Haedal Protocol

Audit Report

contact@bitslab.xyz

https://twitter.com/movebit_

Wed Dec 04 2024

Haedal Protocol Audit Report

1 Executive Summary

1.1 Project Information

Description	A decentralized trading platform powered by the Proactive Market Maker (PMM) algorithm
Type	DeFi
Auditors	MoveBit
Timeline	Mon Nov 25 2024 - Wed Dec 04 2024
Languages	Move
Platform	Sui
Methods	Architecture Review, Unit Testing, Manual Review
Source Code	https://github.com/haedallsd/haedal-pmm-contract
Commits	4ffa43b2fec636cfa0d9cb44f5da4ba1e0de317d 2c75dc8f77c2696cb6c8db3106ec1480ac3c1cd1 00558801ea21e9dc4a8e3bcb877fd6435d0a3fe6 0d1a054a4a2f5e78d21831988c50aeda85764310

1.2 Files in Scope

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

ID	File	SHA-1 Hash
ADM	sources/admin.move	dd6299f06cb376244b72b93985c7d16f562fb287
OWN	sources/ownable.move	e3162f36c166d733769b49cca88363c1c96a969d
VER	sources/version.move	1bb46fe163350684f73e4dea627723a468346634
LPR	sources/liquidity_provider.move	3372e422a3d8352b465d47ecda25674eb4111d54
SET	sources/settlement.move	7c52975844c46c493f8b0f28c5c9c3674817912a
MOV	Move.toml	8d31753b5470c394528fe39cda4b9aa89aab3247
SCR	sources/script.move	191bbbfe8b66fc31f01e1493afe6f7e4abd5ddfd
SMA	sources/safe_math.move	3ade065a4249bfa99b1cbaab990aedc07d84ed44
ODP	sources/oracle_driven_pool.move	ae5096657479efa407a5c2555ab5b84aa683e263
ORA	sources/oracle.move	0d13f7485e7364698154c3a1acbe00bb575ad7d8
TRA	sources/trader.move	5a6c84019d93aaaebde7e3708ce710cb7150624a

1.3 Issue Statistic

Item	Count	Fixed	Acknowledged
Total	6	6	0
Informational	2	2	0
Minor	2	2	0
Medium	1	1	0
Major	1	1	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(Optional)

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 Haedal Protocol to identify any potential issues and vulnerabilities in the source code of the Haedal PMM 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 6 issues of varying severity, listed below.

ID	Title	Severity	Status
ADM-1	Missing `notClosed` Check	Minor	Fixed
ADM-2	Missing the Functionality to Configure the Maintainer	Informational	Fixed
LPR-1	Incorrect Implementation of Withdraw Base	Major	Fixed
ODP-1	The Initialization Settings Differ from DODO	Informational	Fixed
ORA-1	Different Tokens Should Have Different `max_age` Values	Medium	Fixed
ORA-2	The Price Should Include A Non-zero Value Check	Minor	Fixed

3 Participant Process

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

Admin

- `grant_pool_admin_cap` : Grants admin rights for a specific pool to a user.
- `grant_liquidity_operator_cap` : Grants liquidity operator rights to a user.
- `set_liquidity_provider_fee_rate` : Sets the liquidity provider fee rate for a pool.
- `set_k` : Sets the constant product parameter (k) for a pool.
- `set_protocol_fee_rate` : Sets the protocol fee rate for a pool.
- `set_quote_usd_price_age` : Sets the valid duration for the USD price of the quote asset in a pool.
- `set_base_usd_price_age` : Sets the valid duration for the USD price of the base asset in a pool.
- `disable_trading` : Disables trading in a pool.
- `enable_trading` : Enables trading in a pool.
- `disable_deposit_quote` : Disables deposits of the quote asset in a pool.
- `enable_deposit_quote` : Enables deposits of the quote asset in a pool.
- `disable_deposit_base` : Disables deposits of the base asset in a pool.
- `enable_deposit_base` : Enables deposits of the base asset in a pool.
- `disable_buying` : Disables buying in a pool.
- `enable_buying` : Enables buying in a pool.
- `disable_selling` : Disables selling in a pool.
- `enable_selling` : Enables selling in a pool.
- `set_base_balance_limit` : Sets a balance limit for the base asset in a pool.
- `set_quote_balance_limit` : Sets a balance limit for the quote asset in a pool.

- `add_pool` : Creates a new liquidity pool with specified parameters and registers it in the system.

