol#968)
        - dividendTracker.excludeFromDividends(owner()) (CoinMama.sol#969)
        - dividendTracker.excludeFromDividends(address(0)) (CoinMama.sol#970)
        - dividendTracker.excludeFromDividends(address(_uniswapV2Router)) (CoinMama.sol#971)
        State variables written after the call(s):
        - _mint(owner(),1000000000 * (10 ** 18)) (CoinMama.sol#988)
                - _balances[account] = _balances[account].add(amount) (CoinMama.sol#584)
        - excludeFromFees(owner(),true) (CoinMama.sol#974)
                - _isExcludedFromFees[account] = excluded (CoinMama.sol#1021)
        - excludeFromFees(operationalallet,true) (CoinMama.sol#975)
                - _isExcludedFromFees[account] = excluded (CoinMama.sol#1021)
        - excludeFromFees(address(this),true) (CoinMama.sol#976)
                - _isExcludedFromFees[account] = excluded (CoinMama.sol#1021)
        - excludeFromPunishment(owner(),true) (CoinMama.sol#978)
                - _isExcludedFromPunishment[account] = excluded (CoinMama.sol#1028)
        - excludeFromPunishment(operationalallet,true) (CoinMama.sol#979)
                - _isExcludedFromPunishment[account] = excluded (CoinMama.sol#1028)
        - excludeFromPunishment(address(this),true) (CoinMama.sol#980)
```

```
        - excludeFromFees(operationalallet,true) (CoinMama.sol#975)
                - _isExcludedFromFees[account] = excluded (CoinMama.sol#1021)
        - excludeFromFees(address(this),true) (CoinMama.sol#976)
                - _isExcludedFromFees[account] = excluded (CoinMama.sol#1021)
        - excludeFromPunishment(owner(),true) (CoinMama.sol#978)
                - _isExcludedFromPunishment[account] = excluded (CoinMama.sol#1028)
        - excludeFromPunishment(operationalallet,true) (CoinMama.sol#979)
                - _isExcludedFromPunishment[account] = excluded (CoinMama.sol#1028)
        - excludeFromPunishment(address(this),true) (CoinMama.sol#980)
                - _isExcludedFromPunishment[account] = excluded (CoinMama.sol#1028)
        - _mint(owner(),1000000000 * (10 ** 18)) (CoinMama.sol#988)
                - _totalSupply = _totalSupply.add(amount) (CoinMama.sol#583)
Reentrancy in CoinMamaDividendTracker.processAccount(address,bool) (CoinMama.sol#1567-1577):
        External calls:
        - amount = _withdrawDividendOfUser(account) (CoinMama.sol#1568)
                - (success) = user.call{gas: 3000,value: _withdrawableDividend}() (CoinMama.sol#782)
        State variables written after the call(s):
        - lastClaimTimes[account] = block.timestamp (CoinMama.sol#1571)
Reentrancy in CoinMama.swapAndLiquify(uint256) (CoinMama.sol#1299-1320):
        External calls:
        - swapTokensForEth(half) (CoinMama.sol#1311)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (C
oinMama.sol#1333-1339)
        - addLiquidity(otherHalf,newBalance) (CoinMama.sol#1317)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,deadWallet,block.timestamp) (CoinMama.s
ol#1349-1356)
        External calls sending eth:
        - addLiquidity(otherHalf,newBalance) (CoinMama.sol#1317)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,deadWallet,block.timestamp) (CoinMama.s
ol#1349-1356)
        State variables written after the call(s):
        - addLiquidity(otherHalf,newBalance) (CoinMama.sol#1317)
                - _allowances[owner][spender] = amount (CoinMama.sol#630)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2
INFO:Detectors:
Reentrancy in CoinMama._setAutomatedMarketMakerPair(address,bool) (CoinMama.sol#1051-1060):
        External calls:
        - dividendTracker.excludeFromDividends(pair) (CoinMama.sol#1056)
```

