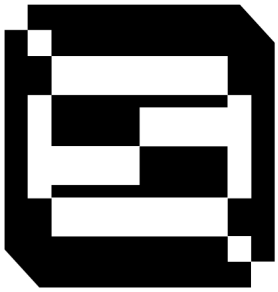

OFFSIDE Labs

Alpha Vault

Smart Contract Security Assessment

May 2024

Prepared for:

Meteora

Prepared by:

Offside Labs

Ronny Xing

Siji Feng

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 *Alpha Vault* smart contracts, starting on May 23, 2024, and concluding on May 25, 2024.

Project Overview

Meteora's *Alpha Vault*, is a new anti-bot tool to guard against sniper bots and allow genuine supporters to be the first to buy tokens at launch.

It helps projects ensure fairer token launches for their community. Vault depositors enjoy the same average price and tokens are only locked for a day or more and vested for a short period.

Audit Scope

The assessment scope contains mainly the smart contracts of the program for the *Alpha Vault* project, and 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
  - Branch: feat/support-snipping-vault
  - Commit Hash: c9a045f9008a811448818584e5edee46986bf926
  - Codebase Link
- Alpha Vault
  - Branch: main
  - Commit Hash: 4d091f1bc3080ad138b8f47adb0d52e7d54538b3
  - Codebase Link

We listed the files we have audited below:

- DLMM PR-269
- Alpha Vault:
  - programs/dlmm-vault/src/*.rs

Findings

The security audit revealed:

- 0 critical issue
- 0 high issues
- 1 medium issues
- 2 low issues
- 3 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	max_cap Should Never be Zero	Medium	Fixed
02	is_over_max_cap Condition for escrow.closable is Not Sufficient	Low	Fixed
03	Recommended to Add a Minimum Check to the max_amount of fill_dlmm	Low	Fixed
04	Condition for Slot is Inaccurate in validate_fill_dlmm	Informational	Fixed
05	DLMM has Removed the Swap Restriction for the whitelisted_wallet	Informational	Acknowledged
06	Reminder About Precision Residues	Informational	Acknowledged

4 Key Findings and Recommendations

4.1 max_cap Should Never be Zero

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

Description

Suppose a scenario where the admin initialized a vault, and temporarily set the `max_cap` to 0 due to uncertainty about the initial liquidity of the pool, with plans to update this value later.

This would result in the `slot.validate_withdraw_remaining_quote` check in the `withdraw_remaining_quote` instruction being bypassed because the `dlmm_vault.get_swappable_amount` method returns 0.

```
let swappable_amount = dlmm_vault.get_swappable_amount(u64::MAX)?;
if swappable_amount != 0 {
    ...
}
```

Impact

In this scenario, an attacker can deposit quote token in advance and then immediately use the `withdraw_remaining_quote` instruction to withdraw. However, after the subsequent swap process, with a updated `max_cap`, is completed, they would hold the same share of the `total_claimable_token`. This effectively amounts to stealing the purchased base tokens.

Recommendation

Ensure that `max_cap` is never 0 in the `initialize_vault` and `update_vault_parameters` instructions.

Mitigation Review Log

Meteora Team: PR-12

Offside Labs: **Fixed**.

4.2 is_over_max_cap Condition for escrow.closable is Not Sufficient

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

Description

`close_escrow` instruction checks if the current escrow is able to be closed by the `closable` function. In cases where there are remaining quote tokens, it is essential to ensure that the escrow has been refunded.

```
        if is_over_max_cap {
            // if it is over max cap, user need to claim refund quote
             ↪  token
            self.refunded == 1
        } else {
            true
        }
```

However, `is_over_max_cap` is not a sufficient condition for this scenario. There is an edge case where, even if `total_deposit <= max_cap`, unexpected situations like insufficient liquidity in the pool could result in `swapped_amount < total_deposit` after the `last_buying_slot`.

Impact

Users will lose any remaining quote tokens that were not refunded.

Recommendation

Check if `swapped_amount < total_deposit`.

Mitigation Review Log

Meteora Team: PR-12

Offside Labs: **Fixed**.

4.3 Recommended to Add a Minimum Check to the max_amount of fill_dlmm

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

Description

`fill_dlmm` IX is permissionless, which means anyone can invoke this instruction to perform a swap after the `pre_activation_start_slot` . The `max_amount` parameter will define the maximum amount of quote tokens for this swap.

The issue is that, without a minimum check for `max_amount` and with the slippage check not applicable in this scenario, numerous dust level swaps could amplify fees and precision losses.

Impact

For instance, if the initial price is quote:base at 100:199 and `max_amount` is set to 2 lamports, the swap price would be 200:100. In this scenario, 1 lamport of quote token would cover the fee, and 1 lamport of base token would be lost due to precision.

Recommendation

Add a minimum check to the `max_amount` of `fill_dlmm` .

Mitigation Review Log

Meteora Team: PR-12

Offside Labs: Fixed.

4.4 Informational and Undetermined Issues

Condition for Slot is Inaccurate in validate_fill_dlmm

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

The `current_slot` should be `>= pre_activation_swap_start_slot` instead of `> last_join_slot` . That is the condition to make sure the swap is activated in the `PermissionLbPairActionAccess.pre_swap_activated` of the *DLMM*:

```
pre_swap_activated: current_slot >= pre_activation_swap_start_slot,
```

DLMM has Removed the Swap Restriction for the whitelisted_wallet

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

Prior to the PR update, whitelist addresses encountered two restrictions when swapping via `add_liquidity` :

1. Swapping before `activated` is not significant since there are currently only two whitelisted users.
2. After `throttled`, swaps are also restricted by the `max_swapped_amount`.

However, following the changes in the PR, these restrictions have been lifted:

1. During the `pre_activation_slot_duration`, `whitelisted_wallet` does not have the permission to swap but can still perform swaps through `add_liquidity`. At this time, the pool already has liquidity provided by the Vault acting as a counterparty.
2. The `max_swapped_amount` restriction in the `add_liquidity` instruction has been entirely removed.

Mitigation Review Log: *DLMM* has removed `max_swap_amount` and throttled duration, which is by design.

Reminder About Precision Residues

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

Because `claimable_amount` and `refund_quote_token` are both calculated by floor rounding according to the ratio of `deposited_escrow_amount` to `total_deposit`, there will inevitably be precision residues left in the vault. The remaining amounts of quote and base tokens might approximately equal the number of participating escrows.

Mitigation Review Log: We will handle residues later (if there are much dust).

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.