OXSCANS

TAO Inu

AI Generated at 09:10 PM, UTC

March 05, 2024

OVERVIEW

This audit has been perpared for 'TAO Inu' to review the main aspects of the project to help investors make an informative decision during their research process

You will find a summarized review of the following key points:

- Contract's source code
- Owner wallets
- Tokenomics
- Team transparency and goals
- Website's age, code, security and UX
- Whitepaper and roadmap
- Social media and online presence

Table of Content

General Information

TAO Inu

Name	TAO Inu
Category	Ethereum Ecosystem
Info	Website, Telegram Bot

General Information

Tokenomics

Ticker	0X4E9FCD48AF4738E3BF1382009DC1E93EBFCE698F
Network	Ethereum
Contract Address	0x4e9fcD48Af4738e3bF1382009dC1e93eBFCE698F

General Analysis

Audit Review Process

1. Testing the smart contracts against both common and uncommon vulnerabilities
2. Assessing the codebase to ensure compliance with current best practices and industry standards
3. Ensuring contract logic meets the specifications and intentions of the client
4. Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders
5. Thorough line-byline AI review of the entire codebase by industry

Token Transfer Stats

Transactions (Latest Mine Block)	Token holders	Compiler
1	2725	v0.8.21

Smart Contract Stats

Functions	Events	Constructor
78	21	1

Detail Analysis

Threat Level

High	Issues on this level are critical to the smart contract's performace/functionality and should be fixed before moving to a live enviroment
Medium	Issues on this level are critical to the smart contract's performace/functionality and should be fixed before moving to a live enviroment
Low	Issues on this level are minor details and warning that can remain unfixed
Informational	Informational level is to offer suggestions for improvement of efficacy or secruity for fratures with risk free factor

Threat Level

High	**0** threats found
Medium	**0** threats found
Low	**0** threats found
Informational	**0** threats found

Detail Analysis

Vulnerability Check

21 Passed

0 Fail

- Arbitrary Jump/Storage Write
- Centralization of Control
- Compiler Issues
- Delegate Call to Untrusted Contract
- Dependence on Predictable Variables
- Ether/Token Theft
- Flash Loans
- Front Running
- Improper Events
- Improper Authorization Scheme
- Integer Over/Underflow
- Logical Issues
- Oracle Issues
- Outdated Compiler Version
- Race Conditions
- Reentrancy
- Signature Issues
- Sybil Attack
- Unbounded Loops
- Unused Code
- Overall Contract Safety

Detail Analysis

Detail Analysis

21 Passed

0 Fail

CATEGORY	STATUS	NOTES
Arbitrary Jump/Storage Write		This contract does not contain low-level calls like delegatecall or callcode that could lead to arbitrary jumps or storage writes.
Centralization of Control		The contract employs a role-based access control mechanism, which mitigates risks of centralized control.
Compiler Issues		The contract is compiled with a recent and stable version of the Solidity compiler (v0.8.7), reducing the risk of known compiler issues.
Delegate Call to Untrusted Contract		The contract does not use delegate calls, thus eliminating risks associated with delegate calls to untrusted contracts.
Dependence on Predictable Variables		There are no critical functionalities in the contract that rely on variables like block.timestamp or block.number, which could be predictable.

Detail Analysis

Detail Analysis

21 Passed

0 Fail

CATEGORY	STATUS	NOTES
Ether/Token Theft		The contract's design and access control mechanisms do not expose functions that could lead to Ether or token theft.
Flash Loans		The contract does not interact with lending protocols and is not susceptible to flash loan attacks.
Front Running		There are no external calls or transactions that could be front-run in a harmful way to the contract or its users.
Improper Events		All external state-changing functions emit events, providing transparency and traceability.
Improper Authorization Scheme		The contract uses a robust role-based access control system, which is a proper authorization scheme for its context.
Integer Over/Underflow		With Solidity 0.8.x, arithmetic operations are checked by default, protecting against overflows and underflows.

Detail Analysis

Detail Analysis

21 Passed

0 Fail

CATEGORY	STATUS	NOTES
Logical Issues		The contract logic is consistent, with no apparent flaws that could lead to unintended behavior.
Oracle Issues		This contract does not use external data feeds or oracles.
Outdated Compiler Version		Compiled with Solidity v0.8.7, which is not outdated.
Race Conditions		There are no functions in the contract that are susceptible to race conditions.
Reentrancy		The contract uses the ReentrancyGuard modifier from OpenZeppelin, effectively preventing reentrancy attacks.
Signature Issues		The contract does not use Ethereum signatures (ECDSA) in a way that could be vulnerable.

