Cyberscope

# Audit Report

# StackUp

June 2022

| | |
|---|---|
| Type | BEP20 |
| Network | BSC |
| Address | 0xbf085b601a4b4d7104ae1d4abdd07099e72c3460 |
| 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 7
Description 7
Recommendation 7
BC - Blacklisted Contracts 9
Description 9
Recommendation 9
Contract Diagnostics 10
MTS - Manipulate Total Supply 11
Description 11
Recommendation 11
L01 - Public Function could be Declared External 12
Description 12
Recommendation 12
L02 - State Variables could be Declared Constant 13
Description 13
Recommendation 13
L04 - Conformance to Solidity Naming Conventions 14
Description 14

Recommendation 14
L05 - Unused State Variable 15
Description 15
Recommendation 15
L07 - Missing Events Arithmetic 16
Description 16
Recommendation 16
L09 - Dead Code Elimination 17
Description 17
Recommendation 17
L13 - Divide before Multiply Operation 18
Description 18
Recommendation 18
Contract Functions 19
Contract Flow 24
Domain Info 25
Summary 26
Disclaimer 27
About Cyberscope 28

# Contract Review

| | |
|---|---|
| **Contract Name** | StackUp |
| **Compiler Version** | v0.7.6+commit.7338295f |
| **Optimization** | 200 runs |
| **Licence** | MIT |
| **Explorer** | https://bscscan.com/token/0xBF085B601A4B4D7104Ae1D4abDD07099e72c3460 |
| **Symbol** | UP |
| **Decimals** | 18 |
| **Total Supply** | 500,000 |
| **Domain** | stackup.finance |

# Source Files

| **Filename** | **SHA256** |
|---|---|
| **contract.sol** | 1ad888ba5d3a35bac51e26858980e6a31d2cd2bc53e7fd57ae63ddf273427e4a |

# Audit Updates

| | |
|---|---|
| **Initial Audit** | 8th June 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#L580,954 |

## Description

The contract owner has the authority to stop transactions for all users excluding the owner. The owner may take advantage of it by setting the `maxSellTransactionAmount` to zero. This will Convert the contract into a **HONEYPOT** and prevent users from selling.

```
if (
    automatedMarketMakerPairs[recipient] &&
    !excludedAccount
) {
    require(amount <= maxSellTransactionAmount, "Error amount");
```

The contract owner has the authority to stop transactions for all users excluding the owner. The owner may take advantage of it by setting the `SellLimit` to zero.

```
uint256 onePercent = balanceOf(sender).mul(SellLimit).div(100); //Should use
variable
            require(amount <= onePercent, "ERR: Can't sell more than set %");
```

Another action that can produce users to be able to sell up to 1% each day is setting the SellLimit to 1 and the TwentyFourhours variable to a very high value.

```
 else if( (blkTime < tradeData[sender].lastTradeTime + TwentyFourhours) && ((
blkTime > tradeData[sender].lastTradeTime)) ){
                require(tradeData[sender].tradeAmount + amount <= onePercent,
"ERR: Can't sell more than 1% in One day");
                tradeData[sender].tradeAmount = tradeData[sender].tradeAmount +
amount;
            }
```

## Recommendation

The contract could embody a check for not allowing setting the maxSellTransactionAmount less than a reasonable amount and similar to the

TwentyFourhours and SellLimit variables. A suggested implementation could check that the maximum amount should be more than a fixed percentage of the total supply.

