G₀ Group

Security Review of
Melon Protocol

February 21, 2020

Melon protocol / February 2020

Files in scope

All solidity files present in the repository at this commit

https://github.com/melonproject/protocol/commit/eeb34fb0a321640d162be86d033a0d52985d8c12

Issues

1. It's possible to bypass paying fund performance fee while divesting

Type: security / Severity: major

Including `addess(0)` in `requestedAssets` of `Participation.redeemWithConstraints` allows attacker to avoid paying performance fees because of `line 314` and `line 331`.

status - fixed

Issue has been fixed and is no longer present in:
https://github.com/melonproject/protocol/commit/b2541173b6d7b51c514d458567af5262dd37aeca

2. It's possible to successfully pass an identifier of an already expired order to cancelOrder function

Type: security / Severity: major

In `ZeroExV3Adapter`, it's possible to pass `cancelOrder` an `_identifier` of already cancelled or expired order instead of the currently expired one, leading to an incorrect amount of tokens being unapproved for the exchange and also current order not being cancelled leading to multiple orders for the same asset and exchange existing at the same time.

status - fixed

Issue has been fixed and is no longer present in:
https://github.com/melonproject/protocol/commit/b2541173b6d7b51c514d458567af5262dd37aeca

3. Partially filled orders can't be properly cancelled

Type: security / Severity: medium

`revokeApproveAssetsCancelOrder` doesn't really consider that cancelled orders can be already partially filled. Which might result in attempts to reduce exchange's allowance below 0.

status - fixed

Issue has been fixed and is no longer present in:
https://github.com/melonproject/protocol/commit/b2541173b6d7b51c514d458567af5262dd37aeca

4. Certain token standards can open system to reentrancy attacks

Type: security

Beware of extensions of ERC20 standard such as ERC223, ERC777 or ERC827 which call the destination address during transfer. If any such token is added as an asset, it activates a host of reentrancy issues.

status - addressed

Clent's response: We think this can be taken care of by having a convention within the council to only permit tokens whose standards are not changeable and have some integration/handling at the protocol level. All tokens that are already permitted would have to be examined to determine which are in accordance with those rules.

5. Fund mamanger can hold investor funds hostage

Type: security / Severity: major

Fund manager can hold investor funds hostage by never calling `returnAssetToVault` after orders have been filled. During the process of share redeeming, users can only withdraw funds from the `Vault` so they won't be able to access funds that are held by the`Trading` component. On the other hand, management and performance fee are still being counted so the situation is beneficial to the fund manager.

status - fixed

Issue has been fixed and is no longer present in:
https://github.com/melonproject/protocol/commit/b2541173b6d7b51c514d458567af5262dd37aeca

6. There are multiple ways in which funds necessary for a pending trade can be prematurely returned to vault

Type: usability / Severity: medium

When a make order to an exchange that doesn't take custody is pending, it's possible that either `takeOrder` or `cancelOrder` (as `makerFeeAsset`) will return maker asset or maker fee asset of the pending order to the vault.

status - fixed

Issue has been fixed and is no longer present in:
https://github.com/melonproject/protocol/commit/b2541173b6d7b51c514d458567af5262dd37aeca

Notes

We don't track status of following notes because they don't affect quality of the product from user's perspective. Issues mentioned concern either code quality or minor inefficiencies.

FundFactory.sol

Why isn't `setSpokes`, `setRouting` and `setPermissions` one function? Why is engine and mlntoken passed to the hub separately if it's part of registry? Why save all routes in all spokes?

Spoke.sol

Why store all routes in all spokes?

Participation.sol

`L 211` might be unnecesary, because `L 218` throws if one of the prices is invalid.`L 275` will probably never be reached because of `L 315` `L 129`, `L 151` why not use `msg.sig`? Does `L 151` need to pass `msg.sender`? `L 298` why not use Accounting.ownedAssets instead since the balances are thrown away?

ManagementFee.sol

`L 17` Shares referrence can be taken directly from`Spoke` (`FeeManager`)

ZeroExV3Adapter.sol

`L 319` even if makerFeeAsset is the same as`makerAsset`, the return to vault already happened so either should be called again or should be called only once at the end. `L 79` unused var.

MaxConcentration.sol

`L 43` can't taker token be denomination asset also?

PriceTolerance.sol

`takeGenericOrder` & `takeOasisDex` + `makeOrder` implement almost identical functionality with a consideraby different code, probably should be unified and ideally abstracted to a function.

KyberPriceFeed.sol

In `getOrderPriceInfo` the buyAsset argument is unused.

Accounting.sol

`calcSharePrice` is not called by any part of the system and`performCalculations` provides the same info.

MaxPositions.sol

`L 3` is unused.