G0 Group

Security Review of

Symphony Finance

September 2021

Symphony Finance / September 2021

Files in scope

https://github.com/symphony-finance/polygon-contracts/blob/75e66247601bb3ebb837e814555352c2496f318e/contracts/

- adapters/AaveYield.sol
- Symphony.sol
- oracles/ChainlinkOracle.sol
- handlers/SushiswapHandler.sol

Current status

All issues have been fixed by the developer. There are no known issues in the relevant contracts in https://github.com/symphony-finance/polygon-contracts/blob/80e434aa34100c945ce7a0261cc7fb24c5772b38

Issues

1. No access control on AaveYield.maxApprove

type: security / severity: major

Anybody can call `AaveYield.maxApprove`, this can lead to a DoS attack making `AaveYield.underlyingAssets` becomoning too expensive by making the `underlyingAssets` array too long.

status - fixed

Issue has been fixed and is no longer present in

https://github.com/symphony-finance/polygon-contracts/blob/80e434aa34100c945ce7a0261cc7fb24c5772b38/contracts

2. `AaveYield.orderRewardDebt` can be incorrectly set to `0` in `AaveYield.setOrderRewardDebt` causing overpaying of rewards

type: security / severity: critical

In `AaveYield.setOrderRewardDebt` on `line 198` `orderDebt` calculation is skipped if either `totalRewardBalance == 0` or `totalShares == 0`, with the assumption that zero value in either variable means no rewards have been paid out yet. This assumption is wrong with regards to both variables, `totalShares` can be lowered back to `0` after being a higher value before through share destruction and `totalRewardBalance` can equal `0` anytime no new rewards have been accumulated since last call of `AaveYield.setOrderRewardDebt` or `AaveYield.withdraw`.

status - fixed

Issue has been fixed and is no longer present in

https://github.com/symphony-finance/polygon-contracts/blob/80e434aa34100c945ce7a0261cc7fb24c5772b38/contracts

3. `AaveYield.withdrawAaveReward` and `AaveYield.claimReward` calls `incentivesController.claimRewards` without increasing `AaveYield.previousAccRewardPerShare` breaking the reward accounting

type: incorrect implementation / severity: major

Before `incentivesController.claimRewards` is called it's necessary to record the `incentivesController.getRewardsBalance` return value increase since the last time it was called into `AaveYield.previousAccRewardPerShare`, otherwise this increase won't be recorded in the internal accounting on the contract and the information will be lost because the counter is zeroed out in the `incentivesController` after `incentivesController.claimRewards` is called.

status - fixed

Issue has been fixed and is no longer present in

https://github.com/symphony-finance/polygon-contracts/blob/80e434aa34100c945ce7a0261cc7fb24c5772b38/contracts

4. `AaveYield.getRewardBalance` will return `0` when `isExternalRewardEnabled == false` breaking `AaveYield.getAccumulatedRewardPerShare` anytime some rewards are pending

type: incorrect implementation / severity: major

`line 305` will throw the safemath underflow error when `rewardBalance < pendingRewards[asset]`, this can happen when `isExternalRewardEnabled` is set to `false` after some rewards have been already counted in the `pendingRewards[asset]`.

status - fixed

Issue has been fixed and is no longer present in

https://github.com/symphony-finance/polygon-contracts/blob/80e434aa34100c945ce7a0261cc7fb24c5772b38/contracts

5. A `Symphony.emergencyWithdrawFromStrategy` call will break `Symphony.cancelOrder` until `rebalanceAsset` is called

type: security / severity: major

On `line 793` it's assumed that `orderAmount + neededAmountInBuffer` is always higher than `bufferAmount` at the time of withdrawal. This won't be the case if asset balance of the contract is suddenly increased without rebalancing. This happens in `Symphony.emergencyWithdrawFromStrategy` practically allowing an emergancy admin to freeze withdrawals at any time, this breaks security assumptions.

status - fixed

Issue has been fixed and is no longer present in

https://github.com/symphony-finance/polygon-contracts/blob/80e434aa34100c945ce7a0261cc7fb24c5772b38/contracts

6. `Symphony.assetBuffer` change without rebalancing can cause user actions to fail

type: incorrect implementation / severity: medium

If `Symphony.assetBufer` is updated without rebalancing the contract, user actions can unexpectedly fail due to `line 453`.

status - fixed

Issue has been fixed and is no longer present in

https://github.com/symphony-finance/polygon-contracts/blob/80e434aa34100c945ce7a0261cc7fb24c5772b38/contracts

7. An attacker can cause reward withdrawal to be skipped by increasing strategy contract asset balance

type: security / severity: medium

`amountToWithdraw` on `line 797` can be manipulated to be equal to `0` by an external attacker since it depends on the `Symphony` contract's asset balance, this will cause the reward payout to be skipped, this can also happen when `bufferAmount` changes without rebalancing.

status - fixed

Issue has been fixed and is no longer present in

https://github.com/symphony-finance/polygon-contracts/blob/80e434aa34100c945ce7a0261cc7fb24c5772b38/contracts

8. `ChainlinkOracle.get` will fail when `inputToken == address(0)`

type: incorrect implementation / severity: medium

In `ChainlinkOracle` a call on `line 58` will fail when `inputToken == address(0)`

status - fixed

Issue has been fixed and is no longer present in

https://github.com/symphony-finance/polygon-contracts/blob/80e434aa34100c945ce7a0261cc7fb24c5772b38/contracts

Issues discovered by the developer

9. Lack of separate accounting of rewards for different assets in AAVE breaks accounting in `AaveYield`

type: incorrect implementation / severity: major

Since AAVE only keeps track of reward totals for each user regardless of the asset that generated the reward, in `AaveYield` rewards won't be correctly assigned to shareholders of a given asset.

status - fixed

Issue has been fixed and is no longer present in

https://github.com/symphony-finance/polygon-contracts/blob/80e434aa34100c945ce7a0261cc7fb24c5772b38/contracts

10. In `Symphony.rebalanceAsset` rebalancing is incorrectly skipped when `assetBufferPercent == 10000`

type: incorrect implementation / severity: medium

When `Symphony.updateBufferPercentage` increases `assetBuffer` to `10000` rebalancing will be skipped due to `line 441`.

status - fixed

Issue has been fixed and is no longer present in

https://github.com/symphony-finance/polygon-contracts/blob/80e434aa34100c945ce7a0261cc7fb24c5772b38/contracts

Notes

- In order for users to not be taken by surprise by the actions of administrators `Symphony.owner` and `AaveYield.governance` should be timelocked contracts.