

Smart Contract Security Audit V1

One Stop Block Shop Token Smart Contract

19/9/2022



<https://saferico.com/>

business@saferico.com

https://t.me/SFI_ANN

—

Table of Contents

Table of Contents

Background

Project Information

Token Information

Executive Summary

File and Function Level Report

File in Scope:

Issues Checking Status

Severity Definitions

Audit Findings

Automatic testing

Testing proves

Inheritance graph

Call graph

Unified Modeling Language (UML)

Functions signature

Automatic general report

Conclusion

Disclaimer

Background

The purpose of the audit was to achieve the following:

- Ensure that the smart contract functions as intended.
- Identify potential security issues with the smart contract.

The information in this report should be used to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

Project Information

- **Platform:** Binance Smart Chain
- **Contract Address:** 0x6228283830c4a9b304f2e4c8562ae45bed82a280
- **Code Source:**

<https://bscscan.com/address/0x6228283830c4a9b304f2e4c8562ae45bed82a280#code>

Token Information

- Name: One Stop Block Shop
- Total Supply: 1,000,000
- Holders:
- Total transactions:

Contracts address deployed to test net (BSC)

One Stop Block Shop Token smart contract on BSC test net by the auditor to test every function (BSC Test Net)

<https://testnet.bscscan.com/address/0x12ffd21b1fb11dcca2cac3d34fe42ecbc0ce2b43>

Executive Summary

According to our assessment, the customer`s solidity smart contract is **Well Secured**.

Well Secured	✓
Secured	
Poor Secured	
Insecure	

Automated checks are with remix IDE. All issues were performed by the team, which included the analysis of code functionality, manual audit found during automated analysis were manually reviewed and applicable vulnerabilities are presented in the audit overview section. The general overview is presented in the Project Information section and all issues found are located in the audit overview section.

Team found 0 critical, 0 high, 0 medium, 3 low, 0 very low-level issues and 2 notes in all solidity files of the contract

The files:

One Stop Block Shop.sol

File and Function Level Report

File in Scope:

Contract Name	SHA 256 hash	Contract Address
One Stop Block Shop.sol	724e1dc72ed20b3673ca1cd977b5753473adfd8e30dd47e28804a6b9e6f74e78	0x6228283830c4a9b304f2e4c8562ae45bed82a280

- Contract: OSBSToken
- Inherit: Context, IERC20, Ownable
- Observation: All passed including security check
- Test Report: passed
- Score: passed
- Conclusion: passed

Function	Test Result	Type / Return Type	Score
name	✓	Read / public	Passed
symbol	✓	Read / public	Passed
decimals	✓	Read / public	Passed
totalSupply	✓	Read / public	Passed
allowance	✓	Read / public	Passed
balanceOf	✓	Read / public	Passed
Owner	✓	Read / public	Passed
isExcludedFromFee	✓	Read / public	Passed
isMarketPair	✓	Read / public	Passed
minimumTokensBefore Swap	✓	Read / public	Passed
ownerWallet	✓	Read / public	Passed
swapAndLiquifyEnabled	✓	Read / public	Passed

swapAndLiquifyByLimit Only	✓	Read / public	Passed
uniswapPair	✓	Read / public	Passed
uniswapV2Router	✓	Read / public	Passed
USDT	✓	Read / public	Passed
zeroAddress	✓	Read / public	Passed
usdtPair	✓	Read / public	Passed
_totalTaxIfBuying	✓	Read / public	Passed
_totalTaxIfSelling	✓	Read / public	Passed
deadAddress	✓	Read / public	Passed
getCirculatingSupply	✓	Read / public	Passed
approve	✓	Write / public	Passed
TransferFrom	✓	Write / public	Passed
increaseAllowance	✓	Write / public	Passed
transfer	✓	Write / public	Passed
decreaseAllowance	✓	Write / public	Passed
changeRouterVersion	✓	Write / public	Passed
rescueFunds	✓	Write / public	Passed
rescueStuckedToken	✓	Write / public	Passed
waiveOwnership	✓	Write / public	Passed
setIsExcludedFromFee	✓	Write / public	Passed
setMarketPair	✓	Write / public	Passed
setMarketPairStatus	✓	Write / public	Passed
setNumTokensBeforeSwap	✓	Write / public	Passed
setSwapAndLiquifyByLimitOnly	✓	Write / public	Passed
setSwapAndLiquifyEnabled	✓	Write / public	Passed
transferOwnership	✓	Write / public	Passed

