Audit Report

Revoluzion

February 2022

Github https://github.com/RevoluzionToken/Revoluzion

SHA256 412608be085f7697eff2e9da35e2ebd9cdbaf2866a7d753e7d7384e1e06d5c08

Audited by © coinscope

Table of Contents

Contract Review

Github	https://github.com/RevoluzionToken/Revoluzion
Commit	d25bf9ebab6299671b18b7e715d55e40cd2a79f1
SHA256	412608be085f7697eff2e9da35e2ebd9cdbaf2866a7d753e7d7384e1e06d5c08
Domain	revoluzion.io

Audit Updates

Initial Audit	14th February 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

Business Logic Consideration

Criticality	minor
Location	contract.sol#L1313

Description

The contract gives an extra reward to the users that hold many tokens. The reward is a dividend of the accumulated fees. According to the business logic of the contract, the recipient is marked as applicable if the balance of the sender is more than a threshold.
The recipient applicability should not be determined by the sender's balance but from the recipient's balance.

```
if (!diamondEligibility[sender]) {
    if (balanceOf(sender) >= diamond.minTokenRequired()) {
        try diamond.setEligibility(sender, true) {} catch {}
    } else {
        try diamond.setEligibility(sender, false) {} catch {}
    }
}
if (!diamondEligibility[recipient]) {
    if (balanceOf(sender) >= diamond.minTokenRequired()) {
        try diamond.setEligibility(recipient, true) {} catch {}
    } else {
        try diamond.setEligibility(recipient, false) {} catch {}
    }
}
```

Recommendation

The contract should check the recipient's balance rather than the sender's balance.

```
if (!diamondEligibility[sender]) {
    if (balanceOf(sender) >= diamond.minTokenRequired()) {
        try diamond.setEligibility(sender, true) {} catch {}
    } else {
        try diamond.setEligibility(sender, false) {} catch {}
    }
}
if (!diamondEligibility[recipient]) {
```

```
    if (balanceOf(recipient) >= diamond.minTokenRequired()) {
        try diamond.setEligibility(recipient, true) {} catch {}
    } else {
        try diamond.setEligibility(recipient, false) {} catch {}
    }
}
```

Contract Diagnostics

Critical Medium Minor

Severity	Code	Description
	CR	Code Repetition
	L01	Public Function could be Declared External
	L04	Conformance to Solidity Naming Conventions
	L09	Dead Code Elimination
	L11	Unnecessary Boolean equality
	L13	Divide before Multiply Operation

CR - Code Repetition

Criticality	minor
Location	contract.sol#L1306

Description

There are code segments that are repetitive in the contract. Those segments increase the code size of the contract unnecessarily.

```
if (!diamondEligibility[sender]) {
    if (balanceOf(sender) >= diamond.minTokenRequired()) {
        try diamond.setEligibility(sender, true) {} catch {}
    } else {
        try diamond.setEligibility(sender, false) {} catch {}
    }
}
if (!diamondEligibility[recipient]) {
    if (balanceOf(sender) >= diamond.minTokenRequired()) {
        try diamond.setEligibility(recipient, true) {} catch {}
    } else {
        try diamond.setEligibility(recipient, false) {} catch {}
    }
}
```

Recommendation

The contract could reuse the expression result in order to avoid the repetition.

```
if (!diamondEligibility[sender]) {
    bool enabled = balanceOf(sender) >= diamond.minTokenRequired()
    try diamond.setEligibility(sender, enabled) {} catch {}
}
if (!diamondEligibility[recipient]) {
    bool enabled = balanceOf(sender) >= diamond.minTokenRequired()
    try diamond.setEligibility(recipient, enabled) {} catch {}
}
```

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L502,509,1373,1437,1441,1470,1474

Description

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

```
setDividendDistributor
distributorInitialization
setDiamondDistributor
diamondInitialization
setFees
unauthorize
authorize
```

Recommendation

Use the external attribute for functions never called from the contract

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L583,717,657,943,879,1373,1394,1460,1501,1524,1531,1577 and 15 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.

```
_period
_amount
_cap
_enabled
_denominator
_target
_minDistribution
_minPeriod
_minTokenRequired
...
```

Recommendation

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

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L100,110,125,135,172,182,149,159,48,75,195,455 and 8 more

Description

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

```
trySub
tryMul
tryMod
tryDiv
tryAdd
mod
div
_msgData
verifyCallResult
...
```

Recommendation

Remove unused functions.

