



Smart Contract Audit

FOR

Tate Twoge

DATED : 04 Apr 23'



AUDIT SUMMARY

Project name – Tate Twoge

Date: 04 April, 2023

Scope of Audit- Audit Ace was consulted to conduct the smart contract audit of the solidity source codes.

Audit Status: **Passed**

Issues Found

Status	Critical	High	Medium	Low	Suggestion
Open	0	2	1	0	0
Acknowledged	0	0	0	0	0
Resolved	0	0	0	0	0



USED TOOLS

Tools:

1- Manual Review:

a line by line code review has been performed by audit ace team.

2- BSC Test Network:

all tests were done on BSC Test network, each test has its transaction has attached to it.

3- Slither : Static Analysis

Testnet Link: all tests were done using this contract, tests are done on BSC Testnet

<https://testnet.bscscan.com/address/0xA9Bd5CEF60b2016BCb84a166bBE4eCa8f444e970#code>



Token Information

Token Name : Tate Twoge

Token Symbol: TATETWOGGE

Decimals: 9

Token Supply: 100,000,000

Token Address:

0x09924d7C4f6aA63baedd23b04BdD3211eb761D35

Checksum:

b62afe267351d26c15a3f4ee29820681b8dcd41c

Owner:

0x78EB79092b63d374588a24570ed5C51b96F74F95



TOKEN OVERVIEW

Fees:

Buy Fees: upto 30%

Sell Fees: upto 30%

Transfer Fees: 0%

Fees Privilege: Owner

Ownership : Owned

Minting: No mint function

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

Other Privileges: changing fee - changing swap
threshold - excluding wallets from fees - including
wallets in fees - including in rewards - excluding from
rewards - opening trade



AUDIT METHODOLOGY

The auditing process will follow a routine as special considerations by Auditace:

- Review of the specifications, sources, and instructions provided to Auditace to make sure the contract logic meets the intentions of the client without exposing the user's funds to risk.
 - Manual review of the entire codebase by our experts, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
 - Specification comparison is the process of checking whether the code does what the specifications, sources, and instructions provided to Auditace describe.
 - Test coverage analysis determines whether the test cases are covering the code and how much code is exercised when we run the test cases.
 - Symbolic execution is analysing a program to determine what inputs cause each part of a program to execute.
 - Reviewing the codebase to improve maintainability, security, and control based on the established industry and academic practices.
-

VULNERABILITY CHECKLIST

- | | |
|--|---|
|  Return values of low-level calls |  Gasless Send |
|  Private modifier |  Using block.timestamp |
|  Multiple Sends |  Re-entrancy |
|  Using Suicide |  Tautology or contradiction |
|  Gas Limitand Loops |  Timestamp Dependence |
|  Address hardcoded |  Revert/require functions |
|  Exception Disorder |  Use of tx.origin |
|  Using inline assembly |  Integer overflow/underflow |
|  Divide before multiply |  Dangerous strict equalities |
|  Missing Zero Address Validation |  Using SHA3 |
|  Compiler version not fixed |  Using throw |
-



CLASSIFICATION OF RISK

Severity

Description

◆ Critical

These vulnerabilities could be exploited easily and can lead to asset loss, data loss, asset, or data manipulation. They should be fixed right away.

◆ High-Risk

A vulnerability that affects the desired outcome when using a contract, or provides the opportunity to use a contract in an unintended way.

◆ Medium-Risk

A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.

◆ Low-Risk

A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.

◆ Gas Optimization /Suggestion

A vulnerability that has an informational character but is not affecting any of the code.

