



## 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

**11**

Inheritance  
Graph

**12**

Contract  
Summary

**13**

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:** TinyTesla (TINT)

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

**Language:** Solidity

**Contract Address:** 0x527aD1d5BCEE13a2952d1832b18011649A78266b

**ContractLink:** <https://bscscan.com/address/0x527aD1d5BCEE13a2952d1832b18011649A78266b>

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

**Decimals:** 9

**Supply:** 1,000,000,000,000,000

**Website:** <https://www.tinytesla.co/>

**Twitter:** -

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

**Report Date:** August 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 fees up to 100%



The owner can't set the max tx amount "0" but can set very low amount



Auto liquidity is going to an externally owned account



The owner can't set max wallet token amount to "0" but can set very low amount



The owner can set a blacklist any account



The owner can exclude accounts from rewards



The owner can exclude accounts from fees



The owner can change swap settings



The owner can withdraw any token from the contract

**TinyTesla (TINT)** has successfully **PASSED** the smart contract audit with **HIGH MEDIUM 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

Context	Implementation			
L	_msgSender	Internal 		
L	_msgData	Internal 		
Ownable	Implementation	Context		
L		Public 		NO 
L	owner	Public 		NO 
L	renounceOwnership	Public 		onlyOwr 
L	transferOwnership	Public 		onlyOwr 
L	_setOwner	Private 		
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 
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 



# Automated Analysis

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 ⓘ
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 ⓘ
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 🔒		



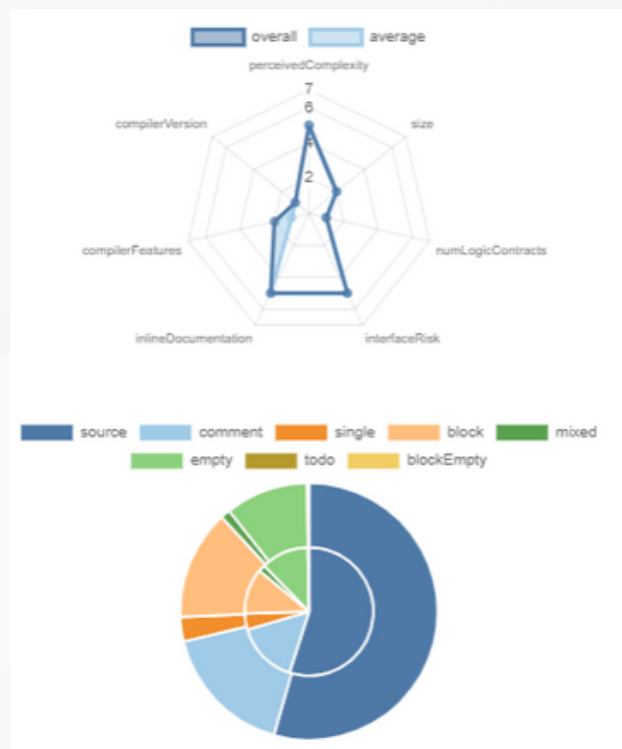
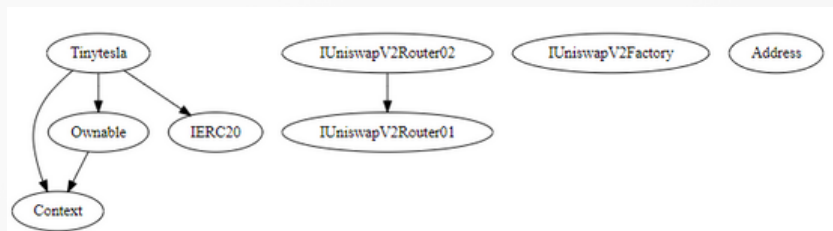
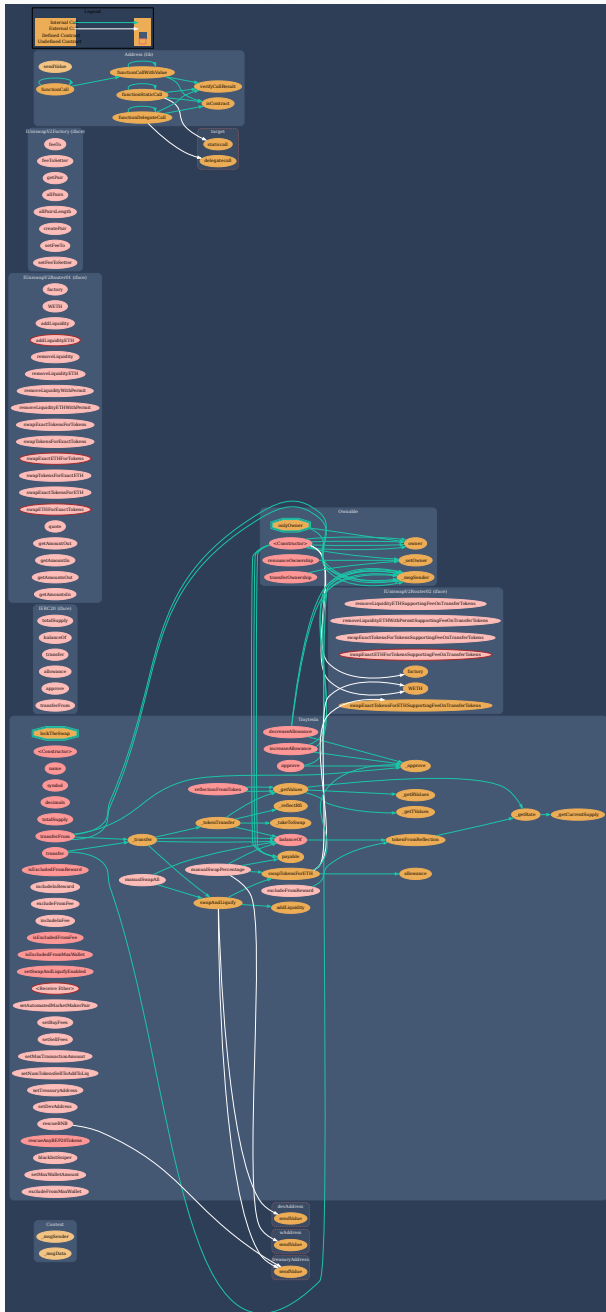
# Automated Analysis