```
        External calls:
        - dividendTracker.excludeFromDividends(pair) (CoinMama.sol#1056)
        Event emitted after the call(s):
        - SetAutomatedMarketMakerPair(pair,value) (CoinMama.sol#1059)
Reentrancy in CoinMama._transfer(address,address,uint256) (CoinMama.sol#1150-1259):
        External calls:
        - swapAndSendToFee(operationalallet,marketingTokens) (CoinMama.sol#1210)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (C
oinMama.sol#1333-1339)
        - swapAndLiquify(swapTokens) (CoinMama.sol#1215)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,deadWallet,block.timestamp) (CoinMama.s
ol#1349-1356)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (C
oinMama.sol#1333-1339)
        External calls sending eth:
        - swapAndSendToFee(operationalallet,marketingTokens) (CoinMama.sol#1210)
                - wallet.transfer(newBalance) (CoinMama.sol#1273)
        - swapAndLiquify(swapTokens) (CoinMama.sol#1215)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,deadWallet,block.timestamp) (CoinMama.s
ol#1349-1356)
        Event emitted after the call(s):
        - Approval(owner,spender,amount) (CoinMama.sol#631)
                - swapAndLiquify(swapTokens) (CoinMama.sol#1215)
        - SwapAndLiquify(half,newBalance,otherHalf) (CoinMama.sol#1319)
                - swapAndLiquify(swapTokens) (CoinMama.sol#1215)
Reentrancy in CoinMama._transfer(address,address,uint256) (CoinMama.sol#1150-1259):
        External calls:
        - swapAndSendToFee(operationalallet,marketingTokens) (CoinMama.sol#1210)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (C
oinMama.sol#1333-1339)
        - swapAndLiquify(swapTokens) (CoinMama.sol#1215)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,deadWallet,block.timestamp) (CoinMama.s
ol#1349-1356)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (C
oinMama.sol#1333-1339)
        - swapAndSendDividends(sellTokens) (CoinMama.sol#1218)
                - (success) = address(dividendTracker).call{value: dividends}() (CoinMama.sol#1363)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (C
```

```
        - swapAndSendDividends(sellTokens) (CoinMama.sol#1218)
                - (success) = address(dividendTracker).call{value: dividends}() (CoinMama.sol#1363)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (C
oinMama.sol#1333-1339)
        External calls sending eth:
        - swapAndSendToFee(operationalallet,marketingTokens) (CoinMama.sol#1210)
                - wallet.transfer(newBalance) (CoinMama.sol#1273)
        - swapAndLiquify(swapTokens) (CoinMama.sol#1215)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,deadWallet,block.timestamp) (CoinMama.s
ol#1349-1356)
        - swapAndSendDividends(sellTokens) (CoinMama.sol#1218)
                - (success) = address(dividendTracker).call{value: dividends}() (CoinMama.sol#1363)
        Event emitted after the call(s):
        - Approval(owner,spender,amount) (CoinMama.sol#631)
                - swapAndSendDividends(sellTokens) (CoinMama.sol#1218)
        - SendDividends(tokens,dividends) (CoinMama.sol#1366)
                - swapAndSendDividends(sellTokens) (CoinMama.sol#1218)
        - Transfer(sender,recipient,amount) (CoinMama.sol#566)
                - super._transfer(from,deadWallet,burnAmt) (CoinMama.sol#1241)
        - Transfer(sender,recipient,amount) (CoinMama.sol#566)
                - super._transfer(from,address(this),fees) (CoinMama.sol#1240)
        - Transfer(sender,recipient,amount) (CoinMama.sol#566)
                - super._transfer(from,to,amount) (CoinMama.sol#1244)
Reentrancy in CoinMama._transfer(address,address,uint256) (CoinMama.sol#1150-1259):
        External calls:
        - swapAndSendToFee(operationalallet,marketingTokens) (CoinMama.sol#1210)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (C
oinMama.sol#1333-1339)
        - swapAndLiquify(swapTokens) (CoinMama.sol#1215)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,deadWallet,block.timestamp) (CoinMama.s
ol#1349-1356)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (C
oinMama.sol#1333-1339)
        - swapAndSendDividends(sellTokens) (CoinMama.sol#1218)
                - (success) = address(dividendTracker).call{value: dividends}() (CoinMama.sol#1363)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (C
oinMama.sol#1333-1339)
        - dividendTracker.setBalance(address(from),balanceOf(from)) (CoinMama.sol#1246)
```

