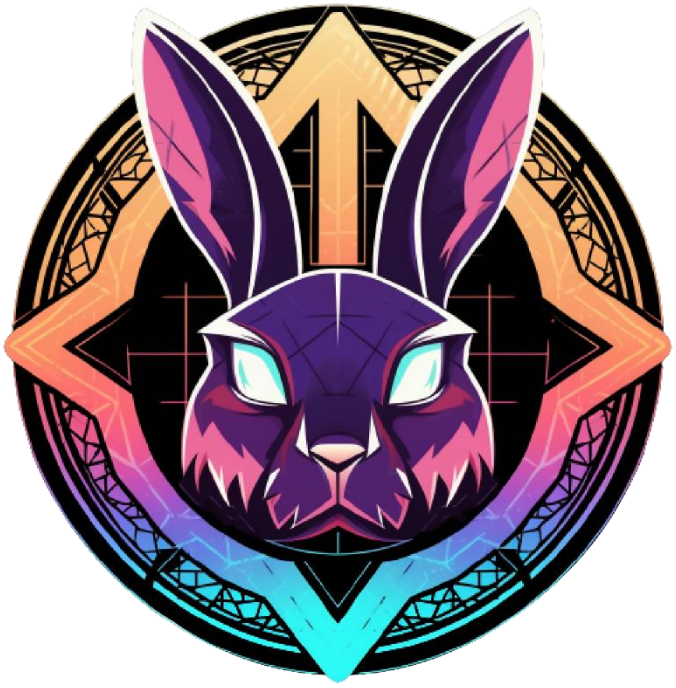
CONTRACT CHECKER
Blockchain Solutions

https://t.me/contractchecker
contact@contractchecker.app
contractchecker.app
Anywhere on the Blockchain

Date: 22.08.2023

Smart Contract Security Audit

ELVISH MAGIC TOKEN



Harry K

Harry Kedelman
General Manager

Project Information

Elvish Magic is a visionary crypto asset project that sets its sights on the creation of an innovative metaverse launchpad, designed to revolutionize the way they incubate and launch metaverse ventures. This groundbreaking initiative aims to seamlessly connect creators, investors, and explorers, providing a comprehensive ecosystem that nurtures metaverse concepts from inception to fruition.

Token Name	Elvish Magic 2.0 Token
Symbol	EMAGIC
Web Site	https://elvishmagic.com/
Twitter	https://twitter.com/elvishmagicpad/
Telegram	https://t.me/elvishmagic
Medium	https://medium.com/@elvishmagic
Whitepaper	https://elvishmagic.com/elvishmagicWhitepaper.pdf
Platform	Binance Smart Chain
Token Type	BEP20
Language	Solidity
Contract Address	0x889563Cb0b549FcC73fFC8E2684d9d64E4dbd4EB
Contract Link	https://bscscan.com/token/0x889563cb0b549fcc73ffc8e2684d9d64e4dbd4eb

Audit Result

Elvish Magic 2.0 Token has PASSED the smart contract audit with below privileges

(Other unknown security vulnerabilities are not included in the audit responsibility scope)

Audit Result:	PASSED
Ownership:	Not renounced yet
KYC Verification:	NA at the date of report edition
Audit Date:	August 22, 2023
Audit Team:	CONTRACTCHECKER

Findings

Privileges

- Contract has 3% buy, 6% sell and 2% transfer fee which is hardcoded and cannot be changed by owner
- Owner can remove the fees and cannot restore back
- Owner can exclude any account from fees
- Owner can claim stuck tokens
- Trading must be enabled by the owner
- Owner can change swap settings

Important Notice for Investors

As the ContractChecker team, our primary objective is to conduct a comprehensive audit of the contract code to assess its functionality and identify any potential risks embedded within the code.

Before making any investment decisions, it is crucial to consider several factors. These include the ownership status, approach of the project team, marketing strategies, general market conditions, liquidity, token holdings, and other relevant aspects.

Investors should always exercise due diligence by conducting their own research and carefully managing their risk, considering the various factors that can impact the success of a project.

