Cyberscope

Audit Report

Eggpot

August 2022

Type	BEP20
Network	BSC
Address	0x65BD6Ed8B252a8a5319EF8A3FB93f657fdCba239
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	Eggpot
Compiler Version	v0.8.15+commit.e14f2714
Optimization	200 runs
Explorer	https://bscscan.com/token/0x65BD6Ed8B252a8a5319EF8A3FB93f657fdCba239
Symbol	EGGPOT
Decimals	18
Total Supply	1,000,000,000
Domain	https://eggpot.io

Audit Updates

Initial Audit	19th August 2022 https://github.com/cyberscope-io/audits/blob/main/eggpot/v1/audit.pdf
Corrected	20th August 2022

Source Files

Filename	SHA256
access/Ownable.sol	65b66e7a5f3633539fbb59bb0dbebd9c29121c76490151e15f589c6bce9d59f6
Eggpot.sol	9e4e820da9ada4fbe5ae02926b6a312330067c7ce355902742355262e332c555
interface/IERC20.sol	9a9ce403bcf5796cccfc9c0eb7514128fc1dca540b0617c0dc3ba9f0c2090e95
interface/IUniswapV2Factory.sol	5626a8cec78d7abc17fdc61fe0a9b6b3527b9b471aed6247a0093889778d1b39
interface/IUniswapV2Pair.sol	944ec57bb4c13e8c79218b9c67ee2ca44248186c8c79b77f8b57c432dcffec37
interface/IUniswapV2Router02.sol	5324618037c9db4cd7a9a9e6e5b924efe1185def3a9cd07a97ecf85d6882cc52
token/ERC20.sol	0c2528c77318e3b660a57fc992c56640fc18ddafd60c2346b3c20ed7cbd609ca
utils/Context.sol	cee91680eba65e7ab59b0ae26401f8006cb78c3b8a0c65679f86e250752a98af
utils/EnumerableSet.sol	67bb227a532561b3f4765db93d0535aa139615053b44e33ecc370d7b4b90b600

Contract Analysis

● Critical ● Medium ● Minor ● 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	Unresolved
●	MT	Mints Tokens	Passed
●	BT	Burns Tokens	Passed
●	BC	Blacklists Addresses	Passed

ULTW - Transfers Liquidity to Team Wallet

Criticality	minor
Location	contract.sol#L618,630
Status	Unresolved

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 withdrawStuckETH and forceSwapBack methods.

```
function withdrawStuckETH() external onlyOwner {
    bool success;
    (success, ) = address(owner()).call{ value: address(this).balance }('');
    require(success, 'Failure! fund not sent');
}

function forceSwapBack() external onlyOwner {
    require(
      balanceOf(address(this)) >= swapTokensAtAmount,
      'Can only swap when token amount is at or higher than restriction'
    );
    swapping = true;
    swapBack();
    swapping = false;
    emit OwnerForcedSwapBack(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.

Contract Diagnostics

🔴 Critical 🟠 Medium ● Minor

Severity	Code	Description	Status
●	STC	Succeeded Transfer Check	Unresolved
●	CO	Code Optimization	Unresolved
●	L02	State Variables could be Declared Constant	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved
●	L07	Missing Events Arithmetic	Unresolved
●	L13	Divide before Multiply Operation	Unresolved
●	L15	Local Scope Variable Shadowing	Unresolved

STC - Succeeded Transfer Check

Criticality	minor
Location	contract.sol#L613
Status	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.

```
_sent = IERC20(_token).transfer(_to, _contractBalance);
```

Recommendation

The contract should check if the result of the transfer methods is successful.

CO - Code Optimization

Criticality	minor
Location	contract.sol#L436
Status	Unresolved

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.

Since the method dexRouter.getAmountsOut returns an array, there is no need to initialize an array.

