auditace.tech

AuditAce

FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

PEPE.sui

DATED : 21 April 23'

AUDIT SUMMARY

Project name – PEPE.sui

Date: 21 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	0	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 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/0x3232f990a3130d527D39db1A20703248c39307b7

Token Information

Token Name : PEPE SUI

Token Symbol: PEPE.sui

Decimals: 9

Token Supply: 420,000,000,000,000

Token Address:
0x8520568339c077ad035E94b660F3258Ee8FD4b45

Checksum:
08be1714d771b34708caee97696fa6cc9c893f37

Owner:
0x42DeC0326A07320B9F97C2e03ACF06Ec4dF67dC3
(at time of writing the audit)

Deployer:
0x42DeC0326A07320B9F97C2e03ACF06Ec4dF67dC3

TOKEN OVERVIEW

Fees:

Buy Fees: up to 12%

Sell Fees: up to 12%

Transfer Fees: 0%

Fees Privilege: Owner

Ownership: Owned

Minting: No mint function

Max Tx Amount/ Max Wallet Amount: No

Blacklist: Yes

Other Privileges: excluding from fees - including in fees - changing fees

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

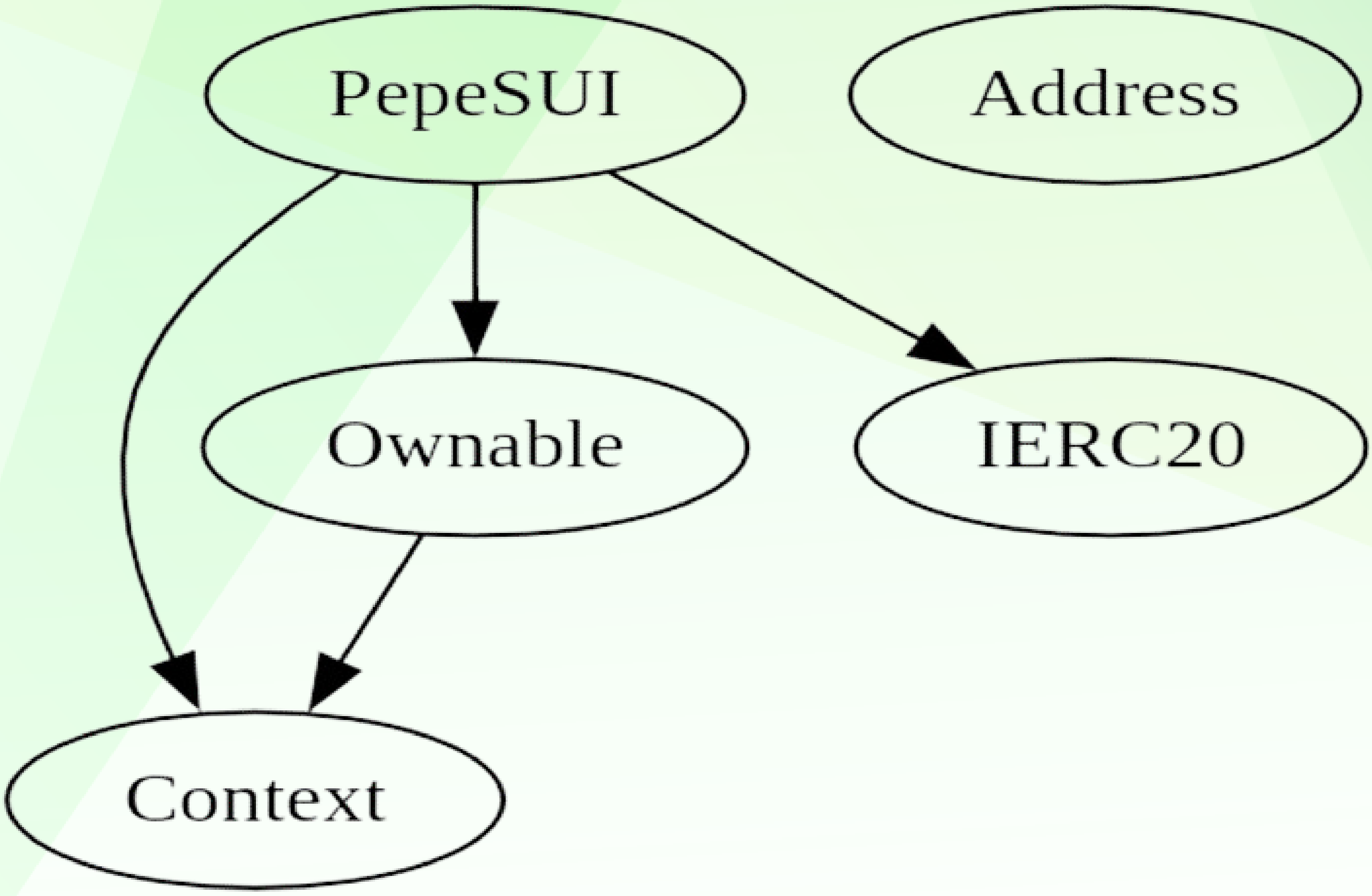
CLASSIFICATION OF RISK

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**	1
◆ **Low-Risk**	0
◆ **Gas Optimization / Suggestions**	0

INHERITANCE TREE



POINTS TO NOTE

- Owner is not able to set buy/sell fees higher than 12%
- 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

CONTRACT ASSESMENT

```
| Contract |        Type        |       Bases      |                  |                 |
|:----------:|:-------------------:|:----------------:|:----------------:|:---------------:|
|     └      |  **Function Name**  |  **Visibility**  |  **Mutability**  |  **Modifiers**  |
||||||
| **Context** | Implementation |  |||
| └ | _msgSender | Internal 🔒 |   | |
| └ | _msgData | Internal 🔒 |   | |
||||||
| **Ownable** | Implementation | Context |||
| └ | <Constructor> | Public ❗️ | 🛑  |NO ❗️ |
| └ | owner | Public ❗️ |   |NO ❗️ |
