Cyberscope

Audit Report

MetaVision Income Token

April 2022

Type	BEP20
Network	BSC
Address	0x3c6C6d32609d253d54951dB187DAEc272b1d848a
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 8

ULTW - Unlimited Liquidity to Team Wallet 9

Description 9

Recommendation 9

BC - Blacklisted Contracts 10

Description 10

Recommendation 10

Contract Diagnostics 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
L07 - Missing Events Arithmetic 15
Description 15
Recommendation 15
L09 - Dead Code Elimination 16
Description 16
Recommendation 16
Contract Functions 17
Contract Flow 22
Domain Info 23
Summary 24
Disclaimer 25
About Cyberscope 26

Contract Review

Contract Name	TokenTool
Compiler Version	v0.8.4+commit.c7e474f2
Optimization	200 runs
Licence	None
Explorer	https://bscscan.com/token/0x3c6C6d32609d253d54951dB187DAEc272b1d848a
Symbol	VIT
Decimals	18
Total Supply	1,000,000,000
Domain	metavisionclubs.com

Source Files

Filename	**SHA256**
contract.sol	91c963bd518cc587976c311aa2628895290384029d52ceda0cc39065946963ca

Audit Updates

Initial Audit	26th April 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#L752, 771, 859

Description

The contract owner has the authority to stop transactions for all users excluding the owner. The owner can do that in multiple ways. The owner may take advantage of it by setting the _maxTxAmount to zero.

```
if(!isTxLimitExempt[sender] && !isTxLimitExempt[recipient]) {
                require(amount <= _maxTxAmount, "Transfer amount exceeds the
maxTxAmount.");
            }
```

The contract owner can also stop transactions by setting the _walletMax to zero.

```
if(checkWalletLimit && !isWalletLimitExempt[recipient])
                require(balanceOf(recipient).add(finalAmount) <= _walletMax);
```

The owner can also convert the contract into a honeypot and prevent users from selling by setting the _totalTaxIfSelling to a high percentage value.

```
 else if(isMarketPair[recipient]) {
            feeAmount =
amount.mul(_totalTaxIfSelling.sub(_sellDestroyFee)).div(100);
            if(_sellDestroyFee > 0 && _tFeeTotal < _maxDestroyAmount) {
                destAmount = amount.mul(_sellDestroyFee).div(100);
                destroyFee(sender,destAmount);
            }
        }
```

Recommendation

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

Also, the contract could embody a check for the maximum acceptable value for the _totalTaxIfSelling. A suggested implementation could check that the maximum amount should be less 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#L621, 626, 631, 639

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 setBuyDestFee , setSellDestFees , setBuyTaxes and setSelTaxes functions with a high percentage value.

```
function setBuyDestFee(uint256 newBuyDestroyFee) public onlyOwner {
        _buyDestroyFee = newBuyDestroyFee;
        _totalTaxIfBuying =
_buyLiquidityFee.add(_buyMarketingFee).add(_buyTeamFee).add(_buyDestroyFee);
    }
```

```
    function setSellDestFee(uint256 newSellDestroyFee) public onlyOwner {
        _sellDestroyFee = newSellDestroyFee;
        _totalTaxIfSelling =
_sellLiquidityFee.add(_sellMarketingFee).add(_sellTeamFee).add(_sellDestroyFee
);
    }
```

```
function setBuyTaxes(uint256 newLiquidityTax, uint256 newMarketingTax, uint256
newTeamTax) external onlyOwner() {
        _buyLiquidityFee = newLiquidityTax;
        _buyMarketingFee = newMarketingTax;
        _buyTeamFee = newTeamTax;

        _totalTaxIfBuying =
_buyLiquidityFee.add(_buyMarketingFee).add(_buyTeamFee).add(_buyDestroyFee);
    }
    function setSelTaxes(uint256 newLiquidityTax, uint256 newMarketingTax,
uint256 newTeamTax) external onlyOwner() {
        _sellLiquidityFee = newLiquidityTax;
        _sellMarketingFee = newMarketingTax;
        _sellTeamFee = newTeamTax;

        _totalTaxIfSelling =
_sellLiquidityFee.add(_sellMarketingFee).add(_sellTeamFee).add(_sellDestroyFee
);
    }
```

