Cyberscope

Audit Report

Shiru

February 2022

Type	BEP20
Network	BSC
Address	0xA9E85F8E01e9BC1ed13bA341A6cF769EfA2A7087
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	Shiru
Compiler Version	v0.8.0+commit.c7dfd78e
Optimization	200 runs
Licence	
Explorer	https://bscscan.com/token/0xA9E85F8E01e9BC1ed13bA341A6cF769EfA2A7087
Symbol	SHIRU
Decimals	18
Total Supply	400,000,000,000,000
Source	contracts/Shiru.sol, contracts/DividendPayingToken.sol, contracts/Ownable.sol, contracts/IUniswapV2Pair.sol, contracts/IUniswapV2Factory.sol, contracts/IUniswapV2Router.sol, contracts/ERC20.sol, contracts/SafeMath.sol, contracts/SafeMathUint.sol, contracts/SafeMathInt.sol, contracts/DividendPayingTokenInterface.sol, contracts/DividendPayingTokenOptionalInterface.sol, contracts/IterableMapping.sol, contracts/IERC20.sol, contracts/IERC20Metadata.sol, contracts/Context.sol
Domain	shirupal.com

Audit Updates

Initial Audit	1st March 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	medium
Location	contract.sol#L348

Description

The contract owner has the authority to stop transactions for all users excluding the owner. The owner may take advantage of it by setting `maxWalletHoldingPercent` to zero.

```
  if(!_isExcludedFromMaxWallet[to]) {
            require(balanceOf(to) + amount <= totalSupply() *
maxWalletHoldingPercent / 100);
        }
```

Recommendation

The contract could embody a check for not allowing setting the maxWalletHoldingPercent less than a reasonable amount. A suggested implementation could check that the maximum amount should be more 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#L180,185,190

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 `setMarketingFee` function with a high percentage value.

```
  function setMarketingFee(uint256 value) external onlyOwner{
        marketingFee = value;
        totalFees = dividendFee + liquidityFee + marketingFee;
}
```

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
●	L01	Public Function could be Declared External
●	L02	State Variables could be Declared Constant
●	L12	Using Variables before Declaration
●	L07	Missing Events Arithmetic
●	L15	Local Scope Variable Shadowing
●	L14	Uninitialized Variables in Local Scope

L01 - Public Function could be Declared External

Criticality	minor
Location	contracts/DividendPayingToken.sol#L46,61,86,100,279,325,372 contracts/ERC20.sol#L63,71,88,114,122,133,151,173,192

Description

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

```
dividendTokenBalanceOf
withdrawableDividendOf
isExcludedFromMaxWallet
isExcludedFromFees
updateGasForProcessing
setAutomatedMarketMakerPair
setSwapTokensAtAmount
setSwapEnabled
excludeMultipleAccountsFromFees
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contracts/Shiru.sol#L32,23

Description

Constant state variables should be declared constant to save gas.

```
deadWallet
burnFee
```

Recommendation

Add the constant attribute to state variables that never change.

L12 - Using Variables before Declaration

Criticality	minor
Location	contracts/Shiru.sol#L364

Description

The contract is using a variable before the declaration. This is usually happening either if it has not been declared yet or the variable has been declared in a different scope.

```
iterations
claims
lastProcessedIndex
```

Recommendation

The variables should be declared before any usage of them.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contracts/Shiru.sol#L180,185,190,200

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.

```
swapTokensAtAmount = newAmount
marketingFee = value
liquidityFee = value
dividendFee = value
```

Recommendation

Emit an event for critical parameter changes.

L15 - Local Scope Variable Shadowing

Criticality	minor
Location	contracts/DividendPayingToken.sol#L44,86,93,100,110 contracts/Shiru.sol#L87

Description

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

```
operator
_owner
_symbol
_name
```

Recommendation

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

L14 - Uninitialized Variables in Local Scope

Criticality	minor
Location	contracts/Shiru.sol#L364

Description

The are variables that are defined in the local scope and are not initialized.