Issues Checking Status

No.	Issue Description	Checking Status
1	Compiler warnings.	Passed
2	Race conditions and Reentrancy. Cross-function race conditions.	Passed
3	Possible delays in data delivery.	Passed
4	Oracle calls.	Passed
5	Design Logic.	Passed
6	Timestamp dependence.	Passed with notes
7	Integer Overflow and Underflow.	Passed
8	DoS with Revert.	Passed
9	DoS with block gas limit.	Passed with notes
10	Methods execution permissions.	Passed
11	Economy model. If application logic is based on an incorrect economic model, the application would not function correctly and participants would incur financial losses. This type of issue is most often found in bonus rewards systems, Staking and Farming contracts, Vault and Vesting contracts, etc.	Passed
12	The impact of the exchange rate on the logic.	Passed
13	Private user data leaks.	Passed
14	Malicious Event log.	Passed
15	Scoping and Declarations.	Passed
16	Uninitialized storage pointers.	Passed
17	Arithmetic accuracy.	Passed

Severity Definitions

Risk Level	Description
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to tokens loss etc.
High	High-level vulnerabilities are difficult to exploit; however, they also have significant impact on smart contract execution, e.g. public access to crucial functions
Medium	Medium-level vulnerabilities are important to fix; however, they can't lead to tokens lose
Low	Low-level vulnerabilities are mostly related to outdated, unused etc. code snippets, that can't have significant impact on execution
Note	Lowest-level vulnerabilities, code style violations and info statements can't affect smart contract execution and can be ignored.

Audit Findings

Critical:

No Critical severity vulnerabilities were found.

High:

No High severity vulnerabilities were found.

Medium:

No Medium severity vulnerabilities were found.

Low:

#Pragma version not fixed

Description

It is a good practice to lock the solidity version for a live deployment (use 0.8.16 instead of ^0.8.4). contracts should be deployed with the same compiler version and flags that they have been tested the most with. Locking the pragma helps ensure that contracts do not accidentally get deployed using, for example, the latest compiler which may have higher risks of undiscovered bugs. Contracts may also be deployed by others and the pragma indicates the compiler version intended by the original authors.

Remediation

Remove the ^ sign to lock the pragma version.

Status: **Acknowledged**.

#Use of block.timestamp for comparisons

Description

The value of block.timestamp can be manipulated by the miner.
And conditions with strict equality is difficult to achieve -
block.timestamp

Remediation

Avoid use of block.timestamp

Status: **Acknowledged**

#Owner privileges (In the period when the owner isn't renounced)

Description

The owner can enable / disable the trade.

The owner can include / exclude any address from Fees.

```
function setSwapAndLiquifyEnabled(bool _enabled) public onlyOwner {
    swapAndLiquifyEnabled = _enabled;
    emit SwapAndLiquifyEnabledUpdated(_enabled);
}

function setSwapAndLiquifyByLimitOnly(bool newValue) public onlyOwner {
    swapAndLiquifyByLimitOnly = newValue;
}

function getCirculatingSupply() public view returns (uint256) {
    return
    _totalSupply.sub(balanceOf(deadAddress)).sub(balanceOf(zeroAddress));
}

function setIsExcludedFromFee(address account, bool newValue) public onlyOwner {
    isExcludedFromFee[account] = newValue;
}
```

Remediation

Make these functions internal in next version or the team should announce the investors before change anything to give them time if they want to do anything.

P.S: This issue is common to the majority of rewards smart contracts.

Status: **Acknowledged**.

Very Low:

No Very Low severity vulnerabilities were found.

Notes:

#Naming Conventions

Description

The contract follows a consistent naming convention where we are private variables with leading "_" and public variables without it. But we have missed to comply to the condition for certain variable names "___totalTaxIfBuying" which is public.

