



# AUDIT REPORT

# SecureWise

## SMART CONTRACT AUDIT



-  <https://github.com/securewise>
-  <https://t.me/securewise>
-  <https://securewise.info/>

# Table of Contents

**03**

Disclaimer

**04**

Overview

**05**

Quick Result

**06**

Auditing  
Approach and  
Methodologies

**07**

Automated  
Analysis

**13**

Inheritance  
Graph

**14**

Contract  
Summary

**15**

Manual Review

# Disclaimer

SecureWise provides the smart contract audit of solidity. Audit and report are for informational purposes only and not, nor should be considered, as an endorsement to engage with, invest in, participate, provide an incentive, or disapprove, criticise, discourage, or purport to provide an opinion on any particular project or team.

This audit report doesn't provide any warranty or guarantee regarding the nature of the technology analysed. These reports, in no way, provide investment advice, nor should be used as investment advice of any sort. Investors must always do their own research and manage their risk.

**DISCLAIMER:** 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 SecureWise and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) SecureWise owe no duty of care towards you or any other person, nor does SecureWise 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 SecureWise 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, SecureWise hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against SecureWise, 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.

# Overview

**Token Name:** SKUNK TOKEN

**Methodology:** Automated Analysis, Manual Code Review

**Language:** Solidity

**Contract Address:** 0x405Be46CE184F0339821Aafe81080A8f3acFc3b3

**ContractLink:** <https://bscscan.com/address/0x405Be46CE184F0339821Aafe81080A8f3acFc3b3>

**Network:** Binance Smart Chain (BSC)

**Decimals:** 16

**Supply:** 100.000.000.000.000.000

**Website:** <https://www.skunktoken.com/>

**Twitter:** [https://twitter.com/Skunk\\_Token](https://twitter.com/Skunk_Token)

**Telegram:** <https://t.me/skunktken2022>

**Report Date:** October 1, 2022

# Quick Result

SecureWise has applied the automated and manual analysis of Smart Contract and were reviewed for common contract vulnerabilities and centralized exploits

## Owner Privileges

-  The owner can set the max tx amount "0"
-  The owner can set fees up to 100%
-  The owner can exclude accounts from fees
-  The owner can change swap settings

**SKUNK TOKEN** has successfully **PASSED** the smart contract audit with **HIGH and LOW** severity issue

# Auditing Approach and Methodologies

SecureWise has performed starting with analyzing the code, issues, code quality, and libraries. Reviewed line-by-line by our team. Finding any potential issue like race conditions, transaction-ordering dependence, timestamp dependence, and denial of service attacks.

## Methodology

- Understanding the size, scope and functionality of your project's source code
- Manual review of code, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
- Testing and automated analysis of the Smart Contract to determine proper logic has been followed throughout the whole process
- Deploying the code on testnet using multiple live test
- Analyzing a program to determine the specific input that causes different parts of a program to execute its functions.
- Checking whether all the libraries used in the code are on the latest version.

## Goals

Smart Contract System is secure, resilient and working according to the specifications and without any vulnerabilities.

## Risk Classification

**High:** Exploits, vulnerabilities or errors that will certainly or probabilistically lead towards loss of funds, control, of the contract and its functions. Must be fixed as soon as possible.

**Medium:** Bugs or issues with that may be subject to exploit, though their impact is somewhat limited. Must be fixed as soon as possible.

**Low:** Effects are minimal in isolation and do not pose a significant danger to the project or its users. Issues under this classification are recommended to be fixed nonetheless.

# Automated Analysis

Symbol	Meaning
	Function can modify state
	Function is payable

Contract	Category	Type	State Modifiers	Paying	Modifiable
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	tryAdd	Internal			
L	trySub	Internal			
L	tryMul	Internal			
L	tryDiv	Internal			
L	tryMod	Internal			
L	add	Internal			
L	sub	Internal			
L	mul	Internal			
L	div	Internal			
L	mod	Internal			
L	sub	Internal			
L	div	Internal			
L	mod	Internal			
Context	Implementation				
L	_msgSender	Internal			
L	_msgData	Internal			

