A&D FORENSICS

Preliminary Audit Report

For

March, 2021

Table of contents

Introduction 01

Audit goals 02

Issues category 03

Manual audit 04

Automated audit 05

Disclaimer 06

Introduction

This Audit Report mainly focuses on the overall security of ENAToken.sol Contract. With this report, we have tried to ensure the reliability and correctness of their smart contract by complete and rigorous assessment of their token contract and associated libraries.

1. Auditing Approach and Methodologies applied

The A&D Forensics team has performed rigorous testing of the project starting with analyzing the code design patterns in which we reviewed the smart contract architecture to ensure it is structured and safe use of third-party smart contracts and libraries.

Our team then performed a formal line by line inspection of the Smart Contract with unit and E2E testing to find any potential issue like race conditions, transaction- ordering dependence, timestamp dependence, and denial of service attacks.

The code was tested in collaboration of our team members and this included -

- Testing the functionality of the Smart Contract to determine proper logic has been followed throughout the whole process.
- Analyzing the complexity of the code in depth and detailed, manual review of the code, line-by-line.
- Deploying the code on testnet using multiple clients to run live tests. Analyzing failure preparations to check how the Smart Contract performs in case of any bugs and vulnerabilities.
- Checking whether all the libraries used in the code are on the latest version.

1. Audit Details

This audit is based of commit hash `bf0966f39b627cecfc7a4c3fc80fde45218c5aea` of the GitHub repository - https://github.com/Earnathon/token-contract

The contract in scope of the audit is ENAToken.sol

2. Audit Goals

The focus of the audit was to verify that the Smart Contract System works according to token standard specifications. The audit activities was based on:

Code Correctness and Quality
A full review of the contract source code. The primary areas of focus include:

- Accuracy
- Readability
- Sections of code with high complexity

2. Issue Categories

Every issue in this report was assigned a severity level from the following:

1. High severity issues

Issues on this level are critical to the smart contract's performance/ functionality and should be fixed before moving to a live environment.

2. Medium severity issues

Issues on this level could potentially bring problems and should eventually be fixed.

3. Low severity issues

Issues on this level are minor details and warnings that can remain unfixed but would be better fixed at some point in the future.

Number of issues per severity

	Low	Medium	High
Open	3	16	0
Closed	0	0	0

Manual Audit

- It implemented all the functions of an ERC20 Token from an Interface
- The contract compiles successfully.
- Contract was not properly locked to its specific version
- Change in allowance in the transferFrom function should come before changes in the Balances.
- Proper comments were not found on some functions like approveAndCall and transferAnyERC20Token.

Automated Audit

Slither

Slither, an open-source static analysis framework. This tool provides rich information about Ethereum smart contracts and has the critical properties. While Slither is built as a security-oriented static analysis framework, it is also used to enhance the user's understanding of smart contracts, assist in code reviews, and detect missing optimizations.

High level severity issues

None

Medium level severity issues

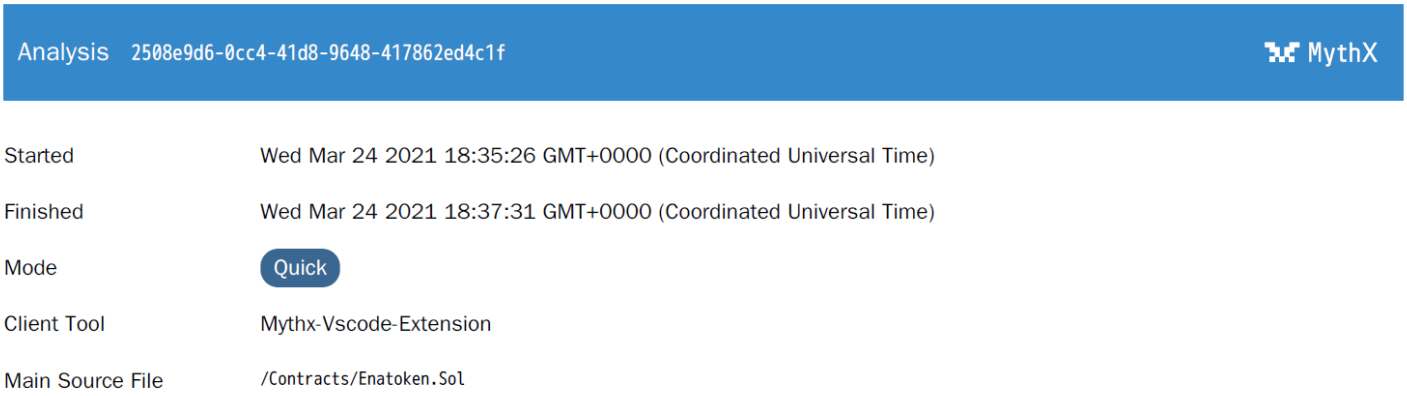
The function transferAnyERC20Token ignores the return value of the line ERC20(_tokenAddress), transfer(_to,_amount)

Low level severity issues

None

MythX

MythX is an enterprise smart contract audit software with community catalog of known smart contract vulnerabilities with detailed descriptions, code samples, and remediations. MythX uses the SWC Registry as its database when scanning smart contracts for security issues.



High level severity issues

None

Medium level severity issues

https://github.com/ConvexityTeam/token-contract/blob/master/ENAtoken-MythX.pdf

Low level severity issues

https://github.com/ConvexityTeam/token-contract/blob/master/ENAtoken-MythX.pdf

3. Disclaimer

This report is not an endorsement or indictment of any particular project or team, and the report does not guarantee the security of any particular project. This report does not consider, and should not be interpreted as considering or having any bearing on, the potential economics of a token, token sale or any other product, service or other asset. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty.

This report does not provide any warranty or representation to any Third-Party in any respect, including regarding the bugfree nature of code, the business model or proprietors of any such business model, and the legal compliance of any such business. No third party should rely on this report in any way, including for the purpose of making any decisions to buy or sell any token, product, service or other asset. Specifically, for the avoidance of doubt, this report does not constitute investment advice, is not intended to be relied upon as investment advice, is not an endorsement of this project or team, and it is not a guarantee as to the absolute security of the project. I owe no duty to any Third-Party by virtue of publishing these Reports.

The scope of my review is limited to a review of Solidity code and only the Solidity code noted as being within the scope of the review within this report. The Solidity language itself remains under development and is subject to unknown risks and flaws. The review does not extend to the compiler layer, or any other areas beyond Solidity that could present security risks. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty.

This audit does not give any warranties on finding all possible security issues of the given smart contracts, i.e., the evaluation result does not guarantee the nonexistence of any further findings of security issues. As one audit-based assessment cannot be considered comprehensive, I always recommend proceeding with several independent audits and a public bug bounty program to ensure the security of smart contracts.