OFFSIDE LABS

DLMM 0.8.2

Smart Contract Security Assessment

November 2024

Prepared for:

Meteora

Prepared by:

Offside Labs

Sirius Xie

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 *DLMM* smart contracts, starting on Nov 12, 2024, and concluding on Nov 15, 2024.

Project Overview

This update primarily expands the flexibility in creating and configuring *Alpha Vault* related vaults and pools in *DLMM*.

DLMM adds a new endpoint allows pool creator to be able to create pool with input customizable parameters.

Audit Scope

The assessment scope contains mainly related changes of lb-clmm program for the *DLMM* project.

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

- DLMM
  - Codebase: https://github.com/MeteoraAg/DLMM
  - Commit Hash: f8c0d959b018f9ec8124f05fb64836c08236338c
  - Branch: feat/init-lb-pair-with-vault-binding

We listed the files we have audited below:

- DLMM PR-318

Findings

The security audit revealed:

- 0 critical issue
- 1 high issues
- 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	Missing Mint Validation in init_customizable _permissionless_lb_pair	High	Fixed
02	InitializePositionByOperator Missing Operator Validation	Medium	Fixed
03	Calculation Error in Pre Activation Start Point Update Validation	Low	Fixed
04	Technical Typo	Informational	Fixed
05	Recommend Sufficient Buffer for Settings Regarding Activation Point	Informational	Fixed

4 Key Findings and Recommendations

4.1 Missing Mint Validation in init_customizable_permissionless_lb_pair

Severity: High	Status: Fixed
Target: Smart Contract	Category: Data Validation

Description

In DLMM, the `InitializeCustomizablePermissionlessLbPair` contains a TokenAccount (`user_token_x`), and the corresponding IX verifies that `user_token_x.amount > 0` to confirm that the `lb_pair` creator holds ownership of the token as the token creator on `token_mint_x` .

```
238     // prove the ownership of token creator
239     require!(
240         ctx.accounts.user_token_x.amount > 0,
241         LBError::MissingTokenAmountAsTokenLaunchProof
242     );
```

programs/lb_clmm/src/instructions/initialize_pool/initialize_customizable_permissionless_lb_pair.rs#L238-L242

However, this instruction only checks the authority of this `TokenAccount` without imposing any restrictions on the mint, allowing a `TokenAccount` with an arbitrary mint to be passed in, as long as the authority is the funder.

```
206     #[account(
207         token::authority = funder,
208     )]
209     pub user_token_x: Box<InterfaceAccount<'info, TokenAccount>>,
```

programs/lb_clmm/src/instructions/initialize_pool/initialize_customizable_permissionless_lb_pair.rs#L206-L209

Impact

Assume the token creator has created a new token. If a malicious user discovers this new token, they can create a `user_token_x` for the token creator with an amount greater than 0 on a mint controlled by the malicious user. Given that this instruction is permissionless, the malicious user can preempt the token creator and create a customizable permissionless `lb_pair` for this token in DLMM, setting the parameters to malicious values.

Since the seeds for a customizable permissionless `lb_pair` are based solely on `token_mint_x` and `token_mint_y` , the token creator would be unable to create the intended customizable permissionless `lb_pair` in DLMM.

Recommendation

It is recommended to add a new constraint for `user_token_x` :

```
token::mint = token_mint_x
```

Mitigation Review Log

Fixed in the commit **8a5aa373544ecb29f2105a16c70b66416ecf042f**.

4.2 InitializePositionByOperator Missing Operator Validation

Severity: Medium	Status: Fixed
Target: Smart Contract	Category: Data Validation

Description

Due to the removal of the `whitelisted_wallet` field of LbPair, the PermissionLbPairActionAccess.validate_initialize_position_by_operator method in the current implementation will no longer validate the operator.

Therefore, any signer can call the `initialize_position_by_operator` instruction to init a position for any other users with customized `fee_owner` and `lock_release_point` .

Impact

This increases phishing risks, as attackers may exploit the instruction to create empty positions for high net-worth users and attempt to capture their LP fees.

Recommendation

Add risk warning in the frontend.

Mitigation Review Log

Fixed in the commit **8a5aa373544ecb29f2105a16c70b66416ecf042f**.

4.3 Calculation Error in Pre Activation Start Point Update Validation

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

Description

```
114 let new_pre_activation_start_point =
    ↪ new_activation_point.safe_sub(self.time_buffer)?;
```

programs/lb_clmm/src/pair_action_access/permission_pair.rs#L114

`new_pre_activation_start_point` should be

`new_activation_point - pre_activation_duration` .

