The Timebox token is a new cryptocurrency based in Ethereum that allows any user to have a permanent, verifiable and secure storage system. Timebox is implemented in Blockchain technology, allowing the user to control the files stored in their system, as well as sharing them with third users (registered or not in Timebox).

As Online Software Engineer, I have had access to the source code of this project with the objective of completing an exhaustive audit of its source code, analyzing different aspects related to the security, stability and efficiency of the project. This is my report.

INFORMATION

Project: Timebox.Network

Report Type: Analysis and Opinion on the stability and reliability of the source code of Smart Contract

Object: Smart Contracts

General features source: Developed in Solidity Contract-Oriented 0.4.11

SCOPE OF TESTS

The methodology used in the process has made it possible to carry out an exhaustive review of the audited application, covering the following main safety aspects detailed below:

Authorization: It is verified that the proper authorization mechanisms have been implemented; clearly defining the types or profiles of users and the rights of such users.

Based: The development of this smart contract has its origins in the FirstBlood project; which was developed without the use of standardized framework.

Data Validation: It verifies that robust Data Validation mechanisms exist and include all data that can be modified by a malicious user such as HTTP headers, input fields, hidden fields, list data, cookies, HTTP headers/data; performing checks of data validation on the server and not on the client side, precluding the existence of “backdoors” in the validation model.

Error handling: We review all methods / functions that return values have proper error handling and return proven and expected values in error conditions. Managing exceptions and error situations; that no system errors are returned to the user.

Logging / Auditing: It is audited that no sensitive information is stored in the application logs: cookies, information in “GET” methods, authentication credentials, etc.; the application logs the actions that occur in the application by users and especially in cases of potentially dangerous actions; all authentication events, failed or not, are logged; etc.

Cryptography: No sensitive data is transmitted without encryption, either internally or externally, and encryption algorithms or methods are recognized and standardized on known BlockChain systems, possessing the necessary robustness.

Secure Code: It analyzes the file’s structure to ensure that no resource that should not be accessible by the user; memory reservations / releases are correctly performed; not all logical decisions have a default clause.

Interoperability: The ERC20 Tokens Standard is implemented, allowing optimal interaction with alternative smart contracts, as well as with decentralized applications in the Ethereum block chain. The use of this standard allows a drastic reduction of the error rate in the interaction. This type of tokens are easily tracked in the blockchain, since they are a specific type of smartcontracts that “lives” in the Ethereum blockchain.

SOURCE CODE COMPLEXITY

Given the type of audit, a study of the Functional Complexity of Source Code is added, allowing an evaluation of the same, following the Cyclomatic Complexity metric, which is based on the flow diagram determined by the control structures of the source code; regardless of the programming language used. From this analysis can be obtained a quantitative measure of the difficulty of creating automatic tests of the code and is also a guideline measurement of the reliability of the same.

The measure given by this study is 7. This measure indicates that the code is properly structured, facilitating the testing, in addition to establishing the minimum associated security risk.

RESULTS

As a result of the audit of the delivered code, the following opinion is issued by the Auditor: There are currently many Blockchain-based projects that opt for the use of Ethereum and the ERC20 standard. It is very likely that this market will continue to grow with new and better applications that meet this standard in order to interact with each other. This allows the software to interoperate with other applications that meet the aforementioned standard.

Also, the use of Solidity Contract-Oriented as High Level Language, allows a short learning curve, thanks to its similarity to the JavaScript Script Language.

The above described, along with the characteristics of the source code audited, allows this auditor to make a positive assessment in the scope of the source code. Therefore, a high reliability in all significant aspects is estimated.