```
uint256[] memory amounts = new uint256[](2);
amounts = dexRouter.getAmountsOut(minBuyAmount, path);
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contracts/Eggpot.sol#L37
Status	Unresolved

Description

Constant state variables should be declared constant to save gas.

```
botsCaught
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contracts/Eggpot.sol#L282,271,270,605,158,64,284,272,283,63,624
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.

```
_operationsFee
_liquidityFee
_token
_presaleAddress
_isExcludedMaxTransactionAmount
_jackpotFee
_to
_isExcludedFromFees
_operationsAddress
...
```

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
Location	contracts/Eggpot.sol#L508,230,281,513,269,208,503
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.

```
percentForJackpot = percent
numberOfBuysForJackpot = num
sellOperationsFee = _operationsFee
minBuyAmount = minBuy
buyOperationsFee = _operationsFee
swapTokensAtAmount = newAmount
timeBetweenBuysForJackpot = timeInMinutes * 60
```

Recommendation

Emit an event for critical parameter changes.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contracts/Eggpot.sol#L307
Status	Unresolved

Description

Performing divisions before multiplications may cause lose of prediction.

```
fees = (amount * (sellTotalFees)) / FEE_DENOMINATOR
fees = (amount * (buyTotalFees)) / FEE_DENOMINATOR
```

Recommendation

The multiplications should be prior to the divisions.

L15 - Local Scope Variable Shadowing

Criticality	minor
Location	contracts/Eggpot.sol#L115
Status	Unresolved

Description

The are variables that are defined in the local scope containing the same name from an upper scope.

