Cyberscope

Audit Report

RUNSTEPN

March 2022

Type	BEP20
Network	BSC
Address	0xfd6B30b9E2c7eb0715e4f516FA79A1f6fB98dEbf
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	RUNTOKEN
Compiler Version	v0.8.6+commit.11564f7e
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0xfd6B30b9E2c7eb0715e4f516FA79A1f6fB98dEbf
Symbol	RUN
Decimals	18
Total Supply	100,000,000
Source	contract.sol
Domain	runstepn.com

Audit Updates

Initial Audit	19th 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

ELFM - Exceed Limit Fees Manipulation

Criticality	critical
Location	contract.sol#L1403,1407,1411,1415

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 `updateTaxFee` function with a high percentage value.

```
function updateTaxFee(uint256 taxFee) external onlyOwner{
    _taxFee = taxFee;
}
```

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.

BC - Blacklisted Contracts

Criticality	medium
Location	contract.sol#L1108

Description

The contract owner has the authority to stop contracts from transactions. The owner may take advantage of it by calling the `updateMarketList` function.

```
require( !_marketList[from], "ERC20: market is not enabled");
require( !_marketList[to], "ERC20: market is not enabled");
```

Recommendation

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
	BLC	Business Logic Concern
	FSA	Fixed Swap Address
	CO	Code Optimization
	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
	L09	Dead Code Elimination
	L13	Divide before Multiply Operation

BLC - Business Logic Concern

Criticality	minor
Location	contract.sol#L1142

Description

The business logic seems peculiar. The implementation may not follow the expected behavior.

The fromAddress and toAddress are class properties. That means that their value persists across transfers. Hence, the setShare is called with the addresses of the previous transfer instead of the current.

```
if (shouldSetInviter) {
    inviter[to] = from;
}
if (fromAddress == address(0))fromAddress = from;
if (toAddress == address(0))toAddress = to;
if (!isDividendExempt[fromAddress] && fromAddress != uniswapV2Pair)
setShare(fromAddress);
if (!isDividendExempt[toAddress] && toAddress != uniswapV2Pair)
setShare(toAddress);

fromAddress = from;
toAddress = to;
```

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#L922

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.

```
IUniswapV2Router02 _uniswapV2Router = IUniswapV2Router02(
    0x10ED43C718714eb63d5aA57B78B54704E256024E
);

// Create a uniswap pair for this new token
uniswapV2Pair =
IUniswapV2Factory(_uniswapV2Router.factory()).createPair(address(this),
USDTAddress);
```

Recommendation

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

CO - Code Optimization

Criticality	minor
Location	contract.sol#L1282

Description

There are code segments that could be optimized. A segment may be optimized so that it becomes a smaller size, consumes less memory, executes more rapidly, or performs fewer operations.

- The *accurRate* is calculated in the loop even if the result does not affected by the flow. The *accurRate* number is always fixed to 700.
- The total fees that are added to the cur balance are fixed to 7%.
- The loop is executed even if the cur does not exist in the inviter map.

```
for (int256 i = 0; i < 7; i++) {
    uint256 rate;
    if (i == 0) {
    rate = 300;
    } else if (i == 1 || i == 2){
    rate = 100;
    } else {
    rate = 50;
    }
    cur = inviter[cur];
    if (cur == address(0)) {
        break;
    }
    accurRate = accurRate.add(rate);

    uint256 curTAmount = tAmount.div(10000).mul(rate);
    _tOwned[cur] = _tOwned[cur].add(curTAmount);
    emit Transfer(sender, cur, curTAmount);
}
```

Recommendation

Rewrite some code segments so the runtime will be more performant.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L106,115,120,945,949,953,957,965,974,983 and 15 more

Description

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

```
setSwapAndLiquifyEnabled
emergencyUSDTWithdraw
emergencyBNBWithdraw
emergencyLPWithdraw
emergencyTokenWithdraw
updateReceiver
includeInFee
includeReward
excludeReward
...
```

Recommendation

Use the external attribute for functions never called from the contract

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L96,95,862,860,861,876

Description

Constant state variables should be declared constant to save gas.

```
_tTotal
_symbol
_name
_decimals
_previousOwner
_lockTime
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L560,562,593,639,1062,1364,1371,1381,1391,1395 and 10 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.

```
LPFeeDividends
_inviterFee
_taxFee
_LPFee
_burnFee
USDTAddress
_enabled
_dividendsMinAmount
LPFee
...
```

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#L95,96

Description

There are segments that contain unused state variables.

```
_lockTime
_previousOwner
```

Recommendation

Remove unused state variables.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L1391,1395,1399,1403,1407,1411,1415,1419

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.

```
dividendsMinAmount = _dividendsMinAmount
_inviterFee = inviterFee
_taxFee = taxFee
_LPFee = LPFee
_burnFee = burnFee
minPeriod = _minPeriod
distributorGas = _distributorGas
minTokenNumberToSell = _minTokenNumberToSell
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L321,350,364,295,465,456,424

Description

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

```
safeTransferFrom
safeIncreaseAllowance
safeDecreaseAllowance
sendValue
functionCallWithValue
functionCall
```

Recommendation

Remove unused functions.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L1282,1427

Description

Performing divisions before multiplications may cause lose of prediction.

```
Transfer(sender,recipient,tAmount.div(10000).mul(recipientRate))
_tOwned[recipient] =
_tOwned[recipient].add(tAmount.div(10000).mul(recipientRate))
_takeTaxFee(sender,tAmount.div(10000).mul(_taxFee))
_takeLPFee(sender,tAmount.div(10000).mul(_LPFee))
_takeburnFee(sender,tAmount.div(10000).mul(_burnFee))
curTAmount = tAmount.div(10000).mul(rate)
Transfer(sender,address(this),tAmount.div(10000).mul(_inviterFee.sub(accurRate))
)
_tOwned[address(this)] =
_tOwned[address(this)].add(tAmount.div(10000).mul(_inviterFee.sub(accurRate)))
Transfer(sender,address(this),tAmount.div(10000).mul(_inviterFee))
...
```

