OXSCANS

MindVerse

AI Generated at 05:38 PM, +0000

March 17, 2024

OVERVIEW

This audit has been perpared for 'MindVerse' 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

MindVerse

Name	**MindVerse**
Category	**Ethereum Ecosystem**
Info	Website Telegram Bot

General Information

Tokenomics

Ticker	0XF67366E83CC9B115EF8CCA93BAED1F03E6D3CA9A
Network	Ethereum
Contract Address	0xf67366e83cc9b115ef8cca93baed1f03e6d3ca9a

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	1020	v0.8.23

Smart Contract Stats

Functions	Events	Constructor
53	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	**0** threats found
Low	**0** threats found
Informational	**0** threats found

Detail Analysis

Vulnerability Check

21 Passed

0 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- Outdated Compiler Version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

21 Passed

0 Fail

CATEGORY	STATUS	NOTES
Reentrancy		ReentrancyGuard is not used, but the contract does not appear to be vulnerable to reentrancy attacks due to the atomic nature of operations.
Flash Loans		The contract does not interact with or utilize flash loans.
Unused Code		No unused code found, all contract code is utilized and has a purpose.
Sybil Attack		Contract is not vulnerable to Sybil attacks since it does not rely on node or user reputation.
Front Running		No apparent opportunities for front running identified within the contract.

Detail Analysis

Detail Analysis

21 Passed

0 Fail

CATEGORY	STATUS	NOTES
Oracle Issues		The contract does not use external oracles.
Logical Issues		No apparent logical issues found in the contract's execution flow.
Compiler Issues		Compiled with a recent Solidity version (0.8.23) with no evident compiler-specific issues.
Improper Events		All external and public functions that alter state emit appropriate events.
Race Conditions		No identifiable race conditions within the contract's logic.
Unbounded Loops		Looping is either not present or contains bounds within functions, preventing excessive gas costs or denial of service.

Detail Analysis

Detail Analysis

21 Passed

0 Fail

CATEGORY	STATUS	NOTES
Signature Issues	●	The contract does not utilize signature verification in its logic.
Ether/Token Theft	●	No functions are present that directly transfer Ether or tokens to arbitrary addresses in an unauthorized manner.
Integer Over/Underflow	●	Uses Solidity 0.8.x, which has built-in overflow/underflow checks.
Overall Contract Safety	●	The contract follows general best practices and does not exhibit critical vulnerabilities.
Centralization of Control	●	No risk of centralization since the owner is a dead address.
Outdated Compiler Version	●	The contract uses a recent Solidity compiler version (0.8.23).

Detail Analysis

Detail Analysis

21 Passed

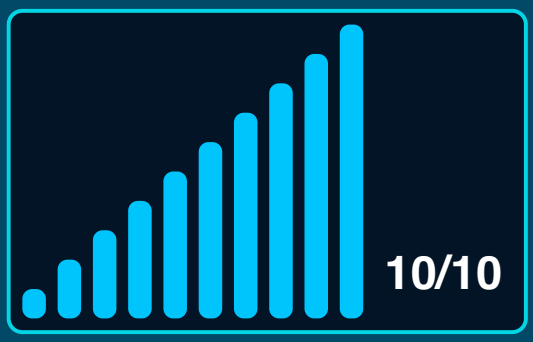
0 Fail

CATEGORY	STATUS	NOTES
Arbitrary Jump/Storage Write		The contract does not exhibit arbitrary jumps or storage writes, as it adheres to standard ERC-20 development patterns.
Improper Authorization Scheme		Authorization is handled using the standard OpenZeppelin Ownable pattern; however, since the owner is a dead address, control is decentralized.
Delegate Call to Untrusted Contract		Contract does not make delegate calls to untrusted contracts.
Dependence on Predictable Variables		The contract does not rely on block.timestamp or block.number for critical functionalities.

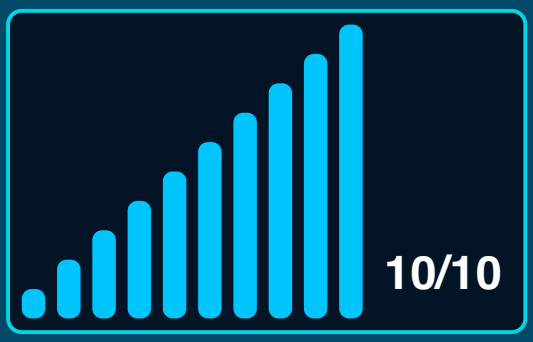
Market Analysis

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