Cyberscope

Audit Report

Jujutsu Inu

April 2022

Type	BEP20
Network	BSC
Address	0xeafa273f5a6f9c04052e62f9d3c8d5135a3b620b
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	JujutsuInu
Compiler Version	v0.8.5+commit.a4f2e591
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0xeafa273f5a6f9c04052e62f9d3c8d5135a3b620b
Symbol	JJI
Decimals	9
Total Supply	1,000,000,000
Domain	jujutsuinu.com

Source Files

Filename	**SHA256**
contract.sol	0ca7f39cf59e3aca23f448aad3e56f916cd66976d8126b9444b51a93f32e9b0b

Audit Updates

Initial Audit	23rd 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

ST - Stop Transactions

Criticality	critical
Location	contract.sol#L535, 545

Description

The contract owner has the authority to stop transactions for all users excluding the owner. The owner may convert the contract into a honeypot by increasing the `sellFee` value.

```
function getTotalFee(bool selling) public view returns (uint256) {
        if(launchedAt + 1 >= block.number){ return feeDenominator.sub(1); }
        if(selling){ return sellFee; }
        return buyFee;
    }
```

```
    function takeFee(address sender, address receiver, uint256 amount)
internal returns (uint256) {
        uint256 feeAmount = amount.mul(getTotalFee(receiver ==
pair)).div(feeDenominator);

        _balances[address(this)] = _balances[address(this)].add(feeAmount);
        emit Transfer(sender, address(this), feeAmount);

        return amount.sub(feeAmount);
    }
```

Recommendation

The contract could embody a check for not allowing setting the sellFee more than a reasonable amount. A suggested implementation could check that the maximum amount should be less than a fixed percentage of the total supply.

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.

ELFM - Exceed Limit Fees Manipulation

Criticality	critical
Location	contract.sol#L631

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 setFees function with a high percentage value.

```
function setFees(uint256 _buyFee, uint256 _sellFee, uint256 _feeDenominator)
external onlyOwner {
        require(_feeDenominator > _buyFee.add(_sellFee), "fee invalid");
        buyFee = _buyFee;
        sellFee = _sellFee;
        feeDenominator = _feeDenominator;
    }
```

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
	L11	Unnecessary Boolean equality
	L13	Divide before Multiply Operation

FSA - Fixed Swap Address

Criticality	minor
Location	contract.sol#L455

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.

```
router = IDEXRouter(_router);
WBNB = router.WETH();

pair = IDEXFactory(router.factory()).createPair(WBNB, address(this));
```

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#L357,364,368,389

Description

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

```
transferOwnership
selfUnauthorize
unauthorize
authorize
```

Recommendation

Use the external attribute for functions never called from the contract

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L412,413,419

Description

Constant state variables should be declared constant to save gas.

```
_totalSupply
ZERO
DEAD
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L275,626,631,637,643,648,411,412,413,415 and 11 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.

```
_allowances
_balances
_totalSupply
_decimals
_symbol
_name
ZERO
DEAD
WBNB
...
```

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#L609,631,637,643,648

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.

```
targetLiquidity = _target
swapThreshold = _amount
liquidityFeeShare = _liquidityFeeShare
buyFee = _buyFee
maxTxAmount = amount
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L541

Description

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

```
isSelling
```

Recommendation

Remove unused functions.

L11 - Unnecessary Boolean equality

Criticality	minor
Location	contract.sol#L527

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)(txLimitEnable == false || (amount <= maxTxAmount ||
isTxLimitExempt[sender]),TX Limit Exceeded)
```

Recommendation

Remove the equality to the boolean constant.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L561

Description

Performing divisions before multiplications may cause lose of prediction.

