auditace.tech

AuditAce

FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

CPEPE

DATED : 17 MAY 23'

AUDIT SUMMARY

Project name – CPEPE

Date: 17 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	0	2	1	2
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.

Testnet version:

The tests were performed using the contract deployed on the BSC Testnet, which can be found at the following address: https://testnet.bscscan.com/token/0x55812462524fdebe2172c90629a928c174942383

Payment Transaction :

0x9760c2426a079ac2d483fa6765ab187f90984cce0d228a6c8b88c263fd84dbcc

Token Information

Name : CPEPE

Symbol : CZPEPE

Decimals: 18

Network: BSC

Token Type: BEP20

Token Address:
0x6c895882Ce0abdbb4e77d6bD24ED7db6D98F6F8a

Owner:
0x97681c12dD3A7889cEC0786Bdcb57fA2CeA84D30 **(at time of writing the audit)**

Deployer:0x4DC331D1dfc2FDDD739782A18A3697d1562Ba3F3

Token Information

Fees:

Buy Fees: 0-25%

Sell Fees: 0-25%

Transfer Fees: 0-5%

Fees Privilige: Owner

Ownership : Owned

Minting: None

Max Tx Amount/ Max Wallet Amount: 0.1%-100% supply

Blacklist: No

Other Priviliges: Changing fees - changing limits - enabling trades

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.

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

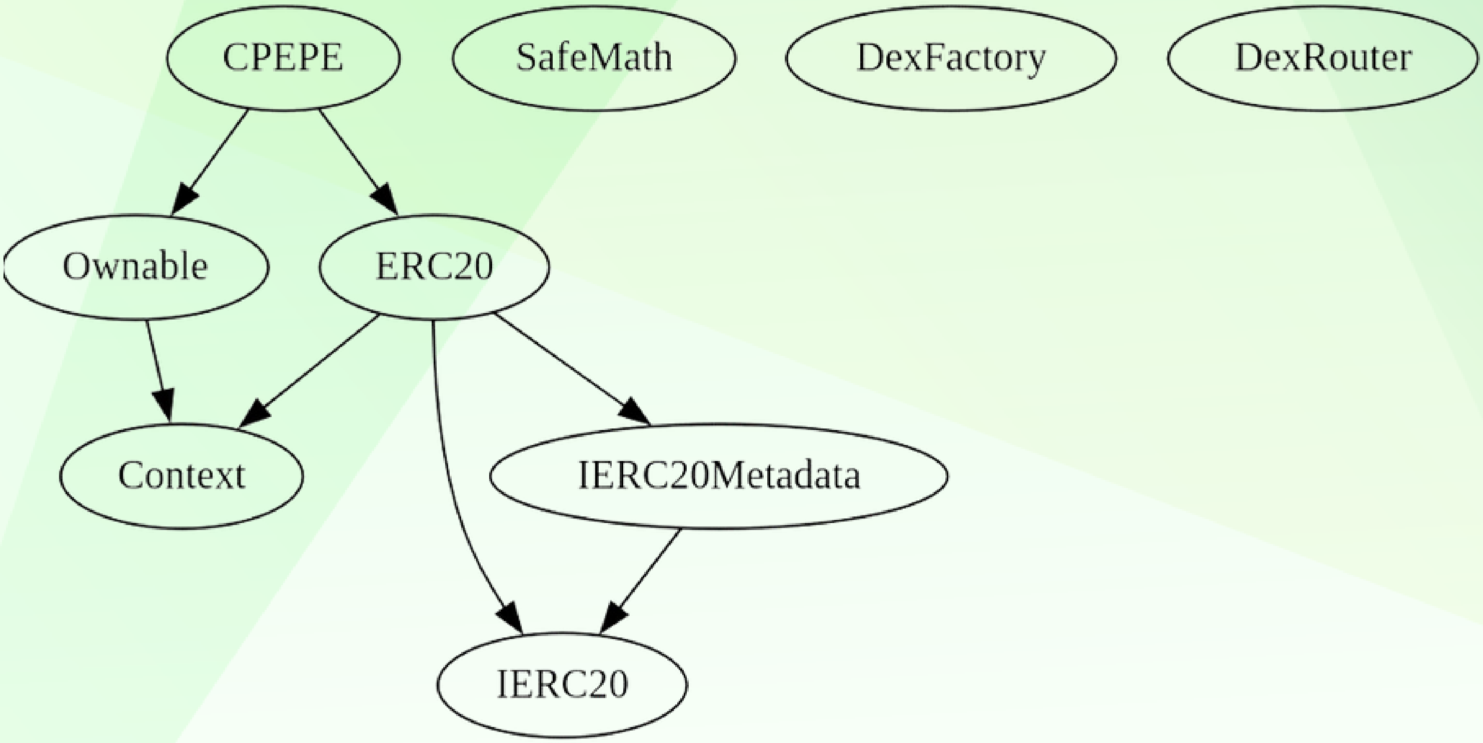
CLASSIFICATION OF RISK

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	0
◆ Medium-Risk	2
◆ Low-Risk	1
◆ Gas Optimization / Suggestions	2

