AuditAce

FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

PEPE DAO

DATED : 7 MAY 23'

AUDIT SUMMARY

Project name – PEPE DAO

Date: 7 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	1	0	0	1
Acknowledged	0	0	0	0	0
Resolved	0	0	0	0	0

USED TOOLS

Tools:

1.Manual Review: The code has undergone a line-by-line review by the Ace team.

2.BSC Test Network: All tests were conducted on the BSC Test network, and each test has a corresponding transaction attached to it. These tests can be found in the "Functional Tests" section of the report.

3.Slither: The code has undergone static analysis using Slither.

Token Information

Name : PEPE DAO

Symbol : PEPED

Decimals: 18

Network: Binance smart chain

Token Type: BEP20

Token Address :
0x181C0f81d56102f64EC805cA444638b18a191dB3

Owner:
0x069a9310B3CB158B918D7ba94086a70b54D286aA

Deployer:
0x069a9310B3CB158B918D7ba94086a70b54D286aA

Token Information

Fees:

Buy Fees: 0%

Sell Fees: 0%

Transfer Fees: 0%

Fees Privilige: Owner

Ownership : Owned

Minting: No

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

Other Priviliges: Including or excluding from fees - changing swap threshold

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 isexercised 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.

auditace.tech

VULNERABILITY CHECKLIST

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

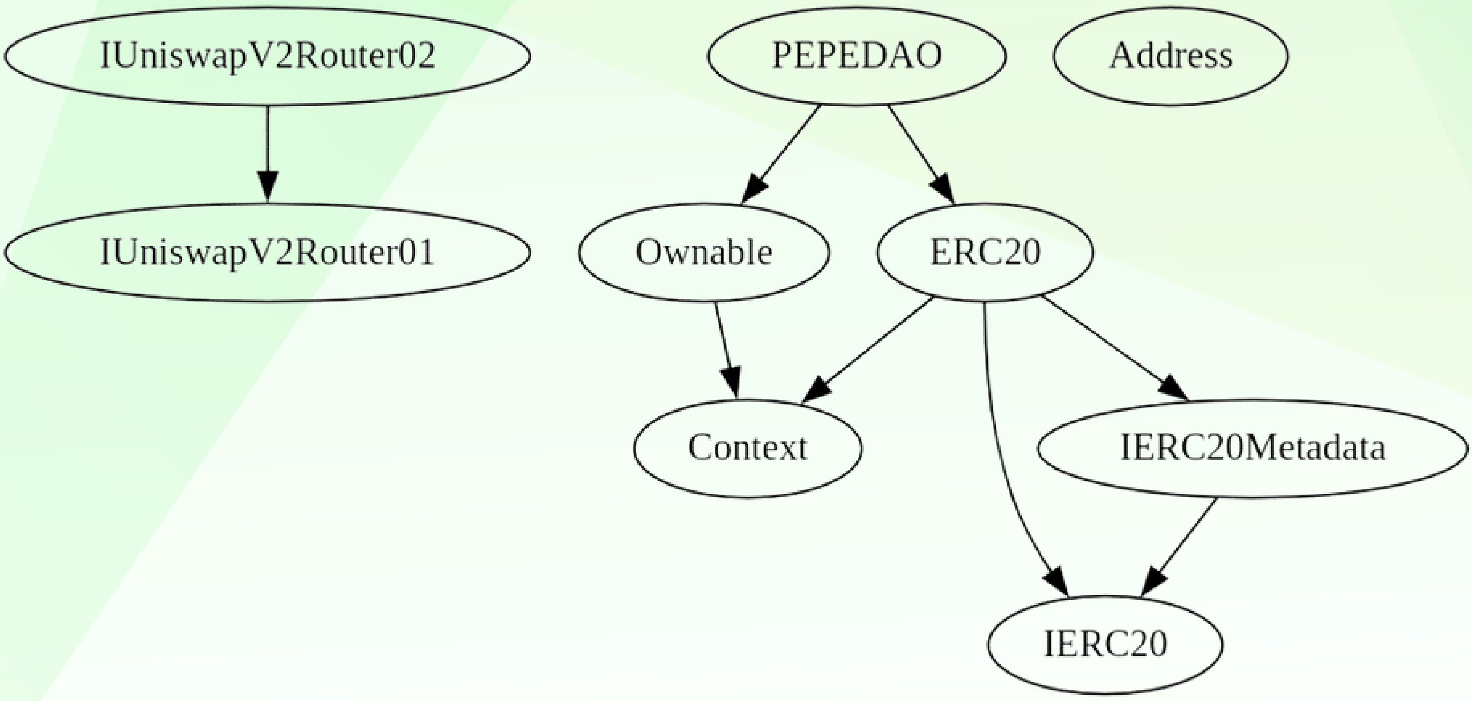
CLASSIFICATION OF RISK

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	0
Low-Risk	0
Gas Optimization / Suggestions	1

