Cyberscope

Audit Report

OTO Protocol V2

April 2022

Type	ERC20
Network	AVAX
Address	0x0ac80e1753dea5e298e8a2b6cf53f161937806a1
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	OTOV2
Compiler Version	v0.7.6+commit.7338295f
Optimization	200 runs
Licence	Unlicense
Explorer	https://snowtrace.io/token/0x0ac80e1753dea5e298e8a2b6cf53f161937806a1
Symbol	OTO
Decimals	5
Total Supply	325,000
Domain	otoprotocol.info

Source Files

Filename	**SHA256**
contract.sol	6a6799935133b4adf5b486d314cae244828bf0a66a65876ebc1fa56249e682b9

Audit Updates

Initial Audit	7th 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	minor
Location	contract.sol#L581

Description

The minimum hold limit that can be set by the contract owner is 0.5%. If the contract owner removes the swap pair from the _excludeFromLimit then the users will be unable to sell their tokens. That will happen because the pair will probably hold more than 0.5% of the total supply. Since this is only checked for the recipient, it can cause the contract to operate as a honeypot.

```
modifier checkLimit(address from, address to, uint256 value) {
    if(!_excludeFromLimit[from]) {
        require(sold[from][getCurrentDay()] + value <= getUserSellLimit(from),
"Cannot sell or transfer more than limit.");
    }
    _;
    if(!_excludeFromLimit[to]) {
        require(_gonBalances[to].div(_gonsPerFragment) <= getUserHoldLimit(),
"Cannot buy more than limit.");
    }
}
```

Recommendation

The contract could explicite check if the recipient is the pair address. It could also apply the same check for the sender.

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#L893

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 calling the *withdrawAllToTreasury* method.

```
function withdrawAllToTreasury() external swapping onlyOwner {

    uint256 amountToSwap = _gonBalances[address(this)].div(_gonsPerFragment);
    require( amountToSwap > 0,"There is no OTO token deposited in token
contract");
    address[] memory path = new address[](2);
    path[0] = address(this);
    path[1] = router.WAVAX();
    router.swapExactTokensForAVAXSupportingFeeOnTransferTokens(
        amountToSwap,
        0,
        path,
        treasuryReceiver,
        block.timestamp
    );
}
```

Recommendation

The contract could embody a check for the maximum amount of funds that can be swapped. Since a huge amount may volatile 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	medium
Location	contract.sol#L734

Description

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

```
require(!blacklist[sender] && !blacklist[recipient], "blacklisted");
```

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
	MTS	Manipulate Total Supply
	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
	L13	Divide before Multiply Operation
	L14	Uninitialized Variables in Local Scope

MTS - Manipulate Total Supply

Criticality	minor
Location	contract.sol#L659

Description

Owner is able to manipulate total supply. This change will have a direct impact on the token price and Market Cap

```
for (uint256 i = 0; i < times; i++) {
    _totalSupply = _totalSupply
        .mul((10**RATE_DECIMALS).add(rebaseRate))
        .div(10**RATE_DECIMALS);
}
```

Recommendation

The contract owner should carefully manage the adjustment of the circulating supply (increases or decreases), according to the token’s price fluctuations.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L471,476,502,506,510,1072,1155

Description

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

```
rescueAVAX
getLiquidityBacking
decimals
symbol
name
transferOwnership
renounceOwnership
```

Recommendation

Use the external attribute for functions never called from the contract

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L557,558,522,555,552,553,566

Description

Constant state variables should be declared constant to save gas.

```
swapEnabled
maxAllowedSellFee
maxAllowedFee
feeDenominator
_decimals
ZERO
DEAD
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L153,155,181,225,942,951,1014,1034,1035,1036,1037,1038,1048,1049,1050,1051,1052,1053,1082,1086,1090,1095,1099,1111,1116,1129,1133,1137,522,525,537,538,539,540,541,542,557,558,571,601,602,606,607,608,609,610,611

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.

```
_totalSupply
_lastAddLiquidityTime
_lastRebasedTime
_initRebaseStartTime
_autoAddLiquidity
_autoRebase
MAX_HOLD_LIMIT
DAILY_SELL_LIMIT
_excludeFromLimit
...
```

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#L9

Description

There are segments that contain unused state variables.

```
MAX_INT256
```

Recommendation

Remove unused state variables.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L1047,1111,1116

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.

```
DAILY_SELL_LIMIT = _dailySellLimit
MAX_HOLD_LIMIT = _maxHoldLimit
liquidityFee = _liquidityFee
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L37

Description

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

```
abs
```

Recommendation

Remove unused functions.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L659,770,1072,1129

Description

Performing divisions before multiplications may cause lose of prediction.

```
_gonBalances[_who].div(_gonsPerFragment).mul(DAILY_SELL_LIMIT).div(LIMIT_DENOMIN
ATOR)
liquidityBalance = _gonBalances[pair].div(_gonsPerFragment)
_gonBalances[serviceFeeReceiver] =
_gonBalances[serviceFeeReceiver].add(gonAmount.div(feeDenominator).mul(serviceFe
e))
_gonBalances[autoLiquidityReceiver] =
_gonBalances[autoLiquidityReceiver].add(gonAmount.div(feeDenominator).mul(liquid
ityFee))
_gonBalances[address(this)] =
_gonBalances[address(this)].add(gonAmount.div(feeDenominator).mul(_treasuryFee.a
dd(otoVaultFundFee)))
_gonBalances[cauldron] =
_gonBalances[cauldron].add(gonAmount.div(feeDenominator).mul(cauldronFee))
feeAmount = gonAmount.div(feeDenominator).mul(_totalFee)
times = deltaTime.div(900)
```

Recommendation

The multiplications should be prior to the divisions.

