Releap Social Smart Contract

Audit Report

MOVEBIT

✉ contact@movebit.xyz

https://twitter.com/movebit_

07/18/2023

Releap Social Smart Contract Audit Report

1 Executive Summary

1.1 Project Information

Description	Releap Social is a Fully Decentralized Social Graph
Type	Social
Auditors	MoveBit
Timeline	July 10, 2023 – July 18, 2023
Languages	Move
Platform	Sui
Methods	Architecture Review, Unit Testing, Manual Review
Source Code	https://github.com/releapxyz/releap-protocol https://github.com/releapxyz/releap-token
Commits	9ee60404fea236ca69eadfd2599016b2b791526a 94495e4d88430c754543467b52e41ecc8671f38f a09bc15346c47156afa13f10f66542c1cd1dc4b2

1.2 Files in Scope

The following are the SHA1 hashes of the last reviewed files.

ID	Files	SHA-1 Hash
ERR	releap-protocol/sources/err.move	dde6c5a4a4b6b91ddc2c856cd075be9d1a5b849c

POS	releap-protocol/sources/post.move	bfdecc6e3e7a057822c7d637cdf1fe7704fa5d8d
PRF	releap-protocol/sources/profile.move	ea2db2e9ab27d94b586355882bafa3d23b3bc8ac
RPS	releap-protocol/sources/releap_social.move	8dbd3accd95c3f833d85329c733f739806cfbef0
TOK	releap-token/sources/reap_token.move	a1c9d4d84a48d2638ece14ddfdb8f574e9810a25
RTM	releap-token/Move.toml	a7355f18f5e371006af25f0a819972a9d8e122bc
RPM	releap-protocol/Move.toml	a7899c80949616b3d3d0e10687f2a4e6799c8a67

1.3 Issue Statistic

Item	**Count**	**Fixed**	**Partially Fixed**	**Acknowledged**
Total	11	8		3
Informational	1	1		
Minor	7	5		2
Medium				
Major	3	2		1
Critical				

1.4 MoveBit Audit BreakDown

MoveBit aims to assess repositories for security-related issues, code quality, and compliance with specifications and best practices. Possible issues our team looked for included (but are not limited to):

- Transaction-ordering dependence
- Timestamp dependence
- Integer overflow/underflow by bit operations
- Number of rounding errors
- Denial of service / logical oversights
- Access control
- Centralization of power
- Business logic contradicting the specification
- Code clones, functionality duplication
- Gas usage
- Arbitrary token minting
- Unchecked CALL Return Values
- The flow of capability
- Witness Type

1.5 Methodology

The security team adopted the "**Testing and Automated Analysis**", "**Code Review**" and "**Formal Verification**" strategy to perform a complete security test on the code in a way that is closest to the real attack. The main entrance and scope of security testing are stated in the conventions in the "Audit Objective", which can expand to contexts beyond the scope according to the actual testing needs. The main types of this security audit include:

(1) Testing and Automated Analysis

Items to check: state consistency / failure rollback / unit testing / value overflows / parameter verification / unhandled errors / boundary checking / coding specifications.

(2) Code Review

The code scope is illustrated in section **1.2**.

(3) Formal Verification

Perform formal verification for key functions with the Move Prover.

(4) Audit Process

- Carry out relevant security tests on the testnet or the mainnet;
- If there are any questions during the audit process, communicate with the code owner in time. The code owners should actively cooperate (this might include providing the latest stable source code, relevant deployment scripts or methods, transaction signature scripts, exchange docking schemes, etc.);
- The necessary information during the audit process will be well documented for both the audit team and the code owner in a timely manner.

2 Summary

This report has been commissioned by **Releap Protocol** to identify any potential issues and vulnerabilities in the source code of the **Releap Social** smart contract, as well as any contract dependencies that were not part of an officially recognized library. In this audit, we have utilized various techniques, including manual code review and static analysis, to identify potential vulnerabilities and security issues.

During the audit, we identified **11** issues of varying severity, listed below.

