EXTROPY.IO

Report for:

Cudo Ventures Limited

July 2021

Version: 2.0

Prepared By:	Extropy.IO
Email:	info@extropy.io
Telephone:	+44 1865261424

Table of Contents

1 Using This Report

To facilitate the dissemination of the information within this report throughout your organisation, this document has been divided into the following clearly marked and separable sections.

Document Breakdown		
0	Executive Summary	Management level, strategic overview of the assessment and the risks posed to the business
1	Technical Summary	An overview of the assessment from a more technical perspective, including a defined scope and any caveats which may apply
2	Technical Findings	Detailed discussion (including evidence and recommendations) for each individual security issue which was identified
3	Methodologies	Audit process and tools used

Disclaimer

The audit makes no statements or warranty about utility of the code, safety of the code, suitability of the business model, regulatory regime for the business model, or any other statements about fitness of the code to purpose, or their bug free status. The audit documentation is for discussion purposes only.`

Document Control

Client Confidentiality

This document contains Client Confidential information and may not be copied without written permission.

Proprietary Information

The content of this document should be considered proprietary information and should not be disclosed outside of Cudo Ventures Limited
Extropy gives permission to copy this report for the purposes of disseminating information within your organisation or any regulatory agency.

Document Version Control			
Data Classification	Client Confidential		
Client Name	Cudo Ventures Limited		
Document Title	Cudo Smart Contracts Audit		
Author	Extropy Audit Team		

Document History			
Issue No.	**Issue Date**	**Issued By**	**Change Description**
1.0	14/07/2021	Laurence Kirk	Released to client
2.0	27/07/2021	Laurence Kirk	Released to client

Document Distribution List	
Dominic Burns	Cudo Ventures Limited
Ethan Illingworth	Cudo Ventures Limited
Laurence Kirk	CEO, Extropy

2 Executive Summary

Extropy was contracted by Cudo Ventures Limited to conduct a code review and smart contracts vulnerability assessment in order to identify security issues that could negatively affect Cudo Ventures Limited's business or reputation if they led to the compromise or abuse of systems.
This report presents the findings of the smart contract security assessment conducted on behalf of Cudo Ventures Limited The assessment was conducted between 05/07/21 and 13/07/2021 and was authorised by Cudo Ventures Limited.

Following the initial audit, the code was retested at commit
9c6bcbe8f937df8fd33593033c20802b01931844

The majority of issues have been resolved since the initial report. We have added another informational issue (4.11) that could help with optimisation.

2.1 Assessment Summary

The contracts are up to accepted standards and showed evidence of good design.
The only issue remaining concerns contract size, since deployment is possible this has been moved to low priority, if the contracts are to be developed further, it would be worth looking at further optimisations.

The following table breaks down the issues which were identified by phase and severity of risk.

Phase	Description	Critical	High	Medium	Low	Info	Total
1	Initial Audit	0	0	1	1	8	10
2	Final Audit	0	0	0	1	3	4

3 Technical Summary

3.1 Scope

File Name	SHA-1 Hash
src/contracts/staking/StakingRewardsGuild.sol	f6e2bc4fc16eaa130aee9d0ba7d568ce347f1bbe
src/contracts/staking/CloneFactory.sol	7c1a04455863c688267c337b121c5ef7610914aa
src/contracts/staking/StakingRewards.sol	eb90af8c7c3a915eee55a93fd7eb9406b586f407
src/contracts/staking/ServiceProvider.sol	ab29b6ec911772d43a305ec0d7bcdc4f8d1fcdba

4 Technical Findings – Code Audit

The remainder of this document is technical in nature and provides additional detail about the items already discussed, for the purposes of remediation and risk assessment.

4.1 Contract Size

Risk Rating	Medium

The size of StakingRewards and ServiceProvider exceeds the 24KB limit

"contracts/staking/StakingRewards.sol:18:1: Warning: Contract code size exceeds 24576 bytes (a limit introduced in Spurious Dragon).