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#L895 |

## 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 `setFees` function with a high percentage value. The MAX_FEE_RATE variable will check each tax individually but not as a total and that will result in a total max buy fee of 25% and sell fee of 65%.

```
  function setFees(uint256 _liquidityFee, uint256 _riskFreeValue, uint256
_treasuryFee, uint256 _sellFeeTreasuryAdded, uint256 _sellFeeRFVAdded, uint256
_feeDenominator) external onlyWhitelisted {
        require(
            _liquidityFee <= MAX_FEE_RATE &&
            _riskFreeValue <= MAX_FEE_RATE &&
            _treasuryFee <= MAX_FEE_RATE &&
            _sellFeeTreasuryAdded <= MAX_FEE_RATE &&
            _sellFeeRFVAdded <= MAX_FEE_RATE,
            "wrong"
        );

        liquidityFee = _liquidityFee;
        buyFeeRFV = _riskFreeValue;
        treasuryFee = _treasuryFee;
        sellFeeTreasuryAdded = _sellFeeTreasuryAdded;
        sellFeeRFVAdded = _sellFeeRFVAdded;
        totalBuyFee = liquidityFee.add(treasuryFee).add(buyFeeRFV);
        totalSellFee =
totalBuyFee.add(sellFeeTreasuryAdded).add(sellFeeRFVAdded);
        feeDenominator = _feeDenominator;
        require(totalBuyFee < feeDenominator / 4);
    }
```

## Recommendation

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

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 - Blacklisted Contracts

| | |
|---|---|
| **Criticality** | medium |
| **Location** | contract.sol#L865 |

## Description

The contract owner has the authority to stop contracts from transactions. The owner may take advantage of it by calling the `setBlackListed` function.

```
if(_isBlackListed[sender] && !automatedMarketMakerPairs[recipient]){
        revert("can't do P2p");
    }
```

## 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

| Severity | Code | Description |
|---|---|---|
| Medium | MTS | Manipulate Total Supply |
| Minor | L01 | Public Function could be Declared External |
| Minor | L02 | State Variables could be Declared Constant |
| Minor | L04 | Conformance to Solidity Naming Conventions |
| Minor | L05 | Unused State Variable |
| Minor | L07 | Missing Events Arithmetic |
| Minor | L09 | Dead Code Elimination |
| Minor | L13 | Divide before Multiply Operation |

# MTS - Manipulate Total Supply

| | |
|---|---|
| **Criticality** | medium |
| **Location** | contract.sol#L805 |

## Description

Owner is able to manipulate total supply. This change will have a direct impact on the token price and Market Cap.

```
if (supplyDelta < 0) {
            _totalSupply = _totalSupply.sub(uint256(-supplyDelta));
        } else {
            _totalSupply = _totalSupply.add(uint256(supplyDelta));
        }
```

The owner can also set `rewardYield` and `rewardYieldDenominator` to any value without limit, this will have a direct impact on the totalSupply.

```
   function setRewardYield(uint256 _rewardYield, uint256
_rewardYieldDenominator) external onlyWhitelisted {
        rewardYield = _rewardYield;
        rewardYieldDenominator = _rewardYieldDenominator;
    }
```

## Recommendation

The contract owner should carefully manage the adjustment of the circulating supply (increases or decreases), according to the token's price fluctuations.

# L01 - Public Function could be Declared External

| | |
|---|---|
| **Criticality** | minor |
| **Location** | contract.sol#L159,163,167,254,263,268,310,315,319,335,339 |

## Description

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

```
removeRebaseAdmin
addRebaseAdmin
renounceWhitelisted
removeWhitelisted
addWhitelisted
transferOwnership
renounceOwnership
owner
decimals
...
```

## Recommendation

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

# L02 - State Variables could be Declared Constant

| | |
|---|---|
| **Criticality** | minor |
| **Location** | contract.sol#L384,385,390 |

## Description

Constant state variables should be declared constant to save gas.

```
busdToken
ZERO
DEAD
```

## Recommendation

Add the constant attribute to state variables that never change.

# L04 - Conformance to Solidity Naming Conventions

| | |
|---|---|
| **Criticality** | minor |
| **Location** | contract.sol#L175,417,483,487,834,855,860,865,870,874,883,888,895,916,925,930,935,940,945,950,954,371,372,373,384,385,409,423 |

## 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.

```
TwentyFourhours
SellLimit
ZERO
DEAD
_markerPairs
_isBlackListed
_isFeeExempt
_maxTxn
_nextRebase
...
```

## 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#L11 |

## Description

There are segments that contain unused state variables.

```
MAX_INT256
```

## Recommendation

Remove unused state variables.

