



Cyberscope

# Audit Report

## **Washer**

November 2023

Network    ETH

Address    0x5f4fcdf64c7d990f5249176e0048e36e9ba5e687

Audited by    © cyberscope

# Analysis

● Critical ● Medium ● Minor / Informative ● Pass

Severity	Code	Description	Status
●	ST	Stops Transactions	Passed
●	OTUT	Transfers User's Tokens	Passed
●	ELFM	Exceeds Fees Limit	Passed
●	MT	Mints Tokens	Passed
●	BT	Burns Tokens	Passed
●	BC	Blacklists Addresses	Passed

# Diagnostics

● Critical ● Medium ● Minor / Informative

Severity	Code	Description	Status
●	MEE	Missing Events Emission	Unresolved
●	RSML	Redundant SafeMath Library	Unresolved
●	MEM	Misleading Error Messages	Unresolved
●	IDI	Immutable Declaration Improvement	Unresolved
●	L02	State Variables could be Declared Constant	Unresolved
●	L17	Usage of Solidity Assembly	Unresolved
●	L20	Succeeded Transfer Check	Unresolved

# Table of Contents

<b>Analysis</b>	<b>1</b>
<b>Diagnostics</b>	<b>2</b>
<b>Table of Contents</b>	<b>3</b>
<b>Review</b>	<b>4</b>
Audit Updates	4
Source Files	4
<b>Findings Breakdown</b>	<b>5</b>
MRM - Missing Revert Messages	6
Description	6
Recommendation	6
MEE - Missing Events Emission	7
Description	7
Recommendation	7
RSM - Redundant SafeMath Library	8
Description	8
Recommendation	8
IDI - Immutable Declaration Improvement	9
Description	9
Recommendation	9
L02 - State Variables could be Declared Constant	10
Description	10
Recommendation	10
L17 - Usage of Solidity Assembly	11
Description	11
Recommendation	11
L20 - Succeeded Transfer Check	12
Description	12
Recommendation	12
<b>Functions Analysis</b>	<b>13</b>
<b>Inheritance Graph</b>	<b>16</b>
<b>Flow Graph</b>	<b>17</b>
<b>Summary</b>	<b>18</b>
<b>Disclaimer</b>	<b>19</b>
<b>About Cyberscope</b>	<b>20</b>

## Review

Contract Name	washer
Compiler Version	v0.8.20+commit.a1b79de6
Optimization	200 runs
Explorer	<a href="https://etherscan.io/address/0x5f4cdf64c7d990f5249176e0048e36e9ba5e687">https://etherscan.io/address/0x5f4cdf64c7d990f5249176e0048e36e9ba5e687</a>
Address	0x5f4cdf64c7d990f5249176e0048e36e9ba5e687
Network	ETH
Symbol	WASH
Decimals	9
Total Supply	100,000,000

## Audit Updates

Initial Audit	22 Nov 2023
---------------	-------------

## Source Files

Filename	SHA256
washer.sol	7ced28447b71b8e4372ea210c6a49f164960804f169c3e0ab76e63e64e8ed335

## Findings Breakdown



● Critical	0
● Medium	0
● Minor / Informative	7

Severity	Unresolved	Acknowledged	Resolved	Other
● Critical	0	0	0	0
● Medium	0	0	0	0
● Minor / Informative	7	0	0	0

## MRM - Missing Revert Messages

<b>Criticality</b>	Minor / Informative
<b>Location</b>	washer.sol#L233,293,298,303
<b>Status</b>	Unresolved

### Description

The contract is missing error messages. These missing error messages are making it difficult to identify and fix the issue. As a result, the user will not be able to find the root cause of the error.

```
require(!isContract(to))  
require(_msgSender() == _taxWallet)  
require(_msgSender() == _taxWallet)
```

### Recommendation

The team is advised to carefully review the source code in order to address these issues. To accelerate the debugging process and mitigate these issues, the team should use more specific and descriptive error messages.

