AuditAce

FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

METADOGE2

DATED : 10 July 23'

FUNCTIONAL TESTING

Centralization – Enabling Trades

Severity: **High**

function: enableTrading

Status: Not Resolved

Overview:

Owner of the contract must enable trades manually for investors, otherwise no one would be able to buy/sell/transfer their tokens.

```
function enableTrading() external onlyOwner {
    require(!isTradingEnabled, "Trading already enabled");
    isTradingEnabled = true;
    emit _enableTrading();
}
```

Suggestion

Its suggested to either enable trades prior to presale, or transfer ownership of the contract to a certified pinsksale safu developer to guearanee enabling of trades.

AUDIT SUMMARY

Project name – METADOGE2

Date: 10 July, 2023

Scope of Audit- Audit Ace was consulted to conduct the smart contract audit of the solidity source codes.

Audit Status: **Passed with High Risk**

Issues Found

Status	Critical	High	Medium	Low	Suggestion
Open	0	1	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/0xC0B7a79Bd06fc7FE3F2607c07f397b26244Ec4fa

Token Information

Token Name : META DOGE 2.0

Token Symbol: METADOGE2

Decimals: 9

Token Supply:100,000,000

Token Address:
0x30Ab698F605a277F129cFc754De309D344Bd92e6

Checksum:
f1bc1e948b9029e1f5028ecd3146efdbd6f0e4bc

Owner:
0xf5F2a0255310F97eda5ed25D6cB23c34e6a1B5Ea
(at time of writing the audit)

Deployer:
0xf5F2a0255310F97eda5ed25D6cB23c34e6a1B5Ea

TOKEN OVERVIEW

Fees:

