



Smart Contract Audit

FOR

Pepe Inu

DATED : 26 Apr 23'



AUDIT SUMMARY

Project name – Pepe Inu

Date: 26 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	1	1	0	1
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/0x4cF85Fa65ba9B9Dc2A65bfDd04c5ae59845Bc87e#code>



Token Information

Token Name : PEPEInu

Token Symbol: PEPE

Decimals: 18

Token Supply: 1,000,000,000

Token Address:

0x882D4Fce78fC7D719EA60bE8F44A3B4587eECd5C

Checksum:

e2328b91c0aceebc3d5a4f85e2699b3e2b67de57

Owner:

0x9BcD1a58a64F2e0FaF8E4eA66beca59a3A4ac50f

Deployer:

0x9BcD1a58a64F2e0FaF8E4eA66beca59a3A4ac50f



TOKEN OVERVIEW

Fees:

Buy Fees: 4%

Sell Fees: 4%

Transfer Fees: 0%

Fees Privilege: none

Ownership : Owned

Minting: No mint function

Max Tx Amount/ Max Wallet Amount: No

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 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

1

◆ Medium-Risk

1

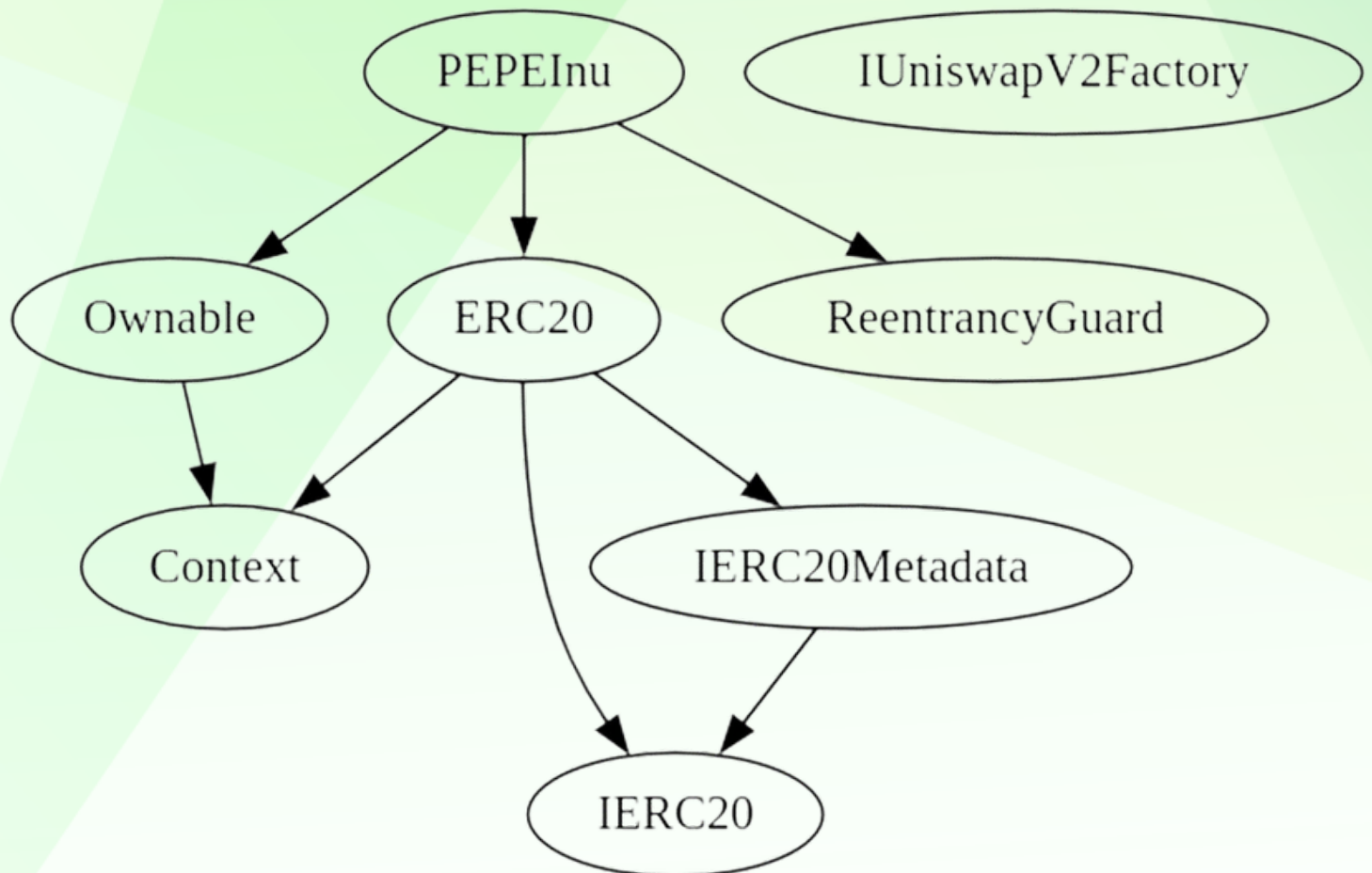
◆ Low-Risk

0

◆ Gas Optimization / Suggestions

1

INHERITANCE TREE



POINTS TO NOTE

- Owner is not able change buy/sell fees (4% for each)
 - Owner is not able to set transfer taxes (0% forever)
 - 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
 - Token doesn't have any rewards or reflections for holders
 - Trades are enabled by default
 - All fees are converted to BNB and sent to marketing wallet
-



CONTRACT ASSESMENT
















































Contract	Type	Bases			
:-----: :-----: :-----: :-----: :-----:					
L	**Function Name**	**Visibility**	**Mutability**	**Modifiers**	
Context Implementation					
L	_msgSender	Internal	🔒		
L	_msgData	Internal	🔒		
IUniswapV2Factory Interface					
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!

CONTRACT ASSESMENT

```
|  | 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! |
|||||