| └ | renounceOwnership | Public ❗️ | 🛑  | onlyOwner |
| └ | transferOwnership | Public ❗️ | 🛑  | onlyOwner |
||||||
| **IERC20** | Interface |  |||
| └ | totalSupply | External ❗️ |   |NO ❗️ |
| └ | balanceOf | External ❗️ |   |NO ❗️ |
| └ | transfer | External ❗️ | 🛑  |NO ❗️ |
| └ | allowance | External ❗️ |   |NO ❗️ |
| └ | approve | External ❗️ | 🛑  |NO ❗️ |
| └ | transferFrom | External ❗️ | 🛑  |NO ❗️ |
||||||
| **Address** | Library |  |||
| └ | isContract | Internal 🔒 |   | |
| └ | sendValue | Internal 🔒 | 🛑  | |
| └ | functionCall | Internal 🔒 | 🛑  | |
| └ | functionCall | Internal 🔒 | 🛑  | |
| └ | functionCallWithValue | Internal 🔒 | 🛑  | |
| └ | functionCallWithValue | Internal 🔒 | 🛑  | |
| └ | _functionCallWithValue | Private 🔐 | 🛑  | |
||||||
| **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 ❗ |
| └ | burn | External ❗ | 🛑 |NO ❗ |
| └ | 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 ❗ |
```

CONTRACT ASSESMENT

```
| └ | 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 ❗ |
||||||
| **PepeSUI** | Implementation | Context, IERC20, Ownable |||
| └ | <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 ❗ |
| └ | isExcludedFromReward | Public ❗ |  |NO ❗ |
| └ | totalReflectionDistributed | Public ❗ |  |NO ❗ |
| └ | deliver | Public ❗ | 🛑 |NO ❗ |
| └ | reflectionFromToken | Public ❗ |  |NO ❗ |
| └ | tokenFromReflection | Public ❗ |  |NO ❗ |
| └ | excludeFromReward | Public ❗ | 🛑 | onlyOwner |
| └ | includeInReward | External ❗ | 🛑 | onlyOwner |
| └ | <Receive Ether> | External ❗ | 💵 |NO ❗ |
| └ | claimStuckTokens | External ❗ | 🛑 | onlyOwner |
| └ | setStakingAddress | External ❗ | 🛑 | onlyOwner |
| └ | lockToken | Public ❗ | 🛑 |NO ❗ |
| └ | unlockToken | Public ❗ | 🛑 |NO ❗ |
| └ | updateFeeBuy | Public ❗ | 🛑 | onlyOwner |
| └ | updateFeeSell | Public ❗ | 🛑 | onlyOwner |
| └ | _reflectFee | Private 🔐 | 🛑 | |
| └ | _getValues | Private 🔐 |  | |
| └ | _getTValues | Private 🔐 |  | |
| └ | _getRValues | Private 🔐 |  | |
```