Buy Fees: 0%

Sell Fees: 5%

Transfer Fees: 0%

Fees Privilege: Immutable Fees

Ownership: owned

Minting: none

Max Tx Amount/ Max Wallet Amount: Yes

Blacklist: No

Other Privileges: - enabling trades

- Initial distribution of the tokens

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

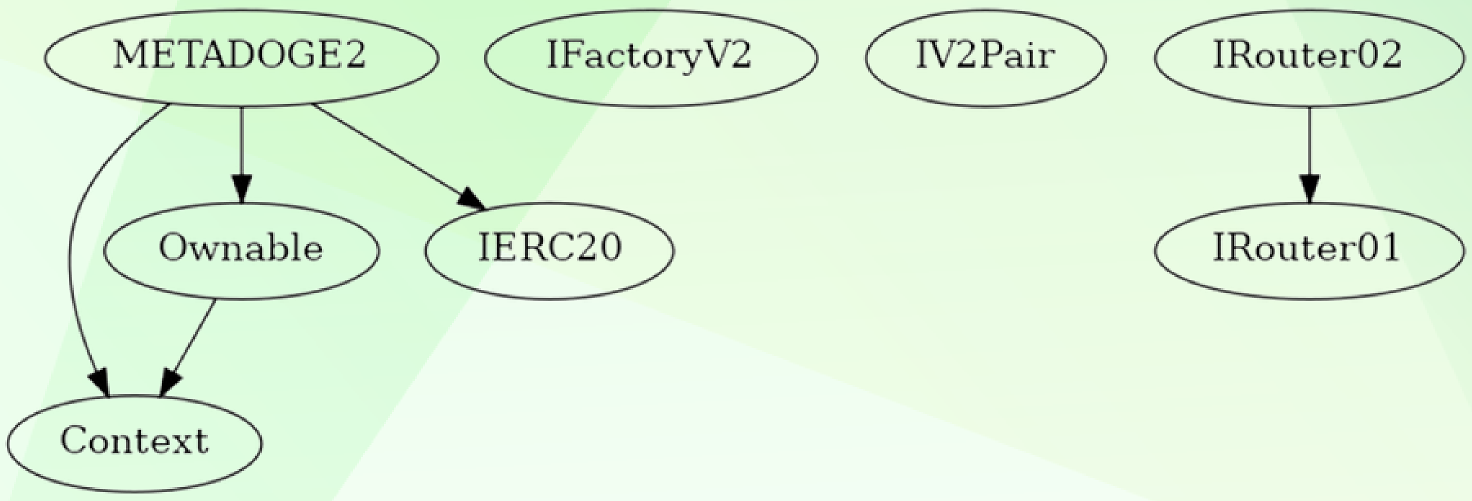
CLASSIFICATION OF RISK

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	1
🟧 Medium-Risk	0
🟨 Low-Risk	0
🟦 Gas Optimization / Suggestions	0