# L07 - Missing Events Arithmetic

| | |
|---|---|
| **Criticality** | minor |
| **Location** | contract.sol#L870,874,878,883,895,930,935,950,954 |

## 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.

```
maxSellTransactionAmount = _maxTxn
nextRebase = _nextRebase
rewardYield = _rewardYield
rebaseFrequency = _rebaseFrequency
liquidityFee = _liquidityFee
gonSwapThreshold = TOTAL_GONS.div(_denom).mul(_num)
targetLiquidity = target
TwentyFourhours = _time
SellLimit = _addr
```

## Recommendation

Emit an event for critical parameter changes.

# L09 - Dead Code Elimination

| | |
|---|---|
| **Criticality** | minor |
| **Location** | contract.sol#L39 |

## Description

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

```
abs
```

## Recommendation

Remove unused functions.

# L13 - Divide before Multiply Operation

| | |
|---|---|
| **Criticality** | minor |
| **Location** | contract.sol#L713,883 |

## Description

Performing divisions before multiplications may cause lose of prediction.

```
gonSwapThreshold = TOTAL_GONS.div(_denom).mul(_num)
contractTokenBalance = _gonBalances[address(this)].div(_gonsPerFragment)
```

## Recommendation

The multiplications should be prior to the divisions.

# Contract Functions

| Contract | Type | Bases | | |
|---|---|---|---|---|
| | Function Name | Visibility | Mutability | Modifiers |
| | | | | |
| **SafeMathInt** | Library | | | |
| | mul | Internal | | |
| | div | Internal | | |
| | sub | Internal | | |
| | add | Internal | | |
| | abs | Internal | | |
| | | | | |
| **IERC20** | Interface | | | |
| | totalSupply | External | | - |
| | balanceOf | External | | - |
| | allowance | External | | - |
| | transfer | External | ✓ | - |
| | approve | External | ✓ | - |
| | transferFrom | External | ✓ | - |
| | | | | |
| **SafeMath** | Library | | | |
| | add | Internal | | |
| | sub | Internal | | |
| | sub | Internal | | |
| | mul | Internal | | |
| | div | Internal | | |
| | div | Internal | | |
| | mod | Internal | | |
| | | | | |
| **InterfaceLP** | Interface | | | |
| | sync | External | ✓ | - |
| | | | | |
| **Roles** | Library | | | |
| | add | Internal | ✓ | |

