OFFSIDE LABS

Kamino Lending WithdrawQueue

Smart Contract Security Assessment

February 2026

Prepared for:

Kamino

Prepared by:

Offside Labs

Ronny Xing



Contents

1 About Offside Labs **2**

2 Executive Summary **3**

3 Summary of Findings **5**

4 Key Findings and Recommendations **6**

4.1 Insufficient Availability Check for user_destination_liquidity_ta 6

4.2 Withdrawal Caps can Block Queue Redeem 6

4.3 Informational and Undetermined Issues 7

5 Disclaimer **9**

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 *Kamino Lending* smart contracts, starting on December 2th, 2025, and concluding on December 5th, 2025.

Project Overview

Kamino Lending is a decentralized lending protocol operating on the Solana blockchain.

The protocol utilizes a novel peer-to-pool model designed to facilitate borrowing and lending while supporting automated leveraged strategies such as Multiply and Long/Short positions.

Key technical characteristics include capital-efficient liquidity mechanisms and a risk management system intended to protect both lenders and borrowers. The protocol also functions as a standalone money market, allowing users to deposit and borrow assets without engaging in leveraged strategies.

Audit Scope

The assessment scope contains mainly the smart contracts of the klend program for the *Kamino Lending* project.

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

- Kamino Lending:
  - Codebase: `https://github.com/Kamino-Finance/klend-private`
  - Commit Hash: bdf8bbe893aaddb001b8f2ab5880ca8079887bda

We listed the files and commits we have audited below:

- Kamino Lending: programs/klend/src/*.rs
  - PR-544, 531, 533, 592, 593: 8f81f968d797f2e542262e72ee1b6bd7b237e123 to 0f839e1b77d2b78d37bf6fbca8b342bfda995fae
  - PR-598: c7bda266122bebdc9e5241cede6d740991391ceb
  - PR-609: 63160d68033ce1d5ee8fda3c24f77f0b4660f1cc
  - PR-617: 0c3dfb4285c2847f34dd7822ad3249454f8213b4
  - PR-634: 4987438f0e06d109953bc49e90a94ee3154c32e8
  - PR-637: ba995c01d38cf0abaf0a6172279e9de5ddf3ca17
  - PR-680: 88eec1e17e2d74c3c9e00333d0c43d299f24886e
  - PR-681: 9ea285a41cada322f4e509d9efaf5ae4a201c350

Findings

The security audit revealed:

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

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	Insufficient Availability Check for user_destination_liquidity_ta	High	Fixed
02	Withdrawal Caps can Block Queue Redeem	Low	Acknowledged
03	Dequeue Operations Should Not Be Blocked by Reserved Obsolete Status	Informational	Fixed
04	Use ATA as Receiving Address for Permissionless IX	Informational	Fixed

4 Key Findings and Recommendations

4.1 Insufficient Availability Check for user_destination_liquidity_ta

Severity: High	Status: Fixed
Target: Smart Contract	Category: Token

Description

The following token-2022 extension of type `AccountType::Account` may cause token reception failures when used as a destination address:

- `MemoTransfer` : when `MemoTransfer` is enabled on the token accounts, all token transfer without a memo will fail.
- `ConfidentialTransferAccount` with `allow_non_confidential_credits` is false: when `allow_non_confidential_credits` is false, all normal transfer of base token account will be blocked.

Impact

An attacker can insert tickets containing invalid `user_destination_liquidity_ta` into the withdraw queue. These tickets cannot be dequeued, blocking the execution of all subsequent withdraw tickets. In the worst-case scenario, as withdraw tickets accumulate, the entire reserve becomes unable to process liquidity withdrawals.

Recommendation

Add checks for `token_acc_info` extersions in the function `validate_liquidity_token_extensions` .

Or add `Invalid` branch for these extersions in function `DestinationTokenAccountValidity::resolve` .

Mitigation Review Log

Fixed in PR-675

4.2 Withdrawal Caps can Block Queue Redeem

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

Description

Instruction `withdraw_queued_liquidity` uses `RedeemCollateralOptions::FOR_WITHDRAW_QUEUED_LIQUIDITY` to call operation `redeem_reserve_collateral` with `use_withdraw_queue: true` .

If the caps config is enabled, requested_amount exceeding the cap limit will be blocked by the `check_capacity_allows_withdrawals` check. However, the ix `enqueue_to_withdraw` is never limited by withdrawal caps config.

