



Smart Contract Audit

FOR
TROLLBNB

DATED : 03 May 23'



AUDIT SUMMARY

Project name - TROLLBNB

Date: 02 May, 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	3	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/token/0x134872f5Da7f2f0e8c97356f865991bB37b94CB3>



Token Information

Token Name : TROLLBNB

Token Symbol: TROLLBNB

Decimals: 18

Token Supply: 1,000,000,000

Token Address:

0xEf425A181893Fa71DE650c5E8995E34eE56637e8

Checksum:

1edf17eccc5497dee4a0678553803bd74717d58f

Owner:

0x50Eae60B6CD960a62cdc767a3A1Eb3b102CB57C1

Deployer:

0x50Eae60B6CD960a62cdc767a3A1Eb3b102CB57C1



TOKEN OVERVIEW

Fees:

Buy Fees: upto 10 %

Sell Fees: upto 10 %

Transfer Fees: 10%

Fees Privilege: none

Ownership : Owned

Minting: No mint function

Max Tx Amount/ Max Wallet Amount: yes

Blacklist: No

Other Privileges: modifying swap threshold - toggling internal swap - excluding wallets from fee - including wallets in fee



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 Limit and 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



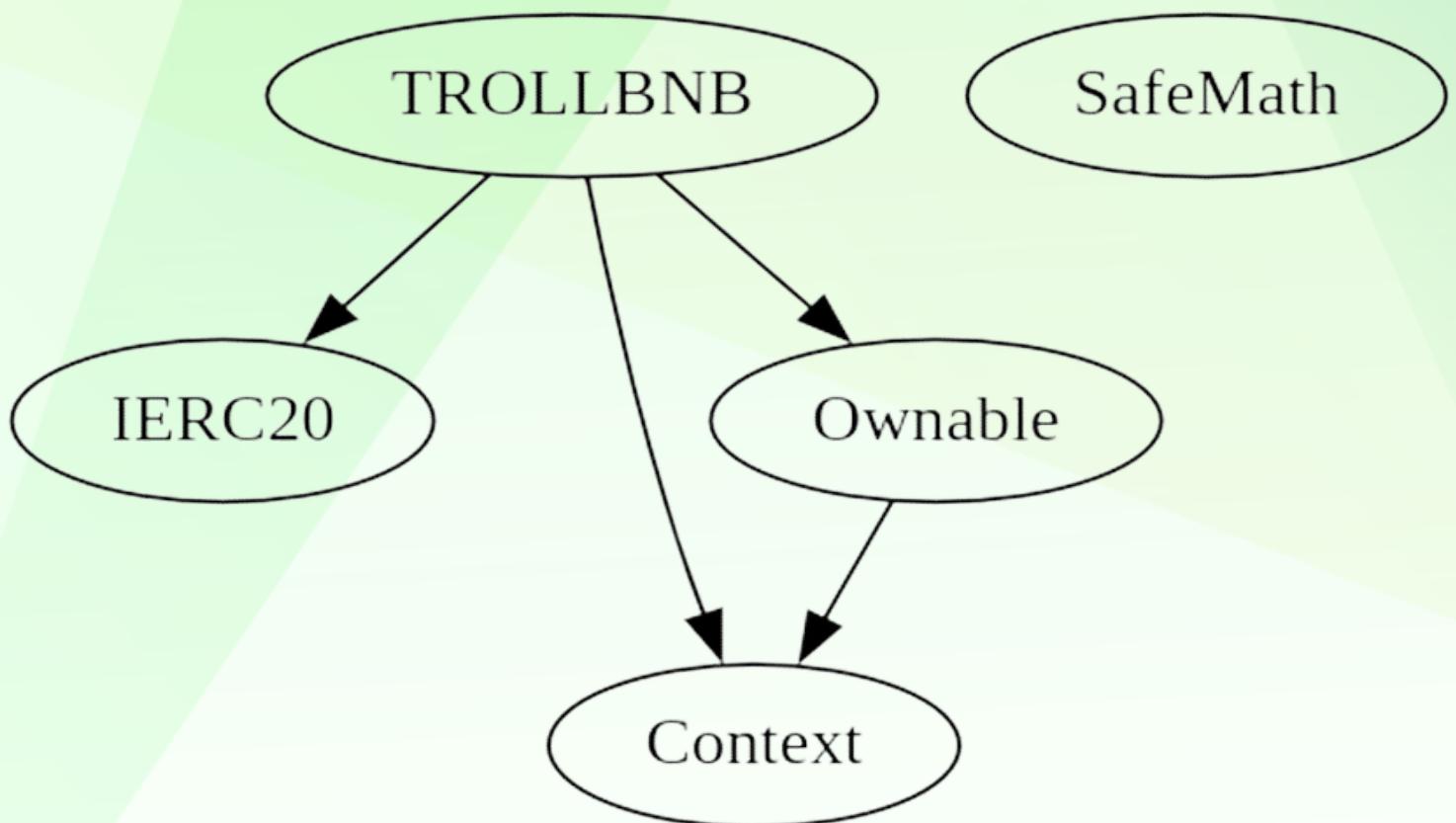
CLASSIFICATION OF RISK

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	3
◆ Medium-Risk	1
◆ Low-Risk	0
◆ Gas Optimization / Suggestions	0

INHERITANCE TREE





POINTS TO NOTE

- Owner is not able to change buy/sell/transfer taxes (10%)
- 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 has to enable trades manually for holders
- Owner is not able to mint new tokens
- Owner is able to set a max wallet/transfer/buy/sell amount.



CONTRACT ASSESSMENT

Contract	Type	Bases			
Function Name **Visibility** **Mutability** **Modifiers**					
IERC20 Interface					
totalSupply External	!	NO	!		
balanceOf External	!	NO	!		
transfer External	!		NO	!	
allowance External	!	NO	!		
approve External	!		NO	!	
transferFrom External	!		NO	!	
SafeMath Library					
add Internal					
sub Internal					
