OXSCANS

DeTensor

AI Generated at 09:39 AM, +0000

March 24, 2024

OVERVIEW

This audit has been perpared for 'DeTensor' 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

DeTensor

Name **DeTensor**

General Information

Tokenomics

Contract Address 0xe6f4a40156c9e8c7addda66848bbb99fdedecf84

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	561	v0.8.23

Smart Contract Stats

Functions	Events	Constructor
57	9	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	**4** threats found
Low	**2** threats found
Informational	**2** threats found

Detail Analysis

Vulnerability Check

15 Passed

6 Fail

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- Outdated Compiler Version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

Detail Analysis

Detail Analysis

15 Passed 6 Fail

CATEGORY	STATUS	NOTES
Reentrancy	●	No external calls that could lead to reentrancy attacks.
Flash Loans	●	Flash loan attack vectors not applicable, no external calls or token price dependencies.
Unused Code	●	Some code paths are not used in current contract logic but do not pose a risk.
Sybil Attack	●	While not directly vulnerable, ERC20 tokens can potentially be affected by sybil attacks in broader ecosystem.
Front Running	●	Some functions might be susceptible to front-running due to public visibility and transfer mechanics, although no direct financial risk observed.

Detail Analysis

Detail Analysis

15 Passed

6 Fail

CATEGORY	STATUS	NOTES
Oracle Issues		No external oracles or dependencies on off-chain data.
Logical Issues		Logic appears sound, but some functions require careful review to ensure they behave as intended.
Compiler Issues		Compiled with a recent Solidity version (v0.8.23) without known compiler issues.
Improper Events		All external state-changing functions emit appropriate events.
Race Conditions		Potential for race conditions in functions like 'transfer' and 'approve', common in ERC20 tokens.
Unbounded Loops		No unbounded loops that could lead to gas limit issues.

Detail Analysis

Detail Analysis

15 Passed | 6 Fail

CATEGORY	STATUS	NOTES
Signature Issues	●	No signature-based functionalities in the contract.
Ether/Token Theft	●	Standard ERC20 transfer mechanisms, no functions that could lead to Ether or token theft.
Integer Over/Underflow	●	Solidity 0.8.23 inherently protects against integer overflow and underflow.
Overall Contract Safety	●	While the contract follows common ERC20 patterns, some functions need careful review, and as the owner address is a dead address, centralization of control is not a concern.
Centralization of Control	●	No risk of centralization as the owner address is a dead address.
Outdated Compiler Version	●	Compiled with a recent and stable version of Solidity.

Detail Analysis

Detail Analysis

15 Passed

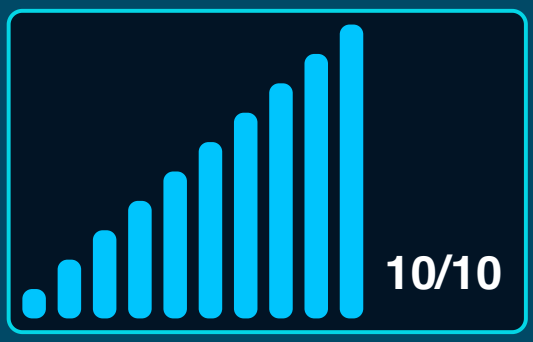
6 Fail

CATEGORY	STATUS	NOTES
Arbitrary Jump/Storage Write		No arbitrary jumps or storage writes detected, standard ERC20 and Ownable patterns used.
Improper Authorization Scheme		No risk of improper authorization scheme as the owner address is a dead address.
Delegate Call to Untrusted Contract		No delegate calls to external contracts present in the contract.
Dependence on Predictable Variables		No critical dependence on variables like block.timestamp or block.number.

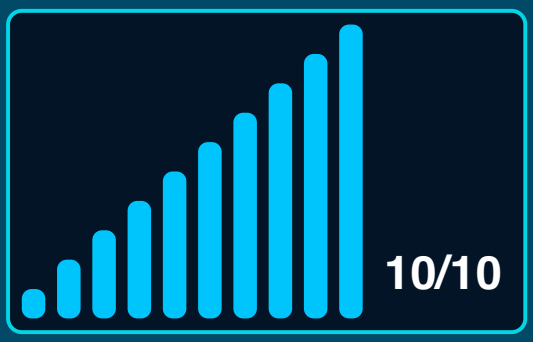
Market Analysis

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