Recommendation:
Optimise the code or move functions to libraries where possible.
Reducing the size of string literals in require statements (see below) would help here.

Outcome: The optimiser is being used to enable deployment.

4.2 Solidity Version

Risk Rating	Low

The version of solidity used 0.6.12 is 2 major versions behind the latest.
A later version would allow the latest versions of libraries to be used, and take advantage of recent optimisations and improvements in the language, such as support for safe math operations and ABI coder V2.

Recommendation:
Upgrade to solidity version 8

Outcome : Resolved, version 0.8.0 is being used.

4.3 Use modifiers for code reuse

Risk Rating	Informational

There are require statements repeated in functions in ServiceProvider that could be replaced with a modifier.
For example ServiceProvider lines 94-96.
This would reduce the contract size and improve readability.

Recommendation:
Replace the require statements with a modifier

Outcome : Resolved as suggested.

4.4 Use library functions where possible

Risk Rating	Informational

The function updateUserActionsPaused in StakingRewards could be replaced by the pausable functionality in Open Zeppelin Pausable

Recommendation:
Use Open Zeppelin Pausable
(https://docs.openzeppelin.com/contracts/4.x/api/security#Pausable)

Outcome : Resolved : Similar functionality has been added with a modifier

4.5 Limit string literals to a length of 32 bytes

Risk Rating	Informational

To reduce gas costs, limit string literals to 32 bytes.
Many require statements have large message string literals, (for example Staking Rewards line 290) which specify the contract name and function, a more compact representation of this would decrease the contract size.

Recommendation:
Reduce the size of the message in require statements.

Outcome : Resolved, String lengths have been reduced

4.6 Code readability

Risk Rating	Informational

A typo could cause confusion if it needs to be referenced externally.
Service Provider - line 50 - typo on Cal*l*ibrated... might cause issues when writing event listeners

Recommendation:
Fix the typo.

Outcome : Resolved

4.7 Use built in time units if possible

Risk Rating	Informational

See Staking Rewards - line 93
"uint256 public unbondingPeriod = numOfBlocksInADay.mul(21); // Equivalent to solidity 21 days"

Recommendation:
Use the built in days time unit.

4.8 Remove unnecessary public functions

Risk Rating	Informational

Recommendation:

Simplify public function _getBlock as this is functionally equivalent to block.number.

4.9 Use event indexes appropriately

Risk Rating	Informational

Do not index event fields unnecessarily, for example where the values indexed are unlikely occur more than once. See the amount field in ServiceProvider lines 46, 47,48,49, 50

Recommendation:
Remove some indexes in these events.

Outcome : Resolved, indexes have been removed.

4.10 Remove unused import statements

Risk Rating	Informational

Remove unused import statements

Recommendation:
In StakingRewards.sol the imports for Enumerable set and Ownable Open Zeppelin contracts are not used

Outcome : Resolved, the import statements have been removed.

4.11 Remove usage of safe math library

Risk Rating	Informational

Recommendation:
Solidity version 8 has checks for underflow and overflow, so the safe math library is not needed. Removing calls to this library would save gas.

5 Tool List

The following tools were used during the assessment:

Tools Used	Description	Resources
Solidity Metrics	Static analysis	https://github.com/ConsenSys/solidity-metrics
SWC Registry	Vulnerability database	https://swcregistry.io/
Mythx	Static Analysis	https://mythx.io/

5.1 Tailored Methodologies

5.1.1 Audit Goals

1. We will audit the code in accordance with the following criteria:

- **Sound Architecture**

  This audit includes assessments of the overall architecture and design choices. Given the subjective nature of these assessments, it will be up to the development team to determine whether any changes should be made.

- **Smart Contract Best Practices**

  This audit will evaluate whether the codebase follows the current established best practices for smart contract development.

- **Code Correctness**

  This audit will evaluate whether the code does what it is intended to do.

