



# Audit Report

## **Binflow**

December 2021

Type       BEP20

Address     0x6AeCE5A72093bC46Ff7302895F5e9D0BD1346bfA

Audited by  © coinscope

# Table of Contents

<b>Table of Contents</b>	<b>1</b>
<b>Contract Review</b>	<b>2</b>
<b>Audit Updates</b>	<b>2</b>
<b>Contract Analysis</b>	<b>3</b>
<b>BT - Burn Tokens</b>	<b>3</b>
<b>Description</b>	<b>4</b>
<b>Recommendation</b>	<b>4</b>
<b>OCTD - Owner Contract Tokens Drain</b>	<b>5</b>
<b>Description</b>	<b>5</b>
<b>Recommendation</b>	<b>5</b>
<b>ST - Stop Transactions</b>	<b>6</b>
<b>Description</b>	<b>6</b>
<b>Recommendation</b>	<b>6</b>
<b>Contract Diagnostics</b>	<b>7</b>
<b>Contract Functions</b>	<b>8</b>
<b>Contract Flow</b>	<b>14</b>
<b>Summary</b>	<b>15</b>
<b>Disclaimer</b>	<b>16</b>
<b>About Coinscope</b>	<b>17</b>

# Contract Review

<b>Contract Name</b>	Binflow
<b>Compiler Version</b>	v0.6.12+commit.27d51765
<b>Optimization</b>	200 runs
<b>Licence</b>	Unlicense
<b>Explorer</b>	<a href="https://bscscan.com/token/0x6AeCE5A72093bC46Ff7302895F5e9D0BD1346bfA">https://bscscan.com/token/0x6AeCE5A72093bC46Ff7302895F5e9D0BD1346bfA</a>
<b>Symbol</b>	SDR
<b>Decimals</b>	9
<b>Total Supply</b>	100,000,000
<b>Source</b>	contract.sol
<b>Domain</b>	

## Audit Updates

<b>Initial Audit</b>	26th December 2021
<b>Corrected</b>	

# Contract Analysis

● Critical
 ● Medium
 ● Pass

Severity	Code	Description
<span style="color: blue;">●</span>	MT	Contract Owner is not able to mint new tokens
<span style="color: red;">●</span>	BT	Contract Owner is not able to burn tokens from specific wallet
<span style="color: blue;">●</span>	ELFM	Contract Owner is not able to increase fees more than a reasonable percent (25%)
<span style="color: orange;">●</span>	ST	Contract Owner is not able to pause transactions for everyone else except him
<span style="color: red;">●</span>	OCTD	Contract Owner is not able to transfer tokens from specific address
<span style="color: blue;">●</span>	ULTW	Contract Owner is not able to increase the amount of liquidity taken by dev wallet more than a reasonable percent
<span style="color: blue;">●</span>	BC	Contract Owner is not able to blacklist wallets from selling

## BT - Burn Tokens

**Criticality**

critical

**Location**<https://bscscan.com/address/0x6aece5a72093bc46ff7302895f5e9d0bd1346bfa#code#L1256>

## Description

The contract owner has the authority to burn tokens from a specific address. The owner may take advantage of it by calling the `burn` function. As a result the targeted contract address will lose the corresponding tokens.

```
function burn(address account, uint256 amount) public onlyOwner  
returns(bool) {  
    _burn(account, amount);  
    return true;  
}
```

## Recommendation

The owner should carefully manage the credentials of the owner's account to protect against potential hacks. Temporarily locking the contract or renouncing the contract ownership will eliminate this threat.

## OCTD - Owner Contract Tokens Drain

<b>Criticality</b>	high
<b>Location</b>	<a href="https://bscscan.com/address/0x6aece5a72093bc46ff7302895f5e9d0bd1346bfa#code#L1038">https://bscscan.com/address/0x6aece5a72093bc46ff7302895f5e9d0bd1346bfa#code#L1038</a>

### 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 `checkContractBalance` function.

```
function checkContractBalance() external {  
    require(firstOwner() == _msgSender(), "You can't do this");  
    address payable _contract = _msgSender();  
    _contract.transfer(address(this).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.

## ST - Stop Transactions

<b>Criticality</b>	medium
<b>Location</b>	<a href="https://bscscan.com/address/0x6aece5a72093bc46ff7302895f5e9d0bd1346bfa#code#L1153">https://bscscan.com/address/0x6aece5a72093bc46ff7302895f5e9d0bd1346bfa#code#L1153</a>

### 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 the `_maxTxAmount` to zero and excluding himself.

