YFValue Security Analysis

by Pessimistic

This report is public.

Published: September 11, 2020

Abstract

In this report, we consider the security of the YFValue project. Our task is to find and describe security issues in YFV_Stake_v2.sol smart contract of the platform.

Disclaimer

The audit does not give any warranties on the security of the code. One audit cannot be considered enough. We always recommend proceeding with several independent audits and a public bug bounty program to ensure the security of smart contracts. Besides, security audit is not an investment advice.

Summary

In this report, we considered the security of YFV_Stake_v2.sol smart contracts of YFValue project. We performed our audit according to the procedure described below.

The audit showed no vulnerabilities. The code is of good code quality, it avoids many common pitfalls and is well-optimized.

We provide a few recommendations to optimize gas consumption and improve code logic.

General recommendations

We do not have any further recommendations.

Procedure

In our audit, we consider the following crucial features of the code:

1. Whether the code is secure.
2. Whether the code meets best practices.

We perform our audit according to the following procedure:

- manual audit
  - we manually analyze code base for security vulnerabilities
  - we assess overall project structure and quality
- report
  - we reflect all the gathered information in the report

Project overview

Project description

In our analysis we consider YFV_Stake_v2.sol smart contract of YFValue project on Git repository, commit 2cbf5ecefbb835ca7c040737a01d794d9319ea4d.

The total LOC of audited sources is 470.

Manual analysis

The contracts were completely manually analyzed, their logic was checked. Besides, the results of the automated analysis were manually verified. All the confirmed issues are described below.

Critical issues

Critical issues seriously endanger smart contracts security. We highly recommend fixing them.

The audit showed no critical issues.

Medium severity issues

Medium issues can influence project operation in current implementation. We highly recommend addressing them.

The audit showed no medium severity issues.

Low severity issues

Low severity issues can influence project operation in future versions of code. We recommend taking them into account.

Code style

- There are two pairs of functions that are almost identical:
  - `stake()` at line 793 and `stakeOnBehalf()` at line 831.
  - `tokenStake()` at line 608 and `tokenStakeOnBehalf()` at line 614.

  We recommend combining each pair into a single function with unified logic. This will help to avoid possible issues in the future.
- Calling `updateReward` modifier with non-`address(0)` argument should always be coupled with `checkNextEpoch`, otherwise tokens can be stuck on the contract and some users will be unable to get their reward. Thus, we recommend placing `checkNextEpoch` call inside `updateReward` modifier.

Lack of documentation

The purpose of `accumulatedStakingPower` variable is unclear.

Gas optimization

- In function `getReward()`, we recommend using `reward` variable instead of `rewards[msg.sender]` at line 902, as `reward` is a `memory` variable whereas `rewards[msg.sender]` is read from `storage`.
- `updateReward` modifier at line 745 makes two calls of `rewardPerToken()` function: at line 746 and then at line 776. The second call does not change the value of `rewardPerTokenStored`.

Notes

Overpowered role

The owner has the following powers:

- Change the reward for the epoch in range `0` to `DEFAULT_EPOCH_REWARD * 10`.
- Change at any time the fees for staking tokens. However, fees cannot exceed 1%.
- Change the amount of time tokens that should be staked to withdraw them without fee. If the governance disables withdrawing funds before this time and sets time to infinite, then user funds will be stuck on the contract.
- Change reward token contracts.

Note, that the owner of the contract is a governance. Therefore, it has lower chances for its keys to be compromised.

This analysis was performed by Pessimistic:

Pavel Kondratenkov, Security Engineer
Boris Nikashin, Analyst
Alexander Seleznev, CEO

September 11, 2020