COINSCOPE

Audit Report

Sgt.SHIB

February 2022

Type	BEP20
Network	BSC
Address	0x694001c8f994D9AF3d6755D411d4e2B327adF4d4
Audited by	© coinscope

Table of Contents

Contract Review

Contract Name	SgtSHIB
Compiler Version	v0.7.6+commit.7338295f
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0x694001c8f994D9AF3d6755D411d4e2B327adF4d4
Symbol	SGTS
Decimals	9
Total Supply	150,000,000,000,000
Source	contract.sol
Domain	sgtshib.com

Audit Updates

Initial Audit	9th February 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#L1084,1087,1097

Description

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

```
if(!swapping && automatedMarketMakerPairs[to] && from !=
address(uniswapV2Router) && !_isExcludedFromFees[to])
{
    require(amount <= maxSellTransactionAmount, "Amount exceeds the
Protocol-X");
}
```

The contract owner has the authority to stop transactions for all buys excluding the owner. The owner may take advantage of it by setting the `contractBalanceRecepient` or `maxBuyTranscationAmount` to zero.

```
require(contractBalanceRecepient + amount <= _maxWalletToken, "Exceeds maximum
wallet token amount.");
```

```
require(amount <= maxBuyTranscationAmount, "Amount exceeds the Protocol-X.");
```

Recommendation

The contract could embody a check for not allowing setting the *maxSellTransactionAmount*, *contractBalanceRecepient*, and *maxBuyTranscationAmount* 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.

OCTD - Owner Contract Tokens Drain

Criticality	medium
Location	contract.sol#L1303

Description

The contract owner has the authority to claim all the balance of the contract. The owner may take advantage of it by calling the *withdrawRemainingToken* function. The swapTokensAtAmount can also be combined with the withdraw function. The contract owner can increase the *swapTokensAtAmount*, so the accumulated fees will enter in the swap and liquify feature.

```
function withdrawRemainingToken(address account) public onlyOwner {
    uint256 balance = balanceOf(address(this));
    super._transfer(address(this), account, balance);
}
```

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.

BC - Blacklisted Contracts

Criticality	critical
Location	contract.sol#L769

Description

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

```
function addToBlackList(address[] calldata addresses) external onlyOwner {
    for (uint256 i; i < addresses.length; ++i) {
    _isBlacklisted[addresses[i]] = true;
    }
}
```

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
●	L05	Unused State Variable
●	L04	Conformance to Solidity Naming Conventions
●	L09	Dead Code Elimination
●	L11	Unnecessary Boolean equality
●	L12	Using Variables before Declaration
●	L07	Missing Events Arithmetic
●	L15	Local Scope Variable Shadowing
●	L14	Uninitialized Variables in Local Scope
●	L08	Tautology or Contradiction
●	L13	Divide before Multiply Operation

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L251,255,262,268,323,328 and 36 more

Description

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

```
process
getAccountAtIndex
burnRemainingToken
...
```

Recommendation

Use the external attribute for functions never called from the contract

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L615,795,799

Description

Constant state variables should be declared constant to save gas.

```
marketingDivisor
_rateLimitSeconds
lastAmount
```

Recommendation

Add the constant attribute to state variables that never change.

L05 - Unused State Variable

Criticality	minor
Location	contract.sol#L803,615

Description

There are segments that contain unused state variables.

```
lastAmount
DefaultLiquidityLockTime
```

Recommendation

Remove unused state variables.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L40,204,205,222,416,698 and 20 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.

```
_account
_isExcludedMaxSellTransactionAmount
DefaultLiquidityLockTime
...
```

Recommendation

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

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L31,715,521,596,586,601 and 7 more

Description

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

```
mul
div
trySub
...
```

Recommendation

Remove unused functions.

L11 - Unnecessary Boolean equality

Criticality	minor
Location	contract.sol#L1069

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.

```
getAntiBotEnabled() == true
```

Recommendation

Remove the equality to the boolean constant.

L12 - Using Variables before Declaration

Criticality	minor
Location	contract.sol#L1151

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
lastProcessedIndex
claims
```

Recommendation

The variables should be declared before any usage of them.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L907,911,915,919,1288,1293 and 1 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.

```
liquidityFee = newFee
marketingFee = newFee
swapTokensAtAmount = amountOfTokens
...
```

Recommendation

Emit an event for critical parameter changes.

L15 - Local Scope Variable Shadowing

Criticality	minor
Location	contract.sol#L627

Description

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

```
_symbol
_name
```

Recommendation

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

L14 - Uninitialized Variables in Local Scope

Criticality	minor
Location	contract.sol#L1151,770

Description

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

```
claims
i
iterations
...
```

Recommendation

All the local scoped variables should be initialized.

L08 - Tautology or Contradiction

Criticality	minor
Location	contract.sol#L1288,1293,1298

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)(newFee >= 0 && newFee <= 15,SgtSHIB: ETH/BNB Rewards tax
must be between 0 and 15)
require(bool,string)(newFee >= 0 && newFee <= 10,SgtSHIB: Liquidity tax must be
between 0 and 10)
require(bool,string)(newFee >= 0 && newFee <= 10,SgtSHIB: Expenses tax must be
between 0 and 10)
```