Remediation

Remove "___" from external variable names and add it to private variable names.

Status: **Acknowledged**.

Constant calculations in the contract

Description

recalculated initialization will save 2847 units of gas in deployment

```
uint256 private _totalSupply = 1000000 * 10**_decimals;
```

Recommendation

Replace the initialization as

```
uint256 private _totalSupply = 1000000000000000000000000;
```

Status **Acknowledged**.

Automatic Testing

1- Check for security

724e1dc72ed20b3673ca1cd977b5753473adfd8e30dd47e28804a6b9e6f74e...

File: One Sto... | Language: solidity | Size: 21718 bytes | Date: 2022-09-19T11:21:29.894Z

Critical	High	Medium	Low	Note
0	0	0	0	0

✓

2- SOLIDITY STATIC ANALYSIS

SOLIDITY STATIC ANALYSIS

☒ Select all ☒ Autorun

Run

Security

☒ Select Security

- ☒ Transaction origin:
'tx.origin' used
- ☒ Check-effects-interaction:
Potential reentrancy bugs
- ☒ Inline assembly:
Inline assembly used
- ☒ Block timestamp:
Can be influenced by miners
- ☒ Low level calls:
Should only be used by experienced devs
- ☒ Block hash:
Can be influenced by miners
- ☒ Selfdestruct:
Contracts using destructed contract can be broken

Gas & Economy

☒ Select Gas & Economy

- ☒ Gas costs:
Too high gas requirement of functions
- ☒ This on local calls:
Invocation of local functions via 'this'
- ☒ Delete dynamic array:
Use require/assert to ensure complete deletion
- ☒ For loop over dynamic array:
Iterations depend on dynamic array's size
- ☒ Ether transfer in loop:
Transferring Ether in a for/while/do-while loop

SOLIDITY STATIC ANALYSIS

ERC

☒ Select ERC

- ☒ ERC20:
'decimals' should be 'uint8'

Miscellaneous

☒ Select Miscellaneous

- ☒ Constant/View/Pure functions:
Potentially constant/view/pure functions
- ☒ Similar variable names:
Variable names are too similar
- ☒ No return:
Function with 'returns' not returning
- ☒ Guard conditions:
Ensure appropriate use of require/assert
- ☒ Result not used:
The result of an operation not used
- ☒ String length:
Bytes length != String length
- ☒ Delete from dynamic array:
'delete' leaves a gap in array
- ☒ Data truncated:
Division on int/uint values truncates the result

3- Inheritance graph

```
graph TD; OSBToken --> IERC20; OSBToken --> Ownable; IERC20 --> Context; Ownable --> Context; SafeMath; IUniswapV2Factory; IUniswapV2Pair; IUniswapV2Router02 --> IUniswapV2Router01;
```

The diagram illustrates an inheritance graph for Solidity contracts. It shows the following relationships:

- OSBToken** inherits from **IERC20** and **Ownable**.
- IERC20** inherits from **Context**.
- Ownable** inherits from **Context**.
- SafeMath**, **IUniswapV2Factory**, and **IUniswapV2Pair** are standalone contracts with no visible inheritance.
- IUniswapV2Router02** inherits from **IUniswapV2Router01**.

4- SOLIDITY UNIT TESTING

SOLIDITY UNIT TESTING

✓ >

Test your smart contract in Solidity.

Select directory to load and generate test files.

Test directory:

☒ Select all

☒ tests/One Stop Block Shop_test.sol

Progress: 1 finished (of 1)

PASS

testSuite (tests/One Stop Block Shop_test.sol)

