EXTROPY.IO

Report for:

PWN Project

July 2021

Version: 1.0

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 without permission,
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	PWN Finance		
Document Title	PWN Finance Smart Contracts Audit		
Author	Extropy Audit Team		

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

Document Distribution List	
Josef Jelacic	PWN Finance development team
Laurence Kirk	CEO, Extropy

2 Executive Summary

Extropy was contracted to conduct a code review and smart contracts vulnerability assessment in order to identify security issues that could negatively affect the business or reputation of the project if they led to the compromise or abuse of systems. This report presents the findings of the smart contract security assessment conducted between 15/07/21 and 27/07/2021.

2.1 Assessment Summary

The contracts are of good design and clearly and concisely written.
The issues identified are mainly low or informational risk, and are all easily remedied

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	4	2	7

3 Technical Summary

3.1 Scope

Contracts

MultiToken

PWNDeed

PWNVault

PWNController

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 Uninitialized Variables

Risk Rating	Medium

The variables
address public collector;
address public DAO;
are not initialized and so will retain their default values when being used.

Affects PWNController

Recommendation:
Set the address variables in the constructor, and mark them as immutable.

4.2 Floating Pragma

Risk Rating	Low

Contracts should be deployed with the same compiler version and flags that they have been tested with
See https://swcregistry.io/docs/SWC-103

Affects MultiToken,PWNDeed, PWNVault, PWNController

Recommendation:
Use the same fixed version of solidity in all contracts

4.3 Insufficient Verification of data authenticity

Risk Rating	Low

Function parameters should be checked for valid values
See https://cwe.mitre.org/data/definitions/345.html

Affects
PWNDeed line 390

Recommendation:
Add a check for a zero address.

4.4 Handle return values appropriately

Risk Rating	Low

Return values from external calls should be handled with a clear error path.
See https://swcregistry.io/docs/SWC-104

This has mostly been achieved in the contracts by the use of require statements, though not in all cases. A consistent approach is recommended.

Affects
Multitoken lines 39 / 63 / 68

Recommendation:
Provide a consistent pattern to handle external calls such as token transfers. One alternative is to use the safe transfer functions available from Open Zeppelin.

4.5 Transaction ordering around token approvals

Risk Rating	Low

See https://swcregistry.io/docs/SWC-114
and the discussions at
https://medium.com/coinmonks/solidity-transaction-ordering-attacks-1193a014884e

https://github.com/OpenZeppelin/openzeppelin-contracts/issues/438

Affects
Multitoken lines 121 / 125

Recommendation:
One alternative is to only allow approval changes from zero to the required amount.

4.6 Irrelevant Code

Risk Rating	Informational

See https://swcregistry.io/docs/SWC-135

Affects
MultiToken line 64 and 109
PWNDeed line 108

Recommendation:
Remove unreachable code or code that has been commented out.

4.7 Add license information to contracts

Risk Rating	Informational

See https://docs.soliditylang.org/en/v0.6.8/layout-of-source-files.html?highlight=spdx#spdx-license-identifier

Affects MultiToken,PWNDeed, PWNVault, PWNController

Recommendation:
Add SPDX license identifiers to the source files.

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:

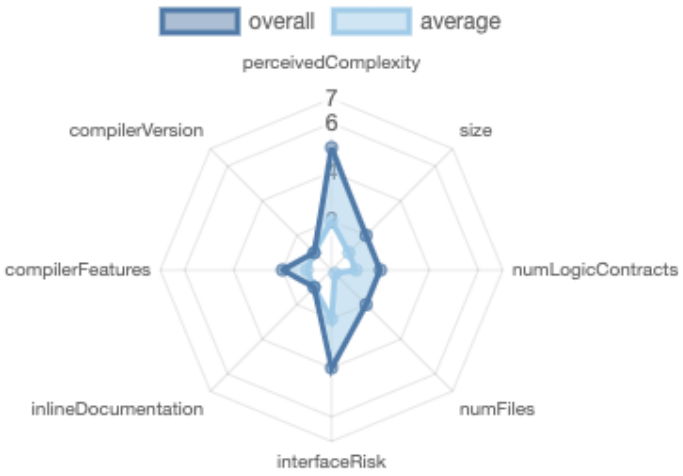
a. **Independent Code Review**

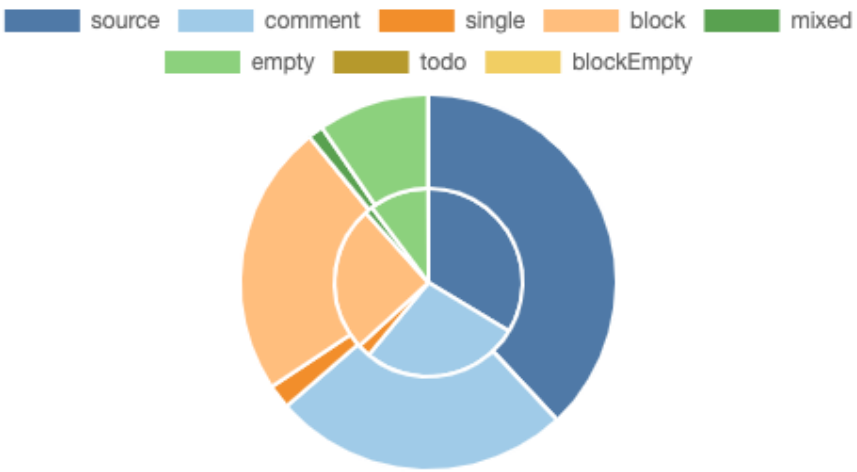
b. The code is inspected separately by four team members checking for software errors and known vulnerabilities.

c. **Static Analysis**

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

5.3 Solidity Code Metrics





5.4 Mythx Findings

MythX

Analysis 82bc20e4-19de-4342-a45a-7615755ca53c MythX

Started	Fri Jul 23 2021 14:58:07 GMT+0000 (Coordinated Universal Time)
Finished	Fri Jul 23 2021 15:45:20 GMT+0000 (Coordinated Universal Time)
Mode	Deep
Client Tool	Remythx
Main Source File	PWNDeed.Sol

DETECTED VULNERABILITIES

HIGH	MEDIUM	LOW
0	0	3

ISSUES

LOW A floating pragma is set.

SWC-103

The current pragma Solidity directive is ""^0.8.0"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file
IERC1155Receiver.sol
Locations

```
1 // SPDX-License-Identifier: MIT
2
3 pragma solidity ^0.8.0;
4
5 import "./IERC165.sol";
```