



Cyberscope

Audit Report

Mafia Ape

April 2022

Type BEP20

Network BSC

Address 0x3e81e7068ce0a0e08444e10e4225a6e205846f880

Audited by © cyberscope

Table of Contents

Table of Contents	1
Contract Review	3
Audit Updates	3
Source Files	4
Contract Analysis	5
ST - Stop Transactions	6
Description	6
Recommendation	6
ELFM - Exceed Limit Fees Manipulation	7
Description	7
Recommendation	7
ULTW - Unlimited Liquidity to Team Wallet	8
Description	8
Recommendation	8
Contract Diagnostics	10
FSA - Fixed Swap Address	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
Contract Functions	18
Contract Flow	21
Domain Info	22
Summary	23
Disclaimer	24
About Cyberscope	25

Contract Review

Contract Name	MafiaApe
Compiler Version	v0.8.4+commit.c7e474f2
Optimization	200 runs
Licence	
Explorer	https://bscscan.com/token/0x3e81e7068ceaaea8444e10e4225a6e205846f880
Symbol	mAPE
Decimals	18
Total Supply	1,000,000,000
Domain	mafiaape.io

Audit Updates

Initial Audit	23rd April 2022
Corrected	

Source Files

Filename	SHA256
Context.sol	4c155cc34bf55030ccc9de46ab76ceb2c66eafaf669bd38214f9a73215f88380
interface/IUniswapV2Factory.sol	9406d8ab7a2aca3e1207cd847049a2969ba5ec8ceb52e131e29585b9272404b1
interface/IUniswapV2Router02.sol	fa569fddc50da2b4a58e6e368fbb9abc2ee05d0f1363257a24d3ab6c2be89e39
library/IBEP20.sol	bafe38c0b7fef77b36027ae55c7e9cbd0d02bd6091748916198dbbc7d8ab0bcb
library/SafeMath.sol	17b0bbbc562d95c2b358837a4427bcca734e917ba01f0b0cbc904fd0e912563e
MafiaApe.sol	850670d9476be077127032684af32b0f91508144102eaea80bd180b350e0426a
Ownable.sol	111f14c1de81bd02139cea6cebf29d7129402ee9c26e03291cc2fde6297671b3

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#L400

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 high fees. This can cause the contract to operate as a honeypot.

```
if (!_isExcludedFromFee[from] || !_isExcludedFromFee[to]) {  
    takeFee = false;  
}
```

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

```
require(  
    _tokenSold[from] <= _maxSoldAmount,  
    "Sold amount exceeds the maxTxAmount."  
);
```

Recommendation

The contract could embody a check for not allowing setting the `_maxSoldAmount` less than a reasonable amount. A suggested implementation could check that the maximum amount should be more than a fixed percentage of the total supply.

Read more about the fees manipulation in the [corresponding 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#L168

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 `setDefaultTaxFee` function with a high percentage value.

```
function setDefaultTaxFee(
    uint256 _defaultTaxFee,
    uint256 _defaultLiquidity,
    uint256 _defaultMarketing,
    uint256 _defaultBuyback
) public onlyOwner {
    require(
        _defaultTaxFee ==
        _defaultLiquidity.add(_defaultMarketing).add(_defaultBuyback),
        "_defaultTaxFee must equal to sum of _defaultLiquidity,
        _defaultMarketing and _defaultBuyback"
    );
    defaultTaxFee = _defaultTaxFee;
    defaultLiquidity = _defaultLiquidity;
    defaultMarketing = _defaultMarketing;
    defaultBuyback = _defaultBuyback;
    _taxFee = defaultTaxFee;
}
```

Recommendation

The contract could embody a check for the 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.

