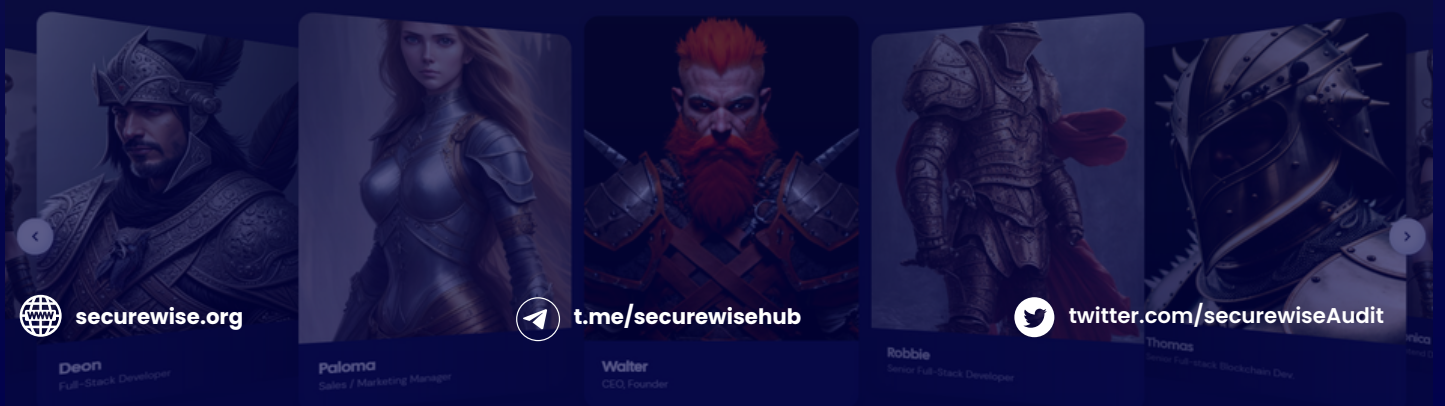




AUDIT REPORT

TOKEN NINJA (TKNJ)



 securewise.org

 t.me/securewisehub

 twitter.com/securewiseAudit

Deon
Full-Stack Developer

Paloma
Sales / Marketing Manager

Walter
CEO, Founder

Robbie
Senior Full-Stack Developer

Thomas
Senior Full-Stack Blockchain Dev

Quick Result

Quick Result	Status
Owner can mint new token?	Not Detected
Owner can update tax over 25% ?	Yes
Owner can pause trade ?	Yes
Owner can enable trading ?	Not Detected
Owner can add Blacklist ?	Pass
Owner can set Max Tx ?	Yes
Owner can set Max Wallet Amount ?	Not Detected
KYC ?	<u>Yes</u>

Page 6,12 for more details

Token Ninja (TKNJ) as **PASSED** the smart contract audit with **Critical 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 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.
Informational	A vulnerability that have informational character but is not effecting any of the code

Severity	Found	Pending	Resolved
High	0	6	0
Medium	0	0	0
Low	0	0	0
Informational	0	7	0
Total	0	13	0

Contents

01	Quick Result
02	Findings
04	Overview
05	Auditing Approach and Methodologies
06	Findings Summary
07	Function Privileges
10	Inheritance Graph
12	Manuel Review
18	Disclaimer

Overview

Token Name: Token Ninja (TKNJ)

Language: Solidity

Contract Address: 0xe8984FFF44cf219B9bA77800232e96AcF2b046cF

Network: Ethereum

Total Supply: 1000000

KYC: Yes

Website: <https://tokenninja.xyz>

Twitter: <https://twitter.com/2crazylive>

Telegram: <https://t.me/TwoCrazyNFT>

Report Date: September 4, 2023

Testnet Link:

<https://testnet.bscscan.com/address/0xeec581fa8554fa6a8b06b2bd74744a8138028e43>

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.

Informational: A vulnerability that have informational character but is not affecting 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

	The owner has the authority to set the tax rates without limit
	The owner has the authority to set the maximum transaction amount without limit
	The owner has the authority to set the uniswapV2router but there is a lack of input check
	Trading Pair Issue
	After the contract is deployed, the owner creates a pair
	The owner has the authority to set the swap token at amount without limit
	The owner has the authority to withdraw tokens from the contract
	The owner has the authority to turn off the max tx amount limits
	The owner has the authority to exclude/include addresses from max tx amount.
	The owner has the authority to turn on/off the bot protection status
	The owner has the authority to exclude/include addresses from bot protection
	The owner has the authority to exclude/include addresses from fees.
	The owner has the authority to turn on/off the tax status

Page 12 for more details

Token Ninja (TKNJ) as PASSED the smart contract audit with Critical Risk

Function Privileges

```

||||| |
| **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 ! |
| |||||
| **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 ! |
| |||||
| **Context** | Implementation | |||
| | _msgSender | Internal 🔒 | | |
| | _msgData | Internal 🔒 | | |
| |||||
| **Ownable** | Implementation | Context |||
| | <Constructor> | Public ! | ● NO ! |
| | owner | Public ! | | NO ! |
| | _checkOwner | Internal 🔒 | | |
| | renounceOwnership | Public ! | ● onlyOwner |
| | transferOwnership | Public ! | ● onlyOwner |
| | _transferOwnership | Internal 🔒 | ● |
| |||||

