Cryptonics

Continuous Audit Report 3

pTokens Bridges

Februar 12, 2020

Table of Contents

Disclaimer

THE CONTENT OF THIS AUDIT REPORT IS PROVIDED “AS IS”, WITHOUT REPRESENTATIONS AND WARRANTIES OF ANY KIND.

THE AUTHOR AND HIS EMPLOYER DISCLAIM ANY LIABILITY FOR DAMAGE ARISING OUT OF, OR IN CONNECTION WITH, THIS AUDIT REPORT.

COPYRIGHT OF THIS REPORT REMAINS WITH THE AUTHOR.

This audit has been performed by

Philip Stanislaus

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 continuous audit of the pTokens 2-way asset transfer bridges, forming part of the pTokens project (https://ptokens.io/). The engagement is ongoing and includes the changes performed to the codebase since the last audit report from October 15, 2020.

The objectives of the audit are as follows:

1. Determine the correct functioning of the contract, 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 changes between the following commits in the code provided by the developers in the following GitHub repository:

https://github.com/provable-things/ptokens-core-private

Start commit: `b5a47d70afbbb9f5084efeb1c9807749e1395ad7` (tag `v1.11.0`)
Final commit: `462167c86a37f6efcccd2a261a93af52d6370b83`

This report presents the findings over 232 commits with 238 changed files, containing 9,107 additions and 6,915 deletions.

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 code implements a cross-blockchain bridge that allows assets to be moved between Ethereum and EOS as well as between Bitcoin and EOS. Bitcoin is represented on Ethereum and EOS as pBTC.

The two-way peg works by depositing (locking) BTC on Bitcoin and minting pBTC on Ethereum or EOS and by burning pBTC on Ethereum or EOS and unlocking BTC on Bitcoin. Transactions are relayed across chains through light clients designed to operate in a secure enclave.

Since the last audit report, support for moving ERC20 assets between Ethereum and EOS has been added.

The enclave is designed to be executed in a protected enclave using a trusted execution environment, such as Intel SGX.

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
1		Major	Acknowledged
2		Medium	Acknowledged
3		Minor	Resolved
4		Minor	Resolved
5		Minor	Resolved
6		Minor	Resolved
7		Minor	Acknowledged
8		Minor	Acknowledged
9		Minor	Acknowledged

This report contains 9 findings on 16 pages (including one cover page).

Code Quality Criteria

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

Detailed Findings

1. Issue

Severity: Major

Recommendation

Status: Pending