Audit Report

Mirror Decentralized Governance

v0.5

January 14, 2022

Table of Contents

License

Disclaimer

This audit has been performed by

Oak Security

https://oaksecurity.io/
info@oaksecurity.io

Introduction

Purpose of This Report

Oak Security has been engaged by Terraform Labs Pte. Ltd. to perform a security audit of the changes introduced in release v2.2.x of the Mirror Protocol (https://mirror.finance).

The objectives of the audit are as follows:

1. Determine the correct functioning of the protocol, in accordance with the project specification.
2. Determine possible vulnerabilities, which could be exploited by an attacker.
3. Determine smart contract bugs, which might lead to unexpected behavior.
4. Analyze whether best practices have been applied during development.
5. Make recommendations to improve code safety and readability.

This report represents a summary of the findings.

As with any code audit, there is a limit to which vulnerabilities can be found, and unexpected execution paths may still be possible. The author of this report does not guarantee complete coverage (see disclaimer).

Codebase Submitted for the Audit

The audit has been performed on the changes introduced in the following GitHub branch:

https://github.com/Mirror-Protocol/mirror-contracts/tree/release/v2.2.x

Commit hash: `23825e418edcaf832e46f81576fd57879ce0d290`

Methodology

The audit has been performed in the following steps:

1. Gaining an understanding of the code base's intended purpose by reading the available documentation.
2. Automated source code and dependency analysis.
3. Manual line by line analysis of the source code for security vulnerabilities and use of best practice guidelines, including but not limited to:
    a. Race condition analysis
    b. Under-/overflow issues
    c. Key management vulnerabilities
4. Report preparation

Functionality Overview

The submitted smart contracts implement release v2.2.x of the Mirror Protocol, a decentralized finance (DeFi) protocol aimed at providing synthetic representations of real-world assets on the Terra blockchain. Mirrored assets are minted by collateralized debt positions (CDP).

This audit focuses on the following subset of changes to the Mirror protocol:

- Decentralized governance including (delegated) contract migrations.
- Migration of LP tokens from one to another AMM protocol.
- Introduction of more generalized oracles.

How to Read This Report

This report classifies the issues found into the following severity categories:

Severity	Description
Critical	A serious and exploitable vulnerability that can lead to loss of funds, unrecoverable locked funds, or catastrophic denial of service.
Major	A vulnerability or bug that can affect the correct functioning of the system, lead to incorrect states or denial of service.
Minor	A violation of common best practices or incorrect usage of primitives, which may not currently have a major impact on security, but may do so in the future or introduce inefficiencies.
Informational	Comments and recommendations of design decisions or potential optimizations, that are not relevant to security. Their application may improve aspects, such as user experience or readability, but is not strictly necessary. This category may also include opinionated recommendations that the project team might not share.

The status of an issue can be one of the following: **Pending,** **Acknowledged** or **Resolved**.

Note, that 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 the 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**. We include a table with these criteria below.

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

Summary of Findings

No	Description	Severity	Status
1	Revoked collateral assets re-activated during contract migration	Major	Resolved
2	Incorrect specification of migration poll voting period	Major	Resolved
3	Collateral oracle's config migration is not performed	Major	Resolved
4	Admin rights cannot be revoked when claimed	Minor	Acknowledged
5	Users can grief creation of admin polls	Minor	Resolved
6	Admin and migration poll configuration can be updated in a default poll	Minor	Resolved
7	Admin manager can be updated in governance contract with default poll config	Minor	Resolved
8	Migrations that run out of gas remain executable and may be triggered in the far future	Minor	Resolved
9	Addresses for migration in migration polls are not validated	Minor	Resolved
10	Best practices for canonical addresses	Informational	Acknowledged
11	Mint asset configuration loaded twice	Informational	Resolved
12	Non-default polls are not covered in the governance tests	Informational	Resolved

Code Quality Criteria

Criteria	Status	Comment
Code complexity	Medium-High	-
Code readability and clarity	Medium	-
Level of documentation	Medium	-
Test coverage	Medium-High	-

Detailed Findings

1. Revoked collateral assets re-activated during contract migration

Severity: Major

When performing the migration of the collateral oracle contract, legacy collateral information is migrated to the new storage layout. During this process, every collateral is marked with `is_revoked: false` in `contracts/mirror_collateral_oracle/src/migration.rs:112`. The consequence of this is that collateral that had previously been revoked is now re-activated.

