CERTIK

Security Assessment

EcoCelium

Jun 9th, 2021

Table of Contents

Summary

This report has been prepared for EcoCelium smart contracts, to discover issues and vulnerabilities in the source code of their Smart Contract as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Manual Review techniques.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

Additionally, this audit is based on a premise that all external smart contracts are implemented safely.

The security assessment resulted in 11 findings that ranged from minor to informational. We recommend addressing these findings to ensure a high level of security standards and industry practices. We suggest recommendations that could better serve the project from the security perspective:

- Enhance general coding practices for better structures of source codes;
- Add enough unit tests to cover the possible use cases given they are currently missing in the repository;
- Provide more comments per each function for readability, especially contracts are verified in public;
- Provide more transparency on privileged activities once the protocol is live.

Overview

Project Summary

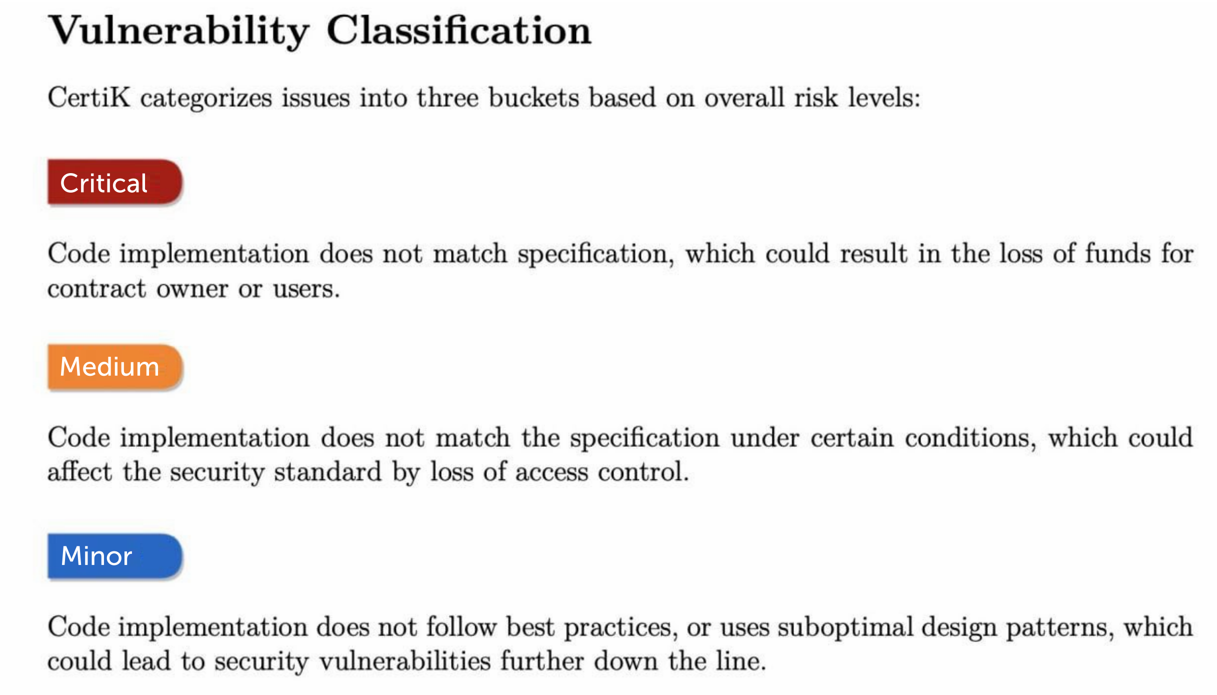
Project Name	EcoCelium
Platform	BSC
Language	Solidity
Codebase	https://gitlab.com/ecocelium/ecooptions
Commits	92d23b479734234c15da0897fb6d900b72c30e45

Audit Summary

Delivery Date	Jun 09, 2021
Audit Methodology	Manual Review
Key Components	

Vulnerability Summary

Total Issues	11
● Critical	0
● Major	0
● Medium	0
● Minor	4
● Informational	7
● Discussion	0



Vulnerability Classification

CertiK categorizes issues into three buckets based on overall risk levels:

Critical

Code implementation does not match specification, which could result in the loss of funds for contract owner or users.

