



Cyberscope

Audit Report

Opin

October 2022

Type BEP20

Network BSC

Address 0xAff60Ed78a7Ba9A6532466A372b546a6bdb885aB

Audited by © cyberscope

Table of Contents

Table of Contents	1
Contract Review	3
Source Files	3
Audit Updates	3
Contract Analysis	4
Contract Diagnostics	5
L01 - Public Function could be Declared External	6
Description	6
Recommendation	6
L04 - Conformance to Solidity Naming Conventions	7
Description	7
Recommendation	7
L07 - Missing Events Arithmetic	8
Description	8
Recommendation	8
L12 - Using Variables before Declaration	9
Description	9
Recommendation	9
L13 - Divide before Multiply Operation	10
Description	10
Recommendation	10
L14 - Uninitialized Variables in Local Scope	11
Description	11
Recommendation	11
Contract Functions	12
Contract Flow	16

Summary	17
Disclaimer	18
About Cyberscope	19

Contract Review

Contract Name	OPIN
Compiler Version	v0.8.17+commit.8df45f5f
Optimization	500 runs
Licence	MIT
Explorer	https://bscscan.com/token/0xAff60Ed78a7Ba9A6532466A372b546a6bdb885aB
Symbol	OPIN
Decimals	9
Total Supply	500,000,000
Domain	-

Source Files

Filename	SHA256
contract.sol	b91a016bd5a2448a275c1adf88e03e736d0d2576273b83372ebf5d39fbcdf086

Audit Updates

Initial Audit	7th October 2022 https://github.com/cyberscope-io/audits/blob/main/v1/opin/audit.pdf
Corrected phase 1	20th October 2022 https://github.com/cyberscope-io/audits/blob/main/v2/opin/audit.pdf
Corrected phase 2	24th October 2022

Contract Analysis

● Critical ● Medium ● Minor / Informative ● Pass

Severity	Code	Description	Status
●	ST	Stops Transactions	Passed
●	OCTD	Transfers Contract's Tokens	Passed
●	OTUT	Transfers User's Tokens	Passed
●	ELFM	Exceeds Fees Limit	Passed
●	ULTW	Transfers Liquidity to Team Wallet	Passed
●	MT	Mints Tokens	Passed
●	BT	Burns Tokens	Passed
●	BC	Blacklists Addresses	Passed

Contract Diagnostics

● Critical ● Medium ● Minor / Informative

Severity	Code	Description	Status
●	L01	Public Function could be Declared External	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved
●	L07	Missing Events Arithmetic	Unresolved
●	L12	Using Variables before Declaration	Unresolved
●	L13	Divide before Multiply Operation	Unresolved
●	L14	Uninitialized Variables in Local Scope	Unresolved

L01 - Public Function could be Declared External

Criticality	minor / informative
Location	contract.sol#L293,382,623,650
Status	Unresolved

Description

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

```
transfer
getCirculatingSupply
enableTrading
isExcludedFromReward
```

Recommendation

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

L04 - Conformance to Solidity Naming Conventions

Criticality	minor / informative
Location	contract.sol#L33,390,114,115,116,117,118,135,141,148,149,150,151,152,172
Status	Unresolved

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.

```
WETH
_antiSnipe
_antiBlock
startingSupply
_name
_symbol
_decimals
_tTotal
_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 / informative
Location	contract.sol#L425,432,449,458
Status	Unresolved

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.

```
_maxTxBuyAmount = (_tTotal * percentBuy) / divisorBuy  
_maxWalletSize = (_tTotal * percent) / divisor  
swapThreshold = (_tTotal * thresholdPercent) / thresholdDivisor  
piSwapPercent = priceImpactSwapPercent
```

Recommendation

Emit an event for critical parameter changes.

L12 - Using Variables before Declaration

Criticality	minor / informative
Location	contract.sol#L753
Status	Unresolved

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 / informative
Location	contract.sol#L559,740
Status	Unresolved

Description

Performing divisions before multiplications may cause lose of prediction.

```
toLiquify = ((contractTokenBalance * ratios.liquidity) / ratios.totalSwap) / 2  
feeAmount = (tAmount * currentFee) / masterTaxDivisor
```

Recommendation

The multiplications should be prior to the divisions.

L14 - Uninitialized Variables in Local Scope

Criticality	minor / informative
Location	contract.sol#L753,741,752
Status	Unresolved

Description

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