Recommendation

We recommend replacing `false` in line `contracts/mirror_collateral_oracle/src/migration.rs:112` with `legacy_collateral_info.is_revoked` to ensure that the collateral status is unchanged.

Status: Resolved

Resolved in `dadf0c3`

2. Incorrect specification of migration poll voting period

Severity: Major

During the creation of a new migration poll, the voting period is defined incorrectly as that of the default poll config in line `contracts/mirror_gov/src/contract.rs:329`. The impact of this would be the reduction in the voting period of migration polls, allowing less time for them to pass with a potential danger of a distributed poll deposit by the protocol.

Recommendation

We recommend altering `contracts/mirror_gov/src/contract.rs:329` to use the voting period defined in the `migration_poll_config` in place of the `default_config`.

Status: Resolved

Resolved in `8787915`

3. Collateral oracle's config migration is not performed

Severity: Major

The function `migrate_config` in `contracts/mirror_collateral_oracle/src/migration.rs:49` is unused. This implies that after the migration, the config will not be decoded properly.

Recommendation

We recommend calling `migrate_config` from the migrate function in `contracts/mirror_collateral_oracle/src/contract.rs:293`.

Status: Resolved

Resolved in `871b7df`

4. Admin rights cannot be revoked when claimed

Severity: Minor

The current implementation of the admin manager contract in `contracts/admin_manager/src/contract.rs` allows an account to claim the contract's admin rights. Upon claiming the admin rights the account becomes the target contract's admin and has the right to perform contract migrations. There is no mechanism to revoke any claimed admin rights. If the third party's account is lost or compromised, there is no way for the protocol to recover, and all locked funds might be lost forever.

Recommendation

We recommend that instead of updating the admin manager contract's admin to an external account, the external account be permissioned to dispatch any update through the admin contract. That would allow expiry and revocation of admin rights. Unfortunately, this mechanism does not work in the case of Terra protocol updates that require an update through an externally owned account, such as the Columbus 5 update.

To mitigate this issue, we recommend using a multi-sig with sufficient signatories and a high threshold.

Lastly, off-chain mechanisms could be used to incentivize proper behavior.

Status: Acknowledged

The Mirror team acknowledges this issue. Due to limitations of the underlying blockchain, it is currently not possible to resolve it fully. At the same time, the feature contributes to decentralization of Mirror, hence it is an improvement over the current setup where an externally owned multisig holds the admin permissions. The separate poll thresholds and quorums introduced with this feature have been designed to mitigate the risks. The Mirror

team intends to use off-chain mechanisms and proper communication with the community to ensure the functionality is used in a benevolent way.

5. Users can grief creation of admin polls

Severity: Minor

Creation of a new admin poll is not possible if the number of existing polls is greater or equal to `MAX_POLLS_IN_PROGRESS` in `contracts/mirror_gov/src/contract.rs:335`. The impact of which is prevention of new admin polls being created through continuous adding of new polls.

We consider this issue to be minor since it is mitigated by the fact that deposits for failed polls are distributed to stakers. Despite this cost, there is still a possibility of attackers trying to grief admin actions.

Recommendation

We recommend increasing the number of maximum polls in the creation of admin polls in `contracts/mirror_gov/src/contract.rs:335`, as is done in the creation of migration polls in `contracts/mirror_gov/src/contract.rs:330`.