# Automated Analysis

Address	Library			
L	isContract	Internal 🔒		
L	sendValue	Internal 🔒	●	
L	functionCall	Internal 🔒	●	
L	functionCall	Internal 🔒	●	
L	functionCallWithValue	Internal 🔒	●	
L	functionCallWithValue	Internal 🔒	●	
L	functionStaticCall	Internal 🔒		
L	functionStaticCall	Internal 🔒		
L	functionDelegateCall	Internal 🔒	●	
L	functionDelegateCall	Internal 🔒	●	
L	verifyCallResult	Internal 🔒		
Ownable	Implementation	Context		
L		Public ⓘ	●	NO ⓘ
L	owner	Public ⓘ		NO ⓘ
L	renounceOwnership	Public ⓘ	●	onlyOwner
L	transferOwnership	Public ⓘ	●	onlyOwner
L	_transferOwnership	Internal 🔒	●	
UniSwapFactory	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 ⓘ

# Automated Analysis

IIUniSwapPair	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	mint	External	●	NO
L	burn	External	●	NO
L	swap	External	●	NO
L	skim	External	●	NO
L	sync	External	●	NO
L	initialize	External	●	NO

# Automated Analysis

IUniswapV2Router01	Interface			
L	factory	External		NO
L	WETH	External		NO
L	WBNB	External		NO
L	WAVAX	External		NO
L	WHT	External		NO
L	addLiquidity	External	●	NO
L	addLiquidityETH	External	■	NO
L	addLiquidityBNB	External	■	NO
L	addLiquidityAVAX	External	■	NO
L	addLiquidityHT	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

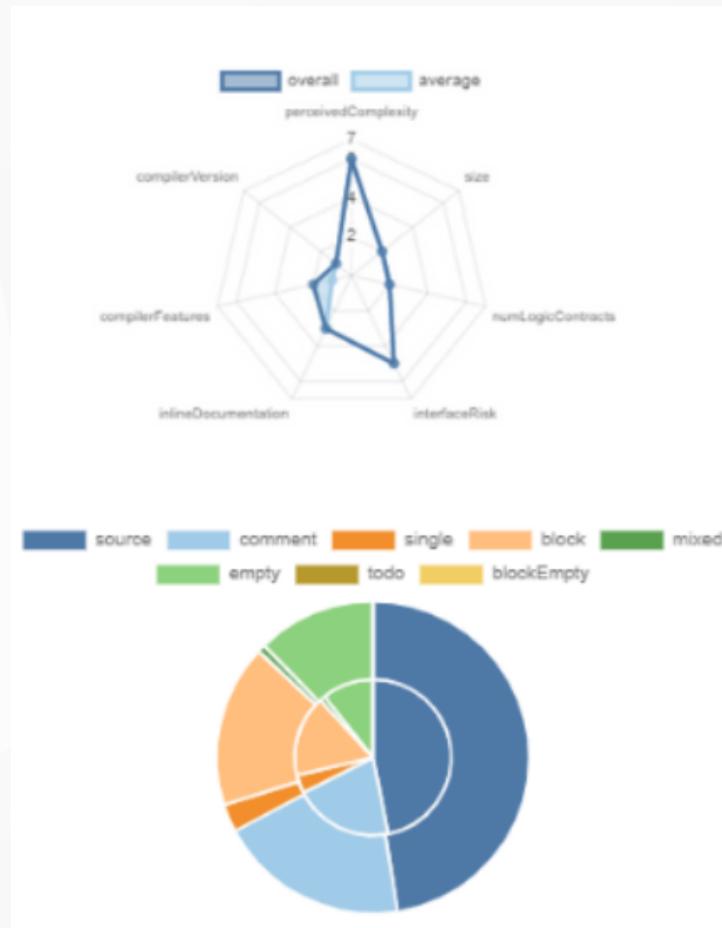
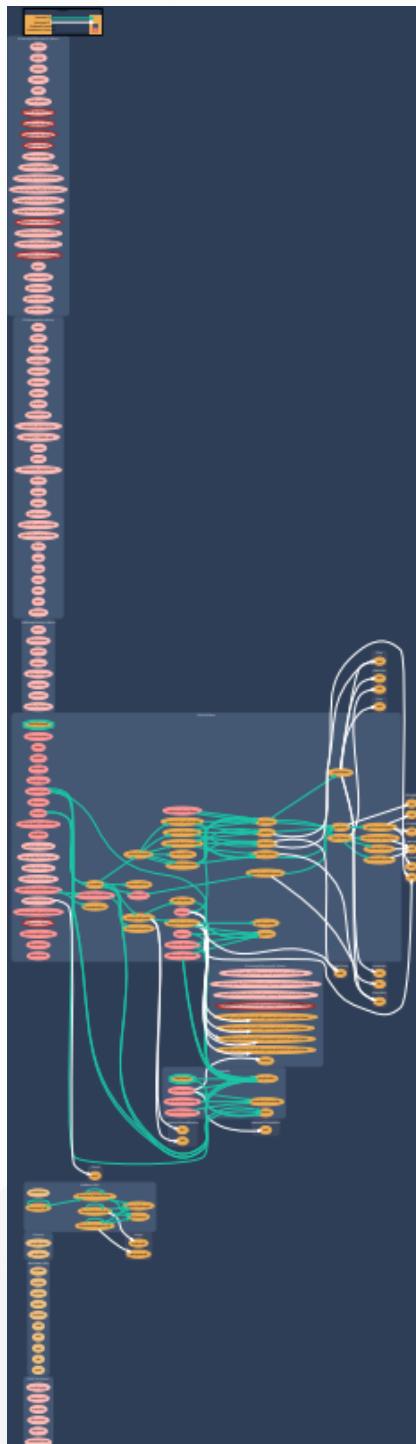
# Automated Analysis