```
        - dividendTracker.setBalance(address(from),balanceOf(from)) (CoinMama.sol#1246)
        - dividendTracker.setBalance(address(to),balanceOf(to)) (CoinMama.sol#1247)
        - dividendTracker.process(gas) (CoinMama.sol#1252-1257)
        External calls sending eth:
        - swapAndSendToFee(operationalallet,marketingTokens) (CoinMama.sol#1210)
                - wallet.transfer(newBalance) (CoinMama.sol#1273)
        - swapAndLiquify(swapTokens) (CoinMama.sol#1215)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,deadWallet,block.timestamp) (CoinMama.s
ol#1349-1356)
        - swapAndSendDividends(sellTokens) (CoinMama.sol#1218)
                - (success) = address(dividendTracker).call{value: dividends}() (CoinMama.sol#1363)
        Event emitted after the call(s):
        - ProcessedDividendTracker(iterations,claims,lastProcessedIndex,true,gas,tx.origin) (CoinMama.sol#1253)
Reentrancy in CoinMama.constructor() (CoinMama.sol#951-989):
        External calls:
        - _uniswapV2Pair = IUniswapV2Factory(_uniswapV2Router.factory()).createPair(address(this),_uniswapV2Router.WETH()) (CoinMama.sol#
958-959)
        - _setAutomatedMarketMakerPair(_uniswapV2Pair,true) (CoinMama.sol#964)
                - dividendTracker.excludeFromDividends(pair) (CoinMama.sol#1056)
        Event emitted after the call(s):
        - SetAutomatedMarketMakerPair(pair,value) (CoinMama.sol#1059)
                - _setAutomatedMarketMakerPair(_uniswapV2Pair,true) (CoinMama.sol#964)
Reentrancy in CoinMama.constructor() (CoinMama.sol#951-989):
        External calls:
        - _uniswapV2Pair = IUniswapV2Factory(_uniswapV2Router.factory()).createPair(address(this),_uniswapV2Router.WETH()) (CoinMama.sol#
958-959)
        - _setAutomatedMarketMakerPair(_uniswapV2Pair,true) (CoinMama.sol#964)
                - dividendTracker.excludeFromDividends(pair) (CoinMama.sol#1056)
        - dividendTracker.excludeFromDividends(address(dividendTracker)) (CoinMama.sol#967)
        - dividendTracker.excludeFromDividends(address(this)) (CoinMama.sol#968)
        - dividendTracker.excludeFromDividends(owner()) (CoinMama.sol#969)
        - dividendTracker.excludeFromDividends(address(0)) (CoinMama.sol#970)
        - dividendTracker.excludeFromDividends(address(_uniswapV2Router)) (CoinMama.sol#971)
        Event emitted after the call(s):
        - ExcludeFromFees(account,excluded) (CoinMama.sol#1023)
                - excludeFromFees(operationalallet,true) (CoinMama.sol#975)
        - ExcludeFromFees(account,excluded) (CoinMama.sol#1023)
                - excludeFromFees(address(this),true) (CoinMama.sol#976)
```

```
                - excludeFromFees(operationalallet,true) (CoinMama.sol#975)
        - ExcludeFromFees(account,excluded) (CoinMama.sol#1023)
                - excludeFromFees(address(this),true) (CoinMama.sol#976)
        - ExcludeFromFees(account,excluded) (CoinMama.sol#1023)
                - excludeFromFees(owner(),true) (CoinMama.sol#974)
        - ExcludeFromPunishment(account,excluded) (CoinMama.sol#1030)
                - excludeFromPunishment(owner(),true) (CoinMama.sol#978)
        - ExcludeFromPunishment(account,excluded) (CoinMama.sol#1030)
                - excludeFromPunishment(address(this),true) (CoinMama.sol#980)
        - ExcludeFromPunishment(account,excluded) (CoinMama.sol#1030)
                - excludeFromPunishment(operationalallet,true) (CoinMama.sol#979)
        - Transfer(address(0),account,amount) (CoinMama.sol#585)
                - _mint(owner(),1000000000 * (10 ** 18)) (CoinMama.sol#988)
Reentrancy in CoinMamaDividendTracker.processAccount(address,bool) (CoinMama.sol#1567-1577):
        External calls:
        - amount = _withdrawDividendOfUser(account) (CoinMama.sol#1568)
                - (success) = user.call{gas: 3000,value: _withdrawableDividend}() (CoinMama.sol#782)
        Event emitted after the call(s):
        - Claim(account,amount,automatic) (CoinMama.sol#1572)
Reentrancy in CoinMama.processDividendTracker(uint256) (CoinMama.sol#1120-1123):
        External calls:
        - (iterations,claims,lastProcessedIndex) = dividendTracker.process(gas) (CoinMama.sol#1121)
        Event emitted after the call(s):
        - ProcessedDividendTracker(iterations,claims,lastProcessedIndex,false,gas,tx.origin) (CoinMama.sol#1122)
Reentrancy in CoinMama.swapAndLiquify(uint256) (CoinMama.sol#1299-1320):
        External calls:
        - swapTokensForEth(half) (CoinMama.sol#1311)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (C
oinMama.sol#1333-1339)
        - addLiquidity(otherHalf,newBalance) (CoinMama.sol#1317)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,deadWallet,block.timestamp) (CoinMama.s
ol#1349-1356)
        External calls sending eth:
        - addLiquidity(otherHalf,newBalance) (CoinMama.sol#1317)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,deadWallet,block.timestamp) (CoinMama.s
ol#1349-1356)
        Event emitted after the call(s):
        - Approval(owner,spender,amount) (CoinMama.sol#631)
```

