Audit Report

SpongeBob Square

January 2022

Type	BEP20
Network	BSC
Address	0x471f883bbd2c705f418ba3d6667ef05342c4ee05
Audited by	© coinscope

Table of Contents

Contract Review

Contract Name	SPONGS
Compiler Version	v0.8.6+commit.11564f7e
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0x471f883bbd2c705f418ba3d6667ef05342c4ee05
Symbol	SPONGS
Decimals	9
Total Supply	1,000,000,000,000,000
Source	contract.sol
Domain	spongebobsquare.com

Audit Updates

Initial Audit	13th 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#L1769,L1809

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 by disabling the `isTradingEnabled` variable.

```
if(!isTradingEnabled) {
    require(_isAllowedDuringDisabled[to] || _isAllowedDuringDisabled[from],
"Trading is currently disabled");
}
```

The contract owner has the authority to prevent the sale of the token and allow only the buy operations. The owner may take advantage of it by setting the maxSellTransactionAmount to zero.

```
if(automatedMarketMakerPairs[to]){

    require(amount <= maxSellTransactionAmount, "BEP20: Exceeds max sell
amount");
```

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 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#L1520

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 `updateFees` function with a high percentage value on a parameter like the `liquidityFee`.

```
function updateFees(uint256 bnbRewardPerc, uint256 liquidityPerc, uint256
operationsPerc, uint256 buyBackPerc) external onlyOwner {
    require (operationsPerc.add(buyBackPerc) <= liquidityPerc, "SPONGS:
updateFees:: Liquidity Perc must be equal to or higher than operations and
buyback combined.");
    emit FeesUpdated(bnbRewardPerc, liquidityPerc, operationsPerc, buyBackPerc);
    BNBRewardsFee = bnbRewardPerc;
    liquidityFee = liquidityPerc;
    operationsFee = operationsPerc;
    buyBackFee= buyBackPerc;

    totalFees = BNBRewardsFee.add(liquidityFee);

}
```

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#L1767

Description

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

```
require(!_isIgnoredAddress[to] || !_isIgnoredAddress[from], "SPONGS: To/from address is ignored");
```

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
	L05	Unused State Variable
	L04	Conformance to Solidity Naming Conventions
	L09	Dead Code Elimination
	L11	Unnecessary Boolean equality
	L07	Missing Events Arithmetic

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L2232,L2188,L2038 and 31 more

Description

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

```
process
getAccountAtIndex
size
...
```

Recommendation

Use the external attribute for functions never called from the contract

L05 - Unused State Variable

Criticality	minor
Location	contract.sol#L243

Description

There are segments that contains unused state variable.

```
MAX_INT256
```

Recommendation

Remove unused state variables.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L2143,L1279,L1273 and 12 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
_maxSellPercent
BNBRewardsFee
...
```

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#L248,L260,L289 and 4 more

Description

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

```
mul
div
abs
...
```

Recommendation

Remove unused functions.

L11 - Unnecessary Boolean equality

Criticality	minor
Location	contract.sol#L1684,L860

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.

```
require(bool,string)(isAMMWhitelisted(ammContractAddress) == true,SPONGS:
setRewardToken:: AMM is not whitelisted!)
userHasCustomRewardToken[holder] == true
```

Recommendation

Remove the equality to the boolean constant.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L2006,L1429,L1399

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.