ID	Title	Severity	Status
ERR–01	Unused Function `not_enough_balance`	Minor	Fixed
POS–01	Missing Event Logging in Multiple Functions	Minor	Fixed
POS–02	The `object::id()` Function is Called Repeatedly	Minor	Fixed
POS–03	Unhandled Branches in Functions `like_post` , `unlike_post` , `profile_follow_` , and `profile_unfollow_`	Minor	Fixed
POS–04	Event Logging for `create_comment` Function Errors	Minor	Fixed
RPS–01	Wrong Amount Charged by `new_profile` Function	Major	Fixed

RPS–02	Creating A `profile` Lacks Assertion Judgment	Minor	Acknowledged
RPS–03	Comments and Code Do Not Match	Informational	Fixed
RPS–04	Admin Privileges are Too Large	Major	Fixed
RPS–05	Unnecessary Entry Function Calls	Minor	Acknowledged
TOK–01	Centralization Risk	Major	Acknowledged

3 Participant Process

Here are the relevant actors with their respective abilities within the `Releap Social` Smart Contract：

Admin

- Admin can create a profile.
- Admin can update the price of creating a profile.
- Admin can update the beneficiary address.
- Admin can update the number of created profiles allowed.
- Admin can mint any number of reap tokens and transfer them to any address.

User

- User can create/update profile.
- User can create a post/comment.
- User can follow/unfollow another user.
- User can like/dislike a post.
- Users can add/remove an address to wallet delegation.

4 Findings

ERR–01 Unused Function `not_enough_balance`

Severity: Minor

Status: Fixed

Code Location: releap-protocol/sources/err.move#L20

Descriptions: The `not_enough_balance` function is not used.

Suggestion: The unused function `not_enough_balance` should be reviewed. If it is no longer needed, it should be removed from the codebase.

Resolution: The client has followed our suggestion and fixed the issue.

POS-01 Missing Event Logging in Multiple Functions

Severity: Minor

Status: Fixed

Code Location: releap-protocol/sources/post.move#L170

Descriptions: The functions such as `unlike_post` , `profile_unfollow` , `admin_update_profile_price` , `admin_update_beneficiary` , `admin_update_profile_cap` , and others do not log events.

Suggestion: The identified functions, including `unlike_post` , `profile_unfollow` , `admin_update_profile_price` , `admin_update_beneficiary` , `admin_update_profile_cap` , and any other relevant functions, should be modified to include event logging.

Resolution: The client has followed our suggestion and fixed the issue.

POS-02 The `object::id()` Function is Called Repeatedly

Severity: Minor

Status: Fixed

Code Location: releap-protocol/sources/post.move#L113

Descriptions: `object::id()` is called repeatedly in the `create_post` and `create_comment` functions.

Suggestion: The code should be reviewed to identify the redundant calls to the `object::id()` function and optimize them. Refactor the code by storing the result of the `object::id()` function in a variable and reusing it where needed, rather than repeatedly calling the function.

Resolution: The client has followed our suggestion and fixed the issue.

POS-03 Unhandled Branches in Functions `like_post`, `unlike_post`, `profile_follow_`, and `profile_unfollow_`

Severity: Minor

Status: Acknowledged

Code Location: releap-protocol/sources/post.move#L155; releap-protocol/sources/post.move#L170; sources/profile.move#L261; releap-protocol/sources/profile.move#L283

Descriptions: The functions `like_post`, `unlike_post`, `profile_follow_`, and `profile_unfollow_` do not handle all possible branching processes. It is recommended to abort directly for any unhandled branches.

Suggestion: By handling all branches and providing a clear abort mechanism for unhandled cases, the code can ensure consistent behavior, and maintain code integrity.

POS-04 Event Logging for `create_comment` Function Errors

Severity: Minor

Status: Fixed

Code Location: releap-protocol/sources/post.move#L105

Descriptions: The `create_comment` function event record error, `CreateCommentEvent.content` should use `comment.content` instead of `post.content`.

Suggestion: Replace `post.content` with `comment.content`.

Resolution: The client has followed our suggestion and fixed the issue.

RPS-01 Wrong Amount Charged by `new_profile` Function

Severity: Major

Status: Fixed

Code Location: releap-protocol/sources/releap_social.move#L96-98

Descriptions: The `new_profile` function charges all the `Sui` amount input by the user, the amount that should be charged is `index.profile_price` , and transfers the remaining amount after deducting `index.profile_price` to the user.

Suggestion: Modify the amount charged by the `new_profile` function to `index.profile_price` .

Resolution: The client has followed our suggestion and fixed the issue.

RPS-02 Creating A `profile` Lacks Assertion Judgment

Severity: Minor

Status: Acknowledged

Code Location: releap-protocol/sources/releap_social.move#L102; releap-protocol/sources/releap_social.move#L183; releap-protocol/sources/releap_social.move#L192

Descriptions: `new_profile` , `new_profile_with_admin_cap` , and `new_profile_with_admin_cap_bypass_name_validation` functions did not determine whether the `name` exists in the `index.profiles` when creating a `Profile` .

