
# Security Audit of Moonfarm Smart Contracts

a report of findings by

Arcadia

*innovative fortuna iuvat*

May 29th, 2021

## Table of Contents

# Document Info

| | |
|---|---|
| Client | Moonfarm |
| Title | Security Audit of Certain Moonfarm Smart Contracts |
| Approved By | Rasikh Morani |

## Contact

For more information on this report, contact The Arcadia Media Group Inc.


| Rasikh Morani |
|---|
| (972) 543-3886 |
| rasikh@arcadiamgroup.com |
| https://t.me/thearcadiagroup |

# Executive Summary

A Representative Party of Moonfarm engaged The Arcadia Group ("Arcadia"), a software development, research, and security company, to conduct a review of the following Moonfarm smart contracts on the Moonfarm repo at Commit #4bfb3fefe3ff37099defc413c449dde42f892023.

Arcadia completed this security review using various methods primarily consisting of dynamic and static analysis. This process included a line-by-line analysis of the in-scope contracts, optimization analysis, analysis of key functionalities and limiters, and reference against intended functionality.

There were 04 issues found, 00 of which were deemed to be ‘critical’, and 02 of which were rated as ‘high’.

| Severity Rating | Number Of Original Occurrences | Number Of Remaining Occurrences |
|---|---|---|
| Critical | 00 | 00 |
| High | 00 | 00 |
| Medium | 01 | 01 |
| Low | 01 | 01 |
| Notice | 00 | 00 |
| Informational | 00 | 00 |

# Findings

## 1. Source code readability.

- MS-1
- Severity: Low
- Impact: Low
- Target:
- Category: Readability

**Problem:** Source code is using a different version of Solidity. There is no specified @openzeppelin contracts version,thus the naming convention should be improved.

- It's very hard to build your source code because of conflicting Solidity versions. We strongly suggest you upgrade your source code to ***^0.6.0*** for better inheritance structure (with **virtual** and **override**).

- If we dive deep into the code, we understand **`y`** stands for LP address pair, **`yfi`** stands for the governance token (MFO). These two variables should be renamed so that other people can more easily understand your code at a glance.

```
IERC20 public y = IERC20(0x0000000000000000000000000000000000000000);
IERC20 public yfi = IERC20(0x0000000000000000000000000000000000000000);
```

- Should add a constructor and get/set to change ***LP address*** and ***governance token***

## 2. Check for a sufficient balance before balance for paying the reward

- MS-2
- Severity: Medium
- Impact: Medium
- Target: RewardPool.sol
- Category: Arithmetic

**Problem:** `notifyRewardAmount` does not transfer the equivalent amount of reward to ***RewardPool;*** instead the owner has to transfer the reward manually via another transaction. Should merge those two transactions into one in order to make sure there's enough rewards to also pay the user.

```
function notifyRewardAmount(uint256 reward)
      external
      onlyRewardDistribution
```

```
        updateReward(address(0))
    {
```

The `leftover` variable is calculated through ***rewardRate*** , if you transfer the reward amount by function ***notifyRewardAmount*** `leftover`can be calculated through ***balanceOf(address(this))***

```
uint256 remaining = periodFinish.sub(block.timestamp);
uint256 leftover = remaining.mul(rewardRate);
// Should be
uint256 leftover = yfi.balanceOf(address(this));
```

# Conclusion

Arcadia identified issues that occurred at hash #4bfb3fefe3ff37099defc413c449dde42f892023.

# Disclaimer

While best efforts and precautions have been taken in the preparation of this document, The Arcadia Group and the Authors assume no responsibility for errors, omissions, or damages resulting from the use of the provided information. Additionally, Arcadia would like to emphasize that the use of Arcadia's services does not guarantee the security of a smart contract or set of smart contracts and does not guarantee against attacks. One audit on its own is not enough for a project to be considered secure; that categorization can only be earned through extensive peer review and battle testing over an extended period.