Status: Resolved

Resolved in 8787915

6. Admin and migration poll configuration can be updated in a default poll

Severity: Minor

If the governance config is updated through a poll, the `default_poll_config` will be used in `contracts/mirror_gov/src/contract.rs:435`. This could enable a whale or colluding token holders to alter the configuration of the admin and migration poll types through a default poll. Subsequently, such malicious actors could execute migration and admin actions with significantly lower poll requirements to extract value from Mirror.

We consider this issue minor since the community could observe such polls and intervene by rejecting them.

Recommendation

We recommend that polls that update the admin or migration poll config respectively apply the `auth_admin_poll_config` or `migration_poll_config`.

Status: Resolved

Resolved in `0f38091`

7. Admin manager can be updated in governance contract with default poll config

Severity: Minor

While all admin actions imply admin or migration poll configs through `contracts/mirror_gov/src/contract.rs:373`, the current implementation allows an update of the `admin_manager` defined in the contract with the default poll config.

After updating the `admin_manager` to any other address, normal messages can be used for admin actions, including contract migrations.

As in the last issue, we only consider this issue minor since the community could observe such polls and intervene by rejecting them.

Recommendation

We recommend that polls updating the `admin_manager` in the governance config apply the `auth_admin_poll_config`.

Status: Resolved

Resolved in `0f38091`

8. Migrations that run out of gas remain executable and may be triggered in the far future

Severity: Minor

The governance contract supports migrations to be run when a poll is executed through the `ExecutePoll` message. Such execution, which is triggered as a sub-message in `contracts/mirror_gov/src/contract.rs:615`, does not have a gas limit set. If the migration (or any other message) runs out of gas, the whole `ExecutePoll` message will revert. As a consequence, the status of the poll will not be changed from `PollStatus::Passed` to `PollStatus::Failed`. This implies that polls will stay open for an unlimited time.

This can become problematic if Terra decides to increase the gas limit for messages in the future. In such a case outdated migration messages might be triggered, which could lead to an inconsistent state of the contracts.

This issue also replies to any other poll running out gas, but is especially devastating with migrations that might change the storage layout.

Recommendation

We recommend setting a reasonable gas limit to the sub-message in `contracts/mirror_gov/src/contract.rs:617`, ideally storing it in the config and allowing governance to update it. Otherwise we recommend adding an expiry date to all polls, after which they cannot be executed, which would invalidate any dangling polls.

Status: Resolved

Resolved in `79201f9`

9. Addresses for migration in migration polls are not validated

Severity: Minor

When creating migration polls, addresses for the contracts to be migrated are currently not validated prior to message creation in `contracts/mirror_gov/src/contract.rs:409`. This could lead to failure of migration executions.

Recommendation

We recommend using `deps.api.addr_validate` to perform a validation of `migration.0` prior to creating the message on `contracts/mirror_gov/src/contract.rs:409`.

Status: Resolved

Resolved in `8787915`

10. Best practices for canonical addresses

Severity: Informational

The use of canonical addresses is no longer recommended as best practice for CosmWasm contracts.

Recommendation

We recommend using `Addr` in place of `CanonicalAddr` throughout the codebase.

Status: Acknowledged

11. Mint asset configuration loaded twice

Severity: Informational

During migration of an asset the `load_mint_asset_config` is called twice `contracts/mirror_factory/src/contract.rs:638` and `contracts/mirror_factory/src/contract.rs:652`. Querying the same data twice is inefficient.

Recommendation

We recommend removing the second query in `contracts/mirror_factory/src/contract.rs:652` and storing all returned variables on the first query.

Status: Resolved

Resolved in `ce68f98`

12.Non-default polls are not covered in the governance tests

Severity: Informational

Testing creation of non-default polls is not covered by the governance contract tests in `contracts/mirror_gov/src/testing/tests.rs`. This could lead to non-default polls not executing as expected and reduces code coverage.

Recommendation

We recommend testing the creation and execution of each non-default poll type in the relevant tests `contracts/mirror_gov/src/testing/tests.rs`.

Status: Resolved

Resolved in `ce68f98`