sub Internal					
mul Internal					
div Internal					
div Internal					
mod Internal					
mod Internal					
Context Implementation					
_msgSender Internal					
_msgData Internal					
Address Library					
isContract Internal					
sendValue Internal					
functionCall Internal					
functionCall Internal					
functionCallWithValue Internal					
functionCallWithValue Internal					
_functionCallWithValue Private					
Ownable Implementation Context					
<Constructor> Public	!		NO	!	
owner Public	!	NO	!		
renounceOwnership Public	!		onlyOwner		
transferOwnership Public	!		onlyOwner		
IUniswapV2Factory Interface					

CONTRACT ASSESSMENT

```
| 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! |
|||||
| **IUniswapV2Pair** | Interface | ||
| L | name | External ! | | NO! |
| L | symbol | External ! | | NO! |
| L | decimals | External ! | | NO! |
| L | totalSupply | External ! | | NO! |
| L | balanceOf | External ! | | NO! |
| L | allowance | External ! | | NO! |
| L | approve | External ! | | NO! |
| L | transfer | External ! | | NO! |
| L | transferFrom | External ! | | NO! |
| L | DOMAIN_SEPARATOR | External ! | | NO! |
| L | PERMIT_TYPEHASH | External ! | | NO! |
| L | nonces | External ! | | NO! |
| L | permit | External ! | | NO! |
| L | MINIMUM_LIQUIDITY | External ! | | NO! |
| L | factory | External ! | | NO! |
| L | token0 | External ! | | NO! |
| L | token1 | External ! | | NO! |
| L | getReserves | External ! | | NO! |
| L | price0CumulativeLast | External ! | | NO! |
| L | price1CumulativeLast | External ! | | NO! |
| L | kLast | External ! | | NO! |
| L | mint | External ! | | NO! |
| L | burn | External ! | | NO! |
| L | swap | External ! | | NO! |
| L | skim | External ! | | NO! |
| L | sync | External ! | | NO! |
| L | initialize | External ! | | NO! |
|||||
| **IUniswapV2Router01** | Interface | ||
| L | factory | External ! | | NO! |
| L | WETH | External ! | | NO! |
```



