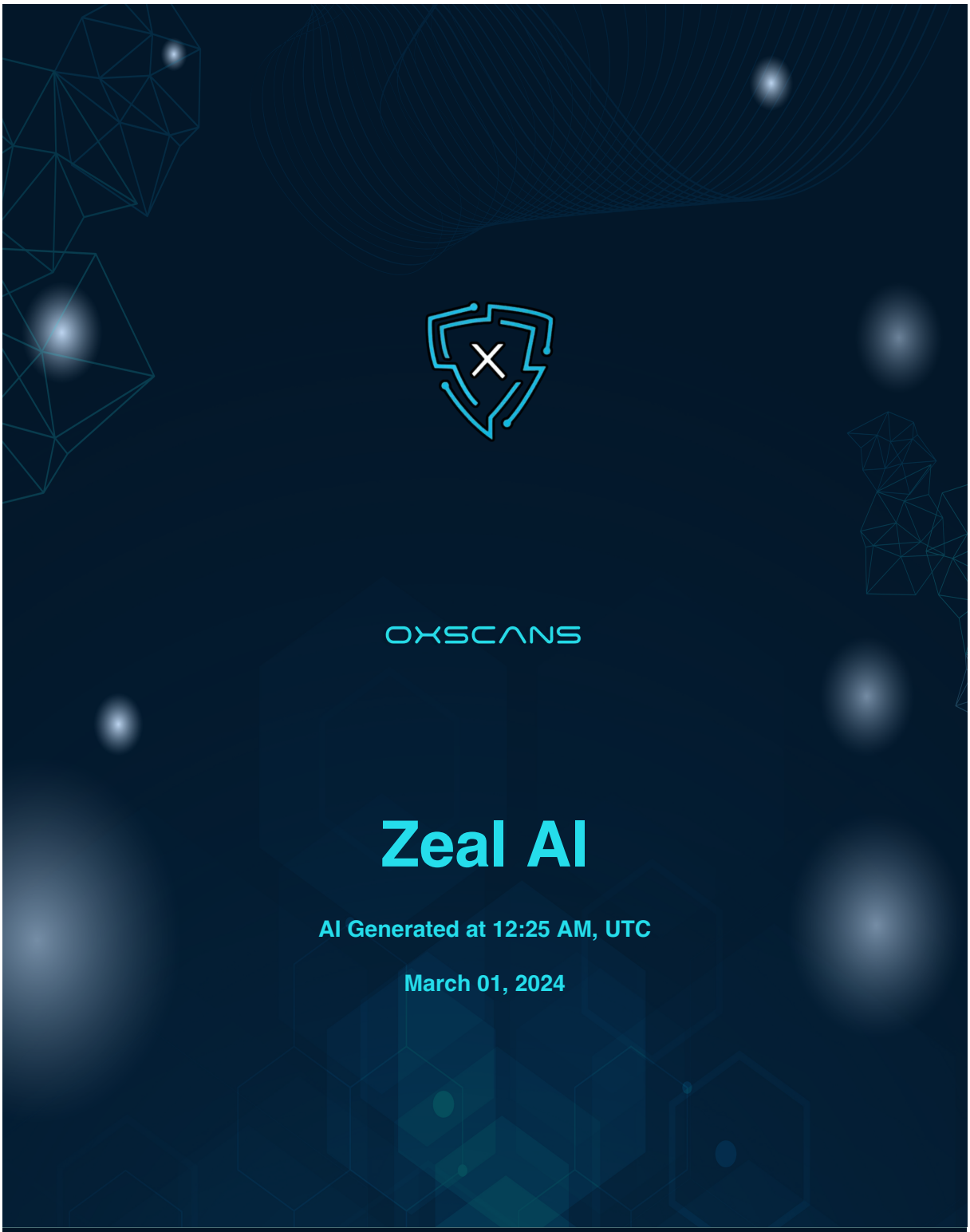


OVERVIEW

This audit has been perpared for 'Zeal AI' 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

