AuditAce
FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

Pepe 2.0
DATED : 20 May 23'

AUDIT SUMMARY

Project name – Pepe 2.0

Date: 20 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	0	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/0x7ce9A36Afc1fA469200f6D820EFF042CE79Ecb7C

Token Information

Token Name : Pepe 2.0

Token Symbol: Pepe 2.0

Decimals: 18

Token Supply:420,690,000,000,000

Token Address:
0x5204802DE0C726F42574D7Ee95Dbc6E0a53497e7

Checksum:
2787a8b9ae4805dca44a8f5181423ea113f6b441

Owner:
0x56d7db7cE1128Cf8cB288650Cf614b8EBa03256b

TOKEN OVERVIEW

Fees:

Buy Fees: upto: 0%

Sell Fees: upto: 0 %

Transfer Fees: 0%

Fees Privilige: none

Ownership: 0x56d7db7cE1128Cf8cB288650Cf614b8EBa03256b

Minting: No mint function

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

Other Priviliges: 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

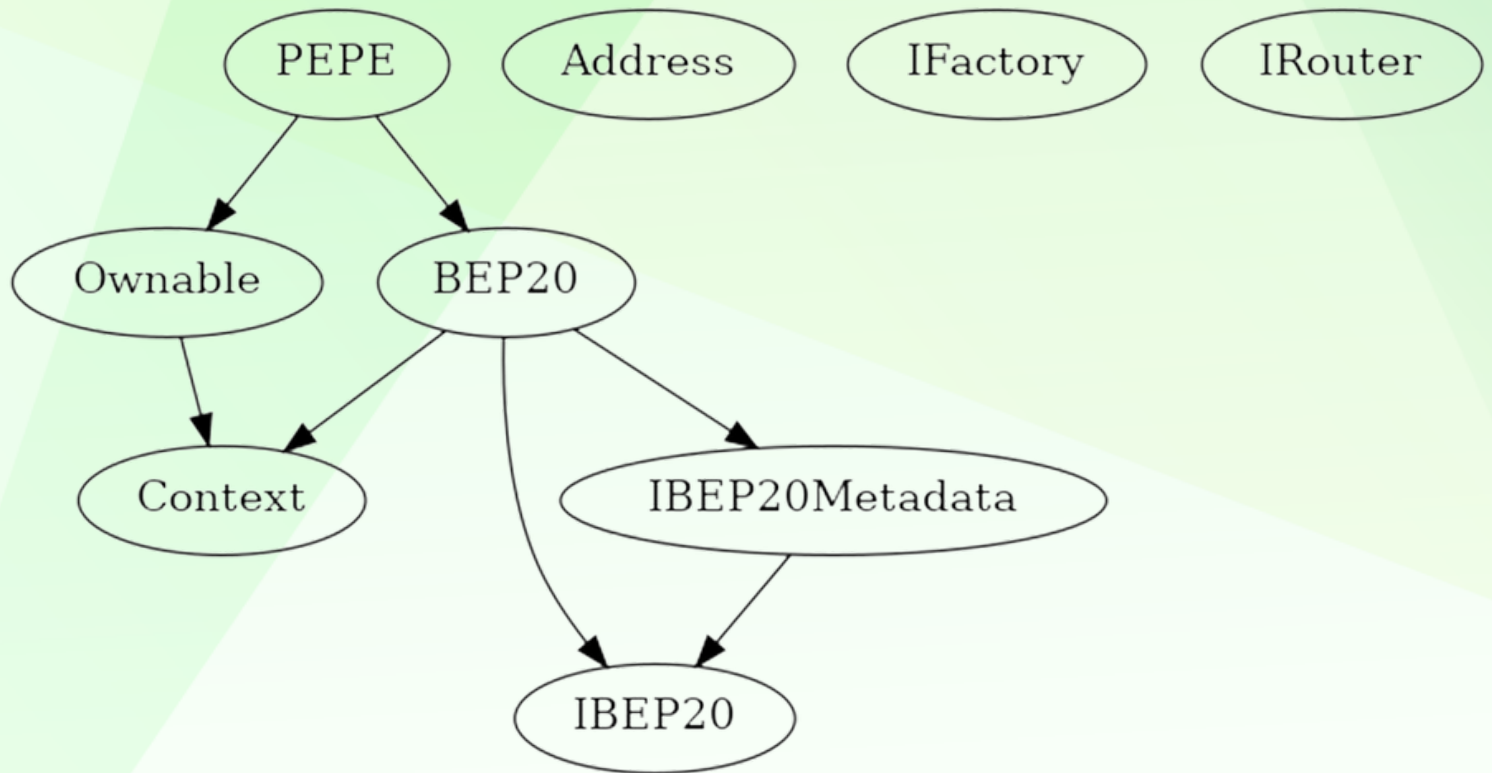| Severity	Found
◆ **Critical**	0
◆ **High-Risk**	0
◆ **Medium-Risk**	0
◆ **Low-Risk**	0
◆ **Gas Optimization / Suggestions**	0