└	_getRate	Private 🔐		
└	_getCurrentSupply	Private 🔐		
└	_takeLiquidity	Private 🔐	🛑	
└	_takeMarketing	Private 🔐	🛑	
└	calculateTaxFee	Private 🔐		
└	calculateLiquidityFee	Private 🔐		
└	calculateMarketingFee	Private 🔐		
└	removeAllFee	Private 🔐	🛑	
└	setBuyFee	Private 🔐	🛑	
└	setSellFee	Private 🔐	🛑	
└	isExcludedFromFee	Public ❗		NO ❗
└	_approve	Private 🔐	🛑	
└	_transfer	Private 🔐	🛑	
└	swapAndLiquify	Private 🔐	🛑	
└	swapAndSendMarketing	Private 🔐	🛑	
└	setSwapTokensAtAmount	External ❗	🛑	onlyOwner
└	setSwapEnabled	External ❗	🛑	onlyOwner
└	_tokenTransfer	Private 🔐	🛑	
└	_transferStandard	Private 🔐	🛑	
└	_transferToExcluded	Private 🔐	🛑	
└	_transferFromExcluded	Private 🔐	🛑	
└	_transferBothExcluded	Private 🔐	🛑	
└	excludeFromFees	External ❗	🛑	onlyOwner
└	isContract	Internal 🔒		

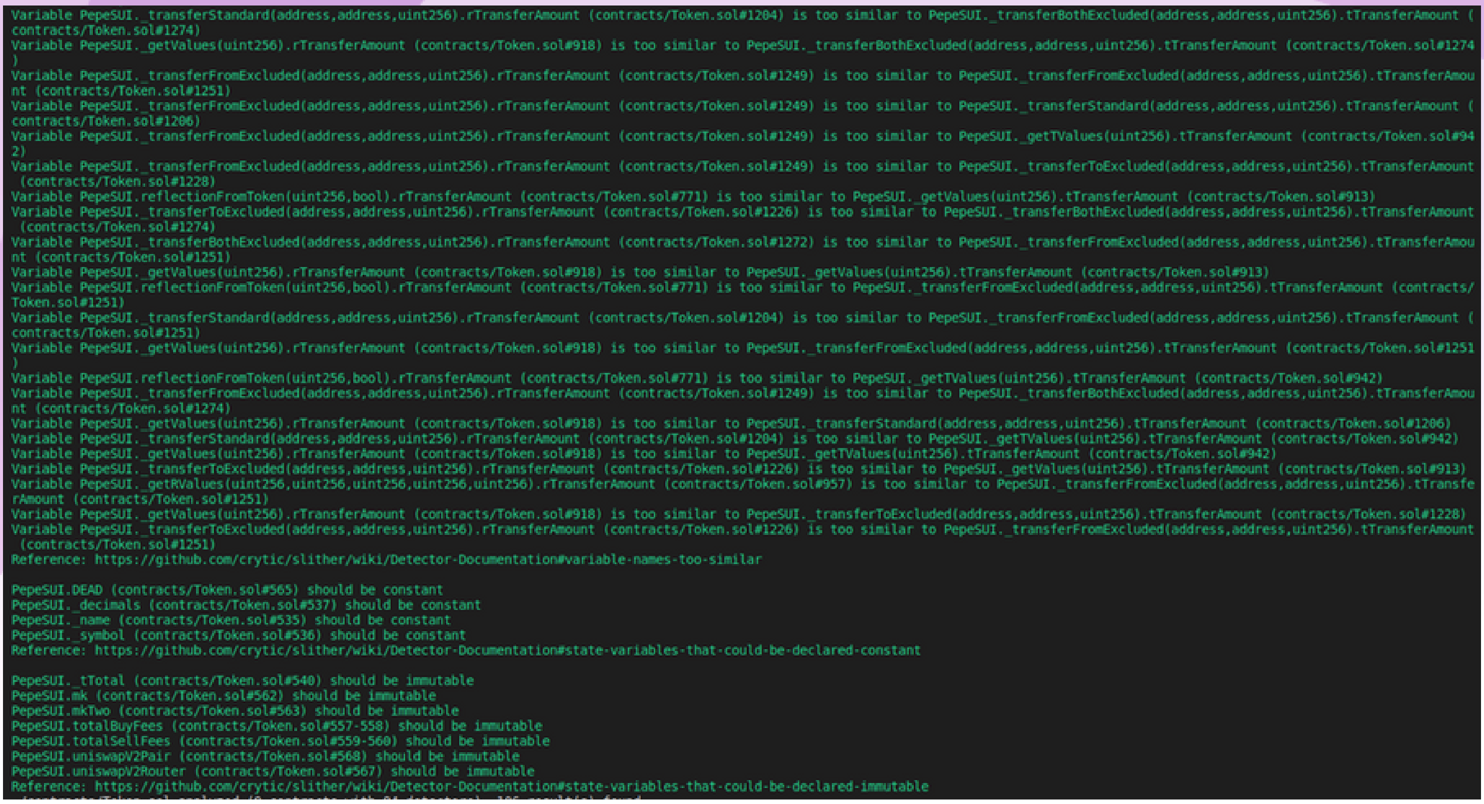
Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

