Pinata

Audit Report

contact@bitslab.xyz

https://twitter.com/movebit_

Fri Aug 30 2024

Pinata Audit Report

1 Executive Summary

1.1 Project Information

Description	A non-custodial Telegram bot
Type	DeFi
Auditors	MoveBit
Timeline	Mon Aug 26 2024 - Fri Aug 30 2024
Languages	Move
Platform	Sui
Methods	Architecture Review, Unit Testing, Manual Review
Source Code	https://github.com/PinataBot/pinata-contract
Commits	d5ae9e2f5f69a49ac13450046bcd5045be27dfd8 5803e08ba86886df151633be3d544c0b39270709

1.2 Files in Scope

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

ID	File	SHA-1 Hash
UTI	pre_market/sources/utils.move	6f76a078375ca017b761c4925440624ebfbda151
GAM	break_sui_pinata/sources/game.move	d70a1bf2995e1ee1481ce7b2fca0097c2e8d2c0e
OFF	pre_market/sources/offer.move	4299c00a240ceca50b2579ae0ea2ff5cebb5c928
MAR	pre_market/sources/market.move	ad56c9ccd1525f663c488b98b7518e1e97e8391b

1.3 Issue Statistic

Item	Count	Fixed	Acknowledged
Total	3	2	1
Informational	0	0	0
Minor	1	1	0
Medium	2	1	1
Major	0	0	0
Critical	0	0	0

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 Pinata to identify any potential issues and vulnerabilities in the source code of the Pinata 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 3 issues of varying severity, listed below.

ID	Title	Severity	Status
MAR-1	Centralization Risk	Medium	Acknowledged
MAR-2	The Data in The `market` Is Manipulable	Medium	Fixed
MAR-3	The `coin_decimals` Is Not Set to `None`	Minor	Fixed

3 Participant Process

Here are the relevant actors with their respective abilities within the Pinata Smart Contract :

Admin

- Admin can create a new game using the `new` function.
- Admin can cancel a game using the `cancel` function.
- Admin can create a new market using the `new` function.
- Admin can initiate the settlement of a market using the `settlement` function.
- Admin can cancel the settlement of a market using the `unsettlement` function.
- Admin can close the market using the `close` function.
- Admin can withdraw funds from the market using the `withdraw` function.

User

- Users can perform tap action using the `tap` function.
- Users can create a new offer (buy or sell) using the create function.
- Users can cancel their unsettled offers using the `cancel` function.
- Users can be filled by other parties using the `fill` function.
- Users can settle the offer using the `settle_and_close` function.
- Users can close the offer using the `close` function.

4 Findings

MAR-1 Centralization Risk

Severity: Medium

Status: Acknowledged

Code Location:

pre_market/sources/market.move#122-132

Descriptions:

Admin can initiate the settlement of a market using the settlement function.

Admin can cancel the settlement of a market using the unsettlement function.

Admin can close the market using the close function.

Suggestion:

It is recommended to implement measures to mitigate this risk.

MAR-2 The Data in The `market` Is Manipulable

Severity: Medium

Status: Fixed

Code Location:

pre_market/sources/market.move#327-338

Descriptions:

The data in the `market` is manipulable. In the `add_offer` function, the `update_stats` updates values such as `total_buy_amount` and `total_sell_amount`, but these values can be manipulated. A user can set `collateral_value` to 1 USDC in the `create` function, then set `amount` to 10**100 (an extremely large number), causing `total_buy_amount` to be updated to a very high value. This can lead to `Average bids` and `Average asks` potentially dropping to zero.

Suggestion:

It is recommended to reset `total_buy_amount` and `total_sell_amount` in the `cancel offer` and `close` functions to prevent such manipulation.

Resolution:

This issue has been fixed. The client has adopted our suggestions.

MAR-3 The `coin_decimals` Is Not Set to `None`

Severity: Minor

Status: Fixed

Code Location:

pre_market/sources/market.move#196-204

Descriptions:

The `settlement` function configures the `market` with `settlement_end_timestamp_ms`, `coin_type`, `coin_decimals`, and `coin_symbol` fields. However, the `coin_symbol` is not set to `None` in the `unsettlement` function.

```
/// Optional function to unsettle the market
/// Call this function if there are settlement issues
entry public fun unsettlement(market: &mut Market, cap: &Publisher) {
    assert_admin(cap);

    market.settlement_end_timestamp_ms = option::none();
    market.coin_type = option::none();
    market.coin_decimals = option::none();

    emit(MarketUnsettlement { market: object::id(market) });
}
```

Suggestion:

It is recommended to set `coin_symbol` to `None` within the `unsettlement` function.

Resolution:

This issue has been fixed. The client has adopted our suggestions.

Appendix 1

Issue Level

- **Informational** issues 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.
- **Partially Fixed:** The issue has been partially 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.