Report for:

Animal Concerts

October 2021

Version: 2.1

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

Table of Contents

1. Executive Summary

Extropy was contracted to conduct a code review and vulnerability assessment of the Animal smart contract
This report presents the findings of that audit, conducted between 21/10/21 and 25/10/2021.
No major issues were found, and there is one recommendation.

The final audit was conducted after consideration by the developers, the recommendation from the initial report was actioned and no issues remain.

1.1. Assessment Summary

The contract avoids unneeded complexity and uses standard libraries which reduces the risk of vulnerabilities.
The only recommendation is to upgrade to newer version of the Open Zeppelin libraries.
This recommendation was followed for the final report.

Phase	Description	Critical	High	Medium	Low	Info	Total
1	Initial Audit	0	0	0	0	1	1
2	Final Audit	0	0	0	0	0	0

2. 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.

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

2.1. Client Confidentiality

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

2.2. Proprietary Information

The content of this document should be considered proprietary information. 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	Animal Concerts		
Document Title	Animal Concerts Audit		
Author	Extropy Audit Team		

Document History			
Issue No.	**Issue Date**	**Issued By**	**Change Description**
1.0	25/10/2021	Laurence Kirk	Released to client
2.0	01/11/2021	Laurence Kirk	Released to client
2.1	02/06/2021	Laurence Kirk	Released to client

2. Technical Summary

2.1. Scope

This audit discusses the ERC20 token Animal in compilation unit Animal_ERC20_noOwner.sol

2.2. Design

The contract is a standard implementation of an ERC20 inheriting from Open Zeppelin contracts.

3. Technical Findings

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.

As this is a simple audit we present our findings as a checklist of potential issues

Issue	Status
Returns bool after transfer	Yes
Prevent transferring tokens to the 0x0 address	No
Prevent transferring tokens to the contract address	No
Re entrant Calls	N/A
Fee on transfer	N/A
Balance Modifications Outside of Transfers	N/A
Upgradable tokens	No
Flash Mintable tokens	No
Tokens with Blocklists	N/A
Revert on Zero Value Transfers	No
Decimals returns uint8	Yes
No Revert on Failure	Yes
Revert on Large Approvals & Transfers	No
Code Injection Via Token Name	N/A
Approval Race protection	No

None of the above items presents a vulnerability.

4. Issues Found

4.1. Upgrade to latest version of Open Zeppelin libraries

Risk Rating	Informational

Description:

The latest stable version of the Open Zeppelin libraries is version 4.3.2 whereas this project uses version 4.1.0. The optional upgrade would allow the use of the hook _afterTokenTransfer though in this instance, this is probably not needed.

Outcome : Resolved, libraries upgraded to v4.3.2

5. Tool List

The following tools were used during the assessment:

Tools Used	Description	Resources
SWC Registry	Vulnerability database	https://swcregistry.io/

6. General Audit Goals

We 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 and Rust 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.

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.

Apart from manually reviewing the code we also checked for:

- Compliance with ERC20 standard
- Token integration
- Token interaction
- Token implementation
- Best practices