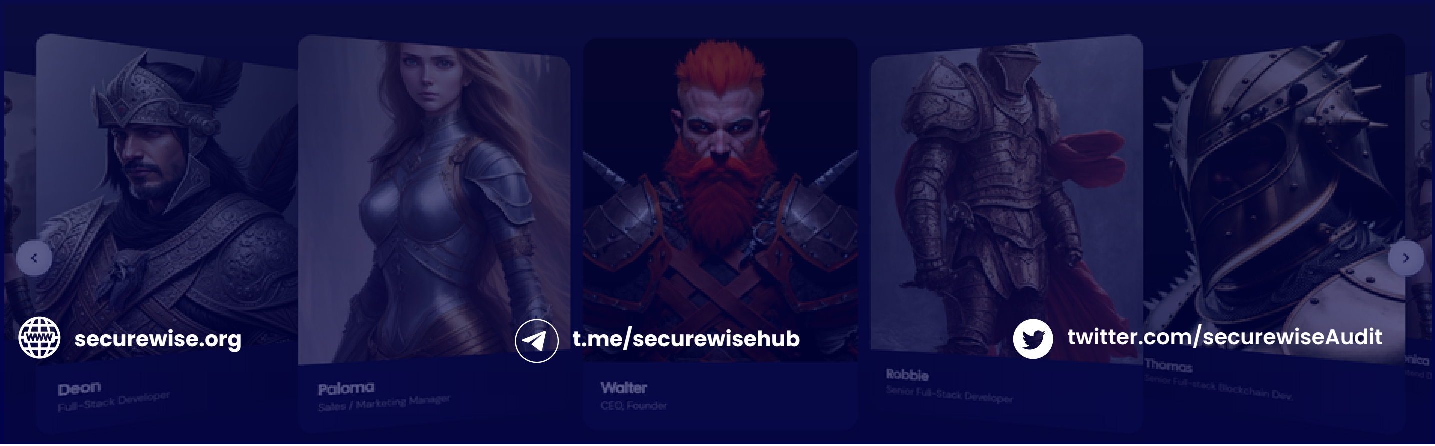
SecureWise

AUDIT REPORT

DORK LORD 2.0 (DORKL2.0)


Deon
Full-Stack Developer
Paloma
Sales / Marketing Manager
Walter
CEO, Founder
Robbie
Senior Full-Stack Developer
Thomas
Senior Full-stack Blockchain Dev

securewise.org

t.me/securewisehub

twitter.com/securewiseAudit

Quick Result

Quick Result	Status
Owner can mint new token?	Not Detected
Owner can update tax over 25% ?	Not Detected
Owner can pause trade ?	Not Detected
Owner can enable trading ?	Not Detected
Owner can add Blacklist ?	Not Detected
Owner can set Max Tx ?	Not Detected
Owner can set Max Wallet Amount ?	Not Detected
KYC ?	No

Page 6,12 for more details

Dork Lord 2.0 (DORKL2.0) as PASSED the smart contract audit with Low Risk

Findings

Risk Classification	Description
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 fxed as soon as possible.
Low	Effects are minimal in isolation and do not pose a signifcant danger to the project or its users. Issues under this classifcation are recommended to be fixed nonetheless.
Informational	A vulnerability that have informational character but is not effecting any of the code

Severity	Found	Pending	Resolved
High	0	0	0
Medium	0	0	0
Low	0	2	0
Informational	0	4	0
Total	0	6	0

Contents

Overview

Token Name: Dork Lord 2.0 (**DORKL2.0**)

Language: Solidity

Contract Address: 0x38571ebd7fa832C4eE78d09aeE52BDd8C08D2185

Network: Binance Smart Chain

Total Supply: 1000000000

KYC: No

Website: DorkLord.wiki

Twitter: https://twitter.com/DorkLord2O

Telegram:

Report Date: September 8, 2023

Testnet Link:

https://testnet.bscscan.com/address/0x50989E7C67256091aE06F89b1c5465f1e3593Dfd

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 fxed as soon as possible.