✓ Before all

✓ Check success

✓ Check success2

✓ Check failure

✓ Check sender and value

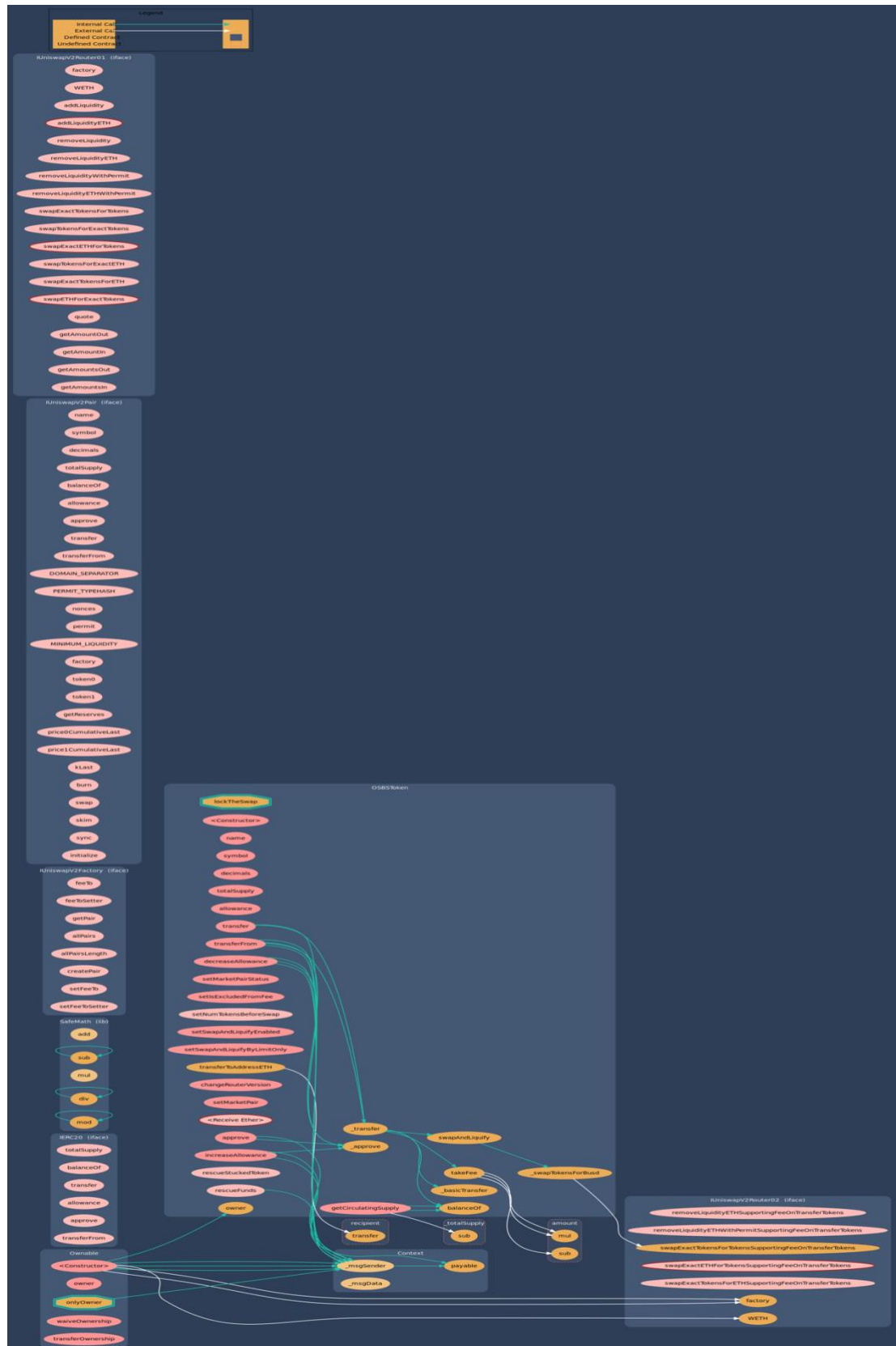
Result for tests/One Stop Block Shop_test.sol