Impact

Due to `pre_activation_duration >= self.time_buffer` , this might cause the pool to potentially enter pre-activation phase immediately.

Recommendation

Use `self.pre_activation_duration` instead of `self.time_buffer` .

Mitigation Review Log

Fixed in the commit **8a5aa373544ecb29f2105a16c70b66416ecf042f**.

4.4 Informational and Undetermined Issues

Technical Typo

Severity: Informational	Status: Fixed
Target: Smart Contract	Category: Code QA

1. For consistency with internal implementation naming, it's better to rename the name of new instruction from `initialize_customizable_permission_lb_pair` to `initialize_customizable_permissionless_lb_pair` . This lb_pair is permissionless rather than permissioned. programs/lb_clmm/src/lib.rs#L147-L152
2. Typo in comments: programs/lb_clmm/src/constants.rs#L116-L116 , it should be

`(10^9 * 10 / 100)`

Recommend Sufficient Buffer for Settings Regarding Activation Point

Severity: Informational	Status: Fixed
Target: Smart Contract	Category: Data Validation

Since adjusting `LbPair.pre_activation_duration` affects pre_activation_start_point, it is recommended to include buffer checks in the PermissionLbPairActionAccess.validate_set_pre_activation_duration method to maintain consistency with other

validations. For instance, in the PermissionLbPairActionAccess.validate_update_new_activation_point method, `new_pre_activation_start_point` should be greater than `current_point + time_buffer (1 hour)` .

Alternatively, a minimum 5-minute buffer should be maintained to ensure the alpha vault's last join point has not been reached.

When creating `CustomizablePermissionless` pairs, it's also recommended to standardize buffer validation checks instead of only checking `vault_last_join_point` :

DLMM/../programs/lb_clmm/src/instructions/initialize_pool/initialize_customizable_permissionless_lb_pair.rs#L127.

Similar validation also exists in Dynamic AMM:

meteora-dynamic-amm/../programs/amm/src/instructions/initialize_pool/initialize_customizable_permissionless_pool.rs#L281-L281

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.

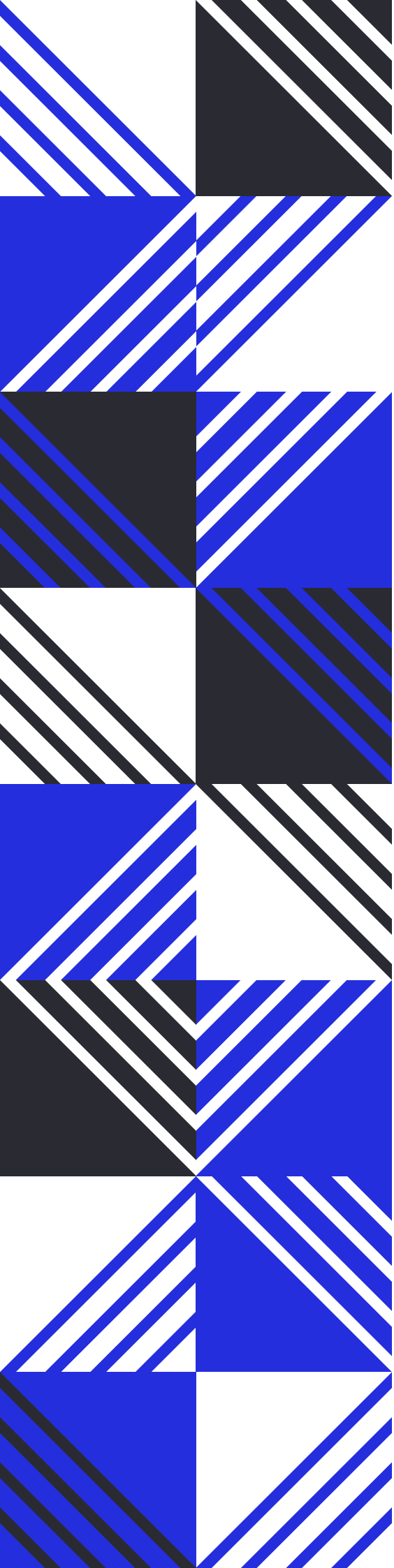
We expressly disclaim any liability for any losses or damages arising from the use of this report or from any security breaches that may occur in the future. We also recommend that our clients engage in multiple independent audits and establish a public bug bounty program as additional measures to bolster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