Cryptonics

Audit Report

pNetworks Safemoon Token and Vault Smart Contracts

May 26, 2021

Table of Contents

Disclaimer

Cryptonics Consulting S.L.
Ramiro de Maeztu 7
46022 Valencia
SPAIN

https://cryptonics.consulting/
info@cryptonics.consulting

Introduction

Purpose of this Report

Cryptonics Consulting has been engaged to perform a security audit of the pNetworks token and vault smart contracts on Ethereum for the pTokens 2-way asset transfer bridge (https://p.network/). The objectives of the audit are as follows:

1. Determine the correct functioning of the contracts, in accordance with the project specification.
2. Determine possible vulnerabilities, which could be exploited by an attacker.
3. Determine contract bugs, which might lead to unexpected behavior.
4. Analyze whether best practices have been applied during development.
5. Make recommendations to improve code safety and readability.

This report represents a summary of the findings.

As with any code audit, there is a limit to which vulnerabilities can be found, and unexpected execution paths may still be possible. The author of this report does not guarantee complete coverage (see disclaimer).

Codebase Submitted for the Audit

The audit has been performed on the following GitHub repositories:

https://github.com/provable-things/ptokens-safemoon-erc777/

latest commit covered: `1393d9dabffa73accb57f9f799646109a02eac4f`

https://github.com/provable-things/ptokens-safemoon-vault/

latest commit covered: `4f1f5ae651756890a91719cad8c8abf1cb2c84c1`

Methodology

The audit has been performed in the following steps:

1. Gaining an understanding of the code base’s intended purpose by reading the available documentation.
2. Automated source code and dependency analysis.
3. Manual line by line analysis of the source code for security vulnerabilities and use of best practice guidelines, including but not limited to:
    a. Race condition analysis
    b. Under- / overflow issues
    c. Key management vulnerabilities
4. Report preparation

Functionality Overview

The submitted contracts implement the pTokens ERC-777 representation, adapted for the Safemoon token model and an adapted vault contract that allows locking and releasing tokens, capable of dealing with the SafeMoon model.

How to read this Report

This report classifies the issues found into the following severity categories:

Severity	Description
Critical	A serious and exploitable vulnerability that can lead to loss of funds, unrecoverable locked funds, or catastrophic denial of service.
Major	A vulnerability or bug that can affect the correct functioning of the system, lead to incorrect states or denial of service.
Minor	A violation of common best practices or incorrect usage of primitives, which may not currently have a major impact on security, but may do so in the future or introduce inefficiencies.
Informational	Comments and recommendations of design decisions or potential optimizations, that are not relevant to security. Their application may improve aspects, such as user experience or readability, but is not strictly necessary. This category may also include opinionated recommendations that the project team might not share.

The status of an issue can be one of the following: **Pending, Acknowledged** or **Resolved**. Informational notes do not have a status, since we consider them optional recommendations.

Note, that audits are an important step to improve the security of smart contracts and can find many issues. However, auditing complex codebases has its limits and a remaining risk is present (see disclaimer).

Users of the system should exercise caution. In order to help with the evaluation of the remaining risk, we provide a measure of the following key indicators: **code complexity**, **code readability**, **level of documentatio**n, and **test coverage**. We include a table with these criteria for each module, in the corresponding findings section.

Note, that high complexity or lower test coverage does not necessarily equate to a higher risk, although certain bugs are more easily detected in unit testing than a security audit and vice versa.

Summary of Findings

No	Description	Severity	Status
Safemoon pToken			
1	pToken.sol: Duplicated Inheritance	Informational	Resolved
Safemoon Vault			
2	Erc20Vault: Unbounded loops may cause issues with block gas limit	Minor	Acknowledged
3	pToken.sol: Duplicated Inheritance	Informational	Resolved

Code Quality Criteria

Safemoon pToken

Criteria	Status	Comment
Code complexity	Low-Medium	-
Code readability and clarity	High	-
Level of Documentation	High	-
Test Coverage	High	-

Safemoon Token

Criteria	Status	Comment
Code complexity	Low-Medium	-
Code readability and clarity	High	-
Level of Documentation	High	-
Test Coverage	High	-

Detailed Findings

Safemoon pToken

1. `pToken.sol`: Duplicated Inheritance

Severity: Information

The `pToken` contract inherits both, `ERC777Upgradeable` and `ERC777GSNUpgreadable`. However, `ERC777GSNUpgreadable` already inherits `ERC777Upgradeable`.

Recommendation

Consider removing duplicate inheritance to simply the inheritance graph.

Status: Resolved

SafeMoon Vault

2. `Erc20Vault`: Unbounded loops may cause issues with block gas limit

Severity: Minor

The `migrate()` and `destroy()` functions iterate over a data structure that may grow very large if many tokens are added. This means, that if the data structures grow too large, the calls to these functions will hit the block gas limit, causing the transactions to always fail.

Recommendation

Implement a limit on the number of tokens that can be added, in line with the block gas limit.

Team Reply

This contract will only ever be used for one token. However, the original vault from which this codebase manages multiple tokens and we will implement a limit on the maximum number of registered tokens.

Status: Acknowledged

3. `Erc20Vault`: Consider zero-check for `setPNetwork()`

Severity: Informational

The `setPNetwork()` function does check for `address(0)` as a parameter. Since this allows renouncing the PNetwork privileged role, it could result in accidentally lose of control over the vault in an irrecoverable way.

Recommendation

Add a zero-check.

Status: Resolved