Zeal AI

Name	Zeal AI
Category	Ethereum Ecosystem
Info	Website Telegram Bot

General Information

Tokenomics

Ticker	0XC2AEEDC081D4CB6797A681E9403A82211F97B308
Network	Ethereum
Contract Address	0xc2aeedc081d4cb6797a681e9403a82211f97b308

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	778	v0.8.19

Smart Contract Stats

Functions	Events	Constructor
20	6	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

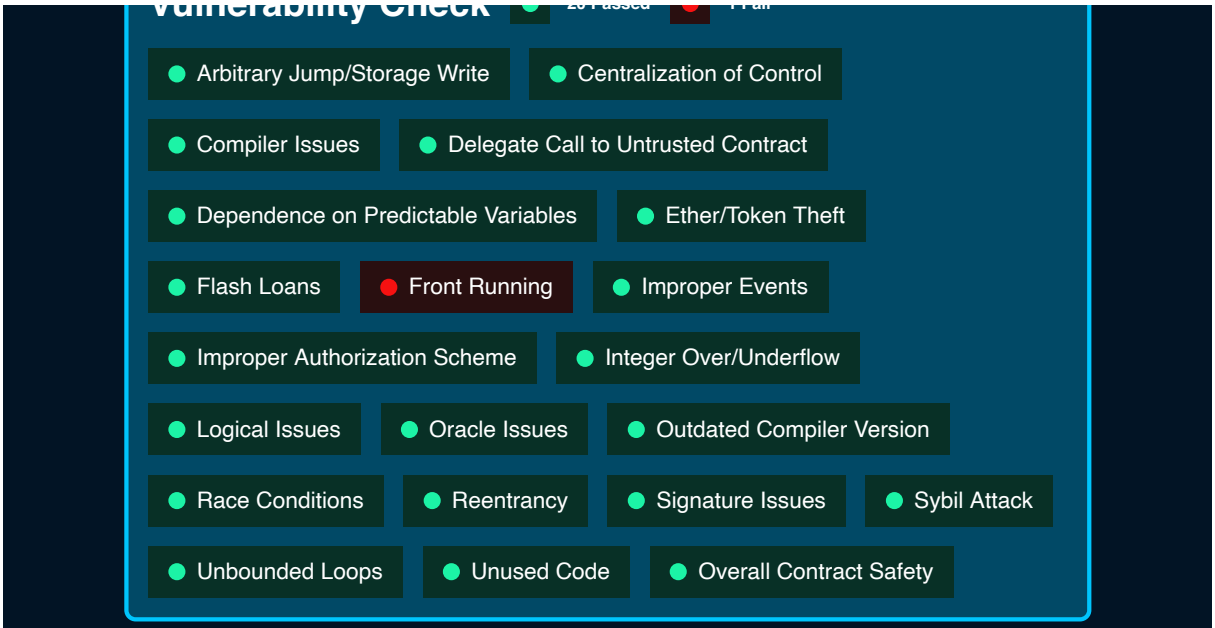
Threat Level

High	**1** threats found
Medium	**0** threats found
Low	**0** threats found
Informational	**0** threats found

Detail Analysis

Vulnerability Check

20 Passed 1 Fail



Detail Analysis

Detail Analysis

20 Passed 1 Fail

CATEGORY	STATUS	NOTES
Arbitrary Jump/Storage Write		The contract does not exhibit arbitrary jumps or storage writes, as it adheres to standard Solidity development patterns.

Centralization of Control	●	No risk of centralization as the contract owner is a dead address, eliminating the potential for owner-based manipulation.
Compiler Issues	●	Compiled with a recent Solidity version (0.8.19) with no known compiler issues.
Delegate Call to Untrusted Contract	●	There is no use of delegatecall to an untrusted contract, mitigating risks associated with delegate calls.
Dependence on Predictable Variables	●	The contract does not rely on variables like block.timestamp or block.number in a way that affects core functionalities or security.