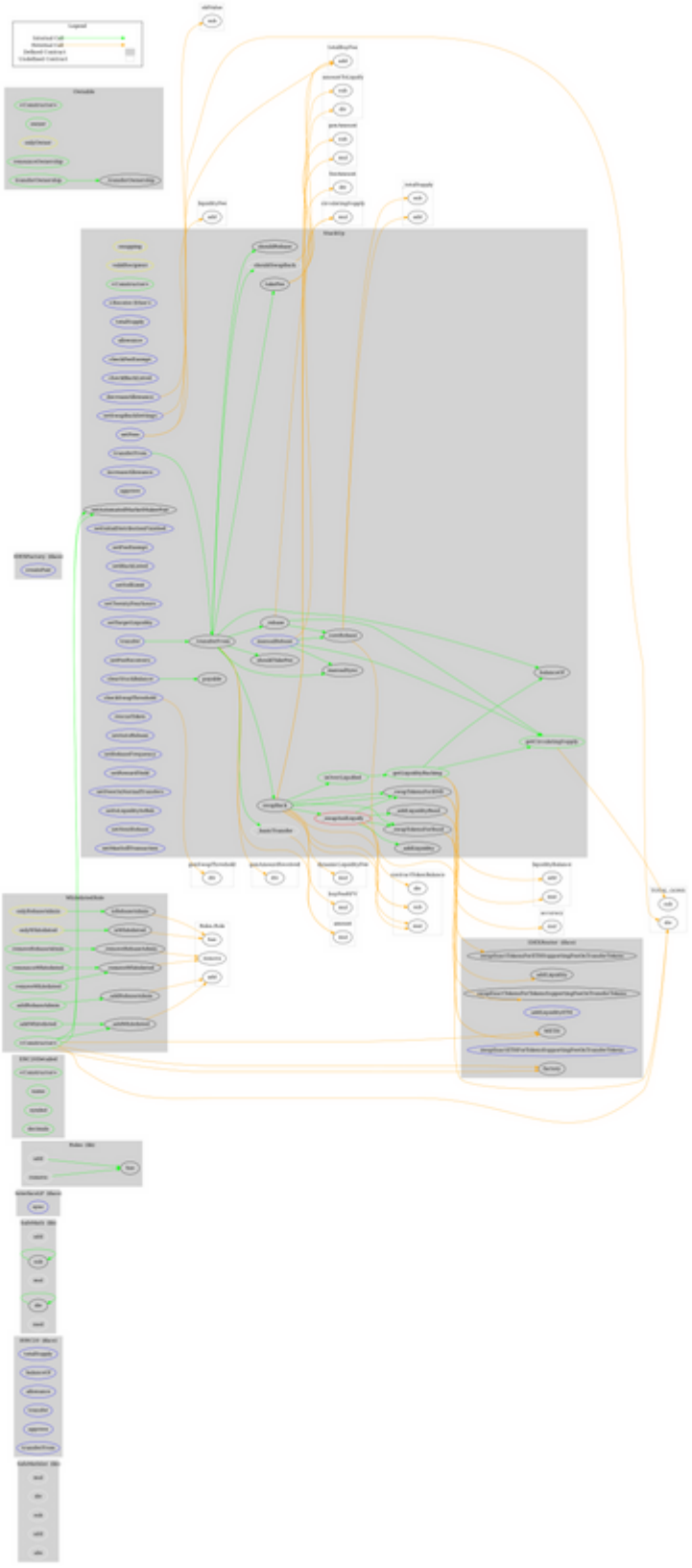
| | | | | |
|---|---|---|---|---|
| | remove | Internal | ✓ | |
| | has | Internal | | |
| | | | | |
| **ERC20Detailed** | Implementation | IERC20 | | |
| | <Constructor> | Public | ✓ | - |
| | name | Public | | - |
| | symbol | Public | | - |
| | decimals | Public | | - |
| | | | | |
| **IDEXRouter** | Interface | | | |
| | factory | External | | - |
| | WETH | External | | - |
| | addLiquidity | External | ✓ | - |
| | addLiquidityETH | External | Payable | - |
| | swapExactTokensForTokensSupportingFeeOnTransferTokens | External | ✓ | - |
| | swapExactETHForTokensSupportingFeeOnTransferTokens | External | Payable | - |
| | swapExactTokensForETHSupportingFeeOnTransferTokens | External | ✓ | - |
| | | | | |
| **IDEXFactory** | Interface | | | |
| | createPair | External | ✓ | - |
| | | | | |
| **Ownable** | Implementation | | | |
| | <Constructor> | Public | ✓ | - |
| | owner | Public | | - |
| | renounceOwnership | Public | ✓ | onlyOwner |
| | transferOwnership | Public | ✓ | onlyOwner |
| | _transferOwnership | Internal | ✓ | |
| | | | | |
| **WhitelistedRole** | Implementation | Ownable | | |
| | <Constructor> | Public | ✓ | - |
| | isWhitelisted | Public | | - |
| | isRebaseAdmin | Public | | - |