## MEE - Missing Events Emission

<b>Criticality</b>	Minor / Informative
<b>Location</b>	washer.sol#L216
<b>Status</b>	Unresolved

### Description

The contract performs actions and state mutations from external methods that do not result in the emission of events. Emitting events for significant actions is important as it allows external parties, such as wallets or dApps, to track and monitor the activity on the contract. Without these events, it may be difficult for external parties to accurately determine the current state of the contract.

```
function setMarketPair(address addr) public onlyOwner {  
    marketPair[addr] = true;  
}
```

### Recommendation

It is recommended to include events in the code that are triggered each time a significant action is taking place within the contract. These events should include relevant details such as the user's address and the nature of the action taken. By doing so, the contract will be more transparent and easily auditable by external parties. It will also help prevent potential issues or disputes that may arise in the future.



## RSML - Redundant SafeMath Library

Criticality	Minor / Informative
Location	washer.sol
Status	Unresolved

### Description

SafeMath is a popular Solidity library that provides a set of functions for performing common arithmetic operations in a way that is resistant to integer overflows and underflows.

Starting with Solidity versions that are greater than or equal to 0.8.0, the arithmetic operations revert to underflow and overflow. As a result, the native functionality of the Solidity operations replaces the SafeMath library. Hence, the usage of the SafeMath library adds complexity, overhead and increases gas consumption unnecessarily.

```
library SafeMath {...}
```

### Recommendation

The team is advised to remove the SafeMath library. Since the version of the contract is greater than `0.8.0` then the pure Solidity arithmetic operations produce the same result.

If the previous functionality is required, then the contract could exploit the `unchecked { ... }` statement.

Read more about the breaking change on

<https://docs.soliditylang.org/en/v0.8.16/080-breaking-changes.html#solidity-v0-8-0-breaking-changes>.

## IDI - Immutable Declaration Improvement

<b>Criticality</b>	Minor / Informative
<b>Location</b>	washer.sol#L160
<b>Status</b>	Unresolved

### Description

The contract declares state variables that their value is initialized once in the constructor and are not modified afterwards. The `immutable` is a special declaration for this kind of state variables that saves gas when it is defined.

```
_taxWallet
```

### Recommendation

By declaring a variable as immutable, the Solidity compiler is able to make certain optimizations. This can reduce the amount of storage and computation required by the contract, and make it more gas-efficient.

## L02 - State Variables could be Declared Constant

Criticality	Minor / Informative
Location	washer.sol#L127,128,129,130,131,132,133,142,143
Status	Unresolved

### Description

State variables can be declared as constant using the constant keyword. This means that the value of the state variable cannot be changed after it has been set. Additionally, the constant variables decrease gas consumption of the corresponding transaction.

```
uint256 private _initialBuyTax=25
uint256 private _initialSellTax=25
uint256 private _finalBuyTax=5
uint256 private _finalSellTax=5
uint256 private _reduceBuyTaxAt=25
uint256 private _reduceSellTaxAt=30
uint256 private _preventSwapBefore=29
uint256 public _taxSwapThreshold= 1000000 * 10**_decimals
uint256 public _maxTaxSwap= 1300000 * 10**_decimals
```

### Recommendation

Constant state variables can be useful when the contract wants to ensure that the value of a state variable cannot be changed by any function in the contract. This can be useful for storing values that are important to the contract's behavior, such as the contract's address or the maximum number of times a certain function can be called. The team is advised to add the constant keyword to state variables that never change.

## L17 - Usage of Solidity Assembly

<b>Criticality</b>	Minor / Informative
<b>Location</b>	washer.sol#L272
<b>Status</b>	Unresolved

### Description

Using assembly can be useful for optimizing code, but it can also be error-prone. It's important to carefully test and debug assembly code to ensure that it is correct and does not contain any errors.

Some common types of errors that can occur when using assembly in Solidity include Syntax, Type, Out-of-bounds, Stack, and Revert.

```
assembly {  
    size := extcodesize(account)  
}
```

### Recommendation

It is recommended to use assembly sparingly and only when necessary, as it can be difficult to read and understand compared to Solidity code.

## L20 - Succeeded Transfer Check

<b>Criticality</b>	Minor / Informative
<b>Location</b>	washer.sol#L299
<b>Status</b>	Unresolved

### Description

According to the ERC20 specification, the transfer methods should be checked if the result is successful. Otherwise, the contract may wrongly assume that the transfer has been established.

```
IERC20(_tokenAddr).transfer(_to, _amount)
```

### Recommendation

The contract should check if the result of the transfer methods is successful. The team is advised to check the SafeERC20 library from the [Openzeppelin library](#).

