OFFSIDE LABS

Alpha Vault 0.4.0

Smart Contract Security Assessment

April 2025

Prepared for:

Meteora

Prepared by:

Offside Labs

Sirius Xie

Ronny Xing

Contents

1 About Offside Labs

Offside Labs stands as a pre-eminent security research team, comprising highly skilled hackers with top - tier talent from both academia and industry.

The team demonstrates extensive and diverse expertise in modern software systems, which encompasses yet are not restricted to *browsers*, *operating systems*, *IoT devices*, and *hypervisors*. Offside Labs is at the forefront of innovative domains such as *cryptocurrencies* and *blockchain technologies*. The team achieved notable accomplishments including the successful execution of remote jailbreaks on devices like the **iPhone** and **PlayStation 4**, as well as the identification and resolution of critical vulnerabilities within the **Tron Network**.

Offside Labs actively involves in and keeps contributing to the security community. The team was the winner and co-organizer for the *DEFCON CTF*, the most renowned CTF competition in Web2. The team also triumphed in the **Paradigm CTF 2023** in Web3. Meanwhile, the team has been conducting responsible disclosure of numerous vulnerabilities to leading technology companies, including *Apple*, *Google*, and *Microsoft*, safeguarding digital assets with an estimated value exceeding **$300 million**.

During the transition to Web3, Offside Labs has attained remarkable success. The team has earned over **$9 million** in bug bounties, and **three** of its innovative techniques were acknowledged as being among the **top 10 blockchain hacking techniques of 2022** by the Web3 security community.

2 Executive Summary

Introduction

Offside Labs completed a security audit of *Alpha Vault* smart contracts, starting on April 16th, 2025, and concluding on April 24, 2025.

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.

This update upgrades Anchor to version 0.31.0, adds support for Token-2022, and integrates with DAMM v2.

Audit Scope

The assessment scope contains mainly the smart contracts of the alpha-vault program for the *Alpha Vault* project. The audit is based on the following specific branches and commit hashes of the codebase repositories:

- Alpha Vault
  - Codebase: `https://github.com/MeteoraAg/alpha-vault`
  - Commit Hash: 9b78346a41b3a73c5f1323452c14e328b89a88bb

We listed the pull request we have audited below:

- Alpha Vault PR-71

Findings

The security audit revealed:

- 0 critical issue
- 0 high issue
- 0 medium issue
- 1 low issues
- 1 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	Unable to Deposit Final Dust into Vault When Transfer Fee Is Enabled	Low	Fixed
02	Missing Memo Support for Token-2022 Transfers	Informational	Fixed

4 Key Findings and Recommendations

4.1 Unable to Deposit Final Dust into Vault When Transfer Fee Is Enabled

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

Description

In the `deposit` IX, the amount a user can deposit is calculated based on the current state of the vault and escrow, as well as the `max_amount` parameter. This amount is then transferred into the `token_vault` via a `transfer`, and the vault records the actual amount of tokens received.

```rust
    anchor_spl::token_2022::transfer_checked(
        CpiContext::new(...),
        amount,
        ctx.accounts.token_mint.decimals,
    )?;

    // exclude transfer fee
    let excluded_transfer_fee_amount =
        calculate_transfer_fee_excluded_amount(&ctx.accounts.token_mint, amount)?.amount;

    vault.deposit(&mut escrow, excluded_transfer_fee_amount)?;
```

programs/alpha-vault/src/instructions/deposit.rs#L51-L69

If the `token_mint` supports the Token-2022 Transfer Fee Extension, the amount recorded in the vault will be the net amount after the transfer fee is deducted. However, this logic does not account for dust. For example, if the amount passed to `transfer_checked` is 1 and the token mint's transfer fee is greater than 0, the resulting `excluded_transfer_fee_amount` will be 0. After the `deposit` IX finishes, no tokens will actually be deposited into the vault. But the user's token account balance will be reduced by 1.

In addition, during the amount calculation, `fee-included` and `fee-excluded` values are mixed together in the computation.

```
    let amount = match vault_mode {
        VaultMode::Prorata => max_amount,
        VaultMode::Fcfs => {
            let remaining_quota_quote =
             ↪  vault.get_remaining_quote_in_fcfs_vault()?;
            if remaining_quota_quote > max_amount {
                max_amount
            } else {
                remaining_quota_quote
            }
        }
    };
    let remaining_quota = escrow.get_remaining_quota(&vault)?;
    let amount = amount.min(remaining_quota);
    require!(amount > 0, VaultError::DepositAmountIsZero);
```

programs/alpha-vault/src/instructions/deposit.rs#L36-L49

The `max_amount` is provided by the user and is generally a `fee-included` amount, whereas `remaining_quota_quote` and `remaining_quota` are both `fee-excluded` amounts. Comparing these two types of values directly may result in an inaccurate final amount.

Impact

If the vault's current deposited amount is just a small dust amount away from `max_cap` , the current implementation could cause that max_cap to be permanently unreachable. Users attempting to deposit tokens to fill this gap will find their deposits never counted toward the vault's deposited amount.

Recommendation

It is recommended that, when the Transfer Fee Extension is enabled, the code explicitly distinguishes whether each intermediate amount in the calculation includes the fee. Only then should values be compared or used together. This ensures the final computed amount is accurate and allows even small dust amounts to be successfully deposited into the vault.

Mitigation Review Log

Fixed in the commit eb09dedadf95eb05886f7b8b52b41a095d876657.

4.2 Informational and Undetermined Issues

Missing Memo Support for Token-2022 Transfers

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

`Vault.token_vault` might be a Token-2022 Mint. In the `withdraw` , `withdraw_remaining_quote` , and `claim_token` IXs, the alpha vault transfers tokens from `token_vault` to the user. In Token-2022, users can choose to enable the `Memo` extension on their token accounts, which requires a separate Memo instruction to be included before any transfer. And whether the user enables Memo is independent of the Mint. However, the `transfer_to_user` function currently does not handle logic related to the Memo extension. It is recommended to add support for Memo to prevent transfer failures caused by this requirement.

5 Disclaimer

This report reflects the security status of the project as of the date of the audit. It is intended solely for informational purposes and should not be used as investment advice. Despite carrying out a comprehensive review and analysis of the relevant smart contracts, it is important to note that Offside Labs' services do not encompass an exhaustive security assessment. The primary objective of the audit is to identify potential security vulnerabilities to the best of the team's ability; however, this audit does not guarantee that the project is entirely immune to future risks.

Offside Labs disclaims any liability for losses or damages resulting from the use of this report or from any future security breaches. The team strongly recommends that clients undertake multiple independent audits and implement a public bug bounty program to enhance the security of their smart contracts.

The audit is limited to the specific areas defined in Offside Labs' engagement and does not cover all potential risks or vulnerabilities. Security is an ongoing process, regular audits and monitoring are advised.

Please note: Offside Labs is not responsible for security issues stemming from developer errors or misconfigurations during contract deployment and does not assume liability for centralized governance risks within the project. The team is not accountable for any impact on the project's security or availability due to significant damage to the underlying blockchain infrastructure.

By utilizing this report, the client acknowledges the inherent limitations of the audit process and agrees that the firm shall not be held liable for any incidents that may occur after the completion of this audit.

This report should be considered null and void in case of any alteration.

OFFSIDE LABS

https://offside.io/

https://github.com/offsidelabs

https://twitter.com/offside_labs