OXSCANS

Syntax AI Node

AI Generated at 05:29 PM, UTC

March 03, 2024

OVERVIEW

This audit has been perpared for 'Syntax AI Node' 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

Syntax AI Node

Name	**Syntax AI Node**

Info

General Information

Tokenomics

Contract Address	0xE2870Ad60442bd6f5634CA2E00a1Eb23cEA9786e

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	0	v0.8.20

Smart Contract Stats

Functions	Events	Constructor
39	4	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	1 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- 🟢 Outdated Compiler Version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	🟢	No arbitrary jumps or storage writes detected; the contract uses standard, well-audited OpenZeppelin libraries.
Centralization of Control	🔴	The contract implements the 'Ownable' pattern, offering centralized control to the owner, which can be a single point of failure or malicious control.
Compiler Issues	🟢	Compiled with a recent and stable version of the Solidity compiler (v0.8.20).
Delegate Call to Untrusted Contract	🟢	The contract does not use delegatecall to untrusted contracts.
Dependence on Predictable Variables	🟢	No dependence on block variables like block.timestamp or block.number that could be manipulated by miners.

Detail Analysis

Detail Analysis 🟢 20 Passed 🔴 1 Fail

CATEGORY	STATUS	NOTES
Ether/Token Theft	🟢	No functions exist that could lead to Ether or token theft. Standard ERC20 functions are properly implemented.
Flash Loans	🟢	Flash loan attacks are not relevant to this contract as it does not have functions that interact with loan mechanisms.
Front Running	🟢	Front running is not a concern for this contract’s main functionalities.
Improper Events	🟢	All external state-changing functions emit proper events.
Improper Authorization Scheme	🟢	Uses a standard and secure authorization scheme with 'onlyOwner' modifiers where necessary.
Integer Over/Underflow	🟢	SafeMath library is used to prevent overflows/underflows.

Detail Analysis

Detail Analysis 🟢 20 Passed 🔴 1 Fail

CATEGORY	STATUS	NOTES
Logical Issues	🟢	No logical issues detected; the contract follows standard ERC721 and Ownable logic.
Oracle Issues	🟢	The contract does not interact with oracles.
Outdated Compiler Version	🟢	Compiler version is not outdated for the contract's deployment context.
Race Conditions	🟢	No race conditions detected due to proper state management.
Reentrancy	🟢	No external calls that could lead to reentrancy attacks.
Signature Issues	🟢	Contract does not use signature verification mechanisms.

Detail Analysis

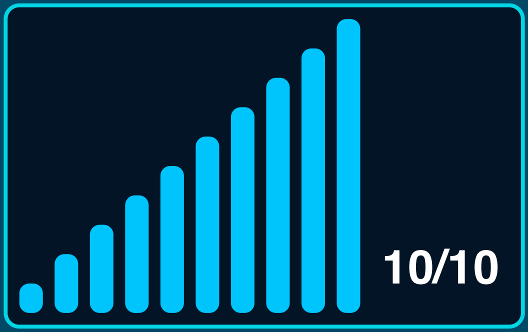
Detail Analysis 🟢 20 Passed 🔴 1 Fail

CATEGORY	STATUS	NOTES
Sybil Attack	🟢	Sybil attacks are not relevant to this contract.
Unbounded Loops	🟢	No unbounded loops that could lead to gas limit issues.
Unused Code	🟢	No significant chunks of unused code found.
Overall Contract Safety	🟢	Contract adheres to standard practices and follows common patterns for security and functionality.

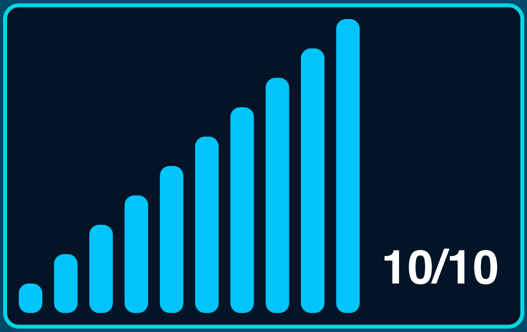
Market Analysis

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