```
iterations
lastProcessedIndex
claims
```

Recommendation

All the local scoped variables should be initialized.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
DividendPayingToken	Implementation	ERC20, Ownable, DividendPayingTokenInterface, DividendPayingTokenOptionalInterface		
	<Constructor>	Public	✓	ERC20 Ownable
	distributeDividends	Public	✓	onlyOwner
	withdrawDividend	Public	✓	-
	_withdrawDividendOfUser	Internal	✓	
	dividendOf	Public		-
	withdrawableDividendOf	Public		-
	withdrawnDividendOf	Public		-
	accumulativeDividendOf	Public		-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_setBalance	Internal	✓	
DividendTracker	Implementation	DividendPayingToken		
	<Constructor>	Public	✓	DividendPayingToken
	setMinimumTokenBalanceForDividends	External	✓	onlyOperator

	_transfer	Internal		
	withdrawDividend	Public		-
	excludeFromDividends	External	✓	onlyOperator
	updateClaimWait	External	✓	onlyOperator
	getLastProcessedIndex	External		-
	getNumberOfTokenHolders	External		-
	getAccount	Public		-
	getAccountAtIndex	Public		-
	canAutoClaim	Private		
	setBalance	External	✓	onlyOwner
	process	Public	✓	-
	sendTokens	Public	✓	onlyOperator
	processAccount	Public	✓	onlyOwner
	<Receive Ether>	External	Payable	-
DividendPayingTokenInterface	Interface			
	dividendOf	External		-
	withdrawDividend	External	✓	-
DividendPayingTokenOptionalInterface	Interface			
	withdrawableDividendOf	External		-
	withdrawnDividendOf	External		-
	accumulativeDividendOf	External		-
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	✓	
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
IterableMapping	Library			
	get	Public		-
	getIndexOfKey	Public		-
	getKeyAtIndex	Public		-
	size	Public		-
	set	Public	✓	-
	remove	Public	✓	-
IUniswapV2Fa	Interface			

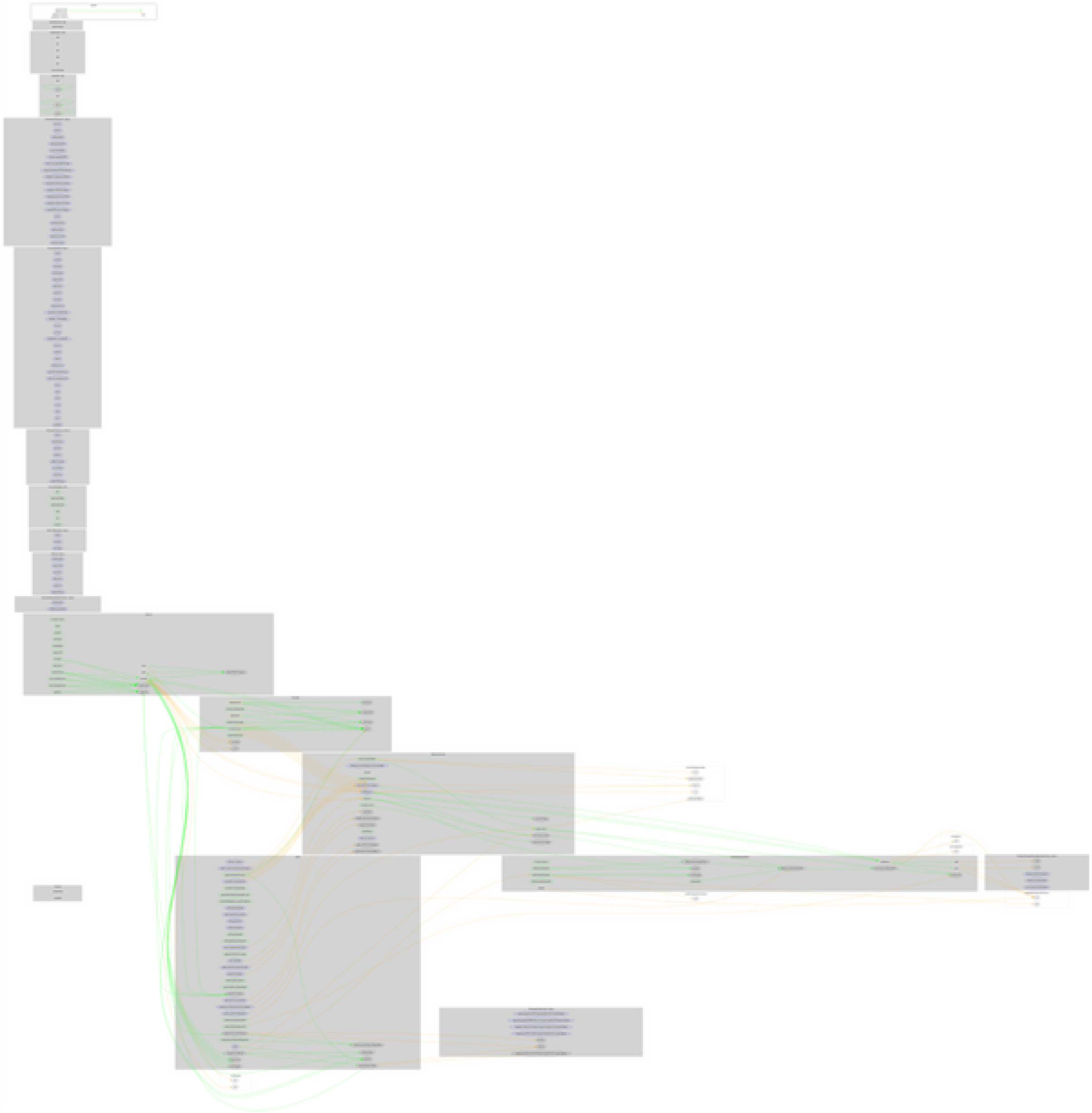
ctory				
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
IUniswapV2Pair	Interface			
	name	External		-
	symbol	External		-
	decimals	External		-
	totalSupply	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	✓	-
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	✓	-

Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	operator	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOperator
	transferOperator	Public	✓	onlyOperator
	_setOwner	Private	✓	
