Cyberscope

Audit Report

SPIDERSHIBA INU

8th March 2023

Type BEP20

Network BSC

Address 0x67866d577c7Fc8F1944a44506c874eFe63A95278

Auditedby © cyberscope

Table of Contents

Contract Review

Contract Name	Spider Shiba Inu
Compiler Version	v0.8.2+commit.e5eed63a
Optimization	200 runs
Licence	None
Explorer	https://bscscan.com/token/0x67866d577c7fc8f1944a44506c874efe63a95278
Symbol	SSB
Decimals	9
Total Supply	1,000,000
Domain	https://spidershiba.space

Source Files

Filename	**SHA256**
contract.sol	34f1746f161650ce6e7f6ba78a93cc0754be6f7c88d81f be78d5699146138a8b

Audit Updates

Initial Audit	8th March 2023
Corrected	8th March 2023

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
●	OTU	Owner Transfer User's Tokens
●	T 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
		Contract Owner is not able to mint new tokens
●	M	Contract Owner is not able to burn tokens from specific wallet
●	T BT	Contract Owner is not able to blacklist wallets from selling
●	BC	

ULTW - Unlimited Liquidity to Team Wallet

Criticality	minor
Location	contract.sol#L1100,1105

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 therescueBNBand rescueAnyBEP20Tokens methods.

```
functionrescueBNB(uint256 weiAmount)externalonlyOwner{
                require(address(this).balance >= weiAmount,"insufficientBNB balance");
payable(msg.sender).transfer(weiAmount);
}

functionrescueAnyBEP20Tokens(address _tokenAddr,address _to, uint _amount)
public onlyOwner {
IERC20(_tokenAddr).transfer(_to, _amount);
}
```

Recommendation

The contract could embody a check for the maximum amount of funds that can be transferred.

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
	STC	Succeeded Transfer Check
	CR	Code Repetition
	L01	Public Function could be Declared External
	L02	State Variables could be Declared Constant
	L04	Conformance to Solidity Naming Conventions
	L05	Unused State Variable
	L07	Missing Events Arithmetic
	L08	Tautology or Contradiction
	L09	Dead Code Elimination
	L13	Divide before Multiply Operation
	L15	Local Scope Variable Shadowing

STC - Succeeded Transfer Check

Criticality	minor
Location	contract.sol#L1290,1298

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.

```
(success,) = address(devWallet).call{value: ethForDev}("");
(success,) = address(marketingWallet).call{value: address(this).balance}("");
```

Recommendation

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

CR - Code Repetition

Criticality	minor
Location	contract.sol#L1198

Description

There are code segments that are repetitive in the contract. Those segments increase the code size of the contract unnecessarily.

The sell and buy calculations share the same functionality.

```
fees = amount.mul(sellTotalFees).div(100);
tokensForLiquidity += fees * sellLiquidityFee / sellTotalFees;
tokensForDev += fees * sellDevFee / sellTotalFees;
tokensForMarketing += fees * sellMarketingFee / sellTotalFees;
```

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.sol#L1067,303,643,240,344,1091,223,266,325,652,215,285,1101,274

Description

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

```
allowance
rescueAnyBEP20Tokens
approve
name
transferOwnership
increaseAllowance
transfer
symbol
isExcludedFromFees

...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L886,887

Description

Constant state variables should be declared constant to save gas.

```
lastManualLpBurnTi
me
manualBurnFrequen
cy
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L1297,1101,927,1054,1046,870,42,59,41,915,732,929

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.

```
_liquidityFee
devWalletUpdated
WETH
_marketingFee
_isExcludedMaxTransactionAmount
_frequencyInSeconds
_percent
DOMAIN_SEPARATOR
MINIMUM_LIQUIDITY

...
```

Recommendation

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

L05 - Unused State Variable

Criticality	minor
Location	contract.sol#L663

Description

There are segments that contain unused state variables.

```
MAX_INT256
```

Recommendation

Remove unused state variables.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L1020,1027,1054,1297,1032,1046

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.

```
buyMarketingFee = _marketingFee
maxWallet = newNum * (10 ** 18)
lpBurnFrequency = _frequencyInSeconds
sellMarketingFee = _marketingFee
maxTransactionAmount = newNum * (10 ** 18)
swapTokensAtAmount = newAmount
```

Recommendation

Emit an event for critical parameter changes.

