



Cyberscope

Audit Report

Munkai

July 2022

Type BEP20

Network BSC

Address 0x8562b410498635930fef78f2ca7e7f067d1ca170

Audited by © cyberscope

Table of Contents

| | |
|---|-----------|
| Table of Contents | 1 |
| Contract Review | 3 |
| Source Files | 3 |
| Audit Updates | 3 |
| Contract Analysis | 4 |
| ST - Stop Transactions | 5 |
| Description | 5 |
| Recommendation | 5 |
| ELFM - Exceed Limit Fees Manipulation | 6 |
| Description | 6 |
| Recommendation | 6 |
| Contract Diagnostics | 7 |
| L01 - Public Function could be Declared External | 8 |
| Description | 8 |
| Recommendation | 8 |
| L02 - State Variables could be Declared Constant | 8 |
| Description | 8 |
| Recommendation | 9 |
| L04 - Conformance to Solidity Naming Conventions | 9 |
| Description | 9 |
| Recommendation | 9 |
| L05 - Unused State Variable | 10 |
| Description | 10 |
| Recommendation | 10 |
| L07 - Missing Events Arithmetic | 10 |
| Description | 10 |

| | |
|---|-----------|
| Recommendation | 11 |
| L09 - Dead Code Elimination | 11 |
| Description | 11 |
| Recommendation | 11 |
| L11 - Unnecessary Boolean equality | 11 |
| Description | 11 |
| Recommendation | 12 |
| Contract Functions | 13 |
| Contract Flow | 17 |
| Domain Info | 18 |
| Summary | 19 |
| Disclaimer | 20 |
| About Cyberscope | 21 |

Contract Review

| | |
|-------------------------|---|
| Contract Name | Munkai |
| Compiler Version | v0.8.0+commit.c7dfd78e |
| Optimization | 200 runs |
| Licence | MIT |
| Explorer | https://bscscan.com/token/0x8562b410498635930fef78f2ca7e7f067d1ca170 |
| Symbol | \$MUNK |
| Decimals | 9 |
| Total Supply | 60,000,000 |
| Domain | munkai.art |

Source Files

| | |
|---------------------|--|
| Filename | SHA256 |
| contract.sol | c5a3208d5f65e81203ee64befbcd589aef7326a339ee94217df17045886d03c8 |

Audit Updates

| | |
|----------------------|---------------|
| Initial Audit | 8th July 2022 |
| Corrected | |

Contract Analysis

● Critical ● Medium ● Minor ● Pass

| Severity | Code | Description |
|----------|------|---|
| ● | ST | Contract Owner is not able to stop or pause transactions |
| ● | OCTD | Contract Owner is not able to transfer tokens from specific address |
| ● | OTUT | Owner Transfer User's Tokens |
| ● | ELFM | Contract Owner is not able to increase fees more than a reasonable percent (25%) |
| ● | ULTW | Contract Owner is not able to increase the amount of liquidity taken by dev wallet more than a reasonable percent |
| ● | MT | Contract Owner is not able to mint new tokens |
| ● | BT | Contract Owner is not able to burn tokens from specific wallet |
| ● | BC | Contract Owner is not able to blacklist wallets from selling |

ST - Stop Transactions

| | |
|-------------|-----------------------|
| Criticality | critical |
| Location | contract.sol#L612,585 |

Description

The contract owner has the authority to stop the sales for all users excluding the owner. The owner may take advantage of it by setting the `totalFee` to zero. As a result the expression will divide to zero and the transaction will revert.

```
uint256 amountToLiquify =  
swapThreshold.mul(dynamicLiquidityFee).div(totalFee).div(2);
```

Additionally, the contract owner has the authority to stop the sales for all users excluding the owner for one day after the initial launch. The owner may take advantage of it by setting the `feeDenominator` to a value greater than 18000. As a result, the tokens will be more than the sender's balance and the transactions will revert.

```
if (launchedAtTimestamp + 1 days > block.timestamp) {  
    return totalFee.mul(18000).div(feeDenominator);  
}
```

Recommendation

The contract should embody a check for preventing the execution of the `swapBack()` when the `totalFee` is zero.

Read more in the [fees manipulation section](#).

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.