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		
SafeMathInt	Library			
	mul	Internal		
	div	Internal		
	sub	Internal		
	add	Internal		
	abs	Internal		
	toUint256Safe	Internal		
SafeMathUint	Library			
	toInt256Safe	Internal		
Shiru	Implementation	ERC20, Ownable		
	<Constructor>	Public	✓	ERC20 Ownable
	<Receive Ether>	External	Payable	-

	updateDividendTracker	Public	✓	onlyOwner
	updateUniswapV2Router	Public	✓	onlyOwner
	excludeFromFees	Public	✓	onlyOwner
	excludeFromMaxWallet	Public	✓	onlyOwner
	updateMaxWalletHoldingPercent	Public	✓	onlyOwner
	excludeMultipleAccountsFromFees	Public	✓	onlyOwner
	setMarketingWallet	External	✓	onlyOwner
	setDividendRewardsFee	External	✓	onlyOwner
	setLiquidityFee	External	✓	onlyOwner
	setMarketingFee	External	✓	onlyOwner
	setSwapEnabled	Public	✓	onlyOwner
	setSwapTokensAtAmount	Public	✓	onlyOwner
	setAutomatedMarketMakerPair	Public	✓	onlyOwner
	_setAutomatedMarketMakerPair	Private	✓	
	updateGasForProcessing	Public	✓	onlyOwner
	updateClaimWait	External	✓	onlyOwner
	getClaimWait	External		-
	getTotalDividendsDistributed	External		-
	isExcludedFromFees	Public		-
	isExcludedFromMaxWallet	Public		-
	withdrawableDividendOf	Public		-
	dividendTokenBalanceOf	Public		-
	excludeFromDividends	External	✓	onlyOwner
	getAccountDividendsInfo	External		-
	getAccountDividendsInfoAtIndex	External		-
	processDividendTracker	External	✓	-
	claim	External	✓	-
	getLastProcessedIndex	External		-
	getNumberOfDividendTokenHolders	External		-
	_transfer	Internal	✓	
	swapAndSendToAll	Private	✓	
	swapTokensForEth	Private	✓	
	addLiquidity	Private	✓	

Contract Flow



Domain Info

Domain Name	shirupal.com
Registry Domain ID	2647102432_DOMAIN_COM-VRSN
Creation Date	2021-10-11 18:17:21
Updated Date	2021-10-11 18:17:22
Registry Expiry Date	
Registrar WHOIS Server	whois.domains.co.za
Registrar URL	https://www.domains.co.za
Registrar	DIAMATRIX C.C.
Registrar IANA ID	1645

The domain has been created 5 months before the creation of the audit.

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

Summary

Shiru is an interesting project with a friendly and growing community. There are some functions that can be abused by the owner, like manipulating fees up to 100% and stopping transactions for everyone but the owner. 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