INHERITANCE TREE



POINTS TO NOTE

- Owner is not change current fees (5% buy / 0% sell / 0% transfer)
- Owner is not able to blacklist an address
- Owner is not able to disable buy/sell/transfers
- Owner is not able to set max wallet limit and minimum wallet limits
- 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 |  |||
| └ | <Constructor> | Public ❗️ | 🛑  |NO ❗️ |
| └ | _msgSender | Internal 🔒 |   | |
| └ | _msgData | Internal 🔒 |   | |
||||||
| **Ownable** | Implementation | Context |||
| └ | <Constructor> | Public ❗️ | 🛑  |NO ❗️ |
| └ | owner | Public ❗️ |   |NO ❗️ |
| └ | renounceOwnership | Public ❗️ | 🛑  | onlyOwner |
| └ | transferOwnership | Public ❗️ | 🛑  | onlyOwner |
| └ | _setOwner | Private 🔐 | 🛑  | |
||||||
| **IFactoryV2** | Interface |  |||
| └ | getPair | External ❗️ |   |NO ❗️ |
| └ | createPair | External ❗️ | 🛑  |NO ❗️ |
||||||
| **IV2Pair** | Interface |  |||
| └ | factory | External ❗️ |   |NO ❗️ |
| └ | getReserves | External ❗️ |   |NO ❗️ |
| └ | sync | External ❗️ | 🛑  |NO ❗️ |
||||||
| **IRouter01** | Interface |  |||
| └ | factory | External ❗️ |   |NO ❗️ |
| └ | WETH | External ❗️ |   |NO ❗️ |
| └ | addLiquidityETH | External ❗️ |  💵 |NO ❗️ |
| └ | addLiquidity | External ❗️ | 🛑  |NO ❗️ |
| └ | swapExactETHForTokens | External ❗️ |  💵 |NO ❗️ |
| └ | getAmountsOut | External ❗️ |   |NO ❗️ |
| └ | getAmountsIn | External ❗️ |   |NO ❗️ |
||||||
| **IRouter02** | Interface | IRouter01 |||
| └ | swapExactTokensForETHSupportingFeeOnTransferTokens | External ❗️ | 🛑  |NO ❗️ |
| └ | swapExactETHForTokensSupportingFeeOnTransferTokens | External ❗️ |  💵 |NO ❗️ |
| └ | swapExactTokensForTokensSupportingFeeOnTransferTokens | External ❗️ | 🛑  |NO ❗️ |
| └ | swapExactTokensForTokens | External ❗️ | 🛑  |NO ❗️ |
||||||
| **IERC20** | Interface |  |||
| └ | totalSupply | External ❗️ |   |NO ❗️ |
| └ | decimals | External ❗️ |   |NO ❗️ |
| └ | symbol | External ❗️ |   |NO ❗️ |
```

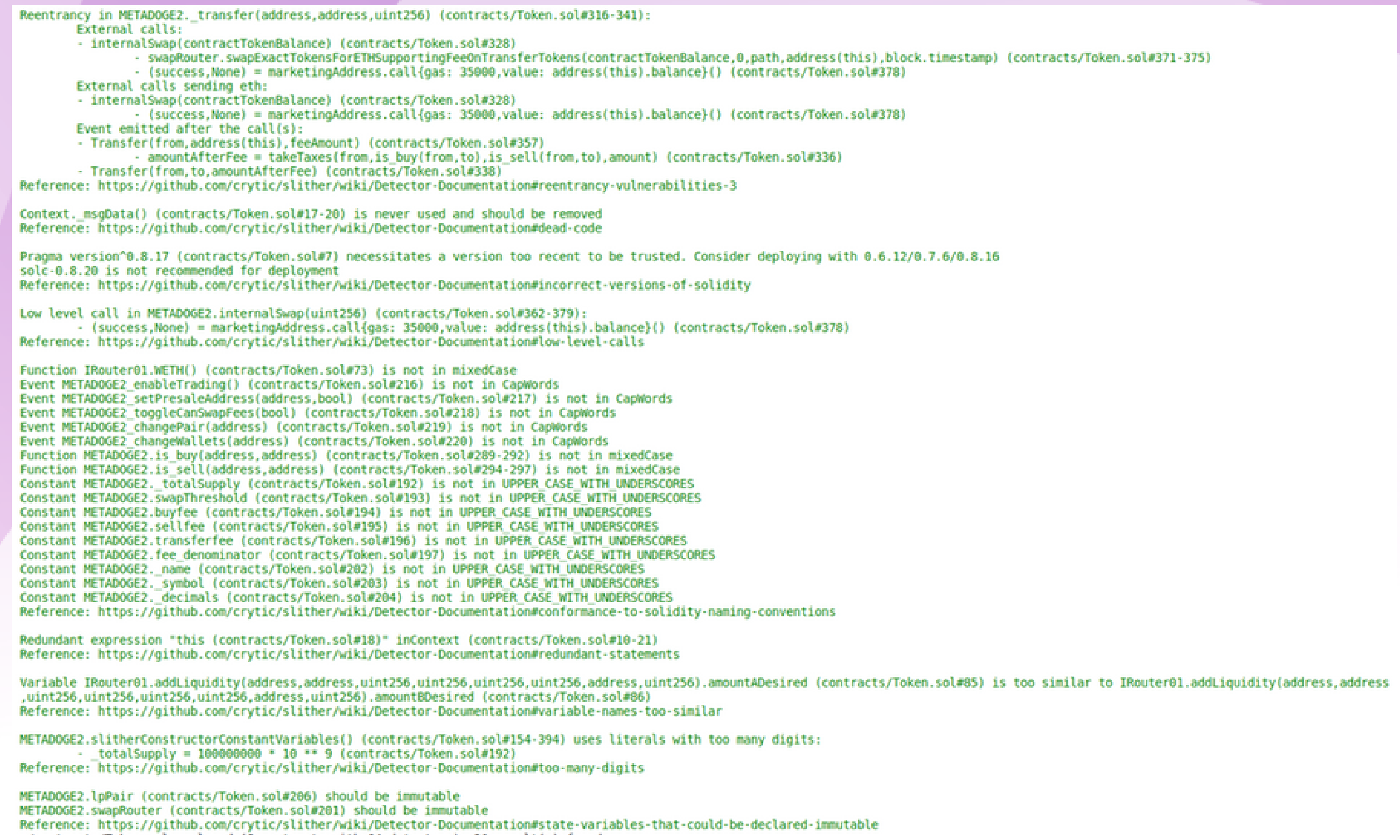
CONTRACT ASSESMENT