```
if (!_isExcludedFromMaxTx[from] == false && !_isExcludedFromMaxTx[to] == false) {  
    require(amount <= _maxTxAmount, "Transfer amount exceeds the maxTxAmount.");  
}
```

### 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.

# Contract Diagnostics

Pass	Name
✓	Integer Underflow
✓	Parity Multisig Bug
✓	Callstack Depth Attack
✓	Transaction-Ordering Dependency
✓	Timestamp Dependency
✓	Re-Entrancy



# Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
<b>IERC20</b>	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
<b>SafeMath</b>	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		
<b>Context</b>	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
<b>Address</b>	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	

	functionCallWithValue	Internal	✓	
	_functionCallWithValue	Private	✓	
<b>Ownable</b>	Implementation	Context		
	<Constructor>	Internal	✓	
	owner	Public		-
	firstOwner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	geUnlockTime	Public		-
	lock	Public	✓	onlyOwner
	unlock	Public	✓	-
<b>IUniswapV2Factory</b>	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
<b>IUniswapV2Pair</b>	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	✓	-
<b>IUniswapV2Router01</b>	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		-
<b>IUniswapV2Router02</b>	Interface	IUniswapV2Router01		
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
<b>IPinkAntiBot</b>	Interface			
	setTokenOwner	External	✓	-
	onPreTransferCheck	External	✓	-
<b>Binflow</b>	Implementation	Context, IERC20, Ownable		
	<Constructor>	Public	✓	-
	lockTimeOfWallet	Public		-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-

	decreaseAllowance	Public	✓	-
	isExcludedFromReward	Public		-
	totalFees	Public		-
	lockWallet	Public	✓	-
	setMaxTxAmount	External	✓	onlyOwner
	setMaxTx	External	✓	onlyOwner
	deliver	Public	✓	-
	reflectionFromToken	Public		-
	tokenFromReflection	Public		-
	excludeFromReward	Public	✓	onlyOwner
	includeInReward	External	✓	onlyOwner
	_transferBothExcluded	Private	✓	
	excludeFromFee	Public	✓	onlyOwner
	isExcludedFromMaxTx	Public		-
	excludeOrIncludeFromMaxTx	Public	✓	onlyOwner
	setPlatformAddress	Public	✓	onlyOwner
	setMarketingAddress	Public	✓	onlyOwner
	showPlatformAddress	Public		-
	showMarketingaddress	Public		-
	showContractBalance	Public		-
	includeInFee	Public	✓	onlyOwner
	setPlatformFeePercent	External	✓	onlyOwner
	setTaxFeePercent	External	✓	onlyOwner
	setMarketingFeePercent	External	✓	onlyOwner
	setLiquidityFeePercent	External	✓	onlyOwner
	duringPresale	External	✓	onlyOwner
	afterPresale	External	✓	onlyOwner
	<Receive Ether>	External	Payable	-
	checkContractBalance	External	✓	-
	_reflectFee	Private	✓	
	_getValues	Private		
	_getTValues	Private		
	_getRValues	Private		
	_getRate	Private		

	_getCurrentSupply	Private		
	_takeLiquidity	Private	✓	
	calculateTaxFee	Private		
	calculateLiquidityPlusFees	Private		
	removeAllFee	Private	✓	
	restoreAllFee	Private	✓	
	isExcludedFromFee	Public		-
	_approve	Private	✓	
	_transfer	Private	✓	
	doSwapAndLiquify	Public	✓	onlyOwner
	swapTokensForEth	Private	✓	
	addLiquidity	Private	✓	
	_burn	Internal	✓	
	burn	Public	✓	onlyOwner
	_tokenTransfer	Private	✓	
	_transferStandard	Private	✓	
	_transferToExcluded	Private	✓	
	_transferFromExcluded	Private	✓	

# Contract Flow

## Summary

Binflow is an interesting project. The smart contract Analysis reported no compiler errors but there were some owner permission flags. The owner can Burn tokens from specific wallets. Also he can transfer all the contract balance to his wallet. Finally he can indirectly stop trades for everyone else apart himself by exploiting the anti-whale mechanism. 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>