INHERITANCE TREE

POINTS TO NOTE

- Owner is not able to set buy/sell/transfer fees
- Owner is not able to blacklist an arbitrary address.
- Owner is not able to disable trades
- Owner is not able to limit buy/sell/transfer/wallet amounts
- Owner is not able to mint new tokens
- Owner must enable trades manually

CONTRACT ASSESMENT

Contract	Type	Bases		
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
Context	Implementation			
└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		
IBEP20	Interface			
└	totalSupply	External ❗️		NO❗️
└	balanceOf	External ❗️		NO❗️
└	transfer	External ❗️	🛑	NO❗️
└	allowance	External ❗️		NO❗️
└	approve	External ❗️	🛑	NO❗️
└	transferFrom	External ❗️	🛑	NO❗️
IBEP20Metadata	Interface	IBEP20		
└	name	External ❗️		NO❗️
└	symbol	External ❗️		NO❗️
└	decimals	External ❗️		NO❗️
BEP20	Implementation	Context, IBEP20, IBEP20Metadata		
└	<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 🔒	🛑	
└	_tokengeneration	Internal 🔒	🛑	
└	_approve	Internal 🔒	🛑	
Address	Library			
└	sendValue	Internal 🔒	🛑	
Ownable	Implementation	Context		

CONTRACT ASSESMENT

└	<Constructor>	Public ❗	🛑	NO❗
└	owner	Public ❗		NO❗
└	renounceOwnership	Public ❗	🛑	onlyOwner
└	transferOwnership	Public ❗	🛑	onlyOwner
└	_setOwner	Private 🔐	🛑	
IFactory	Interface			
└	createPair	External ❗	🛑	NO❗
IRouter	Interface			
└	factory	External ❗		NO❗
└	WETH	External ❗		NO❗
PEPE	Implementation	BEP20, Ownable		
└	<Constructor>	Public ❗	🛑	BEP20
└	approve	Public ❗	🛑	NO❗
└	transferFrom	Public ❗	🛑	NO❗
└	increaseAllowance	Public ❗	🛑	NO❗
└	decreaseAllowance	Public ❗	🛑	NO❗
└	transfer	Public ❗	🛑	NO❗
└	_transfer	Internal 🔒	🛑	
└	EnableTrading	External ❗	🛑	onlyOwner
└	updateWhitelist	External ❗	🛑	onlyOwner
└	bulkWhitelist	External ❗	🛑	onlyOwner
└	rescueBNB	External ❗	🛑	onlyOwner
└	rescueBSC20	External ❗	🛑	onlyOwner
└	burnBSC20	External ❗	🛑	onlyOwner
└	<Receive Ether>	External ❗	💶	NO❗

Legend

Symbol	Meaning
🛑	Function can modify state
💶	Function is payable

STATIC ANALYSIS

```
Address.sendValue(address,uint256) (contracts/Token.sol#338-349) is never used and should be removed
Context._msgData() (contracts/Token.sol#14-17) 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#338-349):
        - (success) = recipient.call{value: amount}() (contracts/Token.sol#344)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Variable BEP20._balances (contracts/Token.sol#70) is not in mixedCase
Variable BEP20._allowances (contracts/Token.sol#72) is not in mixedCase
Function IRouter.WETH() (contracts/Token.sol#402) is not in mixedCase
Function PEPE.EnableTrading() (contracts/Token.sol#511-514) is not in mixedCase
Parameter PEPE.updateWhitelist(address,bool)._address (contracts/Token.sol#516) is not in mixedCase
Constant PEPE.deadWallet (contracts/Token.sol#413-414) is not in UPPER_CASE_WITH_UNDERSCORES
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (contracts/Token.sol#15)" inContext (contracts/Token.sol#9-18)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

PEPE.constructor() (contracts/Token.sol#418-435) uses literals with too many digits:
        - _tokengeneration(msg.sender,420690000000000 * 10 ** decimals()) (contracts/Token.sol#419)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

PEPE.pair (contracts/Token.sol#409) should be immutable
PEPE.router (contracts/Token.sol#408) 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/0xd229fbb14857406f2a8a8d6b7d
6bce6b93414ecef496c7c9c63cd62a987d3f5d

2- Buying (0% tax) (passed):
https://testnet.bscscan.com/tx/0xe142be7b9dc290e7754bd9c2d5
b3c1b6a389c7730a8e9a5dc5aa5294d81554ca

3- Selling (0% tax) (passed):
https://testnet.bscscan.com/tx/0xed3be86745aa51382ef0fda696
dbf0e3fe3d26ff14adf0bcf6d51976b2f40dff

4- Transferring (0% tax) (passed):
https://testnet.bscscan.com/tx/0x2716198a6a789d97b97a525406
6ba7df45588a62d1fd2b9dba6cb0aed785870b

FUNCTIONAL TESTING

No issues Found

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