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

Informational: A vulnerability that have informational character but is not effecting any of the code

Findings Summary

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

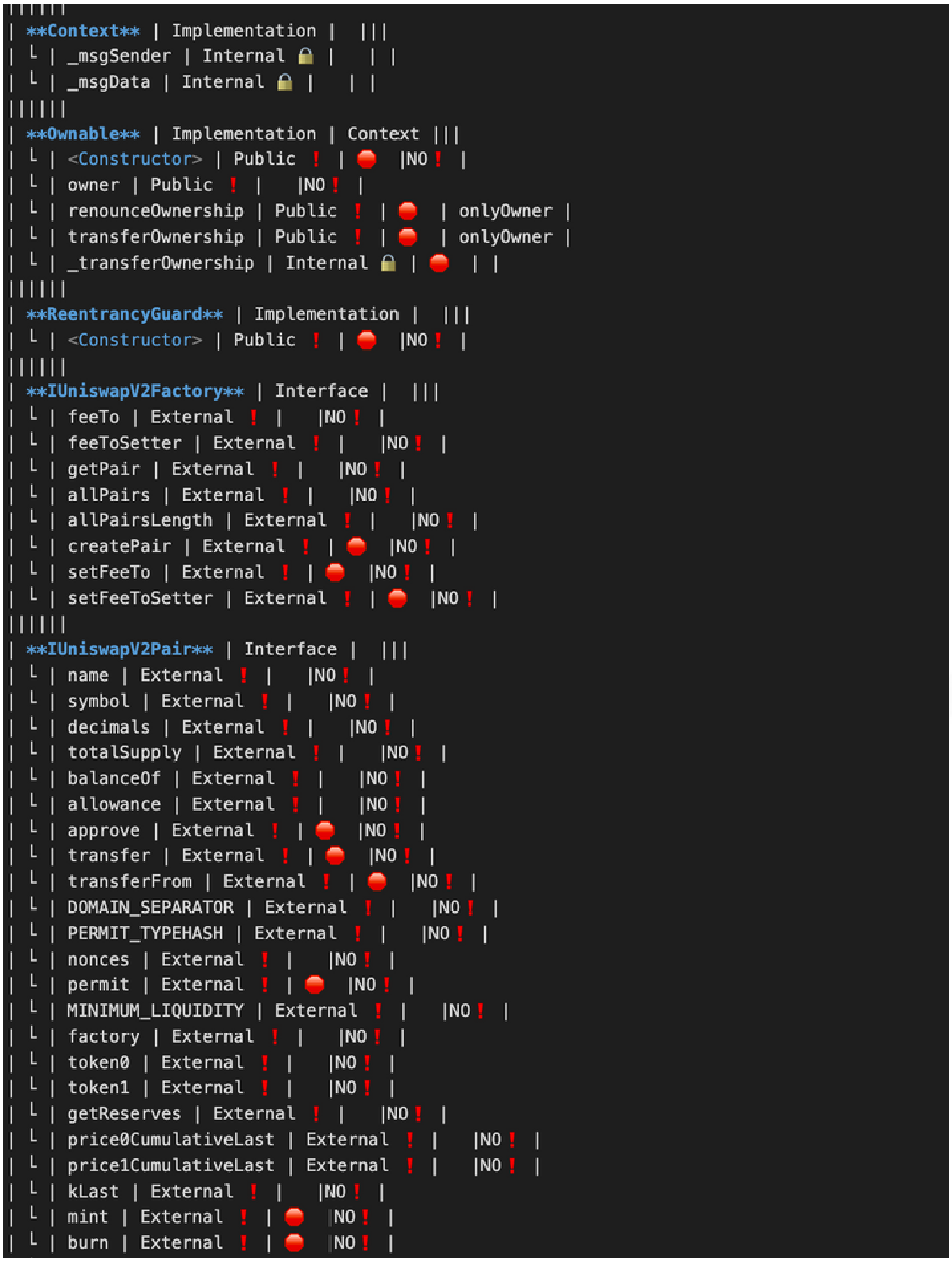
Findings

⚠	Auto liquidity goes to externally owned account
⚠	The owner has the authority to set the swap token at amount “0”
⚠	The owner has the authority to set fees max 6%
⚠	The owner has the authority to exclude/include addresses from fees.
⚠	The owner has the authority to withdraw tokens and bnb from the contract except native token
⚠	Lack of zero wallet address check on the constructor