INHERITANCE TREE



POINTS TO NOTE

- Owner is not able to change buy/sell fees over 25% (buy + sell <= 25%)
- Owner is not able to change transfer fees over 5%
- Owner is not able to blacklist an arbitrary address.
- Owner is not able to disable trades
- Owner is able to set max buy/sell/transfer/wallet amount withing a range of 0.1% – 100% of supply
- Owner is not able to mint new tokens
- Owner must enable trades manually for investors

CONTRACT ASSESMENT

```
| Contract |        Type        |       Bases      |                  |                 |
|:----------:|:-------------------:|:----------------:|:----------------:|:---------------:|
|     └      |  **Function Name**  |  **Visibility**  |  **Mutability**  |  **Modifiers**  |
||||||
| **Context** | Implementation |  |||
| └ | _msgSender | Internal 🔒 |   | |
| └ | _msgData | Internal 🔒 |   | |
||||||
| **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❗️ |
||||||
| **SafeMath** | Library |  |||
| └ | tryAdd | Internal 🔒 |   | |
| └ | trySub | Internal 🔒 |   | |
| └ | tryMul | Internal 🔒 |   | |
| └ | tryDiv | Internal 🔒 |   | |
| └ | tryMod | Internal 🔒 |   | |
| └ | add | Internal 🔒 |   | |
| └ | sub | Internal 🔒 |   | |
| └ | mul | Internal 🔒 |   | |
| └ | div | Internal 🔒 |   | |
| └ | mod | Internal 🔒 |   | |
| └ | sub | Internal 🔒 |   | |
| └ | div | Internal 🔒 |   | |
| └ | mod | Internal 🔒 |   | |
||||||
| **Ownable** | Implementation | Context |||
| └ | <Constructor> | Public ❗️ | 🛑  |NO❗️ |
| └ | owner | Public ❗️ |   |NO❗️ |
| └ | _checkOwner | Internal 🔒 |   | |
| └ | renounceOwnership | Public ❗️ | 🛑  | onlyOwner |
| └ | transferOwnership | Public ❗️ | 🛑  | onlyOwner |
| └ | _transferOwnership | Internal 🔒 | 🛑  | |
```

CONTRACT ASSESMENT

```
||||||
| **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 🔒 | 🛑 | |
| └ | _spendAllowance | Internal 🔒 | 🛑 | |
| └ | _beforeTokenTransfer | Internal 🔒 | 🛑 | |
| └ | _afterTokenTransfer | Internal 🔒 | 🛑 | |
||||||
| **DexFactory** | Interface | |||
| └ | createPair | External ❗️ | 🛑 |NO❗️ |
||||||
| **DexRouter** | Interface | |||
| └ | factory | External ❗️ |   |NO❗️ |
| └ | WETH | External ❗️ |   |NO❗️ |
| └ | addLiquidityETH | External ❗️ | 💵 |NO❗️ |
| └ | swapExactTokensForETHSupportingFeeOnTransferTokens | External ❗️ | 🛑 |NO❗️ |
||||||
| **CPEPE** | Implementation | ERC20, Ownable |||
| └ | <Constructor> | Public ❗️ | 🛑 | ERC20 |
| └ | enableTrading | External ❗️ | 🛑 | onlyOwner |
| └ | setmarketingWallet | External ❗️ | 🛑 | onlyOwner |
| └ | setreliquidityWallet | External ❗️ | 🛑 | onlyOwner |
| └ | setMaxBuy | External ❗️ | 🛑 | onlyOwner |
| └ | setMaxSell | External ❗️ | 🛑 | onlyOwner |
| └ | setMaxTx | External ❗️ | 🛑 | onlyOwner |
| └ | setMaxWallet | External ❗️ | 🛑 | onlyOwner |
| └ | setBuyTaxes | External ❗️ | 🛑 | onlyOwner |
| └ | setSellTaxes | External ❗️ | 🛑 | onlyOwner |
```