User

- `deposit_base` : Deposits a specified amount of the base asset into a pool.
- `deposit_quote` : Deposits a specified amount of the quote asset into a pool.
- `withdraw_base` : Withdraws a specified amount of the base asset from a pool.
- `withdraw_quote` : Withdraws a specified amount of the quote asset from a pool.
- `withdraw_all_base` : Withdraws all available base assets from a pool.
- `withdraw_all_quote` : Withdraws all available quote assets from a pool.
- `sell_base_coin` : Sells a specified amount of the base asset.
- `buy_base_coin` : Buys a specified amount of the base asset.
- `final_settlement` : Finalizes the settlement and shuts down the pool.
- `claim_assets` : Claims the remaining base and quote assets after the final settlement.
- `claim_base` : Claims the remaining base assets after the final settlement.
- `claim_quote` : Claims the remaining quote assets after the final settlement.

4 Findings

ADM-1 Missing `notClosed` Check

Severity: Minor

Status: Fixed

Code Location:

sources/admin.move#94

Descriptions:

When configuring parameters such as `enableTrading` and `enableQuoteDeposit`, a `notClosed` check should be added. These parameters should not be enabled if the pool is in a closed state.

Reference

Suggestion:

Add a `notClosed` check to ensure these parameters cannot be enabled when the pool is closed.

Resolution:

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

ADM-2 Missing the Functionality to Configure the Maintainer

Severity: Informational

Status: Fixed

Code Location:

sources/admin.move#1

Descriptions:

DODO allows configuring the maintainer's address, but the current contract lacks this functionality.

Reference

Suggestion:

Add the functionality to configure the maintainer's address.

Resolution:

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

LPR-1 Incorrect Implementation of Withdraw Base

Severity: Major

Status: Fixed

Code Location:

sources/liquidity_provider.move#344

Descriptions:

In the `withdraw_base_internal` function, the parameter passed to the `base_coin_pay_out` function is incorrect:

```
// handle penalty, penalty may exceed amount
let penalty = pool.get_withdraw_base_penalty(primitive_price, amount);
assert!(penalty <= amount, EPenaltyExceed);

// settlement
let mut base_capital_balance = coin::into_balance(base_capital_coin);
let base_capital_burn = balance::split(&mut base_capital_balance, require_base_capital);

let (target_base_coin_amount, _) = pool.get_target_amount();
// update target
pool.set_target_base_coin_amount(target_base_coin_amount - amount);

let withdraw_amount = amount - penalty;
base_capital_burn(pool, base_capital_burn, penalty);

pool.base_coin_pay_out(amount, ctx.sender(), ctx);  // Incorrect parameter
```

The `base_coin_pay_out` function should receive `amount - penalty` instead of `amount`. This causes incorrect updates to the `base_balance` and transfers the wrong amount of tokens to the user.

Reference: DODO Implementation

The same issue exists in the `withdraw_quote_internal`, `withdraw_all_quote_internal`, and `withdraw_all_base_internal` functions.

Suggestion:

Update the code to pass `amount - penalty` to the `base_coin_pay_out` function.

Resolution:

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

ODP-1 The Initialization Settings Differ from DODO

Severity: Informational

Status: Fixed

Code Location:

sources/oracle_driven_pool.move#939-941

Descriptions:

In DODO, the following parameters are set to `false` during initialization, while the current contract sets them to `true`:

- `deposit_base_allowed: true`
- `deposit_quote_allowed: true`
- `trade_allowed: true`

Reference: DODO Initialization Code

Suggestion:

It is recommended to initialize these parameters as `false`.

Resolution:

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

ORA-1 Different Tokens Should Have Different `max_age` Values

Severity: Medium

Status: Fixed

Code Location:

sources/oracle.move#21-23

Descriptions:

Since the price volatility of each token varies, it is recommended to adjust the `max_age` parameter accordingly. For tokens with high volatility, use a smaller `max_age`, while for tokens with lower volatility, a larger `max_age` can be applied.

```
let max_age = 60;
// Ensure the price is not older than max_age seconds
let pyth_price = pyth::get_price_no_older_than(pyth_price_pair_obj, clock, max_age);
```

Suggestion:

It is advisable to customize the `max_age` value for different tokens based on their volatility characteristics.

Resolution:

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

ORA-2 The Price Should Include A Non-zero Value Check

Severity: Minor

Status: Fixed

Code Location:

sources/oracle.move#37-41

Descriptions:

In Pyth, `price` uses negative numbers and non-negative numbers (positive numbers) to represent values, with a value of `0` interpreted as positive.

```
// Ensure we have a single zero representation: (0, false).
// (0, true) is invalid.
if (magnitude == 0) {
    negative = false;
};
```

However, a price should not have a value of `0`. Therefore, a non-zero value check should be added.

Suggestion:

It is recommended to add a non-zero value check for price.

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.