AuditAce
FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

Pepe Escobar

DATED : 08 May 23'

AUDIT SUMMARY

Project name – Pepe Escobar

Date: 08 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:

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/0xb4dAe005CC549573B742d20e463b289ab7797EdC

Token Information

Token Name : Pepe Escobar

Token Symbol: $ELPEPE

Decimals: 18

Token Supply:1,000,000,000

Token Address:
0xb7374eaBDb80E8b59F45ffe3f08B3BAA4a98DD7f

Checksum:
ff6d6364364fae41a49e053a3bddf9c42529d470

Owner:
0x2BAEA30c14d0e434B78316cFA7742960739D77Dc

Deployer:
0x2BAEA30c14d0e434B78316cFA7742960739D77Dc

TOKEN OVERVIEW

Fees:

Buy Fees: upto 10 %

Sell Fees: upto 10 %

Transfer Fees: upto 10%

Fees Privilige: owner

Ownership : Owned

Minting: No mint function

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

Other Priviliges: including or excluding from fees -
changing swap threshold - modifying fees - 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

✅ **Gasless Send**

✅ Private modifier

✅ Using block.timestamp

✅ Multiple Sends

✅ Re-entrancy

✅ Using Suicide

✅ Tautology or contradiction

✅ Gas Limitand Loops

✅ Timestamp Dependence

✅ Address hardcoded

✅ Revert/require functions

✅ Exception Disorder

✅ Use of tx.origin

✅ Using inline assembly

✅ Integer overflow/underflow

✅ Divide before multiply

✅ Dangerous strict equalities

✅ Missing Zero Address Validation

✅ Using SHA3

✅ Compiler version not fixed

✅ 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
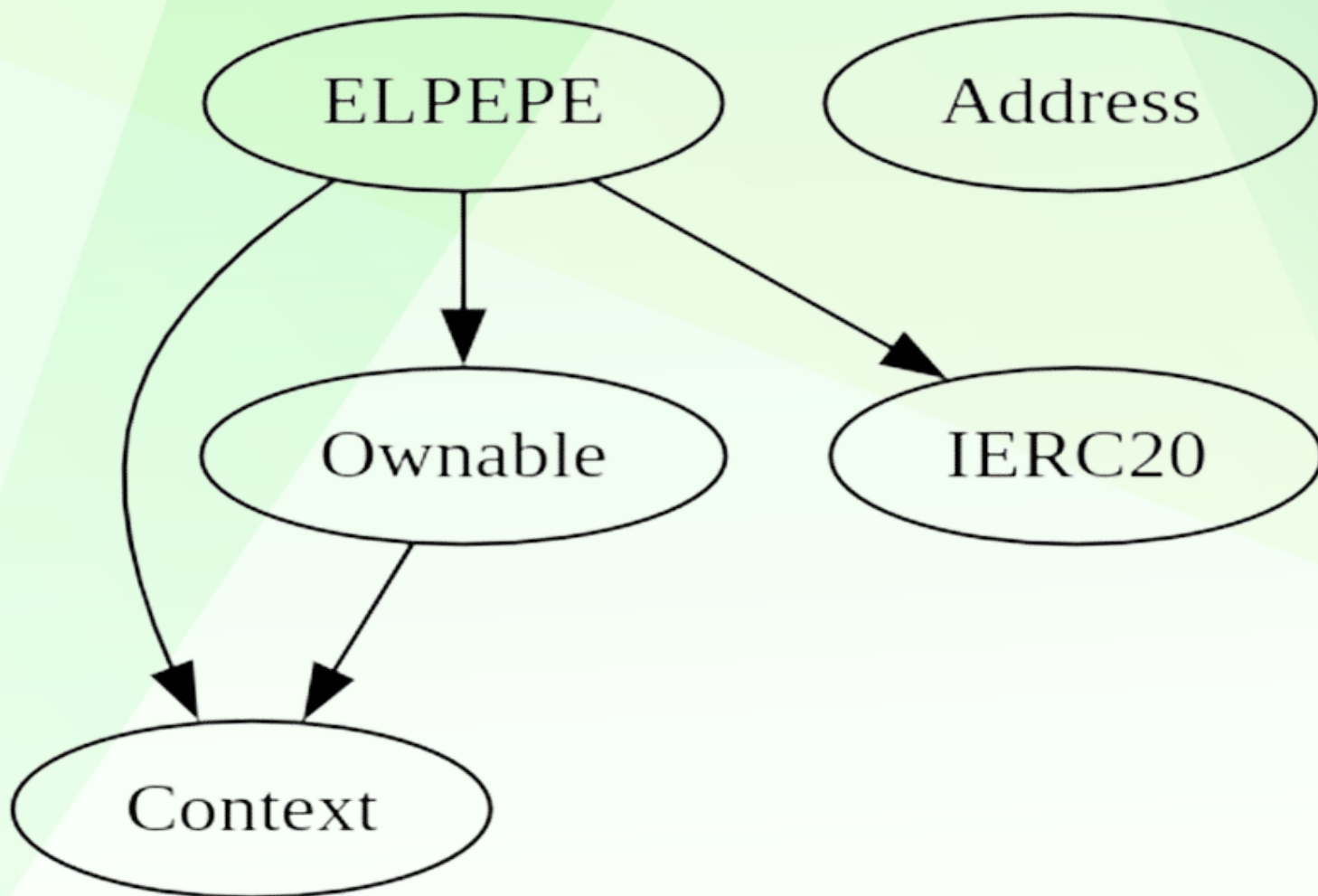
POINTS TO NOTE