L11 - Unnecessary Boolean equality

Criticality	minor
Location	contract.sol#L951,992,1018,1054,1166

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[_msgSender()] == true,Not a buybacker)
diamonds[holder].eligible == false
diamonds[holder].eligible == true && diamonds[holder].eligibleTime +
diamondCycle <= previousCycleEnd
eligible == false && diamonds[holder].eligible == true
eligible == true && diamonds[holder].eligible == false
```

Recommendation

Remove the equality to the boolean constant.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L1380

Description

Performing divisions before multiplications may cause lose of prediction.

```
require(bool,string)(totalFee < feeDenominator.div(100).mul(15),Total fee should not be greater than 15%.)
```

Recommendation

The multiplications should be prior to the divisions.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
Address	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		
	functionStaticCall	Internal		
	functionDelegateCall	Internal	✓	
	functionDelegateCall	Internal	✓	
	verifyCallResult	Internal		
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		
Context	Implementation			

	_msgSender	Internal		
	_msgData	Internal		
	_msgValue	Internal		
Auth	Implementation	Context		
	<Constructor>	Public	✓	-
	authorize	Public	✓	onlyOwner
	unauthorize	Public	✓	onlyOwner
	isOwner	Public		-
	isAuthorized	Public		-
IERC20Extended	Interface			
	totalSupply	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
IUniswapV2Router01	Interface			

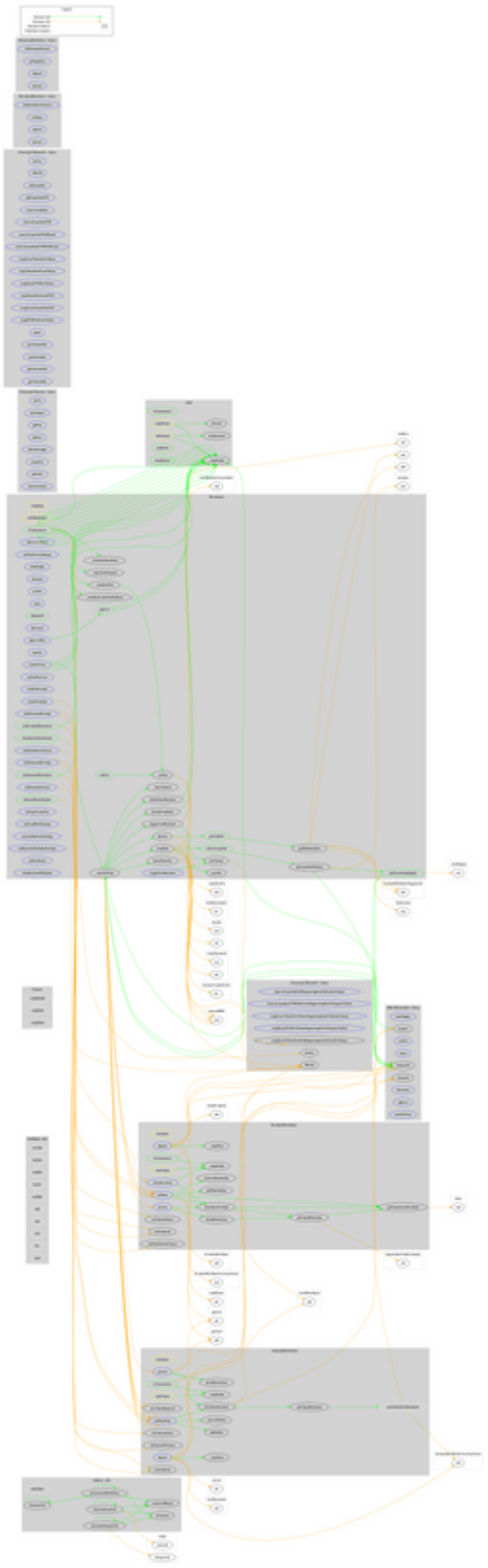
	factory	External		-
	WETH	External		-
	addLiquidity	External	✓	-
	addLiquidityETH	External	Payable	-
	removeLiquidity	External	✓	-
	removeLiquidityETH	External	✓	-
	removeLiquidityWithPermit	External	✓	-
	removeLiquidityETHWithPermit	External	✓	-
	swapExactTokensForTokens	External	✓	-
	swapTokensForExactTokens	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	✓	-
IDividendDistributor	Interface			
	setDistributionCriteria	External	✓	-
	setShare	External	✓	-
	deposit	External	Payable	-

	process	External	✓	-
DividendDistributor	Implementation	IDividendDistributor, Context		
	<Constructor>	Public	✓	-
	unInitialized	External	✓	onlyToken
	setTokenAddress	External	✓	initializer onlyToken
	setDistributionCriteria	External	✓	onlyToken
	setShare	External	✓	onlyToken
	deposit	External	Payable	onlyToken
	process	External	✓	onlyToken
	shouldDistribute	Internal		
	distributeDividend	Internal	✓	
	getCumulativeDividends	Internal		
	getUnpaidEarnings	Public		-
	addShareholder	Internal	✓	
	removeShareholder	Internal	✓	
	claimDividend	External	✓	-
IDiamondDistributor	Interface			
	setDiamondCriteria	External	✓	-
	setEligibility	External	✓	-
	deposit	External	Payable	-
	process	External	✓	-
DiamondDistributor	Implementation	IDiamondDistributor, Context		
	<Constructor>	Public	✓	-
	unInitialized	External	✓	onlyToken
	setTokenAddress	External	✓	initializer onlyToken
	setDiamondCriteria	External	✓	onlyToken
	setEligibility	External	✓	onlyToken
	process	External	✓	onlyToken

	shouldDistribute	Internal		
	deposit	External	Payable	onlyToken
	distributeDiamond	Internal	✓	
	getCumulativeDiamonds	Internal	✓	
	getUnpaidEarnings	Internal	✓	
	addHolder	Internal	✓	
	removeHolder	Internal	✓	
Revoluzion	Implementation	IERC20Extended, Auth		