TOKEN DISTRIBUTION

It should be noted that the owner currently holds 100% of the total supply. However, information about the distribution of these tokens is not available, and it is recommended that investors exercise caution when considering this aspect.

STATIC ANALYSIS



```
Variable PepeSUI._transferStandard(address,address,uint256).rTransferAmount (contracts/Token.sol#1204) is too similar to PepeSUI._transferBothExcluded(address,address,uint256).tTransferAmount (
contracts/Token.sol#1274)
Variable PepeSUI._getValues(uint256).rTransferAmount (contracts/Token.sol#918) is too similar to PepeSUI._transferBothExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1274
)
Variable PepeSUI._transferFromExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1249) is too similar to PepeSUI._transferFromExcluded(address,address,uint256).tTransferAmou
nt (contracts/Token.sol#1251)
Variable PepeSUI._transferFromExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1249) is too similar to PepeSUI._transferStandard(address,address,uint256).tTransferAmount (
contracts/Token.sol#1206)
Variable PepeSUI._transferFromExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1249) is too similar to PepeSUI._getTValues(uint256).tTransferAmount (contracts/Token.sol#94
2)
Variable PepeSUI._transferFromExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1249) is too similar to PepeSUI._transferToExcluded(address,address,uint256).tTransferAmount
 (contracts/Token.sol#1228)
Variable PepeSUI.reflectionFromToken(uint256,bool).rTransferAmount (contracts/Token.sol#771) is too similar to PepeSUI._getValues(uint256).tTransferAmount (contracts/Token.sol#913)
Variable PepeSUI._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1226) is too similar to PepeSUI._transferBothExcluded(address,address,uint256).tTransferAmount
 (contracts/Token.sol#1274)
Variable PepeSUI._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1272) is too similar to PepeSUI._transferFromExcluded(address,address,uint256).tTransferAmou
nt (contracts/Token.sol#1251)
Variable PepeSUI._getValues(uint256).rTransferAmount (contracts/Token.sol#918) is too similar to PepeSUI._getValues(uint256).tTransferAmount (contracts/Token.sol#913)
Variable PepeSUI.reflectionFromToken(uint256,bool).rTransferAmount (contracts/Token.sol#771) is too similar to PepeSUI._transferFromExcluded(address,address,uint256).tTransferAmount (contracts/
Token.sol#1251)
Variable PepeSUI._transferStandard(address,address,uint256).rTransferAmount (contracts/Token.sol#1204) is too similar to PepeSUI._transferFromExcluded(address,address,uint256).tTransferAmount (
contracts/Token.sol#1251)
Variable PepeSUI._getValues(uint256).rTransferAmount (contracts/Token.sol#918) is too similar to PepeSUI._transferFromExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1251
)
Variable PepeSUI.reflectionFromToken(uint256,bool).rTransferAmount (contracts/Token.sol#771) is too similar to PepeSUI._getTValues(uint256).tTransferAmount (contracts/Token.sol#942)
Variable PepeSUI._transferFromExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1249) is too similar to PepeSUI._transferBothExcluded(address,address,uint256).tTransferAmou
nt (contracts/Token.sol#1274)
Variable PepeSUI._getValues(uint256).rTransferAmount (contracts/Token.sol#918) is too similar to PepeSUI._transferStandard(address,address,uint256).tTransferAmount (contracts/Token.sol#1206)
Variable PepeSUI._transferStandard(address,address,uint256).rTransferAmount (contracts/Token.sol#1204) is too similar to PepeSUI._getTValues(uint256).tTransferAmount (contracts/Token.sol#942)
Variable PepeSUI._getValues(uint256).rTransferAmount (contracts/Token.sol#918) is too similar to PepeSUI._getTValues(uint256).tTransferAmount (contracts/Token.sol#942)
Variable PepeSUI._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1226) is too similar to PepeSUI._getValues(uint256).tTransferAmount (contracts/Token.sol#913)
