SOLIDProof

Bring trust into your projects

Blockchain Security | Smart Contract Audits | KYC

MADE IN GERMANY

CarbonCredit

Audit

Security Assessment

17. March, 2022

For

CARBON CREDIT

Disclaimer

SolidProof.io reports are not, nor should be considered, an “endorsement” or “disapproval” of any particular project or team. These reports are not, nor should be considered, an indication of the economics or value of any “product” or “asset” created by any team. SolidProof.io do not cover testing or auditing the integration with external contract or services (such as Unicrypt, Uniswap, PancakeSwap etc’...)

SolidProof.io Audits do not provide any warranty or guarantee regarding the absolute bug- free nature of the technology analyzed, nor do they provide any indication of the technology proprietors. SolidProof Audits should not be used in any way to make decisions around investment or involvement with any particular project. These reports in no way provide investment advice, nor should be leveraged as investment advice of any sort.

SolidProof.io Reports represent an extensive auditing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology. Blockchain technology and cryptographic assets present a high level of ongoing risk. SolidProof’s position is that each company and individual are responsible for their own due diligence and continuous security. SolidProof in no way claims any guarantee of security or functionality of the technology we agree to analyze.

Version	Date	Description
1.0	17. February 2022	• Layout project • Automated- /Manual-Security Testing • Summary

Network
Algorand

Website
https://cctoken.co/

Description

CC Token is the future of carbon cryptocurrencies. By linking our coin to the price of EUA Carbon Credits, we're opening the door for retail investors to join the world's largest regulated carbon market.

A disruptor to a global carbon market which has been dominated for institutions for too long - CC Token allows you to invest in one of the fastest growing asset spaces of the 21st century. Diversify your portfolio and invest in our planet's future.

Project Engagement

During the Date, **CC Team** engaged Solidproof.io to audit smart contracts that they created. The engagement was technical in nature and focused on identifying security flaws in the design and implementation of the contracts. They provided Solidproof.io with access to their code repository and whitepaper.

Logo

Source files

v1.0

- We got the source files in a .zip file.

Vulnerability & Risk Level

Risk represents the probability that a certain source-threat will exploit vulnerability, and the impact of that event on the organization or system. Risk Level is computed based on CVSS version 3.0.

Level	Value	Vulnerability	Risk (Required Action)
Critical	9 - 10	A vulnerability that can disrupt the contract functioning in a number of scenarios, or creates a risk that the contract may be broken.	Immediate action to reduce risk level.
High	7 – 8.9	A vulnerability that affects the desired outcome when using a contract, or provides the opportunity to use a contract in an unintended way.	Implementation of corrective actions as soon aspossible.
Medium	4 – 6.9	A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.	Implementation of corrective actions in a certain period.
Low	2 – 3.9	A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.	Implementation of certain corrective actions or accepting the risk.
Informational	0 – 1.9	A vulnerability that have informational character but is not effecting any of the code.	An observation that does not determine a level of risk

Auditing Strategy and Techniques Applied

Throughout the review process, care was taken to evaluate the repository for security-related issues, code quality, and adherence to specification and best practices. To do so, reviewed line-by-line by our team of expert pentesters and smart contract developers, documenting any issues as there were discovered.

Methodology

The auditing process follows a routine series of steps:

1. Code review that includes the following:
    i) Review of the specifications, sources, and instructions provided to SolidProof to make sure we understand the size, scope, and functionality of the smart contract.
    ii) Manual review of code, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
    iii) Comparison to specification, which is the process of checking whether the code does what the specifications, sources, and instructions provided to SolidProof describe.

2. Testing and automated analysis that includes the following:
    i) Test coverage analysis, which is the process of determining whether the test cases are actually covering the code and how much code is exercised when we run those test cases.
    ii) Symbolic execution, which is analysing a program to determine what inputs causes each part of a program to execute.

3. Best practices review, which is a review of the smart contracts to improve efficiency, effectiveness, clarify, maintainability, security, and control based on the established industry and academic practices, recommendations, and research.

4. Specific, itemized, actionable recommendations to help you take steps to secure your smart contracts.

Used Code from other Frameworks/Smart Contracts (direct imports)

Imported packages:

Algorand SDK
https://github.com/algorand/js-algorand-sdk

Node PostgreSQL
https://www.npmjs.com/package/pg

Tested Contract Files

This audit covered the following files listed below with a SHA-1 Hash.

A file with a different Hash has been modified, intentionally or otherwise, after the security review. A different Hash could be (but not necessarily) an indication of a changed condition or potential vulnerability that was not within the scope of this review.

v1.0

File	SHA-1 Hash
CCUpdateAlgorand.js	63df7786a8cffe75fa2bb2672292e4a43be999dc
mint_tokens.js	060d2060b69e3262f3121b7385b83ec72c344cb9

Metrics

Capabilities

Components

Version	Files	Libraries	Interfaces	Abstract
1.0	2	2	0	0

Scope of Work/Verify Claims

The above token Team provided us with the files that needs to be tested (Github, Bscscan, Etherscan, files, etc.). The scope of the audit is the main contract (usual the same name as team appended with .sol).

We will verify the following claims:

1. Correct implementation of Coding standard
2. Security Issues in the code
3. Testing the logic
4. Overall checkup

Tested	Verified
✓	✓

Legend

Attribute	Symbol
Verfified / Checked	✓
Partly Verified	⚑
Unverified / Not checked	✘
Not available	-

Source Units in Scope
v1.0

Legend

Attribute	Description
Lines	total lines of the source unit
nLines	normalized lines of the source unit (e.g. normalizes functions spanning multiple lines)
nSLOC	normalized source lines of code (only source-code lines; no comments, no blank lines)
Comment Lines	lines containing single or block comments
Complexity Score	a custom complexity score derived from code statements that are known to introduce code complexity (branches, loops, calls, external interfaces, ...)

Audit Results

AUDIT PASSED

Critical issues

No critical issues

High issues

No high issues

Medium issues

No medium issues

Low issues

Issue	File	Type	Line	Description
#1	CCUpdateAlgorand	Escape SQL statements	38	Even if the value is not a user input, we recommend escaping the string e.g. with https://github.com/mysqljs/sqlstring.

Informational issues

Issue	File	Type	Line	Description
#1	CCUpdateAlgorand	Static variable	13	We recommend to add this value to the .env file
#2	CCUpdateAlgorand	Static variable	8	We recommend to add this value to the .env file
#3	mint_tokens.js	Static variable	9	We recommend to add this value to the .env file

Audit Comments

We recommend you to use the special form of comments for your files to provide rich documentation for functions, return variables and more. This helps investors to make clear what that variables, functions etc. do.

. February 2022:

- Read whole report for more information
- All files work as expected. A standard asset can be created on the Algorand chain and then the holders can be saved to the PostgreSQL database.

- The tests were performed using PureStake.io interface.

Testing

Creation of an asset
https://testnet.algoexplorer.io/tx/
62FET4NQDXLATULJBJOTZNX7RJ7XLDWUANFYAMTNPOS3VR3XHELA

Asset created
https://testnet.algoexplorer.io/asset/78628417

Blockchain Security | Smart Contract Audits | KYC

MADE IN GERMANY