ELFM - Exceed Limit Fees Manipulation

| | |
|-------------|-------------------|
| Criticality | critical |
| Location | contract.sol#L743 |

Description

The contract owner has the authority to increase over the allowed limit of 25% for one day after the initial launch. The owner may take advantage of it by setting the `feeDenominator` to a value greater than **18000**.

```
function setFees(uint256 _liquidityFee, uint256 _buybackFee, uint256
_reflectionFee, uint256 _marketingFee, uint256 _feeDenominator) external
authorized {
    liquidityFee = _liquidityFee;
    buybackFee = _buybackFee;
    reflectionFee = _reflectionFee;
    marketingFee = _marketingFee;
    totalFee =
    _liquidityFee.add(_buybackFee).add(_reflectionFee).add(_marketingFee);
    feeDenominator = _feeDenominator;
    require(totalFee < feeDenominator/4);
}
```

Recommendation

The contract could embody a check for the maximum acceptable value of `feeDenominator`.

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

| Severity | Code | Description |
|----------|------|--|
| ● | L01 | Public Function could be Declared External |
| ● | L02 | State Variables could be Declared Constant |
| ● | L04 | Conformance to Solidity Naming Conventions |
| ● | L05 | Unused State Variable |
| ● | L07 | Missing Events Arithmetic |
| ● | L09 | Dead Code Elimination |
| ● | L11 | Unnecessary Boolean equality |

L01 - Public Function could be Declared External

| | |
|--------------------|-------------------------------|
| Criticality | minor |
| Location | contract.sol#L148,155,176,714 |

Description

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

```
launch  
transferOwnership  
unauthorize  
authorize
```

Recommendation

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

L02 - State Variables could be Declared Constant

| | |
|--------------------|---|
| Criticality | minor |
| Location | contract.sol#L256,269,414,416,418,417,424 |

Description

Constant state variables should be declared constant to save gas.

```
_totalSupply  
ZERO  
DEAD_NON_CHECKSUM  
DEAD  
BUSD  
dividendsPerShareAccuracyFactor
```

WBNB

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality

minor

Location

contract.sol#L191,294,247,255,256,694,743,753,758,763,768,414,415,416,417,418,420,421,422,424,425,427,428

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.

```
_allowances
_balances
_maxTxAmount
_totalSupply
_decimals
_symbol
_name
DEAD_NON_CHECKSUM
ZERO
...
```

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 |
| Location | contract.sol#L414,418 |

Description

There are segments that contain unused state variables.

```
DEAD_NON_CHECKSUM  
BUSD
```

Recommendation

Remove unused state variables.

L07 - Missing Events Arithmetic

| | |
|--------------------|---|
| Criticality | minor |
| Location | contract.sol#L294,694,703,720,743,758,763 |

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.

```
targetLiquidity = _target  
swapThreshold = _amount  
liquidityFee = _liquidityFee  
_maxTxAmount = amount  
buybackMultiplierNumerator = numerator  
autoBuybackCap = _cap  
minPeriod = _minPeriod
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

| | |
|--------------------|-------------------|
| Criticality | minor |
| Location | contract.sol#L710 |

Description

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

```
launched
```

Recommendation

Remove unused functions.

L11 - Unnecessary Boolean equality

| | |
|--------------------|-------------------|
| Criticality | minor |
| Location | contract.sol#L509 |

Description

The comparison to boolean constants is redundant. Boolean constants can be used directly and do not need to be compared to true or false.