CONTRACT ASSESSMENT

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!
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!
TROLLBNB Implementation Context, IERC20, Ownable	
L <Constructor> Public !	NO!
L name External !	NO!
L symbol External !	NO!
L decimals External !	NO!
L totalSupply External !	NO!
L isExcludedFromMaxTx External !	NO!
L balanceOf Public !	NO!
L transfer External !	NO!
L allowance External !	NO!
L approve External !	NO!
L transferFrom External !	NO!
L increaseAllowance External !	NO!
L decreaseAllowance External !	NO!
L isExcludedFromReward External !	NO!
L totalFees External !	NO!

CONTRACT ASSESSMENT

```
| L | excludeFromMaxTx | External ! | ○ | onlyOwner | |
| L | deliver | External ! | ○ | NO! |
| L | reflectionFromToken | External ! | | NO! |
| L | tokenFromReflection | Public ! | | NO! |
| L | excludeFromReward | External ! | ○ | onlyOwner |
| L | includeInReward | External ! | ○ | onlyOwner |
| L | _transferBothExcluded | Private 🔒 | ○ || |
| L | excludeFromFee | External ! | ○ | onlyOwner |
| L | includeInFee | External ! | ○ | onlyOwner |
| L | setMarketingWallet | External ! | ○ | onlyOwner |
| L | setDevWallet | External ! | ○ | onlyOwner |
| L | setBurningAddress | External ! | ○ | onlyOwner |
| L | setSwapThresholdAmount | External ! | ○ | onlyOwner |
| L | allowTrading | External ! | ○ | onlyOwner |
| L | setMigrationWallet | External ! | ○ | onlyOwner |
| L | setSwapAndLiquifyEnabled | External ! | ○ | onlyOwner |
| L | setMaxTxPercent | External ! | ○ | onlyOwner |
| L | manualSwap | External ! | ○ | NO! |
| L | <Receive Ether> | External ! | 💸 | NO! |
| L | _reflectFee | Private 🔒 | ○ || |
| L | _getValues | Private 🔒 | || |
| L | _getTValues | Private 🔒 | || |
| L | _getRValues | Private 🔒 | || |
| L | _getRate | Private 🔒 | || |
| L | _getCurrentSupply | Private 🔒 | || |
| L | _takeLiquidity | Private 🔒 | ○ || |
| L | calculateTaxFee | Private 🔒 | || |
| L | calculateLiquidityFee | Private 🔒 | || |
| L | removeAllFee | Private 🔒 | ○ || |
| L | restoreAllFee | Private 🔒 | ○ || |
| L | isExcludedFromFee | External ! | | NO! |
| L | _approve | Private 🔒 | ○ || |
| L | _transfer | Private 🔒 | ○ || |
| L | swapAndLiquify | Private 🔒 | ○ | lockTheSwap |
| L | swapTokensForEth | Private 🔒 | ○ || |
| L | addLiquidity | Private 🔒 | ○ || |
| L | _tokenTransfer | Private 🔒 | ○ || |
| L | _transferStandard | Private 🔒 | ○ || |
| L | _transferToExcluded | Private 🔒 | ○ || |
| L | _transferFromExcluded | Private 🔒 | ○ ||
```