Page 12 for more details

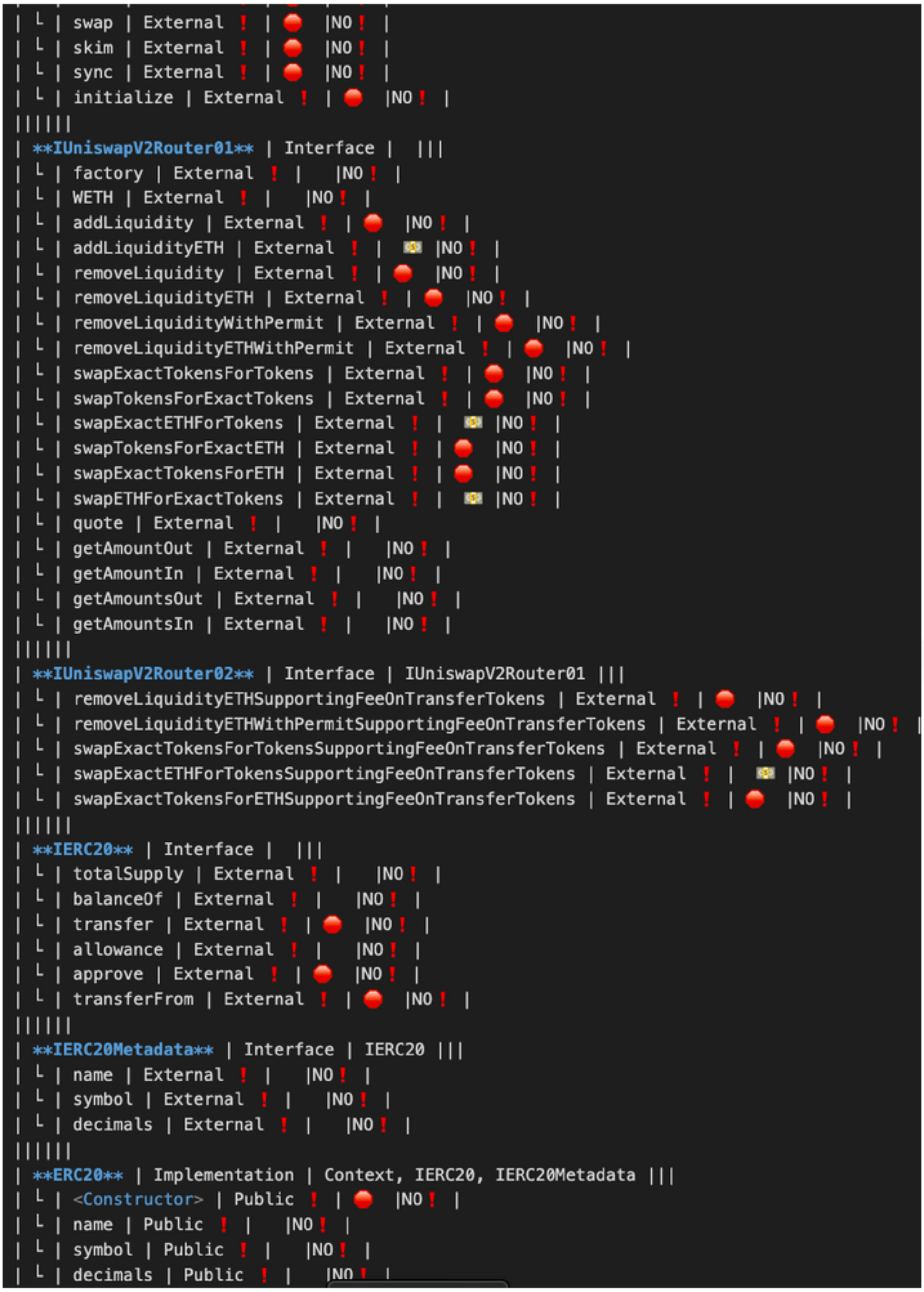
Dork Lord 2.0 (DORKL2.0) as PASSED the smart contract audit with Low Risk