CONTRACT CHECKER

Table of Contents

Project Information 1
Audit Result 2
Findings 2
  Privileges 2
Important Notice for Investors 2
Executive Summary 4
Overview 5
  Applied Methodology 5
  Security Assessment 6
  Sound Architecture 6
  Code Correctness and Quality 6
Risk Classification 7
  High level vulnerability 7
  Medium level vulnerability 7
  Low level vulnerability 7
Manual Audit: 7
Automated Audit 7
  Remix Compiler Warnings 7
Disclaimer 8

Executive Summary

CONTRACTCHECKER received an application on August 21, 2023, from the project team of Elvish Magic 2.0 Token to perform a thorough smart contract security audit. The objective was to identify any vulnerabilities present in the source code of Elvish Magic 2.0 Token, as well as any dependencies within the contract. The audit process employed a combination of Static Analysis and Manual Review techniques, conducted by our expert team.

The audit primarily focused on the following considerations:

- Functionality Testing: The Smart Contract was subjected to rigorous testing to assess if the intended logic was followed consistently throughout the entire process.
- Line-by-Line Manual Examination: Our experts meticulously reviewed the code, examining each line in detail to identify any potential issues or vulnerabilities.
- Live Testing with Multiple Clients: The Smart Contract underwent live testing using a Testnet environment, involving multiple clients. This allowed for real-world usage scenarios to be simulated and evaluated.
- Failure Analysis: The preparations for potential failures were analyzed to determine how the Smart Contract would perform in the event of bugs or vulnerabilities.
- Library Version Analysis: The versions of all libraries utilized in the code were scrutinized to ensure they were up to date, reducing the risk of known vulnerabilities.
- On-chain Data Security Analysis: The security of on-chain data was thoroughly assessed to identify any weaknesses or potential risks associated with data storage and handling.

Furthermore, as part of the smart contract security audit, CONTRACTCHECKER conducted an in-depth review of the Elvish Magic 2.0 Token project, focusing on various key aspects. These included access control and authorization, input validation and sanitization, gas optimization and efficiency, event logging and error handling, as well as compliance with best practices and standards.

Through these comprehensive auditing procedures, CONTRACTCHECKER aimed to provide a detailed evaluation of the Elvish Magic 2.0 Token's smart contract security, highlighting any vulnerabilities and recommending appropriate measures for mitigation.

Overview

This audit report provides a comprehensive assessment of the overall security of the Elvish Magic 2.0 Token smart contract. ContractChecker, a trusted security auditing firm, has conducted a thorough analysis of the contract, evaluating the system architecture and codebase to identify potential vulnerabilities, exploitations, hacks, and backdoors. The objective of this audit is to ensure the reliability, correctness, and robustness of the Elvish Magic 2.0 Token smart contract.

Applied Methodology

The audit process employed by Contract Checker followed industry-leading practices and methodologies. Our expert team utilized a comprehensive approach, including the following key elements:

- Code Design Pattern Analysis: We conducted a thorough analysis of the code design patterns used in the smart contract. This analysis helped identify any design flaws or architectural weaknesses that could potentially impact the contract's security.
- Line-by-Line Inspection: Our experienced auditors performed a meticulous review of the smart contract's code, examining each line in detail. This process aimed to identify any coding errors, vulnerabilities, or potential security risks that could compromise the contract's integrity.
- Unit Testing Phase: We executed a robust unit testing phase, where specific units of the smart contract were tested individually to ensure their functionality and resilience. This phase helped identify and address any functional issues or discrepancies within the contract.
- Automated Testing: Contract Checker employed automated testing techniques to complement the manual review process. Automated tests were designed to simulate various scenarios and interactions with the contract, helping to identify potential vulnerabilities or weaknesses that may not be apparent through manual inspection alone.

These elements were integrated into our methodology to ensure a comprehensive assessment of the smart contract's security. By combining manual inspection, unit testing, and automated testing, we aimed to provide a holistic evaluation of the contract's reliability and correctness.

Security Assessment

During the assessment, Contract Checker scrutinized the smart contract for various security aspects, including but not limited to:

- Vulnerability Identification: We conducted a comprehensive scan of the contract codebase to identify any potential vulnerabilities that could expose the contract to unauthorized access, manipulation, or exploitation.

- Exploitation Analysis: Our team performed rigorous testing and analysis to assess the contract's resistance to common exploitation techniques, ensuring its robustness against potential attacks.

