Cyberscope

Audit Report
GirlesToken

September 2022

Type	ERC20
Network	ETH
Address	0xBd5608Fe87816273307f9769a16089A41Bc6F3b2
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	GirlesToken
Compiler Version	v0.8.15+commit.e14f2714
Optimization	200 runs
Licence	None
Explorer	https://etherscan.io/token/0xBd5608Fe87816273307f9769a16089A41Bc6F3b2
Symbol	GIRLES
Decimals	6
Total Supply	10,000,000,000

Source Files

Filename	**SHA256**
contract.sol	3d3b2a941198292db313a6ab70af79f05f8e1e007414756605b44343f2742584

Audit Updates

Initial Audit	22nd September 2022
Corrected	

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	Unresolved

BC - Blacklists Addresses

Criticality	medium
Location	contract.sol#L953,969
Status	Unresolved

Description

The contract owner has the authority to stop addresses from transactions. The owner may take advantage of it by calling the blacklistAddress function.

```
    function blacklistAddress(address account, bool value) external onlyOwner {
        _isBlacklisted[account] = value;
    }

    function _transfer(address sender, address recipient, uint256 amount)
        internal
        virtual
        override
        antiWhale(sender, recipient, amount) {
        if (recipient == BURN_ADDRESS || transferTaxRate == 0) {
            super._transfer(sender, recipient, amount);
        }
        else {
            require(sender != address(0), "Girles : transfer from the zero address");
            require(recipient != address(0), "Girles : transfer to the zero address");
            require(!_isBlacklisted[sender] && !_isBlacklisted[recipient], 'Girles : Blacklisted
address');
```

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 / Informative

Severity	Code	Description	Status
🟠	ZD	Zero Division	Unresolved
⚪	L01	Public Function could be Declared External	Unresolved
⚪	L02	State Variables could be Declared Constant	Unresolved
⚪	L04	Conformance to Solidity Naming Conventions	Unresolved
⚪	L07	Missing Events Arithmetic	Unresolved
⚪	L09	Dead Code Elimination	Unresolved
⚪	L11	Unnecessary Boolean equality	Unresolved

ZD - Zero Division

Criticality	medium
Location	contract.sol#L989
Status	Unresolved

Description

The contract is using variables that may be set to zero as denominators. As a result, the transactions will revert.

```
uint256 totalFeeRate    = burnRate + liquidityRate + marketingRate;
uint256 burnAmount      = contractTokenBalance * burnRate / totalFeeRate;
uint256 liquidityAmount = contractTokenBalance * liquidityRate / totalFeeRate;
uint256 marketingAmount = contractTokenBalance - burnAmount - liquidityAmount;
```

Recommendation

The contract should prevent those variables to be set to zero or should not allow to execute the corresponding statements.

L01 - Public Function could be Declared External

Criticality	minor / informative
Location	contract.sol#L367,629,1105,945,903,410,937,279,322,390,1023,271,1113,296,345,614
Status	Unresolved

Description

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

```
transferFrom
burnFrom
isExcludedFromAntiWhale
isExcludedFromFees
updateUniswapRouter
decreaseAllowance
excludeMultipleAccountsFromFees
symbol
transfer
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor / informative
Location	contract.sol#L849,820,826,823,822,819
Status	Unresolved

Description

Constant state variables should be declared constant to save gas.

```
maxTransferAmountRate
_symbol
transferTaxRate
_totalSupply
_decimals
_name
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor / informative
Location	contract.sol#L1105,1113,903,853,1023,657
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.

```
_account
_excluded
_router
_isBlacklisted
_amount
WETH
```

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#L926,922,1023,918
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.

```
marketingRate = value
liquidityRate = value
swapAndSendTokensAtAmount = _amount
burnRate = value
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor / informative
Location	contract.sol#L87
Status	Unresolved

Description

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

```
_transferOwnership
```

Recommendation

Remove unused functions.

L11 - Unnecessary Boolean equality

Criticality	minor / informative
Location	contract.sol#L865,954
Status	Unresolved

Description

The comparison to boolean constants is redundant. Boolean constants can be used directly and do not need to be compared to true or false.