```
check  
values  
checked
```

Recommendation

All the local scoped variables should be initialized.

Contract Functions

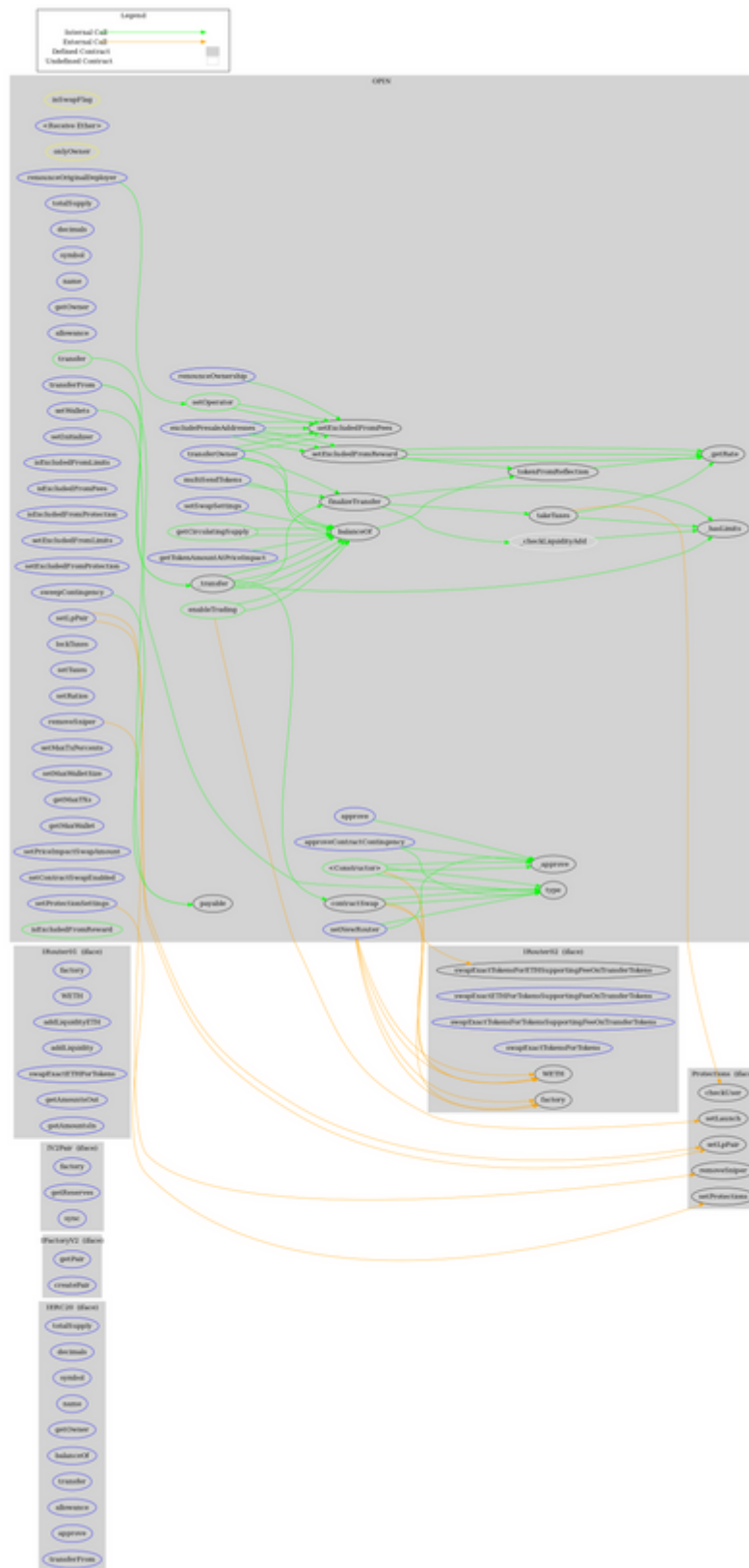
Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
IERC20	Interface			
	totalSupply	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	getOwner	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	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	✓	-
Protections	Interface			
	checkUser	External	✓	-
	setLaunch	External	✓	-
	setLpPair	External	✓	-
	setProtections	External	✓	-
	removeSniper	External	✓	-
OPIN	Implementation	IERC20		
	<Constructor>	Public	Payable	-
	<Receive Ether>	External	Payable	-
	transferOwner	External	✓	onlyOwner
	renounceOwnership	External	✓	onlyOwner
	setOperator	Public	✓	-
	renounceOriginalDeployer	External	✓	-
	totalSupply	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	getOwner	External		-
	allowance	External		-
	balanceOf	Public		-
	transfer	Public	✓	-
	approve	External	✓	-
	_approve	Internal	✓	
	approveContractContingency	External	✓	onlyOwner

	transferFrom	External	✓	-
	setNewRouter	External	✓	onlyOwner
	setLpPair	External	✓	onlyOwner
	setInitializer	External	✓	onlyOwner
	isExcludedFromLimits	External		-
	isExcludedFromFees	External		-
	isExcludedFromProtection	External		-
	setExcludedFromLimits	External	✓	onlyOwner
	setExcludedFromFees	Public	✓	onlyOwner
	setExcludedFromProtection	External	✓	onlyOwner
	getCirculatingSupply	Public		-
	removeSniper	External	✓	onlyOwner
	setProtectionSettings	External	✓	onlyOwner
	lockTaxes	External	✓	onlyOwner
	setTaxes	External	✓	onlyOwner
	setRatios	External	✓	onlyOwner
	setWallets	External	✓	onlyOwner
	setMaxTxPercents	External	✓	onlyOwner
	setMaxWalletSize	External	✓	onlyOwner
	getMaxTXs	External		-
	getMaxWallet	External		-
	getTokenAmountAtPriceImpact	External		-
	setSwapSettings	External	✓	onlyOwner
	setPriceImpactSwapAmount	External	✓	onlyOwner
	setContractSwapEnabled	External	✓	onlyOwner
	excludePresaleAddresses	External	✓	onlyOwner
	_hasLimits	Internal		
	_transfer	Internal	✓	
	contractSwap	Internal	✓	inSwapFlag
	_checkLiquidityAdd	Internal	✓	
	enableTrading	Public	✓	onlyOwner
	sweepContingency	External	✓	onlyOwner
	multiSendTokens	External	✓	onlyOwner
	isExcludedFromReward	Public		-

	setExcludedFromReward	Public	✓	onlyOwner
	tokenFromReflection	Public		-
	finalizeTransfer	Internal	✓	
	takeTaxes	Internal	✓	
	_getRate	Internal		

Contract Flow



Summary

Opin Token is an interesting project that has a friendly and growing community. 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.

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



The Cyberscope team

<https://www.cyberscope.io>