Cyberscope

# Audit Report

# Kouta Kun Inu

October 2022

| | |
|---|---|
| Type | BEP20 |
| Network | BSC |
| Address | 0x23B72DbF901B9B6794499bde0D58D4FED77EEc70 |
| Audited by | © cyberscope |

# Table of Contents

# Contract Review

| | |
|---|---|
| **Contract Name** | ProMax |
| **Compiler Version** | v0.8.15+commit.e14f2714 |
| **Optimization** | 200 runs |
| **Licence** | None |
| **Explorer** | https://bscscan.com/token/0x23B72DbF901B9B6794499bde0D58D4FED77EEc70 |
| **Symbol** | KKI |
| **Decimals** | 18 |
| **Total Supply** | 1,000,000,000 |
| **Domain** | https://kouta-kuninu.me |

# Source Files

| **Filename** | **SHA256** |
|---|---|
| **contract.sol** | 0bf51ce8700e2f3297159879ac31f9003d0765d215e5ff5673677c31a71ab9c2 |

# Audit Updates

| | |
|---|---|
| **Initial Audit** | 11th October 2022 |
| **Corrected** | |

# Contract Analysis

● Critical ● Medium ● Minor / Informative ● Pass

| Severity | Code | Description | Status |
|---|---|---|---|
| ● | ST | Stops Transactions | Passed |
| ● | OCTD | Transfers Contract's Tokens | Passed |
| ● | OTUT | Transfers User's Tokens | Passed |
| ● | ELFM | Exceeds Fees Limit | Unresolved |
| ● | ULTW | Transfers Liquidity to Team Wallet | Passed |
| ● | MT | Mints Tokens | Passed |
| ● | BT | Burns Tokens | Passed |
| ● | BC | Blacklists Addresses | Unresolved |

# ELFM - Exceeds Fees Limit

| | |
|---|---|
| **Criticality** | critical |
| **Location** | contract.sol#L1484 |
| **Status** | Unresolved |

## Description

The contract owner has the authority to increase over the allowed limit of 25%. The owner may take advantage of it by calling the setAllFeePercent function with the maximum transaction fee for each fee.

```
function setAllFeePercent(
        uint8 taxFee,
        uint8 liquidityFee,
        uint8 burnFee,
        uint8 walletFee,
        uint8 buybackFee,
        uint8 walletCharityFee,
        uint8 rewardFee
    ) external onlyOwner {
        require(taxFee >= 0 && taxFee <= maxTaxFee, "TF err");
        require(liquidityFee >= 0 && liquidityFee <= maxLiqFee, "LF err");
        require(burnFee >= 0 && burnFee <= maxBurnFee, "BF err");
        require(walletFee >= 0 && walletFee <= maxWalletFee, "WF err");
        require(buybackFee >= 0 && buybackFee <= maxBuybackFee, "BBF err");
        require(
            walletCharityFee >= 0 && walletCharityFee <= maxWalletFee,
            "WFT err"
        );
        require(rewardFee >= 0 && rewardFee <= maxTaxFee, "RF err");
        //both tax fee and reward fee cannot be set
        require(rewardFee == 0 || taxFee == 0, "RT fee err");
        _taxFee = taxFee;
        _liquidityFee = liquidityFee;
        _burnFee = burnFee;
        _buybackFee = buybackFee;
        _walletFee = walletFee;
        _walletCharityFee = walletCharityFee;
        _rewardFee = rewardFee;
    }
```

## Recommendation

The contract could embody a check for the total maximum acceptable value.

The team should carefully manage the private keys of the owner’s account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions. That risk can be prevented by temporarily locking the contract or renouncing ownership.