Variable PepeSUI._getRValues(uint256,uint256,uint256,uint256,uint256).rTransferAmount (contracts/Token.sol#957) is too similar to PepeSUI._transferFromExcluded(address,address,uint256).tTransfe
rAmount (contracts/Token.sol#1251)
Variable PepeSUI._getValues(uint256).rTransferAmount (contracts/Token.sol#918) is too similar to PepeSUI._transferToExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1228)
Variable PepeSUI._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1226) is too similar to PepeSUI._transferFromExcluded(address,address,uint256).tTransferAmount
 (contracts/Token.sol#1251)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar

PepeSUI.DEAD (contracts/Token.sol#565) should be constant
PepeSUI._decimals (contracts/Token.sol#537) should be constant
PepeSUI._name (contracts/Token.sol#535) should be constant
PepeSUI._symbol (contracts/Token.sol#536) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

PepeSUI._tTotal (contracts/Token.sol#540) should be immutable
PepeSUI.mk (contracts/Token.sol#562) should be immutable
PepeSUI.mkTwo (contracts/Token.sol#563) should be immutable
PepeSUI.totalBuyFees (contracts/Token.sol#557-558) should be immutable
PepeSUI.totalSellFees (contracts/Token.sol#559-560) should be immutable
PepeSUI.uniswapV2Pair (contracts/Token.sol#568) should be immutable
PepeSUI.uniswapV2Router (contracts/Token.sol#567) 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/0x8ec3e2399b6eeb37ccd472f052bbcea05c54b5f7fc257bb9c83bf3d07fa2b44f

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

https://testnet.bscscan.com/tx/0x32e641ae42b33a521fcdc27a36a5ace320284e40df5fb10e6326cab905acb31a

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

https://testnet.bscscan.com/tx/0x35e462325641572b10d926a2b5f716f6ec612fd4bd2406f1b54723ac0b592e0e

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

https://testnet.bscscan.com/tx/0x2a7751abaaf97fc9fd8ae288f19eb3e726e0deffe8438d1d050885d3f4e4aabb

5- Buying when not excluded from fees (up to 12% tax) (passed):

https://testnet.bscscan.com/tx/0xbda8387cd92506695ef93c9b7a3f12923c36cd75577567145598876c62a64ddc

6- Selling when not excluded from fees (up to 12% tax) (passed):

https://testnet.bscscan.com/tx/0x94f0d3732273d88d9508b01fe39c62e8949aa0f19226442d4c8b0cf1645b1f46

FUNCTIONAL TESTING

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

https://testnet.bscscan.com/tx/0x43b9406251d395b7560053f623827acb792eebad2662d42870b0b565e3a3cfa6

8- Internal swap **(passed):**

marketing wallet received BUSD

https://testnet.bscscan.com/address/0x00cbf022aA35F4d2A3eF41145b80307c953D26BE#internaltx

8- Auto Liquidity **(passed):**

https://testnet.bscscan.com/token/0x87298bdd13f8c774c52c776af1eae0fe1324244c?a=0x000000000000000000000000000000000000dead

MANUAL TESTING

Centralization – Staking contract

Severity: Medium

Function: setStakingAddress

Lines: 584

Status: Not Resolved

Overview:

In the current implementation of the code, the owner has the ability to designate any address as a valid staking contract.

```
function setStakingAddress(address stakingAddress, bool _value) external onlyOwner {
require(isStakingAddress[stakingAddress] != _value, "account is already          the
value of '_value'");
require(isContract(stakingAddress), "call to non-contract");

    isStakingAddress[stakingAddress] = _value;
    emit SetStakingAddress(stakingAddress, _value);
}
```

Staking contracts can interact with the token to lock and unlock tokens. The purpose and usage of these lockTokens and unlockTokens functions are unclear at this point, and the staking contract is not within the scope of this audit.

Recommendation:

Provide further information about the staking contract, as well as the lockTokens and unlockTokens functions, including their intended use and the role of locked tokens within the system.

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