## Functions Analysis

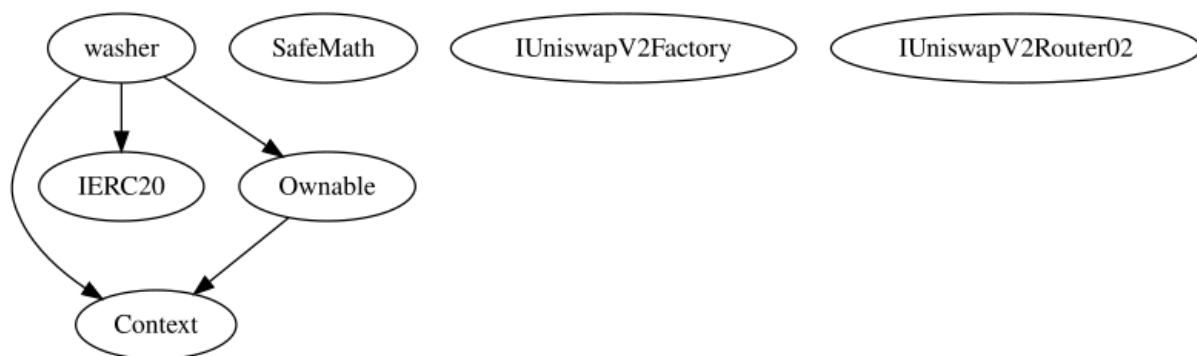
Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
<b>Context</b>	Implementation			
	_msgSender	Internal		
<b>IERC20</b>	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
<b>SafeMath</b>	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		

<b>Ownable</b>	Implementation	Context		
		Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
<b>IUniswapV2Factory</b>	Interface			
	createPair	External	✓	-
<b>IUniswapV2Router02</b>	Interface			
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
	factory	External		-
	WETH	External		-
	addLiquidityETH	External	Payable	-
<b>washer</b>	Implementation	Context, IERC20, Ownable		
		Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-

