QuillAudits

# AUDIT REPORT

February 2025

For

UMBRA

# Table of Content

# Executive Summary

| | |
|---|---|
| **Project Name** | UMBRA |
| **Project URL** | https://umbra.finance/swap |
| **Overview** | UmbraDEX CLMM (Concentrated Liquidity Market Maker) is a decentralized trading mechanism designed to optimize capital efficiency by allowing liquidity providers to concentrate their liquidity within specific price ranges. It is a fork of Raydium's CLMM, with minimal changes. The core swap logic, liquidity provisioning, and fee mechanisms remain the same as Raydium's implementation. The primary modifications include renaming Raydium to Umbra across the codebase and adding event emissions to track contract interactions better. There are no major improvements in gas efficiency, execution logic, or security mechanisms. The protocol structure remains identical, ensuring compatibility with existing Solana-based DEX infrastructure. |
| **Audit Scope** | The scope of this Audit was to analyze the Umbra Smart Contracts changes review for quality, security, and correctness.<br>differs Review from Raydium's. CLMM:<br>https://github.com/UmbraDEX/umbra-clmm/commit/5d23e9f027636c26d94497c967b241fa7039649a |
| **Commit Hash** | 5d23e9f027636c26d94497c967b241fa7039649a |
| **Language** | Rust |
| **Blockchain** | Eclipse |

# Executive Summary

| | |
|---|---|
| **Method** | Manual Analysis, Functional Testing, Automated Testing |
| **Review 1** | 10th February 2025 - 11th February 2025 |

**Summary of Changes in CLMM (*Commit*)**

1. Name Replacement
    - All occurrences of Raydium have been replaced with Umbra in function names, comments, and documentation.
2. Event Emissions
    - Additional events have been introduced in key functions, likely to improve on-chain monitoring and analytics.
3. No Functional or Logic Changes
    - There are no modifications in the core logic, math, or liquidity handling.
    - The protocol retains the same architecture as Raydium's CLMM.

# Number of Issues per Severity

0

Total Issues

| | |
|---|---|
| 🟥 High | 0 (0%) |
| 🟧 Medium | 0 (0%) |
| 🟩 Low | 0 (0%) |
| 🟪 Informational | 0 (0%) |

Severity

Issues

| | 🟥 High | 🟧 Medium | 🟩 Low | 🟪 Informational |
|---|---|---|---|---|
| Open | 0 | 0 | 0 | 0 |
| Resolved | 0 | 0 | 0 | 0 |
| Acknowledged | 0 | 0 | 0 | 0 |
| Partially Resolved | 0 | 0 | 0 | 0 |

# Checked Vulnerabilities

We have scanned the solana program for commonly known and more specific vulnerabilities. Here are some of the commonly known vulnerabilities that we considered:

- ✓ Signer authorization
- ✓ Account data matching
- ✓ Sysvar address checking
- ✓ Owner checks
- ✓ Type cosplay
- ✓ Initialization
- ✓ Arbitrary cpi
- ✓ Duplicate mutable accounts
- ✓ Bump seed canonicalization
- ✓ PDA Sharing
- ✓ Incorrect closing accounts
- ✓ Missing rent exemption checks
- ✓ Arithmetic overflows/underflows
- ✓ Numerical precision errors
- ✓ Solana account confusions
- ✓ Casting truncation
- ✓ Insufficient SPL token account verification
- ✓ Signed invocation of unverified programs

# Techniques and Methods

**Throughout the audit of Rust Programs, care was taken to ensure:**

- The overall quality of code
- Use of best practices
- Code documentation and comments, match logic and expected behavior
- Token distribution and calculations are as per the intended behavior mentioned in the whitepaper
- Implementation of various token standards.
- Efficient use of gas
- Code is safe from re-entrancy and other vulnerabilities

**The following techniques, methods, and tools were used to review all the smart contracts:**

### Structural Analysis

In this step, we have analysed the design patterns and structure of Solana programs. A thorough check was done to ensure the Solana program is structured in a way that will not result in future problems.

