OFFSIDE LABS

Kamino LIMO

Smart Contract Security Assessment

November 2024

Prepared for:

Kamino

Prepared by:

Offside Labs

Yao Li

Ripples Wen

Ronny Xing

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 *LIMO* smart contracts, starting on Nov 25, 2024, and concluding on Nov 29, 2024.

Project Overview

Kamino LIMO (Liquidity Integration & Matching Orders) is a C2C limit order matching platform that prioritizes a seamless and efficient trading experience by offering zero fees and a zero-slippage swap mechanism, ensuring users receive the exact amount expected without incurring additional costs. It also features a flash swap capability, attracting liquidity-constrained searchers by enabling rapid and efficient trades. To further enhance security, LIMO integrates with Pyth Express Relayer, providing robust protection against Miner Extractable Value (MEV) exploitation. This innovative platform is designed to deliver fairness, efficiency, and accessibility for all users.

Audit Scope

The assessment scope contains mainly the smart contracts of the limo program for the *LIMO* project.

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

- LIMO
  - Codebase: https://github.com/Kamino-Finance/limo
  - Commit Hash: 0f4e77e1f33ef3a4f1b85a8b90c7c84287c78e04

We listed the files we have audited below:

- LIMO
  - programs/limo/src/**/*.rs

Findings

The security audit revealed:

- 0 critical issue
- 0 high issue
- 1 medium issues
- 1 low issues
- 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	Failure to Create Intermediary Token Account Due to Pre-existing Lamports	Medium	Fixed
02	Incompatibility with Token-2022 Accounts in Output Token Validation	Low	Fixed
03	Redundant Rent Sysvar Account in initialize_vault Instruction	Informational	Fixed
04	Timestamp of Order Not Updated in close_order_and_claim_tip Instruction	Informational	Fixed

4 Key Findings and Recommendations

4.1 Failure to Create Intermediary Token Account Due to Pre-existing Lamports

Severity: Medium	Status: Fixed
Target: Smart Contract	Category: Logic

Description

During the order-taking process, the intermediary token account is initialized using `system_instruction::create_account` within the `initialize_token_account_with_signer_seeds` function.

```
168 let create_account_ix = system_instruction::create_account(
169     authority.key,             // Payer of the account's rent
170     token_account.key,         // New WSOL token account address
171     rent_exempt_balance,       // Minimum rent-exempt balance
172     TokenAccount::LEN as u64, // Space needed for a token account
173     &spl_token::ID,            // Token program ID
174 );
```

programs/limo/src/token_operations.rs#L168-L174

However, since the address of the intermediary token account is predetermined, any pre-existing lamports in this account will prevent the creation of a new account. Consequently, this will result in the failure of the transaction.

Impact

This introduces a potential vector for a DoS attack on specific orders. Since the issue only impacts the targeted order, the maker can mitigate the problem by canceling the affected order and creating a new one. However, for the liquidity provision for liquidation, this temporary DoS could significantly reduce liquidation efficiency and timeliness.

Recommendation

It is recommended to manually supplement the rent, allocate the necessary data, and assign the appropriate owner when there are existing lamports in the intermediary token account.

Mitigation Review Log

Fixed in https://github.com/Kamino-Finance/limo/pull/42.

4.2 Incompatibility with Token-2022 Accounts in Output Token Validation

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

Description

During the order-taking process, if the output mint is not wSOL, the output token account is validated using `verify_ata`. However, since the associated token address is derived using the SPL token program id, this validation will fail for all Token-2022 accounts.

```
65 pub fn verify_ata(wallet: &Pubkey, mint: &Pubkey, ata_account_key:
   ↪ &Pubkey) -> Result<()> {
66     // Derive the expected ATA address
67     let expected_ata = get_associated_token_address(wallet, mint);
```

programs/limo/src/utils/constraints.rs#L65-L67

Impact

This prevents the possibility of swapping for all Token-2022 tokens.

Recommendation

To support Token-2022 tokens, pass the token program to `verify_ata` and calculate the associated token address using `get_associated_token_address_with_program_id`.

Mitigation Review Log

Fixed in https://github.com/Kamino-Finance/limo/pull/42.

4.3 Informational and Undetermined Issues

Redundant Rent Sysvar Account in initialize_vault Instruction

Severity: Informational	Status: Fixed
Target: Smart Contract	Category: Optimization

The rent sysvar account is not required in the `initialize_vault` instruction. Anchor uses the Rent account from Sysvar directly.

Timestamp of Order Not Updated in close_order_and_claim_tip Instruction

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

In the `close_order_and_claim_tip` instruction, the `order.last_updated_timestamp` field is not being updated. This could lead to inconsistencies, as the outdated timestamp is subsequently used in the `emit_cpi!` log, potentially causing incorrect or misleading log entries.

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.

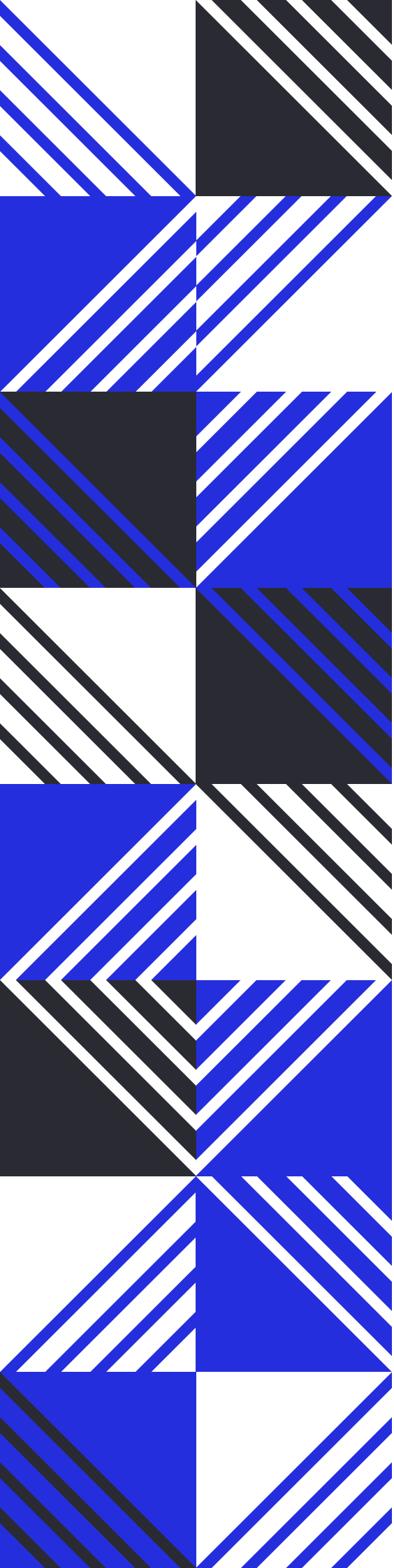
We expressly disclaim any liability for any losses or damages arising from the use of this report or from any security breaches that may occur in the future. We also recommend that our clients engage in multiple independent audits and establish a public bug bounty program as additional measures to bolster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