



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

09

Inheritance
Graph

10

Contract
Summary

11

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: DogeVerseToken (**DVT**)

Methodology: Automated Analysis, Manual Code Review

Language: Solidity

Contract Address: 0xF5646ea3825eD14EE54237310cdB84b4cc84638E

ContractLink: <https://bscscan.com/address/0xf5646ea3825ed14ee54237310cdb84b4cc84638e>

Network: Binance Smart Chain (BSC)

Decimals: 18

Supply: 3.000.000.000.000

Website: <https://dogechampions.io/>

Twitter: <https://twitter.com/DogeChampions>

Telegram: <https://www.t.me/DogeChampionsNFT>

Report Date: July 22, 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



Auto liquidity is going to an externally owned account



The owner can exclude accounts from rewards



The owner can exclude accounts from fees



The owner can set fees with limit up to 25%



The owner can set max transaction amount within reasonable limits



The owner can change reward protocol address



The owner can change swap settings

The audited address **0xf5646ea3825ed14ee54237310cdb84b4cc84638e** implements the contract of the coin. The coin contains a dividends tracker feature that receives tokens from a specific tax on every transaction. The dividend tracker is connected with an NFT contract. It shares rewards proportional to the NFT holdings of every user. **The auditing of the dividend tracker and the NFT ecosystem is out of the scope of this audit.**



DogeVerseToken (DVT) has successfully **PASSED** the smart contract audit with **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