IUniswapV2Router02	Interface	IUniswapV2Router01		
L	removeLiquidityETHSupportingFeeOnTransferTokens	External	●	NO
L	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	●	NO
L	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	●	NO
L	swapExactETHForTokensSupportingFeeOnTransferTokens	External	■	NO
L	swapExactTokensForETHSupportingFeeOnTransferTokens	External	●	NO
L	swapExactTokensForBNBSupportingFeeOnTransferTokens	External	●	NO
L	swapExactTokensForAVAXSupportingFeeOnTransferTokens	External	●	NO
L	swapExactTokensForHTSupportingFeeOnTransferTokens	External	●	NO
<hr/>				
DxFeeToken	Implementation	Context, IERC20, Ownable		
L		Public	●	NO
L	getWrapAddr	Public		NO
L	name	Public		NO
L	symbol	Public		NO
L	decimals	Public		NO
L	totalSupply	Public		NO
L	balanceOf	Public		NO
L	transfer	Public	●	NO
L	allowance	Public		NO
L	approve	Public	●	NO
L	transferFrom	Public	●	NO
L	increaseAllowance	Public	●	NO
L	decreaseAllowance	Public	●	NO
L	isExcludedFromReward	Public		NO
L	totalFees	Public		NO
L	deliver	Public	●	NO
L	reflectionFromToken	Public		NO

# Automated Analysis

L	includeInFee	Public 🚧	●	onlyOwner
L	setTaxFeePercent	External 🚧	●	onlyOwner
L	setLiquidityFeePercent	External 🚧	●	onlyOwner
L	setDevFeePercent	External 🚧	●	onlyOwner
L	setSellTaxFeePercent	External 🚧	●	onlyOwner
L	setMaxTxPercent	External 🚧	●	onlyOwner
L	setDevWalletAddress	Public 🚧	●	onlyOwner
L	replaceDevWalletAddress	Public 🚧	●	onlyOwner
L		External 🚧	⚠️	NO 🚧
L	_getValues	Private 🗂️		
L	_getTValues	Private 🗂️		
L	_getRValues	Private 🗂️		
L	_getRate	Private 🗂️		
L	_getCurrentSupply	Private 🗂️		
L	_takeLiquidity	Private 🗂️	●	
L	_takeDev	Private 🗂️	●	
L	calculateTaxFee	Private 🗂️		
L	calculateLiquidityFee	Private 🗂️		
L	calculateDevFee	Private 🗂️		
L	removeAllFee	Private 🗂️	●	
L	restoreAllFee	Private 🗂️	●	
L	isExcludedFromFee	Public 🚧		NO 🚧
L	_approve	Private 🗂️	●	
L	_transfer	Private 🗂️	●	
L	swapAndLiquify	Private 🗂️	●	lockTheSwap
L	swapTokensForEth	Private 🗂️	●	
L	addLiquidity	Private 🗂️	●	
L	_tokenTransfer	Private 🗂️	●	
L	_transferStandard	Private 🗂️	●	
L	_transferToExcluded	Private 🗂️	●	
L	_transferFromExcluded	Private 🗂️	●	
L	_transferBothExcluded	Private 🗂️	●	
L	_reflectFee	Private 🗂️	●	
L	disableFees	Public 🚧	●	onlyOwner
L	enableFees	Public 🚧	●	onlyOwner

