OFFSIDE LABS

Jupiter Lend

Smart Contract Security Assessment

October 2025

Prepared for:

Jupiter

Prepared by:

Offside Labs

Yao Li

Siji Feng

Contents

1 About Offside Labs

Offside Labs is a leading security research team, composed of top talented hackers from both academia and industry.

We possess a wide range of expertise in modern software systems, including, but not limited to, *browsers*, *operating systems*, *IoT devices*, and *hypervisors*. We are also at the forefront of innovative areas like *cryptocurrencies* and *blockchain technologies*. Among our notable accomplishments are remote jailbreaks of devices such as the **iPhone** and **PlayStation 4**, and addressing critical vulnerabilities in the **Tron Network**.

Our team actively engages with and contributes to the security community. Having won and also co-organized *DEFCON CTF*, the most famous CTF competition in the Web2 era, we also triumphed in the **Paradigm CTF 2023** within the Web3 space. In addition, our efforts in responsibly disclosing numerous vulnerabilities to leading tech companies, such as *Apple*, *Google*, and *Microsoft*, have protected digital assets valued at over **$300 million**.

In the transition towards Web3, Offside Labs has achieved remarkable success. We have earned over **$9 million** in bug bounties, and **three** of our innovative techniques were recognized among the **top 10 blockchain hacking techniques of 2022** by the Web3 security community.

 **https://offside.io/**

 **https://github.com/offsidelabs**

 **https://twitter.com/offside_labs**

2 Executive Summary

Introduction

Offside Labs completed a security audit of *Fluid Solana* smart contracts, starting on October 13th, 2025, and concluding on October 19th, 2025.

Project Overview

The Fluid ecosystem on Solana consists of several core modules each serving a distinct purpose.

The Flashloan Program integrates with Fluid's Liquidity Layer to provide flashloan services and requires repayment within the same transaction ensuring security and efficiency.

The Oracle Program supplies various types of oracle data to support Fluid's Vault Program and plays a key role in borrowing and liquidation processes.

The Merkle Distributor Program implements token vesting using merkle proof technology enabling efficient and verifiable token distribution.

Audit Scope

The assessment scope contains mainly the smart contracts of the flashloan, merkleDistributor and oracle program for the *Fluid Solana* project.

The audit is based on the following specific branches and commit hashes of the codebase repositories:

- Fluid Solana
  - Codebase: `https://github.com/Instadapp/fluid-contracts-solana`
  - Commit Hash: c0d63a497f5e9ab115711d4d9bae328bf8a5dc46

We listed the files we have audited below:

- Fluid Solana
  - programs/flashloan/src/*.rs
  - programs/merkleDistributor/src/*.rs
  - programs/oracle/src/*.rs

Findings

The security audit revealed:

- 0 critical issue
- 0 high issue
- 0 medium issue
- 1 low issue
- 1 informational issue

Further details, including the nature of these issues and recommendations for their remediation, are detailed in the subsequent sections of this report.

3 Summary of Findings

ID	Title	Severity	Status
01	Stake-pool Fee Can Cause Oracle Unavailability	Low	Fixed
02	approve_root Ignores Paused State	Informational	Acknowledged

4 Key Findings and Recommendations

4.1 Stake-pool Fee Can Cause Oracle Unavailability

Severity: Low	Status: Fixed
Target: Smart Contract	Category: Logic Error

Description

If an SPL Stake Pool authority increases any checked fee above 0.5%, `is_fee_too_high` returns `FeeTooHigh`, causing this stake-pool oracle hop to fail. Any consumer (including liquidation) relying on this hop will revert until fees drop or the oracle config changes.

```
62      fn is_fee_too_high(stake_pool: &StakePool) -> Result<()> {
63          check_fee(&stake_pool.sol_withdrawal_fee)?;
64          check_fee(&stake_pool.stake_withdrawal_fee)?;
65
66          check_fee(&stake_pool.sol_deposit_fee)?;
67          check_fee(&stake_pool.stake_deposit_fee)?;
68
69          Ok(())
70      }
```

programs/oracle/src/modules/stake_pool.rs#L62-L70

Impact

Availability (not integrity), driven by external pool config. Fee increases are detectable one epoch in advance via `next_*` fields; actively monitor and pre-decide whether to rotate or temporarily use a cached rate.