Function Privileges



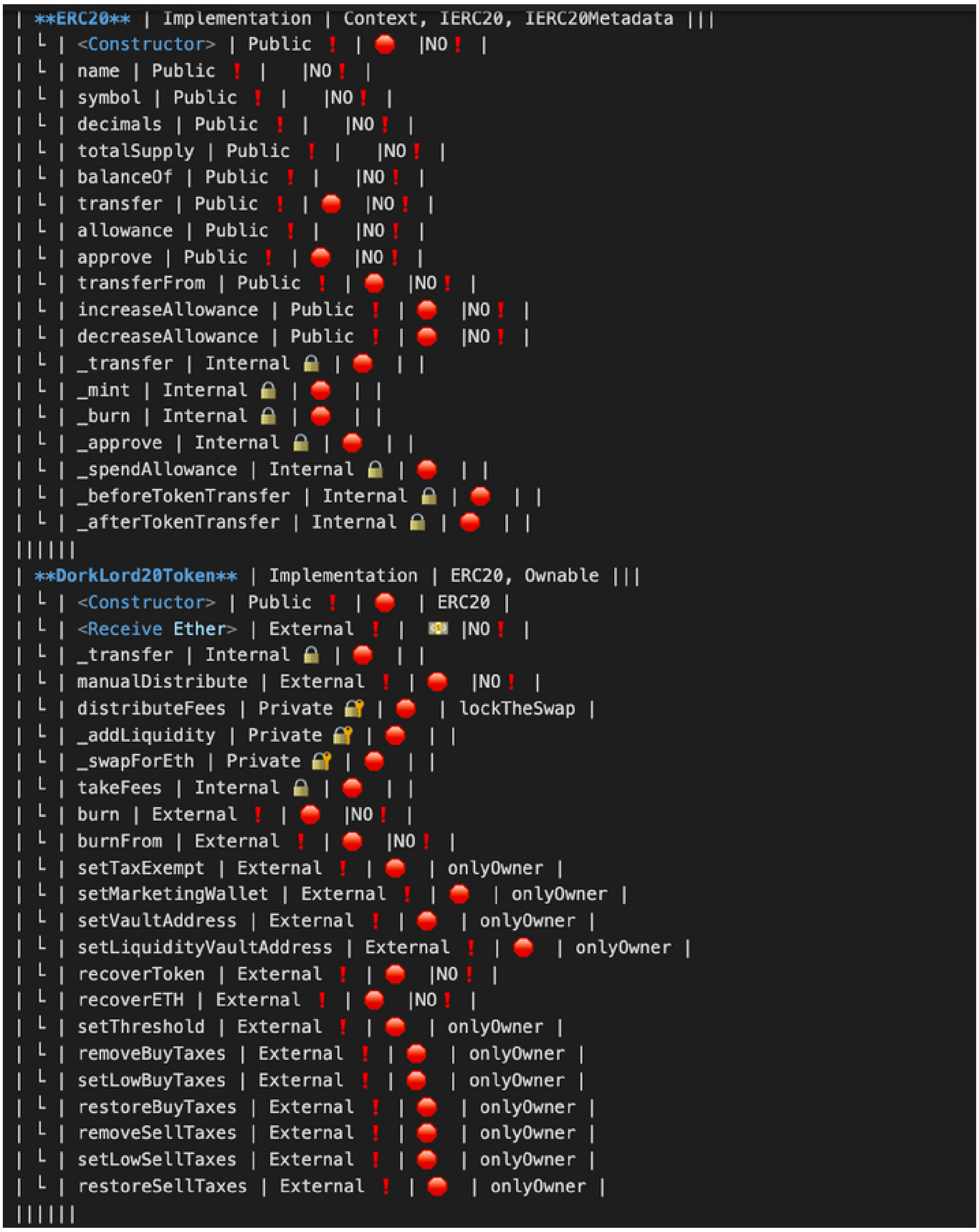
```
||||||
| **Context** | Implementation |  |||
| └ | _msgSender | Internal 🔒 |   | |
| └ | _msgData | Internal 🔒 |   | |
||||||
| **Ownable** | Implementation | Context |||
| └ | <Constructor> | Public ❗️ | 🛑  |NO❗️ |
| └ | owner | Public ❗️ |   |NO❗️ |
| └ | renounceOwnership | Public ❗️ | 🛑  | onlyOwner |
| └ | transferOwnership | Public ❗️ | 🛑  | onlyOwner |
| └ | _transferOwnership | Internal 🔒 | 🛑  | |
||||||
| **ReentrancyGuard** | Implementation |  |||
| └ | <Constructor> | Public ❗️ | 🛑  |NO❗️ |
||||||
| **IUniswapV2Factory** | Interface |  |||
| └ | feeTo | External ❗️ |   |NO❗️ |
| └ | feeToSetter | External ❗️ |   |NO❗️ |
| └ | getPair | External ❗️ |   |NO❗️ |
| └ | allPairs | External ❗️ |   |NO❗️ |
| └ | allPairsLength | External ❗️ |   |NO❗️ |
| └ | createPair | External ❗️ | 🛑  |NO❗️ |
| └ | setFeeTo | External ❗️ | 🛑  |NO❗️ |
| └ | setFeeToSetter | External ❗️ | 🛑  |NO❗️ |
||||||
| **IUniswapV2Pair** | Interface |  |||