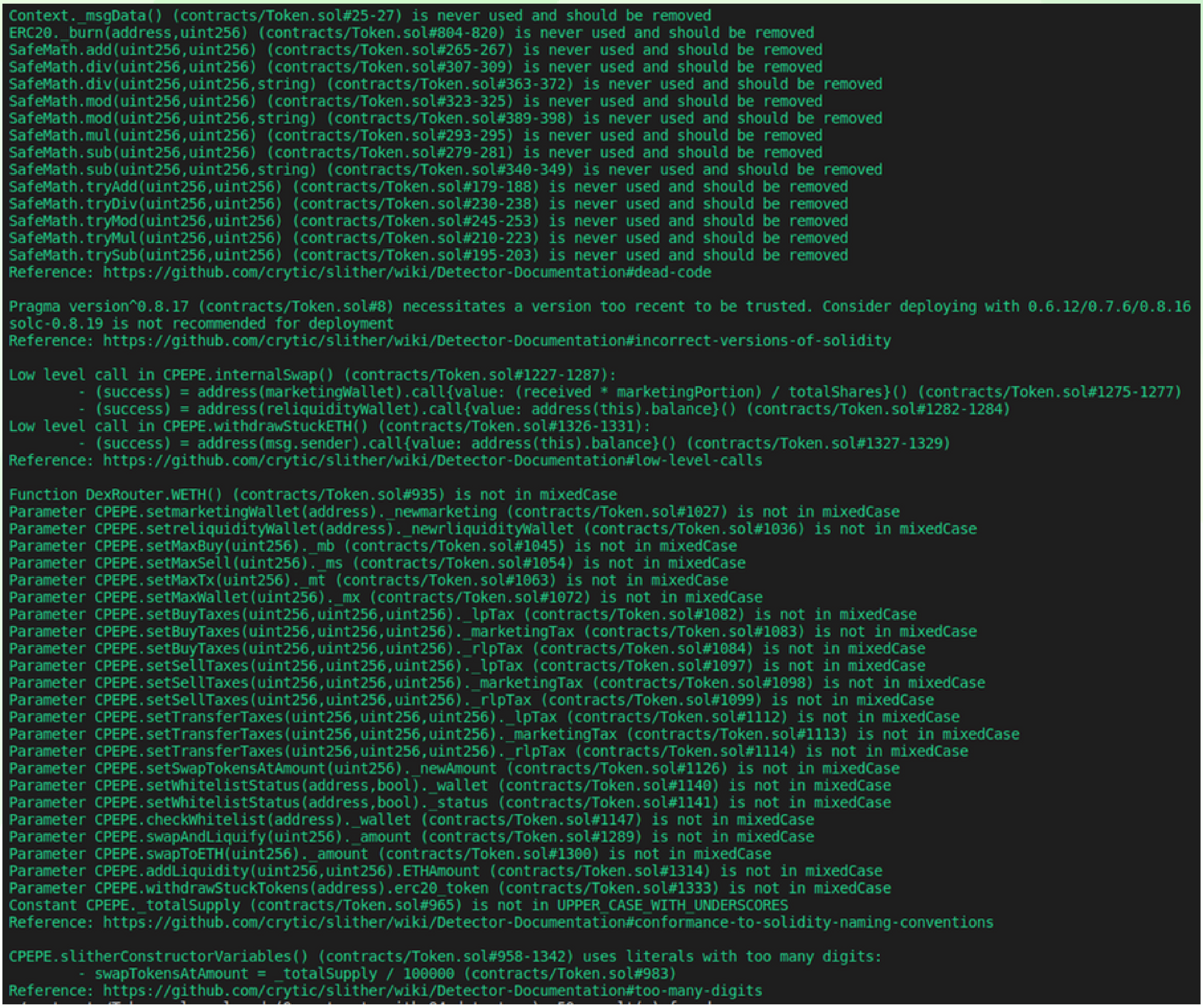
CONTRACT ASSESMENT

```
| └ | setTransferTaxes | External ❗ | 🛑 | onlyOwner |
| └ | setSwapTokensAtAmount | External ❗ | 🛑 | onlyOwner |
| └ | toggleSwapping | External ❗ | 🛑 | onlyOwner |
| └ | setWhitelistStatus | External ❗ | 🛑 | onlyOwner |
| └ | checkWhitelist | External ❗ | |NO ❗ |
| └ | _takeTax | Internal 🔒 | 🛑 | |
| └ | _transfer | Internal 🔒 | 🛑 | |
| └ | internalSwap | Internal 🔒 | 🛑 | |
| └ | swapAndLiquify | Internal 🔒 | 🛑 | |
| └ | swapToETH | Internal 🔒 | 🛑 | |
| └ | addLiquidity | Private 🔐 | 🛑 | |
| └ | withdrawStuckETH | External ❗ | 🛑 | onlyOwner |
| └ | withdrawStuckTokens | External ❗ | 🛑 | onlyOwner |
| └ | <Receive Ether> | External ❗ | 💵 |NO ❗ |
```

Legend

```
| Symbol | Meaning |
|:--------:|-----------|
| 🛑 | Function can modify state |
| 💵 | Function is payable |
```