```
_maxSellPercent = maxSellPercent
maxSellTransactionAmount = maxTxnAmount
swapTokensAtAmount = newAmount
```

Recommendation

Emit an event for critical parameter changes.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
DividendPayingTokenOptionalInterface	Interface			
	withdrawableDividendOf	External		-
	withdrawnDividendOf	External		-
	accumulativeDividendOf	External		-

DividendPayingTokenInterface	Interface			
	dividendOf	External		-
	distributeDividends	External	Payable	-
	withdrawDividend	External	✓	-
SafeMathInt	Library			
	mul	Internal		
	div	Internal		
	sub	Internal		
	add	Internal		
	abs	Internal		
	toUint256Safe	Internal		
SafeMathUint	Library			
	toInt256Safe	Internal		
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		
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	✓	
DividendPayingToken	Implementation	ERC20, DividendPayingTokenInterface, DividendPayingTokenOptionalInterface, Ownable		
	updateDividendUniswapV2Router	External	✓	onlyOwner
	<Constructor>	Public	✓	ERC20
	<Receive Ether>	External	Payable	-
	swapETHForTokens	Private	✓	
	setIgnoreToken	External	✓	onlyOwner
	isIgnoredToken	Public		-
	getRawBNBDividends	External		-
	setWhiteListAMM	External	✓	onlyOwner
	setRewardToken	External	✓	onlyOwner
	unsetRewardToken	External	✓	onlyOwner
	distributeDividends	Public	Payable	-
	withdrawDividend	Public	✓	-
	_withdrawDividendOfUser	Internal	✓	
	dividendOf	Public		-
	withdrawableDividendOf	Public		-

	withdrawnDividendOf	Public		-
	accumulativeDividendOf	Public		-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_setBalance	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	✓	-

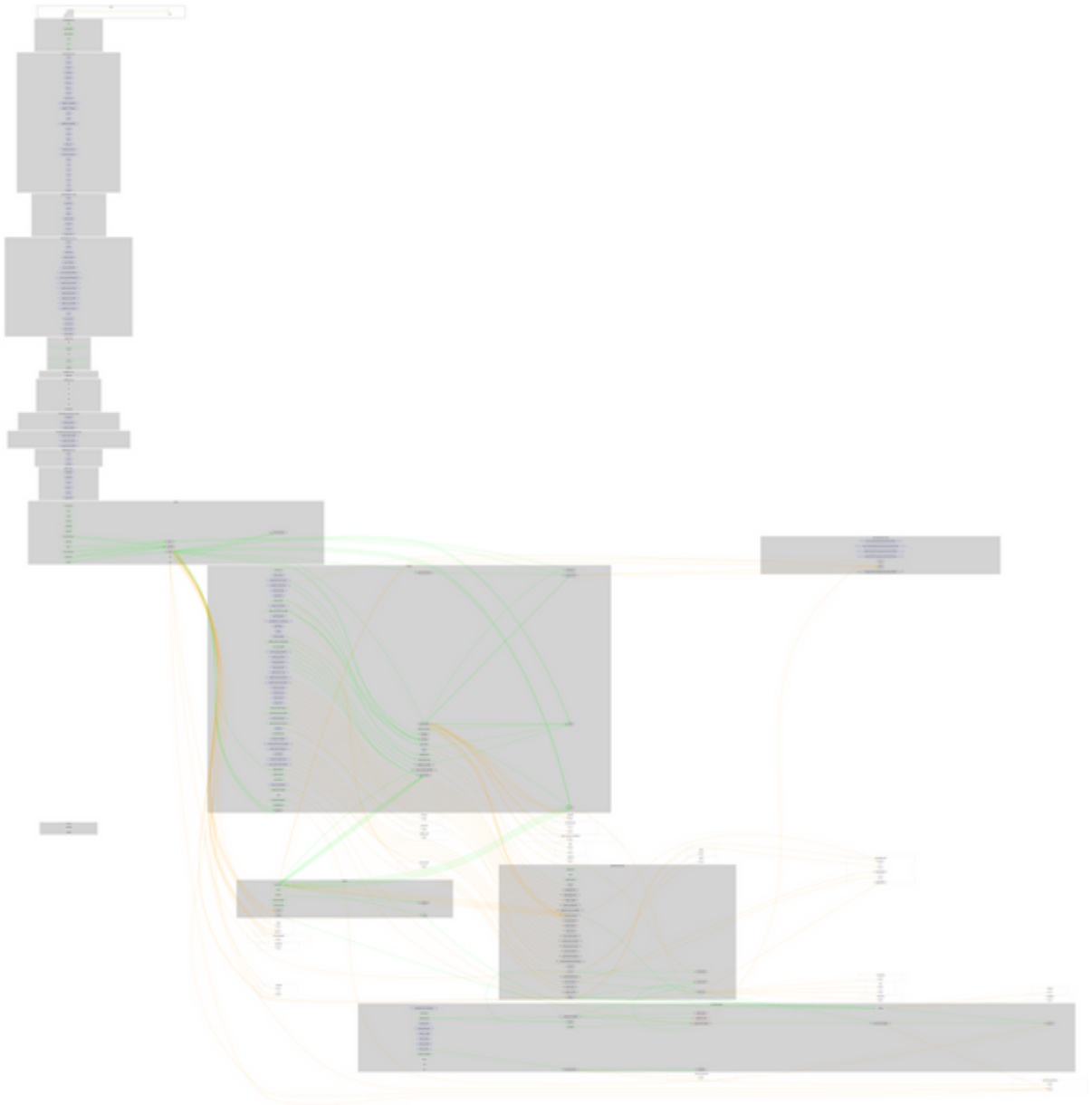
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	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	✓	-
SPONGS	Implementation	ERC20, Ownable		
	<Constructor>	Public	✓	ERC20
	<Receive Ether>	External	Payable	-
	setWhiteListAMM	External	✓	onlyOwner
	updateSwapTokensAtAmount	External	✓	onlyOwner
	disableTransferDelay	External	✓	onlyOwner
	updateDividendTracker	Public	✓	onlyOwner
	updateMaxTxn	External	✓	onlyOwner
	updateDividendTokensMinimum	External	✓	onlyOwner
	updateUniswapV2Router	External	✓	onlyOwner
	updateDividendUniswapV2Router	External	✓	onlyOwner
	updateTradingStatus	External	✓	onlyOwner
	excludeFromFees	Public	✓	onlyOwner
	excludeMultipleAccountsFromFees	External	✓	onlyOwner
	addToWhitelist	External	✓	onlyOwner
	setIsBot	External	✓	onlyOwner
	excludeFromDividends	External	✓	onlyOwner
	includeInDividends	External	✓	onlyOwner
	setAutomatedMarketMakerPair	External	✓	onlyOwner
	updateLiquidityWallet	External	✓	onlyOwner