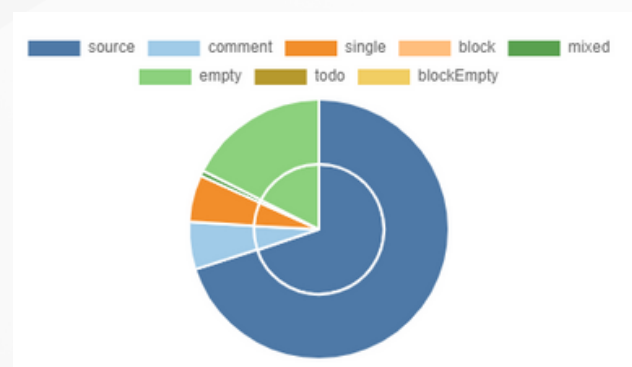
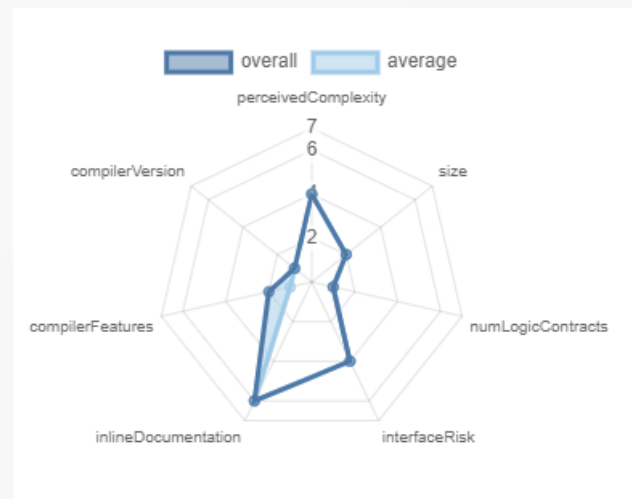
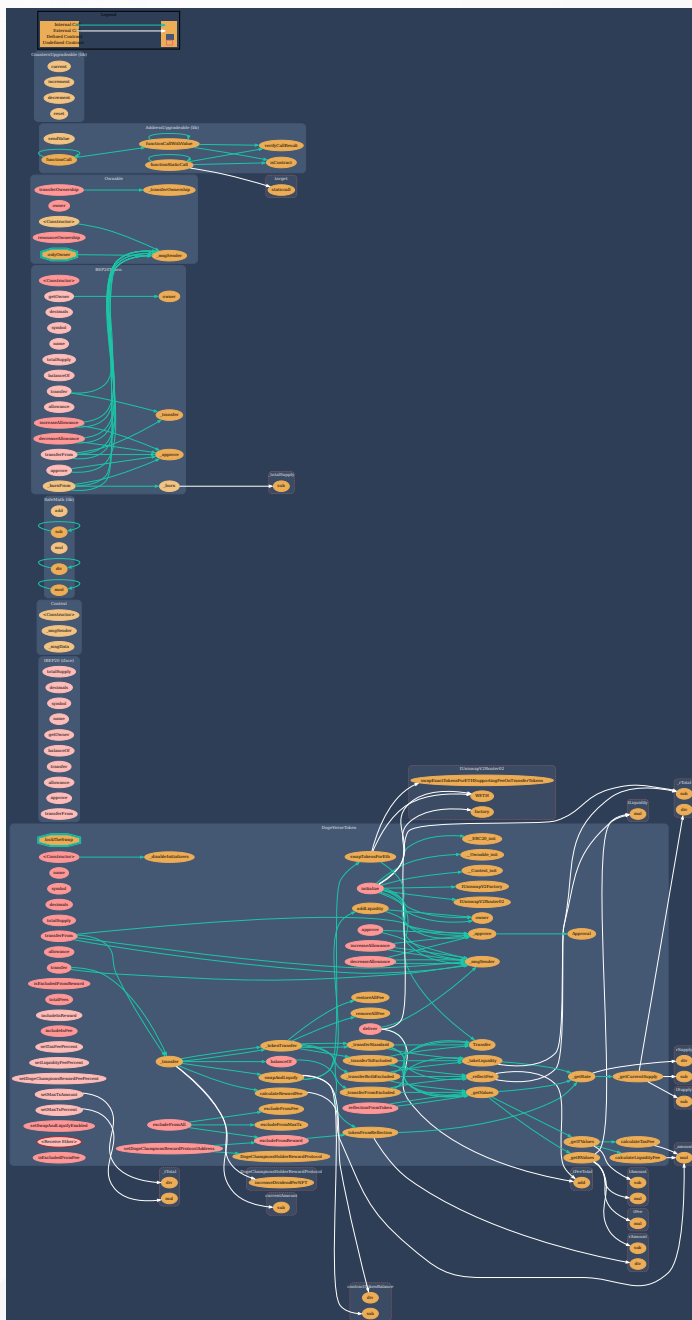
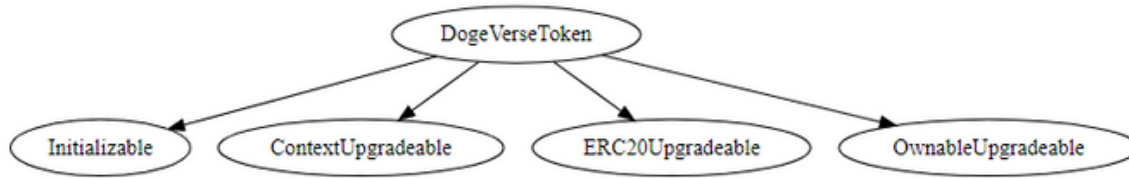
DogeVerseToken	Implementation	Initializable, ContextUpgradeable, ERC20Upgradeable, OwnableUpgradeable		
L		Public 		NO 
L	initialize	Public 		initializer
L	name	Public 		NO 
L	symbol	Public 		NO 
L	decimals	Public 		NO 
L	totalSupply	Public 		NO 
L	balanceOf	Public 		NO 
L	transfer	Public 		NO 
L	allowance	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 
L	tokenFromReflection	Public 		NO 
L	excludeFromReward	Public 		onlyOwner
L	includeInReward	External 		onlyOwner
L	_transferBothExcluded	Private 		
L	excludeFromFee	Public 		onlyOwner
L	excludeFromMaxTx	Public 		onlyOwner
L	excludeFromAll	Public 		onlyOwner
L	includeInFee	Public 		onlyOwner