| └ | name | External ❗️ |   |NO❗️ |
| └ | symbol | External ❗️ |   |NO❗️ |
| └ | decimals | External ❗️ |   |NO❗️ |
| └ | totalSupply | External ❗️ |   |NO❗️ |
| └ | balanceOf | External ❗️ |   |NO❗️ |
| └ | allowance | External ❗️ |   |NO❗️ |
| └ | approve | External ❗️ | 🛑  |NO❗️ |
| └ | transfer | External ❗️ | 🛑  |NO❗️ |
| └ | transferFrom | External ❗️ | 🛑  |NO❗️ |
| └ | DOMAIN_SEPARATOR | External ❗️ |   |NO❗️ |
| └ | PERMIT_TYPEHASH | External ❗️ |   |NO❗️ |
| └ | nonces | External ❗️ |   |NO❗️ |
| └ | permit | External ❗️ | 🛑  |NO❗️ |
| └ | MINIMUM_LIQUIDITY | External ❗️ |   |NO❗️ |
| └ | factory | External ❗️ |   |NO❗️ |
| └ | token0 | External ❗️ |   |NO❗️ |
| └ | token1 | External ❗️ |   |NO❗️ |
| └ | getReserves | External ❗️ |   |NO❗️ |
| └ | price0CumulativeLast | External ❗️ |   |NO❗️ |
| └ | price1CumulativeLast | External ❗️ |   |NO❗️ |
| └ | kLast | External ❗️ |   |NO❗️ |
| └ | mint | External ❗️ | 🛑  |NO❗️ |
| └ | burn | External ❗️ | 🛑  |NO❗️ |
```