Impact

Becuase the `withdraw_queued_liquidity` instruction always tries to withdraw `max_redeemable_collateral_amount` , large withdraw tickets may be rejected by withdraw caps, resulting in partial withdrawals being blocked even when sufficient liquidity exists. In the worst-case scenario, if the `requested_amount` of a single withdraw ticket exceeds the `config_capacity` setting, withdrawals from the withdraw queue will be completely blocked.

Recommendation

When calculating the `max_redeemable_collateral_amount` for a single withdraw ticket, the remaining withdrawable liquidity from the withdraw caps should be taken into account.

Mitigation Review Log

Kamino Team: Yes, that's a known issue, we have a ticket for addressing it, although prio is low.

(also, it has some non-obvious consequences regarding our ATA rent refund policy, given that withdrawal cap will force partial ticket redeems; hence it's not exactly a no-brainer fix)

4.3 Informational and Undetermined Issues

Dequeue Operations Should Not Be Blocked by Reserved Obsolete Status

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

Check functions `recover_invalid_ticket_collateral_checks` and `withdraw_queued_liquidity_checks` all check `withdraw_reserve.config.status()` must not be `ReserveStatus::Obsolete` .

However, ix `withdraw_queued_liquidity` and `recover_invalid_ticket_collateral` are endpoints for liquidity withdrawal, like ix `redeem_reserve_collateral` . They should not be limited by `ReserveStatus::Obsolete` .

Fixed in PR-675

Use ATA as Receiving Address for Permissionless IX

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

Its better to use ATA of withdraw_ticket owner as receiving address for ix `recover_invalid_ticket_collateral` .

```
96      /// The ticket's owner token account to which the ticket-locked
          ↪ collateral should be returned.
97      #[account(mut,
98          token::mint = reserve_collateral_mint,
99          token::authority = withdraw_ticket.load()?.owner,
100     )]
101     pub user_source_collateral: Box<InterfaceAccount<'info,
          ↪ TokenAccount>>,
```

programs/klend/src/handlers/handler_recover_invalid_ticket_-collateral.rs#L96-L101

Fixed in PR-675

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.

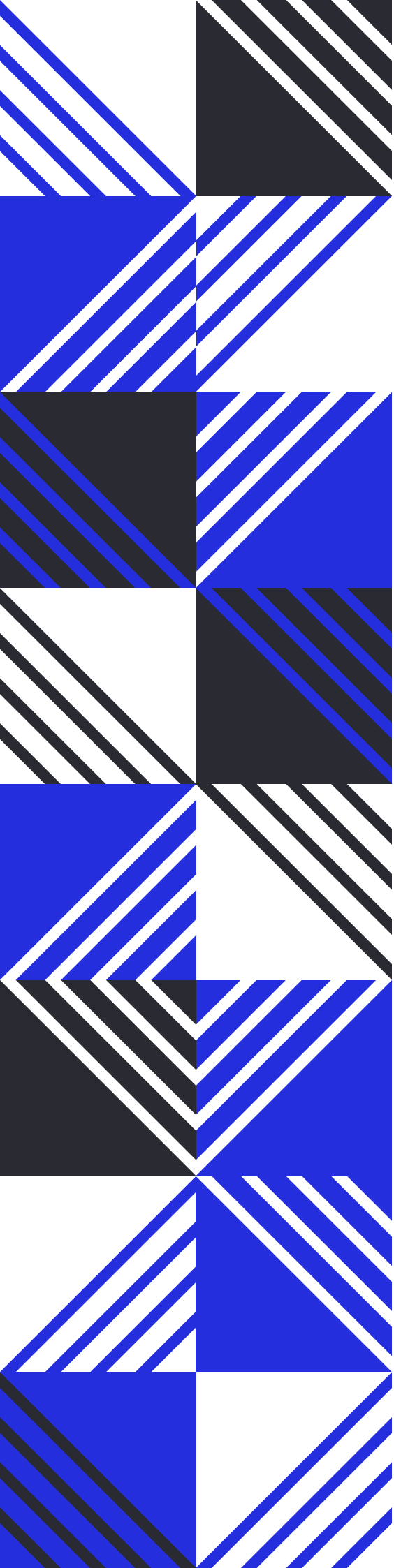
We expressly disclaim any liability for any losses or damages arising from the use of this report or from any security breaches that may occur in the future. We also recommend that our clients engage in multiple independent audits and establish a public bug bounty program as additional measures to bolster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