| | | | | |
|---|---|---|---|---|
| | addWhitelisted | Public | ✓ | onlyWhitelisted |
| | removeWhitelisted | Public | ✓ | onlyWhitelisted |
| | renounceWhitelisted | Public | ✓ | - |
| | _addWhitelisted | Internal | ✓ | |
| | _removeWhitelisted | Internal | ✓ | |
| | addRebaseAdmin | Public | ✓ | onlyWhitelisted |
| | removeRebaseAdmin | Public | ✓ | onlyWhitelisted |
| | _addRebaseAdmin | Internal | ✓ | |
| | _removeRebaseAdmin | Internal | ✓ | |
| | | | | |
| **StackUp** | Implementation | ERC20Detailed, Ownable, WhitelistedRole | | |
| | <Constructor> | Public | ✓ | ERC20Detailed |
| | <Receive Ether> | External | Payable | - |
| | totalSupply | External | | - |
| | allowance | External | | - |
| | balanceOf | Public | | - |
| | checkFeeExempt | External | | - |
| | checkBlackListed | External | | - |
| | checkSwapThreshold | External | | - |
| | shouldRebase | Internal | | |
| | shouldTakeFee | Internal | | |
| | shouldSwapBack | Internal | | |
| | getCirculatingSupply | Public | | - |
| | getLiquidityBacking | Public | | - |
| | isOverLiquified | Public | | - |
| | manualSync | Public | ✓ | - |
| | transfer | External | ✓ | validRecipient |
| | _basicTransfer | Internal | ✓ | |
| | _transferFrom | Internal | ✓ | |

| | | | | |
|---|---|---|---|---|
| | transferFrom | External | ✓ | validRecipient |
| | _swapAndLiquify | Private | ✓ | |
| | _addLiquidity | Private | ✓ | |
| | _addLiquidityBusd | Private | ✓ | |
| | _swapTokensForBNB | Private | ✓ | |
| | _swapTokensForBusd | Private | ✓ | |
| | swapBack | Internal | ✓ | swapping |
| | takeFee | Internal | ✓ | |
| | decreaseAllowance | External | ✓ | - |
| | increaseAllowance | External | ✓ | - |
| | approve | External | ✓ | - |
| | _rebase | Private | ✓ | |
| | coreRebase | Private | ✓ | |
| | manualRebase | External | ✓ | onlyRebaseAdmin |
| | setAutomatedMarketMakerPair | Public | ✓ | onlyWhitelisted |
| | setInitialDistributionFinished | External | ✓ | onlyWhitelisted |
| | setFeeExempt | External | ✓ | onlyWhitelisted |
| | setBlackListed | External | ✓ | onlyWhitelisted |
| | setSellLimit | External | ✓ | onlyWhitelisted |
| | setTwentyFourhours | External | ✓ | onlyWhitelisted |
| | setTargetLiquidity | External | ✓ | onlyWhitelisted |
| | setSwapBackSettings | External | ✓ | onlyWhitelisted |
| | setFeeReceivers | External | ✓ | onlyWhitelisted |
| | setFees | External | ✓ | onlyWhitelisted |
| | clearStuckBalance | External | ✓ | onlyWhitelisted |
| | rescueToken | External | ✓ | onlyWhitelisted |
| | setAutoRebase | External | ✓ | onlyWhitelisted |

| | | | | |
|---|---|---|---|---|
| | | | | d |
| | setRebaseFrequency | External | ✓ | onlyWhitelisted |
| | setRewardYield | External | ✓ | onlyWhitelisted |
| | setFeesOnNormalTransfers | External | ✓ | onlyWhitelisted |
| | setIsLiquidityInBnb | External | ✓ | onlyWhitelisted |
| | setNextRebase | External | ✓ | onlyWhitelisted |
| | setMaxSellTransaction | External | ✓ | onlyWhitelisted |

# Contract Flow



# Domain Info

| | |
|---|---|
| **Domain Name** | stackup.finance |
| **Registry Domain ID** | db91da8b1b0b4c60939991996eecc1db-DONUTS |
| **Creation Date** | 2022-05-23T15:25:10Z |
| **Updated Date** | 2022-06-03T22:33:13Z |
| **Registry Expiry Date** | 2023-05-23T15:25:10Z |
| **Registrar WHOIS Server** | www.whois.domainnameshop.com |
| **Registrar URL** | http://www.domainnameshop.com |
| **Registrar** | Domeneshop AS dba domainnameshop.com |
| **Registrar IANA ID** | 1001 |

The domain has been created 16 days before the creation of the audit. It will expire in 12 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 stopping transactions, manipulating fees and blacklisting addresses. 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.

# 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