AuditAce
FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

Pepe GPT

DATED : 14 May 23'

AUDIT SUMMARY

Project name – Pepe GPT

Date: 14 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	1	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/0xab1ed8473404ee2e447e797363d15c8162c72cc4

Token Information

Token Name : PepeGPT

Token Symbol: PEPEG

Decimals: 9

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

Token Address:
0x96EFe18F7B7E807c2014EF9538BA671b448ecf14

Checksum:
49f8c57e4505f8cb6f3d3058deccff7262bfe5a2

Owner:
0x233382C9b02eF82528e8D3f07573D7055dbE8730

TOKEN OVERVIEW

Fees:

Buy Fees: upto 12 %

Sell Fees: upto 12 %

Transfer Fees: 0%

Fees Privilige: owner

Ownership : owned

Minting: No mint function

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

Other Priviliges: changing swap threshold - changing fees - modifying swap settings

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
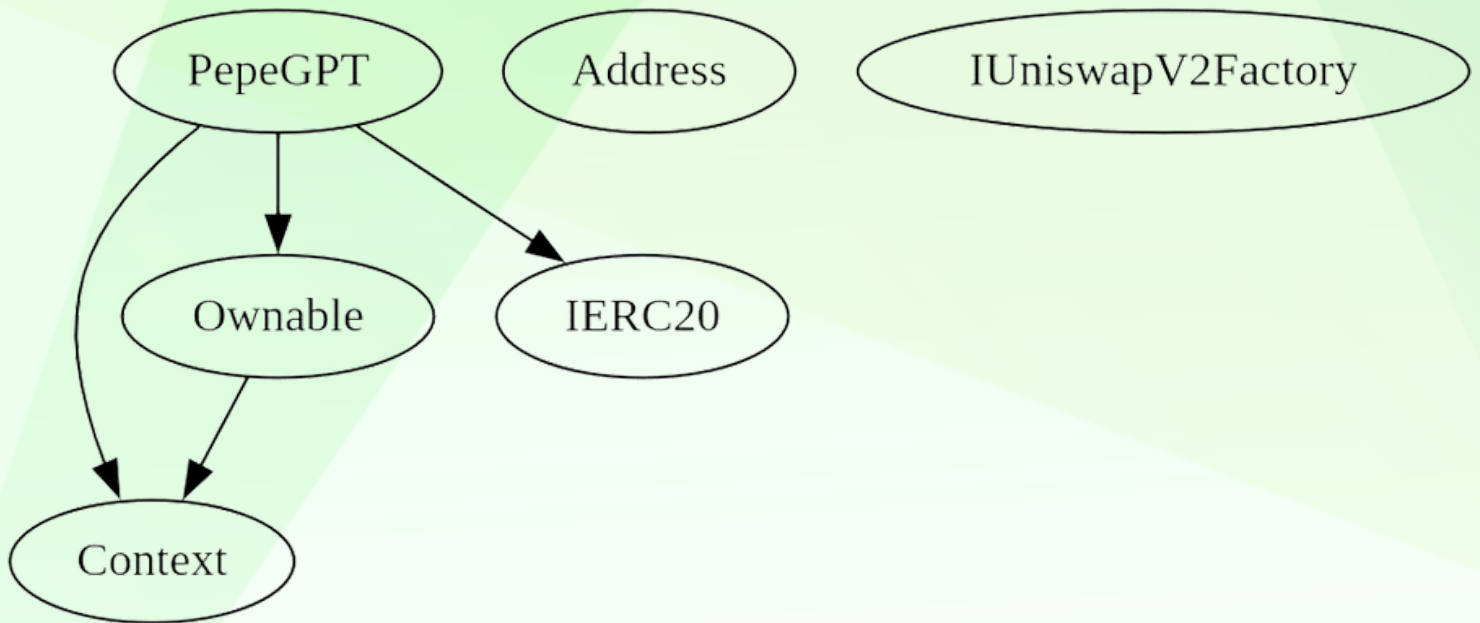
◆ **High-Risk**

0

◆ **Medium-Risk**

0

◆ **Low-Risk**

1

◆ **Gas Optimization / Suggestions**

0

INHERITANCE TREE

POINTS TO NOTE

- owner is not able to set buy/sell fees more than 12%
- 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			