CONTRACT ASSESSMENT

Legend

Symbol	Meaning
-----	-----
 	Function can modify state Function is payable



STATIC ANALYSIS

```
Variable TROLLBNB.reflectionFromToken(uint256,bool).rTransferAmount (contracts/Token.sol#1067) is too similar to TROLLBNB._transferFromExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1502)
Variable TROLLBNB._transferFromExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1500) is too similar to TROLLBNB._getValues(uint256).tTransferAmount (contracts/Token.sol#1204)
Variable TROLLBNB._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1479) is too similar to TROLLBNB._transferStandard(address,address,uint256).tTransferAmount (contracts/Token.sol#1461)
Variable TROLLBNB._transferStandard(address,address,uint256).rTransferAmount (contracts/Token.sol#1459) is too similar to TROLLBNB._transferToExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1481)
Variable TROLLBNB._getValues(uint256).rTransferAmount (contracts/Token.sol#1208) is too similar to TROLLBNB._transferToExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1481)
Variable TROLLBNB._getValues(uint256,uint256,uint256,uint256).rTransferAmount (contracts/Token.sol#1242) is too similar to TROLLBNB._getValues(uint256).tTransferAmount (contracts/Token.sol#1204)
Variable TROLLBNB._transferStandard(address,address,uint256).rTransferAmount (contracts/Token.sol#1459) is too similar to TROLLBNB._transferBothExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1115)
Variable TROLLBNB._getValues(uint256).rTransferAmount (contracts/Token.sol#1208) is too similar to TROLLBNB._transferBothExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1115)
Variable TROLLBNB._transferStandard(address,address,uint256).rTransferAmount (contracts/Token.sol#1459) is too similar to TROLLBNB._getTValues(uint256).tTransferAmount (contracts/Token.sol#1229)
Variable TROLLBNB.reflectionFromToken(uint256,bool).rTransferAmount (contracts/Token.sol#1067) is too similar to TROLLBNB._getValues(uint256).tTransferAmount (contracts/Token.sol#1204)
Variable TROLLBNB._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1479) is too similar to TROLLBNB._transferToExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1481)
Variable TROLLBNB._getValues(uint256).rTransferAmount (contracts/Token.sol#1208) is too similar to TROLLBNB._getTValues(uint256).tTransferAmount (contracts/Token.sol#1229)
Variable TROLLBNB._getValues(uint256,uint256,uint256,uint256).rTransferAmount (contracts/Token.sol#1242) is too similar to TROLLBNB._transferStandard(address,address,uint256).tTransferAmount (contracts/Token.sol#1461)
Variable TROLLBNB._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1113) is too similar to TROLLBNB._getValues(uint256).tTransferAmount (contracts/Token.sol#1204)
Variable TROLLBNB._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1479) is too similar to TROLLBNB._transferBothExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1115)
Variable TROLLBNB._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1479) is too similar to TROLLBNB._getTValues(uint256).tTransferAmount (contracts/Token.sol#1229)
Variable TROLLBNB.reflectionFromToken(uint256,bool).rTransferAmount (contracts/Token.sol#1067) is too similar to TROLLBNB._transferStandard(address,address,uint256).tTransferAmount (contracts/Token.sol#1461)
Variable TROLLBNB._transferStandard(address,address,uint256).rTransferAmount (contracts/Token.sol#1459) is too similar to TROLLBNB._transferFromExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1502)
Variable TROLLBNB._getValues(uint256).rTransferAmount (contracts/Token.sol#1208) is too similar to TROLLBNB._transferFromExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1502)
Variable TROLLBNB._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1113) is too similar to TROLLBNB._transferStandard(address,address,uint256).tTransferAmount (contracts/Token.sol#1461)
Variable TROLLBNB._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1113) is too similar to TROLLBNB._transferToExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1481)
Variable TROLLBNB._transferFromExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1500) is too similar to TROLLBNB._transferStandard(address,address,uint256).tTransferAmount (contracts/Token.sol#1461)
Variable TROLLBNB._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1113) is too similar to TROLLBNB._transferFromExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1502)
Variable TROLLBNB._transferFromExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1500) is too similar to TROLLBNB._transferToExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1481)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar

TROLLBNB.setSwapThresholdAmount(uint256) (contracts/Token.sol#1148-1156) uses literals with too many digits:
- require(bool,string)(swapThresholdAmount > 69000000, Swap Threshold Amount cannot be less than 69 Million) (contracts/Token.sol#1151-1154)
TROLLBNB.slitherConstructorVariables() (contracts/Token.sol#853-1513) uses literals with too many digits:
- _maxTxAmount = 5000000 * 10 ** 6 * 10 ** 9 (contracts/Token.sol#898)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
```