L08 - Tautology or Contradiction

Criticality	minor
Location	contract.sol#L1297

Description

Detects expressions that are tautologies or contradictions. For instance, an uint variable will always be greater than or equal to zero.

```
require(bool,string)(_percent <= 1000 && _percent >= 0,Must set auto LP burn percent between 0% and 10%)
```

Recommendation

Fix the incorrect comparison by changing the value type or the comparison.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L709,722,408,715

Description

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

```
toUint256Safe
_burn
toInt256Safe
abs
```

Recommendation

Remove unused functions.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L1108

Description

Performing divisions before multiplications may cause lose of prediction.

```
tokensForLiquidity += fees * sellLiquidityFee / sellTotalFees fees =
amount.mul(buyTotalFees).div(100)
tokensForDev += fees * sellDevFee / sellTotalFees
tokensForMarketing += fees * sellMarketingFee / sellTotalFees
```

Recommendation

The multiplications should be prior to the divisions.

L15 - Local Scope Variable Shadowing

Criticality	minor
Location	contract.sol#L960

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	V isibility	Mutability	Modifiers
Context	Implementation			
	_msgSender	Interna		
	_msgData	l		
		Interna		
IUniswapV2P a ir	Interface	l		
	name	Externa		-
	symbol	l		-
	decimals	Externa		-
	totalSupply	l		-
	balanceOf	Externa		-
	allowance	l		-
	approve	Externa	✓	-
	transfer	l	✓	-
	transferFrom	Externa	✓	-
	DOMAIN_SEPARATOR	l		-
	PERMIT_TYPEHASH	Externa		-
	nonces	l		-
	permit	Externa	✓	-
	MINIMUM_LIQUIDITY	l		-
	factory	Externa		-
	token0	l		-
	token1	Externa		-
	getReserves	l		-
	price0CumulativeLast	Externa		-
	price1CumulativeLast	l		-
	kLast	Externa		-
	mint	l	✓	-
	burn	Externa	✓	-
		l		
		Externa		
		l		
		Externa		
		l		

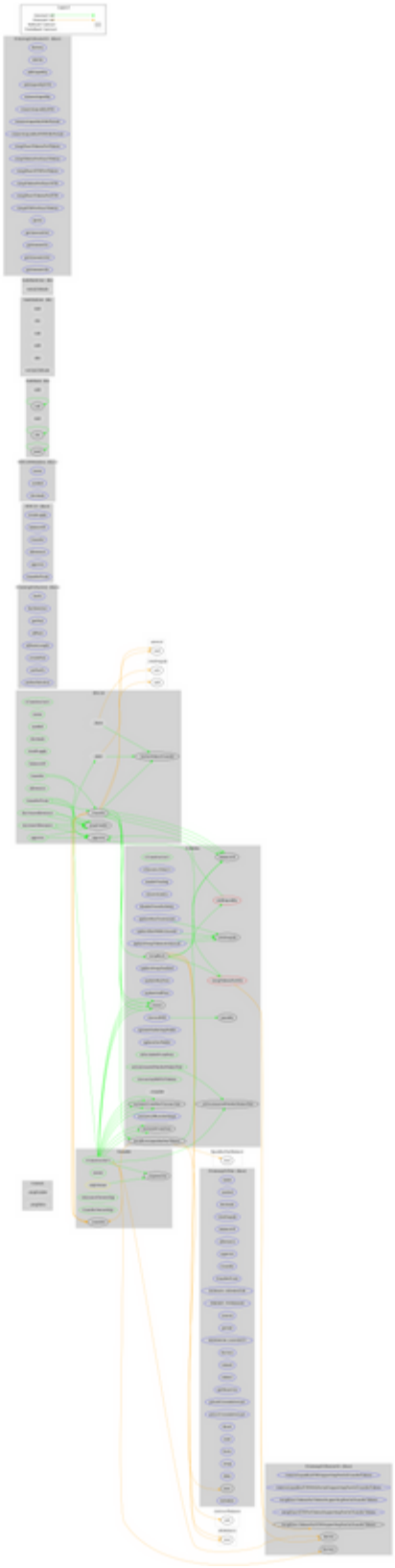
	swap	External	✓	-
	skim	l	✓	-
	sync	External	✓	-
	initialize	l	✓	-
		External		
IUniswapV2Factory	Interface	l External		
	feeTo	External		-
	feeToSetter	l		-
	getPair	External		-
	allPairs	l		-
	allPairsLength	External		-
	createPair	l	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	l	✓	-
		External		
IERC20	Interface	l		
	totalSupply	External		-
	balanceOf	l		-
	transfer	External	✓	-
	allowance	l		-
	approve	External	✓	-
	transferFrom	l	✓	-
		External		
IERC20Metadata	Interface	l IERC20 External		
	name	External		-
	symbol	l External		-
	decimals	External		-
		l		
ERC20	Implementation	Context, IERC20, IERC20Metadata		
	<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	√	
	_mint	Internal	√	
	_burn	Internal	√	
	_approve	Internal	√	
	_beforeTokenTransfer	Internal	√	
SafeMath	Library			
	add	Interna		
	sub	l		
	sub	Interna		
	mul	l		
	div	Interna		
	div	l		
	mod	Interna		
	mod	l		
		Interna		
Ownable	Implementation	Context		
	<Constructor>	Public	√	-
	owner	Public		-
	renounceOwnership	Public	√	onlyOwner
	transferOwnership	Public	√	onlyOwner
		Interna		
SafeMathInt	Librar	l		
	y mul	Interna		
	div	l		
	sub	Interna		
		l		
		Interna		
		l		

