Cyberscope

Audit Report
WATCHN

June 2022

Table of Contents

Contract Review

Contract Name	WTE
Compiler Version	v0.8.10+commit.fc410830
Optimization	200 runs
Licence	None
Explorer	https://bscscan.com/token/0x3a64d5bc87ef25004aa07831ed4a7ee985e1831f
Symbol	WTE
Decimals	18
Total Supply	1,000,000,000
Domain	https://watchn.finance/

Source Files

Filename	**SHA256**
contract.sol	6fd3aab11c76d90e18065122351fa2959ac396aeecf0f5e254001a2900284bde

Audit Updates

Initial Audit	28th June 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

Contract Diagnostics

● Critical   ● Medium   ● Minor

Severity	Code	Description
●	BLC	Business Logic Concern
●	FSA	Fixed Swap Address
●	CO	Code Optimization
●	L01	Public Function could be Declared External
●	L02	State Variables could be Declared Constant
●	L04	Conformance to Solidity Naming Conventions
●	L09	Dead Code Elimination

BLC - Business Logic Concern

Criticality	minor
Location	contract.sol#L1012,L1030

Description

The business logic seems peculiar. The implementation may not follow the expected behavior. The arguments on the _getValue function are called with the wrong order.As a result the tax fee is used as a liquidity fee and vice versa.

```
(uint256 tTransferAmount, uint256 tFee, uint256 tTeam) = _getTValues(
    tAmount,
    _taxFee,
    _liquidityFee
);

function _getTValues(
    uint256 tAmount,
    uint256 liquidityFee,
    uint256 teamFee
)
    private
    pure
    returns (
        uint256,
        uint256,
        uint256
    )
{
```

Recommendation

The team is advised to carefully check if the implementation follows the expected business logic.

FSA - Fixed Swap Address

Criticality	minor
Location	contract.sol#L672

Description

The swap address is assigned once in the constructor and it can not be changed. The decentralized swaps sometimes create a new swap version or abandon the current. A contract that cannot change the swap address may not be able to catch-up the upgrade.

```
constructor() {
    _rOwned[_msgSender()] = _rTotal;
    IUniswapV2Router02 _uniswapV2Router = IUniswapV2Router02(
        0x10ED43C718714eb63d5aA57B78B54704E256024E
    );
    uniswapV2Router = _uniswapV2Router;
    uniswapV2Pair = IUniswapV2Factory(_uniswapV2Router.factory())
        .createPair(address(this), _uniswapV2Router.WETH());
```

Recommendation

It could be better to allow the swap address mutation in case of future swap updates.

CO - Code Optimization

Criticality	minor
Location	contract.sol#L626

Description

There are code segments that could be optimized. A segment may be optimized so that it becomes a smaller size, consumes less memory, executes more rapidly, or performs fewer operations.

The variable _taxFee can be removed from the contract and its mechanisms as it is always zeros. As a result less gas fees will be produced.

```
uint256 private _taxFee = 0;
```

Recommendation

Rewrite some code segments so the runtime will be more performant.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L1090,1097,797,533,772,517,741,753

Description

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

```
allowance
transfer
acceptOwnership
approve
transferOwnership
transferFrom
excludeFromFees
toggleSwap
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L630,631,615,616,627,614,625

Description

Constant state variables should be declared constant to save gas.

```
_marketingAddress
_name
_autoLiquidityReceiver
_decimals
_symbol
_systemDivFee
_marketingDivFee
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L569,1090

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.

```
_feeSwap
WETH
```

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#L317,381,401,414,288,362,349

Description

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

```
functionCall
isContract
_functionCallWithValue
functionCallWithValue
sendValue
```

## Recommendation

Remove unused functions.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IERC20Metad ata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		
Address	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	

	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	_functionCallWithValue	Private	✓	
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	acceptOwnership	Public	✓	-
	transferOwnership	Public	✓	onlyOwner
Token	Interface			
	transferFrom	External	✓	-
	transfer	External	✓	-
IUniswapV2Factory	Interface			
	createPair	External	✓	-
IUniswapV2Router02	Interface			
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
	factory	External		-
	WETH	External		-
	addLiquidityETH	External	Payable	-
WTE	Implementation	Context, IERC20, Ownable, IERC20Metadata		
	<Constructor>	Public	✓	-
	decimals	External		-

symbol	External		-
name	External		-
totalSupply	External		-
balanceOf	Public		-
transfer	Public	✓	-
allowance	Public		-
approve	Public	✓	-
transferFrom	Public	✓	-
tokenFromReflection	Private		
_approve	Private	✓	
_transfer	Private	✓	
swapTokensForEth	Private	✓	lockTheSwap
sendETHToFee	Private	✓	
_tokenTransfer	Private	✓	
_transferStandard	Private	✓	
_takeTeam	Private	✓	
_reflectFee	Private	✓	
<Receive Ether>	External	Payable	-
_getValues	Private		
_getTValues	Private		
_getRValues	Private		
_getRate	Private		
_getCurrentSupply	Private		
toggleSwap	Public	✓	onlyDev
excludeFromFees	Public	✓	onlyDev
burn	External	✓	-
_burn	Internal	✓	
burnFrom	External	✓	-