```


Function Privileges

```

**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 ! |
|||||
**IERC20Metadata** | Interface | IERC20 |||
| L | name | External ! | |NO ! |
| L | symbol | External ! | |NO ! |
| L | decimals | External ! | |NO ! |
|||||
**ERC20** | Implementation | Context, IERC20, IERC20Metadata |||
| L | <Constructor> | 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 | _transfer | Internal 🔒 | ● | |
| L | _mint | Internal 🔒 | ● | |
| L | _burn | Internal 🔒 | ● | |
| L | _approve | Internal 🔒 | ● | |
| L | _spendAllowance | Internal 🔒 | ● | |
| L | _beforeTokenTransfer | Internal 🔒 | ● | |
| L | _afterTokenTransfer | Internal 🔒 | ● | |
|||||
**TKNJ** | Implementation | ERC20, Ownable |||
| L | <Constructor> | Public ! | ● | ERC20 |
| L | _transfer | Internal 🔒 | ● | |
| L | splitAndSwapBack | Private 🔒 | ● | |
| L | _swapTokensForEth | Private 🔒 | ● | |
| L | burn | Public ! | ● |NO ! |
| L | emergencyTransfer | Public ! | ● | onlyOwner |
| L | excludeFromFee | Public ! | ● | onlyOwner |
| L | includeInFee | Public ! | ● | onlyOwner |
| L | excludeFromBotProtection | Public ! | ● | onlyOwner |
| L | includeInBotProtection | Public ! | ● | onlyOwner |
| L | excludeFromMaxTxAmount | Public ! | ● | onlyOwner |
| L | includeInMaxTxAmount | Public ! | ● | onlyOwner |
| L | setTeamWallet | Public ! | ● | onlyOwner |
| L | setPoolWallet | Public ! | ● | onlyOwner |
| L | setSwapTokensAtAmount | Public ! | ● | onlyOwner |
| L | deactivateMaxTransaction | Public ! | ● | onlyOwner |
| L | setTaxStatus | Public ! | ● | onlyOwner |
| L | setTaxRate | Public ! | ● | onlyOwner |

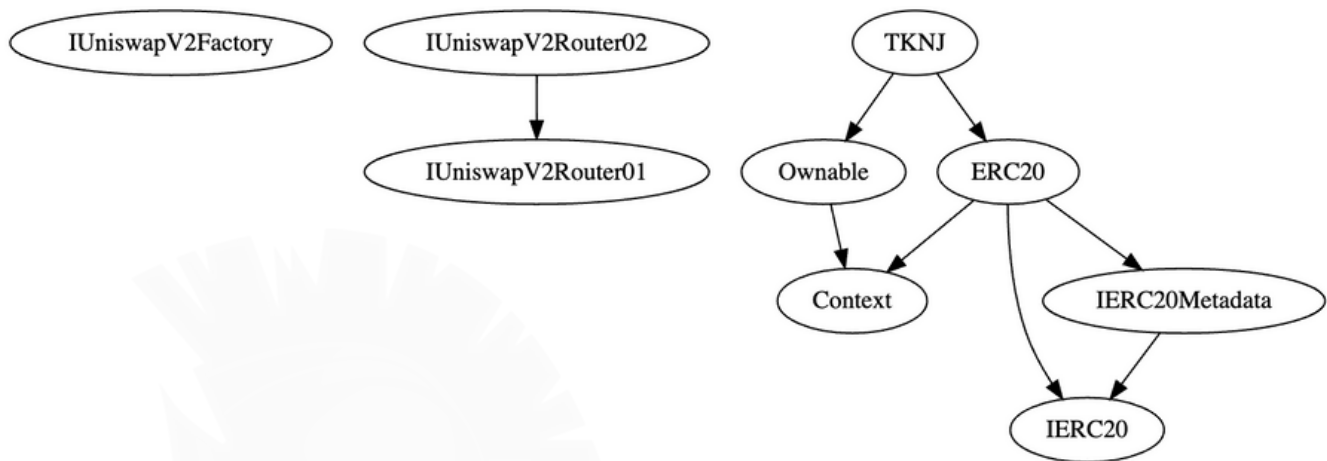
```

Function Privileges

```
| L | setMaxTxAmount | Public ! | ● | onlyOwner |
| L | setBotDetectionStatus | Public ! | ● | onlyOwner |
| L | setUniswapV2Router | Public ! | ● | onlyOwner |
| L | addPair | Public ! | ● | onlyOwner |
| L | removePair | Public ! | ● | onlyOwner |
| L | createPair | Public ! | ● | onlyOwner |
```