STATIC ANALYSIS



```
Context._msgData() (contracts/Token.sol#25-27) is never used and should be removed
ERC20._burn(address,uint256) (contracts/Token.sol#804-820) is never used and should be removed
SafeMath.add(uint256,uint256) (contracts/Token.sol#265-267) is never used and should be removed
SafeMath.div(uint256,uint256) (contracts/Token.sol#307-309) is never used and should be removed
SafeMath.div(uint256,uint256,string) (contracts/Token.sol#363-372) is never used and should be removed
SafeMath.mod(uint256,uint256) (contracts/Token.sol#323-325) is never used and should be removed
SafeMath.mod(uint256,uint256,string) (contracts/Token.sol#389-398) is never used and should be removed
SafeMath.mul(uint256,uint256) (contracts/Token.sol#293-295) is never used and should be removed
SafeMath.sub(uint256,uint256) (contracts/Token.sol#279-281) is never used and should be removed
SafeMath.sub(uint256,uint256,string) (contracts/Token.sol#340-349) is never used and should be removed
SafeMath.tryAdd(uint256,uint256) (contracts/Token.sol#179-188) is never used and should be removed
SafeMath.tryDiv(uint256,uint256) (contracts/Token.sol#230-238) is never used and should be removed
SafeMath.tryMod(uint256,uint256) (contracts/Token.sol#245-253) is never used and should be removed
SafeMath.tryMul(uint256,uint256) (contracts/Token.sol#210-223) is never used and should be removed
SafeMath.trySub(uint256,uint256) (contracts/Token.sol#195-203) 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#8) 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 CPEPE.internalSwap() (contracts/Token.sol#1227-1287):
        - (success) = address(marketingWallet).call{value: (received * marketingPortion) / totalShares}() (contracts/Token.sol#1275-1277)
        - (success) = address(reliquidityWallet).call{value: address(this).balance}() (contracts/Token.sol#1282-1284)
Low level call in CPEPE.withdrawStuckETH() (contracts/Token.sol#1326-1331):
        - (success) = address(msg.sender).call{value: address(this).balance}() (contracts/Token.sol#1327-1329)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function DexRouter.WETH() (contracts/Token.sol#935) is not in mixedCase
Parameter CPEPE.setmarketingWallet(address)._newmarketing (contracts/Token.sol#1027) is not in mixedCase
Parameter CPEPE.setreliquidityWallet(address)._newrliquidityWallet (contracts/Token.sol#1036) is not in mixedCase
Parameter CPEPE.setMaxBuy(uint256)._mb (contracts/Token.sol#1045) is not in mixedCase
Parameter CPEPE.setMaxSell(uint256)._ms (contracts/Token.sol#1054) is not in mixedCase
Parameter CPEPE.setMaxTx(uint256)._mt (contracts/Token.sol#1063) is not in mixedCase
Parameter CPEPE.setMaxWallet(uint256)._mx (contracts/Token.sol#1072) is not in mixedCase
Parameter CPEPE.setBuyTaxes(uint256,uint256,uint256)._lpTax (contracts/Token.sol#1082) is not in mixedCase
Parameter CPEPE.setBuyTaxes(uint256,uint256,uint256)._marketingTax (contracts/Token.sol#1083) is not in mixedCase
Parameter CPEPE.setBuyTaxes(uint256,uint256,uint256)._rlpTax (contracts/Token.sol#1084) is not in mixedCase
Parameter CPEPE.setSellTaxes(uint256,uint256,uint256)._lpTax (contracts/Token.sol#1097) is not in mixedCase
Parameter CPEPE.setSellTaxes(uint256,uint256,uint256)._marketingTax (contracts/Token.sol#1098) is not in mixedCase
Parameter CPEPE.setSellTaxes(uint256,uint256,uint256)._rlpTax (contracts/Token.sol#1099) is not in mixedCase
Parameter CPEPE.setTransferTaxes(uint256,uint256,uint256)._lpTax (contracts/Token.sol#1112) is not in mixedCase
Parameter CPEPE.setTransferTaxes(uint256,uint256,uint256)._marketingTax (contracts/Token.sol#1113) is not in mixedCase
Parameter CPEPE.setTransferTaxes(uint256,uint256,uint256)._rlpTax (contracts/Token.sol#1114) is not in mixedCase
Parameter CPEPE.setSwapTokensAtAmount(uint256)._newAmount (contracts/Token.sol#1126) is not in mixedCase
Parameter CPEPE.setWhitelistStatus(address,bool)._wallet (contracts/Token.sol#1140) is not in mixedCase
Parameter CPEPE.setWhitelistStatus(address,bool)._status (contracts/Token.sol#1141) is not in mixedCase
Parameter CPEPE.checkWhitelist(address)._wallet (contracts/Token.sol#1147) is not in mixedCase
Parameter CPEPE.swapAndLiquify(uint256)._amount (contracts/Token.sol#1289) is not in mixedCase
Parameter CPEPE.swapToETH(uint256)._amount (contracts/Token.sol#1300) is not in mixedCase
Parameter CPEPE.addLiquidity(uint256,uint256).ETHAmount (contracts/Token.sol#1314) is not in mixedCase
Parameter CPEPE.withdrawStuckTokens(address).erc20_token (contracts/Token.sol#1333) is not in mixedCase
Constant CPEPE._totalSupply (contracts/Token.sol#965) is not in UPPER_CASE_WITH_UNDERSCORES
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

CPEPE.slitherConstructorVariables() (contracts/Token.sol#958-1342) uses literals with too many digits:
        - swapTokensAtAmount = _totalSupply / 100000 (contracts/Token.sol#983)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
```

Static Analysis

an static analysis of the code were performed using slither. No issues were found

