



Cyberscope

Audit Report

EverIncome

April 2022

Type BEP20

Network BSC

Address 0x981f843475a0b9adBDd1b09607f995502f843BE2

Audited by © cyberscope

Table of Contents

Table of Contents	1
Contract Review	3
Source Files	3
Audit Updates	3
Contract Analysis	4
ELFM - Exceed Limit Fees Manipulation	5
Description	5
Recommendation	5
Contract Diagnostics	6
L01 - Public Function could be Declared External	7
Description	7
Recommendation	7
L02 - State Variables could be Declared Constant	8
Description	8
Recommendation	8
L04 - Conformance to Solidity Naming Conventions	9
Description	9
Recommendation	9
L05 - Unused State Variable	10
Description	10
Recommendation	10
L07 - Missing Events Arithmetic	11
Description	11
Recommendation	11
L09 - Dead Code Elimination	12
Description	12

Recommendation	12
L11 - Unnecessary Boolean equality	13
Description	13
Recommendation	13
Contract Functions	14
Contract Flow	18
Domain Info	19
Summary	20
Disclaimer	21
About Cyberscope	22

Contract Review

Contract Name	EverIncome
Compiler Version	v0.8.0+commit.c7dfd78e
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0x981f843475a0b9adBDd1b09607f995502f843BE2
Symbol	EIC
Decimals	9
Total Supply	100,000,000
Domain	everincomecoin.com

Source Files

Filename	SHA256
contract.sol	c1641498292d4d2dd74ae476db3c95dcad097f9db1bf058dfb7878b47abfa2b9

Audit Updates

Initial Audit	22nd 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

ELFM - Exceed Limit Fees Manipulation

Criticality	minor
Location	contract.sol#L570

Description

The contract owner has the authority to increase over the allowed limit of 25% during the first day of trading. The owner may take advantage of it by setting the total fees to 25%. Then, the sales of the first day will be increased to $25\% * 1.8 = 45\%$.

```
function getMultipliedFee() public view returns (uint256) {
    if (launchedAtTimestamp + 1 days > block.timestamp) {
        return totalFee.mul(18000).div(feeDenominator);
    } else if (buybackMultiplierTriggeredAt.add(buybackMultiplierLength) >
block.timestamp) {
        uint256 remainingTime =
buybackMultiplierTriggeredAt.add(buybackMultiplierLength).sub(block.timestamp);
        uint256 feeIncrease =
totalFee.mul(buybackMultiplierNumerator).div(buybackMultiplierDenominator).sub(t
otalFee);
        return
totalFee.add(feeIncrease.mul(remainingTime).div(buybackMultiplierLength));
    }
    return totalFee;
}
```

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.

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#L135,142,163,701

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#L243,256,401,403,405,404,411

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#L178,281,234,242,243,681,730,740,745,750,755,401,402,403,404,405,407,408,409,411,412,414,415

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#L401,405

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#L281,681,690,707,730,745,750

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#L697

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#L496

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

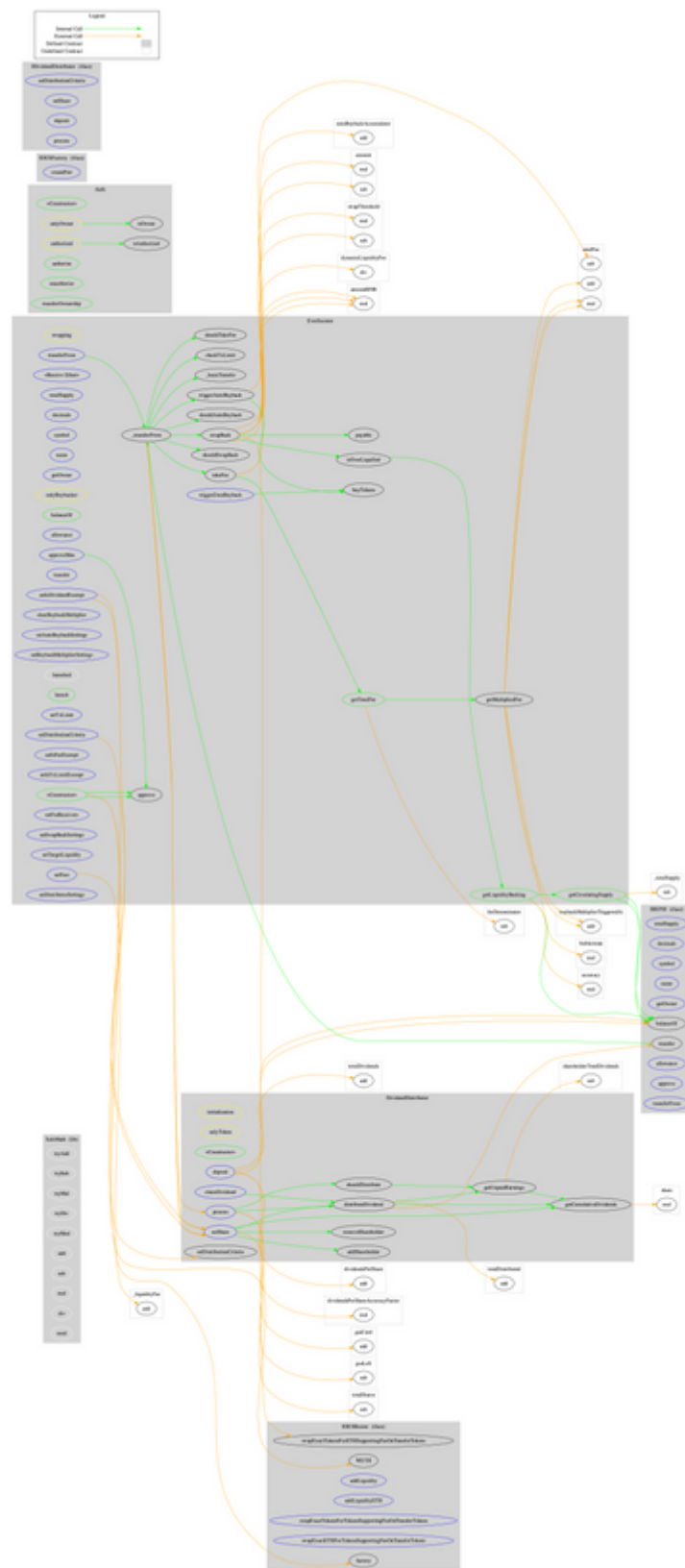
Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
SafeMath	Library			
	tryAdd	Internal		
	trySub	Internal		
	tryMul	Internal		
	tryDiv	Internal		
	tryMod	Internal		
	add	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	mod	Internal		
	sub	Internal		
	div	Internal		
	mod	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	✓	
EverIncome	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	everincomecoin.com
Registry Domain ID	2683687902_DOMAIN_COM-VRSN
Creation Date	2022-03-23T08:57:34Z
Updated Date	2022-03-23T08:57:35Z
Registry Expiry Date	2023-03-23T08:57:34Z
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 about 1 month 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

EverIncome is an interesting project that has a friendly and growing community. The contract owner has the authority to increase the sale fees to 45% over the first day of trading. Other than that, the contract Owner can access some admin functions that can not be used in a malicious way to disturb the users' transactions.

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>