



Cyberscope

Audit Report

LuffyInu

April 2022

Type BEP20

Network BSC

Address 0x720f6edd18f5e2f414dba8fcd81e98724708c39e

Audited by © cyberscope

Table of Contents

Table of Contents	1
Contract Review	3
Source Files	3
Audit Updates	3
Contract Analysis	4
Contract Diagnostics	5
BLC - Business Logic Concern	6
Description	6
Recommendation	6
FSA - Fixed Swap Address	7
Description	7
Recommendation	7
L01 - Public Function could be Declared External	8
Description	8
Recommendation	8
L04 - Conformance to Solidity Naming Conventions	9
Description	9
Recommendation	9
L09 - Dead Code Elimination	10
Description	10
Recommendation	10
L11 - Unnecessary Boolean equality	11
Description	11
Recommendation	11
L15 - Local Scope Variable Shadowing	12
Description	12

Recommendation	12
Contract Functions	13
Contract Flow	17
Domain Info	18
Summary	19
Disclaimer	20
About Cyberscope	21

Contract Review

Contract Name	LuffyInu
Compiler Version	v0.8.13+commit.abaa5c0e
Optimization	200 runs
Licence	None
Explorer	https://bscscan.com/token/0x720f6edd18f5e2f414dba8fcd81e98724708c39e
Symbol	LUFFY
Decimals	18
Total Supply	1,000,000,000
Domain	luffyinutoken.com

Source Files

Filename	SHA256
contract.sol	e17ce4b84710e49472af1f5421a29075d20b19005a9ba61b860d855b68869b6b

Audit Updates

Initial Audit	28th 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

Contract Diagnostics

● Critical ● Medium ● Minor

Severity	Code	Description
●	BLC	Business Logic Concern
●	FSA	Fixed Swap Address
●	L01	Public Function could be Declared External
●	L04	Conformance to Solidity Naming Conventions
●	L09	Dead Code Elimination
●	L11	Unnecessary Boolean equality
●	L15	Local Scope Variable Shadowing

BLC - Business Logic Concern

Criticality	critical
Location	contract.sol#L1165,1178

Description

The contract is transferring the liquidity fee to the pair. This is not working as a liquidity fee since it is not adding the pair tokens.

```
super._transfer(sender, uniswapV2Pair, _LiquidityFee);
```

The contract takes a liquidity fee from the sale transactions but it does not transfer it to any recipient. As a result the total supply diverges from the balances.

```
if(sender == uniswapV2Pair && // Swap BNB to Tokens (BUY)
    recipient != owner() &&
    !excludeFee[recipient]){

    if(taxStatus == true) {
        uint256 _MarketingFee = amount.mul(marketingBuyFee).div(100);
        uint256 _LiquidityFee = amount.mul(liquidityBuyFee).div(100);
        uint256 _BurnFee = amount.mul(burnBuyFee).div(100);

        super._transfer(sender, marketingAddress, _MarketingFee);
        super._burn(sender, _BurnFee);

        amount = amount.sub(_MarketingFee.add(_BurnFee).add(_LiquidityFee));
    }
}
super._transfer(sender, recipient, amount);
```

Recommendation

The team is advised to carefully check if the implementation follows the expected business logic.

FSA - Fixed Swap Address

Criticality	minor
Location	contract.sol#L1136

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.

```
uniswapV2Router = IUniswapV2Router02(routerAddress);
    uniswapV2Pair =
    IUniswapV2Factory(uniswapV2Router.factory()).createPair(address(this),
    uniswapV2Router.WETH());
    _approve(address(this), address(uniswapV2Router), ~uint256(0));
```

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#L788,796

Description

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

```
transferOwnership  
renounceOwnership
```

Recommendation

Use the external attribute for functions never called from the contract

L04 - Conformance to Solidity Naming Conventions

Criticality

minor

Location

contract.sol#L38,1120,1123,1124,1125,1126,1127,1128,1129,1130

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.

```
routerAddress
marketingAddress
burnBuyFee
burnSellFee
liquidityBuyFee
liquiditySellFee
marketingBuyFee
marketingSellFee
maxSupply
...
```

Recommendation

Follow the Solidity naming convention.

<https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions>

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L352,362,381,395,441,451,414,424,303,327 and 2 more

Description

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

```
_burnFrom  
verifyCallResult  
sendValue  
isContract  
functionStaticCall  
functionDelegateCall  
functionCallWithValue  
functionCall  
...
```

Recommendation

Remove unused functions.

L11 - Unnecessary Boolean equality

Criticality

minor

Location

contract.sol#L1149

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.

```
taxStatus == true
```

Recommendation

Remove the equality to the boolean constant.

L15 - Local Scope Variable Shadowing

Criticality

minor

Location

contract.sol#L860,1135

Description

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

```
_symbol  
_name  
symbol  
name
```

Recommendation

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

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	✓	-
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	✓	-
IERC20	Interface			
	totalSupply	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	getOwner	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
Address	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		

	functionStaticCall	Internal		
	functionDelegateCall	Internal	✓	
	functionDelegateCall	Internal	✓	
	verifyCallResult	Internal		
SafeMath	Library			
	tryAdd	Internal		
	trySub	Internal		
	tryMul	Internal		
	tryDiv	Internal		
	tryMod	Internal		
	add	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	mod	Internal		
	sub	Internal		
	div	Internal		
	mod	Internal		
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
ERC20	Implementation	Context, IERC20, Ownable		
	<Constructor>	Public	✓	-
	getOwner	External		-

	name	External		-
	decimals	External		-
	symbol	External		-
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
	increaseAllowance	External	✓	-
	decreaseAllowance	External	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_burnFrom	Internal	✓	
LuffyInu	Implementation	ERC20		
	<Constructor>	Public	✓	ERC20
	setTax	External	✓	onlyOwner
	addExcludeFee	External	✓	onlyOwner
	_transfer	Internal	✓	

Contract Flow



Domain Info

Domain Name	luffyinutoken.com
Registry Domain ID	2692212238_DOMAIN_COM-VRSN
Creation Date	2022-04-27T04:45:08Z
Updated Date	2022-04-27T15:33:23Z
Registry Expiry Date	2023-04-27T04:45:08Z
Registrar WHOIS Server	whois.publicdomainregistry.com
Registrar URL	www.publicdomainregistry.com
Registrar	PDR Ltd. d/b/a PublicDomainRegistry.com
Registrar IANA ID	303

The domain has been created 1 day 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

LuffyInu is an interesting project that has a friendly and growing community. The Smart Contract analysis reported no compiler error 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 fees are fixed to 3%.

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.

Cyberscope 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>