Findings

Severity

Found

◆ Critical

0

◆ High-Risk

2

◆ Medium-Risk

1

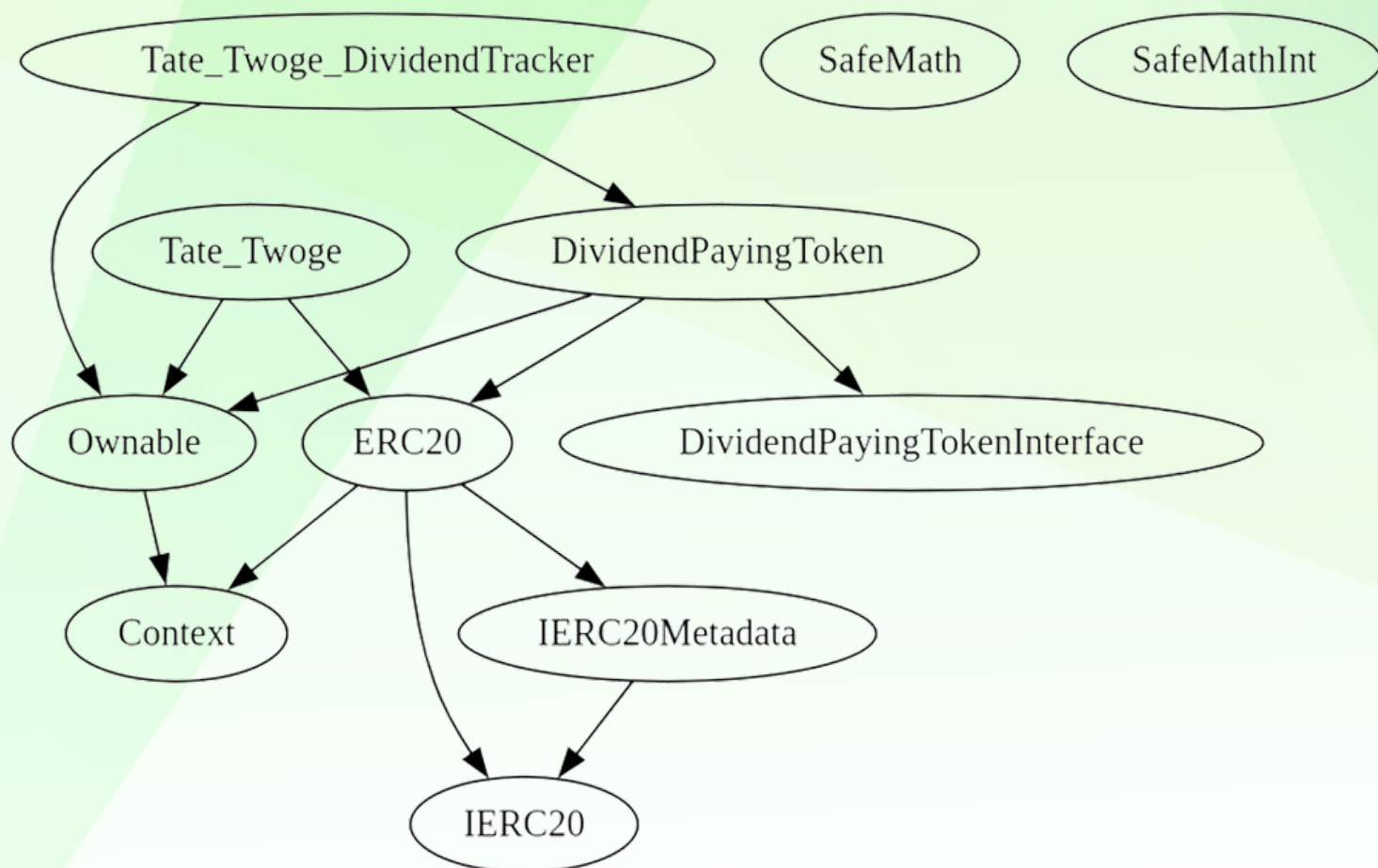
◆ Low-Risk

0

◆ Gas Optimization / Suggestions

0

INHERITANCE TREE



POINTS TO NOTE

- Owner is able to set buy/sell fees up to 30% each
 - Owner is not able to set transfer fees (0% always)
 - Owner is not able to set max buy/sell/transfer/hold amount
 - Owner is not able to blacklist an arbitrary wallet
 - Owner is not able to disable trades
 - Owner is not able to mint new tokens
 - **Owner must enable trading for investors**
-

CONTRACT ASSESMENT

Contract	Type	Bases			
:-----: :-----: :-----: :-----: :-----:					
L	**Function Name**	**Visibility**	**Mutability**	**Modifiers**	
Context Implementation					
L	_msgSender	Internal	🔒		
L	_msgData	Internal	🔒		
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	_tokengeneration	Internal	🔒		
L	_burn	Internal	🔒		
L	_approve	Internal	🔒		
L	_beforeTokenTransfer	Internal	🔒		
SafeMath Library					
L	add	Internal	🔒		