Automated Analysis

L	setTaxFeePercent	External !	●	onlyOwner
L	setLiquidityFeePercent	External !	●	onlyOwner
L	setDogeChampionsRewardFeePercent	External !	●	onlyOwner
L	setMaxTxPercent	External !	●	onlyOwner
L	setMaxTxAmount	External !	●	onlyOwner
L	setSwapAndLiquifyEnabled	Public !	●	onlyOwner
L	setDogeChampionsRewardProtocolAddress	Public !	●	onlyOwner
L		External !	■	NO !
L	_reflectFee	Private 🗑	●	
L	_getValues	Private 🗑		
L	_getTVValues	Private 🗑		
L	_getRValues	Private 🗑		
L	_getRate	Private 🗑		
L	_getCurrentSupply	Private 🗑		
L	_takeLiquidity	Private 🗑	●	
L	calculateTaxFee	Private 🗑		
L	calculateLiquidityFee	Private 🗑		
L	calculateRewardFee	Private 🗑		
L	removeAllFee	Private 🗑	●	
L	restoreAllFee	Private 🗑	●	
L	isExcludedFromFee	Public !		NO !
L	_approve	Internal 🗑	●	
L	_transfer	Internal 🗑	●	
L	swapAndLiquify	Private 🗑	●	lockTheSwap

L	swapTokensForEth	Private 🗑	●	
L	addLiquidity	Private 🗑	●	
L	_tokenTransfer	Private 🗑	●	
L	_transferStandard	Private 🗑	●	
L	_transferToExcluded	Private 🗑	●	
L	_transferFromExcluded	Private 🗑	●	

Inheritance Graph



Contract Summary

Logic Contracts	Interfaces	Lines	nLines	nSLOC	Comment Lines	Complex. Score	Capabilities
1	————	566	559	421	36	337	💰
1	————	566	559	421	36	337	💰

Components

📄 Contracts	📖 Libraries	🔍 Interfaces	🗨 Abstract
1	0	0	0

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
32	1

External	Internal	Private	Pure	View
7	37	20	1	18

StateVariables

Total	🌐 Public
29	9

Capabilities

Solidity Versions observed	 Experimental Features	 Can Receive Funds	 Uses Assembly	 Has Destroyable Contracts	
<input type="text" value="^0.8.4"/>	<input type="text"/>	<input type="text" value="yes"/>	<input type="text"/>	<input type="text"/>	
 Transfers ETH	 Low-Level Calls	 DelegateCall	 Uses Hash Functions	 ECRecover	 New/Create/Create2
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
 TryCatch	Σ Unchecked				
<input type="text"/>	<input type="text"/>				

Manual Review

Auto liquidity is going to an externally owned account

```
function addLiquidity(uint256 tokenAmount, uint256 ethAmount) private {
    // approve token transfer to cover all possible scenarios
    _approve(address(this), address(uniswapV2Router), tokenAmount);

    // add the liquidity
    uniswapV2Router.addLiquidityETH{value: ethAmount}(
        address(this),
        tokenAmount,
        0, // slippage is unavoidable
        0, // slippage is unavoidable
        owner(),
        block.timestamp
    );
}
```

Recommendation

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

The owner can exclude accounts from rewards

```
218 function excludeFromReward(address account) public onlyOwner() {
219     // require(account != 0x7a250d5630B4cF539739dF2C5dAcb4c659F2488D, 'We can not exclude Uniswap router. ');
220     require(!_isExcluded[account], "Account is already excluded");
221     if(_rOwned[account] > 0) {
222         _tOwned[account] = tokenFromReflection(_rOwned[account]);
223     }
224     _isExcluded[account] = true;
225     _excluded.push(account);
226 }
227
228 function includeInReward(address account) external onlyOwner() {
229     require(_isExcluded[account], "Account is already excluded");
230     for (uint256 i = 0; i < _excluded.length; i++) {
231         if (_excluded[i] == account) {
232             _excluded[i] = _excluded[_excluded.length - 1];
233             _tOwned[account] = 0;
234             _isExcluded[account] = false;
235             _excluded.pop();
236             break;
237         }
238     }
239 }
```