	add	Interna		
	abs	l		
	toUint256Safe	Interna		
		l		
SafeMathUint	Library	Interna		
	toInt256Safe	Internal		
IUniswapV2R o uter01	Interface			
	factory	Externa		-
	WETH	l		-
	addLiquidity	Externa	✓	-
	addLiquidityETH	l	Payable	-
	removeLiquidity	Externa	✓	-
	removeLiquidityETH	l	✓	-
	removeLiquidityWithPermit	Externa	✓	-
	removeLiquidityETHWithPermit	l	✓	-
	swapExactTokensForTokens	Externa	✓	-
	swapTokensForExactTokens	l	✓	-
	swapExactETHForTokens	Externa	Payable	-
	swapTokensForExactETH	l	✓	-
	swapExactTokensForETH	Externa	✓	-
	swapETHForExactTokens	l	Payable	-
	quote	Externa		-
	getAmountOut	l		-
	getAmountIn	Externa		-
	getAmountsOut	l		-
	getAmountsIn	Externa		-
		l		
IUniswapV2R o uter02	Interface	Externa l		
	removeLiquidityETHSupportingFeeO nTransferTokens	Externa	✓	-
	removeLiquidityETHWithPermitSupp ortingFeeOnTransferTokens	l External	✓	-
	swapExactTokensForTokensSupport i ngFeeOnTransferTokens	Externa l External	✓	-
		Externa l Externa l Externa		

	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	√	-
SpiderShiba Inu	Implementation	ERC20, Ownable		
	<Constructor>	Public	√	ERC20
	<Receive Ether>	External	Payable	-
	enableTrading	External	√	onlyOwner
	removeLimits	External	√	onlyOwner
	disableTransferDelay	External	√	onlyOwner
	updateSwapTokensAtAmount	External	√	onlyOwner
	updateMaxTxnAmount	External	√	onlyOwner
	updateMaxWalletAmount	External	√	onlyOwner
	excludeFromMaxTransaction	Public	√	onlyOwner
	updateSwapEnabled	External	√	onlyOwner
	updateBuyFees	External	√	onlyOwner
	updateSellFees	External	√	onlyOwner
	excludeFromFees	Public	√	onlyOwner
	setAutomatedMarketMakerPair	Public	√	onlyOwner
	_setAutomatedMarketMakerPair	Private	√	
	updateMarketingWallet	External	√	onlyOwne
	updateDevWallet	External	√	r
	isExcludedFromFees	Public		onlyOwne
	rescueBNB	External	√	r -
	rescueAnyBEP20Tokens	Public	√	onlyOwne
	_transfer	Internal	√	r
	swapTokensForEth	Private	√	onlyOwne
	addLiquidity	Private	√	r
	swapBack	Private	√	
	setAutoLPBurnSettings	External	√	onlyOwner
	autoBurnLiquidityPairTokens	Internal	√	

Contract Flow



Domain Info

Domain Name	spidershiba.space
Registry Domain ID	2715570708_DOMAIN_NET-VRSN
Creation Date	2023-02-18T19:01:52Z
Updated Date	2023-02-18T21:03:35Z
Registry Expiry Date	2024-02-18T19:01:52Z
Registrar WHOIS Ser ver	whois.launchpad.com
Registrar URL	LaunchPad.com
Registrar	Launchpad, Inc. (HostGator)
Registrar IANA ID	955

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. 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 fees can be set up to 20% for buys and 25% for sales.

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 auditingauthorityfirm. 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://w w w.cyberscope.io