```
| └ | name | External ❗ |  |NO ❗ |
| └ | getOwner | External ❗ |  |NO ❗ |
| └ | balanceOf | External ❗ |  |NO ❗ |
| └ | transfer | External ❗ | 🛑 |NO ❗ |
| └ | allowance | External ❗ |  |NO ❗ |
| └ | approve | External ❗ | 🛑 |NO ❗ |
| └ | transferFrom | External ❗ | 🛑 |NO ❗ |
||||||
| **METADOGE2** | Implementation | Context, Ownable, IERC20 |||
| └ | totalSupply | External ❗ |  |NO ❗ |
| └ | decimals | External ❗ |  |NO ❗ |
| └ | symbol | External ❗ |  |NO ❗ |
| └ | name | External ❗ |  |NO ❗ |
| └ | getOwner | External ❗ |  |NO ❗ |
| └ | allowance | External ❗ |  |NO ❗ |
| └ | balanceOf | Public ❗ |  |NO ❗ |
| └ | <Constructor> | Public ❗ | 🛑 |NO ❗ |
| └ | <Receive Ether> | External ❗ | 💵 |NO ❗ |
| └ | transfer | Public ❗ | 🛑 |NO ❗ |
| └ | approve | External ❗ | 🛑 |NO ❗ |
| └ | _approve | Internal 🔒 | 🛑 | |
| └ | transferFrom | External ❗ | 🛑 |NO ❗ |
| └ | isNoFeeWallet | External ❗ |  |NO ❗ |
| └ | setNoFeeWallet | Public ❗ | 🛑 | onlyOwner |
| └ | isLimitedAddress | Internal 🔒 |  | |
| └ | is_buy | Internal 🔒 |  | |
| └ | is_sell | Internal 🔒 |  | |
| └ | canSwap | Internal 🔒 |  | |
| └ | changeLpPair | External ❗ | 🛑 | onlyOwner |
| └ | toggleCanSwapFees | External ❗ | 🛑 | onlyOwner |
| └ | _transfer | Internal 🔒 | 🛑 | |
| └ | changeWallets | External ❗ | 🛑 | onlyOwner |
| └ | takeTaxes | Internal 🔒 | 🛑 | |
| └ | internalSwap | Internal 🔒 | 🛑 | inSwapFlag |
| └ | setPresaleAddress | External ❗ | 🛑 | onlyOwner |
| └ | enableTrading | External ❗ | 🛑 | onlyOwner |

### Legend

|  Symbol  |  Meaning  |
|:--------:|-----------|
|    🛑    | Function can modify state |
|    💵    | Function is payable |
```

STATIC ANALYSIS



```
Reentrancy in METADOGE2._transfer(address,address,uint256) (contracts/Token.sol#316-341):
        External calls:
        - internalSwap(contractTokenBalance) (contracts/Token.sol#328)
                - swapRouter.swapExactTokensForETHSupportingFeeOnTransferTokens(contractTokenBalance,0,path,address(this),block.timestamp) (contracts/Token.sol#371-375)
                - (success,None) = marketingAddress.call{gas: 35000,value: address(this).balance}() (contracts/Token.sol#378)
        External calls sending eth:
        - internalSwap(contractTokenBalance) (contracts/Token.sol#328)
                - (success,None) = marketingAddress.call{gas: 35000,value: address(this).balance}() (contracts/Token.sol#378)
        Event emitted after the call(s):
        - Transfer(from,address(this),feeAmount) (contracts/Token.sol#357)
                - amountAfterFee = takeTaxes(from,is_buy(from,to),is_sell(from,to),amount) (contracts/Token.sol#336)
        - Transfer(from,to,amountAfterFee) (contracts/Token.sol#338)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