Function Privileges



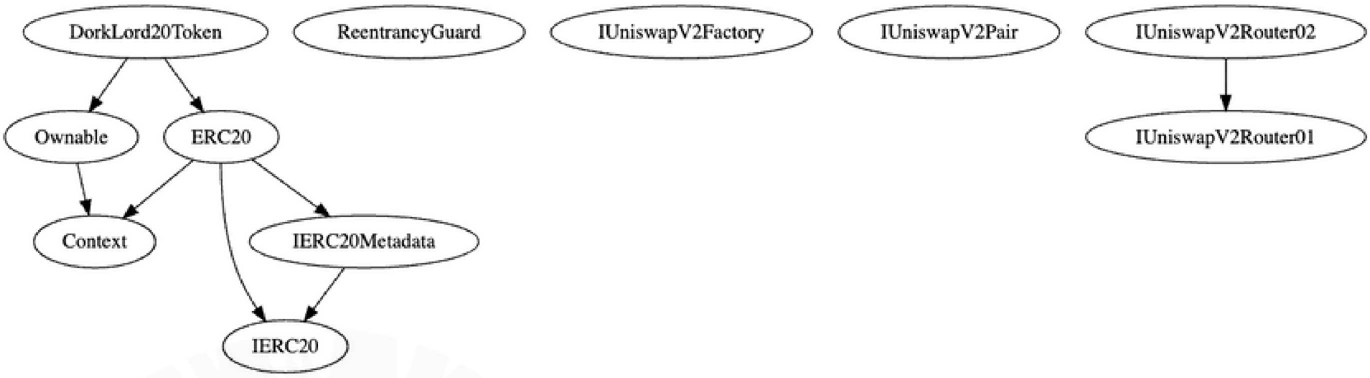
```
| └ | swap | External ❗ | 🛑  |NO❗ |
| └ | skim | External ❗ | 🛑  |NO❗ |
| └ | sync | External ❗ | 🛑  |NO❗ |
| └ | initialize | External ❗ | 🛑  |NO❗ |
||||||
| **IUniswapV2Router01** | Interface |  |||
| └ | factory | External ❗ |   |NO❗ |
| └ | WETH | External ❗ |   |NO❗ |
| └ | addLiquidity | External ❗ | 🛑  |NO❗ |
| └ | addLiquidityETH | External ❗ |  💵 |NO❗ |
| └ | removeLiquidity | External ❗ | 🛑  |NO❗ |
| └ | removeLiquidityETH | External ❗ | 🛑  |NO❗ |
| └ | removeLiquidityWithPermit | External ❗ | 🛑  |NO❗ |
| └ | removeLiquidityETHWithPermit | External ❗ | 🛑  |NO❗ |
| └ | swapExactTokensForTokens | External ❗ | 🛑  |NO❗ |
| └ | swapTokensForExactTokens | External ❗ | 🛑  |NO❗ |
| └ | swapExactETHForTokens | External ❗ |  💵 |NO❗ |
| └ | swapTokensForExactETH | External ❗ | 🛑  |NO❗ |
| └ | swapExactTokensForETH | External ❗ | 🛑  |NO❗ |
| └ | swapETHForExactTokens | External ❗ |  💵 |NO❗ |
| └ | quote | External ❗ |   |NO❗ |
| └ | getAmountOut | External ❗ |   |NO❗ |
| └ | getAmountIn | External ❗ |   |NO❗ |
| └ | getAmountsOut | External ❗ |   |NO❗ |
| └ | getAmountsIn | External ❗ |   |NO❗ |
||||||
| **IUniswapV2Router02** | Interface | IUniswapV2Router01 |||
| └ | removeLiquidityETHSupportingFeeOnTransferTokens | External ❗ | 🛑  |NO❗ |
| └ | removeLiquidityETHWithPermitSupportingFeeOnTransferTokens | External ❗ | 🛑  |NO❗ |
| └ | swapExactTokensForTokensSupportingFeeOnTransferTokens | External ❗ | 🛑  |NO❗ |
| └ | swapExactETHForTokensSupportingFeeOnTransferTokens | External ❗ |  💵 |NO❗ |
| └ | swapExactTokensForETHSupportingFeeOnTransferTokens | External ❗ | 🛑  |NO❗ |
||||||
| **IERC20** | Interface |  |||
| └ | totalSupply | External ❗ |   |NO❗ |
| └ | balanceOf | External ❗ |   |NO❗ |
| └ | transfer | External ❗ | 🛑  |NO❗ |
| └ | allowance | External ❗ |   |NO❗ |
| └ | approve | External ❗ | 🛑  |NO❗ |
| └ | transferFrom | External ❗ | 🛑  |NO❗ |
||||||
| **IERC20Metadata** | Interface | IERC20 |||
| └ | name | External ❗ |   |NO❗ |
| └ | symbol | External ❗ |   |NO❗ |
| └ | decimals | External ❗ |   |NO❗ |
||||||
| **ERC20** | Implementation | Context, IERC20, IERC20Metadata |||
| └ | <Constructor> | Public ❗ | 🛑  |NO❗ |
| └ | name | Public ❗ |   |NO❗ |
| └ | symbol | Public ❗ |   |NO❗ |
| └ | decimals | Public ❗ |   |NO❗ |
```