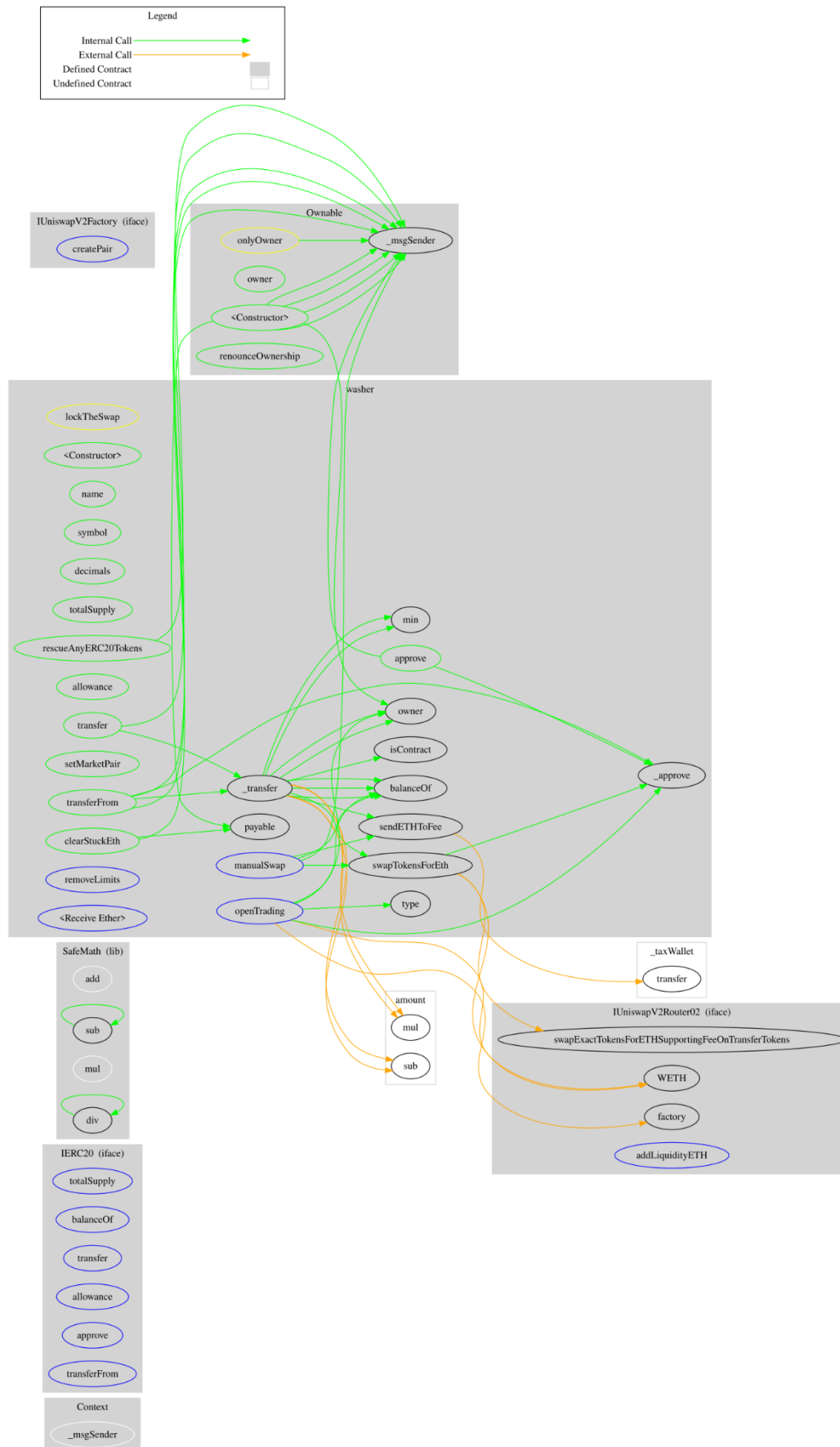
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	_approve	Private	✓	
	setMarketPair	Public	✓	onlyOwner
	_transfer	Private	✓	
	min	Private		
	isContract	Private		
	swapTokensForEth	Private	✓	lockTheSwap
	clearStuckEth	Public	✓	-
	rescueAnyERC20Tokens	Public	✓	-
	manualSwap	External	✓	-
	removeLimits	External	✓	onlyOwner
	sendETHToFee	Private	✓	
	openTrading	External	✓	onlyOwner
		External	Payable	-



## Inheritance Graph



# Flow Graph



## Summary

Washer contract implements a token mechanism. This audit investigates security issues, business logic concerns and potential improvements. Washer is an interesting project that has a friendly and growing community. The Smart Contract analysis reported no compiler errors 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.

The contract's ownership has been renounced. The information regarding the transaction can be accessed through the following link:

<https://etherscan.io/tx/0xde1b6523c504db3e7439951092c0c5d839b72a81f5839ddb648ed0e4ca216236>

The fees are locked at 5% on both buy and sell transactions.

## Disclaimer

The information provided in this report does not constitute investment, financial or trading advice and you should not treat any of the document's content as such. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes nor may copies be delivered to any other person other than the Company without Cyberscope's prior written consent. This report is not nor should be considered an "endorsement" or "disapproval" of any particular project or team. This report is not nor should be regarded as an indication of the economics or value of any "product" or "asset" created by any team or project that contracts Cyberscope to perform a security assessment. This document does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors' business, business model or legal compliance. This report should not be used in any way to make decisions around investment or involvement with any particular project. This report represents an extensive assessment process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk. Cyberscope's position is that each company and individual are responsible for their own due diligence and continuous security. Cyberscope's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies and in no way claims any guarantee of security or functionality of the technology we agree to analyze. The assessment services provided by Cyberscope are subject to dependencies and are under continuing development. You agree that your access and/or use including but not limited to any services reports and materials will be at your sole risk on an as-is where-is and as-available basis. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives and other unpredictable results. The services may access and depend upon multiple layers of third parties.

# About Cyberscope

Cyberscope is a blockchain cybersecurity company that was founded with the vision to make web3.0 a safer place for investors and developers. Since its launch, it has worked with thousands of projects and is estimated to have secured tens of millions of investors' funds.

Cyberscope is one of the leading smart contract audit firms in the crypto space and has built a high-profile network of clients and partners.



**The Cyberscope team**

<https://www.cyberscope.io>