Passed: 5

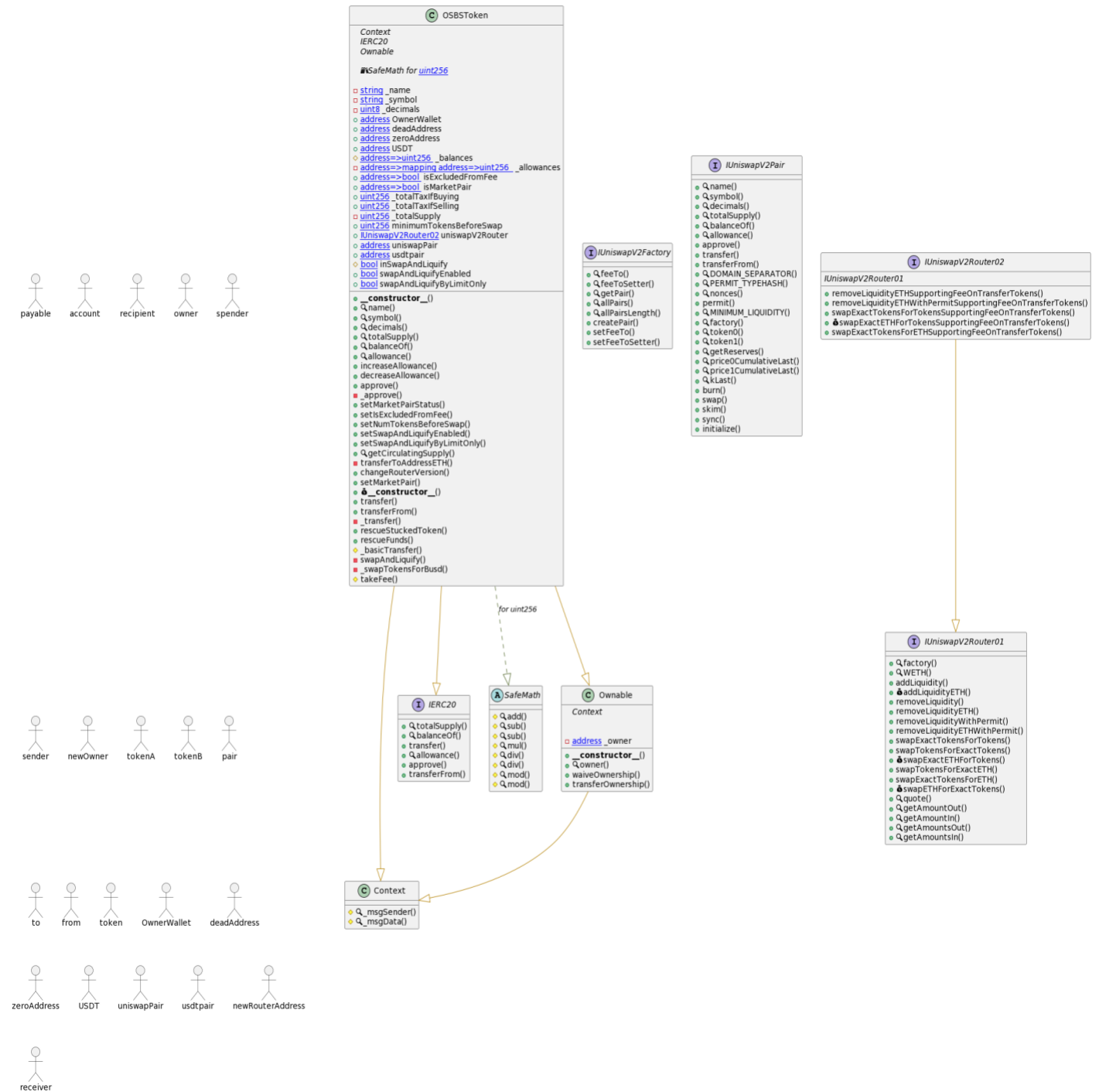
Failed: 0

Time Taken: 0.29s

5- Call graph



Unified Modeling Language (UML)



