The Graph RewardsManager Upgrade Audit

APRIL 7, 2021 | IN SECURITY AUDITS | BY OPENZEPPELIN SECURITY

RewardsManager Upgrade

The Graph team found an edge case condition in which the amount of accumulated rewards retrieved from the `RewardsManager` for a specific signal can be zero.

The main condition is that, on a subgraph having allocations, the curation signal is removed entirely.

In this case, if someone calls the `_closeAllocation` function of the `Staking` contract, this will internally call the `_distributeRewards` function and finally the `takeRewards` function of the `RewardsManager`.

The problem is that the `takeRewards` function is internally calling the `onSubgraphAllocationUpdate` function which will call `getAccRewardsPerAllocatedToken` and the `getAccRewardsForSubgraph` functions. This last function should return the rewards accumulated over time for a given subgraph. Whenever the execution reach this function call, if there are no tokens in the curation pool the result will be zero and the result of the `onSubgraphAllocationUpdate` call will return `accRewardsPerAllocatedToken == 0`.

Finally the `takeRewards` function will call the `_calcRewards` function that will return zero too, and zero rewards will be minted.

The intention of the `getAccRewardsForSubgraph` function is to retrieve an ever increasing number, but under such conditions, zero would be returned, no matter the previous returned value.

The Graph team addressed this issue in PR#452.

The changes are the following:

- The `newAccrued` and `newValue` variables of the `getAccRewardsForSubgraph` function are now `newRewardsPerSignal` and `newRewards` accordingly. The `newAccrued` and `newValue` variables of the `getAccRewardsPerAllocatedToken` are now `newRewardsForSubgraph` and `newRewardsPerAllocatedToken`. Finally, the `newAccrued` variable of the `_calcRewards` function has been renamed to `newRewardsPerAllocatedTokens`.
- The check in lines 219-221 has been removed and the `getAccRewardsForSubgraph` function is not returning zero anymore on an empty pool. Instead it will return now the old value with no new rewards added.
- A check has been added in the `takeRewards` function to avoid minting zero rewards.

We are very happy with the small and modular changes that The Graph is performing and we are glad that this edge case has been spotted and solved.

This pull request has been audited during the course of two days by one auditor and reviewed by a reviewer during the course of one day.

The only note to report is the fact that the `takeRewards` function is returning zero whenever `accRewardsPerAllocatedToken` has the same value as `acc.accRewardsPerAllocatedToken` (when there are no new rewards), but before that, the `_calcRewards` functions and the `if` statement are evaluated even if they will have no effects. To be more gas efficient, consider returning earlier to avoid performing useless operations.

Security Audits

- If you are interested in smart contract security, you can continue the discussion in our forum, or even better, join the team 🚀
- If you are building a project of your own and would like to request a security audit, please do so here.