| **IERC20Permit** | Interface | |||
|  | permit | External ! |  | NO! |
|  | nonces | External ! | | NO! |
|  | DOMAIN_SEPARATOR | External ! | | NO! |
|||||
| **Address** | Library | |||
|  | isContract | Internal  | | |
|  | sendValue | Internal   | | |
|  | functionCall | Internal   | | |
```

CONTRACT ASSESMENT






```

|  | functionCall | Internal |  |  | | | | |
|  | functionCallWithValue | Internal |  |  | |
|  | functionCallWithValue | Internal |  |  | |
|  | functionStaticCall | Internal |  | | |
|  | functionStaticCall | Internal |  | | |
|  | functionDelegateCall | Internal |  |  | |
|  | functionDelegateCall | Internal |  |  | |
|  | verifyCallResultFromTarget | Internal |  | | |
|  | verifyCallResult | Internal |  | | |
|  | _revert | Private |  | | |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  | **IERC20** | Interface | | | |
|  |  | totalSupply | External |  | | | NO! |
|  |  | balanceOf | External |  | | | NO! |
|  |  | transfer | External |  |  | | NO! |
|  |  | allowance | External |  | | | NO! |
|  |  | approve | External |  |  | | NO! |
|  |  | transferFrom | External |  |  | | NO! |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  | **SafeERC20** | Library | | | |
|  |  |  | safeTransfer | Internal |  |  | |
|  |  |  | safeTransferFrom | Internal |  |  | |
|  |  |  | safeApprove | Internal |  |  | |
|  |  |  | safeIncreaseAllowance | Internal |  |  | |
|  |  |  | safeDecreaseAllowance | Internal |  |  | |
|  |  |  | safePermit | Internal |  |  | |
|  |  |  | _callOptionalReturn | Private |  |  | |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  | **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! |
|  |  |  | totalSupply | Public |  | | | NO! |
|  |  |  | balanceOf | Public |  | | | NO! |
|  |  |  | transfer | Public |  |  | | NO! |
|  |  |  | allowance | Public |  | | | NO! |