Functions signature

Sighash		Function Signature
=====		
39509351	=>	increaseAllowance(address,uint256)
119df25f	=>	_msgSender()
8b49d47e	=>	_msgData()
18160ddd	=>	totalSupply()
70a08231	=>	balanceOf(address)
a9059cbb	=>	transfer(address,uint256)
dd62ed3e	=>	allowance(address,address)
095ea7b3	=>	approve(address,uint256)
23b872dd	=>	transferFrom(address,address,uint256)
771602f7	=>	add(uint256,uint256)
b67d77c5	=>	sub(uint256,uint256)
e31bdc0a	=>	sub(uint256,uint256,string)
c8a4ac9c	=>	mul(uint256,uint256)
a391c15b	=>	div(uint256,uint256)
b745d336	=>	div(uint256,uint256,string)
f43f523a	=>	mod(uint256,uint256)
71af23e8	=>	mod(uint256,uint256,string)
8da5cb5b	=>	owner()
914eb66a	=>	waiveOwnership()
f2fde38b	=>	transferOwnership(address)
017e7e58	=>	feeTo()
094b7415	=>	feeToSetter()
e6a43905	=>	getPair(address,address)
1e3dd18b	=>	allPairs(uint256)
574f2ba3	=>	allPairsLength()
c9c65396	=>	createPair(address,address)
f46901ed	=>	setFeeTo(address)
a2e74af6	=>	setFeeToSetter(address)
06fdde03	=>	name()
95d89b41	=>	symbol()
313ce567	=>	decimals()
3644e515	=>	DOMAIN_SEPARATOR()
30adf81f	=>	PERMIT_TYPEHASH()
7ecebe00	=>	nonces(address)
d505accf	=>	permit(address,address,uint256,uint256,uint8,bytes32,bytes32)
ba9a7a56	=>	MINIMUM_LIQUIDITY()
c45a0155	=>	factory()
0dfe1681	=>	token0()
d21220a7	=>	token1()
0902f1ac	=>	getReserves()
5909c0d5	=>	price0CumulativeLast()
5a3d5493	=>	price1CumulativeLast()
7464fc3d	=>	kLast()
89afcb44	=>	burn(address)
022c0d9f	=>	swap(uint256,uint256,address,bytes)
bc25cf77	=>	skim(address)
fff6cae9	=>	sync()
485cc955	=>	initialize(address,address)
ad5c4648	=>	WETH()
e8e33700	=>	
		addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256)
f305d719	=>	addLiquidityETH(address,uint256,uint256,uint256,address,uint256)
baa2abde	=>	
		removeLiquidity(address,address,uint256,uint256,uint256,address,uint256)


```

02751cec => removeLiquidityETH(address,uint256,uint256,uint256,address,uint256)
2195995c =>
removeLiquidityWithPermit(address,address,uint256,uint256,uint256,address,uint256,boo
ool,uint8,bytes32,bytes32)
ded9382a =>
removeLiquidityETHWithPermit(address,uint256,uint256,uint256,address,uint256,bool,u
int8,bytes32,bytes32)
38ed1739 => swapExactTokensForTokens(uint256,uint256,address[],address,uint256)
8803dbee => swapTokensForExactTokens(uint256,uint256,address[],address,uint256)
7ff36ab5 => swapExactETHForTokens(uint256,address[],address,uint256)
4a25d94a => swapTokensForExactETH(uint256,uint256,address[],address,uint256)
18cbafe5 => swapExactTokensForETH(uint256,uint256,address[],address,uint256)
fb3bdb41 => swapETHForExactTokens(uint256,address[],address,uint256)
ad615dec => quote(uint256,uint256,uint256)
054d50d4 => getAmountOut(uint256,uint256,uint256)
85f8c259 => getAmountIn(uint256,uint256,uint256)
d06ca61f => getAmountsOut(uint256,address[])
1f00ca74 => getAmountsIn(uint256,address[])
af2979eb =>
removeLiquidityETHSupportingFeeOnTransferTokens(address,uint256,uint256,uint256,add
ress,uint256)
5b0d5984 =>
removeLiquidityETHWithPermitSupportingFeeOnTransferTokens(address,uint256,uint256,u
int256,address,uint256,bool,uint8,bytes32,bytes32)
5c11d795 =>
swapExactTokensForTokensSupportingFeeOnTransferTokens(uint256,uint256,address[],add
ress,uint256)
b6f9de95 =>
swapExactETHForTokensSupportingFeeOnTransferTokens(uint256,address[],address,uint25
6)
791ac947 =>
swapExactTokensForETHSupportingFeeOnTransferTokens(uint256,uint256,address[],addres
s,uint256)
a457c2d7 => decreaseAllowance(address,uint256)
104e81ff => _approve(address,address,uint256)
844d591c => setMarketPairStatus(address,bool)
ef422a18 => setIsExcludedFromFee(address,bool)
3b97084a => setNumTokensBeforeSwap(uint256)
c49b9a80 => setSwapAndLiquifyEnabled(bool)
a5d69d1f => setSwapAndLiquifyByLimitOnly(bool)
2b112e49 => getCirculatingSupply()
4a8da1c5 => transferToAddressETH(address,uint256)
5881f3ef => changeRouterVersion(address)
c16dd4a4 => setMarketPair(address,bool)
30e0789e => _transfer(address,address,uint256)
f4554446 => rescueStuckedToken(address,uint256)
e6b2603b => rescueFunds()
f0774e71 => _basicTransfer(address,address,uint256)
173865ad => swapAndLiquify(uint256)
ae503da1 => _swapTokensForBUSD(uint256,address)
20cb7bce => takeFee(address,address,uint256)

