OFFSIDE LABS

Dynamic Vault

Smart Contract Security Assessment

September 2024

Prepared for:

Meteora

Prepared by:

Offside Labs

Yao Li

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 *Dynamic Vault* smart contracts, starting on September 22, 2024, and concluding on September 29, 2024.

Project Overview

Meteora Dynamic Vaults is a DeFi infrastructure that optimizes yield by rebalancing vaults across lending platforms every minute, prioritizing both returns and accessibility of user funds. A trusted capital allocation layer mitigates risks, efficiently distributing liquidity while ensuring safety through monitored utilization rates and reserve levels. Vaults automatically withdraw funds when risk thresholds are met, and allocation limits are based on audits, open-source code, and insurance coverage. These security measures will evolve with community input, driving innovation and growth in the ecosystem.

Audit Scope

The assessment scope contains mainly the smart contracts of the *vault* program for the *Dynamic Vault* project.

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

- Dynamic Vault
  - Branch: main
  - Commit Hash: ef240c488838d327e92e55ec24e659deb66d3bf8
  - Codebase Link

We listed the files we have audited below:

- Dynamic Vault
  - programs/vault/src/*.rs

Findings

The security audit revealed:

- 0 critical issue
- 0 high issues
- 1 medium issues
- 3 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	Performance Fee Minting Causes Instant LP Price Drop	Medium	Fixed
02	Missing Check for collateral_vault in CollectDust	Low	Fixed
03	LP Instant Price May Change after update_locked_profit_degradation	Low	Fixed
04	Accumulated Rounding Loss by Users through withdraw_directly_from_strategy	Low	Fixed
05	Missing Type Check for Vault in withdraw2	Informational	Fixed
06	Inflation Attack Under the Edge Case in deposit Instruction	Informational	Fixed

4 Key Findings and Recommendations

4.1 Performance Fee Minting Causes Instant LP Price Drop

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

Description

At the end of the `rebalance_strategy_wrapper` method, a portion of the profit gained is minted as LP tokens, serving as the performance fee.

```
    let lp_mint_more =
        calculate_performance_fee(total_amount_before,
↪   total_amount_after, total_lp_supply)
            .ok_or(VaultError::MathOverflow)?;
```

programs/vault/src/instructions/remove_strategy.rs#L237-L239

Since the total amount of underlying asset tokens is calculated based on the `total_amount` , which is `total_amount_after - performance_fee` , while the current actual value of LP tokens is calculated based on the unlocked amount, this leads to an instantaneous drop in LP price after the `performance_fee` LP minting.

Even worse, within the `withdraw_directly_from_strategy` instruction, after this LP is minted, `lp_mint` is not reloaded. This will cause the `unmint_amount` to be calculated using the larger LP price from before the rebalance.

Impact

The instantaneous drop in Vault LP price may lead to potential manipulation of the invariant in the AMM. When the Vault LP price drops, the corresponding underlying asset price in the AMM curve will increase. This will open up some arbitrage opportunities, potentially causing slight losses for Vault LP holders.

Recommendation

If we want to mint LP tokens while ensuring the LP price remains constant, we can only immediately unlock a portion of the profit as a portion of performance fee for minting. However, if we unlock all the performance fee at once, the LP tokens minted from this fee will share in more of the locked profit. To ensure both that the LP price remains unchanged and that the performance fee is not excessively collected, we solve the following equation to determine the proportion of performance fee to unlock immediately.

Let:

1. x: fee_unlock_immediately

2. U: unlocked_amount
3. S: lp_original_supply
4. m: minting_fee_lp
5. P: profit
6. R: performance_fee_rate
7. F: performance_fee

Derivation:

a) Solve for minting_fee_lp(m):

$$\frac{x+U}{S+m}=\frac{U}{S}$$

—>

$$m=\frac{xS}{U}$$

i.e.

$$\text{minting\_fee\_lp}=\frac{\text{fee\_unlock\_immediately}}{\text{unlocked\_amount}}\times\text{lp\_original\_supply}$$

b) Solve for fee_unlock_immediately(x):

$$\begin{cases}F-x=\frac{(P-x)\times m}{S+m}\\ P\times R=F\end{cases}$$

—>

$$P\times R-x=\frac{(P-x)\times x\times S}{U\times\left(S+\frac{x\times S}{U}\right)}$$

—>

$$x=\frac{F\times U}{P+U-F}$$

—>

$$x=\frac{F\times U}{P+U-F}$$

i.e.

$$\text{fee\_unlock\_immediately}=\frac{\text{unlocked\_amount}}{\text{profit}+\text{unlocked\_amount}-\text{fee}}\times\text{fee}$$

In addition, after calling the `rebalance_strategy_wrapper` method, it needs to add `lp_mint.reload()` in the `withdraw_directly_from_strategy` instruction. This is because `performance_fee` has minted more LP tokens, so `lp_mint.supply` has increased at this point programs/vault/src/instructions/deposit_withdraw_liquidity.rs#L370-L373, which would lead to an incorrect calculation of `unmint_amount` .

Mitigation Review Log

Meteora Team:

Fixed in relevant code implementation.

Offside Labs: Fixed.