```
require(bool,string)(buyBacker[msg.sender] == true,)
```

Recommendation

Remove the equality to the boolean constant.

Contract Functions

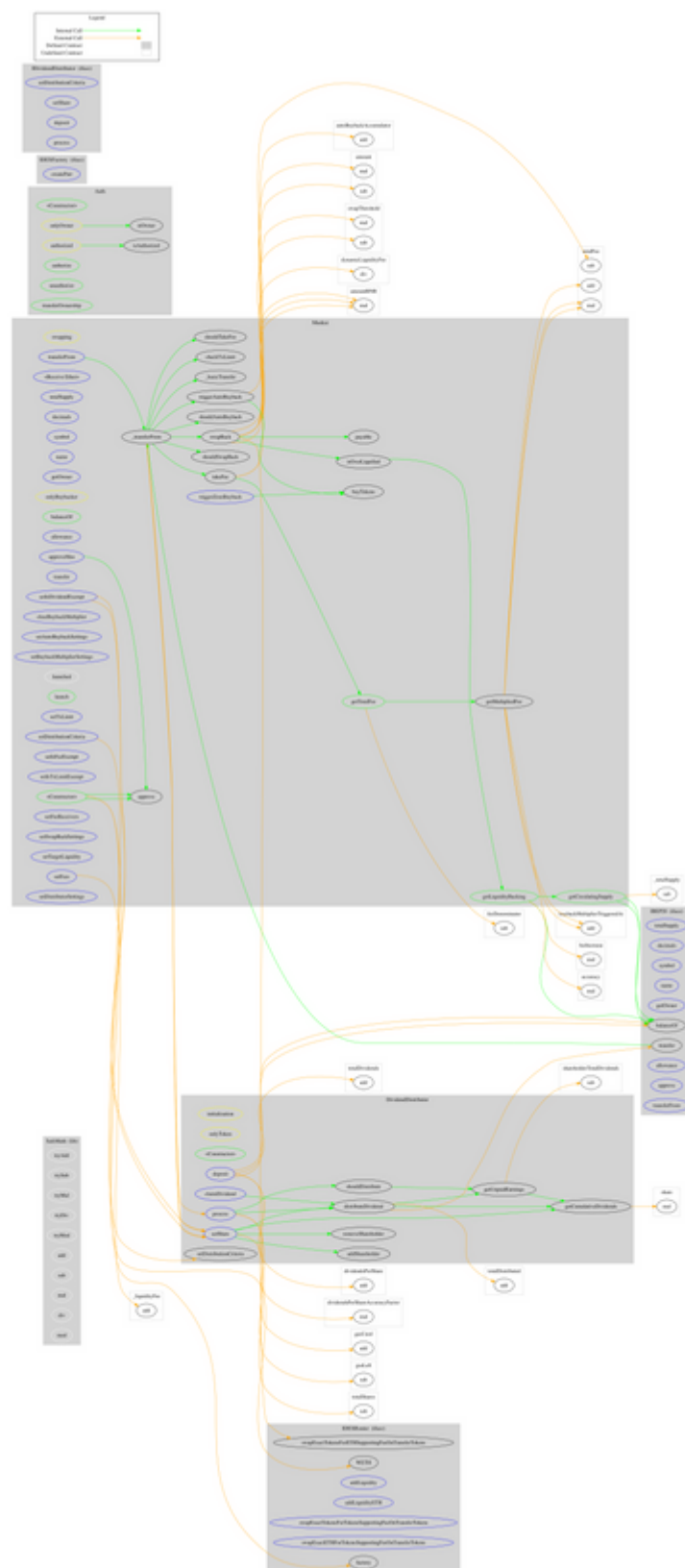
| Contract | Type | Bases | | |
|-----------------|----------------|------------|------------|-----------|
| | Function Name | Visibility | Mutability | Modifiers |
| | | | | |
| SafeMath | Library | | | |
| | tryAdd | Internal | | |
| | trySub | Internal | | |
| | tryMul | Internal | | |
| | tryDiv | Internal | | |
| | tryMod | Internal | | |
| | add | Internal | | |
| | sub | Internal | | |
| | mul | Internal | | |
| | div | Internal | | |
| | mod | Internal | | |
| | sub | Internal | | |
| | div | Internal | | |
| | mod | Internal | | |
| | | | | |
| IBEP20 | Interface | | | |
| | totalSupply | External | | - |
| | decimals | External | | - |
| | symbol | External | | - |
| | name | External | | - |
| | getOwner | External | | - |
| | balanceOf | External | | - |
| | transfer | External | ✓ | - |
| | allowance | External | | - |
| | approve | External | ✓ | - |
| | transferFrom | External | ✓ | - |
| | | | | |
| Auth | Implementation | | | |
| | <Constructor> | Public | ✓ | - |

| | | | | |
|-----------------------------|---|----------------------|---------|-----------|
| | authorize | Public | ✓ | onlyOwner |
| | unauthorize | Public | ✓ | onlyOwner |
| | isOwner | Public | | - |
| | isAuthorized | Public | | - |
| | transferOwnership | Public | ✓ | onlyOwner |
| | | | | |
| IDEXFactory | Interface | | | |
| | createPair | External | ✓ | - |
| | | | | |
| IDEXRouter | Interface | | | |
| | factory | External | | - |
| | WETH | External | | - |
| | addLiquidity | External | ✓ | - |
| | addLiquidityETH | External | Payable | - |
| | swapExactTokensForTokensSupportingFeeOnTransferTokens | External | ✓ | - |
| | swapExactETHForTokensSupportingFeeOnTransferTokens | External | Payable | - |
| | swapExactTokensForETHSupportingFeeOnTransferTokens | External | ✓ | - |
| | | | | |
| IDividendDistributor | Interface | | | |
| | setDistributionCriteria | External | ✓ | - |
| | setShare | External | ✓ | - |
| | deposit | External | Payable | - |
| | process | External | ✓ | - |
| | | | | |
| DividendDistributor | Implementation | IDividendDistributor | | |
| | <Constructor> | Public | ✓ | - |
| | setDistributionCriteria | External | ✓ | onlyToken |
| | setShare | External | ✓ | onlyToken |
| | deposit | External | Payable | onlyToken |
| | process | External | ✓ | onlyToken |
| | shouldDistribute | Internal | | |
| | distributeDividend | Internal | ✓ | |

| | | | | |
|---------------|------------------------|-----------------|---------|------------|
| | claimDividend | External | ✓ | - |
| | getUnpaidEarnings | Public | | - |
| | getCumulativeDividends | Internal | | |
| | addShareholder | Internal | ✓ | |
| | removeShareholder | Internal | ✓ | |
| | | | | |