Recommendation

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

Manual Review

The owner can exclude accounts from fees

```
251     function excludeFromFee(address account) public onlyOwner {
252         _isExcludedFromFee[account] = true;
253     }
```

Recommendation

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

The owner can exclude accounts from max transaction

```
255     function excludeFromMaxTx(address account) public onlyOwner {
256         _isExcludedFromMaxTx[account] = true;
257     }
```

Recommendation

Authorizing privileged roles to exclude accounts from max transaction. These cause affect decentralization and might be issue in the future for big buy and sell .

The owner can set fees with limit up to 25%

```
269     function setTaxFeePercent(uint256 taxFee) external onlyOwner() {
270         require(taxFee + _liquidityFee + _dogeChampionsRewardFee <= 25, "Total tax amount can't exceed 25% per transaction.");
271         _taxFee = taxFee;
272     }
273
274     function setLiquidityFeePercent(uint256 liquidityFee) external onlyOwner() {
275         require(_taxFee + liquidityFee + _dogeChampionsRewardFee <= 25, "Total tax amount can't exceed 25% per transaction.");
276         _liquidityFee = liquidityFee;
277     }
278
279     function setDogeChampionsRewardFeePercent(uint256 dogeChampionsRewardFee) external onlyOwner() {
280         require(_taxFee + _liquidityFee + dogeChampionsRewardFee <= 25, "Total tax amount can't exceed 25% per transaction.");
281         _dogeChampionsRewardFee = dogeChampionsRewardFee;
282     }
283
```

Recommendation

Authorizing privileged roles to can set fees with limit up to 25%.

Manual Review

The owner can set max transaction amount within reasonable limits

```
284     function setMaxTxPercent(uint256 maxTxPercent) external onlyOwner() {
285         require(maxTxPercent >= 1, "Maximum transaction percent can't be set lower than 1% of total supply. Use setMaxTxAmount for setting up to half of percent.");
286         _maxTxAmount = _tTotal.mul(maxTxPercent).div(
287             10**2
288         );
289     }
```

Recommendation

Authorizing privileged roles to set max transaction amount within reasonable limits. can't set 0 or very close to 0. so amount within reasonable limits.

The owner can change reward protocol address

```
301     function setDogeChampionsRewardProtocolAddress(address rewardProtocolAddress) public onlyOwner {
302         dogeChampionsHolderRewardProtocol = DogeChampionsHolderRewardProtocol(rewardProtocolAddress);
303         dogeChampionsRewardProtocolAddress = rewardProtocolAddress;
304         _isExcludedFromFee[dogeChampionsRewardProtocolAddress] = true;
305         _isExcludedFromMaxTx[dogeChampionsRewardProtocolAddress] = true;
306         excludeFromReward(dogeChampionsRewardProtocolAddress);
307     }
```

Recommendation

Authorizing privileged roles to change reward protocol address. it is excluded from fees and rewards. If call this function and change new address, previous address remains excluded even if it is not used a reward address. Should be removed from exclude part.

The owner can change swap settings

```
296     function setSwapAndLiquifyEnabled(bool _enabled) public onlyOwner {
297         swapAndLiquifyEnabled = _enabled;
298         emit SwapAndLiquifyEnabledUpdated(_enabled);
299     }
```

Recommendation

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

Manual Review

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

```
isExcludedFromFee  
setDogeChampionsRewardProtocolAddress  
setSwapAndLiquifyEnabled  
includeInFee  
excludeFromAll  
reflectionFromToken  
deliver  
totalFees  
isExcludedFromReward
```


Recommendation

Should use external attribute for functions never called from the contract.

AUDIT REPORT

SecureWise

SMART CONTRACT AUDIT

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

