OFFSIDE LABS

Kamino Lending BorrowOrder

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 Suggesting Liquidation Against Lowest LLTV Collateral Upon Debt Maturity . . 6

4.2 Informational and Undetermined Issues 7

5 Disclaimer **8**

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 November 24th, 2025, and concluding on December 2th, 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-522: b4bbebc7acbc39ae787b173d175431996506cf8f
  - PR-524: 47460664156dd0ea1304f339edefbe34244571a6
  - PR-526: aeeea311dad725cf7847b7fc3948ae14150b2696
  - PR-537: 41aa82be0dcfd246df2fe8c82f98356fbee26e9d
  - PR-547: dba1bd0cd6eb687941c069597f547c4caa5712e5
  - PR-538: 895c74f57beba381e93c830e65942347b7ff6734
  - PR-540: 71d238bbabdb16f0eff110ac30923406ffc3f56f
  - PR-541: 8b3f3975bcbadb15f132dde3e32d71982b9542b9
  - PR-560: d9841935911ed26e035f2a77fbd2ae137af2e6ea
  - PR-572: b370e808d93c2d21546712d41ff8a3cef4747e80
  - PR-574: ad24c97bedb6014b1c839ee6d46490b6c56d6048
  - PR-580: bc979be6c040953613bec8c9060c33338cee0f3d
  - PR-581: 78139016caadb82c54bd5fe887baa36c8b2f82b7
  - PR-610 + PR-614: 42ea02892b445f077f8cd7a11357593ed44c7e58

- PR-631: d806f3c58f83fa1aca3d209d891c91da6b293256
- PR-619: edf3476b45e557c265e673b1f8a8efefcbbe55c3

Findings

The security audit revealed:

- 0 critical issue
- 0 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	Suggesting Liquidation Against Lowest LLTV Collateral Upon Debt Maturity	Low	Fixed
02	Repayment and Liquidation Could Reactivate Borrow Order	Informational	Acknowledged

4 Key Findings and Recommendations

4.1 Suggesting Liquidation Against Lowest LLTV Collateral Upon Debt Maturity

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

Description

It is suggested adding `is_collateral_reserve_lowest_liquidation_ltv` check for liquidation with reason `ReserveDebtMaturityReached` and `ReserveDebtMaturityReached` .

Impact

Liquidations triggered by `ReserveDebtMaturityReached` and `ReserveDebtMaturityReached` will fully liquidate the entire `borrowed_amount` of the specified debt. For composite obligations under extreme configurations, prioritizing the liquidation of collaterals with higher LLTV may cause the remaining positions' LTV to exceed the new LLTV threshold, then triggering LtvExceeded liquidations.

Proof of Concept

Initial scene:

- Debt-A: $20; borrow factor: x3;
- Collateral-B: $100, max LTV 50%, LLTV 60%;
- Debt-C: $800; borrow factor: x1;
- Collateral-D: $1000, max LTV 80%, LLTV 90%;
- Current LTV: (800+20*3)/1100 = 78.18%

After Debt-C maturity:

- Liquidate $800 debt-C with liquidation bonus 25%
- Withdraw collateral-D $800 * 1.25 = $1000
- LTV after liquidation: 20*3/100 = 60%
- Trigger LtvExceeded liquidation

Recommendation

Unlike ADL, debt maturity triggered liquidations do not target the LTV of the obligation but instead focus on liquidating specific debts. As such, these liquidations should avoid dragging the entire obligation into a cascading liquidation scenario.

It is suggested adding `is_collateral_reserve_lowest_liquidation_ltv` check for liquidation with reason `ReserveDebtMaturityReached` and `ReserveDebtMaturityReached` .

Mitigation Review Log

Fixed in PR-670 .

4.2 Informational and Undetermined Issues

Repayment and Liquidation Could Reactivate Borrow Order

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

Repay and liquidate operations can reset the corresponding ObligationLiquidity to default(). This resets the `first_borrowed_at_timestamp` field to 0.

If a borrow order was previously blocked due to the `seconds_until_reserve_debt_maturity` check , it can be re-filled after `first_borrowed_at_timestamp` is reset in repay or liquidation.

```
115     // Cannot fill using a reserve with a debt maturity timestamp set too
        ↪  early for this order:
116     if !is_term_satisfied(
117         borrow_order.get_min_debt_term_seconds(),
118         seconds_until_reserve_debt_maturity,
119     ) {
```

programs/klend/src/state/borrow_order_operations.rs#L115-L119

Kamino Team: We discussed it internally and we consider this “working as intended” (even if a business/ux decision driving this may seem somewhat arbitrary, that’s what we want). We will make sure to properly point out this behavior to the users on UI, possibly also offering them e.g. to cancel their BO when they repay a “related” loan.

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.

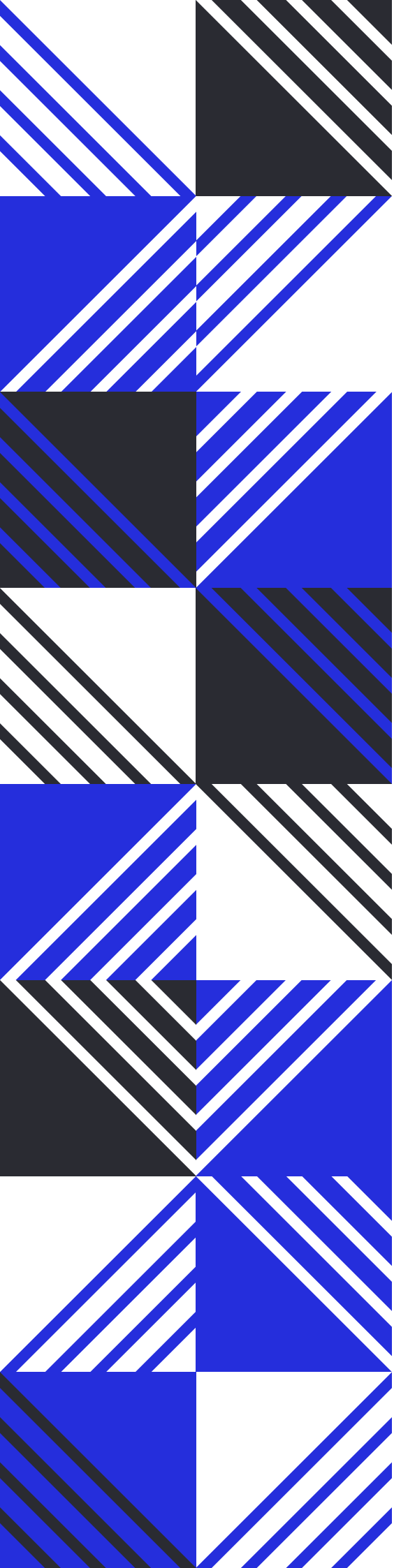
We expressly disclaim any liability for any losses or damages arising from the use of this report or from any security breaches that may occur in the future. We also recommend that our clients engage in multiple independent audits and establish a public bug bounty program as additional measures to bolster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