# Inheritance Graph



# Contract Summary

Logic Contracts	Interfaces	Lines	nLines	nSLOC	Comment Lines	Complex. Score	Capabilities
5	5	1618	1198	728	406	607	
5	5	1618	1198	728	406	607	

## Components

Contracts	Libraries	Interfaces	Abstract
1	2	5	2

## Exposed Functions

This section lists functions that are explicitly declared public or payable. Please note that getter methods for public stateVars are not included.

Public	Payable
108	8

External	Internal	Private	Pure	View
80	104	23	28	47

## StateVariables

Total	Public
43	23

## Capabilities

Solidity Versions observed	Experimental Features	Can Receive Funds	Uses Assembly	Has Destroyable Contracts
^0.8.7	ABIEncoderV2	yes	yes (1 asm blocks)	_____
Transfers ETH	Low-Level Calls	DelegateCall	Uses Hash Functions	ECRecover
_____	_____	yes	_____	_____
TryCatch	Unchecked			
yes	yes			

# Manual Review

## The owner can set the max tx amount "0"

```
function setMaxTxPercent(uint256 maxTxPercent) external onlyOwner() {
    require(maxTxPercent >= minMxTxPercentage && maxTxPercent <=100,"maxTxPercent out of range");
    _maxTxAmount = _tTotal.mul(maxTxPercent).div(
        10**2
    );
}
```

### Recommendation

These functions should be provided arbitrary limits, e.g., put a **require** check that allows maximum limit etc. if **set 0** these cause pause the trading.

## The owner can set fees up to 100%

```
1150     function setTaxFeePercent(uint256 taxFee) external onlyOwner() {
1151         require(taxFee >= 0 && taxFee <=maxTaxFee,"taxFee out of range");
1152         _taxFee = taxFee;
1153     }
1154
1155     function setLiquidityFeePercent(uint256 liquidityFee) external onlyOwner() {
1156         require(liquidityFee >= 0 && liquidityFee <=maxLiqFee,"liquidityFee out of range");
1157         _liquidityFee = liquidityFee;
1158     }
1159
1160     function setDevFeePercent(uint256 devFee) external onlyOwner() {
1161         require(devFee >= 0 && devFee <=maxDevFee,"teamFee out of range");
1162         _devFee = devFee;
1163     }
1164
1165     function setSellTaxFeePercent(uint256 sellTaxFee) external onlyOwner() {
1166         require(sellTaxFee >= 0 && sellTaxFee <=maxSellTaxFee,"taxFee out of range");
1167         _sellTaxFee = sellTaxFee;
1168     }
```

### Recommendation

These functions should be provided arbitrary limits, e.g., put a **require** check that allows maximum limit etc.

# Manual Review

## The owner can exclude accounts from fees

```
1140     function excludeFromFee(address account) public onlyOwner {
1141         require(!_isExcludedFromFee[account], "Account is already excluded");
1142         _isExcludedFromFee[account] = true;
1143     }
1144
1145     function includeInFee(address account) public onlyOwner {
1146         require(_isExcludedFromFee[account], "Account is already included");
1147         _isExcludedFromFee[account] = false;
1148     }
```

### Recommendation

Authorizing privileged roles to exclude accounts from fees. These cause affect decentralization

## The owner can change swap settings

```
1197     function setSwapAndLiquifyEnabled(bool _enabled) public onlyOwner {
1198         swapAndLiquifyEnabled = _enabled;
1199         emit SwapAndLiquifyEnabledUpdated(_enabled);
1200     }
```

### Recommendation

Authorizing privileged roles to enable or disable the swap. These cause affect decentralization

# **AUDIT REPORT**

# **SecureWise**

## **SMART CONTRACT AUDIT**

-  <https://github.com/securewise>
-  <https://t.me/securewise>
-  <https://securewise.info/>