Suggestion: Add an assertion to determine whether the input name exists, and abort if it exists.

RPS-03 Comments and Code Do Not Match

Severity: Informational

Status: Fixed

Code Location: releap-protocol/sources/releap_social.move#L47

Descriptions: The value of `profile_price` in the `init` function is 1 SUI, and the comment is 0.01 SUI, the comment is wrong.

Suggestion: Modify the comment to the correct value.

Resolution: The client has followed our suggestion and fixed the issue.

RPS-04 Admin Privileges are Too Large

Severity: Major

Status: Fixed

Code Location: releap-protocol/sources/releap_social.move#L200; releap-protocol/sources/releap_social.move#L206; releap-protocol/sources/releap_social.move#L210; releap-protocol/sources/releap_social.move#L214; releap-protocol/sources/releap_social.move#L218; releap-protocol/sources/releap_social.move#L222

Descriptions: Admin can use anyone's `Profile` to perform some operations. Because `Profile` is a shared Object, which results in

1. The admin can use the profile of any user to create posts through the `profile::create_post_` function.

2. The admin can use the profile of any user to create comments through the `profile::create_comment_` function.

3. The admin can use the profile of any user to follow or unfollow the user through the `profile::profile_follow_` and `profile::profile_unfollow_` functions.

4. The admin can use the profile of any user to like or unlike posts through the `post::like_post` and `post::unlike_post` functions.

Suggestion: Restrict admin privileges, the admin cannot use profiles created by users to perform these operations.

Resolution: The client has followed our suggestion and fixed the issue.

RPS-05 Unnecessary Entry Function Calls

Severity: Minor

Status: Acknowledged

Code Location: sources/releap_social.move#148

Descriptions: Some entry functions in the `releap_social` module directly call the entry function in the `profile` module without adding additional logic, for example, `releap_social::update_profile_cover_image` calls `profile::update_profile_image` , `release_social::update_profile_image` calls `profile::update_profile_cover_image` , `release_social::update_profile_description` call `profile::update_profile_description` , so we can use `profile::update_profile_image` , `profile::update_profile_image` , and `profile::update_profile_cover_image` as the entry function.

Suggestion: Delete such as `releap_social::update_profile_cover_image`, `releap_social::update_profile_image`, and `releap_social::update_profile_description` functions. These functions directly call the entry function of the `profile` module without adding additional logic. Use `profile::update_profile_image`, `profile::update_profile_cover_image`, and `profile::update_profile_description` functions instead.

Resolution: Our client replied that they need to keep the entry function to maintain backward compatibility.

TOK-01 Centralization Risk

Severity: Major

Status: **Acknowledged**
Code Location: releap-token/sources/reap_token.move
Descriptions: There is a risk of centralization, with privileged accounts able to mint **unlimited tokens** and burn their token.

Suggestion: It is recommended to take measures to mitigate this issue.

Appendix 1

Issue Level

- **Informational:** Informational items are often recommendations to improve the style of the code or to optimize code that does not affect the overall functionality.
- **Minor** issues are general suggestions relevant to best practices and readability. They don't post any direct risk. Developers are encouraged to fix them.
- **Medium** issues are non-exploitable problems and not security vulnerabilities. They should be fixed unless there is a specific reason not to.
- **Major** issues are security vulnerabilities. They put a portion of users' sensitive information at risk, and often are not directly exploitable. All major issues should be fixed.
- **Critical** issues are directly exploitable security vulnerabilities. They put users' sensitive information at risk. All critical issues should be fixed.

Issue Status

- **Fixed:** The issue has been resolved.
- **Acknowledged:** The issue has been acknowledged by the code owner, and the code owner confirms it's as designed, and decides to keep it.

Appendix 2

Disclaimer

This report is based on the scope of materials and documents provided, with a limited review at the time provided. Results may not be complete and do not include all vulnerabilities. The review and this report are provided on an as–is, where–is, and as–available basis. You agree that your access and/or use, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your own risk. A report does not imply an endorsement of any particular project or team, nor does it guarantee its security. These reports should not be relied upon in any way by any third party, including for the purpose of making any decision to buy or sell products, services, or any other assets. TO THE FULLEST EXTENT PERMITTED BY LAW, WE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, IN CONNECTION WITH THIS REPORT, ITS CONTENT, RELATED SERVICES AND PRODUCTS, AND YOUR USE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NOT INFRINGEMENT.