Detail Analysis

Detail Analysis ● 20 Passed ● 1 Fail

CATEGORY	STATUS	NOTES
Ether/Token Theft	●	No functions are present that directly transfer Ether or tokens to arbitrary addresses in an unauthorized manner.
Flash Loans	●	The contract does not interact with flash loan functions, making it unaffected by flash loan attacks.
Front Running	●	The contract may be susceptible to front running, as it does not appear to have mechanisms such as max transaction amount or other protections.

Improper Events	●	All critical functions emit events correctly, providing transparency and traceability.
Improper Authorization Scheme	●	The contract uses an ownership model with renounceOwnership function, and since the owner is a dead address, there is no risk of improper authorization.
Integer Over/Underflow	●	SafeMath library is used consistently for arithmetic operations, mitigating risks of overflows and underflows.

Detail Analysis

Detail Analysis ● 20 Passed ● 1 Fail

CATEGORY	STATUS	NOTES
Logical Issues	●	No apparent logical issues or inconsistencies in the contract logic.
Oracle Issues	●	The contract does not interact with oracles, thus not exposing it to oracle-related risks.
Outdated Compiler Version	●	The contract uses a recent Solidity compiler version (0.8.19), which is not outdated.
Race Conditions	●	No functions or patterns were found that could lead to race conditions.
Reentrancy	●	The contract's functions are structured in a way that avoids reentrancy vulnerabilities.
Signature Issues	●	The contract does not rely on external signatures, hence is not exposed to signature-related risks.

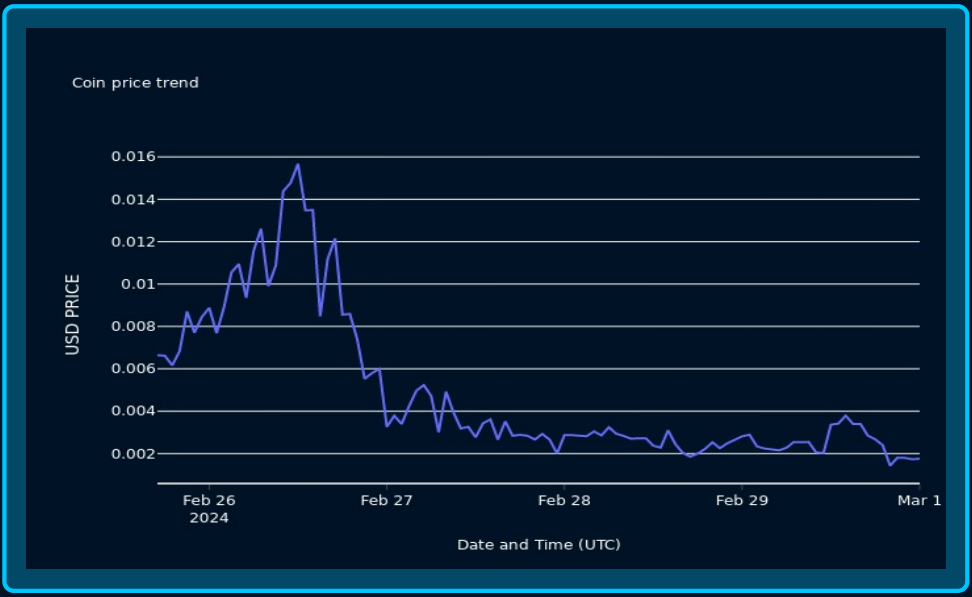
Detail Analysis

Detail Analysis

20 Passed 1 Fail

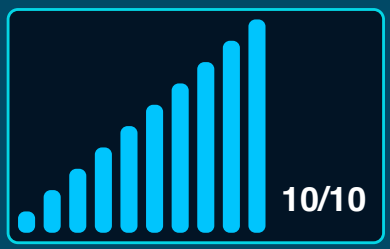
CATEGORY	STATUS	NOTES
Sybil Attack		The nature of the contract does not make it susceptible to Sybil attacks.
Unbounded Loops		All loops in the contract have bounded conditions, avoiding risks of gas limit issues or denial-of-service.
Unused Code		The contract's code does not contain redundant or unused code, ensuring efficiency and reducing attack surface.
Overall Contract Safety		The contract follows general best practices and does not exhibit critical vulnerabilities, though improvements are suggested in front running protection.