INHERITANCE TREE



POINTS TO NOTE

- Owner is not able to set set buy/sell/transfer tax (0% always)
- Owner is not able to set a max buy/transfer/wallet/sell 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 trades for holders to be able to trade**

CONTRACT ASSESMENT

```
|  Contract  |        Type        |       Bases      |                  |                 |
|:----------:|:------------------:|:----------------:|:----------------:|:---------------:|
|     └      |  **Function Name**  |  **Visibility**  |  **Mutability**  |  **Modifiers**  |
| **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 ❗  |
| └ | swap | External ❗  | 🛑  |NO ❗  |
| └ | skim | External ❗  | 🛑  |NO ❗  |
| └ | sync | External ❗  | 🛑  |NO ❗  |
| └ | initialize | External ❗  | 🛑  |NO ❗  |
||||||
```

CONTRACT ASSESMENT

```
| **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 ❗ |
||||||
| **Address** | Library |  |||
```

CONTRACT ASSESMENT

```
| └ | isContract | Internal 🔒 |  | |
| └ | sendValue | Internal 🔒 | 🛑 | |
| └ | functionCall | Internal 🔒 | 🛑 | |
| └ | functionCall | Internal 🔒 | 🛑 | |
| └ | functionCallWithValue | Internal 🔒 | 🛑 | |
| └ | functionCallWithValue | Internal 🔒 | 🛑 | |
| └ | functionStaticCall | Internal 🔒 |  | |
| └ | functionStaticCall | Internal 🔒 |  | |
| └ | functionDelegateCall | Internal 🔒 | 🛑 | |
| └ | functionDelegateCall | Internal 🔒 | 🛑 | |
| └ | verifyCallResultFromTarget | Internal 🔒 |  | |
| └ | verifyCallResult | Internal 🔒 |  | |
| └ | _revert | Private 🔐 |  | |
||||||
| **Context** | Implementation |  |||
| └ | _msgSender | Internal 🔒 |  | |
| └ | _msgData | Internal 🔒 |  | |
||||||
| **Ownable** | Implementation | Context |||
| └ | <Constructor> | Public ❗️ | 🛑 |NO ❗️ |
| └ | owner | Public ❗️ |  |NO ❗️ |
| └ | renounceOwnership | Public ❗️ | 🛑 | onlyOwner |
| └ | transferOwnership | Public ❗️ | 🛑 | onlyOwner |
||||||
| **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 🔒 | 🛑 | |
| └ | _beforeTokenTransfer | Internal 🔒 | 🛑 | |
```