CONTRACT ASSESMENT

└	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	

CONTRACT ASSESMENT

└	swapETHForExactTokens	External 🔓	🆔	NO🔲
└	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🔲
PepeGPT	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 🔓	⬛	NO🔲
└	setStakingAddress	External 🔓	⬛	onlyOwner
└	lockToken	Public 🔓	⬛	NO🔲
└	unlockToken	Public 🔓	⬛	NO🔲
└	updateFeeBuy	Public 🔓	⬛	onlyOwner
└	updateFeeSell	Public 🔓	⬛	onlyOwner

CONTRACT ASSESMENT

└	_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

STATIC ANALYSIS

```
Variable PepeGPT._transferStandard(address,address,uint256).rTransferAmount (contracts/Token.sol#1300) is too similar to PepeGPT._getValues(uint256).tTransferAmount (contracts/To
ken.sol#1009)
Variable PepeGPT._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1368) is too similar to PepeGPT._transferStandard(address,address,uint256).tT
ransferAmount (contracts/Token.sol#1302)
Variable PepeGPT._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1368) is too similar to PepeGPT._transferFromExcluded(address,address,uint256
).tTransferAmount (contracts/Token.sol#1347)
Variable PepeGPT._getRValues(uint256,uint256,uint256,uint256,uint256).rTransferAmount (contracts/Token.sol#1053) is too similar to PepeGPT._getTValues(uint256).tTransferAmount (c
ontracts/Token.sol#1038)
Variable PepeGPT._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1322) is too similar to PepeGPT._transferStandard(address,address,uint256).tTra
nsferAmount (contracts/Token.sol#1302)
Variable PepeGPT._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1368) is too similar to PepeGPT._getValues(uint256).tTransferAmount (contract
s/Token.sol#1009)
Variable PepeGPT._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1368) is too similar to PepeGPT._transferToExcluded(address,address,uint256).
tTransferAmount (contracts/Token.sol#1324)
Variable PepeGPT._transferStandard(address,address,uint256).rTransferAmount (contracts/Token.sol#1300) is too similar to PepeGPT._getTValues(uint256).tTransferAmount (contracts/T
oken.sol#1038)
Variable PepeGPT._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1322) is too similar to PepeGPT._getValues(uint256).tTransferAmount (contracts/
Token.sol#1009)
Variable PepeGPT._transferFromExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1345) is too similar to PepeGPT._transferStandard(address,address,uint256).tT
ransferAmount (contracts/Token.sol#1302)
Variable PepeGPT._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1368) is too similar to PepeGPT._getTValues(uint256).tTransferAmount (contrac
ts/Token.sol#1038)
Variable PepeGPT._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1322) is too similar to PepeGPT._getTValues(uint256).tTransferAmount (contracts
/Token.sol#1038)
Variable PepeGPT._getValues(uint256).rTransferAmount (contracts/Token.sol#1014) is too similar to PepeGPT._transferStandard(address,address,uint256).tTransferAmount (contracts/To
ken.sol#1302)
Variable PepeGPT._transferFromExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1345) is too similar to PepeGPT._getValues(uint256).tTransferAmount (contract
s/Token.sol#1009)
Variable PepeGPT._transferFromExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1345) is too similar to PepeGPT._transferToExcluded(address,address,uint256).
tTransferAmount (contracts/Token.sol#1324)
Variable PepeGPT._getRValues(uint256,uint256,uint256,uint256,uint256).rTransferAmount (contracts/Token.sol#1053) is too similar to PepeGPT._transferStandard(address,address,uint2
56).tTransferAmount (contracts/Token.sol#1302)
Variable PepeGPT._getValues(uint256).rTransferAmount (contracts/Token.sol#1014) is too similar to PepeGPT._getValues(uint256).tTransferAmount (contracts/Token.sol#1009)
Variable PepeGPT._getRValues(uint256,uint256,uint256,uint256,uint256).rTransferAmount (contracts/Token.sol#1053) is too similar to PepeGPT._transferFromExcluded(address,address,u