```
totalSupply
```

Recommendation

The local variables should have different names from the upper scoped variables.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	External	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
Eggpot	Implementation	ERC20, Ownable		
	<Constructor>	Public	Payable	ERC20
	<Receive Ether>	External	Payable	-
	addPresaleAddressForExclusions	External	✓	onlyOwner
	enableTrading	External	✓	onlyOwner
	removeLimits	External	✓	onlyOwner
	enableLimits	External	✓	onlyOwner
	setJackpotEnabled	External	✓	onlyOwner
	updateMaxBuyAmount	External	✓	onlyOwner
	updateMaxSellAmount	External	✓	onlyOwner
	updateMaxWallet	External	✓	onlyOwner
	updateSwapTokensAtAmount	External	✓	onlyOwner
	_excludeFromMaxTransaction	Private	✓	
	airdropToWallets	External	✓	onlyOwner
	setNumberOfBuysForJackpot	External	✓	onlyOwner
	excludeFromMaxTransaction	External	✓	onlyOwner
	setAutomatedMarketMakerPair	External	✓	onlyOwner
	_setAutomatedMarketMakerPair	Private	✓	
	updateBuyFees	External	✓	onlyOwner
	updateSellFees	External	✓	onlyOwner
	disableJeetTaxes	External	✓	onlyOwner

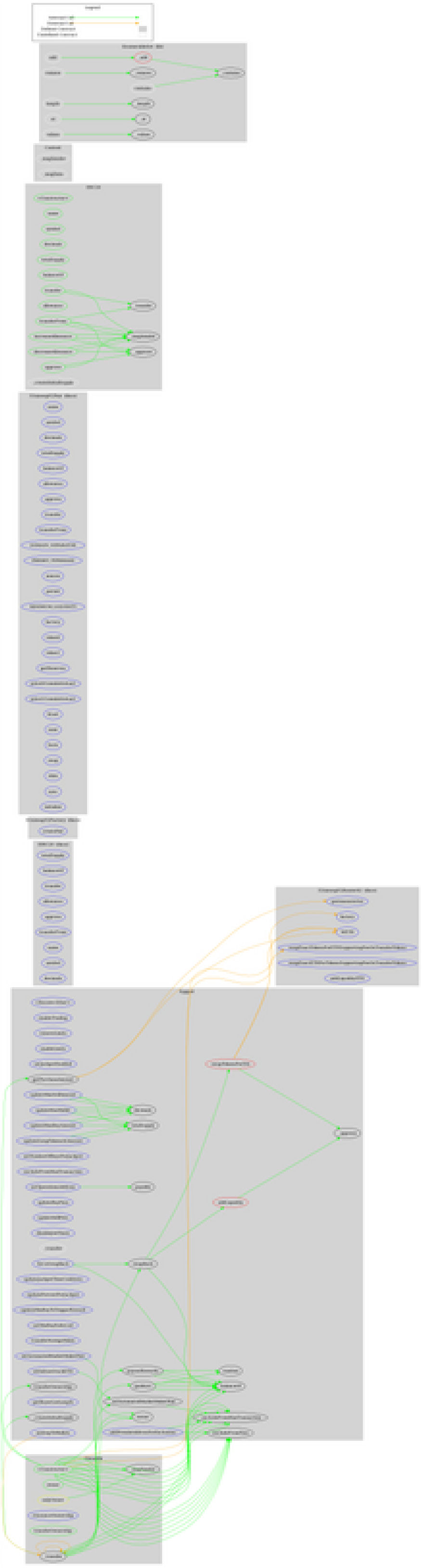
	excludeFromFees	Public	✓	onlyOwner
	_transfer	Internal	✓	
	getPurchaseAmount	Public		-
	gasBurn	Private	✓	
	payoutRewards	Private	✓	
	random	Private		
	updateJackpotTimeCooldown	External	✓	onlyOwner
	updatePercentForJackpot	External	✓	onlyOwner
	updateMinBuyToTriggerReward	External	✓	onlyOwner
	setMinBuyEnforced	External	✓	onlyOwner
	swapTokensForEth	Private	✓	
	addLiquidity	Private	✓	
	swapBack	Private	✓	
	transferForeignToken	External	✓	onlyOwner
	withdrawStuckETH	External	✓	onlyOwner
	setOperationsAddress	External	✓	onlyOwner
	forceSwapBack	External	✓	onlyOwner
	getBuyerListLength	External		-
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
	name	External		-
	symbol	External		-
	decimals	External		-
IUniswapV2Factory	Interface			
	createPair	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		-
	mint	External	✓	-
	burn	External	✓	-
	swap	External	✓	-
	skim	External	✓	-
	sync	External	✓	-
	initialize	External	✓	-
IUniswapV2Router02	Interface			
	factory	External		-
	WETH	External		-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-

	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	addLiquidityETH	External	Payable	-
	getAmountsOut	External		-
ERC20	Implementation	Context, IERC20		
	<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	✓	
	_createInitialSupply	Internal	✓	
	_approve	Internal	✓	
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
EnumerableSet	Library			
	_add	Private	✓	
	_remove	Private	✓	
	_contains	Private		
	_length	Private		
	_at	Private		
	_values	Private		
	add	Internal	✓	

	remove	Internal	✓	
	contains	Internal		
	length	Internal		
	at	Internal		
	values	Internal		

Contract Flow



Domain Info

Domain Name	eggpot.io
Registry Domain ID	40e9b1c23d66463e9de89405c5e6ad50-DONUTS
Creation Date	2022-08-16T12:02:53Z
Updated Date	2022-08-16T12:09:39Z
Registry Expiry Date	2023-08-16T12:02:53Z
Registrar WHOIS Server	whois.namecheap.com
Registrar URL	https://www.namecheap.com/
Registrar	NameCheap, Inc.
Registrar IANA ID	1068

The domain has been created in 12 months before the creation of the audit.

There is no public billing information, the creator is protected by the privacy settings.

Summary

The Smart Contract analysis reported one minor severity issue. The contract owner has the authority to transfer funds to the team's wallet. Other than that, the contract owner can access some admin functions that can not be used in a malicious way to disturb the users' transactions.

The contract has a reward mechanism for every buyer. If the reward mechanism is enabled, the users that buy tokens greater than a threshold are applicable to win. There is also a limit of max 15% buy fees and max limit of 20% for sell 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