```

Automatic general report

Files Description Table

File Name	SHA-1 Hash
/Users/macbook/Desktop/smart contracts/One Stop Block Shop.sol	c6dcd69a84e19c664e4ec60edae6a1b90edb1091

Contracts Description Table






























Contract	Type	Bases		
:-----: :-----: :-----: :-----: :-----:				
L	**Function Name**	**Visibility**	**Mutability**	
Modifiers				
Context	Implementation			
L	_msgSender	Internal		
L	_msgData	Internal		
IERC20	Interface			
L	totalSupply	External		NO
L	balanceOf	External		NO
L	transfer	External		NO
L	allowance	External		NO
L	approve	External		NO
L	transferFrom	External		NO
SafeMath	Library			
L	add	Internal		
L	sub	Internal		
L	sub	Internal		
L	mul	Internal		
L	div	Internal		
L	div	Internal		
L	mod	Internal		
L	mod	Internal		
Ownable	Implementation	Context		
L	<Constructor>	Public		NO
L	owner	Public		NO
L	waiveOwnership	Public		onlyOwner
L	transferOwnership	Public		onlyOwner
IUniswapV2Factory	Interface			
L	feeTo	External		NO
L	feeToSetter	External		NO
L	getPair	External		NO
L	allPairs	External		NO
L	allPairsLength	External		NO
L	createPair	External		NO
L	setFeeTo	External		NO
L	setFeeToSetter	External		NO

```

| | | | | |
| **IUniswapV2Pair** | Interface | | |
| L | name | External | ! | NO |
| L | symbol | External | ! | NO |
| L | decimals | External | ! | NO |
| L | totalSupply | External | ! | NO |
| L | balanceOf | External | ! | NO |
| L | allowance | External | ! | NO |
| L | approve | External | ! | NO |
| L | transfer | External | ! | NO |
| L | transferFrom | External | ! | NO |
| L | DOMAIN_SEPARATOR | External | ! | NO |
| L | PERMIT_TYPEHASH | External | ! | NO |
| L | nonces | External | ! | NO |
| L | permit | External | ! | NO |
| L | MINIMUM_LIQUIDITY | External | ! | NO |
| L | factory | External | ! | NO |
| L | token0 | External | ! | NO |
| L | token1 | External | ! | NO |
| L | getReserves | External | ! | NO |
| L | price0CumulativeLast | External | ! | NO |
| L | price1CumulativeLast | External | ! | NO |
| L | kLast | External | ! | NO |
| L | burn | External | ! | NO |
| L | swap | External | ! | NO |
| L | skim | External | ! | NO |
| L | sync | External | ! | NO |
| L | initialize | External | ! | NO |
| | | |
| **IUniswapV2Router01** | Interface | | |
| L | factory | External | ! | NO |
| L | WETH | External | ! | NO |
| L | addLiquidity | External | ! | NO |
| L | addLiquidityETH | External | ! | NO |
| L | removeLiquidity | External | ! | NO |
| L | removeLiquidityETH | External | ! | NO |
| L | removeLiquidityWithPermit | External | ! | NO |
| L | removeLiquidityETHWithPermit | External | ! | NO |
| L | swapExactTokensForTokens | External | ! | NO |
| L | swapTokensForExactTokens | External | ! | NO |
| L | swapExactETHForTokens | External | ! | NO |
| L | swapTokensForExactETH | External | ! | NO |
| L | swapExactTokensForETH | External | ! | NO |
| L | swapETHForExactTokens | External | ! | NO |
| L | quote | External | ! | NO |
| L | getAmountOut | External | ! | NO |
| L | getAmountIn | External | ! | NO |
| L | getAmountsOut | External | ! | NO |
| L | getAmountsIn | External | ! | NO |
| | | |
| **IUniswapV2Router02** | Interface | IUniswapV2Router01 | | |
| L | removeLiquidityETHSupportingFeeOnTransferTokens | External | ! | NO |
| L | removeLiquidityETHWithPermitSupportingFeeOnTransferTokens | External | ! | NO |
| L | swapExactTokensForTokensSupportingFeeOnTransferTokens | External | ! | NO |
| L | swapExactETHForTokensSupportingFeeOnTransferTokens | External | ! | NO |
| L | swapExactTokensForETHSupportingFeeOnTransferTokens | External | ! | NO |



