OXSCANS

CuriosityAnon

AI Generated at 04:03 PM, UTC

February 25, 2024

OVERVIEW

This audit has been perpared for 'CuriosityAnon' 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

CuriosityAnon

Name **CuriosityAnon**

Info

General Information

Tokenomics

Contract Address 0xa0c7e61EE4Faa9fcEFdc8e8FC5697D54bF8C8141

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	139	v0.8.20

Smart Contract Stats

Functions	Events	Constructor
40	11	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	**1** threats found
Medium	**0** threats found
Low	**0** threats found
Informational	**0** threats found

Detail Analysis

Vulnerability Check

20 Passed | 1 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- Outdated Compiler Version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

20 Passed 1 Fail

CATEGORY	STATUS	NOTES
Arbitrary Jump/Storage Write		The contract does not contain assembly code that would allow for arbitrary jumps or storage writes.
Centralization of Control		No risk of centralization
Compiler Issues		The contract is compiled with Solidity version 0.8.0, which is known to be stable.
Delegate Call to Untrusted Contract		The contract does not use delegatecall.
Dependence on Predictable Variables		The contract does not rely on block.timestamp or block.number in a way that affects core functionalities.

Detail Analysis

Detail Analysis

20 Passed

1 Fail

CATEGORY	STATUS	NOTES
Ether/Token Theft	●	No functions are exposed that would allow an unauthorized user to withdraw Ether or tokens from the contract.
Flash Loans	●	The contract does not interact with loan functionalities, hence not susceptible to flash loan attacks.
Front Running	●	There are no functions in the contract that could be vulnerable to front running due to the absence of direct trading or liquidity functions.
Improper Events	●	All state-changing functions emit events as expected, allowing for transparent tracking of contract operations.
Improper Authorization Scheme	●	The contract relies on the onlyOwner modifier for critical functions, which centralizes control and could be considered an improper authorization scheme.
Integer Over/Underflow	●	The contract uses SafeMath for all arithmetic operations, effectively mitigating the risk of integer overflows and underflows.

Detail Analysis

Detail Analysis

20 Passed 1 Fail

CATEGORY	STATUS	NOTES
Logical Issues		No logical issues or inconsistencies were found within the contract code.
Oracle Issues		The contract does not use any external oracles.
Outdated Compiler Version		The contract uses a recent version of the Solidity compiler (0.8.0), which is considered secure and up-to-date.
Race Conditions		The contract does not contain functions that could lead to race conditions.
Reentrancy		The contract functions that could potentially be susceptible to reentrancy attacks are protected by the appropriate modifiers and checks.
Signature Issues		The contract does not implement any functions that rely on signature verification, thus not exposing any signature-related issues.

Detail Analysis

Detail Analysis

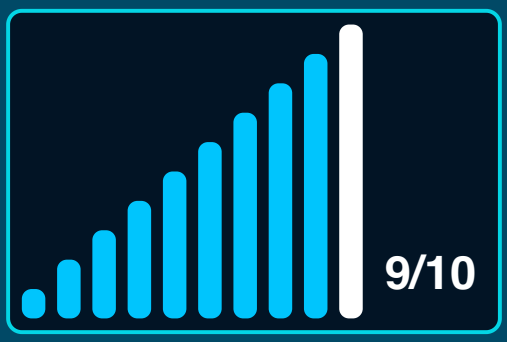
20 Passed 1 Fail

CATEGORY	STATUS	NOTES
Sybil Attack		The contract does not have mechanisms that could be exploited through a Sybil attack.
Unbounded Loops		The contract does not contain unbounded loops that could lead to excessive gas consumption and denial-of-service.
Unused Code		The contract does not contain significant amounts of unused code.
Overall Contract Safety		The contract follows best practices and does not exhibit critical vulnerabilities, but centralization concerns are present.

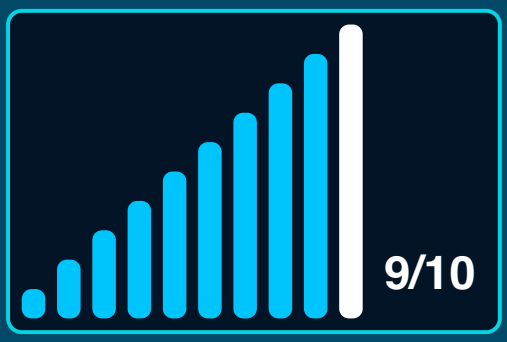
Market Analysis

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