Context._msgData() (contracts/Token.sol#17-20) 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-0.8.20 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in METADOGE2.internalSwap(uint256) (contracts/Token.sol#362-379):
        - (success,None) = marketingAddress.call{gas: 35000,value: address(this).balance}() (contracts/Token.sol#378)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function IRouter01.WETH() (contracts/Token.sol#73) is not in mixedCase
Event METADOGE2_enableTrading() (contracts/Token.sol#216) is not in CapWords
Event METADOGE2_setPresaleAddress(address,bool) (contracts/Token.sol#217) is not in CapWords
Event METADOGE2_toggleCanSwapFees(bool) (contracts/Token.sol#218) is not in CapWords
Event METADOGE2_changePair(address) (contracts/Token.sol#219) is not in CapWords
Event METADOGE2_changeWallets(address) (contracts/Token.sol#220) is not in CapWords
Function METADOGE2.is_buy(address,address) (contracts/Token.sol#289-292) is not in mixedCase
Function METADOGE2.is_sell(address,address) (contracts/Token.sol#294-297) is not in mixedCase
Constant METADOGE2._totalSupply (contracts/Token.sol#192) is not in UPPER_CASE_WITH_UNDERSCORES
Constant METADOGE2.swapThreshold (contracts/Token.sol#193) is not in UPPER_CASE_WITH_UNDERSCORES
Constant METADOGE2.buyfee (contracts/Token.sol#194) is not in UPPER_CASE_WITH_UNDERSCORES
Constant METADOGE2.sellfee (contracts/Token.sol#195) is not in UPPER_CASE_WITH_UNDERSCORES
Constant METADOGE2.transferfee (contracts/Token.sol#196) is not in UPPER_CASE_WITH_UNDERSCORES
Constant METADOGE2.fee_denominator (contracts/Token.sol#197) is not in UPPER_CASE_WITH_UNDERSCORES
Constant METADOGE2._name (contracts/Token.sol#202) is not in UPPER_CASE_WITH_UNDERSCORES
Constant METADOGE2._symbol (contracts/Token.sol#203) is not in UPPER_CASE_WITH_UNDERSCORES
Constant METADOGE2._decimals (contracts/Token.sol#204) 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#18)" inContext (contracts/Token.sol#10-21)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

Variable IRouter01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountADesired (contracts/Token.sol#85) is too similar to IRouter01.addLiquidity(address,address
,uint256,uint256,uint256,uint256,address,uint256).amountBDesired (contracts/Token.sol#86)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar

METADOGE2.slitherConstructorConstantVariables() (contracts/Token.sol#154-394) uses literals with too many digits:
        - _totalSupply = 100000000 * 10 ** 9 (contracts/Token.sol#192)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

METADOGE2.lpPair (contracts/Token.sol#206) should be immutable
METADOGE2.swapRouter (contracts/Token.sol#201) 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/0x7ccd11ee8592d61a13aa9fad7c70e30ff16e57ff0e1796b04b7b5bb6534af382

2- Buying when excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0x244b38d6302738a466913ceffacbe069a8b5684b2aecda1a1177f8cfab17071e

3- Selling when excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0x2ca510c32952695990a4c8d12fceae2ed4abf898cd62c5088f4fe1efda6d3103

4- Transferring when excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0xd14bc462f790f6f5b96d46d1416b9877a0111580b1004b02da3f4b4c385683b9

5- Buying(0% tax) (passed):
https://testnet.bscscan.com/tx/0xe620c6963fb3ec2b38b94e4d8cc03b86b7b257d61ebedf61ec09a1a37b2fc209

6- Selling (5% tax) (passed):
https://testnet.bscscan.com/tx/0x962b38225c61462503066923 8e72fdaaa574e0a27c5a8916ac220a829da0ea7c

FUNCTIONAL TESTING

4- Transferring (0% tax) (passed):
https://testnet.bscscan.com/tx/0xfcaa7ecc43e4210b2e2ea9b92ce9946097c513687784f9c8c7d9a9a5b867866c

4- Internal swap (ETH sent to marketing wallet) (passed):
https://testnet.bscscan.com/address/0xC150d340dC05c73EBB35373dC5E06354bC6B95B3#internaltx

FUNCTIONAL TESTING

Centralization – Enabling Trades

Severity: **High**

function: enableTrading

Status: Not Resolved

Overview:

Owner of the contract must enable trades manually for investors, otherwise no one would be able to buy/sell/transfer their tokens.

```
function enableTrading() external onlyOwner {
    require(!isTradingEnabled, "Trading already enabled");
    isTradingEnabled = true;
    emit _enableTrading();
}
```

Suggestion

Its suggested to either enable trades prior to presale, or transfer ownership of the contract to a certified pinsksale safu developer to guearanee enabling of trades.

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