Recommendation

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

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L1069

Description

Performing divisions before multiplications may cause lose of prediction.

```
fees = amount.mul(totalFees).div(100)
transferToMarketingWallet(address(marketingWallet),address(this).balance.div(10
** 2).mul(marketingDivisor))
```

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	✓	-
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
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	✓	-
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	✓	-
IterableMapping	Library			
	get	Public		-
	getIndexOfKey	Public		-
	getKeyAtIndex	Public		-
	size	Public		-
	set	Public	✓	-
	remove	Public	✓	-
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner

IDividendPayingTokenOptional	Interface			
	withdrawableDividendOf	External		-
	withdrawnDividendOf	External		-
	accumulativeDividendOf	External		-
IDividendPayingToken	Interface			
	dividendOf	External		-
	distributeDividends	External	Payable	-
	withdrawDividend	External	✓	-
SafeMathInt	Library			
	mul	Internal		
	div	Internal		
	sub	Internal		
	add	Internal		
	toUint256Safe	Internal		
SafeMathUint	Library			
	toInt256Safe	Internal		
ERC20	Implementation	Context, IERC20		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-

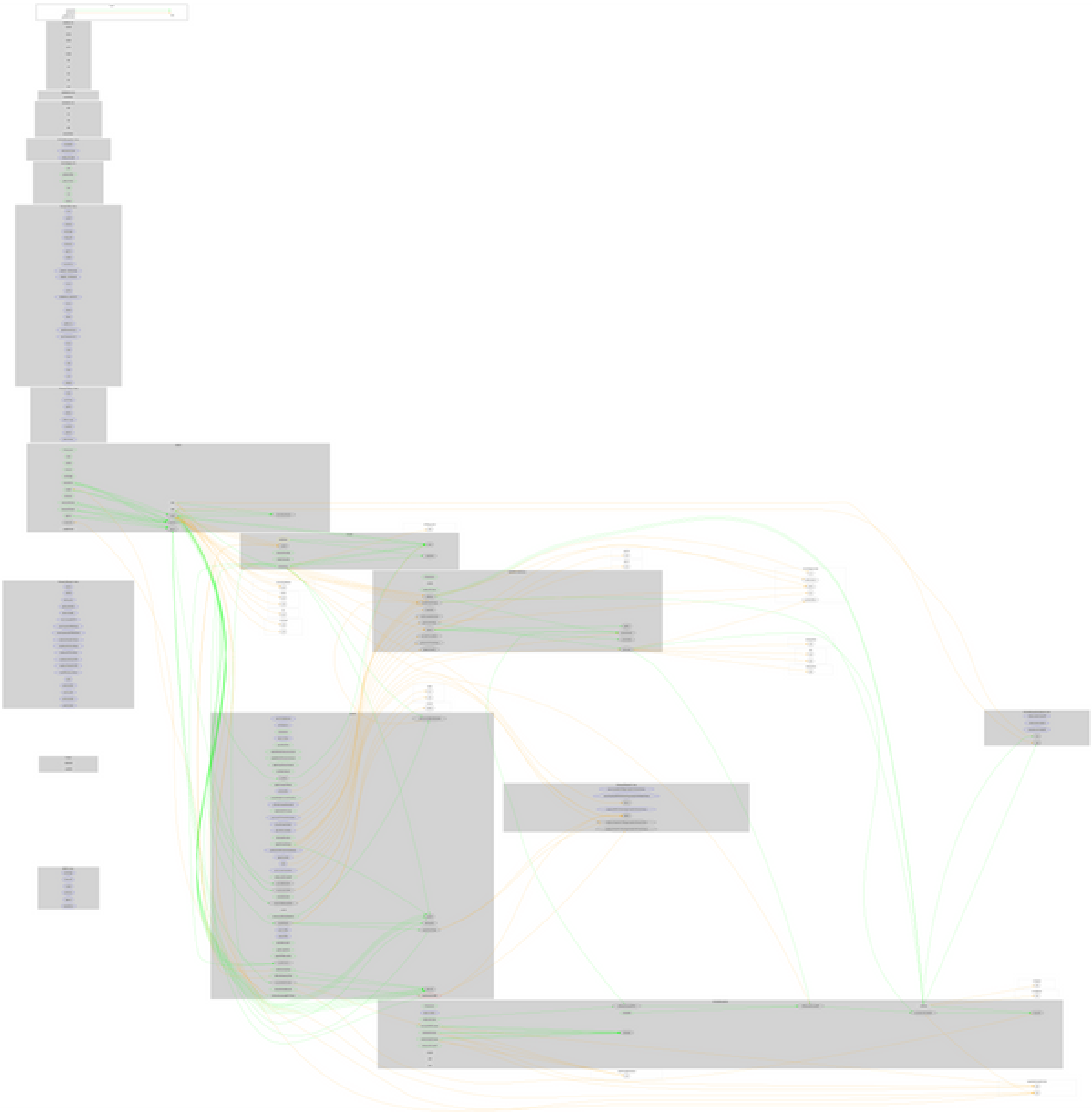
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_setupDecimals	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
SafeMath	Library			
	tryAdd	Internal		
	trySub	Internal		
	tryMul	Internal		
	tryDiv	Internal		
	tryMod	Internal		
	add	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	mod	Internal		
	sub	Internal		
	div	Internal		
	mod	Internal		
DividendPayingToken	Implementation	ERC20, IDividendPayingToken, IDividendPayingTokenOptional		
	<Constructor>	Public	✓	ERC20
	<Receive Ether>	External	Payable	-
	distributeDividends	Public	Payable	-
	distributeSHIBDividends	Public	✓	-
	distributeOpt2Dividends	Public	✓	-
	withdrawDividend	Public	✓	-
	_withdrawDividendOfUser	Internal	✓	
	dividendOf	Public		-
	withdrawableDividendOf	Public		-