- Backdoor Detection: We meticulously examined the codebase to identify any backdoors or hidden functionalities that could compromise the security and integrity of the smart contract.

Sound Architecture

The smart contract incorporates a robust and efficient architecture. It follows a modular design, separating functionalities into distinct modules for reusability and scalability. Access control mechanisms ensure authorized operations, while optimized data structures minimize storage costs. The event-driven architecture enables real-time communication, and upgradeability allows for future enhancements. The contract's sound architecture reflects best practices, ensuring secure and efficient operations.

Note: While keeping the content concise, it's important to maintain clarity and ensure that essential information is conveyed effectively.

Code Correctness and Quality

The smart contract underwent a comprehensive review of its source code, with a primary focus on accuracy, readability, sections of high complexity, and the quantity and quality of test coverage. Contract Checker conducted a thorough assessment to ensure that the code is error-free, easily understandable, and properly tested. By examining these critical areas, the contract's overall code correctness and quality were evaluated, providing assurance of reliable and robust execution.

Risk Classification

Vulnerabilities are classified in 3 main levels as below based on possible effect to the contract.

High level vulnerability

Vulnerabilities on this level must be fixed immediately as they might lead to fund and data loss and open to manipulation. Any High-level finding will be highlighted with **RED** text

Medium level vulnerability

Vulnerabilities on this level also important to fix as they have potential risk of future exploit and manipulation. Any Medium-level finding will be highlighted with **ORANGE** text

Low level vulnerability

Vulnerabilities on this level are minor and may not affect the smart contract execution. Any Low-level finding will be highlighted with **BLUE** text

Manual Audit:

In the manual audit phase, our developers conducted a comprehensive line-by-line examination of the code. This meticulous process involved a thorough review of each line to identify any potential issues or vulnerabilities. To further validate the contract's functionality, we utilized Remix IDE's JavaScript VM and Kovan networks for testing. By combining manual code inspection with real-world testing environments, we aimed to ensure the accuracy and effectiveness of the smart contract.

Automated Audit

Remix Compiler Warnings

During the audit, the smart contract was tested using Solidity's compiler in Remix. While conducting the analysis, we found no issues or warnings reported by the compiler. This indicates that the contract's codebase complied with Solidity's syntax and best practices, further ensuring the contract's integrity and reliability.

Disclaimer

This report provides a limited overview of our findings based on our analysis, in accordance with industry best practices at the time of this report, regarding cybersecurity vulnerabilities and issues in the smart contract framework and algorithms, as detailed within this report. It is essential for you to read the full report to obtain a comprehensive understanding of our analysis. While we have conducted our analysis and prepared this report to the best of our abilities, it is important to note that you should not solely rely on this report and cannot hold us liable based on its content, production, or lack thereof. It is crucial for you to conduct your own independent investigations before making any decisions. We provide further details and clarification in the following disclaimer, which we advise you to read in its entirety.

DISCLAIMER: By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to these terms, please discontinue reading this report, delete and destroy all downloaded and/or printed copies. This report is for informational purposes only and should not be relied upon for investment advice. No party shall have the right to rely on this report or its contents. ContractChecker and its affiliates, including holding companies, shareholders, subsidiaries, employees, directors, officers, and representatives (collectively referred to as "ContractChecker"), owe no duty of care to you or any other person and make no warranty or representation regarding the accuracy or completeness of the report. The report is provided on an "as is" basis, without any conditions, warranties, or other terms, except as explicitly stated in this disclaimer. ContractChecker hereby excludes all representations, warranties, conditions, and other terms that might have effect, but for this clause, in relation to the report, including, but not limited to, warranties of satisfactory quality, fitness for a particular purpose, and the use of reasonable care and skill. Except to the extent prohibited by law, ContractChecker excludes all liability and responsibility and disclaims any claims for any kind of loss or damage that may result from the use of this report, including, but not limited to, direct, indirect, special, punitive, consequential, or purely economic loss or damages, loss of income, profits, goodwill, data, contracts, use of money, or business interruption, regardless of whether the claim is based on delict, tort (including negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent), or otherwise, and regardless of the nature or jurisdiction of the claim. The security analysis is solely based on the smart contracts and does not cover the security of applications or operations. No product code has been reviewed. If you have any doubts about the authenticity of this document, please verify using the provided QR code.