```
        Event emitted after the call(s):
        - Approval(owner,spender,amount) (CoinMama.sol#631)
                - addLiquidity(otherHalf,newBalance) (CoinMama.sol#1317)
        - SwapAndLiquify(half,newBalance,otherHalf) (CoinMama.sol#1319)
Reentrancy in CoinMama.swapAndSendDividends(uint256) (CoinMama.sol#1360-1368):
        External calls:
        - swapTokensForEth(tokens) (CoinMama.sol#1361)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (C
oinMama.sol#1333-1339)
        - (success) = address(dividendTracker).call{value: dividends}() (CoinMama.sol#1363)
        External calls sending eth:
        - (success) = address(dividendTracker).call{value: dividends}() (CoinMama.sol#1363)
        Event emitted after the call(s):
        - SendDividends(tokens,dividends) (CoinMama.sol#1366)
Reentrancy in CoinMama.updateDividendTracker(address) (CoinMama.sol#995-1011):
        External calls:
        - newDividendTracker.excludeFromDividends(address(newDividendTracker)) (CoinMama.sol#1002)
        - newDividendTracker.excludeFromDividends(address(this)) (CoinMama.sol#1003)
        - newDividendTracker.excludeFromDividends(owner()) (CoinMama.sol#1004)
        - newDividendTracker.excludeFromDividends(address(0)) (CoinMama.sol#1005)
        - newDividendTracker.excludeFromDividends(address(uniswapV2Router)) (CoinMama.sol#1006)
        Event emitted after the call(s):
        - UpdateDividendTracker(newAddress,address(dividendTracker)) (CoinMama.sol#1008)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
INFO:Detectors:
CoinMama._transfer(address,address,uint256) (CoinMama.sol#1150-1259) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp >= userCanSellTime[from],You are punished) (CoinMama.sol#1158)
        - timePassed >= 60 * 60 (CoinMama.sol#1208)
CoinMamaDividendTracker.getAccount(address) (CoinMama.sol#1431-1474) uses timestamp for comparisons
        Dangerous comparisons:
        - nextClaimTime > block.timestamp (CoinMama.sol#1471-1473)
CoinMamaDividendTracker.canAutoClaim(uint256) (CoinMama.sol#1495-1501) uses timestamp for comparisons
        Dangerous comparisons:
        - lastClaimTime > block.timestamp (CoinMama.sol#1496)
        - block.timestamp.sub(lastClaimTime) >= claimWait (CoinMama.sol#1500)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
```

```
INFO:Detectors:
Context._msgData() (CoinMama.sol#8-11) is never used and should be removed
DividendPayingToken._transfer(address,address,uint256) (CoinMama.sol#833-839) is never used and should be removed
SafeMath.mod(uint256,uint256) (CoinMama.sol#1701-1703) is never used and should be removed
SafeMath.mod(uint256,uint256,string) (CoinMama.sol#1717-1720) is never used and should be removed
SafeMathInt.abs(int256) (CoinMama.sol#1771-1774) is never used and should be removed
SafeMathInt.div(int256,int256) (CoinMama.sol#1742-1748) is never used and should be removed
SafeMathInt.mul(int256,int256) (CoinMama.sol#1730-1737) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
CoinMama.totalFees (CoinMama.sol#897) is set pre-construction with a non-constant function or state variable:
        - BNBRewardsFee.add(liquidityFee).add(marketingAndBuybackFee)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#function-initializing-state-variables
INFO:Detectors:
Low level call in DividendPayingToken._withdrawDividendOfUser(address) (CoinMama.sol#777-793):
        - (success) = user.call{gas: 3000,value: _withdrawableDividend}() (CoinMama.sol#782)
Low level call in CoinMama.swapAndSendDividends(uint256) (CoinMama.sol#1360-1368):
        - (success) = address(dividendTracker).call{value: dividends}() (CoinMama.sol#1363)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Function IUniswapV2Pair.DOMAIN_SEPARATOR() (CoinMama.sol#93) is not in mixedCase
Function IUniswapV2Pair.PERMIT_TYPEHASH() (CoinMama.sol#94) is not in mixedCase
Function IUniswapV2Pair.MINIMUM_LIQUIDITY() (CoinMama.sol#111) is not in mixedCase
Function IUniswapV2Router01.WETH() (CoinMama.sol#148) is not in mixedCase
Parameter DividendPayingToken.dividendOf(address)._owner (CoinMama.sol#799) is not in mixedCase
Parameter DividendPayingToken.withdrawableDividendOf(address)._owner (CoinMama.sol#806) is not in mixedCase
Parameter DividendPayingToken.withdrawnDividendOf(address)._owner (CoinMama.sol#813) is not in mixedCase
Parameter DividendPayingToken.accumulativeDividendOf(address)._owner (CoinMama.sol#823) is not in mixedCase
Constant DividendPayingToken.magnitude (CoinMama.sol#714) is not in UPPER_CASE_WITH_UNDERSCORES
Parameter CoinMama.setTrigger(bool)._bool (CoinMama.sol#1041) is not in mixedCase
Variable CoinMama.BNBRewardsFee (CoinMama.sol#893) is not in mixedCase
Parameter CoinMamaDividendTracker.getAccount(address)._account (CoinMama.sol#1431) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
Redundant expression "this (CoinMama.sol#9)" inContext (CoinMama.sol#3-12)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements
INFO:Detectors:
```