L14 - Uninitialized Variables in Local Scope

Criticality	minor
Location	contract.sol#L662

Description

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

```
rebaseRate
```

Recommendation

All the local scoped variables should be initialized.

Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
SafeMathInt	Library			
	mul	Internal		
	div	Internal		
	sub	Internal		
	add	Internal		
	abs	Internal		
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	allowance	External		-
	transfer	External	✓	-
	approve	External	✓	-
	transferFrom	External	✓	-
IJoeSwapPair	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		-
	mint	External	✓	-
	burn	External	✓	-
	swap	External	✓	-
	skim	External	✓	-
	sync	External	✓	-
	initialize	External	✓	-
IJoeSwapRouter	Interface			
	factory	External		-
	WAVAX	External		-
	addLiquidity	External	✓	-
	addLiquidityAVAX	External	Payable	-
	removeLiquidity	External	✓	-
	removeLiquidityAVAX	External	✓	-
	removeLiquidityWithPermit	External	✓	-
	removeLiquidityAVAXWithPermit	External	✓	-
	swapExactTokensForTokens	External	✓	-
	swapTokensForExactTokens	External	✓	-

	swapExactAVAXForTokens	External	Payable	-
	swapTokensForExactAVAX	External	✓	-
	swapExactTokensForAVAX	External	✓	-
	swapAVAXForExactTokens	External	Payable	-
	quote	External		-
	getAmountOut	External		-
	getAmountIn	External		-
	getAmountsOut	External		-
	getAmountsIn	External		-
	removeLiquidityAVAXSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityAVAXWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactAVAXForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForAVAXSupportingFeeOnTransferTokens	External	✓	-
IJoeSwapFactory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	migrator	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
	setMigrator	External	✓	-
Ownable	Implementation			
	<Constructor>	Public	✓	-
	owner	Public		-
	isOwner	Public		-

	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
ERC20Detailed	Implementation	IERC20		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
OTOV2	Implementation	ERC20Detailed, Ownable		
	<Constructor>	Public	✓	ERC20Detailed Ownable
	rebase	Internal	✓	
	transfer	External	✓	validRecipient
	transferFrom	External	✓	validRecipient
	_basicTransfer	Internal	✓	
	_transferFrom	Internal	✓	checkLimit
	takeFee	Internal	✓	
	addLiquidity	Internal	✓	swapping
	swapBack	Internal	✓	swapping
	withdrawAllToTreasury	External	✓	swapping onlyOwner
	shouldTakeFee	Internal		
	shouldRebase	Internal		
	shouldAddLiquidity	Internal		
	shouldSwapBack	Internal		
	setAutoRebase	External	✓	onlyOwner
	setAutoAddLiquidity	External	✓	onlyOwner
	allowance	External		-
	decreaseAllowance	External	✓	-
	increaseAllowance	External	✓	-
	approve	External	✓	-
	checkFeeExempt	External		-
	getCirculatingSupply	Public		-

	isNotInSwap	External		-
	manualSync	External	✓	-
	setFeeReceivers	External	✓	onlyOwner
	setFees	External	✓	onlyOwner
	getTotalFee	Public		-
	getLiquidityBacking	Public		-
	setFeeExempt	External	✓	onlyOwner
	removeFeeExempt	External	✓	onlyOwner
	setBotBlacklist	External	✓	onlyOwner
	setBlacklist	External	✓	onlyOwner
	setLP	External	✓	onlyOwner
	totalSupply	External		-
	balanceOf	External		-
	setMaxHoldLimit	External	✓	onlyOwner
	setDailySellLimit	External	✓	onlyOwner
	getCurrentDay	Public		-
	getUserHoldLimit	Public		-
	getUserSellLimit	Public		-
	excludeFromLimit	External	✓	onlyOwner
	removeFromLimitExclusion	External	✓	onlyOwner
	isContract	Internal		
	minZero	Private		
	rescueAVAX	Public	✓	onlyOwner
	restoreInitTax	External	✓	onlyOwner
	<Receive Ether>	External	Payable	-

Contract Flow

Domain Info

Domain Name	otoprotocol.info
Registry Domain ID	c3ae8b825c1a4208a0a684b877b5965a-DONUTS
Creation Date	2022-03-09T21:35:04Z
Updated Date	2022-03-14T21:35:19Z
Registry Expiry Date	2023-03-09T21:35:04Z
Registrar WHOIS Server	whois.namecheap.com
Registrar URL	https://www.namecheap.com/
Registrar	NameCheap, Inc.
Registrar IANA ID	1068

The domain has been created 29 days 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

There are some functions that can be abused by the owner, like blacklisting addresses, transferring funds to the team’s wallet or limiting the sales. The maximum fee percentage that can be set is 25% for buys and 28 for sales. 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