



Audit Report

Solium Rise

January 2022

Type	BEP20
Network	BSC
Address	0xD169fAF8119Ca3B339a323D0911c14947a603f08
Audited by	© coinscope

Table of Contents

Table of Contents	1
Contract Review	3
Audit Updates	3
Contract Analysis	4
ST - Stop Transactions	5
Description	5
Recommendation	5
ELFM - Exceed Limit Fees Manipulation	7
Description	7
Recommendation	7
BC - Blacklisted Contracts	8
Description	8
Recommendation	8
Contract Diagnostics	9
L01 - Public Function could be Declared External	10
Description	10
Recommendation	10
L02 - State Variables could be Declared Constant	11
Description	11
Recommendation	11
L04 - Conformance to Solidity Naming Conventions	12
Description	12
Recommendation	12
L07 - Missing Events Arithmetic	13
Description	13
Recommendation	13

Contract Functions	14
Contract Flow	18
Domain Info	19
Summary	20
Disclaimer	21
About Coinscope	22

Contract Review

Contract Name	SOLIUMRISE
Compiler Version	v0.7.6+commit.7338295f
Optimization	500 runs
Licence	Unlicense
Explorer	https://bscscan.com/token/0xD169fAF8119Ca3B339a323D0911c14947a603f08
Symbol	SOLAR
Decimals	4
Total Supply	50,000,000
Source	contract.sol
Domain	soliumrise.io

Audit Updates

Initial Audit	20th January 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#L521,L466

Description

The contract owner has the authority to stop the sales. The owner may take advantage of it by setting a high value to the `sellMultiplier`.

```
uint256 multiplier = isSell ? sellMultiplier : 100;
uint256 feeAmount = amount.mul(totalFee).mul(multiplier).div(feeDenominator * 100);
```

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

```
require(amount <= _maxTxAmount || isTxLimitExempt[sender], "TX Limit Exceeded");
```

```
if(!authorizations[sender] && !authorizations[recipient]){
    require(tradingOpen, "Trading not open yet");
}
```

Recommendation

The contract could embody a check for not allowing setting the `_maxTxAmount` 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 contract should not allow setting the `tradingOpen` variable to `false` once it's `true`.

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.

ELFM - Exceed Limit Fees Manipulation

Criticality	critical
Location	contract.sol#L564,L663

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` or the `set_sell_multiplier` function with a high percentage value.

```
function setFees(uint256 _liquidityFee, uint256 _reflectionFee, uint256
_marketingFee, uint256 _teamFee, uint256 _burnFee, uint256 _charityFee, uint256
_feeDenominator) external authorized {
    liquidityFee = _liquidityFee;
    reflectionFee = _reflectionFee;
    marketingFee = _marketingFee;
    teamFee = _teamFee;
    charityFee = _charityFee;
    burnFee = _burnFee;
    totalFee = _liquidityFee + _reflectionFee + _marketingFee + _teamFee +
    _burnFee + _charityFee;
    feeDenominator = _feeDenominator;
    require(totalFee < feeDenominator/2, "Fees cannot be more than 50%");
}
```

```
function set_sell_multiplier(uint256 Multiplier) external onlyOwner{
    sellMultiplier = Multiplier;
}
```

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	critical
Location	contract.sol#L644

Description

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

```
function manage_blacklist(address[] calldata addresses, bool status) public  
onlyOwner {  
    for (uint256 i; i < addresses.length; ++i) {  
        isBlacklisted[addresses[i]] = status;  
    }  
}
```

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
●	L01	Public Function could be Declared External
●	L02	State Variables could be Declared Constant
●	L04	Conformance to Solidity Naming Conventions
●	L07	Missing Events Arithmetic

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L644,L640,L574 and 4 more

Description

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

```
manage_blacklist  
enable_blacklist  
cooldownEnabled  
...
```

Recommendation

Use the external attribute for functions never called from the contract

L02 - State Variables could be Declared Constant

Criticality

minor

Location

contract.sol#L337,L331,L329 and 3 more

Description

Constant state variables should be declared constant to save gas.

```
_totalSupply  
ZERO  
WBNB  
...
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality

minor

Location

contract.sol#L343,L342,L340 and 46 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  
_maxWalletToken  
...
```