	updateOperationsWallet	External	✓	onlyOwner
	updateBuyBackWallet	External	✓	onlyOwner
	updateFees	External	✓	onlyOwner
	updateGasForProcessing	External	✓	onlyOwner
	updateClaimWait	External	✓	onlyOwner
	setIgnoreToken	External	✓	onlyOwner

	isAMMWhitelisted	Public		-
	isContract	Internal		
	getUserCurrentRewardToken	Public		-
	getUserHasCustomRewardToken	Public		-
	getRewardTokenSelectionCount	Public		-
	getLastProcessedIndex	External		-
	getNumberOfDividendTokenHolders	External		-
	getHolderSellFactor	Public		-
	getDividendTokensMinimum	External		-
	getClaimWait	External		-
	getTotalDividendsDistributed	External		-
	isExcludedFromFees	Public		-
	withdrawableDividendOf	Public		-
	dividendTokenBalanceOf	Public		-
	getAccountDividendsInfo	External		-
	getAccountDividendsInfoAtIndex	External		-
	getRawBNBDividends	Public		-
	getBNBAvailableForHolderBuyBack	Public		-
	isIgnoredToken	Public		-
	setRewardToken	Public	✓	-
	setRewardTokenWithCustomAMM	Public	✓	-
	unsetRewardToken	Public	✓	-
	activateContract	Public	✓	onlyOwner
	buyBackTokensWithNoFees	External	Payable	-
	claim	External	✓	-
	processDividendTracker	External	✓	-
	_setAutomatedMarketMakerPair	Private	✓	
	_transfer	Internal	✓	
	swapAndLiquify	Private	✓	
	swapTokensForEth	Private	✓	
	addLiquidity	Private	✓	
	swapAndSendDividends	Private	✓	
	recoverContractBNB	Public	✓	onlyOwner
	sendToOperationsWallet	Private	✓	
	setMaxSellPercent	Public	✓	onlyOwner

IterableMapping	Library			
	get	Public		-
	getIndexOfKey	Public		-
	getKeyAtIndex	Public		-
	size	Public		-
	set	Public	✓	-
	remove	Public	✓	-
SPONGSDividendTracker	Implementation	DividendPayingToken		
	<Constructor>	Public	✓	DividendPayingToken
	_transfer	Internal		
	withdrawDividend	Public		-
	excludeFromDividends	External	✓	onlyOwner
	includeInDividends	External	✓	onlyOwner
	updateDividendMinimum	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	spongebobsquare.com
Registry Domain ID	2666001178_DOMAIN_COM-VRSN
Creation Date	2022-01-04T19:12:52
Updated Date	2022-01-04T19:12:53
Registry Expiry Date	
Registrar WHOIS Server	whois.wix.com
Registrar URL	http://www.wix.com
Registrar	Wix.Com Ltd.
Registrar IANA ID	3817

The domain has been created 9 days before the creation of the audit.

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

Summary

SpongeBob Square is a meme token that is aiming to build a web marketplace. The token has a friendly and growing community. There are some functions that can be abused by the owner, like manipulating fees, blacklisting wallets, stopping the transactions and preventing users from selling tokens. 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