Cyberscope

Audit Report

ElonGoatToken

August 2022

SHA256 b60e81049de188fafe2b061eedc13db0b82d6f20a43e8eb899516401d776f1d1

Audited by © cyberscope

Table of Contents

Contract Review

Contract Name	EGT
Symbol	EGT
Decimals	9
Total Supply	9,000,000,000

Audit Updates

Initial Audit	10th August 2022
Corrected	

Source Files

Filename	**SHA256**
EGT.sol	b60e81049de188fafe2b061eedc13db0b82d6f20a43e8eb899516401d776f1d1
Utils.sol	9dc9f671e853cfd49be5720faf4de2b169ba55e60358a2c0613522a3028b735d

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	medium
Location	contract.sol#L331

Description

The contract owner has the authority to stop transactions for all users including the owner. The owner may take advantage of it by setting the acceptSlippageReduceFactor to a high value. As a result, the ‘swapExactTokensForETHSupportingFeeOnTransferTokens’ will revert.

```
swapTokensForEth(
    tokensToSwap,
    outAmount.mul(acceptSlippageReduceFactor).div(10)
);
```

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.

ULTW - Unlimited Liquidity to Team Wallet

Criticality	minor
Location	contract.sol#L458

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 sendShares methods.

```
function sendShares() external onlyRole(ADMIN_AUTH) {
        _sendShares();
    }
```

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.

Contract Diagnostics

● Critical ● Medium ● Minor

Severity	Code	Description
●	SU	Subtraction Underflow
●	CR	Code Repetition
●	L01	Public Function could be Declared External
●	L03	Redundant Statements
●	L04	Conformance to Solidity Naming Conventions
●	L11	Unnecessary Boolean equality
●	L12	Using Variables before Declaration
●	L14	Uninitialized Variables in Local Scope

SU - Subtraction Underflow

Criticality	critical
Location	contract.sol#L289,349

Description

The contract contains subtraction expressions where the minuend is not chekced if it is greater than subtrahend. As a result, the contract may underflow.

```
_deltaLPReserve = _deltaLPReserve.sub(amountToken);
//
balanceOf(address(this)).sub(
                    _deltaLPReserve
                )
```

Recommendation

The contract should yield zero value if the subtrahend greater than the minuend.

CR - Code Repetition

Criticality	minor
Location	contract.sol#L493,510

Description

There are code segments that are repetitive in the contract. Those segments increase the code size of the contract unnecessarily. Them methods '_transferToPair' and '_transferFromPair' are sharing the same statements.

```
_balances[sender] = _balances[sender].sub(amount);

 uint256 tax = amount.mul(_sellTax).div(1000);
uint256 receiveAmount = amount.sub(tax);

 _balances[address(this)] = _balances[address(this)].add(tax);

 _balances[recipient] = _balances[recipient].add(receiveAmount);

emit Transfer(sender, recipient, receiveAmount);
```

Recommendation

Create an internal function that contains the code segment and remove it from all the sections.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract/EGT.sol#L171,159,163,155,180,189,151,215,228,198

Description

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

```
transferFrom
decreaseAllowance
increaseAllowance
name
approve
allowance
symbol
totalSupply
decimals
...
```

Recommendation

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

L03 - Redundant Statements

Criticality	minor
Location	contract/EGT.sol#L117,119,118,116

Description

Detect the usage of redundant statements that have no effect.

EGT

Recommendation

Remove redundant statements if they congest code but offer no value.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract/EGT.sol#L708,342,583,620,619,839,835,726,409,618,364,566,697,545,721

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.

```
_enabled
_marketingAddress
_minimumTokensValueBeforeSwap
ETHAmount
_buyBackAddress
O_BuyValue
O_SellValue
_developmentAddress
_minimumETHToTransfer
...
```

Recommendation

Follow the Solidity naming convention.
https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions.

L11 - Unnecessary Boolean equality

Criticality	minor
Location	contract/EGT.sol#L258,730

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.

```
require(bool,string)(enableUniSwap == false,Already enabled!)
taxesEnabled == false || excludedFromFee[from] || excludedFromFee[to]
```

Recommendation

Remove the equality to the boolean constant.

L12 - Using Variables before Declaration

Criticality	minor
Location	contract/EGT.sol#L379

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.

```
amountToken
```

Recommendation

The variables should be declared before any usage of them.