```
excludedFromAntiWhale[sender] == false && excludedFromAntiWhale[recipient] == false
canSwapAndSendFee == true && ! swapping && sender != uniswapPair && recipient !=
uniswapPair && sender != owner()
```

Recommendation

Remove the equality to the boolean constant.

Contract Functions

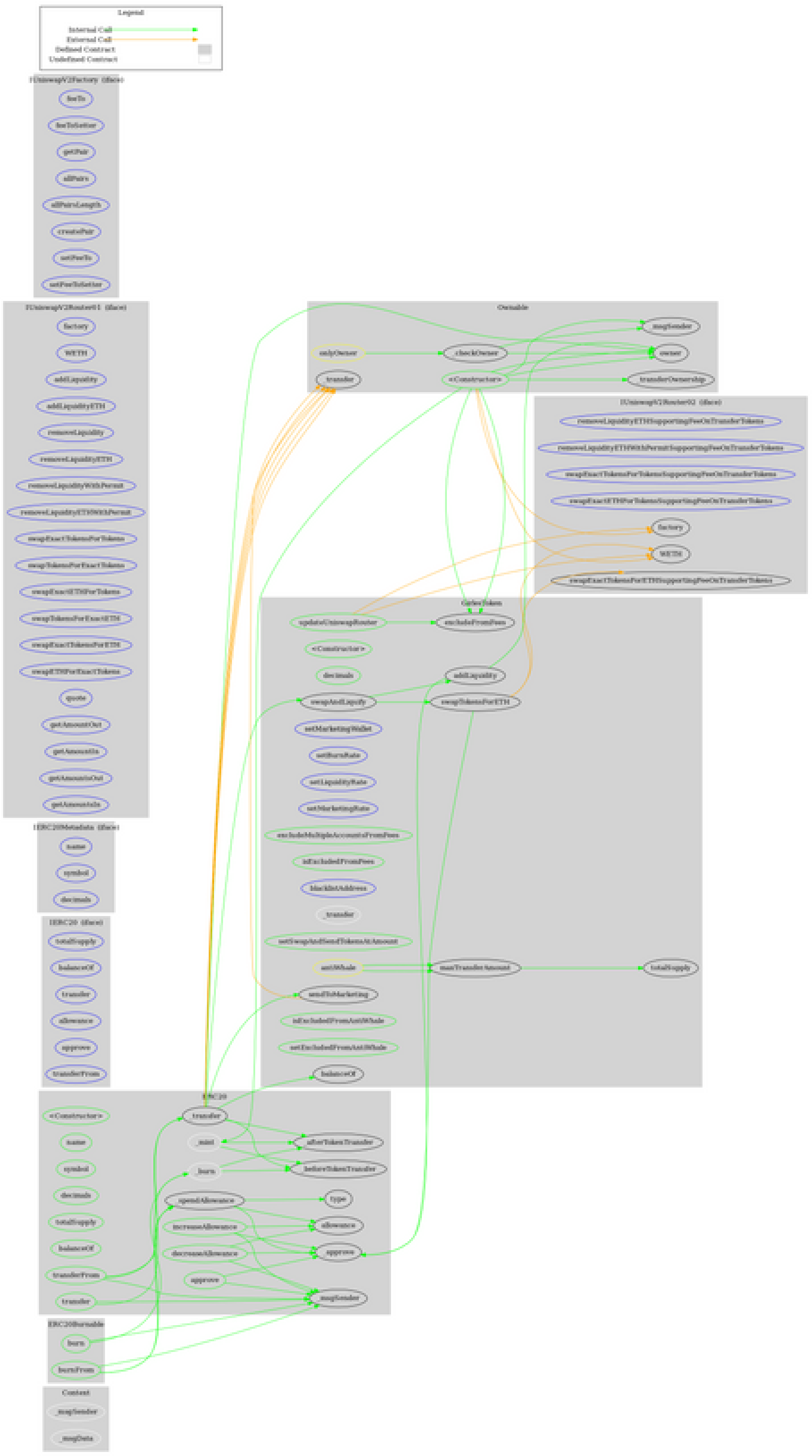
Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	_checkOwner	Internal		
	_transferOwnership	Internal	✓	
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
ERC20	Implementation	Context, IERC20, IERC20Metadata		
	<Constructor>	Public	✓	-
	name	Public		-

	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_spendAllowance	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
	_afterTokenTransfer	Internal	✓	
ERC20Burnable	Implementation	Context, ERC20		
	burn	Public	✓	-
	burnFrom	Public	✓	-
IUniswapV2Router01	Interface			
	factory	External		-
	WETH	External		-
	addLiquidity	External	✓	-
	addLiquidityETH	External	Payable	-
	removeLiquidity	External	✓	-
	removeLiquidityETH	External	✓	-
	removeLiquidityWithPermit	External	✓	-
	removeLiquidityETHWithPermit	External	✓	-
	swapExactTokensForTokens	External	✓	-
	swapTokensForExactTokens	External	✓	-

	swapExactETHForTokens	External	Payable	-
	swapTokensForExactETH	External	✓	-
	swapExactTokensForETH	External	✓	-
	swapETHForExactTokens	External	Payable	-
	quote	External		-
	getAmountOut	External		-
	getAmountIn	External		-
	getAmountsOut	External		-
	getAmountsIn	External		-
IUniswapV2Router02	Interface	IUniswapV2Router01		
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
GirlesToken	Implementation	ERC20, Ownable		
	<Constructor>	Public	✓	ERC20
	decimals	Public		-

	updateUniswapRouter	Public	✓	onlyOwner
	setMarketingWallet	External	✓	onlyOwner
	setBurnRate	External	✓	onlyOwner
	setLiquidityRate	External	✓	onlyOwner
	setMarketingRate	External	✓	onlyOwner
	excludeFromFees	Public	✓	onlyOwner
	excludeMultipleAccountsFromFees	Public	✓	onlyOwner
	isExcludedFromFees	Public		-
	blacklistAddress	External	✓	onlyOwner
	_transfer	Internal	✓	antiWhale
	setSwapAndSendTokensAtAmount	Public	✓	onlyOwner
	swapAndLiquify	Private	✓	
	swapTokensForETH	Private	✓	
	addLiquidity	Private	✓	
	sendToMarketing	Private	✓	
	maxTransferAmount	Public		-
	isExcludedFromAntiWhale	Public		-
	setExcludedFromAntiWhale	Public	✓	onlyOwner

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 fixed fee of 10%.

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