# BC - Blacklists Addresses

| | |
|---|---|
| **Criticality** | medium |
| **Location** | contract.sol#L2536 |
| **Status** | Unresolved |

## Description

The contract owner has the authority to stop addresses from transactions. The owner may take advantage of it by calling the blacklistAddress function.

```
function blacklistAddress(address account, bool value) external onlyOwner {
        _isBlacklisted[account] = value;
    }
```

## Recommendation

The team should carefully manage the private keys of the owner’s account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions. That risk can be prevented by temporarily locking the contract or renouncing ownership.

# Contract Diagnostics

● Critical ● Medium ● Minor / Informative

| Severity | Code | Description | Status |
|---|---|---|---|
| ● | STC | Succeeded Transfer Check | Unresolved |
| ● | MC | Missing Check | Unresolved |
| ● | L01 | Public Function could be Declared External | Unresolved |
| ● | L02 | State Variables could be Declared Constant | Unresolved |
| ● | L03 | Redundant Statements | Unresolved |
| ● | L04 | Conformance to Solidity Naming Conventions | Unresolved |
| ● | L05 | Unused State Variable | Unresolved |
| ● | L07 | Missing Events Arithmetic | Unresolved |
| ● | L08 | Tautology or Contradiction | Unresolved |
| ● | L09 | Dead Code Elimination | Unresolved |
| ● | L13 | Divide before Multiply Operation | Unresolved |
| ● | L15 | Local Scope Variable Shadowing | Unresolved |

# STC - Succeeded Transfer Check

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | contract.sol#L2178 |
| **Status** | Unresolved |

## Description

According to the ERC20 specification, the transfer methods should be checked if the result is successful. Otherwise, the contract may wrongly assume that the transfer has been established.

```
IERC20(tokenAddress).transfer(owner(), tokenAmount);
```

## Recommendation

The contract should check if the result of the transfer methods is successful.

# MC - Missing Check

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | contract.sol#L1234 |
| **Status** | Unresolved |

## Description

The contract is processing variables that have not properly sanitized and checked that they form the proper shape. These variables may produce vulnerability issues.

The contract does not sanitize the setMxTxPer and setMxWalletPer variables on the constructor. These variables can lead to issues with transfer transactions.

```
 _maxTxAmount = _tTotal.mul(setMxTxPer).div(10**4);
_maxWalletAmount = _tTotal.mul(setMxWalletPer).div(10**4);
```

## Recommendation

The contract should properly check the variables according to the required specifications. It is recommended to check for acceptable percentages.

- setMxTxPer could be greater than 1% an lower than or equal of total supply
- setMxWalletPer could be greater than 1% and lower or equal of total supply

# L01 - Public Function could be Declared External

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | contract.sol#L2228,2387,1386,1422,768,2508,1737,1292,2167,1410,1338,1447,1510,1535,2240,1373,1476,777,786,1356,2302,2191,1347,1296,1316,799,1472,1312,1402,1329,791,1406 |
| **Status** | Unresolved |

## Description

Public functions that are never called by the contract should be declared external to save gas.

```
dividendOf
getAccountDividendsInfoAtIndex
decreaseAllowance
reflectionFromToken
renounceOwnership
updateGasForProcessing
isExcludedFromFee
name
recoverBEP20
...
```

## Recommendation

Use the external attribute for functions never called from the contract.

# L02 - State Variables could be Declared Constant

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | contract.sol#L1062,1068,1126,1065,1064,1063,1060,1067,1066 |
| **Status** | Unresolved |

## Description

Constant state variables should be declared constant to save gas.

```
maxLiqFee
minMxWalletPercentage
mintedByMoonDeploy
maxWalletFee
maxBurnFee
maxTaxFee
dead
minMxTxPercentage
maxBuybackFee
```

## Recommendation

Add the constant attribute to state variables that never change.

# L03 - Redundant Statements

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | contract.sol#L255 |
| **Status** | Unresolved |

## Description

The contract contains statements that are not used and have no effect. As a result, those segments increase the code size of the contract unnecessarily.

```
Context
```

## Recommendation

Remove the redundant statements in order to decrease the code size.

