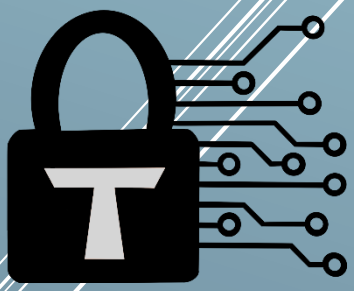
Trust Security

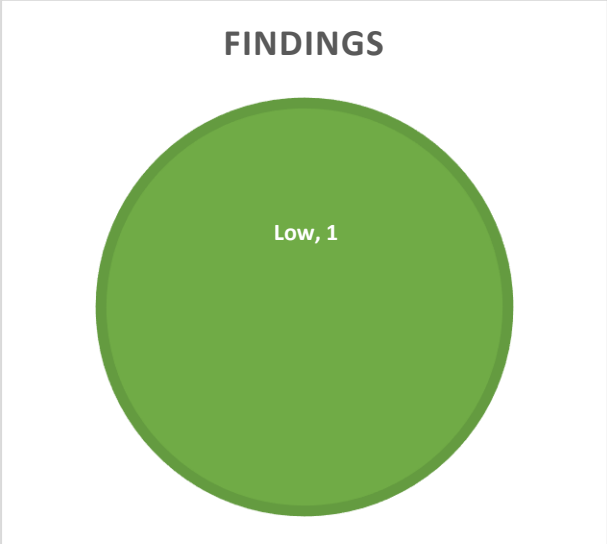


Smart Contract Audit

Reserve Protocol – MetaMorpho Collateral Plugins

02/05/24

Executive summary

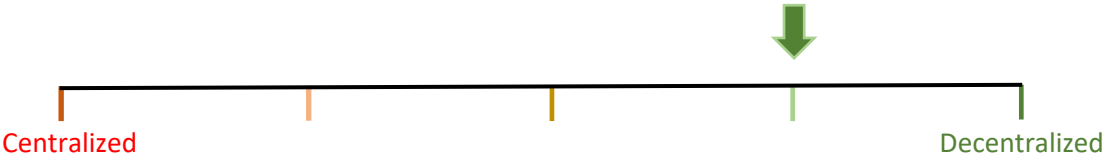


Category	Stablecoin
Auditor	HollaDieWaldfee gjaldon

Findings

Severity	Total	Fixed	Open	Acknowledged
High	-	-	-	-
Medium	-	-	-	-
Low	1	1	-	-

Centralization score



Signature

Digitally signed by Trust
Date: 2024.05.02 12:43:32 +03'00'

Document properties

Versioning

Version	Date	Description
0.1	30/04/2024	Client report
0.2	02/05/2024	Mitigation review

Contact

Trust

trust@trust-security.xyz

Introduction

Trust Security serves as a long-term security partner of the Reserve Protocol. It has conducted the audit at the customer's request. The audit is focused on uncovering security issues and additional bugs contained in the code defined in scope. Additional recommendations have been given when appropriate.

Scope

The following files are in scope of the audit:

- MetaMorphoFiatCollateral.sol
- MetaMorphoSelfReferentialCollateral.sol

In addition, the integration with the codebase at large and the deployment scripts for the collateral plugins have also been assessed.

Repository details

- **Repository URL:** https://github.com/reserve-protocol/protocol
- **Commit hash:** 5ff5c2beb67bfbfe16e80eee9303e0a6cea61080
- **Mitigation hash**: 33ca053ae28930fa9ec2dec99f364af3f0ebac9b

About Trust Security

Trust Security has been established by top-end blockchain security researcher Trust, in order to provide high quality auditing services. Trust is the leading auditor at competitive auditing service Code4rena, reported several critical issues to Immunefi bug bounty platform and is currently a Code4rena judge.

About the Auditors

HollaDieWaldfee is a renowned security expert with a track record of multiple first places in competitive audits. He is a Lead Senior Watson at Sherlock and Lead Auditor for Trust Security and Renascence Labs.

Gjaldon is a DeFi specialist who enjoys numerical and economic incentives analysis. He transitioned to Web3 after 10+ years working as a Web2 engineer. His first foray into Web3 was achieving first place in a smart contracts hackathon and then later securing a project grant to write a contract for Compound III. He shifted to Web3 security and in 3 months achieved

top 2-5 in two contests with unique High and Medium findings and joined exclusive top-tier auditing firms.

Disclaimer

Smart contracts are an experimental technology with many known and unknown risks. Trust Security assumes no responsibility for any misbehavior, bugs or exploits affecting the audited code or any part of the deployment phase.

Furthermore, it is known to all parties that changes to the audited code, including fixes of issues highlighted in this report, may introduce new issues and require further auditing.