Recommendation

Actively monitor pending withdrawal-fee changes (e.g., `next_stake_withdrawal_fee`, `next_sol_withdrawal_fee`) and alert/rotate before they activate; pre-decide whether to switch to a cached rate or replace this hop if a pending fee breaches the 0.5% ceiling.

Mitigation Review Log

Fixed in commit 4cdbb19389cdb66465ac4dc55b546745f70892c9.

4.2 Informational and Undetermined Issues

approve_root Ignores Paused State

Severity: Informational	Status: Acknowledged
Target: Smart Contract	Category: Logic Error

`approve_root` does not enforce the `paused` state, while `propose_root` does. If `pause()` is intended to freeze the distributor, both proposing and approving roots should be blocked during pause.

5 Disclaimer

This audit report is provided for informational purposes only and is not intended to be used as investment advice. While we strive to thoroughly review and analyze the smart contracts in question, we must clarify that our services do not encompass an exhaustive security examination. Our audit aims to identify potential security vulnerabilities to the best of our ability, but it does not serve as a guarantee that the smart contracts are completely free from security risks.

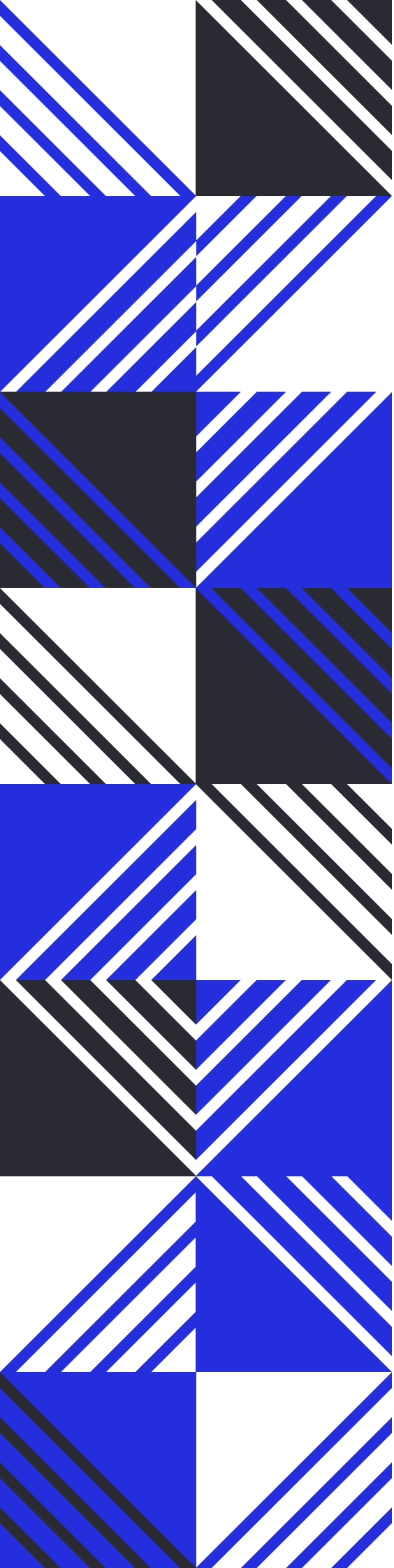
We expressly disclaim any liability for any losses or damages arising from the use of this report or from any security breaches that may occur in the future. We also recommend that our clients engage in multiple independent audits and establish a public bug bounty program as additional measures to bolster the security of their smart contracts.

It is important to note that the scope of our audit is limited to the areas outlined within our engagement and does not include every possible risk or vulnerability. Continuous security practices, including regular audits and monitoring, are essential for maintaining the security of smart contracts over time.

Please note: we are not liable for any security issues stemming from developer errors or misconfigurations at the time of contract deployment; we do not assume responsibility for any centralized governance risks within the project; we are not accountable for any impact on the project's security or availability due to significant damage to the underlying blockchain infrastructure.

By using this report, the client acknowledges the inherent limitations of the audit process and agrees that our firm shall not be held liable for any incidents that may occur subsequent to our engagement.

This report is considered null and void if the report (or any portion thereof) is altered in any manner.



OFFSIDE LABS

https://offside.io/

https://github.com/offsidelabs

https://twitter.com/offside_labs