Cyberscope

Audit Report

HODLXRP

June 2022

SHA256 8213a3aa9eb9545fc3437f5e595ad6a0ec1e45fe856fbe2cce01ae90315e64cc

Audited by © cyberscope

Table of Contents

Table of Contents 1
Source Files 3
Audit Updates 3
Contract Analysis 4
BC - Blacklisted Contracts 5
Description 5
Recommendation 5
Contract Diagnostics 6
L01 - Public Function could be Declared External 7
Description 7
Recommendation 7
L04 - Conformance to Solidity Naming Conventions 8
Description 8
Recommendation 8
L07 - Missing Events Arithmetic 9
Description 9
Recommendation 9
L12 - Using Variables before Declaration 10
Description 10
Recommendation 10
L13 - Divide before Multiply Operation 11
Description 11
Recommendation 11
L14 - Uninitialized Variables in Local Scope 12
Description 12
Recommendation 12

Contract Functions 13
Contract Flow 17
Summary 18
Disclaimer 19
About Cyberscope 20

Source Files

Filename	SHA256
contract.sol	8213a3aa9eb9545fc3437f5e595ad6a0ec1e45fe856fbe2cce01ae90315e64cc

Audit Updates

Initial Audit	17th June 2022
Corrected	22nd June 2022

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

BC - Blacklisted Contracts

Criticality	medium
Location	contract.sol#L421

Description

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

```
    function setBlacklistEnabledMultiple(address[] memory accounts, bool
enabled) external onlyOwner {
        antiSnipe.setBlacklistEnabledMultiple(accounts, enabled);
        for(uint256 i = 0; i < accounts.length; i++){
            setDividendExcluded(accounts[i], enabled);
        }
    }
```

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
	L04	Conformance to Solidity Naming Conventions
	L07	Missing Events Arithmetic
	L12	Using Variables before Declaration
	L13	Divide before Multiply Operation
	L14	Uninitialized Variables in Local Scope

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L325,342

Description

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

```
setNewRouter
approveContractContingency
```

Recommendation

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

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L33,438,510,120,123,134,136,137,138,140,156,162,170,171,172,173,174,189,209

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.

```
_hasLiqBeenAdded
_taxWallets
masterTaxDivisor
maxRoundtripTax
maxTransferTaxes
maxSellTaxes
maxBuyTaxes
_ratios
_taxRates
...
```

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#L483,492,498,515

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.

```
reflectorGas = gas
piSwapPercent = priceImpactSwapPercent
swapThreshold = (_tTotal * thresholdPercent) / thresholdDivisor
_maxTxAmount = (_tTotal * percent) / divisor
```

Recommendation

Emit an event for critical parameter changes.

L12 - Using Variables before Declaration

Criticality	minor
Location	contract.sol#L604

Description

The contract is using a variable before the declaration. This is usually happening either if it has not been declared yet or the variable has been declared in a different scope.

```
check
```

Recommendation

The variables should be declared before any usage of them.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L667

Description

Performing divisions before multiplications may cause lose of prediction.

```
toLiquify = ((contractTokenBalance * ratios.liquidity) / (ratios.total)) / 2
```

Recommendation

The multiplications should be prior to the divisions.

L14 - Uninitialized Variables in Local Scope

Criticality	minor
Location	contract.sol#L603,604

Description

The are variables that are defined in the local scope and are not initialized.

```
check
checked
```

Recommendation

All the local scoped variables should be initialized.

Contract Functions

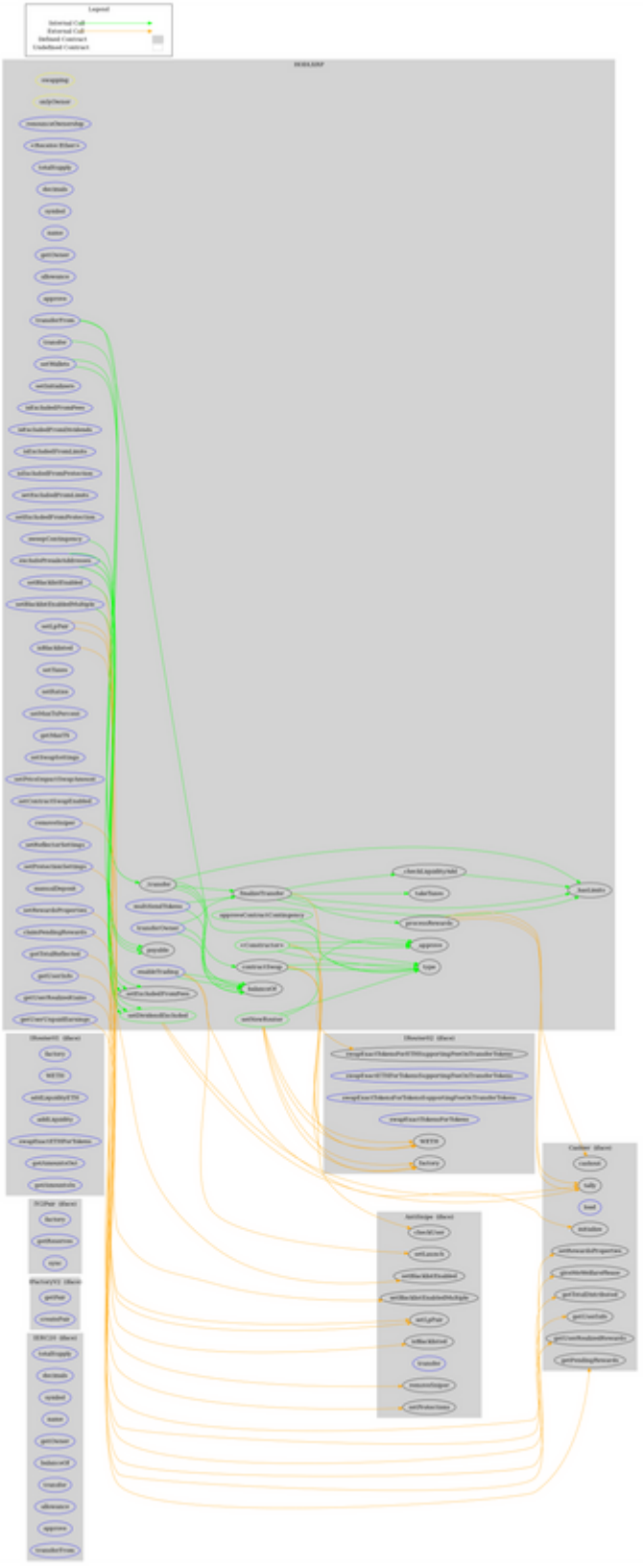
Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
IERC20	Interface			
	totalSupply	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	getOwner	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IFactoryV2	Interface			
	getPair	External		-
	createPair	External	✓	-
IV2Pair	Interface			
	factory	External		-
	getReserves	External		-
	sync	External	✓	-
IRouter01	Interface			
	factory	External		-
	WETH	External		-
	addLiquidityETH	External	Payable	-
	addLiquidity	External	✓	-
	swapExactETHForTokens	External	Payable	-
	getAmountsOut	External		-

