SOLIDIFIED

Audit Report for Origin - July 11, 2022

Summary

Audit Report prepared by Solidified covering the staking and reward smart contracts of the Origin Story NFT platform.

Process and Delivery

Three (3) independent Solidified experts performed an unbiased and isolated audit of the code below. The final debrief took place on July 01, 2022, and the results are presented here.

Audited Files

The source code has been supplied in several source code repositories:

https://github.com/OriginProtocol/nft-launchpad
Commit hash: `069536a9a8392f1d6f9c2a968298eb1f89672268`

Files in scope for this audit:

```
contracts/contracts/staking
├── FeeVault.sol
├── ISeason.sol
├── Season.sol
├── Series.sol
└── proxies.sol
```

Intended Behavior

The audited codebase implements a vault for protocol fees (NFT sale commissions), and related staking fee distribution mechanisms.

Findings

Smart contract audits are an important step to improve the security of smart contracts and can find many issues. However, auditing complex codebases has its limits and a remaining risk is present (see disclaimer).

Users of a smart contract system should exercise caution. In order to help with the evaluation of the remaining risk, we provide a measure of the following key indicators: **code complexity**, **code readability**, **level of documentation**, and **test coverage**.

Note, that high complexity or lower test coverage does not necessarily equate to a higher risk. However, certain bugs are more easily detected in unit testing than in a security audit and vice versa.

Criteria	Status	Comment
Code complexity	Low	-
Code readability and clarity	High	-
Level of Documentation	Medium	-
Test Coverage	High	-

SOLIDIFIED

Audit Report for Origin - July 11, 2022

Issues Found

Solidified found that the Origin contracts contain no critical issues, no major issues and 1 minor issue, and no informational notes.

We recommend issues are amended, while informational notes are up to the team's discretion, as they refer to best practices.

Issue #	Description	Severity	Status
1	Moving on to next season will fail when no new stakes are received in a long time	Minor	Resolved

Critical Issues

No critical issues have been found

Major Issues

No major issues have been found

Minor Issues

1. Moving on to the next season will fail when no new stakes are received in a long time

The function `_acquireStakingSeason()` is triggered from `stake()` and causes seasons to move on to the next season if the current season's end time is exceeded. However, because this process is only triggered when a user provides a new stake, there may be cases in which the end time of the following season in the array has already been exceeded as well. In such a situation, the next season cannot be bootstrapped.
This problem also applies to the claiming process, in which the claiming season is detected using the same process.

Recommendation
Consider iterating over the seasons in both `_acquireStakingSeason()` and `_acquireClaimingSeason()` and advancing to a valid season in one step.

Update
The issue has been resolved by providing a governor-controlled way to bootstrap seasons in case stakes are not received frequently enough.

Informational Notes

No additional notes

Disclaimer

Solidified audit is not a security warranty, investment advice, or an endorsement of Origin or its products. This audit does not provide a security or correctness guarantee of the audited smart contract. Securing smart contracts is a multistep process, therefore running a bug bounty program as a complement to this audit is strongly recommended.

The individual audit reports are anonymized and combined during a debrief process, in order to provide an unbiased delivery and protect the auditors of Solidified platform from legal and financial liability.

Oak Security GmbH