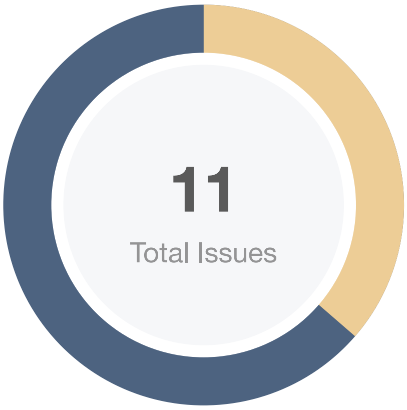
Medium

Code implementation does not match the specification under certain conditions, which could affect the security standard by loss of access control.

Minor

Code implementation does not follow best practices, or uses suboptimal design patterns, which could lead to security vulnerabilities further down the line.

Findings



🟥 **Critical**	**0** (0.00%)
🟧 **Major**	**0** (0.00%)
🟨 **Medium**	**0** (0.00%)
🟨 **Minor**	**4** (36.36%)
🟦 **Informational**	**7** (63.64%)
🟩 **Discussion**	**0** (0.00%)

ID	Title	Category	Severity	Status
ONF-01	Overly-Privilege Granted To Owner	**Centralization / Privilege**	● **Minor**	ⓘ **Acknowledged**
ONF-02	Missing Emit Event	Coding Style	● Informational	ⓘ Acknowledged
SAO-01	Overly-Privilege Granted To Owner	**Centralization / Privilege**	● **Minor**	ⓘ **Acknowledged**
SAO-02	Incorrect Contract Name	Compiler Error	● Minor	✓ Resolved
SAO-03	Check Effect Interaction Pattern Violated	Logical Issue	● Informational	✓ Resolved
SAO-04	Missing Emit Event	Coding Style	● Informational	ⓘ Acknowledged
SAO-05	Zero Address	Logical Issue	● Informational	✓ Resolved
SAV-01	Overly-Privilege Granted To Owner	**Centralization / Privilege**	● **Minor**	ⓘ **Acknowledged**
SAV-02	Check Effect Interaction Pattern Violated	Logical Issue	● Informational	Partially Resolved
SAV-03	Missing Emit Event	Coding Style	● Informational	ⓘ Acknowledged
SAV-04	Zero Address	Logical Issue	● Informational	✓ Resolved

Appendix

Finding Categories

Centralization / Privilege

Centralization / Privilege findings refer to either feature logic or implementation of components that act against the nature of decentralization, such as explicit ownership or specialized access roles in combination with a mechanism to relocate funds.

Logical Issue

Logical Issue findings detail a fault in the logic of the linked code, such as an incorrect notion on how block.timestamp works.

Coding Style

Coding Style findings usually do not affect the generated byte-code but rather comment on how to make the codebase more legible and, as a result, easily maintainable.

Compiler Error

Compiler Error findings refer to an error in the structure of the code that renders it impossible to compile using the specified version of the project.

Checksum Calculation Method

The "Checksum" field in the "Audit Scope" section is calculated as the SHA-256 (Secure Hash Algorithm 2 with digest size of 256 bits) digest of the content of each file hosted in the listed source repository under the specified commit.

The result is hexadecimal encoded and is the same as the output of the Linux "sha256sum" command against the target file.

Disclaimer

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to the Company in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes without CertiK's prior written consent.

This report is not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any "product" or "asset" created by any team or project that contracts CertiK to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing 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. CertiK's position is that each company and individual are responsible for their own due diligence and continuous security. CertiK's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.

About

Founded in 2017 by leading academics in the field of Computer Science from both Yale and Columbia University, CertiK is a leading blockchain security company that serves to verify the security and correctness of smart contracts and blockchain-based protocols. Through the utilization of our world-class technical expertise, alongside our proprietary, innovative tech, we're able to support the success of our clients with best-in-class security, all whilst realizing our overarching vision; provable trust for all throughout all facets of blockchain.

CERTIK