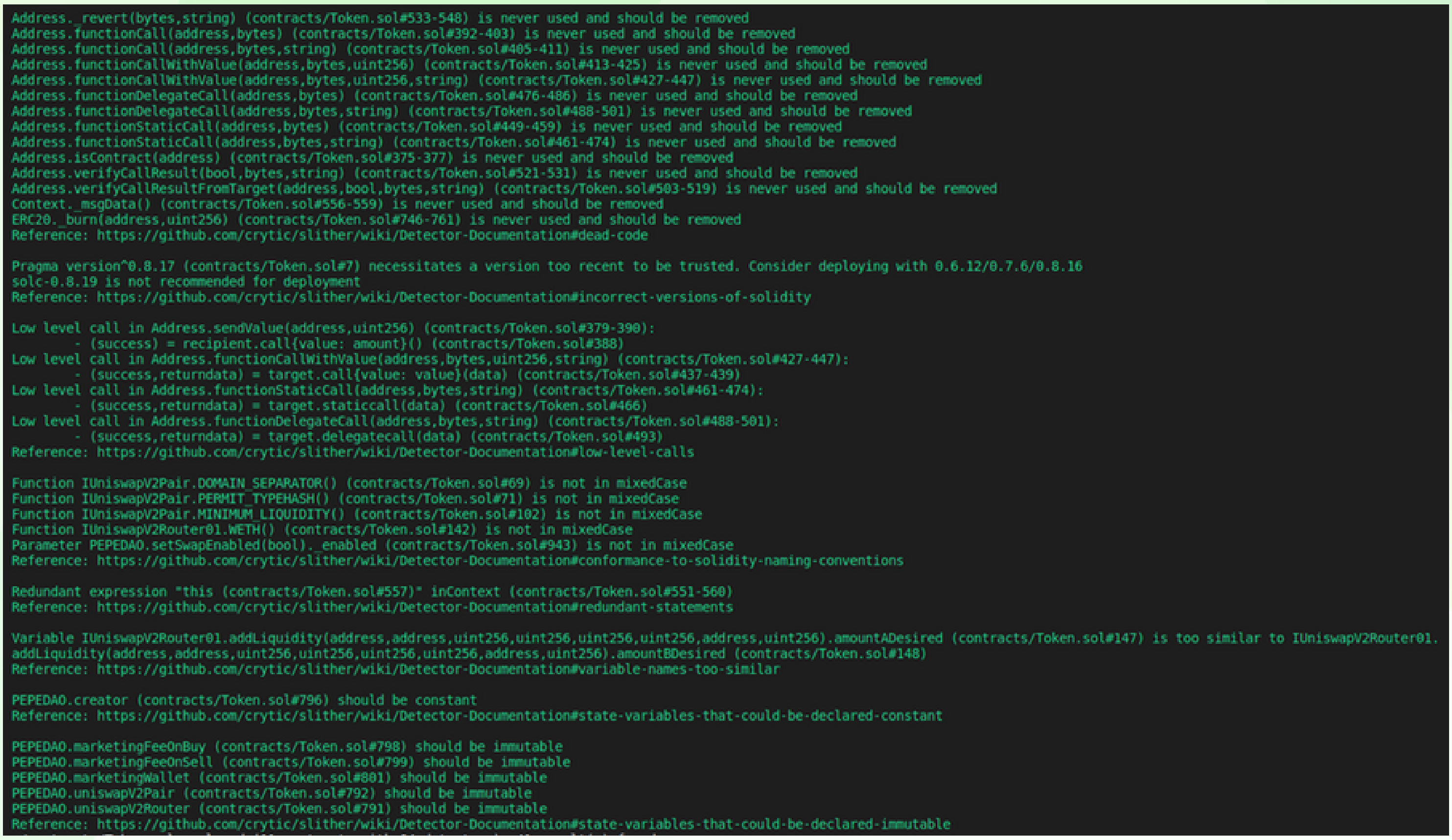
CONTRACT ASSESMENT

└	_afterTokenTransfer	Internal 🔒	🛑	
PEPEDAO	Implementation	ERC20, Ownable		
└	<Constructor>	Public ❗️	🛑	ERC20
└	<Receive Ether>	External ❗️	💵	NO❗️
└	claimStuckTokens	External ❗️	🛑	onlyOwner
└	excludeFromFees	External ❗️	🛑	onlyOwner
└	isExcludedFromFees	Public ❗️		NO❗️
└	enableTrading	External ❗️	🛑	onlyOwner
└	_transfer	Internal 🔒	🛑	
└	setSwapEnabled	External ❗️	🛑	onlyOwner
└	setSwapTokensAtAmount	External ❗️	🛑	onlyOwner
└	swapAndSendMarketing	Private 🔐	🛑	

Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

STATIC ANALYSIS



```
Address._revert(bytes,string) (contracts/Token.sol#533-548) is never used and should be removed
Address.functionCall(address,bytes) (contracts/Token.sol#392-403) is never used and should be removed
Address.functionCall(address,bytes,string) (contracts/Token.sol#405-411) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (contracts/Token.sol#413-425) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (contracts/Token.sol#427-447) is never used and should be removed
Address.functionDelegateCall(address,bytes) (contracts/Token.sol#476-486) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (contracts/Token.sol#488-501) is never used and should be removed
Address.functionStaticCall(address,bytes) (contracts/Token.sol#449-459) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (contracts/Token.sol#461-474) is never used and should be removed
Address.isContract(address) (contracts/Token.sol#375-377) is never used and should be removed
Address.verifyCallResult(bool,bytes,string) (contracts/Token.sol#521-531) is never used and should be removed
Address.verifyCallResultFromTarget(address,bool,bytes,string) (contracts/Token.sol#503-519) is never used and should be removed
Context._msgData() (contracts/Token.sol#556-559) is never used and should be removed
ERC20._burn(address,uint256) (contracts/Token.sol#746-761) 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#379-390):
        - (success) = recipient.call{value: amount}() (contracts/Token.sol#388)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (contracts/Token.sol#427-447):
        - (success,returndata) = target.call{value: value}(data) (contracts/Token.sol#437-439)
Low level call in Address.functionStaticCall(address,bytes,string) (contracts/Token.sol#461-474):
        - (success,returndata) = target.staticcall(data) (contracts/Token.sol#466)
Low level call in Address.functionDelegateCall(address,bytes,string) (contracts/Token.sol#488-501):
        - (success,returndata) = target.delegatecall(data) (contracts/Token.sol#493)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function IUniswapV2Pair.DOMAIN_SEPARATOR() (contracts/Token.sol#69) is not in mixedCase
Function IUniswapV2Pair.PERMIT_TYPEHASH() (contracts/Token.sol#71) is not in mixedCase
Function IUniswapV2Pair.MINIMUM_LIQUIDITY() (contracts/Token.sol#102) is not in mixedCase
Function IUniswapV2Router01.WETH() (contracts/Token.sol#142) is not in mixedCase
Parameter PEPEDAO.setSwapEnabled(bool)._enabled (contracts/Token.sol#943) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (contracts/Token.sol#557)" inContext (contracts/Token.sol#551-560)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

Variable IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountADesired (contracts/Token.sol#147) is too similar to IUniswapV2Router01.
addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountBDesired (contracts/Token.sol#148)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar

PEPEDAO.creator (contracts/Token.sol#796) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

PEPEDAO.marketingFeeOnBuy (contracts/Token.sol#798) should be immutable
PEPEDAO.marketingFeeOnSell (contracts/Token.sol#799) should be immutable
PEPEDAO.marketingWallet (contracts/Token.sol#801) should be immutable
PEPEDAO.uniswapV2Pair (contracts/Token.sol#792) should be immutable
PEPEDAO.uniswapV2Router (contracts/Token.sol#791) 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 issues found

FUNCTIONAL TESTING

1- Adding liquidity (passed):
https://testnet.bscscan.com/tx/0x80308493117cc47706ebfbb5115bc517a2e8ad15e4f1578db69273936de7c271

2- Buying when excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0xe985f449839e2f0aa8cd21a60046917498b1fba6cffb56aa4293ed51adf064b6

3- Selling when excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0x6f756fc71c75c823b81a410f5eb84057ffeae7e088e0723cf5e04c6325a5a721

4- Transferring when excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0x04283601eba8b5497488ff2332ac4028573ee9bb79a91a56b847ce06b471eae3

5- Buying when not excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0x66c634c5f0dadbf6f2be89eeb9de487c92d11d095887604a2684bf7ba4fbd96f

6- Selling when not excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0xce6fb32d7e297cb0e48f55bcf7babce7e917931fc60ebbe0bcf49fa3e8415631

7- Transferring when not excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0x1115b61256aa2c94fe0fa4e44d8fa0e5cb0276afbe1e0321f234a265de844d3e

MANUAL TESTING

Centralization - Trades must be enabled

Severity: High

function: enableTrading

Status: Not Resolved

Overview:

The smart contract owner must enable trades for holders. If trading remain disabled, no one would be able to buy/sell/transfer tokens.

```
function enableTrading() external onlyOwner {
require(!tradingEnabled, "Trading already enabled.");
tradingEnabled = true;
swapEnabled = true;
   }
```

Suggestion

To mitigate this centralization issue, we propose the following options:

1.Renounce Ownership: Consider relinquishing control of the smart contract by renouncing ownership. This would remove the ability for a single entity to manipulate the router, reducing centralization risks.

2.Multi-signature Wallet: Transfer ownership to a multi-signature wallet. This would require multiple approvals for any changes to the mainRouter, adding an additional layer of security and reducing the centralization risk.

3.Transfer ownership to a trusted and valid 3rd party in order to guarantee enabling of the trades

MANUAL TESTING

Informational - Redundant code

Status: Not Resolved

Overview:

Marketing buy and sell tax is zero and can not be changed later, this means some features in the contract (like internal swap) are not needed until there are no tokens in the contract.

Suggestion:

delete sections of the code that are related to tax (like internal swap, etc…) in order to reduce overall gas usage

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