Report for:

PWN Finance

February 2023

Version 1.0

Email	Telephone
info@extropy.io	+44 1865261424

Document Version Control

Data Classification	Client Confidential
Client Name	PWN Finance
Document Title	PWN Protocol Audit Final Report
Author	Extropy Audit Team

Executive Summary

Extropy was contracted to conduct a code review and vulnerability assessment of the project. The review was carried out between 3rd and 16th January 2023.
A retest was carried between 25th January and 3rd February.
The design and code is of the high quality , of 2 major issues highlighted in the initial report, one was found to be a false positive, the other although possible does have possible mitigation and will be addressed in a later version.

Assessment Summary

The issue relating to the use of tax tokens or rebase tokens as collateral is understood by the development team.
The remaining issues have either been fixed, or deemed to be acceptable at this stage.

Issue Count

Stage	Critical	Medium	Low	Informational
Initial Report	0	1	2	0

Scope

The code audited is taken from repo Github repo
at commit e0c6802e4fe39dd04bea32f8e50c2960897357ec

Contracts in scope

PWNErrors
PWNHubTags
PWNHub
PWNHubAccessControl
PWNVault
PWNSimpleLoan
PWNSimpleLoanRequest
PWNSimpleLoanSimpleRequest
PWNSimpleLoanOffer
PWNSimpleLoanSimpleOffer
PWNSimpleLoanListOffer
IPWNSimpleLoanTermsFactory
PWNLOANTerms
IERC5646
IPWNLoanMetadataProvider
PWNLOAN
PWNSignatureChecker
PWNFeeCalculator
PWNConfig
PWNDeployer
PWNContractDeployerSalt
PWNRevokedNonce

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.

ERC-1271 Invalidates EIP-712 signatures

On discussion with the development team this issue was found to be a false positive.

Status

Closed

Tax / Rebase Tokens - insufficient loan repayment

Risk	**Medium**
Affects	PWNSignatureChecker.sol

Tax token scenario:

Alice applies for a loan with using 10 1% taxed token.
Bob makes an offer.
Alice accepts the offer. 10 taxed tokens get transferred to the loan contract. But contract received 99 taxed tokens.
Alice closes the loan, but the transaction fails as the loan contract has only 9.9 taxed tokens, whereas 10 are needed.

Rebase Tokens scenario

Alice applies for a loan with using 10 rebase tokens.
Bob makes an offer.
Alice accepts the offer. 10 rebase tokens get transferred to the loan contract.
Negative rebase takes place, the loan contract has 9 rebase tokens.
Alice closes the loan. But the transaction fails as loan contract has only 9 rebase tokens.

Rebase Tokens from multiple loans in one contract

Alice creates a loan with 10 rebase tokens.
Bob created a loan with 10 rebase tokens.
Loan contract has 20 rebase tokens.
Rebase takes place. 20 rebase tokens balance get reduced to 18 tokens.

Alice closes the loan to avoid negative rebase loss. Alice gets 10 rebase tokens back. Alice is in profit. Loan contract has 8 rebase tokens.
Bob wants to close the loan. Bob should get 9 rebase tokens. Contract thinks that Bob should get 10 rebase tokens. Contract has only 8 rebase tokens. Loan repayment transaction fails.

Recommendation

Keep collateral for each loan in a separate contract, and then return the collateral based on the contract's token balance.

Status

The development team were aware of this issue, and will resolve it in a later version. Some mitigation is possible through the front end.

Missing Check for zero address

Risk	Low
Affects	PWNConfig.sol

Line 63 : function initialize(address *owner, uint16 _fee, address _feeCollector)*
There are no checks on the addresses `owner` *or `_feeCollector`*

Recommendation

Add require statements to check the addresses passed are valid.

Status

This risk associated with this issue has been deemed acceptable at the moment and no action is being taken.

Missing Check for empty arrays

Risk	Low
Affects	PWNHub.sol

If an empty array is passed to function setTag, an error will be thrown.

Recommendation

Check the length of arrays `_addresses` and `_tags`

Status

This risk associated with this issue has been deemed acceptable at the moment and no action is being taken.

Possible Reentrancy

Risk	Low
Affects	PWNSimpleLoan.sol

After discussion with the development team we felt that reentrancy could not be exploited, and therefore this issue has been closed.

Status

Closed

Possible overflow in calculateFeeAmount function

Risk	Low
Affects	PWNFeeCalculator.sol

Since a uint16 is used (`function calculateFeeAmount(uint16 fee, uint256 loanAmount)`) overflow is possible, with this version of Solidity this would revert rather than produce an incorrect value.

Status

Closed - This has been fixed and tests created for the issue

`domainSeparator` can be hardcoded

Risk	Informational
Affects	`PWNSimpleLoanSimpleOffer.sol`, `PWNSimpleLoanListOffer.sol`
	`PWNSimpleLoanSimpleRequest.sol`

Status

Closed - This has been fixed.

Offer struct optimisations

Risk	Informational
Affects	`PWNSimpleLoanSimpleOffer.sol`, `PWNSimpleLoanListOffer.sol`,
	`PWNSimpleLoanSimpleRequest.sol`

Status

Closed - This optimisation has been applied

Unnecessary checks

Risk	Informational
Affects	`PWNSimpleLoanSimpleOffer.sol`, `PWNSimpleLoanListOffer.sol`,
	`PWNSimpleLoanSimpleRequest.sol`, `PWNSimpleLoan`

Status

Closed - This optimisation has been applied

Appendix A

Tools used

Static Analysis was performed, using Slither

Appendix B

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 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.