# L04 - Conformance to Solidity Naming Conventions

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | contract.sol#L2244,1128,1150,1144,1122,1075,1169,1562,2240,1135,1165,1681,1677,1164,1076,1147,1132,2228,2338,1138,1129,847,1535,1141,2232 |
| **Status** | Unresolved |

## Description

Solidity defines a naming convention that should be followed. Rule exceptions:

- Allow constant variable name/symbol/decimals to be lowercase.
- Allow _ at the beginning of the mixed_case match for private variables and unused parameters.

```
_owner
_name
_buybackFee
_walletFee
_tTotal
_tDividendTotal
_isBlacklisted
_minimumTokenBalanceForDividends
_rewardFee
...
```

## Recommendation

Follow the Solidity naming convention.
https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions.

# L05 - Unused State Variable

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | contract.sol#L272 |
| **Status** | Unresolved |

## Description

There are segments that contain unused state variables.

```
MAX_INT256
```

## Recommendation

Remove unused state variables.

# L07 - Missing Events Arithmetic

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | contract.sol#L1561,1514,1518,2508,1526,1480 |
| **Status** | Unresolved |

## Description

Detected missing events for critical arithmetic parameters. There are functions that have no event emitted, so it is difficult to track off-chain changes.

```
minimumTokenBalanceForDividends = _minimumTokenBalanceForDividends
buyBackUpperLimit = buyBackLimit * 10 ** uint256(_decimals)
_maxTxAmount = _tTotal.mul(maxTxPercent).div(10 ** 4)
gasForProcessing = newValue
_maxWalletAmount = _tTotal.mul(maxWalletPercent).div(10 ** 4)
_taxFee = taxFee
```

## Recommendation

Emit an event for critical parameter changes.

# L08 - Tautology or Contradiction

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | contract.sol#L1480,1201 |
| **Status** | Unresolved |

## Description

Detects expressions that are tautologies or contradictions. For instance, an uint variable will always be greater than or equal to zero.

```
require(bool,string)(rewardFee >= 0 && rewardFee <= maxTaxFee,RF err)
require(bool,string)(fee.setBurnFee >= 0 && fee.setBurnFee <= maxBurnFee,BF err)
require(bool,string)(fee.setRewardFee >= 0 && fee.setRewardFee <= maxTaxFee,RF err)
require(bool,string)(walletCharityFee >= 0 && walletCharityFee <= maxWalletFee,WFT err)
require(bool,string)(taxFee >= 0 && taxFee <= maxTaxFee,TF err)
require(bool,string)(buybackFee >= 0 && buybackFee <= maxBuybackFee,BBF err)
require(bool,string)(fee.setBuybackFee >= 0 && fee.setBuybackFee <= maxBuybackFee,BBF err)
require(bool,string)(fee.setLiqFee >= 0 && fee.setLiqFee <= maxLiqFee,LF err)
require(bool,string)(burnFee >= 0 && burnFee <= maxBurnFee,BF err)
...
```

## Recommendation

Fix the incorrect comparison by changing the value type or the comparison.

# L09 - Dead Code Elimination

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | contract.sol#L462,526,318,695,603,652,2257,350,633,614,546,559,670,433,507,494 |
| **Status** | Unresolved |

## Description

Functions that are not used in the contract, and make the code's size bigger.

```
sendValue
functionCallWithValue
abs
_callOptionalReturn
safeTransfer
safeIncreaseAllowance
_dtransfer
get
safeApprove
...
```

## Recommendation

Remove unused functions.

# L13 - Divide before Multiply Operation

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | contract.sol#L1850 |
| **Status** | Unresolved |

## Description

Performing divisions before multiplications may cause lose of prediction.

```
spentAmount = contractTokenBalance.div(totFee).mul(_rewardFee)
spentAmount = contractTokenBalance.div(totFee).mul(_walletCharityFee)
spentAmount = contractTokenBalance.div(totFee).mul(_buybackFee)
spentAmount = contractTokenBalance.div(totFee).mul(_walletFee)
spentAmount = contractTokenBalance.div(totFee).mul(_burnFee)
```

## Recommendation

The multiplications should be prior to the divisions.