CONTRACT ASSESMENT



































```

|  | sub | Internal | | |
|  | sub | Internal | | |
|  | mul | Internal | | |
|  | div | Internal | | |
|  | div | Internal | | |
|  | mod | Internal | | |
|  | mod | Internal | | |
|  |  |  |  |  |
| **SafeMathInt** | Library | | |
|  | mul | Internal | | |
|  | div | Internal | | |
|  | sub | Internal | | |
|  | add | Internal | | |
|  | abs | Internal | | |
|  | toUint256Safe | Internal | | |
|  |  |  |  |  |
| **SafeMathUint** | Library | | |
|  | toInt256Safe | Internal | | |
|  |  |  |  |  |
| **Ownable** | Implementation | Context | | |
|  | <Constructor> | Public | | NO |
|  | owner | Public | | NO |
|  | renounceOwnership | Public | | onlyOwner |
|  | transferOwnership | Public | | onlyOwner |
|  |  |  |  |  |
| **IPair** | Interface | | |
|  | sync | External | | NO |
|  |  |  |  |  |
| **IFactory** | Interface | | |
|  | createPair | External | | NO |
|  | getPair | External | | NO |
|  |  |  |  |  |
| **IRouter** | Interface | | |
|  | factory | External | | NO |
|  | WETH | External | | NO |
|  | addLiquidityETH | External | | NO |
|  | swapExactTokensForTokensSupportingFeeOnTransferTokens | External | | NO |
|  | swapExactETHForTokens | External | | NO |
|  | swapExactTokensForETHSupportingFeeOnTransferTokens | External | | NO |
|  |  |  |  |  |
| **DividendPayingTokenInterface** | Interface | | |
|  | dividendOf | External | | NO |

```

CONTRACT ASSESMENT

```

|  | distributeDividends | External ! |  | NO ! |
|  | withdrawableDividendOf | External ! | | NO ! |
|  | withdrawnDividendOf | External ! | | NO ! |
|  | accumulativeDividendOf | External ! | | NO ! |
|  |  |
| **DividendPayingToken** | Implementation | ERC20, DividendPayingTokenInterface, Ownable | |
|  | <Constructor> | Public ! |  | ERC20 |
|  | <Receive Ether> | External ! |  | NO ! |
|  | distributeDividends | Public ! |  | NO ! |
|  | _withdrawDividendOfUser | Internal  |  | |
|  | setRewardToken | External ! |  | onlyOwner |
|  | swapBnbForCustomToken | Internal  |  | |
|  | dividendOf | Public ! | | NO ! |
|  | withdrawableDividendOf | Public ! | | NO ! |
|  | withdrawnDividendOf | Public ! | | NO ! |
|  | accumulativeDividendOf | Public ! | | NO ! |
|  | _transfer | Internal  |  | |
|  | _tokengeneration | Internal  |  | |
|  | _burn | Internal  |  | |
|  | _setBalance | Internal  |  | |
|  |  |
| **IterableMapping** | Library | | |
|  | get | Internal  | | |
|  | getIndexOfKey | Internal  | | |
|  | getKeyAtIndex | Internal  | | |
|  | size | Internal  | | |
|  | set | Internal  |  | |
|  | remove | Internal  |  | |
|  |  |
| **Address** | Library | | |
|  | sendValue | Internal  |  | |
|  |  |
| **Tate_Twoge** | Implementation | ERC20, Ownable | |
|  | <Constructor> | Public ! |  | ERC20 |
|  | <Receive Ether> | External ! |  | NO ! |
|  | updateDividendTracker | Public ! |  | onlyOwner |
|  | processDividendTracker | External ! |  | NO ! |
|  | claim | External ! |  | NO ! |
|  | rescueBEP20Tokens | External ! |  | onlyOwner |
|  | rescueBNB | External ! |  | NO ! |
|  | excludeFromFees | Public ! |  | onlyOwner |
|  | excludeMultipleAccountsFromFees | Public ! |  | onlyOwner |