	<Constructor>	Public	Payable	Auth
	<Receive Ether>	External	Payable	-
	getCirculatingSupply	Public		-
	setDistributorSettings	External	✓	authorized
	totalSupply	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	balanceOf	Public		-
	allowance	External		-
	approve	Public	✓	-
	approveMax	External	✓	-
	transfer	External	✓	-
	transferFrom	External	✓	-
	_transferFrom	Internal	✓	
	_basicTransfer	Internal	✓	
	shouldTakeFee	Internal		
	shouldSwapBack	Internal		
	shouldAutoBuyback	Internal		
	isOverLiquified	Public		-
	_initializeFees	Internal	✓	
	setFees	Public	✓	authorized
	_setFees	Internal	✓	
	setFeeReceivers	External	✓	authorized
	setIsFeeExempt	External	✓	authorized
	getTotalFee	Public		-

	getMultipliedFee	Public		-
	takeFee	Internal	✓	
	_initializeDiamond	Internal	✓	
	diamondInitialization	Public	✓	authorized
	setDiamondDistributor	Public	✓	authorized
	setIsDiamondExempt	External	✓	authorized
	setDiamondCriteria	External	✓	authorized
	_initializeDistributor	Internal	✓	
	distributorInitialization	Public	✓	authorized
	setDividendDistributor	Public	✓	authorized
	setIsDividendExempt	External	✓	authorized
	setDistributionCriteria	External	✓	authorized
	claimDividend	External	✓	-
	_initializeLiquidityBuyBack	Internal	✓	
	getLiquidityBacking	Public		-
	setTargetLiquidity	External	✓	authorized
	setSwapBackSettings	External	✓	authorized
	swapBack	Internal	✓	swapping
	setAutoBuybackSettings	External	✓	authorized
	setBuybackMultiplierSettings	External	✓	authorized
	setBuyBacker	External	✓	authorized
	clearBuybackMultiplier	External	✓	authorized
	triggerAutoBuyback	Internal	✓	
	triggerZeusBuyback	External	✓	authorized
	buyTokens	Internal	✓	swapping

Contract Flow



Domain Info

Domain Name	revoluzion.io
Registry Domain ID	c91950bd4aee4d628de33f7a27365c0a-DONUTS
Creation Date	2021-12-16T17:18:18Z
Updated Date	2021-12-21T17:19:13Z
Registry Expiry Date	2022-12-16T17:18:18Z
Registrar WHOIS Server	whois.advancedregistrar.com
Registrar URL	http://www.netearthone.com
Registrar	NetEarth One Inc. dba NetEarth
Registrar IANA ID	1005

The domain has been created about 2 months before the creation of the audit. It will expire in 10 months.

There is no public billing information, the creator is protected by the privacy settings.

Summary

Revoluzion implements an interesting feature. It contains an additional dividend layer that rewards the users that hold many tokens. The Smart Contract analysis reported no compiler error or critical issues. The contract Owner can access some admin functions that can not be used in a malicious way to disturb the users' transactions. There is also a limit of max 15% fees.

The audit focuses on performance improvements, security notes and business logic concerns.

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.

Coinscope 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 Coinscope 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 Coinscope 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 Coinscope team disclaims any liability for the resulting losses.

About Coinscope

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.

Coinscope is aiming to make crypto discoverable and efficient globally. It provides all the essential tools to assist users draw their own conclusions.

The Coinscope.co team

https://www.coinscope.co