# L15 - Local Scope Variable Shadowing

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | contract.sol#L2228,2244,2232,2240 |
| **Status** | Unresolved |

## Description

The are variables that are defined in the local scope containing the same name from an upper scope.

```
_owner
```

## Recommendation

The local variables should have different names from the upper scoped variables.

# Contract Functions

| Contract | Type | Bases | | |
|---|---|---|---|---|
| | **Function Name** | **Visibility** | **Mutability** | **Modifiers** |
| | | | | |
| **IERC20** | Interface | | | |
| | totalSupply | External | | - |
| | balanceOf | External | | - |
| | transfer | External | ✓ | - |
| | allowance | External | | - |
| | approve | External | ✓ | - |
| | transferFrom | External | ✓ | - |
| | | | | |
| **SafeMath** | Library | | | |
| | add | Internal | | |
| | sub | Internal | | |
| | sub | Internal | | |
| | mul | Internal | | |
| | div | Internal | | |
| | div | Internal | | |
| | mod | Internal | | |
| | mod | Internal | | |
| | | | | |
| **Context** | Implementation | | | |
| | _msgSender | Internal | | |
| | _msgData | Internal | | |
| | | | | |
| **SafeMathInt** | Library | | | |
| | mul | Internal | | |
| | div | Internal | | |
| | sub | Internal | | |
| | add | Internal | | |

| | | | | |
|---|---|---|---|---|
| | abs | Internal | | |
| | toUint256Safe | Internal | | |
| | | | | |
| **SafeMathUint** | Library | | | |
| | toInt256Safe | Internal | | |
| | | | | |
| **IterableMapping** | Library | | | |
| | get | Internal | | |
| | getIndexOfKey | Internal | | |
| | getKeyAtIndex | Internal | | |
| | size | Internal | | |
| | set | Internal | ✓ | |
| | remove | Internal | ✓ | |
| | | | | |
| **Address** | Library | | | |
| | isContract | Internal | | |
| | sendValue | Internal | ✓ | |
| | functionCall | Internal | ✓ | |
| | functionCall | Internal | ✓ | |
| | functionCallWithValue | Internal | ✓ | |
| | functionCallWithValue | Internal | ✓ | |
| | _functionCallWithValue | Private | ✓ | |
| | | | | |
| **SafeERC20** | Library | | | |
| | safeTransfer | Internal | ✓ | |
| | safeTransferFrom | Internal | ✓ | |
| | safeApprove | Internal | ✓ | |
| | safeIncreaseAllowance | Internal | ✓ | |
| | safeDecreaseAllowance | Internal | ✓ | |
| | _callOptionalReturn | Private | ✓ | |
| | | | | |
| **Ownable** | Implementation | Context | | |
| | <Constructor> | Public | ✓ | - |
| | owner | Public | | - |

| | | | | |
|---|---|---|---|---|
| | renounceOwnership | Public | ✓ | onlyOwner |
| | transferOwnership | Public | ✓ | onlyOwner |
| | geUnlockTime | Public | | - |
| | lock | Public | ✓ | onlyOwner |
| | unlock | Public | ✓ | - |
| | | | | |
| **IUniswapV2Factory** | Interface | | | |
| | feeTo | External | | - |
| | feeToSetter | External | | - |
| | getPair | External | | - |
| | allPairs | External | | - |
| | allPairsLength | External | | - |
| | createPair | External | ✓ | - |
| | setFeeTo | External | ✓ | - |
| | setFeeToSetter | External | ✓ | - |
| | | | | |
| **IUniswapV2Router01** | Interface | | | |
| | factory | External | | - |
| | WETH | External | | - |
| | addLiquidity | External | ✓ | - |
| | addLiquidityETH | External | Payable | - |
| | removeLiquidity | External | ✓ | - |
| | removeLiquidityETH | External | ✓ | - |
| | removeLiquidityWithPermit | External | ✓ | - |
| | removeLiquidityETHWithPermit | External | ✓ | - |
| | swapExactTokensForTokens | External | ✓ | - |
| | swapTokensForExactTokens | External | ✓ | - |
| | swapExactETHForTokens | External | Payable | - |
| | swapTokensForExactETH | External | ✓ | - |
| | swapExactTokensForETH | External | ✓ | - |
| | swapETHForExactTokens | External | Payable | - |
| | quote | External | | - |
| | getAmountOut | External | | - |