L14 - Uninitialized Variables in Local Scope

Criticality	minor
Location	contract/EGT.sol#L379

Description

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

```
amountToken
```

Recommendation

All the local scoped variables should be initialized.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
EGT	Implementation	Context, IERC20, AccessControl		
	initialize	External	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_approve	Private	✓	
	_transfer	Private	✓	
	safeTransferETH	Internal	✓	
	swapAndLiquify	Internal	✓	lockForSwap
	reAddLiquidity	Internal	✓	
	addLiquidity	Internal	✓	
	swapTokensForEth	Internal	✓	
	getBuyValue	Internal		
	getSellValue	Internal		

	_sendShares	Internal	✓	lockForSplitShare
	sendShares	External	✓	onlyRole
	provideLP	External	Payable	-
	_transferStandard	Private	✓	
	_transferToPair	Private	✓	
	_transferFromPair	Private	✓	
	isExcludedFromFee	External		-
	includeInFee	External	✓	onlyRole
	excludeFromFee	External	✓	onlyRole
	setMarketingAddress	External	✓	onlyRole
	setBuyBackAddress	External	✓	onlyRole
	setDevelopmentAddress	External	✓	onlyRole
	_setOperatorsAddresses	Internal	✓	
	setOperatorsAddresses	External	✓	onlyRole
	setBuyBackShare	External	✓	onlyRole
	setMarketingShare	External	✓	onlyRole
	setDevelopmentShare	External	✓	onlyRole
	setLPShare	External	✓	onlyRole
	setBuyTax	External	✓	onlyRole
	getBuyTax	External		-
	setSellTax	External	✓	onlyRole
	getSellTax	External		-
	getTokenAutoSwapLimit	External		-
	setTokenAutoSwapLimit	External	✓	onlyRole
	getETHAutoTransferLimit	External		-
	setETHAutoTransferLimit	External	✓	onlyRole
	setSwapAndLiquifyEnabled	External	✓	onlyRole
	setAutoSplitSharesEnables	External	✓	onlyRole
	enableUniswap	External	✓	onlyRole

	setAcceptedSlippage	External	✓	onlyRole
	getAcceptedSlippage	External		onlyRole
	setAcceptedFeeOnAdd	External	✓	onlyRole
	getAcceptedFeeOnAdd	External		onlyRole
	_setRouterAddress	Internal	✓	
	setRouterAddress	External	✓	onlyRole
	totalDevelopmentTaxCollected	External		onlyRole
	totalMarketingTaxCollected	External		onlyRole
	totalBuyBackTaxCollected	External		onlyRole
	totalLPTaxCollected	External		onlyRole
	totalTaxCollected	External		onlyRole
	getDeltaReserve	External		-
	depositIntoReserve	External	✓	-
	burn	External	✓	-
	O_BuyValue	External		-
	O_SellValue	External		-
	<Receive Ether>	External	Payable	-
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	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	✓	
IAccessControl	Interface			
	hasRole	External		-
	getRoleAdmin	External		-
	grantRole	External	✓	-
	revokeRole	External	✓	-
	renounceRole	External	✓	-
Strings	Library			
	toString	Internal		
	toHexString	Internal		
	toHexString	Internal		
IERC165	Interface			
	supportsInterface	External		-

ERC165	Implementation	IERC165		
	supportsInterface	Public		-
AccessControl	Implementation	Context, IAccessControl, ERC165		
	supportsInterface	Public		-
	hasRole	Public		-
	_checkRole	Internal		
	_checkRole	Internal		
	getRoleAdmin	Public		-
	grantRole	Public	✓	onlyRole
	revokeRole	Public	✓	onlyRole
	renounceRole	Public	✓	-
	_setupRole	Internal	✓	
	_setRoleAdmin	Internal	✓	
	_grantRole	Internal	✓	
	_revokeRole	Internal	✓	
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
IUniswapV2Pair	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		-
	burn	External	✓	-
	swap	External	✓	-
	skim	External	✓	-
	sync	External	✓	-
	initialize	External	✓	-
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	✓	-

Contract Flow

Summary

There are some functions that can be abused by the owner like stopping transactions and transferring funds to the team’s wallet. 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 limit of max 25% 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