	withdrawnDividendOf	Public		-
	accumulativeDividendOf	Public		-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_setBalance	Internal	✓	
SgtSHIB	Implementation	ERC20, Ownable		
	removeFromBlackList	External	✓	onlyOwner
	addToBlackList	External	✓	onlyOwner
	<Constructor>	Public	✓	ERC20
	<Receive Ether>	External	Payable	-
	updateMaxWallet	Public	✓	onlyOwner
	updateMaxBuyTranscationAmount	Public	✓	onlyOwner
	updateMaxSellTransactionAmount	Public	✓	onlyOwner
	updateSwapTokensAtAmount	Public	✓	onlyOwner
	multiWalletTransfer	Public	✓	onlyOwner
	updateDividendTracker	Public	✓	onlyOwner
	updateUniswapV2Router	Public	✓	onlyOwner
	excludeFromFees	Public	✓	onlyOwner
	excludeMultipleAccountsFromFees	Public	✓	onlyOwner
	setAutomatedMarketMakerPair	Public	✓	onlyOwner
	_setAutomatedMarketMakerPair	Private	✓	
	updateGasForProcessing	Public	✓	onlyOwner
	updateClaimWait	External	✓	onlyOwner
	getClaimWait	External		-
	getTotalDividendsDistributed	External		-
	isExcludedFromFees	Public		-
	withdrawableDividendOf	Public		-
	dividendTokenBalanceOf	Public		-
	getAccountDividendsInfo	External		-
	getAccountDividendsInfoAtIndex	External		-
	processDividendTracker	External	✓	-
	claim	External	✓	-
	getLastProcessedIndex	External		-

	getNumberOfDividendTokenHolders	External		-
	setAntiBotEnabled	Public	✓	onlyOwner
	getAntiBotEnabled	Public		-
	_transfer	Internal	✓	
	swapAndLiquify	Private	✓	
	swapTokensForEth	Private	✓	
	swapTokensForSHIB	Private	✓	
	addLiquidity	Private	✓	
	swapAndSendDividends	Private	✓	
	removeAllFee	External	✓	onlyOwner
	restoreAllFee	External	✓	onlyOwner
	updateMarketingFee	Public	✓	onlyOwner
	updateLiquidityFee	Public	✓	onlyOwner
	updateETHRewardsFee	Public	✓	onlyOwner
	withdrawRemainingToken	Public	✓	onlyOwner
	withdrawRemainingBEP20Token	Public	✓	onlyOwner
	burnRemainingToken	Public	✓	onlyOwner
	checkBot	Private	✓	
	swapTokensForBNB	Private	✓	
	transferToMarketingWallet	Private	✓	
SgtSHIBDividendTracker	Implementation	DividendPayingToken, Ownable		
	<Constructor>	Public	✓	DividendPayingToken
	_transfer	Internal		
	withdrawDividend	Public		-
	excludeFromDividends	External	✓	onlyOwner
	updateClaimWait	External	✓	onlyOwner
	getLastProcessedIndex	External		-
	getNumberOfTokenHolders	External		-
	getAccount	Public		-
	getAccountAtIndex	Public		-
	canAutoClaim	Private		
	setBalance	External	✓	onlyOwner

	process	Public	✓	-
	processAccount	Public	✓	onlyOwner

Contract Flow



Domain Info

Domain Name	sgtshib.com
Registry Domain ID	2666167887_DOMAIN_COM-VRSN
Creation Date	2022-01-05T15:19:51Z
Updated Date	2022-01-06T14:54:49Z
Registry Expiry Date	
Registrar WHOIS Server	whois.launchpad.com
Registrar URL	LaunchPad.com
Registrar	Launchpad, Inc. (HostGator)
Registrar IANA ID	955

The domain has been created about 1 month before the creation of the audit.

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 massively blacklisting addresses, transferring funds to the team's wallet and stopping the transactions. The contract Owner can have the ability to stop the sales and the buys for all the users excluding him. If this functionality is abused by the contract owner, then the contract will operate as a honeypot. The fees are fixed, 12% for the buys and 15.6% for the 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