```
amountBNBLiquidity =
amountBNB.mul(dynamicLiquidityFee.div(2)).div(totalFeeShare.sub(dynamicLiquidi
tyFee.div(2)))
amountToLiquify =
swapThreshold.mul(dynamicLiquidityFee.div(2)).div(totalFeeShare)
```

Recommendation

The multiplications should be prior to the divisions.

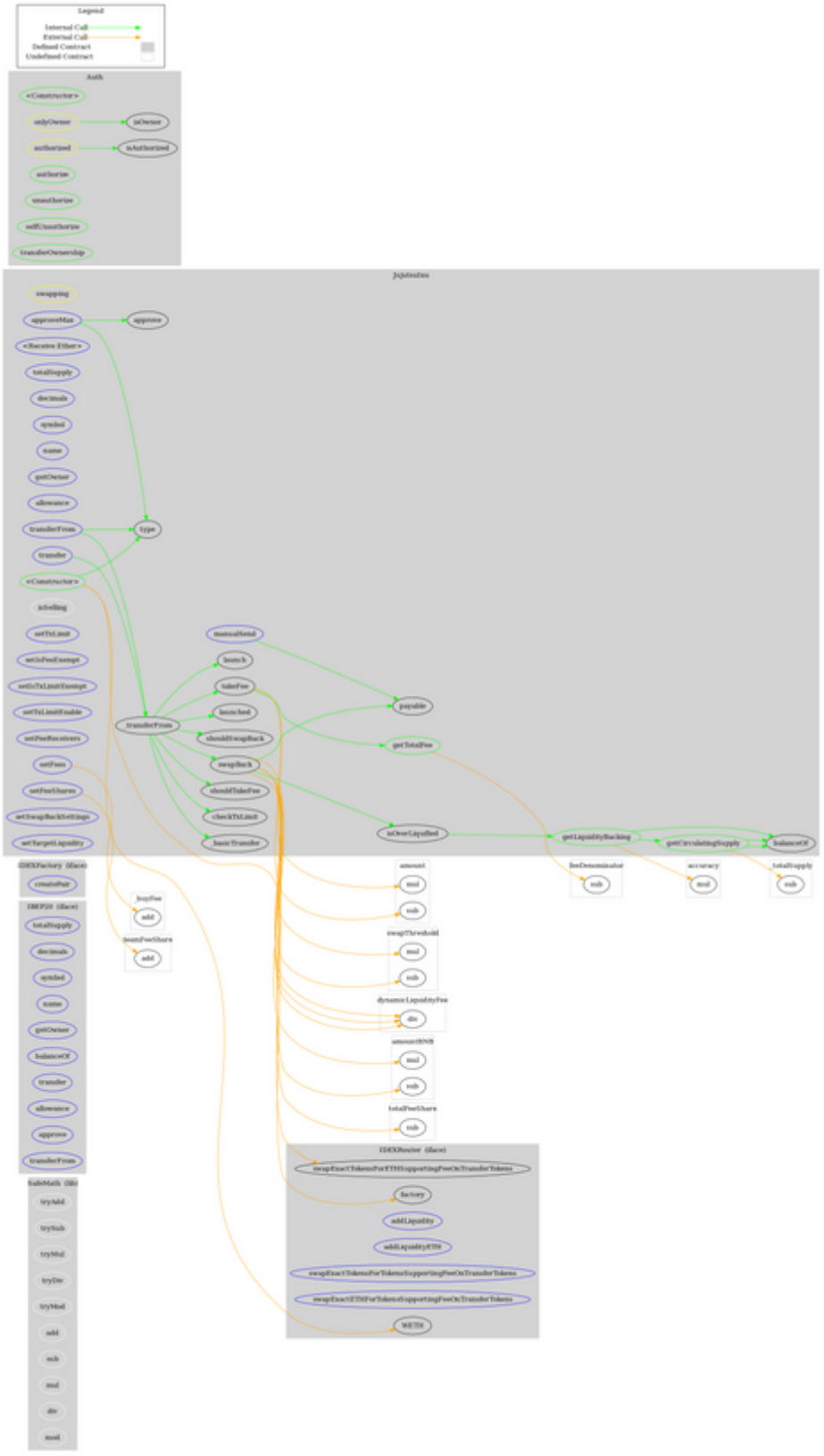
Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
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		
IBEP20	Interface			
	totalSupply	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	getOwner	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IDEXFactory	Interface			
	createPair	External	✓	-

IDEXRouter	Interface			
	factory	External		-
	WETH	External		-
	addLiquidity	External	✓	-
	addLiquidityETH	External	Payable	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
Auth	Implementation			
	<Constructor>	Public	✓	-
	authorize	Public	✓	onlyOwner
	unauthorize	Public	✓	onlyOwner
	selfUnauthorize	Public	✓	authorized
	isOwner	Public		-
	isAuthorized	Public		-
	transferOwnership	Public	✓	onlyOwner
JujutsuInu	Implementation	IBEP20, Auth		
	<Constructor>	Public	✓	Auth
	<Receive Ether>	External	Payable	-
	totalSupply	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	getOwner	External		-
	balanceOf	Public		-
	allowance	External		-
	approve	Public	✓	-
	approveMax	External	✓	-
	transfer	External	✓	-

	transferFrom	External	✓	-
	_transferFrom	Internal	✓	
	_basicTransfer	Internal	✓	
	checkTxLimit	Internal		
	shouldTakeFee	Internal		
	getTotalFee	Public		-
	isSelling	Internal		
	takeFee	Internal	✓	
	shouldSwapBack	Internal		
	swapBack	Internal	✓	swapping
	launched	Internal		
	launch	Internal	✓	
	setTxLimit	External	✓	authorized
	setIsFeeExempt	External	✓	authorized
	setIsTxLimitExempt	External	✓	authorized
	setTxLimitEnable	External	✓	authorized
	setFeeReceivers	External	✓	authorized
	setFees	External	✓	onlyOwner
	setFeeShares	External	✓	onlyOwner
	setSwapBackSettings	External	✓	authorized
	setTargetLiquidity	External	✓	authorized
	manualSend	External	✓	authorized
	getCirculatingSupply	Public		-
	getLiquidityBacking	Public		-
	isOverLiquified	Public		-

Contract Flow



Domain Info

Domain Name	jujutsuinu.com
Registry Domain ID	2691291324_DOMAIN_COM-VRSN
Creation Date	2022-04-23T09:10:53Z
Updated Date	2022-04-23T09:10:54Z
Registry Expiry Date	2023-04-23T09:10:53Z
Registrar WHOIS Server	whois.wix.com
Registrar URL	http://www.wix.com
Registrar	Wix.com Ltd.
Registrar IANA ID	3817

The domain has been created about 1 hour 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

There are some functions that can be abused by the owner, like manipulating fees to 99% and stopping transactions for everyone except the owner. The contract can also be converted into a honeypot and prevent users from selling if the owner abuses the admin functions. 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