### Static Analysis

Static analysis of Solana programs was done to identify contract vulnerabilities. In this step, a series of automated tools are used to test the security of Solana programs.

# Techniques and Methods

### Code Review / Manual Analysis

Manual Analysis or review of code was done to identify new vulnerabilities or verify the vulnerabilities found during the static analysis. Contracts were completely manually analysed, and their logic was checked and compared with the one described in the whitepaper. Besides, the results of the automated analysis were manually verified.

### Gas Consumption

In this step, we have checked the behaviour of Solana programs in production. Checks were done to know how much gas gets consumed and the possibilities of optimising code to reduce gas consumption.

# Types of Severity

Every issue in this report has been assigned to a severity level. There are four levels of severity, and each of them has been explained below.

### 🟥 High Severity Issues

A high severity issue or vulnerability means that your smart contract can be exploited. Issues on this level are critical to the smart contract’s performance or functionality, and we recommend these issues be fixed before moving to a live environment.

### 🟧 Medium Severity Issues

The issues marked as medium severity usually arise because of errors and deficiencies in the smart contract code. Issues on this level could potentially bring problems, and they should still be fixed.

### 🟩 Low Severity Issues

Low-level severity issues can cause minor impact and are just warnings that can remain unfixed for now. It would be better to fix these issues at some point in the future.

### 🟪 Informational

These are four severity issues that indicate an improvement request, a general question, a cosmetic or documentation error, or a request for information. There is low-to-no impact.

# Types of Issues

**Open**

Security vulnerabilities identified that must be resolved and are currently unresolved.

**Resolved**

These are the issues identified in the initial audit and have been successfully fixed.

**Acknowledged**

Vulnerabilities which have been acknowledged but are yet to be resolved.

**Partially Resolved**

Considerable efforts have been invested to reduce the risk/impact of the security issue, but are not completely resolved.

## High Severity Issues

No Issues Found.

## Medium Severity Issues

No Issues Found.

## Low Severity Issues

No Issues Found.

## Informational Issues

No Issues Found.

# Closing Summary

In this report, we have considered the security of Changes made in the UMbra Codebase. We performed our audit according to the procedure described above.

No Issues Found.

# Disclaimer

QuillAudits Smart contract security audit provides services to help identify and mitigate potential security risks in Umbra CLMM. However, it is important to understand that no security audit can guarantee complete protection against all possible security threats. QuillAudits audit reports are based on the information provided to us at the time of the audit, and we cannot guarantee the accuracy or completeness of this information. Additionally, the security landscape is constantly evolving, and new security threats may emerge after the audit has been completed.

Therefore, it is recommended that multiple audits and bug bounty programs be conducted to ensure the ongoing security of Umbra CLMM. One audit is not enough to guarantee complete protection against all possible security threats. It is important to implement proper risk management strategies and stay vigilant in monitoring your smart contracts for potential security risks.

QuillAudits cannot be held liable for any security breaches or losses that may occur subsequent to and despite using our audit services. It is the responsibility of  Umbra CLMM to implement the recommendations provided in our audit reports and to take appropriate steps to mitigate potential security risks.

# About QuillAudits

QuillAudits is a leading name in Web3 security, offering top-notch solutions to safeguard projects across DeFi, GameFi, NFT gaming, and all blockchain layers. With six years of expertise, we've secured over 1000 projects globally, averting over $30 billion in losses. Our specialists rigorously audit smart contracts and ensure DApp safety on major platforms like Ethereum, BSC, Arbitrum, Algorand, Tron, Polygon, Polkadot, Fantom, NEAR, Solana, and others, guaranteeing your project's security with cutting-edge practices.

QuillAudits

**6+**

Years of Expertise

**1M+**

Lines of Code Audited

**$30B+**

Secured in Digital Assets

**1K+**

Projects Secured

Follow Our Journey

# AUDIT REPORT

February 2025

For

UMBRA

QuillAudits

Canada, India, Singapore, UAE, UK

www.quillaudits.com audits@quillhash.com