ULTW - Unlimited Liquidity to Team Wallet

Criticality	minor
Location	contract.sol#L522

Description

The contract owner has the authority to transfer funds without limit to the team wallet. These funds have been accumulated from fees collected from the contract. The owner may take advantage of it by calling the `contractBalanceSwap` and the `contractBalanceSend` methods.

```
function contractBalanceSwap() external onlyOwner {
    uint256 contractBalance = balanceOf(address(this));
    swapTokensForEth(contractBalance);
}

/**
 * @dev the Owner can send regarding the mAPE token's amount of contract
balance
 * this is for manual function
 * 0.1BNB will remain in contract balance for swap and transfer fees.
 */

function contractBalanceSend(uint256 amount, address payable _destAddr)
    external
    onlyOwner
{
    uint256 contractETHBalance = address(this).balance - 1 * 10**17;
    if (contractETHBalance > amount) {
        _destAddr.transfer(amount);
    }
}
```

Recommendation

The contract could embody a check for the maximum amount of funds that can be swapped. Since a huge amount may volatile the token's price.

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
●	FSA	Fixed Swap Address
●	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

FSA - Fixed Swap Address

Criticality	minor
Location	contract.sol#L1

Description

The swap address is assigned once in the constructor and it can not be changed. The decentralized swaps sometimes create a new swap version or abandon the current. A contract that cannot change the swap address may not be able to catch-up the upgrade.

here

Recommendation

It could be better to allow the swap address mutation in case of future swap updates.

L01 - Public Function could be Declared External

Criticality

minor

Location

contracts/MafiaApe.sol#L160,164,168,228,272,294,324,351

Description

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

```
decreaseAllowance  
increaseAllowance  
transferFrom  
approve  
burn  
setDefaultTaxFee  
includeInFee  
excludeFromFee
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality

minor

Location

contracts/MafiaApe.sol#L62

Description

Constant state variables should be declared constant to save gas.

```
swapEnabled
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contracts/MafiaApe.sol#L169,170,171,172,533,604,616,624,632,639,44,49,50,58,59

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.

```
_buybackWallet  
_marketingWallet  
_minBalance  
_taxFee  
_maxSoldAmount  
_setBuybackWalletAddress  
_setMarketingWalletAddress  
_setApplyContractFee  
_setMinBalance  
...
```

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

contracts/MafiaApe.sol#L42

Description

There are segments that contain unused state variables.

```
_blockTime
```

Recommendation

Remove unused state variables.

L07 - Missing Events Arithmetic

Criticality

minor

Location

contracts/MafiaApe.sol#L168,604,616

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.

```
_minBalance = minValue  
_maxSoldAmount = maxvalue  
defaultTaxFee = _defaultTaxFee
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality

minor

Location

contracts/MafiaApe.sol#L195

Description

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