Recommendation

The multiplications should be prior to the divisions.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
Ownable	Implementation			
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
Address	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	

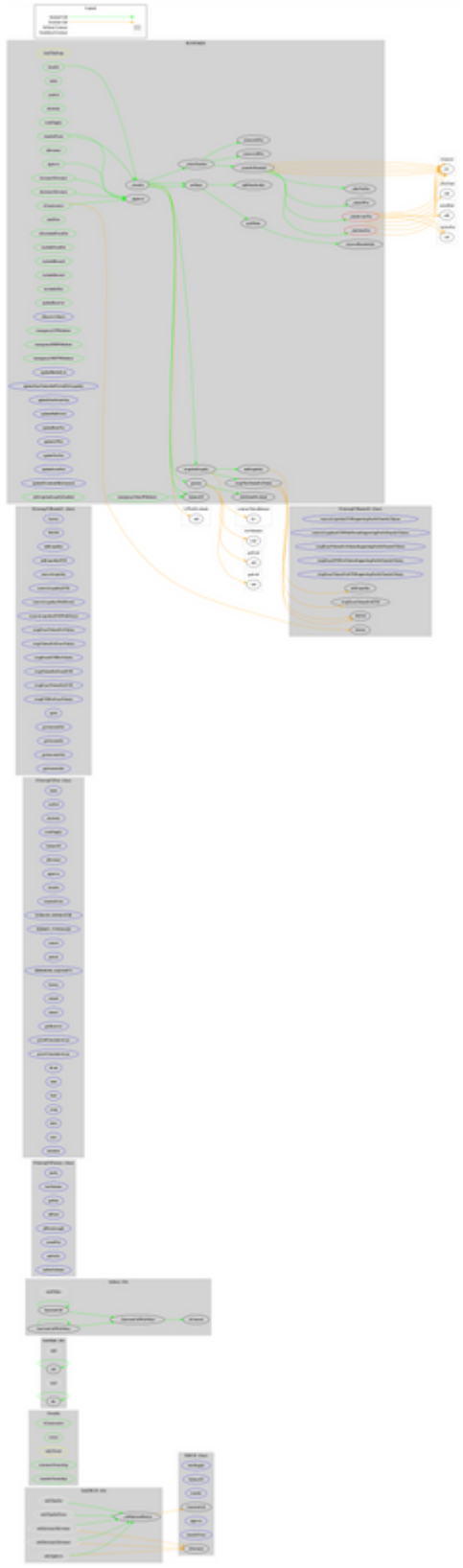
	_functionCallWithValue	Private	✓	
SafeERC20	Library			
	safeTransfer	Internal	✓	
	safeTransferFrom	Internal	✓	
	safeApprove	Internal	✓	
	safeIncreaseAllowance	Internal	✓	
	safeDecreaseAllowance	Internal	✓	
	_callOptionalReturn	Private	✓	
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	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	✓	-
RUNTOKEN	Implementation	IERC20, Ownable		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	totalFees	Public		-
	isExcludedFromFee	Public		-
	excludeFromFee	Public	✓	onlyOwner
	excludeReward	Public	✓	onlyOwner
	includeReward	Public	✓	onlyOwner
	includeInFee	Public	✓	onlyOwner
	updateReceiver	Public	✓	onlyOwner
	<Receive Ether>	External	Payable	-
	removeAllFee	Private	✓	
	restoreAllFee	Private	✓	

	_approve	Private	✓	
	_transfer	Private	✓	
	swapAndLiquify	Private	✓	lockTheSwap
	process	Private	✓	
	distributeDividend	Internal	✓	
	setShare	Private	✓	
	addShareholder	Internal	✓	
	quitShare	Private	✓	
	removeShareholder	Internal	✓	
	_tokenTransfer	Private	✓	
	_takeburnFee	Private	✓	
	_takeLPFee	Private	✓	
	_takeTaxFee	Private	✓	
	_takeInviterFee	Private	✓	
	swapThisTokenForToken	Private	✓	
	addLiquidity	Private	✓	
	emergencyTokenWithdraw	Public	✓	onlyOwner
	emergencyLPWithdraw	Public	✓	onlyOwner
	emergencyBNBWithdraw	Public	✓	onlyOwner
	emergencyUSDTWithdraw	Public	✓	onlyOwner
	updateMarketList	External	✓	onlyOwner
	updateNumTokensSellToAddToLiquidity	External	✓	onlyOwner
	updateDistributorGas	External	✓	onlyOwner
	updateMinPeriod	External	✓	onlyOwner
	updateBurnFee	External	✓	onlyOwner
	updateLPFee	External	✓	onlyOwner
	updateTaxFee	External	✓	onlyOwner
	updateInviteFee	External	✓	onlyOwner
	updateDividendsMinAmount	External	✓	onlyOwner
	setSwapAndLiquifyEnabled	Public	✓	onlyOwner
	_transferStandard	Private	✓	

Contract Flow



Domain Info

Domain Name	runstepn.com
Registry Domain ID	2682522031_DOMAIN_COM-VRSN
Creation Date	2022-03-18T12:11:01Z
Updated Date	2022-03-18T12:11:01Z
Registry Expiry Date	2023-03-18T12:11:01Z
Registrar WHOIS Server	whois.name.com
Registrar URL	http://www.name.com
Registrar	Name.com, Inc.
Registrar IANA ID	625

The domain has been created about 23 hours 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 and blacklisting addresses. There are also some concerns regarding the business logic of the contract. 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