HACKEN Request an audit

Mind AI

Audit name:
[SCA] Mind AI / Token / Jan2025

Date:
Jan 29, 2025

Table of Content

View Report

Introduction

We express our gratitude to the Mind AI team for the collaborative engagement that enabled the execution of this Smart Contract Security Assessment.

Mind AI is a solution to revolutionize crypto investment decisions by making tools and data accessible to everyone.

Document

Name
Smart Contract Code Review and Security Analysis Report for Mind AI

Audited By
Nataliia Balashova

Approved By
Grzegorz Trawinski

Website
https://www.mind-ai.io

Changelog
30/01/2025 – Final Report

Platform
EVM

Language
Solidity

Tags
ERC20, ERC20Burnable

Methodology
https://hackenio.cc/sc_methodology

Review Scope

Repository
https://github.com/Decubate-com/smart-contracts

Commit
ce9e9131d94ec79275f847a8242494d4d15bd9e3

View Full Scope →

Audit Summary

0

Total Findings

0
Resolved

0
Accepted

0
Mitigated

The system users should acknowledge all the risks summed up in the risks section of the report

Documentation quality

- Technical description is provided.

Code quality

- The code is an implementation of the ERC20 token with burnable functionality.
- The development environment is configured.

Test coverage

- Tests are not provided.

System Overview

The Mind-AI (MA) contract is an implementation of an ERC-20 token with burnable functionality.

MA — is a ERC-20 token that mints all initial supply to a hardcoded address `0xB66C1027D2eb89E386C44019cCb129FBcC727f09` on deployment.

This contract broadens the ERC20 token standard and additionally implements the ERC20Burnable extension, enabling the token holders to burn their tokens to reduce the total supply.

Additional minting is not allowed.

Mind-AI token has the following attributes:

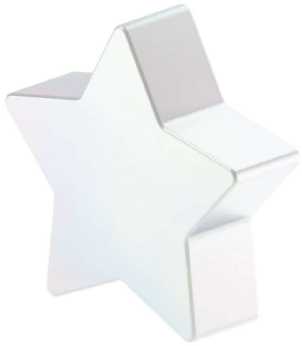
- Name: Mind-AI.
- Symbol: MA.
- Decimals: 18.

- Total supply: 500m tokens.

Potential Risks

The contract allows burning but not additional minting, limiting flexibility if new tokens are needed in the future and increasing the risk of supply issues.

Findings



No vulnerabilities were found

Appendix 1. Definitions

Severities

When auditing smart contracts, Hacken is using a risk-based approach that considers **Likelihood**, **Impact**, **Exploitability** and **Complexity** metrics to evaluate findings and score severities.

Reference on how risk scoring is done is available through the repository in our Github organization:

hknio/severity-formula

Severity

Critical

Description

Critical vulnerabilities are usually straightforward to exploit and can lead to the loss of user funds or contract state manipulation.

Severity

High

Description

High vulnerabilities are usually harder to exploit, requiring specific conditions, or have a more limited scope, but can still lead to the loss of user funds or contract state manipulation.

Severity

Medium

Description

Medium vulnerabilities are usually limited to state manipulations and, in most cases, cannot lead to asset loss. Contradictions and requirements violations. Major deviations from best practices are also in this category.

Severity

Low

Description

Major deviations from best practices or major Gas inefficiency. These issues will not have a significant impact on code execution.

Potential Risks

The "Potential Risks" section identifies issues that are not direct security vulnerabilities but could still affect the project's performance, reliability, or user trust. These risks arise from design choices, architectural decisions, or operational practices that, while not immediately exploitable, may lead to problems under certain conditions. Additionally, potential risks can impact the quality of the audit itself, as they may involve external factors or components beyond the scope of the audit, leading to incomplete assessments or oversight of key areas. This section aims to provide a broader perspective on factors that could affect the project's long-term security, functionality, and the comprehensiveness of the audit findings.

Appendix 2. Scope

The scope of the project includes the following smart contracts from the provided repository:

Scope Details

Deployed Address

https://bscscan.com/token/0xb5ab5cf2e2c686ae43f01f23859f3a55a8629c1c?a=0xb66c1027d2eb89e386c44019ccb129fbcc727f09

Repository

https://github.com/Decubate-com/smart-contracts

Commit

ce9e9131d94ec79275f847a8242494d4d4d15bd9e3

Whitepaper

-

Requirements

-

Technical Requirements

-

Assets in Scope

MA.sol - `MA.sol`

Disclaimer

Hacken Disclaimer

Technical Disclaimer