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#L677, 757

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 setting a high value to the _minimumTokensBeforeSwap variable.

```
    function setNumTokensBeforeSwap(uint256 newLimit) external onlyOwner() {
        _minimumTokensBeforeSwap = newLimit;
    }
```

```
bool overMinimumTokenBalance = contractTokenBalance >=
_minimumTokensBeforeSwap;

            if (overMinimumTokenBalance && !inSwapAndLiquify &&
!isMarketPair[sender] && swapAndLiquifyEnabled)
            {
                if(swapAndLiquifyByLimitOnly)
                    contractTokenBalance = _minimumTokensBeforeSwap;
                swapAndLiquify(contractTokenBalance);
            }
```

Recommendation

The contract could embody a check for the maximum amount of funds that can be swapped. Since a huge amount may violate 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.

BC - Blacklisted Contracts

Criticality	critical
Location	contract.sol#L744

Description

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

```
require(!_isBlacklisted[sender] && !_isBlacklisted[recipient], 'Blacklisted
address');
```

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
⚪	L01	Public Function could be Declared External
⚪	L02	State Variables could be Declared Constant
⚪	L04	Conformance to Solidity Naming Conventions
⚪	L07	Missing Events Arithmetic
⚪	L09	Dead Code Elimination

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L165,170,176,554,558,562,566,574,578,583 and 14 more

Description

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

```
transferFrom
transfer
changeRouterVersion
getCirculatingSupply
setSwapAndLiquifyByLimitOnly
setSwapAndLiquifyEnabled
multipleBotlistAddress
setSellDestFee
setBuyDestFee
...
```

Recommendation

Use the external attribute for functions never called from the contract

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L434

Description

Constant state variables should be declared constant to save gas.

```
deadAddress
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L151,224,226,243,273,689,435,436,444,445 and 16 more

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.

```
_walletMax
_maxTxAmount
_maxDestroyAmount
_tFeeTotal
_totalTaxIfSelling
_totalTaxIfBuying
_totalDistributionShares
_teamShare
_marketingShare
...
```

Recommendation

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

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L617,621,626,631,639,653,661,673,677

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.

```
_minimumTokensBeforeSwap = newLimit
_walletMax = newLimit
_maxTxAmount = maxTxAmount
_liquidityShare = newLiquidityShare
_totalTaxIfSelling =
_sellLiquidityFee.add(_sellMarketingFee).add(_sellTeamFee).add(_sellDestroyFee)
_totalTaxIfBuying =
_buyLiquidityFee.add(_buyMarketingFee).add(_buyTeamFee).add(_buyDestroyFee)
_sellDestroyFee = newSellDestroyFee
_buyDestroyFee = newBuyDestroyFee
_maxDestroyAmount = maxDestroy
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L130,113,117,121,125,94,105

Description

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

```
sendValue
isContract
functionCallWithValue
functionCall
_functionCallWithValue
```

Recommendation

Remove unused functions.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
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		
Address	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	

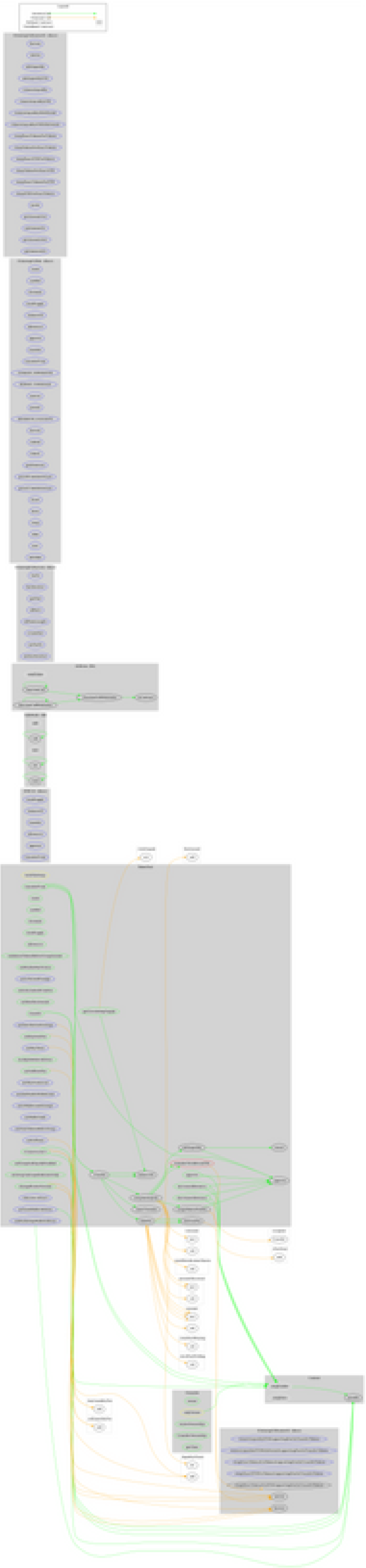
	functionCallWithValue	Internal	✓	
	_functionCallWithValue	Private	✓	
Ownable	Implementation	Context		
	owner	Public		-
	waiveOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	getTime	Public		-
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
IUniswapV2Pair	Interface			
	name	External		-
	symbol	External		-
	decimals	External		-
	totalSupply	External		-
	balanceOf	External		-
	allowance	External		-
	approve	External	✓	-
	transfer	External	✓	-
	transferFrom	External	✓	-
	DOMAIN_SEPARATOR	External		-
	PERMIT_TYPEHASH	External		-
	nonces	External		-
	permit	External	✓	-
	MINIMUM_LIQUIDITY	External		-