```
INFO:Detectors:
Redundant expression "this (CoinMama.sol#9)" inContext (CoinMama.sol#3-12)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements
INFO:Detectors:
Reentrancy in CoinMama._transfer(address,address,uint256) (CoinMama.sol#1150-1259):
        External calls:
        - swapAndSendToFee(operationalallet,marketingTokens) (CoinMama.sol#1210)
                - wallet.transfer(newBalance) (CoinMama.sol#1273)
        State variables written after the call(s):
        - lastSwap = block.timestamp (CoinMama.sol#1211)
Reentrancy in CoinMama._transfer(address,address,uint256) (CoinMama.sol#1150-1259):
        External calls:
        - swapAndSendToFee(operationalallet,marketingTokens) (CoinMama.sol#1210)
                - wallet.transfer(newBalance) (CoinMama.sol#1273)
        External calls sending eth:
        - swapAndSendToFee(operationalallet,marketingTokens) (CoinMama.sol#1210)
                - wallet.transfer(newBalance) (CoinMama.sol#1273)
        - swapAndLiquify(swapTokens) (CoinMama.sol#1215)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,deadWallet,block.timestamp) (CoinMama.s
ol#1349-1356)
        State variables written after the call(s):
        - swapAndLiquify(swapTokens) (CoinMama.sol#1215)
                - _allowances[owner][spender] = amount (CoinMama.sol#630)
        Event emitted after the call(s):
        - Approval(owner,spender,amount) (CoinMama.sol#631)
                - swapAndLiquify(swapTokens) (CoinMama.sol#1215)
        - SwapAndLiquify(half,newBalance,otherHalf) (CoinMama.sol#1319)
                - swapAndLiquify(swapTokens) (CoinMama.sol#1215)
Reentrancy in CoinMama._transfer(address,address,uint256) (CoinMama.sol#1150-1259):
        External calls:
        - swapAndSendToFee(operationalallet,marketingTokens) (CoinMama.sol#1210)
                - wallet.transfer(newBalance) (CoinMama.sol#1273)
        External calls sending eth:
        - swapAndSendToFee(operationalallet,marketingTokens) (CoinMama.sol#1210)
                - wallet.transfer(newBalance) (CoinMama.sol#1273)
        - swapAndLiquify(swapTokens) (CoinMama.sol#1215)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,deadWallet,block.timestamp) (CoinMama.s
```

```
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,deadWallet,block.timestamp) (CoinMama.s
ol#1349-1356)
        - swapAndSendDividends(sellTokens) (CoinMama.sol#1218)
                - (success) = address(dividendTracker).call{value: dividends}() (CoinMama.sol#1363)
        State variables written after the call(s):
        - swapAndSendDividends(sellTokens) (CoinMama.sol#1218)
                - _allowances[owner][spender] = amount (CoinMama.sol#630)
        - super._transfer(from,address(this),fees) (CoinMama.sol#1240)
                - _balances[sender] = _balances[sender].sub(amount,ERC20: transfer amount exceeds balance) (CoinMama.sol#564)
                - _balances[recipient] = _balances[recipient].add(amount) (CoinMama.sol#565)
        - super._transfer(from,deadWallet,burnAmt) (CoinMama.sol#1241)
                - _balances[sender] = _balances[sender].sub(amount,ERC20: transfer amount exceeds balance) (CoinMama.sol#564)
                - _balances[recipient] = _balances[recipient].add(amount) (CoinMama.sol#565)
        - super._transfer(from,to,amount) (CoinMama.sol#1244)
                - _balances[sender] = _balances[sender].sub(amount,ERC20: transfer amount exceeds balance) (CoinMama.sol#564)
                - _balances[recipient] = _balances[recipient].add(amount) (CoinMama.sol#565)
        - swapping = false (CoinMama.sol#1220)
        Event emitted after the call(s):
        - Approval(owner,spender,amount) (CoinMama.sol#631)
                - swapAndSendDividends(sellTokens) (CoinMama.sol#1218)
        - ProcessedDividendTracker(iterations,claims,lastProcessedIndex,true,gas,tx.origin) (CoinMama.sol#1253)
        - SendDividends(tokens,dividends) (CoinMama.sol#1366)
                - swapAndSendDividends(sellTokens) (CoinMama.sol#1218)
        - Transfer(sender,recipient,amount) (CoinMama.sol#566)
                - super._transfer(from,address(this),fees) (CoinMama.sol#1240)
        - Transfer(sender,recipient,amount) (CoinMama.sol#566)
                - super._transfer(from,to,amount) (CoinMama.sol#1244)
        - Transfer(sender,recipient,amount) (CoinMama.sol#566)
                - super._transfer(from,deadWallet,burnAmt) (CoinMama.sol#1241)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-4
INFO:Detectors:
Variable IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountADesired (CoinMama.sol#15
3) is too similar to IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountBDesired (Coi
nMama.sol#154)
Variable DividendPayingToken._withdrawDividendOfUser(address)._withdrawableDividend (CoinMama.sol#778) is too similar to CoinMamaDividend
Tracker.getAccount(address).withdrawableDividends (CoinMama.sol#1436)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar
INFO:Detectors:
```