int256).tTransferAmount (contracts/Token.sol#1347)
Variable PepeGPT._transferFromExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1345) is too similar to PepeGPT._getTValues(uint256).tTransferAmount (contrac
ts/Token.sol#1038)
Variable PepeGPT._getRValues(uint256,uint256,uint256,uint256,uint256).rTransferAmount (contracts/Token.sol#1053) is too similar to PepeGPT._getValues(uint256).tTransferAmount (co
ntracts/Token.sol#1009)
Variable PepeGPT._getRValues(uint256,uint256,uint256,uint256,uint256).rTransferAmount (contracts/Token.sol#1053) is too similar to PepeGPT._transferToExcluded(address,address,uin
t256).tTransferAmount (contracts/Token.sol#1324)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar

PepeGPT.DEAD (contracts/Token.sol#655) should be constant
PepeGPT._decimals (contracts/Token.sol#627) should be constant
PepeGPT._name (contracts/Token.sol#625) should be constant
PepeGPT._symbol (contracts/Token.sol#626) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

PepeGPT.DEV (contracts/Token.sol#656) should be immutable
PepeGPT._tTotal (contracts/Token.sol#630) should be immutable
PepeGPT.mk (contracts/Token.sol#652) should be immutable
PepeGPT.mkTwo (contracts/Token.sol#653) should be immutable
PepeGPT.totalBuyFees (contracts/Token.sol#647-648) should be immutable
PepeGPT.totalSellFees (contracts/Token.sol#649-650) should be immutable
PepeGPT.uniswapV2Pair (contracts/Token.sol#659) should be immutable
PepeGPT.uniswapV2Router (contracts/Token.sol#658) 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/0xbff3d2ee08bc34c7c4ee87092c78971f101c9446cfe1b46791ee64638e238fdf

2- Buying when excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0xa35404088c75c6fd0cc763a0c1f58a9625543a8aab0f596c1dc6d4c8e0362b56

3- Selling when excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0x463f66297b155b371fcf24d89a5341f2812e351be7e62c852f4c583d54b06f84

4- Transferring when excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0xa8bb9ef5114b0d8e7edb423902ead674f9038dbc16c23589983b4f9415b41e59

5- Buying when not excluded (0-12% tax) (passed):
https://testnet.bscscan.com/tx/0xafe099ad68ee7c90badfc75a03d235a31bbb5fd6b83ce78da5ebd28249032b74

6- Selling when not excluded (0- 12% tax) (passed):
https://testnet.bscscan.com/tx/0x4cd42d5184fdba9d3c11bd275153d0b5fc22787cfa0fbd96a4596905c4536281

FUNCTIONAL TESTING

7- Transferring when not excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0xed9b80aeae2083e7a34765e5fe594b6f0a8e02c7495acc6580eb22eafe8fe3c8

8- Auto liquidity, Marketing fee (passed**):**
https://testnet.bscscan.com/tx/0x4cd42d5184fdba9d3c11bd275153d0b5fc22787cfa0fbd96a4596905c4536281

MANUAL TESTING

Logical – swap threshold

Severity: Low

function: setSwapTokensAtAmount

Status: Not Resolved

Overview:

Owner is able to set swapTokensAtAmount to a large number, this can increase slippage up to 49% if contract balance is more than swapTokensAtAmount

additionaly, 1e5 (100000) is not equal to 0.001% of total supply.

```
function setSwapTokensAtAmount(uint256 newAmount) external onlyOwner() {
    require(newAmount > totalSupply() / 1e5, "SwapTokensAtAmount must be greater than
        0.001% of total supply");
    swapTokensAtAmount = newAmount;
    emit SwapTokensAtAmountUpdated(newAmount);
}
```

Suggestion

To mitigate this Logical issue, make sure that swapTokensAtAmount is always less than 1% of supply

```
function setSwapTokensAtAmount(uint256 newAmount) external onlyOwner() {
    require(newAmount > totalSupply() / 4e12, "SwapTokensAtAmount must be greater than
0.001% of total supply");
    require(newAmount < totalSupply() / 4e14, "SwapTokensAtAmount must be greater than
0.001% of total supply");
    swapTokensAtAmount = newAmount;
    emit SwapTokensAtAmountUpdated(newAmount);
}
```

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