FUNCTIONAL TESTING

Router (PCS V2):
0xD99D1c33F9fC3444f8101754aBC46c52416550D1

1- Adding liquidity (passed):
https://testnet.bscscan.com/tx/0x4c745fbcddc2a6d6039f7e5cd846d4560b50731a8999dacbf4cd291e0605f16a

2- Buying when excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0xccfec11f6e45997b6352f35a34c72ecac9dec9b1fe2f78171535512aca3e95fb

3- Selling when excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0xda8539d483623871e5a26def0d08b4ae31e522ae825ec684f7eca6642855bf70

4- Transferring when excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0x425f0ec8022e5e36053c2adf183157638393f178d18922ee0258a2daa264d40a

5- Buying when not excluded from fees (0-25% tax) (passed):
https://testnet.bscscan.com/tx/0x292134b14c8e5cc3434d854f9d577bdb5d7c344dd9f69d1ac3c41ed2d42fddcc

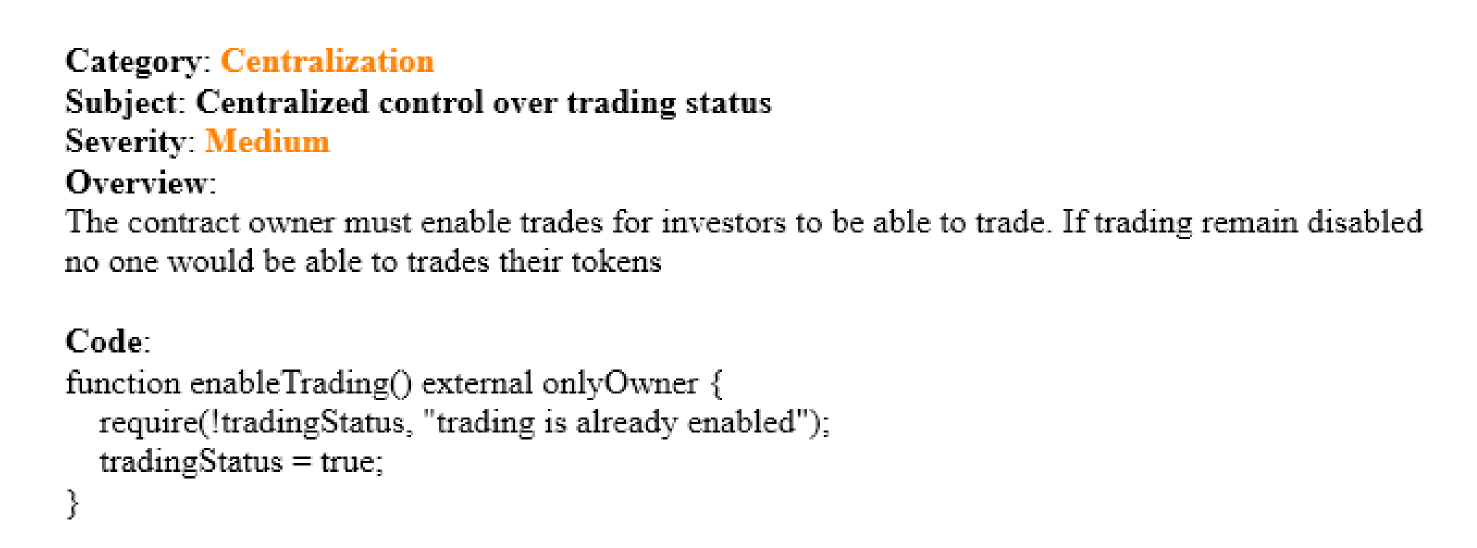
6- Selling when not excluded from fees (0-25% tax) (passed):
https://testnet.bscscan.com/tx/0x4582c9807ff4378826b2eedfaf24ed9f1ef4e8a007394d5366d8d636b2e2feab

7- Transferring when not excluded from fees (0-5% tax) (passed):
https://testnet.bscscan.com/tx/0x23d685f8d6ba7be56715a2f7e59bce9980228c1cd10aa88ea5077413f9c060e5

FUNCTIONAL TESTING

7- Internal swap (auto-liquidity and bnb fees) **(passed):**
https://testnet.bscscan.com/tx/0x4582c9807ff4378826b2eedfaf24ed9f1ef4e8a007394d5366d8d636b2e2feab