```

CONTRACT ASSESMENT

```

|  | 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  |  | |
|  |  |  |  |  |
| **Ownable** | Implementation | Context | |
|  | <Constructor> | Public ! |  | NO! |
|  | owner | Public ! | | NO! |
|  | _checkOwner | Internal  | | |
|  | renounceOwnership | Public ! |  | onlyOwner |
|  | transferOwnership | Public ! |  | onlyOwner |
|  | _transferOwnership | Internal  |  | |
|  |  |  |  |  |
| **ReentrancyGuard** | Implementation | | |
|  | <Constructor> | Public ! |  | NO! |
|  | _nonReentrantBefore | Private  |  | |
|  | _nonReentrantAfter | Private  |  | |
|  | _reentrancyGuardEntered | Internal  | | |
|  |  |  |  |  |
| **PEPEInu** | Implementation | ERC20, Ownable, ReentrancyGuard | |
|  | <Constructor> | Public ! |  | ERC20 |
|  | <Receive Ether> | External ! |  | NO! |
|  | <Fallback> | External ! |  | NO! |
|  | getRouterAddress | Public ! | | NO! |
|  | claimStuckTokens | External ! |  | onlyOwner |
|  | excludeFromFees | External ! |  | onlyOwner |
|  | isExcludedFromFees | Public ! | | NO! |
|  | setAutomatedMarketMakerPair | Public ! |  | onlyOwner |
|  | isAutomatedMarketMakerPair | Public ! | | NO! |
|  | toggleSwapBack | External ! |  | onlyOwner |
|  | setSwapTokensAtAmount | External ! |  | onlyOwner |
|  | _transfer | Internal  |  | |
|  | _swapBack | Internal  |  | |
|  | sendBNB | Internal  |  | nonReentrant |

```




CONTRACT ASSESMENT


| ^L | manualSwapBack | External  |  | NO  |

Legend

| Symbol | Meaning |

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

|  | Function can modify state |

|  | Function is payable |



STATIC ANALYSIS

```
Address.functionStaticCall(address,bytes,string) (contracts/Token.sol#454-467) is never used and should be removed
Address.sendValue(address,uint256) (contracts/Token.sol#372-383) is never used and should be removed
Address.verifyCallResult(bool,bytes,string) (contracts/Token.sol#512-522) is never used and should be removed
Context._msgData() (contracts/Token.sol#14-16) is never used and should be removed
ERC20._burn(address,uint256) (contracts/Token.sol#823-838) is never used and should be removed
ReentrancyGuard._reentrancyGuardEntered() (contracts/Token.sol#952-954) is never used and should be removed
SafeERC20.safeApprove(ERC20,address,uint256) (contracts/Token.sol#590-603) is never used and should be removed
SafeERC20.safeDecreaseAllowance(ERC20,address,uint256) (contracts/Token.sol#621-642) is never used and should be removed
SafeERC20.safeIncreaseAllowance(ERC20,address,uint256) (contracts/Token.sol#605-619) is never used and should be removed
SafeERC20.safePermit(IERC20Permit,address,address,uint256,uint256,uint8,bytes32,bytes32) (contracts/Token.sol#644-661) is never used and should be removed
SafeERC20.safeTransferFrom(IERC20,address,address,uint256) (contracts/Token.sol#578-588) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.17 (contracts/Token.sol#7) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.16
solc-0.8.19 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (contracts/Token.sol#372-383):
- (success) = recipient.call{value: amount}() (contracts/Token.sol#378)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (contracts/Token.sol#420-440):
- (success,returndata) = target.call{value: value}(data) (contracts/Token.sol#430-432)
Low level call in Address.functionStaticCall(address,bytes,string) (contracts/Token.sol#454-467):
- (success,returndata) = target.staticcall(data) (contracts/Token.sol#459)
Low level call in Address.functionDelegateCall(address,bytes,string) (contracts/Token.sol#481-494):
- (success,returndata) = target.delegatecall(data) (contracts/Token.sol#486)
Low level call in PEPEInu.sendBNB(address,uint256) (contracts/Token.sol#1177-1186):
- (success) = address(_to).call{value: amount}() (contracts/Token.sol#1183)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function IUniswapV2Pair.DOMAIN_SEPARATOR() (contracts/Token.sol#83) is not in mixedCase
Function IUniswapV2Pair.PERMIT_TYPEHASH() (contracts/Token.sol#85) is not in mixedCase
Function IUniswapV2Pair.MINIMUM_LIQUIDITY() (contracts/Token.sol#116) is not in mixedCase
Function IUniswapV2Router01.WETH() (contracts/Token.sol#158) is not in mixedCase
Function IERC20Permit.DOMAIN_SEPARATOR() (contracts/Token.sol#364) is not in mixedCase
Parameter PEPEInu.sendBNB(address,uint256)._to (contracts/Token.sol#1177) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Variable IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountADesired (contracts/Token.sol#163) is too similar to IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountBDesired (contracts/Token.sol#164)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar

PEPEInu.marketingWallet (contracts/Token.sol#965) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

PEPEInu.buyTax (contracts/Token.sol#961) should be immutable
PEPEInu.marketingWalletShares (contracts/Token.sol#967) should be immutable
PEPEInu.sellTax (contracts/Token.sol#962) should be immutable
PEPEInu.setSwapTokensLimit (contracts/Token.sol#970) should be immutable
PEPEInu.taxDenominator (contracts/Token.sol#960) should be immutable
PEPEInu.totalTax (contracts/Token.sol#963) should be immutable
PEPEInu.uniswapV2Pair (contracts/Token.sol#975) should be immutable
PEPEInu.uniswapV2Router (contracts/Token.sol#974) 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/0x75043e29f6f19ac7ddf5edcf73aed60ae8f0b28e12fa042c693bd4606c0ea23d>

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

<https://testnet.bscscan.com/tx/0x605659aa6173c9ad186bc2bea5e15bf7dc24533e8f4be445a6e2809904cd3ca5>

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

<https://testnet.bscscan.com/tx/0xe7f73a170c947035da6c4becc2836042764abdde08000b12d6e9b81a7ea33c36>

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

<https://testnet.bscscan.com/tx/0x3b1ac5c1d20ceea910551e933becd067cc78cfaae57f554c6bf51ef7b014b3ed>

5- Buying when not excluded (4% tax) (passed):

<https://testnet.bscscan.com/tx/0x8b28a3e41f48d67c1b7ac8d9324c5affdf50a5c79a9d8b978183db829151351d>

6- Selling when not excluded (4% tax) (passed):

<https://testnet.bscscan.com/tx/0x50ee2ad770d19db23c0357cb82d279af7e55238f6cda41af29b230cba9259692>



FUNCTIONAL TESTING

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

<https://testnet.bscscan.com/tx/0x40d3d964aca461c9bcd7a655b3dd03957aef8902e25f8022774c2b93d5e7dff8>

8- Internal swap (passed):

Marketing wallet received BNB

<https://testnet.bscscan.com/address/0x550d207ff87fe727bb10c2bfd731beb5910f2ebd#internaltx>

MANUAL TESTING

Logical – Unauthorized internal swap

Severity: **Medium**

Function: manualSwapBack

Lines: 1225

Status: Not Resolved

Overview:

The **manualSwapBack** function can be called by any user, even when internal swaps are disabled by the owner through the **toggleSwapBack** function. This could create a scenario where the owner disables internal swaps to prevent contract sales and stabilize the price. However, as fees accumulate in the contract, a malicious actor may call the manualSwapBack function, resulting in a large contract sale and a significant price reduction.