- **Code Quality**

  This audit will evaluate whether the code has been written in a way that ensures readability and maintainability.

- **Security**

  This audit will look for any exploitable security vulnerabilities, or other potential threats to the users.

- **Testing and testability**

- This audit will examine how easily tested the code is, and review how thoroughly tested the code is.

Although we have commented on the application design, issues of crypto-economics, game theory and suitability for business purposes as they relate to this project are beyond the scope of this audit.

5.2 Test Methodology

The security audit is performed in two phases:

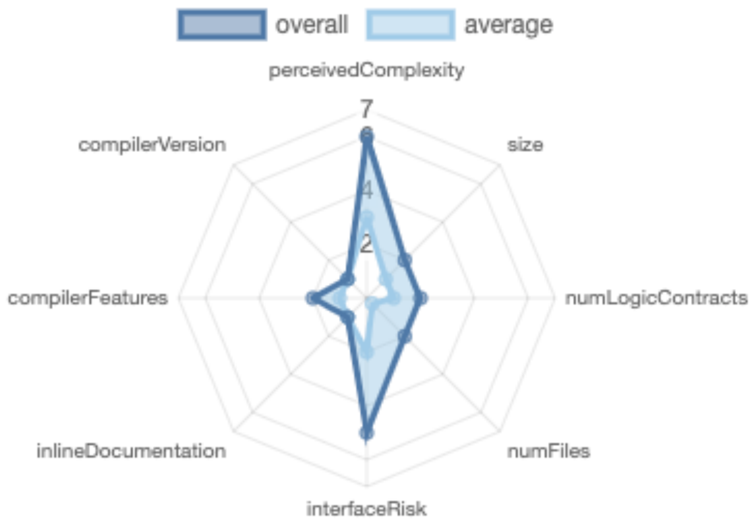
a. **Independent Code Review**

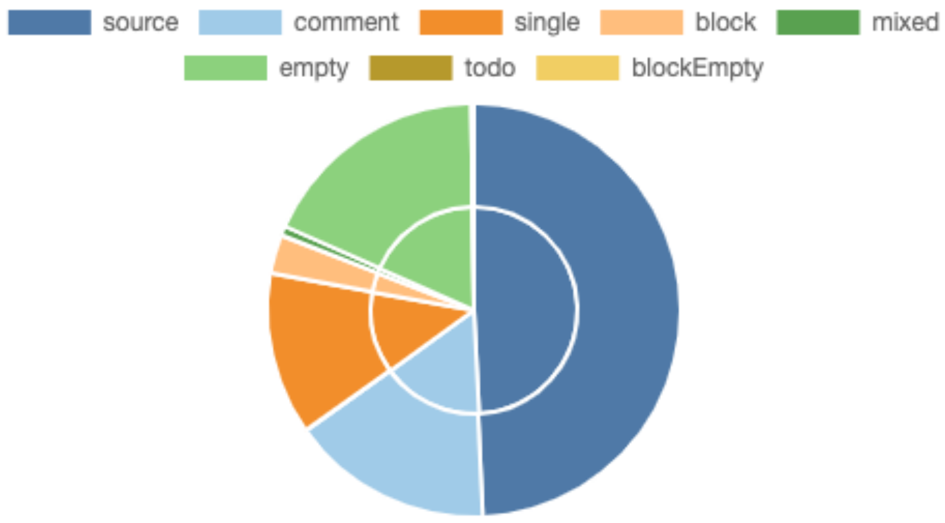
b. The code is inspected separately by four team members checking for software errors and known vulnerabilities.

c. **Static Analysis**

The code is subject to static analysis using Solidity Metrics and Mythx

5.3 Solidity Code Metrics


overall
average
perceivedComplexity
size
numLogicContracts
numFiles
interfaceRisk
inlineDocumentation
compilerFeatures
compilerVersion


source
comment
single
block
mixed
empty
todo
blockEmpty