FUNCTIONAL TESTING

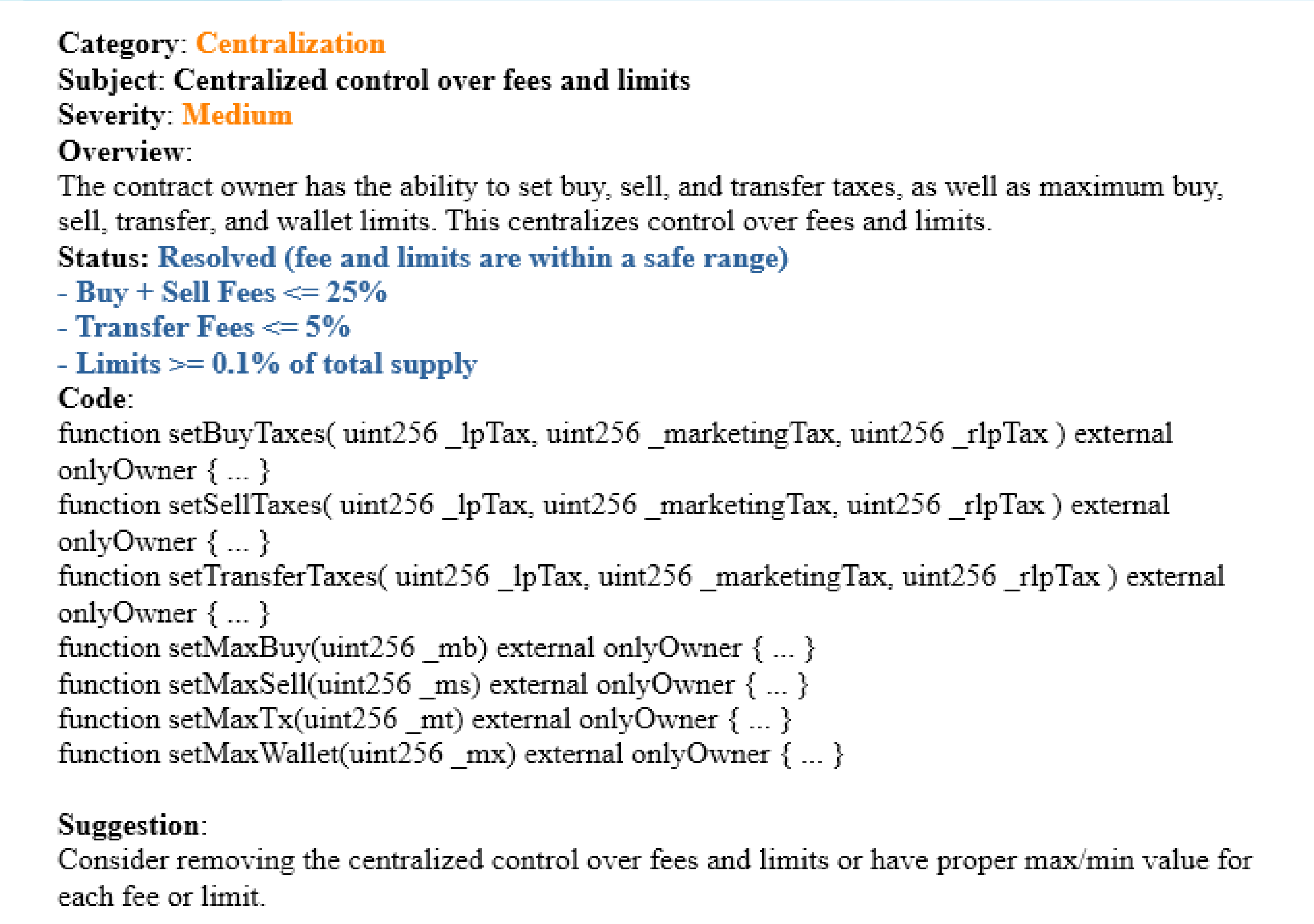


Category: Centralization
Subject: **Centralized control over trading status**
Severity: Medium
Overview:
The contract owner must enable trades for investors to be able to trade. If trading remain disabled no one would be able to trades their tokens

Code:

```
function enableTrading() external onlyOwner {
   require(!tradingStatus, "trading is already enabled");
   tradingStatus = true;
}
```