Function Privileges

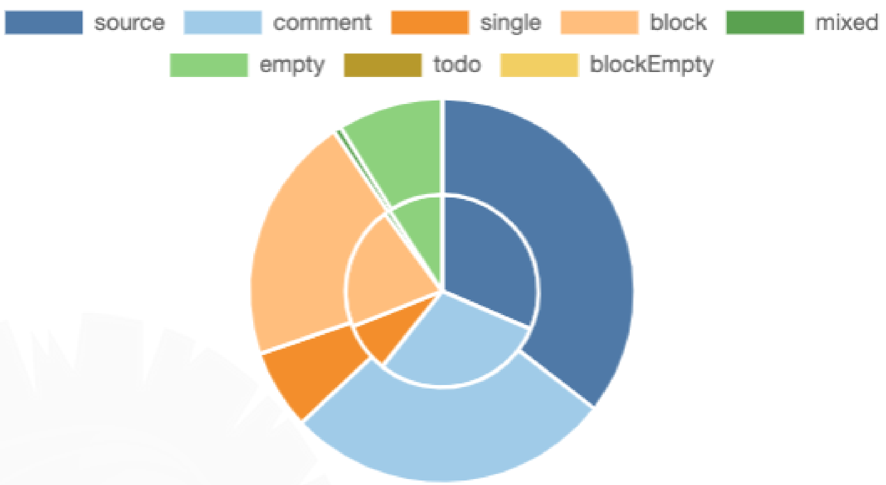


```
| **ERC20** | Implementation | Context, IERC20, IERC20Metadata |||
| └ | <Constructor> | Public ❗️ | 🛑  |NO❗️ |
| └ | name | Public ❗️ |   |NO❗️ |
| └ | symbol | Public ❗️ |   |NO❗️ |
| └ | decimals | Public ❗️ |   |NO❗️ |
| └ | totalSupply | Public ❗️ |   |NO❗️ |
| └ | balanceOf | Public ❗️ |   |NO❗️ |
| └ | transfer | Public ❗️ | 🛑  |NO❗️ |
| └ | allowance | Public ❗️ |   |NO❗️ |
| └ | approve | Public ❗️ | 🛑  |NO❗️ |
| └ | transferFrom | Public ❗️ | 🛑  |NO❗️ |
| └ | increaseAllowance | Public ❗️ | 🛑  |NO❗️ |
| └ | decreaseAllowance | Public ❗️ | 🛑  |NO❗️ |
| └ | _transfer | Internal 🔒 | 🛑  | |
| └ | _mint | Internal 🔒 | 🛑  | |
| └ | _burn | Internal 🔒 | 🛑  | |
| └ | _approve | Internal 🔒 | 🛑  | |
| └ | _spendAllowance | Internal 🔒 | 🛑  | |
| └ | _beforeTokenTransfer | Internal 🔒 | 🛑  | |
| └ | _afterTokenTransfer | Internal 🔒 | 🛑  | |
||||||
| **DorkLord20Token** | Implementation | ERC20, Ownable |||
| └ | <Constructor> | Public ❗️ | 🛑  | ERC20 |
| └ | <Receive Ether> | External ❗️ |  💵 |NO❗️ |
| └ | _transfer | Internal 🔒 | 🛑  | |
| └ | manualDistribute | External ❗️ | 🛑  |NO❗️ |
| └ | distributeFees | Private 🔐 | 🛑  | lockTheSwap |
| └ | _addLiquidity | Private 🔐 | 🛑  | |
| └ | _swapForEth | Private 🔐 | 🛑  | |
| └ | takeFees | Internal 🔒 | 🛑  | |
| └ | burn | External ❗️ | 🛑  |NO❗️ |
| └ | burnFrom | External ❗️ | 🛑  |NO❗️ |
| └ | setTaxExempt | External ❗️ | 🛑  | onlyOwner |
| └ | setMarketingWallet | External ❗️ | 🛑  | onlyOwner |
| └ | setVaultAddress | External ❗️ | 🛑  | onlyOwner |
| └ | setLiquidityVaultAddress | External ❗️ | 🛑  | onlyOwner |
| └ | recoverToken | External ❗️ | 🛑  |NO❗️ |
| └ | recoverETH | External ❗️ | 🛑  |NO❗️ |
| └ | setThreshold | External ❗️ | 🛑  | onlyOwner |
| └ | removeBuyTaxes | External ❗️ | 🛑  | onlyOwner |
| └ | setLowBuyTaxes | External ❗️ | 🛑  | onlyOwner |
| └ | restoreBuyTaxes | External ❗️ | 🛑  | onlyOwner |
| └ | removeSellTaxes | External ❗️ | 🛑  | onlyOwner |
| └ | setLowSellTaxes | External ❗️ | 🛑  | onlyOwner |
| └ | restoreSellTaxes | External ❗️ | 🛑  | onlyOwner |
||||||
```