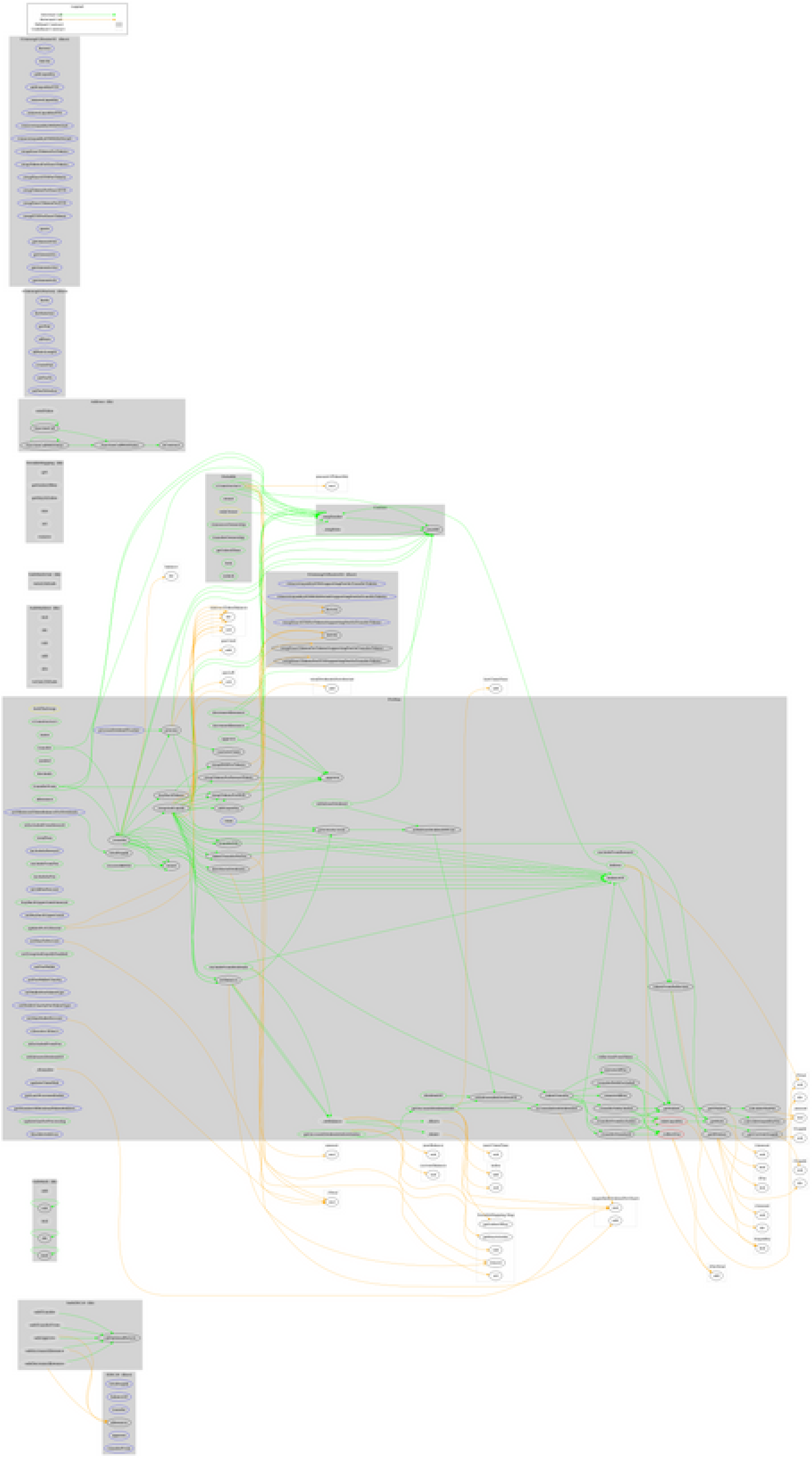
| | | | | |
|---|---|---|---|---|
| | getAmountIn | External | | - |
| | getAmountsOut | External | | - |
| | getAmountsIn | External | | - |
| | | | | |
| **IUniswapV2Router02** | Interface | IUniswapV2Router01 | | |
| | removeLiquidityETHSupportingFeeOnTransferTokens | External | ✓ | - |
| | removeLiquidityETHWithPermitSupportingFeeOnTransferTokens | External | ✓ | - |
| | swapExactTokensForTokensSupportingFeeOnTransferTokens | External | ✓ | - |
| | swapExactETHForTokensSupportingFeeOnTransferTokens | External | Payable | - |
| | swapExactTokensForETHSupportingFeeOnTransferTokens | External | ✓ | - |
| | | | | |
| **ProMax** | Implementation | Context, IERC20, Ownable | | |
| | <Constructor> | Public | Payable | - |
| | name | Public | | - |
| | updatePcsV2Router | Public | ✓ | onlyOwner |
| | symbol | Public | | - |
| | decimals | Public | | - |
| | totalSupply | Public | | - |
| | balanceOf | Public | | - |
| | transfer | Public | ✓ | - |
| | allowance | Public | | - |
| | approve | Public | ✓ | - |
| | transferFrom | Public | ✓ | - |
| | increaseAllowance | Public | ✓ | - |
| | decreaseAllowance | Public | ✓ | - |
| | isExcludedFromReward | Public | | - |
| | totalFees | Public | | - |
| | deliver | Public | ✓ | - |
| | reflectionFromToken | Public | | - |
| | tokenFromReflection | Public | | - |