**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/0x58aa682df13fae685b392da55b759f07e4a25611f4db3fb00959a1a8906617d8>

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

<https://testnet.bscscan.com/tx/0x148353c20b75561707774c8eafff8223ade0b37dd2847e04eeb71e9445140070>

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

<https://testnet.bscscan.com/tx/0x1ff70dba76c0019e93635f330203d935fd8e9b1fdf203da2596d4a6d0d4b1f88>

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

<https://testnet.bscscan.com/tx/0x15227b475814dbe967f4a120c8fd672a50c1b7a6d141a4ab17761309fa2498e6>

5- Buying when not excluded (10% tax) (passed):

<https://testnet.bscscan.com/tx/0x7289223f4d3d96819fae35a8cb82ba91ec0306d6fa92f440970749ab1bb3e4e6>

6- Selling when not excluded (10% tax) (passed):

<https://testnet.bscscan.com/tx/0x8b4c768ec2aba846e50e7d8488d1335c74e0fd8d059d8c9c87085eacdc11d476>



FUNCTIONAL TESTING

7- Transferring when not excluded (10% tax) (passed):

<https://testnet.bscscan.com/tx/0x169d73becd01ff7f68d3a7a9fd4acb4a144bd4462bc12ad737b7cd90e90e62b3>

8- Internal swap (auto-liquidity, fee wallets receiving bnb)

(passed):

<https://testnet.bscscan.com/tx/0x8b4c768ec2aba846e50e7d8488d1335c74e0fd8d059d8c9c87085eacdc11d476>



MANUAL TESTING

Centralization - Owner Must Enable Trades

Severity: High / Informational

Function: allowtrading

Status: Not Resolved

Overview:

The owner is required to enable trading for investors. If trading remains disabled, token holders will not have the ability to buy, sell, or transfer their tokens.

```
function allowtrading() external onlyOwner {  
    canTrade = true;  
}
```

Recommendation:

While the presence of this function is considered a feature rather than a flaw, it is crucial to highlight the centralization risk it inherently poses. To address this issue and ensure the enablement of trades, one possible solution would be to transfer the contract's ownership to a trusted third party, such as a Pinksale Safu developer. This would help mitigate the centralization risk associated with this function.



MANUAL TESTING

Centralization – Max tx excluded wallets

Severity: High

Status: Not Resolved

Overview:

The current implementation of the contract is not using `_excludedFromMaxTx` to check whether a wallet is excluded from max tx or not, instead its only checking if sender and receivers are both owner wallet (which is a wrong condition)

```
if (from != owner() && to != owner())
    require(
        amount <= _maxTxAmount,
        "Transfer amount exceeds the maxTxAmount."
    );
```

Recommendation:

Check if wallet is excluded from max tx

```
if (!_excludedFromMaxTx[from] && !_excludedFromMaxTx[to])
    require(
        amount <= _maxTxAmount,
        "Transfer amount exceeds the maxTxAmount."
    );
```



MANUAL TESTING

Logical – Tokens may become unclaimable after presale until trades are enabled

Severity: High

Status: Not Resolved

Overview:

The current implementation of the contract requires the sender of the tokens to either be the owner or the migrationWallet, if trades are not enabled yet. This means, after presale since sender is presale address, users might not be able to claim their tokens until owner enable trades

```
if (!canTrade) {  
    require(sender == owner() || sender == migrationWallet); // only  
    trade or add liquidity  
}
```

owner allowed to

Recommendation:

If trades are not enabled yet, make sure that either sender or receiver are excluded from the condition to be able to transfer tokens.



MANUAL TESTING

Logical – Owner receiving LP tokens

Severity: Medium

Function: addLiquidity

Status: Not Resolved

Overview:

Owner is receiving LP tokens from auto liquidity. This accumulated tokens can be used to remove a portion of tokens from the liquidity pool.

```
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:

There are several options to mitigate this issue

- Lock LP tokens
- Burn LP 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 specialize 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>
