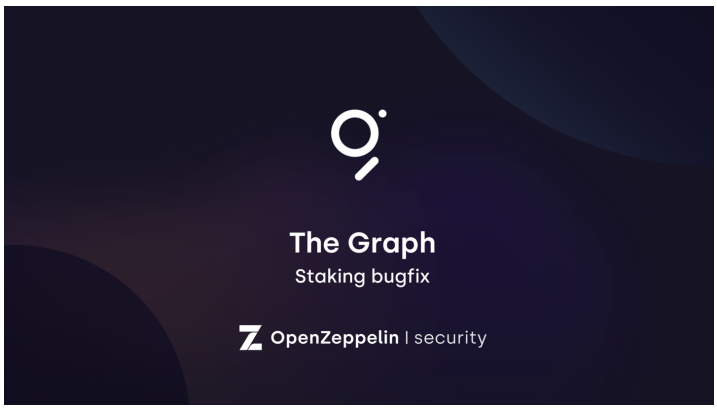
The Graph – Staking Bugfix #2 Audit

APRIL 27, 2021 | IN SECURITY AUDITS | BY OPENZEPPELIN SECURITY



Introduction

The Graph team asked us to audit a bugfix present in the `Staking` contract when an allocation is closed with a `poi=0x0`.

The pull request that we have audited is the PR#459 at commit `168c05b87346e3c0acfbe4abb40ae21d8b39b5e4` and the audited file is:

```
contracts/staking/Staking.sol
```

Overview

Indexers can get rewards for indexing subgraphs, and these rewards are minted when an indexer closes an allocation referencing a non-trivial proof of indexing. These allocations are closed by calling one of many functions within the `Staking` contract that call as a subroutine the `_closeAllocation` function. A valid call to `_closeAllocation` with a non-trivial proof of indexing will update the snapshots since they are used to calculate the minted rewards.

An indexer is able to opt-out of rewards when closing out an allocation by passing in `_poi` set to `0x0`. But in this case, the `takeRewards` method of the `RewardsManager` is not called, and thus the snapshots are not updated before unallocating tokens. This makes it to where the `RewardsManager`'s calculations of various values will use non updated snapshots.

The Graph team made a simple fix addressing this issue by having the `_closeAllocation` function call the `_updateRewards` function in the case the proof of indexing is trivial. As a subroutine of this `_updateRewards` function is a call to the `rewardsManager`'s `onSubgraphAllocation` function. This `onSubgraphAllocation` function is the subroutine responsible for updating the rewards snapshot within the `takeRewards` function. So now with this fix, this subroutine is called in all cases, and thus the snapshots are always updated.

Summary

We are satisfied by the solution proposed, and we are glad that The Graph team is being proactive in hardening their contracts against such bugs. We must note that the PR in question is still not merged; we assume that The Graph team will merge it as it is and that no other bugs are introduced in eventual changes.

The code has been audited by two auditors during the course of one day.

Critical Severity

None.

High Severity

None.

Medium Severity

None.

Low Severity

None.

Notes & Additional Information

None.

Conclusions

We found this bugfix to be straightforward and clean, and we detected no issues worth mentioning.