auditace.tech

AuditAce
FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

20bPEPE

DATED : 7 July 23'

AUDIT SUMMARY

Project name – 20bPEPE

Date: 7 July, 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 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/0x0a6a37991f738ef171d0030b0e7FaBa087d0189d

Token Information

Token Name : 2.0 Baby Pepe

Token Symbol: 20bPEPE

Decimals: 9

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

Token Address:
0x103F09E68566655674266654DC5C0130865fcFbC

Checksum:
a53705f3eb183560fbd781b3424a8bd538c2271c

Owner:
0xBF78D4C6B1dfef656C8803d17509714c8f1E3066
(at time of writing the audit)

Deployer:
0xBF78D4C6B1dfef656C8803d17509714c8f1E3066

TOKEN OVERVIEW

Fees:

Buy Fees: 0-3%

Sell Fees: 0-3%

Transfer Fees: 0%

Fees Privilege: Owner

Ownership: owned

Minting: none

Max Tx Amount/ Max Wallet Amount: Yes

Blacklist: No

Other Privileges: - Initial distribution of the tokens

- modifying 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

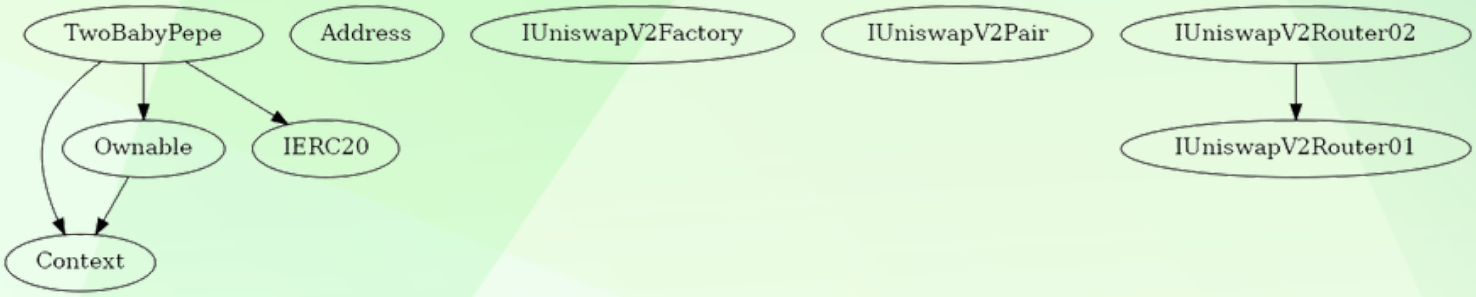
CLASSIFICATION OF RISK

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	0
Low-Risk	0
Gas Optimization / Suggestions	0

INHERITANCE TREE



POINTS TO NOTE

- owner is not able to set buy/sell fees more than 3%
- owner is not able to set transfer fees (0%)
- owner is not able to blacklist an arbitrary wallet
- owner is not able to set limit for buy/sell/transfer/holding amounts
- owner is not able to mint new tokens
- owner is not able to disable trades
- owner can exclude/include an address from fees
- owner can set staking address
- owner can lock/unlock tokens for staking
- owner can update buy/sell fees
- owner can set swap tokens at amount
- owner can enable/disable swap
- owner can claim stuck tokens
- owner can exclude/include an address from rewards
- owner can transfer ownership
- owner can renounce ownership

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 ❗

CONTRACT ASSESMENT

└	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 ❗
TwoBabyPepe	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 ❗
└	clearStuckTokens	External ❗	🛑	NO ❗
└	updateFeeBuy	Public ❗	🛑	onlyOwner
└	updateFeeSell	Public ❗	🛑	onlyOwner
└	_reflectFee	Private 🔐	🛑	
└	_getValues	Private 🔐		
└	_getTValues	Private 🔐		
└	_getRValues	Private 🔐		
└	_getRate	Private 🔐		
└	_getCurrentSupply	Private 🔐		
└	_takeLiquidity	Private 🔐	🛑	