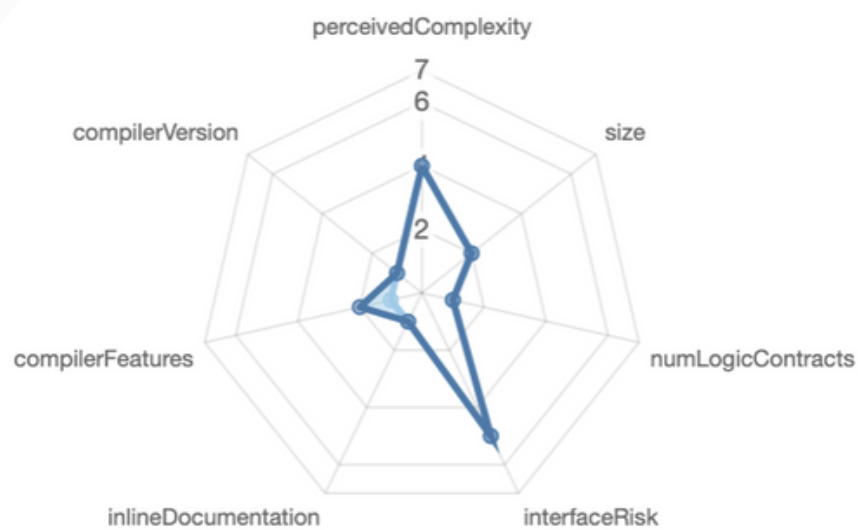
Inheritance Graph



Source Lines



Risk



Manuel Review

High Risk

The owner has the authority to set the tax rates without limit

```
function setTaxRate(uint256 newRate) public onlyOwner {  
    TAX = newRate;  
}
```

Recommendation

The owner can set tax rates without any restrictions. Mitigate the security risk associated with unlimited access to tax rate changes by the owner. Consider using a multi-signature wallet for tax rate changes. This requires multiple parties to collectively approve any changes, reducing the risk of unauthorized or malicious changes. Also Specify a maximum acceptable range for the tax rate. This range should be based on your contract's requirements and business logic.

Manuel Review

High Risk

The owner has the authority to set the maximum transaction amount without limit

```
function setMaxTxAmount(uint256 newMaxTx) public onlyOwner {  
    MAX_TX_AMOUNT = newMaxTx;  
}
```

Recommendation

If owner set newMaxTx to "0". It impossible to trade for anyone except the owner. Add an additional condition that requires the maximum transaction percentage to be greater than 0.

Manuel Review

High Risk

The owner has the authority to set the `uniswapV2router` but there is a lack of input check

```
function setUniswapV2Router(address uniswapV2Router) public onlyOwner {  
    _uniswapV2Router = uniswapV2Router;  
}
```

Recommendation

`setUniswapV2Router` function allows the owner to set the **`_uniswapV2Router`** address without performing any input validation or checks. If the owner sets an invalid or malicious address (e.g., address `0x0` or a contract that is not a valid Uniswap V2 router), it could result in a situation where no one can sell tokens except the owner, potentially causing a liquidity lock or other critical issues. Implement input validation checks within the **`setUniswapV2Router`** function to ensure that the provided address is not **`null`** or **`zero`** and is a **`valid`** contract address.

```
require(uniswapV2Router != address(0), "Invalid router address");  
require(Address.isContract(uniswapV2Router), "Router address must be a contract");
```

Manuel Review

High Risk

Trading Pair Issue

```
function addPair(address pair) public onlyOwner {  
    pairs[pair] = true;  
}  
  
function removePair(address pair) public onlyOwner {  
    pairs[pair] = false;  
}
```

Recommendation

addPair and **removePair** functions allow the owner to add and remove trading pairs, respectively. However, if the owner removes the current trading pair or makes changes to the pairs without proper checks and safeguards, it can lead to severe security risks and issues. Specifically, if the current trading pair is removed or altered, users may not be able to sell or may buy without tax, potentially compromising the contract's functionality and investor interests.

Ensure that there is a default trading pair that is set during contract deployment in the constructor. This default pair should be essential for the contract's functionality, and its removal should not be allowed. Implement restrictions on the removal of trading pairs. For example, you can restrict the removal of the current trading pair, or you can introduce a delay mechanism that prevents immediate changes to pairs. Consider using a time lock or multisignature approval for critical pair changes.

Manuel Review

High Risk

After the contract is deployed, the owner creates a pair

```
function createPair() public onlyOwner {  
    pairs[IUniswapV2Factory(IUniswapV2Router02(_uniswapV2Router).factory()).createPair(address(this), IUniswapV2Router02(_uniswapV2Router).WETH())] = true;  
}
```

Recommendation

createPair function allows the owner to create a pair after the contract is deployed. However, it is a potential issue that the pair creation is not performed in the constructor, which means the contract may not work correctly if the owner fails to call the createPair function after deployment. Ensure the contract's functionality is not dependent on the owner calling createPair separately after deployment, you should create the pair in the constructor.

Manuel Review

High Risk

The owner has the authority to set the swap token at amount without limit

```
function setSwapTokensAtAmount(uint256 amount) public onlyOwner {  
    swapTokensAtAmount = amount;  
}
```

Recommendation

swapTokensAtAmount variable represents the minimum amount of tokens required to trigger a swap for ETH, with a value of 0 indicating that no swaps should occur. The owner has the authority to set this value without any restrictions. Ensure proper control over the minimum swap amount while allowing the owner to configure this feature.

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

