Smart Contract Audit for Forthewin

Overlord Security

July 29, 2023

Contents

1 Project Overview

Created by: ForTheWin

Based on: Ethereum

Date Conducted: April, 2023

ForTheWin

Contracts: **FTW-Farm**
Github: `https://github.com/ForTheWinn/FTW-Solidity-Contracts/tree/main/contracts/FTWFarm`
Commit: **03001d**
Programming Language: **Solidity**
Development Env: **solidity** $\wedge$ **0.8.0**

2 Project Introduction

Forthewin ecosystem will create a platform where ordinary users and businesses can easily use both Fungible tokens and NFTs in their daily lives and find more use cases. The motivation is to give everyone the opportunity to create and manage both Fungible tokens and NFTs, help them be successful and allow their tokens to be more heavily adopted into every day life.

3 Findings and Recommendations

3.1 Summary

The following findings and recommendations after analyzing the **FortheWin FARM contract** implementation. Any additional recommendations beyond what any scanning tools supply are included as necessary.

Severity	Number of findings
Critical	0
Medium	0
Low	2
Informational	4

Issue Id	Severity	Title	Category	Fixed
MS-01	Informative	Potential initialization issues	Coding Practices	Fixed
MS-02	Informative	Unnecessary SafeTransfer	Business logic	Fixed
MS-03	Low	Unnecessary UserLPTokenIdMap	Coding Practices	Fixed
MS-04	Low	Potential risk in block.timestamp	Coding Practices	Fixed
MS-05	Informative	Error message	Optimization	Fixed
MS-06	Informative	Error message	Optimization	Fixed

3.2 Low Vulnerabilities

MS-03: Unnecessary UserLPTokenIdMap and removeElement

Unnecessary UserLPTokenIdMap and removeElement

Source Code link

https://github.com/ForTheWinn/FTW-Solidity-Contracts/blob/78b36254d118a5463d2ae68f820c64ed7040b23b/contracts/FTWFarm/FTWFarm.sol#L482-L494

Description

In solidity, there is no built-in map that can be iterated. Therefore FTW author chooses an uint256[] array for storing a user's LP token. But it is very expensive to iterate over an unknown length of array using a for loop.

Solution

Strongly recommend to drop the usage of UserLPTokenIdMap and store the information elsewhere because no on chain read is needed here.

Status

The issue has been confirmed by team and fixed in commit `e905c0c`

MS-04: Potential risk in block.timestamp

Potential risk in block.timestamp

Source Code link

https://github.com/ForTheWinn/FTW-Solidity-Contracts/blob/233e562d85c6fff7059e84c6a7826ab4f47046b5/contracts/FTWFarm/FTWFarm.sol#L316-L342

Description

In solidity, block.timestamp is not a completely safe data

Solution

It is generally recommended to use block.number instead, and approximate dates with expected block heights and time periods.

Status

The issue has been confirmed by team,

3.3 Informational Vulnerabilities

MS-01: Potential initialization issues

Potential initialization issues

Source Code link

https://github.com/ForTheWinn/FTW-Solidity-Contracts/blob/78b36254d118a5463d2ae68f820c64ed7040b23b/contracts/FTWFarm/FTWFarm.sol#L109-L110

Description

As Openzeppelin didn’t implement safety checks now, we recommend doing initialization unchained manually so that when you update the code later, you could notice double-initialization related problems.

Solution

__Ownable_init should be changed to __Ownable_init_unchained __ReentrancyGuard_init should be changed to __ReentrancyGuard_init_unchained

Status

The issue has been confirmed by the team and fixed in commit `233e562`

MS-02: Unnecessary _safeTransfer

Unnecessary _safeTransfer

Source Code link

https://github.com/ForTheWinn/FTW-Solidity-Contracts/blob/78b36254d118a5463d2ae68f820c64ed7040b23b/contracts/FTWFarm/FTWFarm.sol#L451-L468

Description

the using statement gives you the power to use new methods like safeTransfer and safeTransferFrom for an IERC20 object. You need to use it manually and remove own _safeTransferFrom and _safeWithdraw.

Solution

change it to something like IERC20(NEP_ADDRESS).safeTransfer(account, rewardsToHarvest);

Status

The issue has been confirmed by the team and fixed in commit `3f89bf1`

MS-05: Error message

Error message

Source Code link

https://github.com/ForTheWinn/FTW-Solidity-Contracts/blob/233e562d85c6fff7059e84c6a7826ab4f47046b5/contracts/FTWFarm/FTWFarm.sol#L246

Description

In the createPool function, the require statement checks if the pool doesn’t exist, but the error message says "Pool doesn’t exist."

Solution

The error message should be corrected to "Pool already exists."

Status

The issue has been confirmed by the team and fixed in commit `40dc868`

MS-06: Error Message

Error Message

Source Code link

https://github.com/ForTheWinn/FTW-Solidity-Contracts/blob/233e562d85c6fff7059e84c6a7826ab4f47046b5/contracts/FTWFarm/FTWFarm.sol#L153

Description

The error message says "No authotized."

Solution

The error message should be corrected to "No Authorization" or "Not Authorized" or "No authority"

Status

The issue has been confirmed by the team.

4 Conclusion

In this audit, we have analyzed the **Forthewin Farm contract** design and implementation. The current code base is well organized and those identified issues are promptly confirmed and fixed.

Meanwhile, we need to emphasize that smart contracts as a whole are still in an early, but active stage of development. To improve this report, we greatly appreciate any constructive feedbacks or suggestions, on our methodology, audit findings, or potential gaps in scope/coverage.

For more information regarding this audit report, please send email to contact@overlord.wtf