	factory	External		-
	token0	External		-
	token1	External		-
	getReserves	External		-
	price0CumulativeLast	External		-
	price1CumulativeLast	External		-
	kLast	External		-
	burn	External	✓	-
	swap	External	✓	-
	skim	External	✓	-
	sync	External	✓	-
	initialize	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	✓	-
TokenTool	Implementation	Context, IERC20, Ownable		
	<Constructor>	Public	Payable	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	allowance	Public		-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	minimumTokensBeforeSwapAmount	Public		-
	approve	Public	✓	-
	_approve	Private	✓	
	setMarketPairStatus	Public	✓	onlyOwner
	setIsTxLimitExempt	External	✓	onlyOwner
	setIsExcludedFromFee	Public	✓	onlyOwner
	setMaxDesAmount	Public	✓	onlyOwner
	setBuyDestFee	Public	✓	onlyOwner
	setSellDestFee	Public	✓	onlyOwner
	setBuyTaxes	External	✓	onlyOwner
	setSelTaxes	External	✓	onlyOwner
	multipleBotlistAddress	Public	✓	onlyOwner
	setDistributionSettings	External	✓	onlyOwner

	setMaxTxAmount	External	✓	onlyOwner
	enableDisableWalletLimit	External	✓	onlyOwner
	setIsWalletLimitExempt	External	✓	onlyOwner
	setWalletLimit	External	✓	onlyOwner
	setNumTokensBeforeSwap	External	✓	onlyOwner
	setMarketingWalletAddress	External	✓	onlyOwner
	setTeamWalletAddress	External	✓	onlyOwner
	setSwapAndLiquifyEnabled	Public	✓	onlyOwner
	setSwapAndLiquifyByLimitOnly	Public	✓	onlyOwner
	getCirculatingSupply	Public		-
	transferToAddressETH	Private	✓	
	changeRouterVersion	Public	✓	onlyOwner
	<Receive Ether>	External	Payable	-
	transfer	Public	✓	-
	transferFrom	Public	✓	-
	_transfer	Private	✓	
	_basicTransfer	Internal	✓	
	swapAndLiquify	Private	✓	lockTheSwap
	swapTokensForEth	Private	✓	
	addLiquidity	Private	✓	
	takeFee	Internal	✓	
	destroyFee	Private	✓	

Contract Flow



Domain Info

Domain Name	metavisionclubs.com
Registry Domain ID	2688938394_DOMAIN_COM-VRSN
Creation Date	2022-04-13T06:42:12Z
Updated Date	2022-04-13T06:42:13Z
Registry Expiry Date	2024-04-13T06:42:12Z
Registrar WHOIS Server	whois.godaddy.com
Registrar URL	https://www.godaddy.com
Registrar	GoDaddy.com, LLC
Registrar IANA ID	146

The domain has been created 13 days before the creation of the audit. It will expire in almost 2 years.

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, blacklisting wallets and stopping transactions. 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 provides all the essential tools to assist users draw their own conclusions.

The Cyberscope team

https://www.cyberscope.io