	getAmountsIn	External		-
IRouter02	Interface	IRouter01		
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokens	External	✓	-
AntiSnipe	Interface			
	checkUser	External	✓	-
	setLaunch	External	✓	-
	setLpPair	External	✓	-
	setProtections	External	✓	-
	removeSniper	External	✓	-
	isBlacklisted	External		-
	transfer	External	✓	-
	setBlacklistEnabled	External	✓	-
	setBlacklistEnabledMultiple	External	✓	-
Cashier	Interface			
	setRewardsProperties	External	✓	-
	tally	External	✓	-
	load	External	Payable	-
	cashout	External	✓	-
	giveMeWelfarePlease	External	✓	-
	getTotalDistributed	External		-
	getUserInfo	External		-
	getUserRealizedRewards	External		-
	getPendingRewards	External		-
	initialize	External	✓	-
HODLXRP	Implementation	IERC20		
	<Constructor>	Public	Payable	-

	transferOwner	External	✓	onlyOwner
	renounceOwnership	External	✓	onlyOwner
	<Receive Ether>	External	Payable	-
	totalSupply	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	getOwner	External		-
	balanceOf	Public		-
	allowance	External		-
	approve	External	✓	-
	_approve	Private	✓	
	approveContractContingency	Public	✓	onlyOwner
	transfer	External	✓	-
	transferFrom	External	✓	-
	setNewRouter	Public	✓	onlyOwner
	setLpPair	External	✓	onlyOwner
	setInitializers	External	✓	onlyOwner
	isExcludedFromFees	External		-
	isExcludedFromDividends	External		-
	isExcludedFromLimits	External		-
	isExcludedFromProtection	External		-
	setExcludedFromLimits	External	✓	onlyOwner
	setDividendExcluded	Public	✓	onlyOwner
	setExcludedFromFees	Public	✓	onlyOwner
	setExcludedFromProtection	External	✓	onlyOwner
	setBlacklistEnabled	External	✓	onlyOwner
	setBlacklistEnabledMultiple	External	✓	onlyOwner
	isBlacklisted	External		-
	removeSniper	External	✓	onlyOwner
	setProtectionSettings	External	✓	onlyOwner
	enableTrading	External	✓	onlyOwner
	setWallets	External	✓	onlyOwner
	setTaxes	External	✓	onlyOwner
	setRatios	External	✓	onlyOwner

	setMaxTxPercent	External	✓	onlyOwner
	getMaxTX	External		-
	setSwapSettings	External	✓	onlyOwner
	setPriceImpactSwapAmount	External	✓	onlyOwner
	setContractSwapEnabled	External	✓	onlyOwner
	setRewardsProperties	External	✓	onlyOwner
	setReflectorSettings	External	✓	onlyOwner
	excludePresaleAddresses	External	✓	onlyOwner
	_hasLimits	Private		
	_transfer	Internal	✓	
	finalizeTransfer	Internal	✓	
	processRewards	Internal	✓	
	takeTaxes	Internal	✓	
	contractSwap	Internal	✓	swapping
	_checkLiquidityAdd	Private	✓	
	multiSendTokens	External	✓	onlyOwner
	manualDeposit	External	✓	onlyOwner
	sweepContingency	External	✓	onlyOwner
	claimPendingRewards	External	✓	-
	getTotalReflected	External		-
	getUserInfo	External		-
	getUserRealizedGains	External		-
	getUserUnpaidEarnings	External		-

Contract Flow



Summary

The Smart Contract analysis reported one medium severity issue. The contract owner has the authority to blacklist addresses. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats. There is also a limit of max 15% fees.

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