KAIZEN GLOBAL

Title

Blockchain Security

Smart Contract
Final Audit Report

KaizenGlobal.org

contact@KaizenGlobal.org

+92 334 0738638

info@thekaizenglobal.co

S-19 Second Floor, Gold Mall
Murree Road Rawalpindi

Contents

1. Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice at the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the disclaimer below – please make sure to read it in full.

By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and KaizenGlobal and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers, and other representatives) (KaizenGlobal) owe no duty of care towards you or any other person, nor does KaizenGlobal make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "asis", without any conditions, warranties, or other terms of any kind except as set out in this disclaimer, and KaizenGlobal hereby excludes all representations, warranties, conditions, and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Exceptand only to the extent that it is prohibited by law, KaizenGlobal hereby excludes all liability and responsibility,and neither you nor any other person shall have any claim against KaizenGlobal, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of the use of this report, and any reliance on this report. The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed. KaizenGlobal owns all copyright rights to the text, images, photographs, and other content provided in the following document. When using or sharing partly or in full, third parties must provide a direct link to the original document mentioning the author (thekaizenglobal.com).

2. Overview

Kaizen Global was commissioned by the Documenda team to perform an audit of their smart contract.The audit was conducted between 14/04/2023 and 02/05/2023.

The purpose of this audit was to achieve the following:

- Identify potential security issues with smart contracts
- Formally check the logic behind given smart contracts.

Information in this report should be used for understanding the risk exposure of smart contracts, and as a guide to improving the security posture of smart contracts by remediating the issues that were identified.

The code is available at Polygon Mumbai Test Network:

Document 0x224793c65Eb051F3b5a9184e8852EdB36d0FbDEf

2.1 Summary

Project	**Documenda**
URL	https://mumbai.polygonscan.com/address/0x224793c65eb051f3b5a9184e8852edb36d0fbdef#code
Platform	Polygon Mumbai

Language	Solidity

2.2 Contracts

Name	Address
Document	0x224793c65Eb051F3b5a9184e8852EdB36d0FbDEf

3. Found issues

High	3
Medium	1
Low	1
Info	1

C1. Document.sol

ID	Severity	Title	Status
C1-01	High	Check for document expiration in verifyDocument()	Acknowledged
C1-02	High	Check for document is verified or not in verifyDocument()	Acknowledged
C1-03	High	Fee (0.5) floating-point number	Acknowledged
C1-04	Medium	SPDX - license missing	Acknowledged
C1-05	Low	Record of total documents	Acknowledged
C1-06	Info	Missing functions, variables	Acknowledged

4. Contracts

C1. Document.sol

Overview

The Medical Document Repository Smart Contract is a decentralized application (dApp) built on the Polygon network that allows users to securely store and manage their medical documents. The smart contract facilitates interactions between three types of users: General Users, Documenters, and Verifiers. General Users are everyday users who want to store and manage their documents, Documenters are healthcare providers (e.g., doctors, hospitals, labs) who upload documents on behalf of General Users, and Verifiers are agencies (e.g., embassies, schools) who verify the authenticity of medical documents provided by General Users.

Issues

C1-01 Issue# 1

High Acknowledged

Check for document expiration in verifyDocument()

Recommendation

Use a check like require statement to check
either the document is expired or not.

C1-02 Issue# 2

High Acknowledged

Check for document is already verified or

not in verifyDocument()

Recommendation

Use a check like require statement to check
document is already verified or not

C1-03 Issue# 3

Fee (0.5) floating-point number

High Acknowledged

Recommendation

This error message indicates that you cannot assign a floating-point value to an integer variable without first converting it to an integer. To fix this error, you can either change the type of the variable to a floating-point type like float or double, or you can convert the floating-point value to an integer using a typecast, like this:

C1-04 Issue# 4

SPDX-License missing

Medium Acknowledged

Recommendation

Use SPDX license to compile the contract properly

C1-05 Issue# 5

Record of total documents uploaded

Low Acknowledged

Recommendation

Store all the uploaded documents in an array
to keep tracking all the uploaded documents

C1-06 Issue# 6

Missing functions and variables in the contract

Info Acknowledged

Recommendation

Missing functions and variables like: documenterRole, scheduleAppointment(), documentCategory, retrieveDocument().

Note: These missing functionalities have been included in the contract.

5. Conclusion

3 high, 1 medium, 1 low severity issues were found during the audit. Issues were resolved in the update.

The reviewed contract is dependent on the role defined in the contract. Users using the project have to trust the documenters to upload documents on their behalf and should wait for the verifiers to verify their contracts

Appendix A. Issues' severity classification

- **Critical.** Issues that may cause an unlimited loss of funds or entirely break the contract workflow. Malicious code (including malicious modification of libraries) is also treated as a critical severity issue. These issues must be fixed before deployments or fixed in already running projects as soon as possible.
- **High.** Issues that may lead to a limited loss of funds, break interaction with users, or other contracts under specific conditions. Also, issues in a smart contract, that allow a privileged account the ability to steal or block other users' funds.
- **Medium.** Issues that do not lead to a loss of funds directly, but break the contract logic. May lead to failures in contracts operation.
- **Low.** Issues that are of a non-optimal code character, for instance, gas optimization tips, unused variables, errors in messages.
- **Informational.** Issues that do not impact the contract operation. Usually, informational severity issues are related to code best practices, e.g. style guide, missing or additional things.

Appendix B. List of examined issue types

- Business logic overview
- Functionality checks
- Following best practices
- Access control and authorization
- Reentrancy attacks
- Front-run attacks
- DoS with (unexpected) revert
- DoS with block gas limit
- Transaction-ordering dependence
- ERC/BEP and other standards violation
- Unchecked math
- Implicit visibility levels
- Excessive gas usage
- Timestamp dependence
- Forcibly sending ether to a contract
- Weak sources of randomness
- Shadowing state variables
- Usage of deprecated code

Kaizen Global

Blockchain Security

+92 334 0738638

info@thekaizenglobal.co

S-19 Second Floor, Gold Mall
Murree Road Rawalpindi