L		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	reflectionFromToken	Public		NO
L	tokenFromReflection	Public		NO
L	excludeFromReward	External		onlyOwner
L	includeInReward	External		onlyOwner
L	excludeFromFee	External		onlyOwner
L	includeInFee	External		onlyOwner
L	isExcludedFromFee	Public		NO
L	isExcludedFromMaxWallet	Public		NO
L	setSwapAndLiquifyEnabled	Public		onlyOwner
L		External		NO
L	_reflectRfi	Private		
L	_takeToSwap	Private		
L	_getValues	Private		
L	_getTValues	Private		
L	_getRValues	Private		
L	_getRate	Private		
L	_getCurrentSupply	Private		
L	_approve	Private		
L	_transfer	Private		
L	_tokenTransfer	Private		

# Automated Analysis

L	swapAndLiquify	Private 🗝️	🔴	lockTheSwap
L	swapTokensForETH	Private 🗝️	🔴	
L	addLiquidity	Private 🗝️	🔴	
L	setAutomatedMarketMakerPair	External 🚫	🔴	onlyOwner
L	setBuyFees	External 🚫	🔴	onlyOwner
L	setSellFees	External 🚫	🔴	onlyOwner
L	setMaxTransactionAmount	External 🚫	🔴	onlyOwner
L	setNumTokensSellToAddToLiq	External 🚫	🔴	onlyOwner
L	setTreasuryAddress	External 🚫	🔴	onlyOwner
L	setDevAddress	External 🚫	🔴	onlyOwner
L	manualSwapAll	External 🚫	🔴	onlyOwner
L	manualSwapPercentage	External 🚫	🔴	onlyOwner
L	rescueBNB	External 🚫	🔴	onlyOwner
L	rescueAnyBEP20Tokens	Public 🚫	🔴	onlyOwner
L	blacklistSniper	External 🚫	🔴	onlyOwner
L	setMaxWalletAmount	External 🚫	🔴	onlyOwner
L	excludeFromMaxWallet	External 🚫	🔴	onlyOwner

# Inheritance Graph



# Contract Summary

Logic Contracts	Interfaces	Lines	nLines	nSLOC	Comment Lines	Complex. Score	Capabilities
4	4	1252	891	608	263	469	
4	4	1252	891	608	263	469	

## Components

Contracts	Libraries	Interfaces	Abstract
1	1	4	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
77	5







External	Internal	Private	Pure	View
56	76	14	10	28



## StateVariables

Total	Public
35	19

## Capabilities

Solidity Versions observed	 Experimental Features	 Can Receive Funds	 Uses Assembly	 Has Destroyable Contracts
<div>^0.8.0</div> <div>&gt;=0.6.2</div> <div>&gt;=0.5.0</div>		<div>yes</div>	<div>yes</div> <div>(2 asm blocks)</div>	

 Transfers ETH	 Low-Level Calls	 DelegateCall	 Uses Hash Functions	 ECRRecover	 New/Create/Create2
<div>yes</div>		<div>yes</div>			