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#L688,L683,L663 and 4 more

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  
liquidityFee = _liquidityFee  
...
```

Recommendation

Emit an event for critical parameter changes.

Contract Functions

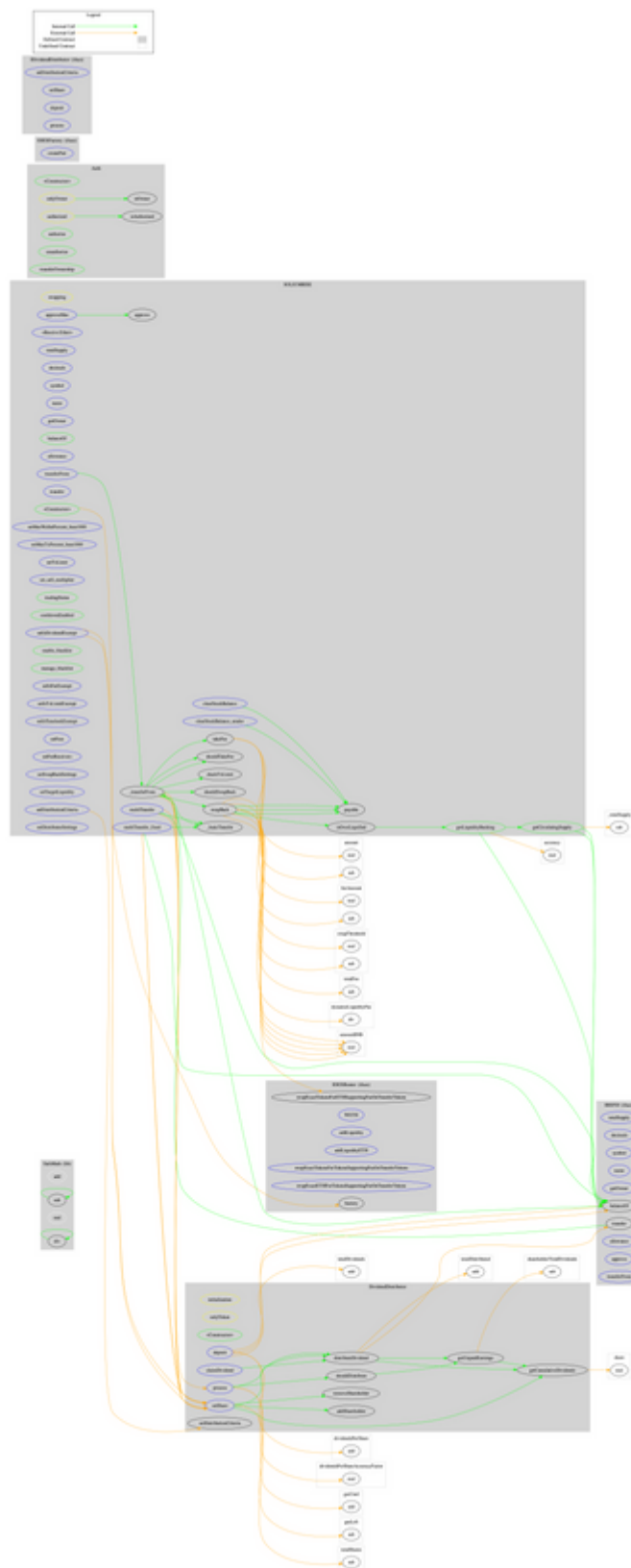
Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
IBEP20	Interface			
	totalSupply	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	getOwner	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
Auth	Implementation			
	<Constructor>	Public	✓	-
	authorize	Public	✓	onlyOwner
	unauthorize	Public	✓	onlyOwner
	isOwner	Public		-
	isAuthorized	Public		-
	transferOwnership	Public	✓	onlyOwner
IDEXFactory	Interface			

	createPair	External	✓	-
IDEXRouter	Interface			
	factory	External		-
	WETH	External		-
	addLiquidity	External	✓	-
	addLiquidityETH	External	Payable	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
IDividendDistributor	Interface			
	setDistributionCriteria	External	✓	-
	setShare	External	✓	-
	deposit	External	Payable	-
	process	External	✓	-
DividendDistributor	Implementation	IDividendDistributor		
	<Constructor>	Public	✓	-
	setDistributionCriteria	External	✓	onlyToken
	setShare	External	✓	onlyToken
	deposit	External	Payable	onlyToken
	process	External	✓	onlyToken
	shouldDistribute	Internal		
	distributeDividend	Internal	✓	
	claimDividend	External	✓	-
	getUnpaidEarnings	Public		-
	getCumulativeDividends	Internal		
	addShareholder	Internal	✓	
	removeShareholder	Internal	✓	
SOLIUMRISE	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	✓	-
	setMaxWalletPercent_base1000	External	✓	onlyOwner
	setMaxTxPercent_base1000	External	✓	onlyOwner
	setTxLimit	External	✓	authorized
	_transferFrom	Internal	✓	
	_basicTransfer	Internal	✓	
	checkTxLimit	Internal		
	shouldTakeFee	Internal		
	takeFee	Internal	✓	
	shouldSwapBack	Internal		
	clearStuckBalance	External	✓	authorized
	clearStuckBalance_sender	External	✓	authorized
	set_sell_multiplier	External	✓	onlyOwner
	tradingStatus	Public	✓	onlyOwner
	cooldownEnabled	Public	✓	onlyOwner
	swapBack	Internal	✓	swapping
	setIsDividendExempt	External	✓	authorized
	enable_blacklist	Public	✓	onlyOwner
	manage_blacklist	Public	✓	onlyOwner
	setIsFeeExempt	External	✓	authorized
	setIsTxLimitExempt	External	✓	authorized
	setIsTimelockExempt	External	✓	authorized

	setFees	External	✓	authorized
	setFeeReceivers	External	✓	authorized
	setSwapBackSettings	External	✓	authorized
	setTargetLiquidity	External	✓	authorized
	setDistributionCriteria	External	✓	authorized
	setDistributorSettings	External	✓	authorized
	getCirculatingSupply	Public		-
	getLiquidityBacking	Public		-
	isOverLiquified	Public		-
	multiTransfer	External	✓	onlyOwner
	multiTransfer_fixed	External	✓	onlyOwner

Contract Flow



Domain Info

Domain Name	soliumrise.io
Registry Domain ID	957e050cd06f41608dc7951f82f6ea58-DONUTS
Creation Date	2021-11-16T11:18:37Z
Updated Date	2021-12-10T21:48:07Z
Registry Expiry Date	2024-11-16T11:18:37Z
Registrar WHOIS Server	whois.namecheap.com
Registrar URL	https://www.namecheap.com/
Registrar	NameCheap, Inc.
Registrar IANA ID	1068

The domain has been created 2 months before the creation of the audit. It will expire in almost 3 years.

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, stopping transactions and blacklisting contracts. The contract also has the ability to stop all sales. 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.

Coinscope 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 Coinscope 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 Coinscope 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 Coinscope team disclaims any liability for the resulting losses.

About Coinscope

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.

Coinscope is aiming to make crypto discoverable and efficient globally. It provides all the essential tools to assist users draw their own conclusions.



The Coinscope.co team

<https://www.coinscope.co>