- Owner is not able to set set buy/sell/transfer fees over (10%) each
- 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**
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❗

CONTRACT ASSESMENT

└	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❗

CONTRACT ASSESMENT

└	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❗
ELPEPE	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
└	_reflectFee	Private 🔒	🛑	
└	_getValues	Private 🔒		
└	_getTValues	Private 🔒		
└	_getRValues	Private 🔒		
└	_getRate	Private 🔒		
└	_getCurrentSupply	Private 🔒		

CONTRACT ASSESMENT

└	_takeLiquidity	Private 🔐	🛑	
└	_takeMarketing	Private 🔐	🛑	
└	calculateTaxFee	Private 🔐		
└	calculateLiquidityFee	Private 🔐		
└	calculateMarketingFee	Private 🔐		
└	removeAllFee	Private 🔐	🛑	
└	setBuyFee	Private 🔐	🛑	
└	setSellFee	Private 🔐	🛑	
└	isExcludedFromFee	Public ❗		NO❗
└	_approve	Private 🔐	🛑	
└	enableTrading	External ❗	🛑	onlyOwner
└	_transfer	Private 🔐	🛑	
└	buybackAndBurn	Private 🔐	🛑	
└	setSwapTokensAtAmount	External ❗	🛑	onlyOwner
└	setSwapEnabled	External ❗	🛑	onlyOwner
└	_tokenTransfer	Private 🔐	🛑	
└	_transferStandard	Private 🔐	🛑	
└	_transferToExcluded	Private 🔐	🛑	
└	_transferFromExcluded	Private 🔐	🛑	
└	_transferBothExcluded	Private 🔐	🛑	
└	excludeFromFees	External ❗	🛑	onlyOwner
└	changeMarketingWallet	External ❗	🛑	onlyOwner
└	setBuyFeePercentages	External ❗	🛑	onlyOwner
└	setSellFeePercentages	External ❗	🛑	onlyOwner
└	enableWalletToWalletTransferWithoutFee	External ❗	🛑	onlyOwner

Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

STATIC ANALYSIS

```
nt256).tBuybackBurn (contracts/Token.sol#1106)
Variable ELPEPE._takeLiquidity(uint256).rBuybackBurn (contracts/Token.sol#900) is too similar to ELPEPE._takeLiquidity(uint256).tBuybackBurn (contracts/Token.sol#897)
Variable ELPEPE._takeLiquidity(uint256).rBuybackBurn (contracts/Token.sol#900) is too similar to ELPEPE._transferStandard(address,address,uint256).tBuybackBurn (contract
s/Token.sol#1106)
Variable ELPEPE._transferFromExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1147) is too similar to ELPEPE._transferBothExcluded(address,address,
uint256).tTransferAmount (contracts/Token.sol#1172)
Variable ELPEPE._getRValues(uint256,uint256,uint256,uint256).rTransferAmount (contracts/Token.sol#873) is too similar to ELPEPE._transferStandard(address,address
,uint256).tTransferAmount (contracts/Token.sol#1104)
Variable ELPEPE._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1170) is too similar to ELPEPE._transferFromExcluded(address,address,
uint256).tTransferAmount (contracts/Token.sol#1149)
Variable ELPEPE._getRValues(uint256,uint256,uint256,uint256).rTransferAmount (contracts/Token.sol#873) is too similar to ELPEPE._transferToExcluded(address,addre
ss,uint256).tTransferAmount (contracts/Token.sol#1126)
Variable ELPEPE._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1170) is too similar to ELPEPE._transferBothExcluded(address,address,
uint256).tTransferAmount (contracts/Token.sol#1172)
Variable ELPEPE._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1124) is too similar to ELPEPE._transferBothExcluded(address,address,ui
nt256).tTransferAmount (contracts/Token.sol#1172)
Variable ELPEPE._getValues(uint256).rTransferAmount (contracts/Token.sol#834) is too similar to ELPEPE._transferBothExcluded(address,address,uint256).tTransferAmount (co
ntracts/Token.sol#1172)
Variable ELPEPE._transferFromExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1147) is too similar to ELPEPE._transferStandard(address,address,uint
256).tTransferAmount (contracts/Token.sol#1104)
Variable ELPEPE._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1170) is too similar to ELPEPE._getValues(uint256).tTransferAmount (c
ontracts/Token.sol#829)
Variable ELPEPE._getRValues(uint256,uint256,uint256,uint256,uint256).rTransferAmount (contracts/Token.sol#873) is too similar to ELPEPE._transferBothExcluded(address,add
ress,uint256).tTransferAmount (contracts/Token.sol#1172)
Variable ELPEPE._transferFromExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1147) is too similar to ELPEPE._transferToExcluded(address,address,ui
nt256).tTransferAmount (contracts/Token.sol#1126)
Variable ELPEPE.reflectionFromToken(uint256,bool).rTransferAmount (contracts/Token.sol#765) is too similar to ELPEPE._transferBothExcluded(address,address,uint256).tTran
sferAmount (contracts/Token.sol#1172)
Variable ELPEPE._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1170) is too similar to ELPEPE._transferStandard(address,address,uint
256).tTransferAmount (contracts/Token.sol#1104)
Variable ELPEPE._transferStandard(address,address,uint256).rTransferAmount (contracts/Token.sol#1102) is too similar to ELPEPE._transferBothExcluded(address,address,uint
256).tTransferAmount (contracts/Token.sol#1172)
Variable ELPEPE._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1124) is too similar to ELPEPE._transferStandard(address,address,uint25
6).tTransferAmount (contracts/Token.sol#1104)
Variable ELPEPE._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1170) is too similar to ELPEPE._transferToExcluded(address,address,ui
nt256).tTransferAmount (contracts/Token.sol#1126)
Variable ELPEPE._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1170) is too similar to ELPEPE._getTValues(uint256).tTransferAmount (
contracts/Token.sol#858)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar

ELPEPE.DEAD (contracts/Token.sol#566) is never used in ELPEPE (contracts/Token.sol#525-1262)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

ELPEPE._decimals (contracts/Token.sol#539) should be constant
ELPEPE._name (contracts/Token.sol#537) should be constant
ELPEPE._symbol (contracts/Token.sol#538) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

ELPEPE._tTotal (contracts/Token.sol#542) should be immutable
ELPEPE.uniswapV2Pair (contracts/Token.sol#569) should be immutable
ELPEPE.uniswapV2Router (contracts/Token.sol#568) 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/0xa789eca58a32a9c9b05301c2ab
ae61fae1999ee38ad309dee6e5cb8056879192

2- Buying when excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0x6f14f7675da5c3d374906ab075
e900ea44201f4a20e95ccdbc59cc3c4a010f2c

3- Selling when excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0xe6ef912c8ddcd9c6a309937a23f
fa8503b9d4f0593418b1daae8d0a16b752f1c

4- Transferring when excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0x36bf015c441eb5f455132fe390a
c62d48a6843b8e75e7326f444e88d7f790e2c

5- Buying when not excluded (up to 10% tax) (passed):
https://testnet.bscscan.com/tx/0x1141f5b74a02f5a02d0083753fe
19cdd434a8e0fdb3ea030a8c69a0f3fadae02

6- Selling when not excluded (up to 10% tax) (passed):
https://testnet.bscscan.com/tx/0x1141f5b74a02f5a02d0083753fe
19cdd434a8e0fdb3ea030a8c69a0f3fadae02

FUNCTIONAL TESTING

7- Transferring when not excluded (upto 10% tax) (passed):
https://testnet.bscscan.com/tx/0x9551bd98b8fa6e16e8cee19d7f2aa35d96695d6d1ce127d3b2cc85fc42945e3c

8- Internal swap (buyback And Burn and marketing fees)
(passed):
https://testnet.bscscan.com/tx/0x1a42d4b8140b25d82f8fddf5c59d8348756b3d19c22f9dd9f6145a5fdf69f1f8

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 == false, "Trading is already enabled");
    tradingEnabled = 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 – naming convention

Status: Not Resolved

Overview:

_takeLiquidity is collecting fees for buyback not liquidity

Suggestion:

change the name to something more related to buyback, like
"_takeBuyBackFee"

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