Armen Arakelian

Artem Chistyakov

tornado-relayer-registry

AUDIT

Distributed Lab

STRUCTURE

- **Informational** - The issue has no impact on the contract's ability to operate.
- **Low** - The issue has minimal impact on the contract's ability to operate.
- **Medium** - The issue affects the ability of the contract to operate in a way that doesn't significantly hinder its behavior.
- **High** - The issue affects the ability of the contract to compile or operate in a significant way.
- **Critical** - The issue affects the contract in such a way that funds may be lost, allocated incorrectly, or otherwise result in a significant loss.

SUMMARY

All smart contracts from the provided repository along with the contracts from all repositories of the project that anyhow are connected to **tornado-relayer-registry** repo or needed for understanding the logic of it were audited for security, usability and logical issues.
As a result no medium high or critical issues were identified. So we consider the system secure.
In spite of this we also noted a lot of architectural issues from our point of view. The code in the repo is tangled and appears difficult to read sometimes.
To provide some general examples I will point to a few parts.
First of all, the system contains a few registry contracts for different data that need to be used in multiple parts of the system. It is common and good practice but in most cases it requires no other dependencies between the system parts. In our case dependencies between the smart contracts are mostly chaotic. Meaning that smart contracts usually "know" about the other once without any structure. But suppose to know only about registries.

ANALYSIS

FeeManager.sol

Low: the relayer's burn fee is not updated when the user withdraws

> The fee manager is implemented the way that requires the fee to be updated externally. That means that an external service has to monitor the fee deviations and update fees accordingly. This logic might be overcomplicated a bit and the recommendation is to calculate the pool fee on the fly. `calculatePoolFee()` function is not that expensive to execute but it might prevent a lot of debugging headache.

The flow:

TornadoRouter.withdraw() -> RelayerRegistry.burn() -> FeeManager.instanceFee()

Instead of calling instanceFee() method, call calculatePoolFee()

Recommendation:

Calculate the pool fee on the fly

Issue status: Fixed.

Governance.sol

Informational: Executor storage has to be maintained

The executor of the governance proposal has to have the same (or none) storage layout as the governance contract itself, otherwise, the execution may lead to unexpected behavior.

Recommendation:

Double-check that the executor has valid storage (or descends from the governance contract).

Issue status: Fixed.

RelayerRegistry.sol

Informational: the memory variables can be calldata

Functions register(), registerPermit() and _register() have several parameters declared as memory, all these variables can be changed to calldata to save gas.

Recommendation:

Change ensName and workersToRegister variables to calldata variables.

Issue status: Fixed.

TornadoStakingRewards.sol

Informational: Alone standing interface should be replaced with one that contracts really inherit from.

> Interface ITornadoGovernance is not a real interface of any smart contract existing in the repo. Contract TornadoStakingRewards relies on it just as it is a Governance interface. Such usage of the interface can cause a mistake while bounding a smart contract that implies this interface.

Recommendation:

> Create one interface that all Governance contracts would inherit, import it as a code file and replace existing alone standing interface.

The same point for all such interfaces:

1. GovernanceStakingUpgrade.sol
   interface:
   ITornadoStakingRewards.

2. RelayerRegistry.sol
   interfaces:
   ITornadoStakingRewards,
   IENS.

Issue status: Not fixed with comment from developers:

We are aware of the non-canonical use of interfaces and plan to fix this code style issue. But not within the audited project as fixes on related projects are needed.

RelayerRegistryProposal.sol

Informational:

> File imports AdminUpgradeableProxycontract but never uses it.

Recommendation:

> Remove the import line of code.

Issue status: Fixed.