```
_mint
```

Recommendation

Remove unused functions.

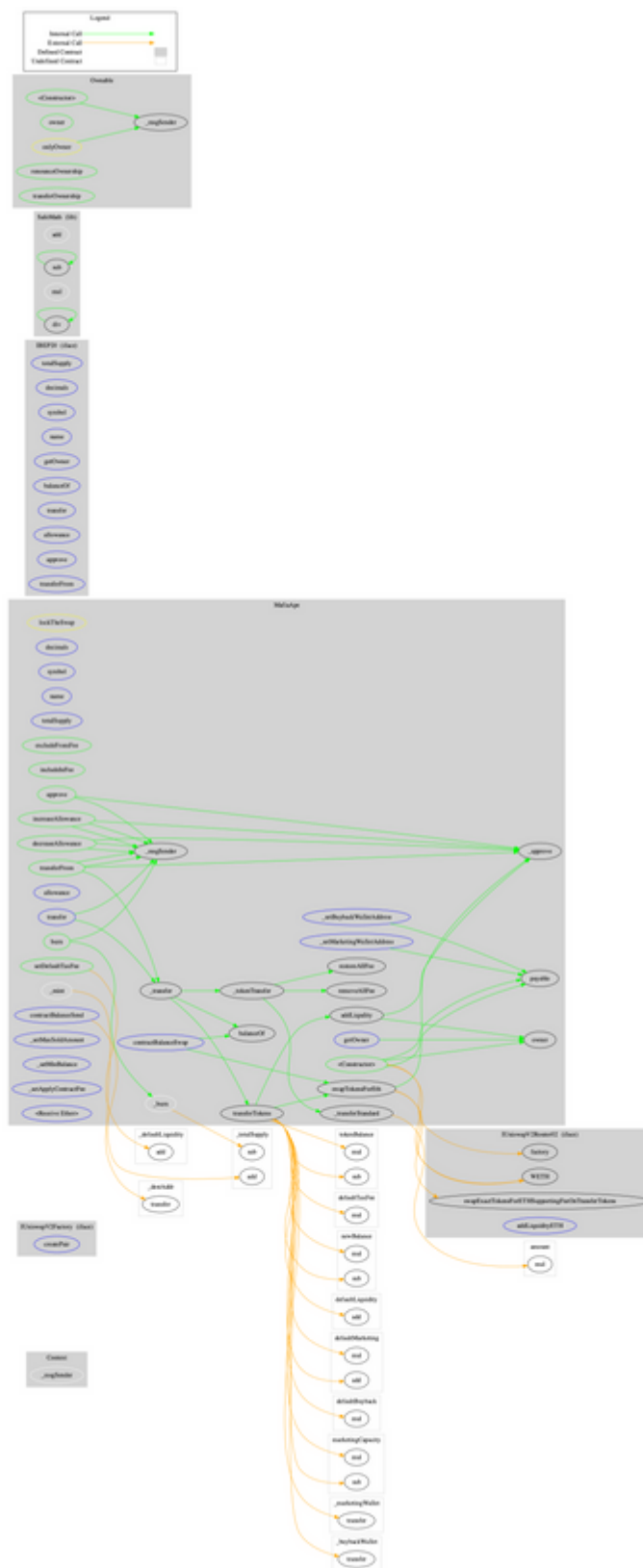
Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
Context	Implementation			
	_msgSender	Internal		
IUniswapV2Factory	Interface			
	createPair	External	✓	-
IUniswapV2Router02	Interface			
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
	factory	External		-
	WETH	External		-
	addLiquidityETH	External	Payable	-
IBEP20	Interface			
	totalSupply	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	getOwner	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		
MafiaApe	Implementation	Context, IBEP20, Ownable		
	<Constructor>	Public	✓	-
	getOwner	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	totalSupply	External		-
	balanceOf	Public		-
	excludeFromFee	Public	✓	onlyOwner
	includeInFee	Public	✓	onlyOwner
	setDefaultTaxFee	Public	✓	onlyOwner
	_mint	Internal	✓	
	_burn	Internal	✓	
	burn	Public	✓	-
	transfer	External	✓	-
	allowance	External		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_approve	Internal	✓	
	transferTokens	Private	✓	lockTheSwap
	swapTokensForEth	Private	✓	
	addLiquidity	Private	✓	
	contractBalanceSwap	External	✓	onlyOwner
	contractBalanceSend	External	✓	onlyOwner
	removeAllFee	Private	✓	
	restoreAllFee	Private	✓	

	_tokenTransfer	Private	✓	
	_transferStandard	Private	✓	
	_setMaxSoldAmount	External	✓	onlyOwner
	_setMinBalance	External	✓	onlyOwner
	_setApplyContractFee	External	✓	onlyOwner
	_setMarketingWalletAddress	External	✓	onlyOwner
	_setBuybackWalletAddress	External	✓	onlyOwner
	<Receive Ether>	External	Payable	-
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner

Contract Flow



Domain Info

Domain Name	mafiaape.io
Registry Domain ID	2eb6e1fb6b1c4c848c185c5936142db1-DONUTS
Creation Date	2022-04-20T15:22:03Z
Updated Date	2022-04-21T11:02:06Z
Registry Expiry Date	2023-04-20T15:22:03Z
Registrar WHOIS Server	whois.name.com
Registrar URL	http://www.name.com
Registrar	Name.com, Inc.
Registrar IANA ID	625

The domain has been created 3 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 manipulating fees, stopping transactions and transferring funds to the team's wallet. 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.

Cyberscope 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 provides all the essential tools to assist users draw their own conclusions.



The Cyberscope team

<https://www.cyberscope.io>