```
INFO:Detectors:
CoinMama.constructor() (CoinMama.sol#951-989) uses literals with too many digits:
        - _mint(owner(),1000000000 * (10 ** 18)) (CoinMama.sol#988)
CoinMama.updateGasForProcessing(uint256) (CoinMama.sol#1063-1068) uses literals with too many digits:
        - require(bool,string)(newValue >= 200000 && newValue <= 500000,CoinMama: gasForProcessing must be between 200,000 and 500,000) (
CoinMama.sol#1064)
CoinMama._transfer(address,address,uint256) (CoinMama.sol#1150-1259) uses literals with too many digits:
        - amount = amount.sub(amount.mul(10).div(100000)) (CoinMama.sol#1184)
CoinMama.slitherConstructorVariables() (CoinMama.sol#877-1369) uses literals with too many digits:
        - deadWallet = 0x000000000000000000000000000000000000dEaD (CoinMama.sol#884)
CoinMama.slitherConstructorVariables() (CoinMama.sol#877-1369) uses literals with too many digits:
        - swapTokensAtAmount = 20000000000 * (10 ** 18) (CoinMama.sol#891)
CoinMama.slitherConstructorVariables() (CoinMama.sol#877-1369) uses literals with too many digits:
        - maxSoldAmount = 100000000000000000000000000 (CoinMama.sol#898)
CoinMama.slitherConstructorVariables() (CoinMama.sol#877-1369) uses literals with too many digits:
        - gasForProcessing = 300000 (CoinMama.sol#903)
CoinMamaDividendTracker.constructor() (CoinMama.sol#1391-1394) uses literals with too many digits:
        - minimumTokenBalanceForDividends = 300000 * (10 ** 18) (CoinMama.sol#1393)
CoinMamaDividendTracker.getAccountAtIndex(uint256) (CoinMama.sol#1476-1493) uses literals with too many digits:
        - (0x0000000000000000000000000000000000000000,-1,-1,0,0,0,0) (CoinMama.sol#1487)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
INFO:Detectors:
SafeMathInt.MAX_INT256 (CoinMama.sol#1725) is never used in SafeMathInt (CoinMama.sol#1723-1781)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variables
INFO:Detectors:
CoinMama.deadWallet (CoinMama.sol#884) should be constant
CoinMama.maxSoldAmount (CoinMama.sol#898) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
INFO:Detectors:
renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (CoinMama.sol#52-55)
previousOwner() should be declared external:
        - Ownable.previousOwner() (CoinMama.sol#57-59)
getUnlockTime() should be declared external:
        - Ownable.getUnlockTime() (CoinMama.sol#61-63)
transferOwnership(address,uint256) should be declared external:
        - Ownable.transferOwnership(address,uint256) (CoinMama.sol#69-75)
```