```
function manualSwapBack() external {
    uint256 contractTokenBalance = balanceOf(address(this));

    require(contractTokenBalance > 0, "Cant Swap Back 0 Token!");

    _swapBack(contractTokenBalance);
}

function toggleSwapBack(bool status) external onlyOwner {
    require(isSwapBackEnabled != status, "SwapBack already on status");

    isSwapBackEnabled = status;
    emit SwapBackStatusChanged(status);
}
```

Recommendation:

Ensure that manualSwapBack function can only be called by contract owner

```
function manualSwapBack() external onlyOwner {
    uint256 contractTokenBalance = balanceOf(address(this));

    require(contractTokenBalance > 0, "Cant Swap Back 0 Token!");

    _swapBack(contractTokenBalance);
}
```

MANUAL TESTING

Logical – Swap threshold not used

Severity: Low

Function: _transfer

Lines: 1162

Status: Not Resolved

Overview:

One of the benefits of having a swap threshold is to limit the contract's internal sales, preventing high slippage for buy/sell transactions and avoiding the sale of excessive token amounts that could negatively impact the price. The current implementation of the contract does not use the swapThreshold value to determine the number of tokens to be swapped for BNB. Instead, it utilizes the entire contract balance:

```
if (
    canSwap &&
    !swapping &&
    _isAutomatedMarketMakerPair[to] &&
    isSwapBackEnabled
){
    _swapBack(contractTokenBalance);
}
```

Recommendation:

Modify the _transfer function to use swapThreshold instead of the entire contract balance when swapping tokens for BNB. This ensures that only a specified amount of tokens are swapped at a time, preventing excessive token sales and protecting the token price:

```
if (
    canSwap &&
    !swapping &&
    _isAutomatedMarketMakerPair[to] &&
    isSwapBackEnabled
){
    _swapBack(swapTokensAtAmount);
}
```



MANUAL TESTING

Gas Optimizations:

- define $\text{buyTax} - \text{sellTax} - \text{totalTax} - \text{taxDenominator} - \text{marketingWallet} - \text{marketingWalletShare}$
as **constant**



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>