| | | | | |
|---|---|---|---|---|
| | excludeFromReward | Public | ✓ | onlyOwner |
| | includeInReward | External | ✓ | onlyOwner |
| | excludeFromFee | Public | ✓ | onlyOwner |
| | includeInFee | Public | ✓ | onlyOwner |
| | setAllFeePercent | External | ✓ | onlyOwner |
| | buyBackUpperLimitAmount | Public | | - |
| | setBuybackUpperLimit | External | ✓ | onlyOwner |
| | setMaxTxPercent | External | ✓ | onlyOwner |
| | setMaxWalletPercent | External | ✓ | onlyOwner |
| | setSwapAndLiquifyEnabled | Public | ✓ | onlyOwner |
| | setFeeWallet | External | ✓ | onlyOwner |
| | setFeeWalletCharity | External | ✓ | onlyOwner |
| | setWalletFeeTokenType | External | ✓ | onlyOwner |
| | setWalletCharityFeeTokenType | External | ✓ | onlyOwner |
| | setMinimumTokenBalanceForDividends | External | ✓ | onlyOwner |
| | <Receive Ether> | External | Payable | - |
| | _reflectFee | Private | ✓ | |
| | _getValues | Private | | |
| | _getTValues | Private | | |
| | _getRValues | Private | | |
| | _getRate | Private | | |
| | _getCurrentSupply | Private | | |
| | _takeLiquidity | Private | ✓ | |
| | calculateTaxFee | Private | | |
| | calculateLiquidityFee | Private | | |
| | removeAllFee | Private | ✓ | |
| | restoreAllFee | Private | ✓ | |
| | isExcludedFromFee | Public | | - |
| | _approve | Private | ✓ | |
| | _transfer | Private | ✓ | |
| | swapAndLiquify | Private | ✓ | lockTheSwap |
| | buyBackTokens | Private | ✓ | lockTheSwap |
| | swapTokensForBNB | Private | ✓ | |
| | swapBNBForTokens | Private | ✓ | |