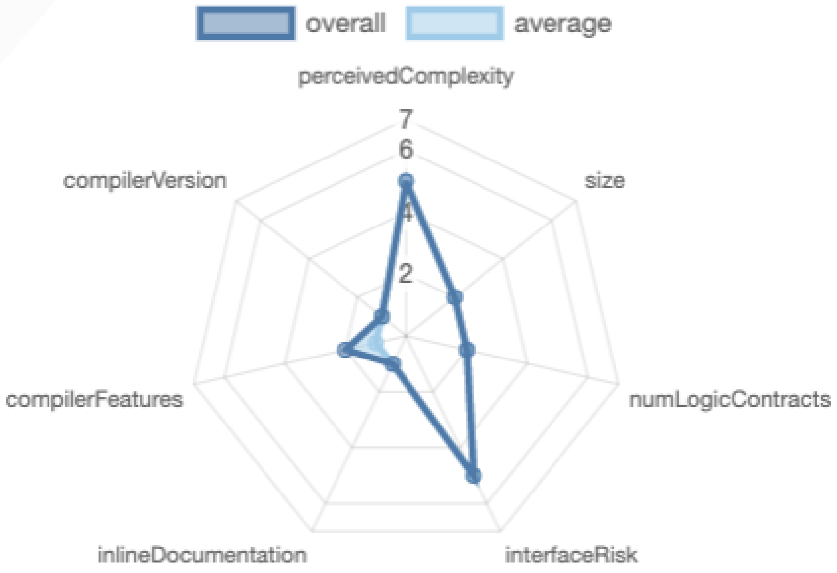
Inheritance Graph



Source Lines



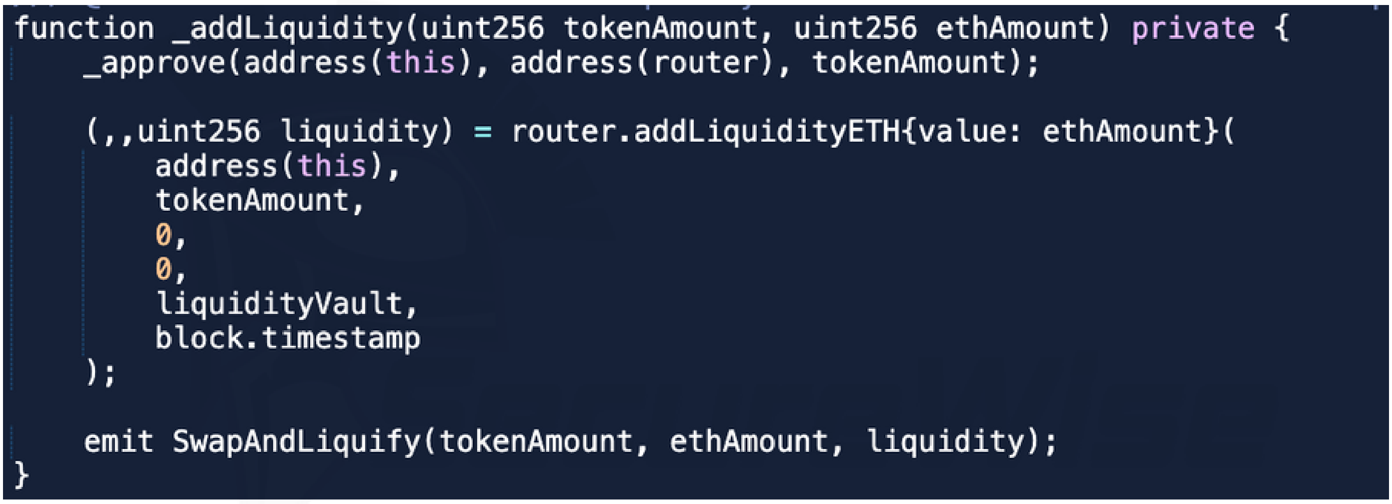
Risk



Manuel Review

Low Risk

Auto liquidity goes to externally owned account



```
function _addLiquidity(uint256 tokenAmount, uint256 ethAmount) private {
    _approve(address(this), address(router), tokenAmount);

    (,,uint256 liquidity) = router.addLiquidityETH{value: ethAmount}(
        address(this),
        tokenAmount,
        0,
        0,
        liquidityVault,
        block.timestamp
    );

    emit SwapAndLiquify(tokenAmount, ethAmount, liquidity);
}
```

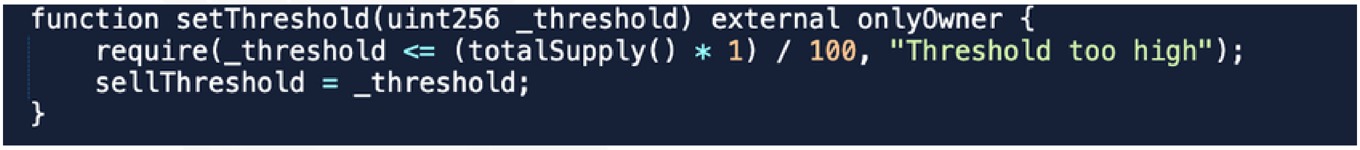
Recommendation

When liquidity tokens are sent to an EOA, there is a risk that the owner of that EOA can withdraw or manipulate the liquidity tokens without any restrictions. This loss of control can be concerning, especially if the owner of the EOA is not a trusted party. it's common to send it to a "dead" address. Dead address is an address that is known to be unspendable because no one has the private key associated with it

Manuel Review

Low Risk

The owner has the authority to set the swap token at amount “0”



```
function setThreshold(uint256 _threshold) external onlyOwner {
    require(_threshold <= (totalSupply() * 1) / 100, "Threshold too high");
    sellThreshold = _threshold;
}
```

Recommendation

If the owner can set the _threshold to "0" or very low with using the setThreshold function. Any positive balance or low balance in the contract will immediately trigger the tax distribution mechanism. Setting the threshold to zero or very low effectively eliminates the threshold condition, making it so that tax distribution occurs on every transfer involving the contract. Add a minimum threshold too requirement that prevents the owner from setting the _threshold to zero or very low.

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.

AUDIT REPORT

SecureWise

securewise.org

t.me/securewisehub

twitter.com/securewiseAudit

SecureWise Scanner

t.me/pinksalefreechecks