 TryCatch	 Σ Unchecked
	<div>yes</div>

# Manual Review

## The owner can set fees up to 100%

```
1156 function setBuyFees(uint8 _rfi,uint8 _treasury,uint8 _dev,uint8 _lp) external onlyOwner {
1157     buyRates.rfi = _rfi;
1158     buyRates.treasury = _treasury;
1159     buyRates.dev = _dev;
1160     buyRates.lp = _lp;
1161     buyRates.toSwap = _treasury + _dev + _lp;
1162 }
1163
1164 function setSellFees(uint8 _rfi,uint8 _treasury,uint8 _dev,uint8 _lp) external onlyOwner {
1165     sellRates.rfi = _rfi;
1166     sellRates.treasury = _treasury;
1167     sellRates.dev = _dev;
1168     sellRates.lp = _lp;
1169     sellRates.toSwap = _treasury + _dev + _lp;
1170 }
```

### Recommendation

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

## The owner can't set the max tx amount "0" but can set very low amount

```
1182 function setMaxTransactionAmount(
1183     uint256 _maxTxAmountBuyPct,
1184     uint256 _maxTxAmountSellPct
1185 ) external onlyOwner {
1186     maxTxAmountBuy = _tTotal / _maxTxAmountBuyPct; // 100 = 1%, 50 = 2% etc.
1187     maxTxAmountSell = _tTotal / _maxTxAmountSellPct; // 100 = 1%, 50 = 2% etc.
1188 }
```

### Recommendation

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

## Auto liquidity is going to an externally owned account

```
1127 function addLiquidity(uint256 tokenAmount, uint256 ETHAmount) private {
1128     // add the liquidity
1129     UniswapV2Router.addLiquidityETH(value: ETHAmount)(
1130         address(this),
1131         tokenAmount,
1132         0, // slippage is unavoidable
1133         0, // slippage is unavoidable
1134         devAddress,
1135         block.timestamp
1136     );
1137     emit LiquidityAdded(tokenAmount, ETHAmount);
1138 }
1139
```

### Recommendation

Authorizing privileged roles to externally-owned-account (EOA) is dangerous. Send LP tokens to dead address or unreachable address.

# Manual Review

## The owner can't set max wallet token amount to "0" but can set very low amount

```
1225     function setMaxWalletAmount(uint256 _maxWalletAmountPct) external onlyOwner {
1226         maxWalletAmount = _tTotal / _maxWalletAmountPct; // 100 = 1%, 50 = 2% etc.
1227         emit MaxWalletAmountUpdated(maxWalletAmount);
1228     }
1229 }
```

### Recommendation

Authorizing privileged roles to change max wallet amount. Put a **require** check that allows minimum reasonable limit etc. if **set totalsupply** might cause pause trading.

## The owner can set a blacklist any account

```
1220     // Blacklist or Unblacklist bots or sniper
1221     function blacklistSniper(address botAddress, bool isban) external onlyOwner {
1222         isBot[botAddress] = isban;
1223     }
1224 }
```

### Recommendation

Authorizing privileged roles to add an account to blacklist and pause trade for any accounts. These cause can affect decentralization. remove blacklist function.

## The owner can exclude accounts from rewards

```
844     //No current rfi - Tiered Rewarding Feature Applied at APP Launch
845     function excludeFromReward(address account) external onlyOwner {
846         require(!_isExcluded[account], "Account is already excluded");
847         if (_rOwned[account] > 0) {
848             _tOwned[account] = tokenFromReflection(_rOwned[account]);
849         }
850         _isExcluded[account] = true;
851         _excluded.push(account);
852     }
```

### Recommendation

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

# Manual Review

## The owner can exclude accounts from fees

```
867     function excludeFromFee(address account) external onlyOwner {
868         _isExcludedFromFee[account] = true;
869     }
870
871
872     function includeInFee(address account) external onlyOwner {
873         _isExcludedFromFee[account] = false;
874     }
```

### Recommendation

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

## The owner can change swap settings

```
888     function setSwapAndLiquifyEnabled(bool _enabled) public onlyOwner {
889         swapAndLiquifyEnabled = _enabled;
890         emit SwapAndLiquifyEnabledUpdated(_enabled);
891     }
```

### Recommendation

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

## The owner can withdraw any token from the contract

```
1210     //Use this in case BNB are sent to the contract by mistake
1211     function rescueBNB(uint256 weiAmount) external onlyOwner{
1212         require(address(this).balance >= weiAmount, "insufficient BNB balance");
1213         treasuryAddress.sendValue(weiAmount);
1214     }
1215
1216     function rescueAnyBEP20Tokens(address _tokenAddr, address _to, uint _amount) public onlyOwner {
1217         IERC20(_tokenAddr).transfer(_to, _amount);
1218     }
1219
```

### Recommendation




Authorizing privileged roles to withdraw any token from the contract. Include native token too. Put a require check that allows **token != address(this)**



## AUDIT REPORT

# SecureWise

## SMART CONTRACT AUDIT

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

