Code Audit

for

StreamFlow Labs

15 Rue des Halles
75001 Paris
France

https://opcodes.fr
contact@opcodes.fr

Siret :89435788800011
RCS : Paris B 894 357 888
TVA : FR35894357888

Project Information

Project	
Mission	Code audit
Client	Streamflow Labs
Start Date	12/20/2021
End Date	12/29/2021

Document Revision			
Version	Date	Details	Authors
1.0	12/28/2021	Document creation	Thibault MARBOUD Xavier BRUNI
1.0	12/29/2021	Peer review	Baptiste OUERIAGLI
2.0	01/06/2022	Public version	Thibault MARBOUD

Table of Contents

Overview

Mission Context

The purpose of the mission was to perform a code audit to discover issues and vulnerabilities in the mission scope. A comprehensive testing has been performed utilizing automated and manual testing techniques.

Mission Scope

As defined with Streamflow Labs prior to the mission, the scope of this assessment was two Solana programs. This is the second report that only concerns the Timelock implementation, a first one has been edited for the Timelock-crate. The code source was supplied through the following GitHub repositories:

- https://github.com/StreamFlow-Finance/timelock-crate / b1caedd (community)
- https://github.com/StreamFlow-Finance/timelock / 08f8468 (community)

OPCODES engineers are due to strictly respect the perimeter agreed with Streamflow Labs as well as respecting ethical hacking behavior.

Note: OPCODES engineers audited the community branch.

Project Summary

Timelock repository contains a Solana program that aim to give an example on how to implement the Timelock-crate. No logic is added on top of the library, each instruction is a proxy to one of the functions provided by Timelock library.

We strongly recommend looking at the Timelock-crate report before reading this report.

Synthesis

Security Level: GOOD

The overall security level is considered as good.

Timelock implementation is done using Anchor framework and is mostly used as an “entrypoint” to the Timelock-crate without adding any logic.

OPCODES reported two informational vulnerabilities that concern some dependencies update and a test private key inside the git repository.

Vulnerabilities summary

Total vulnerabilities	0
🟥 Critical	0
🟧 Major	0
🟨 Medium	0
🟩 Minor	0
🟦 Informational	2

Vulnerabilities & issues table

Identified vulnerabilities

Ref	Vulnerability title	Severity	Remediation effort
#1	Outdated dependencies	🟦 Informational	🟩 Low
#2	Private key in git repository	🟦 Informational	🟩 Low

Outdated dependencies

Severity	Remediation effort
🟦 Informational	🟩 Low

Description

The following crates could be updated:

Crate	Current version	Latest version
anchor-lang	0.17.0	0.19.0
anchor-spl	0.17.0	0.19.0

Scope

Timelock

Risk

This is an informational issue, it does not represent any risk but may enforce bad practice.

Remediation

It is considered as a good practice to update dependencies when possible.

Private key in git repository

Severity	Remediation effort
🟦 Informational	🟩 Low

Description

The file `tests/test-wallet.json` contains a private key that was probably used for testing purposes.

Scope

Timelock

Risk

As the private key seems to only be used with devnet network, it does not represent any risk but may enforce bad practice.

Remediation

We strongly discourage the storage of private key inside a git repository. As the file seems to not be used anymore, we recommend deleting it.

Conclusion

Timelock implementation is straightforward and OPCODES investigation did not produce any probative results. If possible, we still recommend remediating the informational vulnerabilities as the efforts needed should be low.

Timelock is a simple wrapper around Timelock-crate, hence it heavily relies on Timelock-crate security hygiene. As long as Timelock relies on a safe and audited version of Timelock-crate, there are few chances that any vulnerability would arise on it.