```

CONTRACT ASSESMENT

```

|  | excludeFromDividends | External ! |  | onlyOwner |
|  | setMarketingWallet | External ! |  | onlyOwner |
|  | setSwapTokensAtAmount | External ! |  | onlyOwner |
|  | setBuyTaxes | External ! |  | onlyOwner |
|  | setSellTaxes | External ! |  | onlyOwner |
|  | setSwapEnabled | External ! |  | onlyOwner |
|  | OpenTrade | External ! |  | onlyOwner |
|  | setBotBlocks | External ! |  | onlyOwner |
|  | setMinBalanceForDividends | External ! |  | onlyOwner |
|  | _setAutomatedMarketMakerPair | Private  |  |
|  | setGasForProcessing | External ! |  | onlyOwner |
|  | setClaimWait | External ! |  | onlyOwner |
|  | getClaimWait | External ! |  | NO! |
|  | getTotalDividendsDistributed | External ! |  | NO! |
|  | isExcludedFromFees | Public ! |  | NO! |
|  | withdrawableDividendOf | Public ! |  | NO! |
|  | getCurrentRewardToken | External ! |  | NO! |
|  | dividendTokenBalanceOf | Public ! |  | NO! |
|  | getAccountDividendsInfo | External ! |  | NO! |
|  | getAccountDividendsInfoAtIndex | External ! |  | NO! |
|  | getLastProcessedIndex | External ! |  | NO! |
|  | getNumberOfDividendTokenHolders | External ! |  | NO! |
|  | _transfer | Internal  |  |
|  | swapAndLiquify | Private  |  |
|  | swapTokensForBNB | Private  |  |
|  | addLiquidity | Private  |  |
|  |
|  |
|  | **Tate_Twoge_DividendTracker** | Implementation | Ownable, DividendPayingToken ||
|  | <Constructor> | Public ! |  | DividendPayingToken |
|  | _transfer | Internal  |  |
|  | setMinBalanceForDividends | External ! |  | onlyOwner |
|  | excludeFromDividends | External ! |  | onlyOwner |
|  | updateClaimWait | External ! |  | onlyOwner |
|  | getLastProcessedIndex | External ! |  | NO! |
|  | getNumberOfTokenHolders | External ! |  | NO! |
|  | getCurrentRewardToken | External ! |  | NO! |
|  | getAccount | Public ! |  | NO! |
|  | getAccountAtIndex | Public ! |  | NO! |
|  | canAutoClaim | Private  |  |
|  | setBalance | Public ! |  | onlyOwner |
|  | process | Public ! |  | NO! |