4.2 Missing Check for collateral_vault in CollectDust

Severity: Low	**Status: Fixed**
Target: Smart Contract	**Category: Data Validation**

Description

In the `CollectDust` instruction, there is a constraint to ensure that `token_vault.key()` is not equal to `vault.token_vault` to prevent the admin from transferring user funds.

```
    pub struct CollectDust<'info> {
        /// vault
        #[account(has_one = admin)]
        pub vault: Box<Account<'info, Vault>>,
        /// Token vault, must be different from vault.token_vault
        #[account(mut, constraint = token_vault.key() !=
          vault.token_vault)]
        pub token_vault: Box<Account<'info, TokenAccount>>,
        ...
    }
```

programs/vault/src/instructions/collect_dust.rs#L6-L12

However, there is no check to ensure that `token_vault.key()` is not equal to `collateral_vault` of a strategy. This could potentially allow the admin to transfer the `collateral_vault` by using the `CollectDust` instruction, and only `SolendWithoutLM` strategy has vaulable `collateral_vault` .

Impact

An admin may be able to transfer the `collateral_vault` of a `SolendWithoutLM` strategy by the `CollectDust` instruction.

Recommendation

Add an additional check to ensure `token_vault.key()` is not equal to `collateral_vault` of a strategy.

Mitigation Review Log

Meteora Team:

We have removed the `CollectDust` instruction.

- relevant code implementation

Offside Labs: Fixed.

4.3 LP Instant Price May Change after update_locked_profit_degradation

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

Description

In the `update_locked_profit_degradation` instruction, only `locked_profit_degradation` is updated. If there is any locked profit with `last_report < current_time` at the time of execution, the unlocked amount will change immediately after the update, leading to an instant change in the LP price.

Impact

This could result in unintended LP price fluctuations, similar to the impact in the issue *"Performance Fee Minting Causes Instant LP Price Drop"*.

Recommendation

Before updating the `locked_profit_degradation` , refresh the locked profit by calling `update_locked_profit` .

Mitigation Review Log

Meteora Team:

Fixed in relevant code implementation

Offside Labs: Fixed.

4.4 Accumulated Rounding Loss by Users through withdraw_directly_from_strategy

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

Description

In the `rebalance_strategy_wrapper` , there may be a mismatch between the difference in `liquidity_in_vault` and `liquidity_in_strategy` due to rounding losses inherent in the strategy. Each time the vault unwraps the corresponding ctoken from a strategy, it will incur a precision loss of no more than 1 underlying token amount. The vault currently absorbs this rounding loss.

Users can exploit this by frequently invoking the `withdraw_directly_from_strategy` instruction, which triggers the `rebalance_strategy_wrapper` and may introduce a rounding loss with 1 token amount. Over time, these losses accumulate, and the vault, rather than the users who invoke the instruction, bears the cumulative effect of the rounding loss.

Impact

Frequent use of `withdraw_directly_from_strategy` by users could lead to material rounding losses for the vault.

Recommendation

Modify the `withdraw_directly_from_strategy` instruction to ensure that users, rather than the vault, bear the rounding losses. In fact, users also suffer some precision loss each time they unmint vault LP, but the current implementation cannot guarantee that this loss (which is effectively a donation) is not less than the precision loss suffered by the vault when withdrawing from the strategy.

Mitigation Review Log

Meteora Team:

Fixed in relevant code implementation

Offside Labs: Fixed.

4.5 Informational and Undetermined Issues

Missing Type Check for Vault in withdraw2

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

In the `withdraw2` instruction, whether to call `withdraw_directly_from_strategy` or `withdraw` is based on `remaining_account` . The `withdraw_directly_from_strategy` function requires the vault type to be `VaultType::RebalanceVault` .

The code attempts to call `withdraw_directly_from_strategy` when `remaining_account` is not empty. However, this is not a CPI call, it won't trigger the check enforced by the constraint in `WithdrawDirectlyFromStrategy` . Besides, `StrategyAccounts.check_accounts` doesn't verify the vault type, which allows passing a vault account with a type other than `VaultType::RebalanceVault` into `withdraw_directly_from_strategy` , bypassing the vault type restriction.

Although this will not have any practical impact, as other types of vaults do not have any strategies, it is still recommended to add a check before `withdraw2` calls `WithdrawDirectlyFromStrategy` .

Mitigation Review Log:

Meteora Team:

We have removed the `withdraw2` instruction.

- relevant code implementation

Offside Labs: Fixed.

Inflation Attack Under the Edge Case in deposit Instruction

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

In the `depsoit` instruction, if `lp_supply == 0` while locked profit exists, a user could deposit a very small amount (e.g., 1 token). Once all the profit is unlocked and no other users have deposited in the interim, the LP price could become significantly inflated, leading to unexpected precision loss.

It's recommended to add a check in the branch with `lp_supply == 0` , to ensure that the minting LP amount(unlocked token amount of the vault after depositing) is not significantly smaller than the locked profit. For example, if the locked profit is 100 times the minting LP amount, then according to the default unlock degradation, after 36 minutes, the LP price will be amplified tenfold, resulting in an unacceptable precision loss.

Mitigation Review Log:

Meteora Team:

Fixed in relevant code implementation

Offside Labs: Fixed.

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