OXSCANS

Alea

AI Generated at 05:28 PM, UTC

February 13, 2024

OVERVIEW

This audit has been perpared for 'Alea' 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

Alea

Name	Alea
Category	Ethereum Ecosystem
Info	Website, Telegram Bot

General Information

Tokenomics

Ticker	0X24BFF4FE25B5807BAD49B2C08D79BB271766E68A
Network	Ethereum
Contract Address	0x24bff4fe25b5807bad49b2c08d79bb271766e68a

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	451	v0.8.8

Smart Contract Stats

Functions	Events	Constructor
49	3	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- Outdated Compiler Version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		No arbitrary jump or storage write detected in the contract code.
Centralization of Control		The contract uses an Ownable pattern, which is a common practice to allow certain functions to be called by the owner only, reducing centralization risks.
Compiler Issues		Compiled with a recent version of the compiler (v0.8.8) and no optimization issues detected.
Delegate Call to Untrusted Contract		No delegatecall to untrusted contracts found in the contract.
Dependence on Predictable Variables		The contract does use block.timestamp for setting sell timestamps, but this does not lead to a critical vulnerability.

Detail Analysis

Detail Analysis

21 Passed

0 Fail

CATEGORY	STATUS	NOTES
Ether/Token Theft		No functions found that could allow Ether/Token theft; transfer functions are standard and secure.
Flash Loans		Flash loan attack vectors are not applicable to this contract as it does not have functions that could be manipulated via flash loans.
Front Running		No obvious front-running vulnerabilities detected; however, the use of block.timestamp could potentially be manipulated by miners to a small extent.
Improper Events		All external and sensitive internal transactions emit events properly.
Improper Authorization Scheme		Authorization is properly managed with governance controls using the Ownable pattern.
Integer Over/Underflow		The contract is using Solidity 0.8.x, which has built-in overflow/underflow protection.

Detail Analysis

Detail Analysis

21 Passed

0 Fail

CATEGORY	STATUS	NOTES
Logical Issues		No logical issues detected in the contract code.
Oracle Issues		No oracle is used in the contract, hence no related issues.
Outdated Compiler Version		Uses a recent version of the Solidity compiler.
Race Conditions		No functions in the contract are susceptible to race conditions.
Reentrancy		ReentrancyGuard is not used, but the contract does not appear to have reentrancy vulnerabilities due to the use of a reentrant lock modifier.
Signature Issues		No signature-related issues found as the contract does not use ecrecover or similar functions.

Detail Analysis

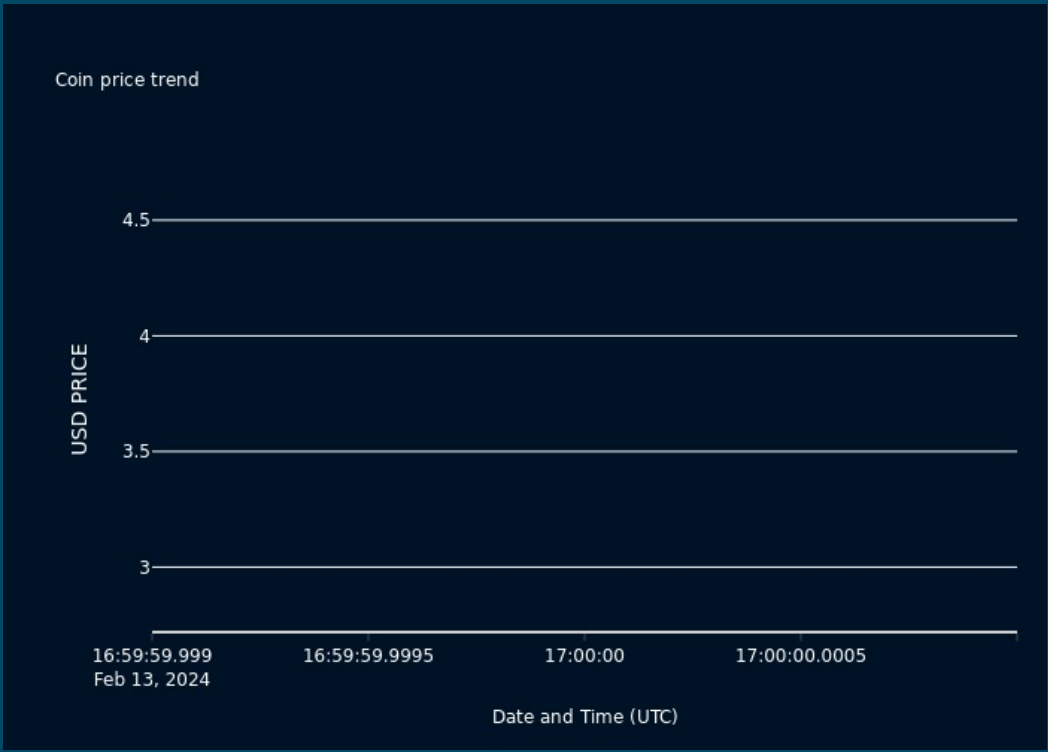
Detail Analysis

21 Passed

0 Fail

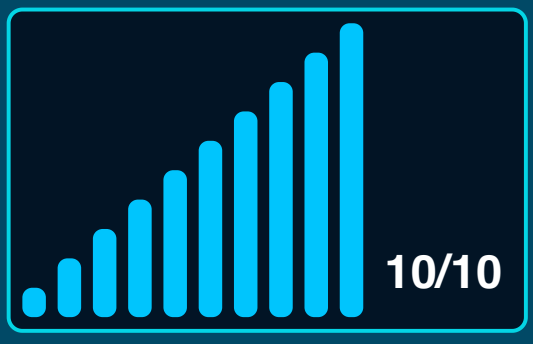
CATEGORY	STATUS	NOTES
Sybil Attack		No susceptibility to Sybil attacks as the contract does not interact with untrusted external contracts in a way that would enable this attack.
Unbounded Loops		No unbounded loops that could lead to gas limit issues.
Unused Code		No significant amount of unused code present in the contract.
Overall Contract Safety		The contract follows general best practices and no critical vulnerabilities were found.

Market Analysis

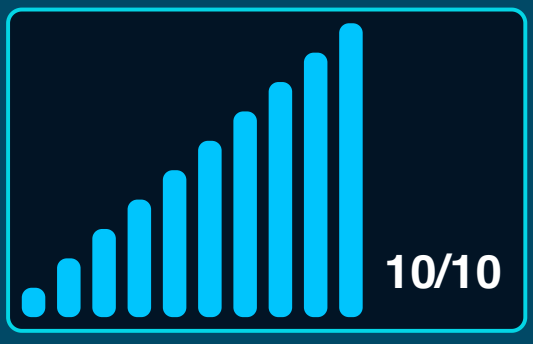


Score

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