```




CONTRACT ASSESMENT


| ^L | processAccount | Public ! |  | onlyOwner |

Legend

| Symbol | Meaning |

| :-----: | ----- |

|  | Function can modify state |

|  | Function is payable |



STATIC ANALYSIS

```
Function IRouter.WETH() (contracts/Token.sol#740) is not in mixedCase
Parameter DividendPayingToken.dividendOf(address). owner (contracts/Token.sol#956) is not in mixedCase
Parameter DividendPayingToken.withdrawableDividendOf(address). owner (contracts/Token.sol#964) is not in mixedCase
Parameter DividendPayingToken.withdrawnDividendOf(address). owner (contracts/Token.sol#973) is not in mixedCase
Parameter DividendPayingToken.accumulativeDividendOf(address). owner (contracts/Token.sol#984) is not in mixedCase
Constant DividendPayingToken.magnitude (contracts/Token.sol#830) is not in UPPER_CASE_WITH_UNDERSCORES
Contract Tate_Two (contracts/Token.sol#1135-1619) is not in CapWords
Parameter Tate_Two.setMarketingWallet(address). newWallet (contracts/Token.sol#1301) is not in mixedCase
Parameter Tate_Two.setBuyTaxes(uint256,uint256,uint256). rewards (contracts/Token.sol#1315) is not in mixedCase
Parameter Tate_Two.setBuyTaxes(uint256,uint256,uint256). marketing (contracts/Token.sol#1316) is not in mixedCase
Parameter Tate_Two.setBuyTaxes(uint256,uint256,uint256). liquidity (contracts/Token.sol#1317) is not in mixedCase
Parameter Tate_Two.setSellTaxes(uint256,uint256,uint256). rewards (contracts/Token.sol#1327) is not in mixedCase
Parameter Tate_Two.setSellTaxes(uint256,uint256,uint256). marketing (contracts/Token.sol#1328) is not in mixedCase
Parameter Tate_Two.setSellTaxes(uint256,uint256,uint256). liquidity (contracts/Token.sol#1329) is not in mixedCase
Parameter Tate_Two.setSwapEnabled(bool). enabled (contracts/Token.sol#1338) is not in mixedCase
Function Tate_Two.OpenTrade() (contracts/Token.sol#1342-1346) is not in mixedCase
Constant Tate_Two.deadWallet (contracts/Token.sol#1148-1149) is not in UPPER_CASE_WITH_UNDERSCORES
Contract Tate_Two.DividendTracker (contracts/Token.sol#1621-1874) is not in CapWords
Parameter Tate_Two.DividendTracker.getAccount(address). account (contracts/Token.sol#1705) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (contracts/Token.sol#15)" inContext (contracts/Token.sol#9-18)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

Variable DividendPayingToken.withdrawDividendOfUser(address). withdrawableDividend (contracts/Token.sol#887) is too similar to Tate_Two.DividendTracker.getAccount(address).withdrawableDividend
s (contracts/Token.sol#1713)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar

Tate_Two.setGasForProcessing(uint256) (contracts/Token.sol#1372-1383) uses literals with too many digits:
- require(bool,string)(newValue >= 200000 && newValue <= 500000,GasForProcessing must be between 200,000 and 500,000) (contracts/Token.sol#1373-1376)
Tate_Two.slitherConstructorVariables() (contracts/Token.sol#1135-1619) uses literals with too many digits:
- gasForProcessing = 300000 (contracts/Token.sol#1165)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

SafeMathInt.MAX_INT256 (contracts/Token.sol#594) is never used in SafeMathInt (contracts/Token.sol#592-649)
Tate_Two.currentRewardToken (contracts/Token.sol#1154) is never used in Tate_Two (contracts/Token.sol#1135-1619)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

Tate_Two.currentRewardToken (contracts/Token.sol#1154) should be constant
Tate_Two.launchtax (contracts/Token.sol#1168) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

DividendPayingToken.router (contracts/Token.sol#832) should be immutable
Tate_Two.pair (contracts/Token.sol#1139) should be immutable
Tate_Two.router (contracts/Token.sol#1138) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
```

Result => A static analysis of contract's source code has been performed using slither,

No major issues were found in the output



FUNCTIONAL TESTING

Router (PCS V2):

0xD99D1c33F9fC3444f8101754aBC46c52416550D1

All the functionalities have been tested, no issues were found

1- Adding liquidity (passed):

<https://testnet.bscscan.com/tx/0x62e40d2225cd55e7eb51daca28c635155ffc46b00ac5f400b64f94693f917070>

2- Buying when excluded (0% tax) (passed):

<https://testnet.bscscan.com/tx/0x3a300623e22dfcd6a24e60b687c4d38d3813e5e2fc4a3453f50bd6008635e03e>

3- Selling when excluded (0% tax) (passed):

<https://testnet.bscscan.com/tx/0xa1e3a325e4e6477a0752ba35f4971791b73e04df325d71d6c4186248a485abdc>

4- Transferring when excluded (0% tax) (passed):

<https://testnet.bscscan.com/tx/0x48f0da846537ee3470c48ff681d633e58fb5a4f6ecae1c4595561599d7794a51>

5- Buying when not excluded (upto 30% tax) (passed):

<https://testnet.bscscan.com/tx/0x9ba63f6d7ab12f088ee3e608119f4a1828479b147081e0f2aea74a2d81c9d1fa>

6- Selling when not excluded (upto 30% tax) (passed):

<https://testnet.bscscan.com/tx/0xe721f1de80e4e03de7b863acbf1414cc5a334860816b73a81da8588cd034894f>



FUNCTIONAL TESTING

7- Transferring when not excluded (0% tax) (passed):