Detail Analysis

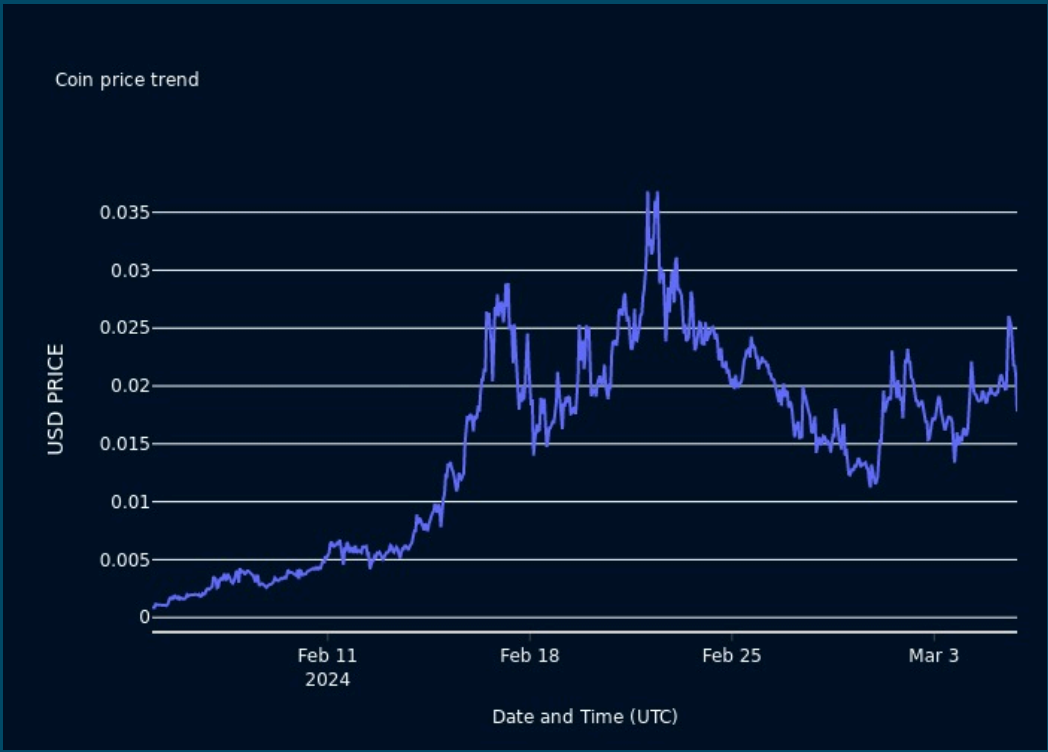
Detail Analysis

21 Passed

0 Fail

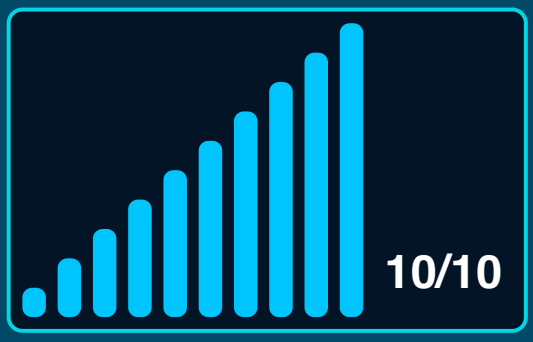
CATEGORY	STATUS	NOTES
Sybil Attack		The nature of this contract makes it not susceptible to Sybil attacks.
Unbounded Loops		Loop operations in the contract are bound and do not pose a risk of unbounded gas consumption.
Unused Code		The contract does not contain significant amounts of unused code.
Overall Contract Safety		Overall, the contract follows good practices, with no critical security issues detected.

Market Analysis

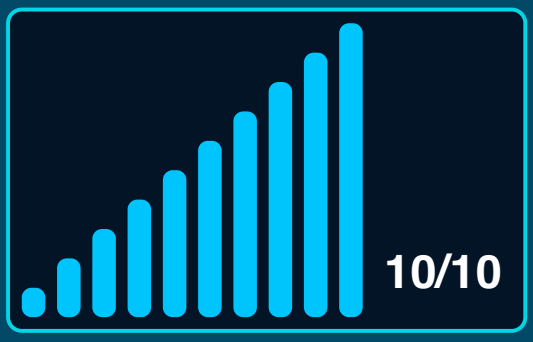


Score

Total Audit Score



Security Score



Legal Disclaimer

0xscans operates as an automated system for smart contract due diligence, acknowledging the possibility of bugs or vulnerabilities impacting token values. We do not hold specific obligations regarding your trading outcomes or the utilization of audit content. Users release 0xscans from any liability associated with content obtained through the tool.

AI generated by 0xscans AI technology

Chat with us

Telegram

For more information. Visit below:

Twitter

Github