| | | | | |
|---|---|---|---|---|
| | swapTokensForRewardToken | Private | ✓ | |
| | addLiquidity | Private | ✓ | |
| | _tokenTransfer | Private | ✓ | |
| | _transferStandard | Private | ✓ | |
| | _transferToExcluded | Private | ✓ | |
| | _transferFromExcluded | Private | ✓ | |
| | _transferBothExcluded | Private | ✓ | |
| | _tokenTransferNoFee | Private | ✓ | |
| | transferEth | Private | ✓ | |
| | recoverBEP20 | Public | ✓ | onlyOwner |
| | distributeDividends | Internal | ✓ | |
| | withdrawDividend | Public | ✓ | - |
| | _withdrawDividendOfUser | Internal | ✓ | |
| | dividendOf | Public | | - |
| | withdrawableDividendOf | Public | | - |
| | withdrawnDividendOf | Public | | - |
| | accumulativeDividendOf | Public | | - |
| | _dtransfer | Internal | ✓ | |
| | _dmint | Internal | ✓ | |
| | _dburn | Internal | ✓ | |
| | _setBalance | Internal | ✓ | |
| | excludeFromDividends | Public | ✓ | onlyOwner |
| | updateClaimWait | External | ✓ | onlyOwner |
| | getLastProcessedIndex | External | | - |
| | getNumberOfDividendTokenHolders | External | | - |
| | getAccountDividendsInfo | Public | | - |
| | getAccountDividendsInfoAtIndex | Public | | - |
| | canAutoClaim | Private | | |
| | setBalance | Private | ✓ | |
| | process | Public | ✓ | - |
| | processAccount | Internal | ✓ | |
| | updateGasForProcessing | Public | ✓ | onlyOwner |
| | processDividendTracker | External | ✓ | - |
| | blacklistAddress | External | ✓ | onlyOwner |

| | | | | |
|---|---|---|---|---|
| | claim | External | ✓ | - |

# Contract Flow



# Domain Info

| | |
|---|---|
| **Domain Name** | kouta-kuninu.me |
| **Registry Domain ID** | D425500000340043206-AGRS |
| **Creation Date** | 2022-09-15T18:42:31Z |
| **Updated Date** | 2022-09-16T11:08:23Z |
| **Registry Expiry Date** | 2023-09-15T18:42:31Z |
| **Registrar WHOIS Server** | whois.namecheap.com |
| **Registrar URL** | www.namecheap.com |
| **Registrar** | NameCheap, Inc. |
| **Registrar IANA ID** | 1068 |

The domain was created 26 days before the creation of the audit. It will expire in 11 months.

There is no public billing information, the creator is protected by the privacy settings.

# Summary

There are some functions that can be abused by the owner like manipulating fees, transferring funds to the team's wallet and blacklisting addresses. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats.

# Disclaimer

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment.

Cyberscope team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed.

The Cyberscope team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Cyberscope receive a payment to manipulate those results or change the awarding badge that we will be adding in our website.

Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token.

The Cyberscope team disclaims any liability for the resulting losses.

# About Cyberscope

Coinscope audit and K.Y.C. service has been rebranded to Cyberscope.

Coinscope is the leading early coin listing, voting and auditing authority firm. The audit process is analyzing and monitoring many aspects of the project. That way, it gives the community a good sense of security using an informative report and a generic score.

Cyberscope and Coinscope are aiming to make crypto discoverable and efficient globally. They provide all the essential tools to assist users draw their own conclusions.

The Cyberscope team

https://www.cyberscope.io