```

```

| | | | | | |
| **OSBSToken** | Implementation | Context, IERC20, Ownable | | |
| L | <Constructor> | Public | ! |  | NO! |
| L | name | Public | ! | NO! |
| L | symbol | Public | ! | NO! |
| L | decimals | Public | ! | NO! |
| L | totalSupply | Public | ! | NO! |
| L | balanceOf | Public | ! | NO! |
| L | allowance | Public | ! | NO! |
| L | increaseAllowance | Public | ! |  | NO! |
| L | decreaseAllowance | Public | ! |  | NO! |
| L | approve | Public | ! |  | NO! |
| L | _approve | Private |  |  |
| L | setMarketPairStatus | Public | ! |  | onlyOwner |
| L | setIsExcludedFromFee | Public | ! |  | onlyOwner |
| L | setNumTokensBeforeSwap | External | ! |  | onlyOwner |
| L | setSwapAndLiquifyEnabled | Public | ! |  | onlyOwner |
| L | setSwapAndLiquifyByLimitOnly | Public | ! |  | onlyOwner |
| L | getCirculatingSupply | Public | ! | NO! |
| L | transferToAddressETH | Private |  |  |
| L | changeRouterVersion | Public | ! |  | onlyOwner |
| L | setMarketPair | Public | ! |  | onlyOwner |
| L | <Receive Ether> | External | ! |  | NO! |
| L | transfer | Public | ! |  | NO! |
| L | transferFrom | Public | ! |  | NO! |
| L | _transfer | Private |  |  |
| L | rescueStuckedToken | External | ! |  | onlyOwner |
| L | rescueFunds | External | ! |  | onlyOwner |
| L | _basicTransfer | Internal |  |
| L | swapAndLiquify | Private |  |  | lockTheSwap |
| L | _swapTokensForBusd | Private |  |  |
| L | takeFee | Internal |  |  |

```

Legend

Symbol	Meaning
	Function can modify state
	Function is payable

Conclusion

The contracts are written systematically. Team found no critical issues. So, it is good to go for production.

Since possible test cases can be unlimited and developer level documentation (code flow diagram with function level description) not provided, for such an extensive smart contract protocol, we provide no such guarantee of future outcomes. We have used all the latest static tools and manual observations to cover maximum possible test cases to scan Everything.

Security state of the reviewed contract is “Well Secured”.

- ✓ No mint function.
- ✓ No volatile code.
- ✓ No high severity issues were found.

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as of the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against the team on the basis of what it says or doesn't say, or how team produced it, and it is important for you to conduct your own independent investigations before making any decisions. team go into more detail on this in the below disclaimer below – please make sure to read it in full.

By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and Saferico and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (Saferico s) owe no duty of care towards you or any other person, nor does Saferico make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and Saferico hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, Saferico hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against Saferico, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report. The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.