FUNCTIONAL TESTING

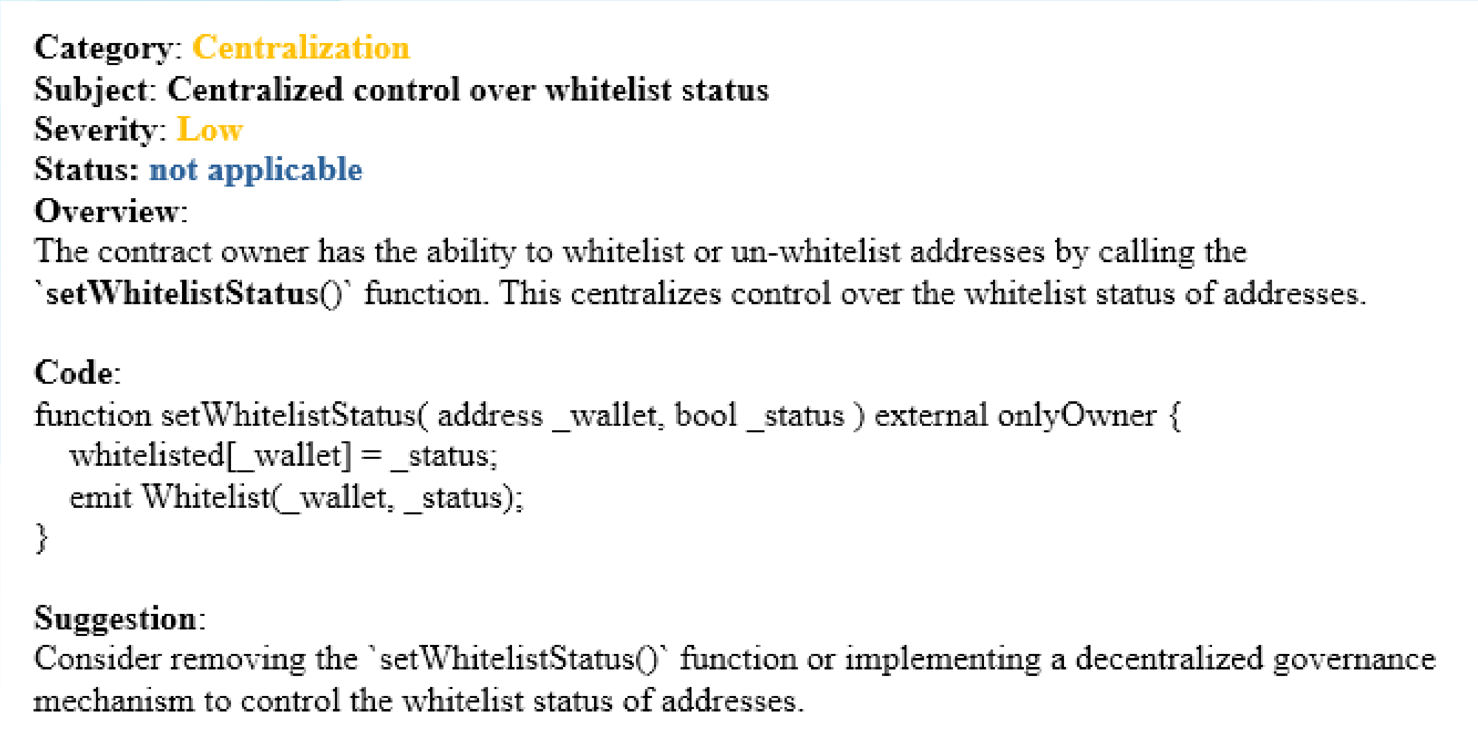


Category: Centralization
Subject: **Centralized control over fees and limits**
Severity: Medium
Overview:
The contract owner has the ability to set buy, sell, and transfer taxes, as well as maximum buy, sell, transfer, and wallet limits. This centralizes control over fees and limits.
Status: Resolved (fee and limits are within a safe range)
- Buy + Sell Fees <= 25%
- Transfer Fees <= 5%
- Limits >= 0.1% of total supply
Code:

```
function setBuyTaxes( uint256 _lpTax, uint256 _marketingTax, uint256 _rlpTax ) external onlyOwner { ... }
function setSellTaxes( uint256 _lpTax, uint256 _marketingTax, uint256 _rlpTax ) external onlyOwner { ... }
function setTransferTaxes( uint256 _lpTax, uint256 _marketingTax, uint256 _rlpTax ) external onlyOwner { ... }
function setMaxBuy(uint256 _mb) external onlyOwner { ... }
function setMaxSell(uint256 _ms) external onlyOwner { ... }
function setMaxTx(uint256 _mt) external onlyOwner { ... }
function setMaxWallet(uint256 _mx) external onlyOwner { ... }
```

Suggestion:
Consider removing the centralized control over fees and limits or have proper max/min value for each fee or limit.

FUNCTIONAL TESTING



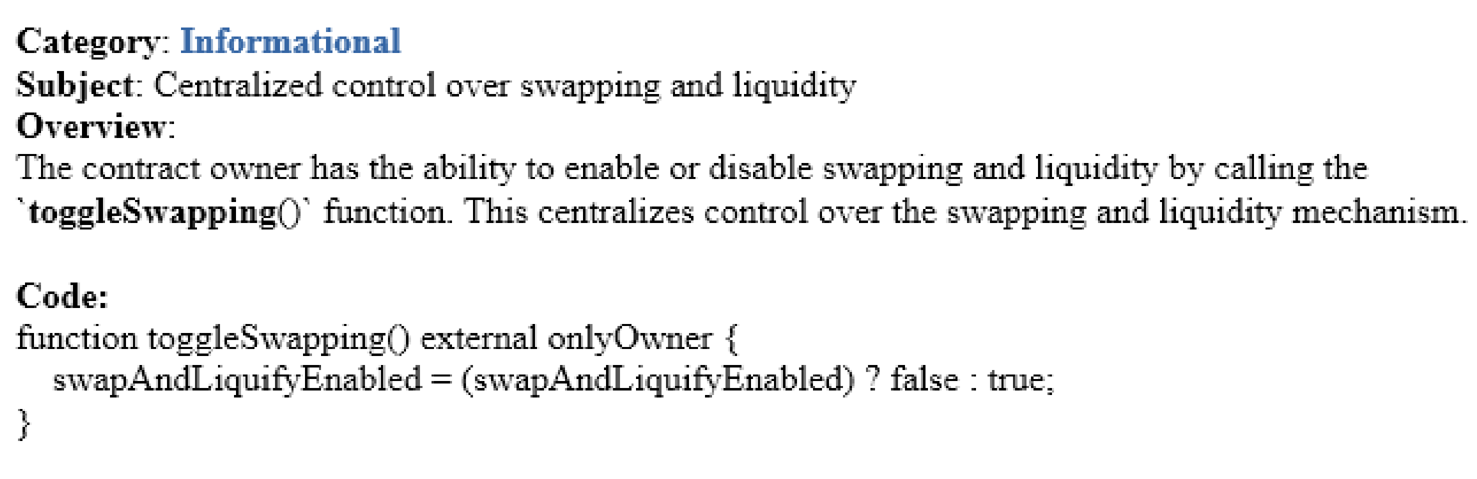
Category: Centralization
Subject: **Centralized control over whitelist status**
Severity: Low
Status: not applicable
Overview:
The contract owner has the ability to whitelist or un-whitelist addresses by calling the `**setWhitelistStatus()**` function. This centralizes control over the whitelist status of addresses.