<https://testnet.bscscan.com/tx/0x96f172a397aed90722f734e416d5b1cdcd40d51fe6b7bb3d8e284b76dc176a7e>

8- Internal swap (passed):

As can seen in this transaction, marketing wallet received BNB

<https://testnet.bscscan.com/address/0x6dc6c49eccc10adcdce3004763bb889d41361370#internaltx>

9- Auto Liquidity (passed):

Auto liquidity generated tokens are sent to owner's wallet

[https://testnet.bscscan.com/token/0x8b1af2de871f7d5c29f19a3d9fb66aa2dc4ffedf?
a=0x322dab6325de6f5bc2ba8efecc2bcbecac4f89f3](https://testnet.bscscan.com/token/0x8b1af2de871f7d5c29f19a3d9fb66aa2dc4ffedf?a=0x322dab6325de6f5bc2ba8efecc2bcbecac4f89f3)

10- Distribution of rewards (passed):

BUSD tokens are distributed between holders, this can be seen in this transaction

<https://testnet.bscscan.com/tx/0xbb8c146fac94637190030414214397e688a11ee398c7e5dbd1707fa52e241390>

MANUAL TESTING

Centralization - Owner must enable trading

Severity: High

Function: openTrade

Lines: 1342

Status: Not Resolved

Overview:

The owner must activate trading for investors to buy, sell, or transfer tokens. If trading remains disabled, token holders will be unable to trade their tokens.

```
function OpenTrade() external onlyOwner {  
    require(!tradingEnabled, "Trading is already enabled");  
    tradingEnabled = true;  
    startTradingBlock = block.number;  
}
```

Recommendation:

Incorporate a safety mechanism that allows investors to activate trading if a specified duration has elapsed since the conclusion of the presale or consider alternative ways such as allowing trades after investors claimed their presale tokens.

MANUAL TESTING

Centralization – Excessive max buy/sell fees

Severity: High

Function: setBuyTaxes - setSellTaxes

Lines: 1314 and 1326

Status: No Resolved

The owner has the ability to set up to 30% tax for buys and 30% tax for sells, which can result in a 60% total tax in a buy and then sell transaction if both fees are set to their maximum value. These high fees can negatively impact the token's economy and make trading the token unprofitable for investors.

```
function setBuyTaxes(
    uint256 _rewards,
    uint256 _marketing,
    uint256 _liquidity
) external onlyOwner {
    buyTaxes = Taxes(_rewards, _marketing, _liquidity);
    require(
        (_rewards + _marketing + _liquidity) <= 30,
        "Must keep fees at 30% or less"
    );
}

function setSellTaxes(
    uint256 _rewards,
    uint256 _marketing,
    uint256 _liquidity
) external onlyOwner {
    sellTaxes = Taxes(_rewards, _marketing, _liquidity);
    require(
        (_rewards + _marketing + _liquidity) <= 30,
        "Must keep fees at 30% or less"
    );
}
```

Recommendation:

In accordance with Pinksale's safu criteria, it is recommended to set a more reasonable tax limit, such as a maximum of 24% for the combined buy and sell fees. This would help maintain a healthier token economy and encourage more investors to trade the token.

MANUAL TESTING

Centralization – Owner receives LP tokens

Severity: Medium

Function: addLiquidity

Lines: 1615

Status: No Resolved

Auto liquidity generated pool tokens are sent to owner's wallet. Overtime this accumulated LP tokens will contain a larger share of liquidity and if used in a malicious way, may impact token price in a negative way.

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

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

Recommendation:

consider burning auto-liquidity generated pool tokens.



DISCLAIMER

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment. Team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed. The Auditace team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Auditace receive a payment to manipulate those results or change the awarding badge that we will be adding in our website. Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token. The Auditace team disclaims any liability for the resulting losses.



ABOUT AUDITACE

We specializes in providing thorough and reliable audits for Web3 projects. With a team of experienced professionals, we use cutting-edge technology and rigorous methodologies to evaluate the security and integrity of blockchain systems. We are committed to helping our clients ensure the safety and transparency of their digital assets and transactions.



<https://auditace.tech/>



https://t.me/Audit_Ace



https://twitter.com/auditace_



<https://github.com/Audit-Ace>