| Munkai | Implementation | IBEP20, Auth | | |
| | <Constructor> | Public | ✓ | Auth |
| | <Receive Ether> | External | Payable | - |
| | totalSupply | External | | - |
| | decimals | External | | - |
| | symbol | External | | - |
| | name | External | | - |
| | getOwner | External | | - |
| | balanceOf | Public | | - |
| | allowance | External | | - |
| | approve | Public | ✓ | - |
| | approveMax | External | ✓ | - |
| | transfer | External | ✓ | - |
| | transferFrom | External | ✓ | - |
| | _transferFrom | Internal | ✓ | |
| | _basicTransfer | Internal | ✓ | |
| | checkTxLimit | Internal | | |
| | shouldTakeFee | Internal | | |
| | getTotalFee | Public | | - |
| | getMultipliedFee | Public | | - |
| | takeFee | Internal | ✓ | |
| | shouldSwapBack | Internal | | |
| | swapBack | Internal | ✓ | swapping |
| | shouldAutoBuyback | Internal | | |
| | triggerZeusBuyback | External | ✓ | authorized |
| | clearBuybackMultiplier | External | ✓ | authorized |
| | triggerAutoBuyback | Internal | ✓ | |
| | buyTokens | Internal | ✓ | swapping |
| | setAutoBuybackSettings | External | ✓ | authorized |

| | | | | |
|--|------------------------------|----------|---|------------|
| | setBuybackMultiplierSettings | External | ✓ | authorized |
| | launched | Internal | | |
| | launch | Public | ✓ | authorized |
| | setTxLimit | External | ✓ | authorized |
| | setIsDividendExempt | External | ✓ | authorized |
| | setIsFeeExempt | External | ✓ | authorized |
| | setIsTxLimitExempt | External | ✓ | authorized |
| | setFees | External | ✓ | authorized |
| | setFeeReceivers | External | ✓ | authorized |
| | setSwapBackSettings | External | ✓ | authorized |
| | setTargetLiquidity | External | ✓ | authorized |
| | setDistributionCriteria | External | ✓ | authorized |
| | setDistributorSettings | External | ✓ | authorized |
| | getCirculatingSupply | Public | | - |
| | getLiquidityBacking | Public | | - |
| | isOverLiquified | Public | | - |

Contract Flow



Domain Info

| | |
|-------------------------------|---|
| Domain Name | munkai.art |
| Registry Domain ID | D280013267-CNIC |
| Creation Date | 2022-03-05T16:58:17+00:00 |
| Updated Date | 2022-06-04T09:47:34+00:00 |
| Registry Expiry Date | 2024-03-05T23:59:59+00:00 |
| Registrar WHOIS Server | whois.godaddy.com |
| Registrar URL | https://www.godaddy.com/ |
| Registrar | Go Daddy, LLC |
| Registrar IANA ID | 146 |

The domain has been created in over 1 year before the creation of the audit.

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 stopping transactions and manipulating fees. The contract can be converted into a honeypot and prevent users from selling if the owner abuses the admin functions. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats. There is also a limit of max 25% fees for the trades one day after the launch.

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>