Contract Flow
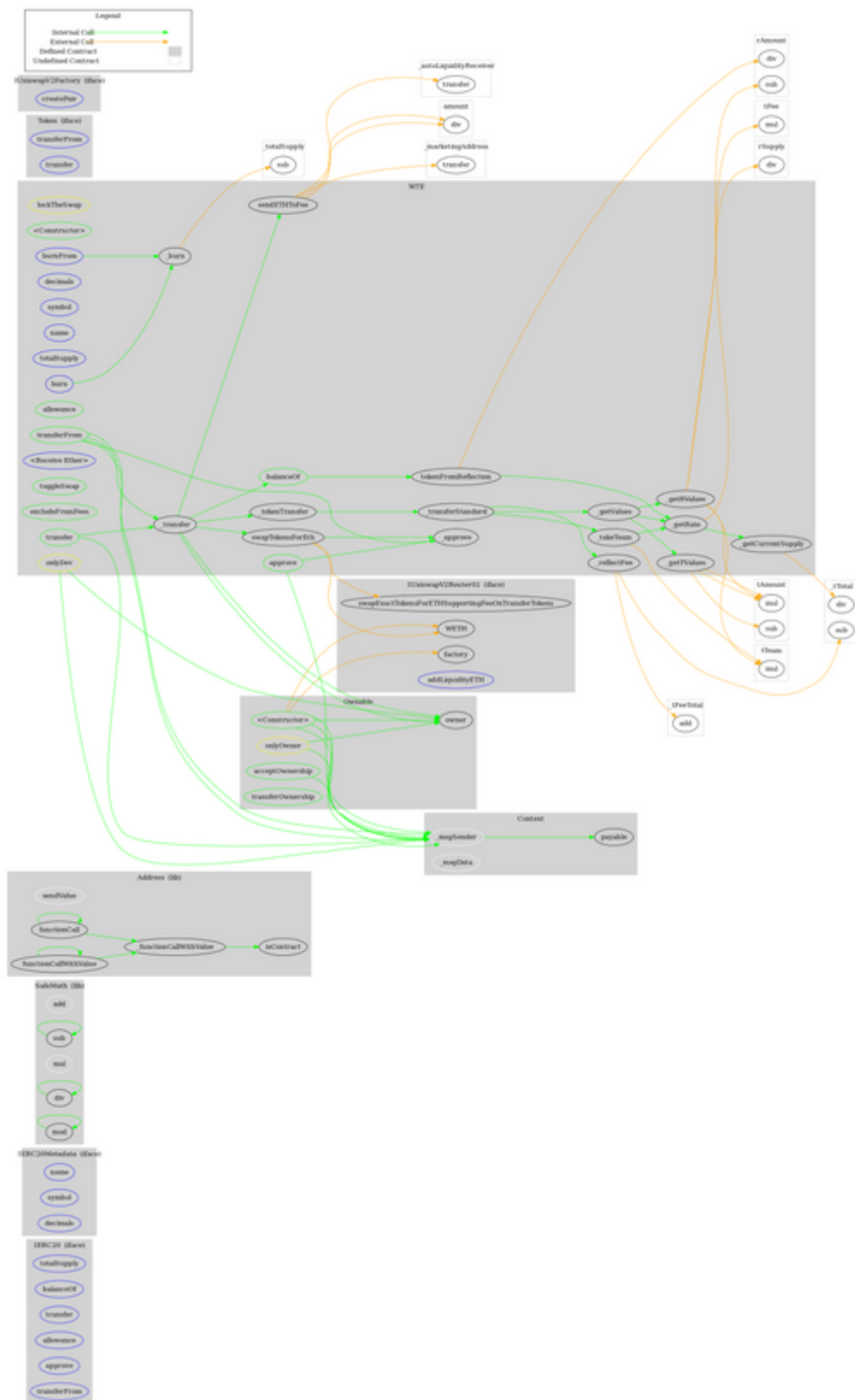
Domain Info

Domain Name	watchn.finance
Registry Domain ID	f58baefe2f5b4a3c82dda8ab34b66c5e-DONUTS
Creation Date	2022-05-26T13:39:53Z
Updated Date	2022-06-27T04:31:23Z
Registry Expiry Date	2023-05-26T13:39:53Z
Registrar WHOIS Server	whois.godaddy.com/
Registrar URL	http://www.godaddy.com/domains/search.aspx?ci=8990
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

WATCHN 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 5% 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