Market Analysis

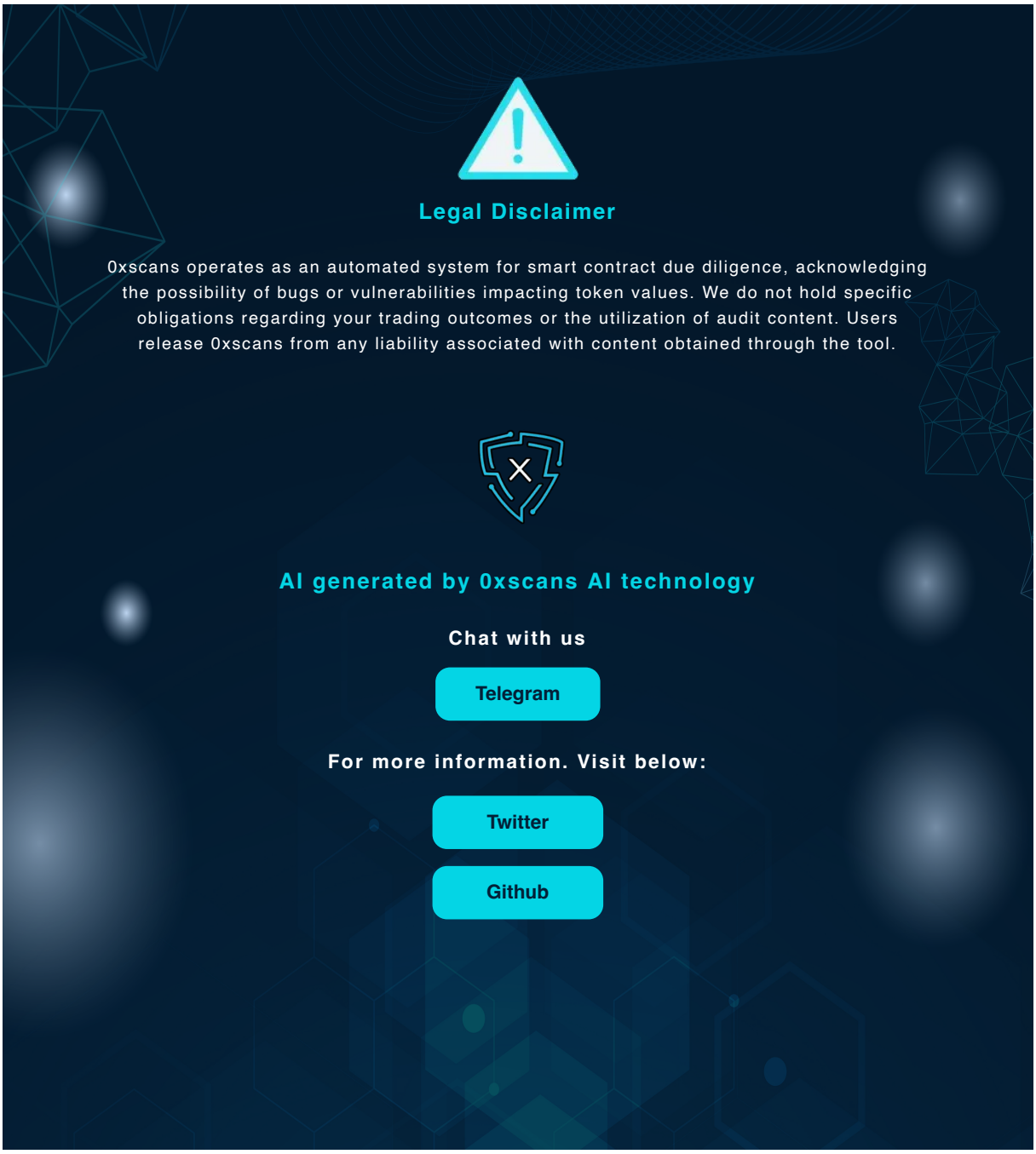


Score

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