Code:

```
function setWhitelistStatus( address _wallet, bool _status ) external onlyOwner {
   whitelisted[_wallet] = _status;
   emit Whitelist(_wallet, _status);
}
```

Suggestion:
Consider removing the `setWhitelistStatus()` function or implementing a decentralized governance mechanism to control the whitelist status of addresses.

FUNCTIONAL TESTING

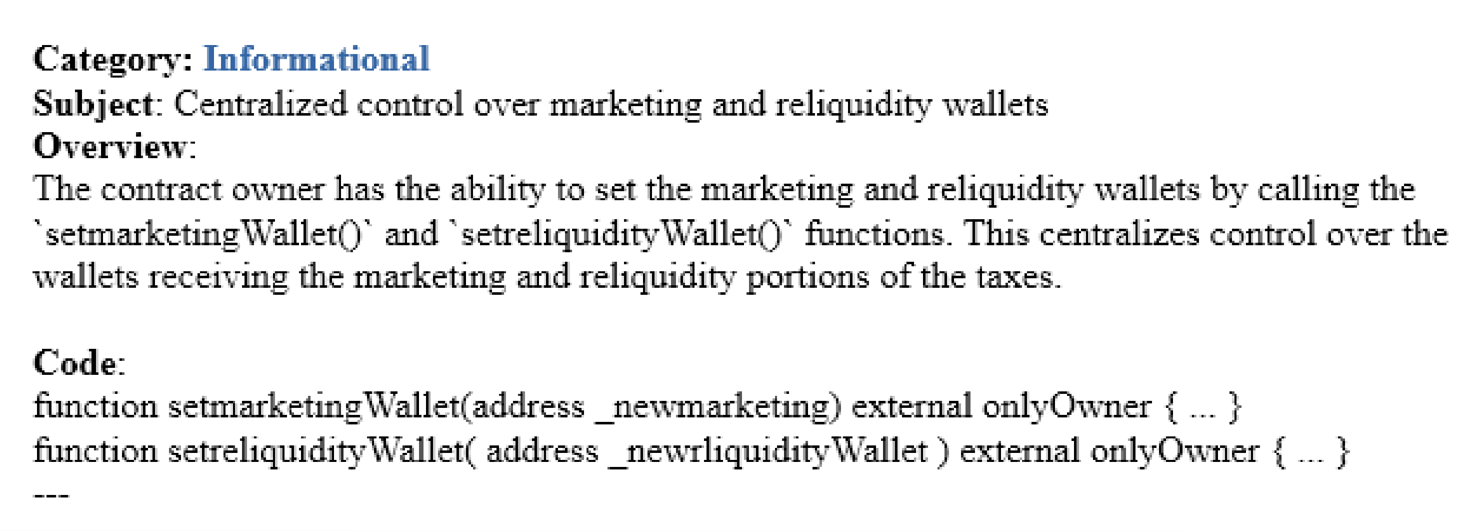


Category: Informational
Subject: Centralized control over swapping and liquidity
Overview:
The contract owner has the ability to enable or disable swapping and liquidity by calling the `**toggleSwapping**()` function. This centralizes control over the swapping and liquidity mechanism.

Code:

```
function toggleSwapping() external onlyOwner {
    swapAndLiquifyEnabled = (swapAndLiquifyEnabled) ? false : true;
}
```

FUNCTIONAL TESTING



Category: Informational
Subject: Centralized control over marketing and reliquidity wallets
Overview:
The contract owner has the ability to set the marketing and reliquidity wallets by calling the `setmarketingWallet()` and `setreliquidityWallet()` functions. This centralizes control over the wallets receiving the marketing and reliquidity portions of the taxes.

Code:

```
function setmarketingWallet(address _newmarketing) external onlyOwner { ... }
function setreliquidityWallet( address _newrliquidityWallet ) external onlyOwner { ... }
---
```

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