Cyberscope

Audit Report

BULLGOLD

April 2022

Type	BEP20
Network	BSC
Address	0x46eF8d452d415f8c7E4E49Ba3782E394c919D794
Audited by	© cyberscope

Table of Contents

Table of Contents 1
Contract Review 3
Source Files 3
Audit Updates 3
Contract Analysis 4
ELFM - Exceed Limit Fees Manipulation 5
Description 5
Recommendation 5
Contract Diagnostics 6
FSA - Fixed Swap Address 7
Description 7
Recommendation 7
L01 - Public Function could be Declared External 8
Description 8
Recommendation 8
L02 - State Variables could be Declared Constant 9
Description 9
Recommendation 9
L04 - Conformance to Solidity Naming Conventions 10
Description 10
Recommendation 10
L07 - Missing Events Arithmetic 11
Description 11
Recommendation 11
L09 - Dead Code Elimination 12
Description 12

Recommendation 12
L13 - Divide before Multiply Operation 13
Description 13
Recommendation 13
Contract Functions 14
Contract Flow 19
Domain Info 20
Summary 21
Disclaimer 22
About Cyberscope 23

Contract Review

Contract Name	BULLGOLD
Compiler Version	v0.6.12+commit.27d51765
Optimization	200 runs
Licence	Unlicense
Explorer	https://bscscan.com/token/0x46eF8d452d415f8c7E4E49Ba3782E394c919D794
Symbol	BULLG
Decimals	18
Total Supply	1,000,000,000
Domain	bullgold.finance

Source Files

Filename	**SHA256**
contract.sol	2dadf13363d0f020ce9e0c7967d85e0d06ef2e6782dce2d152a8dc659e5528c0

Audit Updates

Initial Audit	24th April 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

ELFM - Exceed Limit Fees Manipulation

Criticality	critical
Location	contract.sol#L729

Description

The contract owner has the authority to increase over the allowed limit of 25%. The owner may take advantage of it by calling the `setLiquidityFeePercent` function with a high percentage value.

```
function setLiquidityFeePercent(uint256 liquidityFee) external onlyOwner() {
    _liquidityFee = liquidityFee;
}
```

Recommendation

The contract could embody a check for the maximum acceptable value.

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
	FSA	Fixed Swap Address
	L01	Public Function could be Declared External
	L02	State Variables could be Declared Constant
	L04	Conformance to Solidity Naming Conventions
	L07	Missing Events Arithmetic
	L09	Dead Code Elimination
	L13	Divide before Multiply Operation

FSA - Fixed Swap Address

Criticality	minor
Location	contract.sol#L617

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.

```
        IPancakeRouter02 _pancakeswapV2Router =
IPancakeRouter02(0x10ED43C718714eb63d5aA57B78B54704E256024E);
         // Create a uniswap pair for this new token
        pcsV2Pair =
IPancakeFactory(_pancakeswapV2Router.factory()).createPair(address(this),
_pancakeswapV2Router.WETH());
```

Recommendation

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

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L332,341,632,636,640,644,652,657,661,665 and 6 more

Description

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

```
reflectionFromToken
reflect
totalFees
decreaseAllowance
increaseAllowance
transferFrom
approve
allowance
excludeFromFee
...
```

Recommendation

Use the external attribute for functions never called from the contract

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L603,594,596,601,595,602

Description

Constant state variables should be declared constant to save gas.

```
_symbol
_numTokensSellToAddToLiquidity
_name
_maxWalletToken
_maxTxAmount
_decimals
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L359,513,534,535,552,715,585,588,589,591 and 4 more

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.

```
_maxWalletToken
_numTokensSellToAddToLiquidity
_maxTxAmount
_previousLiquidityFee
_previousTaxFee
_liquidityFee
_taxFee
_tTotal
_amount
...
```

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	contract.sol#L729

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.

```
_liquidityFee = liquidityFee
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L715

Description

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

```
calculateLiquidityFee
```

Recommendation

Remove unused functions.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L821

Description

Performing divisions before multiplications may cause lose of prediction.

```
tLiquidity = tAmount.div(10 ** 2).mul(_liquidityFee)
tFee = tAmount.div(10 ** 2).mul(_taxFee)
```

Recommendation

The multiplications should be prior to the divisions.

Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
IBEP20	Interface			
	totalSupply	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
Context	Implementation			
	<Constructor>	Internal	✓	
	_msgSender	Internal		
	_msgData	Internal		
SafeMath	Library			
	add	Internal		

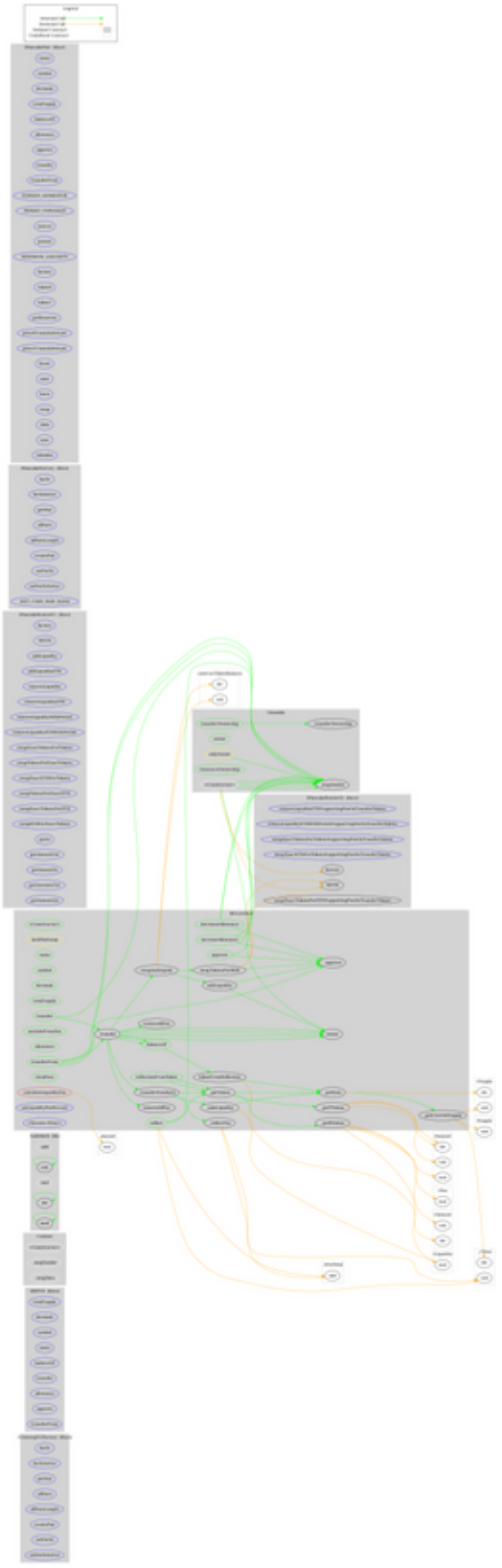
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		
Ownable	Implementation	Context		
	<Constructor>	Internal	✓	
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
IPancakeRouter01	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		-
IPancakeRouter02	Interface	IPancakeRouter01		
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
IPancakeFactory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
	INIT_CODE_PAIR_HASH	External		-
IPancakePair	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	✓	-
BULLGOLD	Implementation	Context, IBEP20, Ownable		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	excludeFromFee	Public	✓	onlyOwner
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-

	totalFees	Public		-
	reflect	Public	✓	-
	reflectionFromToken	Public		-
	tokenFromReflection	Public		-
	calculateLiquidityFee	Private		
	_approve	Private	✓	
	setLiquidityFeePercent	External	✓	onlyOwner
	removeAllFee	Private	✓	
	restoreAllFee	Private	✓	
	_transfer	Private	✓	
	_transferStandard	Private	✓	
	_reflectFee	Private	✓	
	_getValues	Private		
	_getTValues	Private		
	_getRValues	Private		
	_getRate	Private		
	_getCurrentSupply	Private		
	_takeLiquidity	Private	✓	
	swapAndLiquify	Private	✓	lockTheSwap
	swapTokensForBNB	Private	✓	
	addLiquidity	Private	✓	
	<Receive Ether>	External	Payable	-

Contract Flow



Domain Info

Domain Name	bullgold.finance
Registry Domain ID	5457e04c8a8e46cd9b16f84d9265fe48-DONUTS
Creation Date	2022-04-08T14:25:41Z
Updated Date	2022-04-13T14:25:50Z
Registry Expiry Date	2023-04-08T14:25:41Z
Registrar WHOIS Server	http://www.hostinger.com
Registrar URL	http://www.hostinger.com
Registrar	Hostinger, UAB
Registrar IANA ID	1636

The domain has been created 16 days before the creation of the audit. It will expire in 12 months.

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

Summary

The Smart Contract analysis reported one critical issue. The contract Owner can manipulate fees without limit. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats.

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 provides all the essential tools to assist users draw their own conclusions.

The Cyberscope team

https://www.cyberscope.io