CONTRACT ASSESMENT

└	_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

STATIC ANALYSIS

```
Variable TwoBabyPepe._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#907) is too similar to TwoBabyPepe._transferFromExcl
uded(address,address,uint256).tTransferAmount (contracts/Token.sol#928)
Variable TwoBabyPepe._getValues(uint256).rTransferAmount (contracts/Token.sol#650) is too similar to TwoBabyPepe._getTValues(uint256).tTransferAmount (cont
racts/Token.sol#659)
Variable TwoBabyPepe._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#945) is too similar to TwoBabyPepe._transferToExcl
uded(address,address,uint256).tTransferAmount (contracts/Token.sol#909)
Variable TwoBabyPepe._transferStandard(address,address,uint256).rTransferAmount (contracts/Token.sol#889) is too similar to TwoBabyPepe._transferBothExclud
ed(address,address,uint256).tTransferAmount (contracts/Token.sol#947)
Variable TwoBabyPepe.reflectionFromToken(uint256,bool).rTransferAmount (contracts/Token.sol#567) is too similar to TwoBabyPepe._getValues(uint256).tTransfe
rAmount (contracts/Token.sol#649)
Variable TwoBabyPepe._getRValues(uint256,uint256,uint256,uint256,uint256).rTransferAmount (contracts/Token.sol#672) is too similar to TwoBabyPepe._transfer
BothExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#947)
Variable TwoBabyPepe._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#945) is too similar to TwoBabyPepe._transferFromEx
cluded(address,address,uint256).tTransferAmount (contracts/Token.sol#928)
Variable TwoBabyPepe._transferStandard(address,address,uint256).rTransferAmount (contracts/Token.sol#889) is too similar to TwoBabyPepe._getTValues(uint256
).tTransferAmount (contracts/Token.sol#659)
Variable TwoBabyPepe._getRValues(uint256,uint256,uint256,uint256,uint256).rTransferAmount (contracts/Token.sol#672) is too similar to TwoBabyPepe._getTValu
es(uint256).tTransferAmount (contracts/Token.sol#659)
Variable TwoBabyPepe.reflectionFromToken(uint256,bool).rTransferAmount (contracts/Token.sol#567) is too similar to TwoBabyPepe._transferToExcluded(address,
address,uint256).tTransferAmount (contracts/Token.sol#909)
Variable TwoBabyPepe._getRValues(uint256,uint256,uint256,uint256,uint256).rTransferAmount (contracts/Token.sol#672) is too similar to TwoBabyPepe._transfer
Standard(address,address,uint256).tTransferAmount (contracts/Token.sol#891)
Variable TwoBabyPepe._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#907) is too similar to TwoBabyPepe._transferBothExcl
uded(address,address,uint256).tTransferAmount (contracts/Token.sol#947)
Variable TwoBabyPepe._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#907) is too similar to TwoBabyPepe._getTValues(uint2
56).tTransferAmount (contracts/Token.sol#659)
Variable TwoBabyPepe._getValues(uint256).rTransferAmount (contracts/Token.sol#650) is too similar to TwoBabyPepe._getValues(uint256).tTransferAmount (contr
acts/Token.sol#649)
Variable TwoBabyPepe.reflectionFromToken(uint256,bool).rTransferAmount (contracts/Token.sol#567) is too similar to TwoBabyPepe._transferFromExcluded(addres
s,address,uint256).tTransferAmount (contracts/Token.sol#928)
Variable TwoBabyPepe._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#907) is too similar to TwoBabyPepe._transferStandard
(address,address,uint256).tTransferAmount (contracts/Token.sol#891)
Variable TwoBabyPepe._transferFromExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#926) is too similar to TwoBabyPepe._transferToExcl
uded(address,address,uint256).tTransferAmount (contracts/Token.sol#909)
Variable TwoBabyPepe._getRValues(uint256,uint256,uint256,uint256,uint256).rTransferAmount (contracts/Token.sol#672) is too similar to TwoBabyPepe._getValue
s(uint256).tTransferAmount (contracts/Token.sol#649)
Variable TwoBabyPepe._transferStandard(address,address,uint256).rTransferAmount (contracts/Token.sol#889) is too similar to TwoBabyPepe._getValues(uint256)
.tTransferAmount (contracts/Token.sol#649)
Variable TwoBabyPepe._getValues(uint256).rTransferAmount (contracts/Token.sol#650) is too similar to TwoBabyPepe._transferToExcluded(address,address,uint25
6).tTransferAmount (contracts/Token.sol#909)
Variable TwoBabyPepe._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#945) is too similar to TwoBabyPepe._getTValues(uin
t256).tTransferAmount (contracts/Token.sol#659)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar

TwoBabyPepe.DEAD (contracts/Token.sol#420) should be constant
TwoBabyPepe._decimals (contracts/Token.sol#395) should be constant
TwoBabyPepe._name (contracts/Token.sol#393) should be constant
TwoBabyPepe._symbol (contracts/Token.sol#394) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

TwoBabyPepe.DEV (contracts/Token.sol#421) should be immutable
TwoBabyPepe._tTotal (contracts/Token.sol#398) should be immutable
TwoBabyPepe.mk (contracts/Token.sol#418) should be immutable
TwoBabyPepe.totalBuyFees (contracts/Token.sol#415) should be immutable
TwoBabyPepe.totalSellFees (contracts/Token.sol#416) should be immutable
TwoBabyPepe.uniswapV2Pair (contracts/Token.sol#424) should be immutable
TwoBabyPepe.uniswapV2Router (contracts/Token.sol#423) 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

1- Adding liquidity (passed):
https://testnet.bscscan.com/tx/0x0ce395e42e201418514fa6fb2fd2fda3c352d10633df5bfbf67f962eb92a6fbf

2- Buying when excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/https://testnet.bscscan.com/tx/0x43b3d665f6420d2e1a400f7dbfd9f85e10751338b815ee48728cf477357542e0

3- Selling when excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0x3813a9097a4fb81b2839e268df8b0d54b5c193a2077ebf3c8ab3b9e82c04ae4f

4- Transferring when excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0xf4ff0cf28f4b56c233d25cdae95f588dd6f888009126057c252b7b026c1b6d4d

5- Buying when not excluded from fees (0-3% tax)(passed):
https://testnet.bscscan.com/tx/0xebb0c314715c35fc801d9e3db029e5dac1fe28f8ba06274a3964e1f8f3cb61e4

6- Selling when not excluded from fees (0-6% tax) (passed):
https://testnet.bscscan.com/tx/0x2dae928822c430cf497384d408dd0551118ed4b0000cffaa4c51d75598546fce

FUNCTIONAL TESTING

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

https://testnet.bscscan.com/tx/0x0eceff0d166796cc214287f1b9d5f391c81634dd8f1242e84a3212b350295ff8

8- Internal swap (BNB fee + auto liquidity) (passed):

https://testnet.bscscan.com/tx/0x2dae928822c430cf497384d408dd0551118ed4b0000cffaa4c51d75598546fce

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