```
        - Ownable.getUnlockTime() (CoinMama.sol#61-63)
transferOwnership(address,uint256) should be declared external:
        - Ownable.transferOwnership(address,uint256) (CoinMama.sol#69-75)
name() should be declared external:
        - ERC20.name() (CoinMama.sol#406-408)
symbol() should be declared external:
        - ERC20.symbol() (CoinMama.sol#414-416)
decimals() should be declared external:
        - ERC20.decimals() (CoinMama.sol#431-433)
transfer(address,uint256) should be declared external:
        - ERC20.transfer(address,uint256) (CoinMama.sol#457-460)
allowance(address,address) should be declared external:
        - ERC20.allowance(address,address) (CoinMama.sol#465-467)
approve(address,uint256) should be declared external:
        - ERC20.approve(address,uint256) (CoinMama.sol#476-479)
transferFrom(address,address,uint256) should be declared external:
        - ERC20.transferFrom(address,address,uint256) (CoinMama.sol#494-502)
increaseAllowance(address,uint256) should be declared external:
        - ERC20.increaseAllowance(address,uint256) (CoinMama.sol#516-519)
decreaseAllowance(address,uint256) should be declared external:
        - ERC20.decreaseAllowance(address,uint256) (CoinMama.sol#535-538)
withdrawDividend() should be declared external:
        - CoinMamaDividendTracker.withdrawDividend() (CoinMama.sol#1400-1402)
        - DividendPayingToken.withdrawDividend() (CoinMama.sol#771-773)
dividendOf(address) should be declared external:
        - DividendPayingToken.dividendOf(address) (CoinMama.sol#799-801)
withdrawnDividendOf(address) should be declared external:
        - DividendPayingToken.withdrawnDividendOf(address) (CoinMama.sol#813-815)
updateDividendTracker(address) should be declared external:
        - CoinMama.updateDividendTracker(address) (CoinMama.sol#995-1011)
updateUniswapV2Router(address) should be declared external:
        - CoinMama.updateUniswapV2Router(address) (CoinMama.sol#1013-1017)
excludeMultipleAccountsFromFees(address[],bool) should be declared external:
        - CoinMama.excludeMultipleAccountsFromFees(address[],bool) (CoinMama.sol#1033-1039)
setAutomatedMarketMakerPair(address,bool) should be declared external:
        - CoinMama.setAutomatedMarketMakerPair(address,bool) (CoinMama.sol#1045-1049)
updateGasForProcessing(uint256) should be declared external:
        - CoinMama.updateGasForProcessing(uint256) (CoinMama.sol#1063-1068)
```

```
updateGasForProcessing(uint256) should be declared external:
        - CoinMama.updateGasForProcessing(uint256) (CoinMama.sol#1063-1068)
isExcludedFromFees(address) should be declared external:
        - CoinMama.isExcludedFromFees(address) (CoinMama.sol#1082-1084)
withdrawableDividendOf(address) should be declared external:
        - CoinMama.withdrawableDividendOf(address) (CoinMama.sol#1086-1088)
dividendTokenBalanceOf(address) should be declared external:
        - CoinMama.dividendTokenBalanceOf(address) (CoinMama.sol#1090-1092)
getAccountAtIndex(uint256) should be declared external:
        - CoinMamaDividendTracker.getAccountAtIndex(uint256) (CoinMama.sol#1476-1493)
process(uint256) should be declared external:
        - CoinMamaDividendTracker.process(uint256) (CoinMama.sol#1520-1565)
get(IterableMapping.Map,address) should be declared external:
        - IterableMapping.get(IterableMapping.Map,address) (CoinMama.sol#1800-1802)
getIndexOfKey(IterableMapping.Map,address) should be declared external:
        - IterableMapping.getIndexOfKey(IterableMapping.Map,address) (CoinMama.sol#1804-1809)
getKeyAtIndex(IterableMapping.Map,uint256) should be declared external:
        - IterableMapping.getKeyAtIndex(IterableMapping.Map,uint256) (CoinMama.sol#1811-1813)
size(IterableMapping.Map) should be declared external:
        - IterableMapping.size(IterableMapping.Map) (CoinMama.sol#1817-1819)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:CoinMama.sol analyzed (18 contracts with 75 detectors), 108 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

Results

No major issues were found. Some false positive errors were reported by the tools. All the other issues have been categorized above according to their level of severity.

SOLIDITY STATIC ANALYSIS

SOLIDITY STATIC ANALYSIS

CoinMama.sol

Security

Transaction origin:

INTERNAL ERROR in module Transaction origin: can't convert undefined to object
Pos: not available

Check-effects-interaction:

INTERNAL ERROR in module Check-effects-interaction: can't convert undefined to object
Pos: not available

Inline assembly:

INTERNAL ERROR in module Inline assembly: can't convert undefined to object
Pos: not available

Block timestamp:

INTERNAL ERROR in module Block timestamp: can't convert undefined to object
Pos: not available

Low level calls:

INTERNAL ERROR in module Low level calls: can't convert undefined to object
Pos: not available

Selfdestruct:

INTERNAL ERROR in module Selfdestruct: can't convert undefined to object
Pos: not available

Gas & Economy

This on local calls:

INTERNAL ERROR in module This on local calls: can't convert undefined to object
Pos: not available

Delete dynamic array:

INTERNAL ERROR in module Delete dynamic array: can't convert undefined to object
Pos: not available

For loop over dynamic array:

INTERNAL ERROR in module For loop over dynamic array: can't convert undefined to object
Pos: not available

Ether transfer in loop:

INTERNAL ERROR in module Ether transfer in loop: can't convert undefined to object
Pos: not available

ERC

ERC20:

INTERNAL ERROR in module ERC20: can't convert undefined to object
Pos: not available

Miscellaneous

Constant/View/Pure functions:

INTERNAL ERROR in module Constant/View/Pure functions: can't convert undefined to object
Pos: not available

Similar variable names:

INTERNAL ERROR in module Similar variable names: can't convert undefined to object
Pos: not available

No return:

INTERNAL ERROR in module No return: can't convert undefined to object
Pos: not available

Guard conditions:

INTERNAL ERROR in module Guard conditions: can't convert undefined to object
Pos: not available

String length:

INTERNAL ERROR in module String length: can't convert undefined to object
Pos: not available

SOLHINT LINTER

```
 contracts/CoinMama.sol:1:1: Error: Compiler version ^0.6.2 does not satisfy the r semver requirement

 contracts/CoinMama.sol:93:5: Error: Function name must be in mixedCase

 contracts/CoinMama.sol:94:5: Error: Function name must be in mixedCase

 contracts/CoinMama.sol:111:5: Error: Function name must be in mixedCase

 contracts/CoinMama.sol:148:5: Error: Function name must be in mixedCase

 contracts/CoinMama.sol:652:24: Error: Code contains empty blocks

 contracts/CoinMama.sol:714:29: Error: Constant name must be in capitalized SNAKE_CASE

 contracts/CoinMama.sol:734:88: Error: Code contains empty blocks

 contracts/CoinMama.sol:782:25: Error: Avoid using low level calls.

 contracts/CoinMama.sol:877:1: Error: Contract has 20 states declarations but allowed no more than 15

 contracts/CoinMama.sol:893:21: Error: Variable name must be in mixedCase

 contracts/CoinMama.sol:991:32: Error: Code contains empty blocks

 contracts/CoinMama.sol:1122:85: Error: Avoid to use tx.origin

 contracts/CoinMama.sol:1158:21: Error: Avoid to make time-based decisions in your business logic

 contracts/CoinMama.sol:1169:51: Error: Use double quotes for string literals

 contracts/CoinMama.sol:1177:45: Error: Avoid to make time-based decisions in your business logic

 contracts/CoinMama.sol:1207:34: Error: Avoid to make time-based decisions in your business logic

 contracts/CoinMama.sol:1211:28: Error: Avoid to make time-based decisions in your business logic

 contracts/CoinMama.sol:1246:72: Error: Code contains empty blocks
```

```
contracts/CoinMama.sol:1246:81: Error: Code contains empty blocks

contracts/CoinMama.sol:1247:68: Error: Code contains empty blocks

contracts/CoinMama.sol:1247:77: Error: Code contains empty blocks

contracts/CoinMama.sol:1253:89: Error: Avoid to use tx.origin

contracts/CoinMama.sol:1255:13: Error: Code contains empty blocks

contracts/CoinMama.sol:1338:13: Error: Avoid to make time-based decisions in your business logic

contracts/CoinMama.sol:1355:13: Error: Avoid to make time-based decisions in your business logic

contracts/CoinMama.sol:1363:27: Error: Avoid using low level calls.

contracts/CoinMama.sol:1471:58: Error: Avoid to make time-based decisions in your business logic

contracts/CoinMama.sol:1472:71: Error: Avoid to make time-based decisions in your business logic

contracts/CoinMama.sol:1496:25: Error: Avoid to make time-based decisions in your business logic

contracts/CoinMama.sol:1500:13: Error: Avoid to make time-based decisions in your business logic

contracts/CoinMama.sol:1571:33: Error: Avoid to make time-based decisions in your business logic
```

Disclaimer

Quillhash audit is not a security warranty, investment advice, or an endorsement of the CoinMama Token platform. This audit does not provide a security or correctness guarantee of the audited smart contracts. The statements made in this document should not be interpreted as investment or legal advice, nor should its authors be held accountable for decisions made based on them. Securing smart contracts is a multistep process. One audit cannot be considered enough. We recommend that the CoinMama Token Team put in place a bug bounty program to encourage further analysis of the smart contract by other third parties.

Closing Summary

Overall, smart contracts are very well written and adhere to guidelines.

No instances of Integer Overflow and Underflow vulnerabilities or Back-Door Entry were found in the contract, but relying on other contracts might cause Reentrancy Vulnerability.

The majority of the concerns addressed above have been acknowledged, implemented and verified.

CoinMama

QuillAudits

Canada, India, Singapore and United Kingdom

audits.quillhash.com

audits@quillhash.com