Methodology

In general, the primary methodology used is manual auditing. The entire in-scope code has been deeply looked at and considered from different adversarial perspectives. Any additional dependencies on external code have also been reviewed.

Findings

Low severity findings

TRST-L-1 Small bad debt liquidations can cause hard-default in collateral plugins

- **Category:** Logical flaws
- **Source:** Morpho.sol (external), MetaMorpho deployment scripts
- **Status:** Fixed

Description

All MetaMorpho deployment scripts set **revenueHiding=1e-6**. This specifies that the **ref/tok** exchange rate can drop by a factor of one million before the collateral hard-defaults and is disabled.

MorphoBlue socializes bad debt, and so liquidations that create bad debt reduce **totalSupplyAssets**, which in turn causes **ref/tok** to drop.

revenueHiding=1e-6 is overly restrictive since for example, for a vault with $100 million In TVL, any amount of bad debt >$100 is sufficient to disable the collateral.

Recommended mitigation

To determine a suitable value for **revenueHiding**, it is useful to start with the APY of the vaults, which varies from 1% to 8%. An acceptable amount of interest that is lost before the collateral is disabled could be 1 week of interest. This gives a range of roughly **1%/52=0.00019 - 8%/52=0.0015**. As a result, **revenueHiding** should be **1e-4 – 1e-3** in terms of orders of magnitude to allow for small amounts of bad debt, while also being sensitive to **ref/tok** rate breaking.

It is possible to set different values for different vaults. However, **revenueHiding** cannot be changed after deployment, so the value must be suitable for different vault configurations and APYs.

Team response

Fixed.

Mitigation review

MetaMorpho vaults with their asset pegged to USD have **1e-4** revenue hiding and vaults with their asset pegged to ETH have **1e-3** revenue hiding. This is according to the recommendation and reflects the fact that Re7WETH has a higher interest rate than the USD based vaults.

Additional recommendations

Consider deprecating Re7WETH

According to the client, it is not intended to support high risk vaults that contain Liquid Restaking Tokens. However, **Re7WETH** currently allocates roughly 50% of its deposits to MorphoBlue markets with a LRT as collateral (**weETH**, **ezETH**).

Consider deprecating **Re7WETH** and reach out to the owners of the remaining vaults to determine if their risk-profile matches with what Reserve is willing to offer as collateral.

Remove duplicated sanity checks in ERC4626Collateral

ERC4626Collateral performs two sanity checks which are already performed in the downstream *FiatCollateral* and *Asset* parent contracts.

It is recommended to remove the redundant code in *ERC4626Collateral*.

Wrong comments state that tok==ref

The following two comments incorrectly state that **tok==ref**.

- https://github.com/reserve-protocol/protocol/blob/5ff5c2beb67bfbfe16e80eee9303e0a6cea61080/contracts/plugins/assets/meta-morpho/MetaMorphoSelfReferentialCollateral.sol#L14
- https://github.com/reserve-protocol/protocol/blob/5ff5c2beb67bfbfe16e80eee9303e0a6cea61080/contracts/plugins/assets/morpho-aave/MorphoSelfReferentialCollateral.sol#L15

It is recommended to replace **tok==ref** with **tok!=ref**. There is no issue in the smart contracts, only the comments are wrong.

Implement missing Etherscan verification scripts

There are no verification scripts for the **steakPYUSD** and **bbUSDT** vaults. It is recommended to implement them.

MorphoSelfReferentialCollateral should implement claimRewards() function

Morpho rewards are currently not transferable, so claiming rewards is not possible. Still, it is recommended to implement the *claimRewards()* function in *MorphoSelfReferentialCollateral* just like it is implemented in *MorphoFiatCollateral*. Thereby, once **Morpho** becomes transferable, the collateral plugin does not have to be upgraded.

MorphoBlue collateral has no de-peg checks

A MetaMorpho vault acts as a lender that provides liquidity to one or multiple MorphoBlue markets. Some MetaMorpho vaults only lend to MorphoBlue markets where the **loanToken** is pegged to the **collateralToken**. For example, the Re7WETH vault only has collateral that is pegged to WETH. Having a peg to the **loanToken** is not a requirement for a **collateralToken** and there exist vaults for which this does not hold true.

However, if vaults assume there exists a peg, this influences their risk management. And so, while non-pegged collateral is allowed on the smart contract level, a collateral de-pegging can lead to a loss for lenders due to bad debt liquidations.

The bad debt liquidations would eventually lead to a hard-default in the Reserve collateral plugin, due to the **ref/tok** rate decreasing. Nonetheless, Reserve generally aims to protect against de-pegs by soft-defaulting in case the price deviates more than **defaultThreshold** from the expected peg. Since issuance gets paused, this also prevents more **IFFY** collateral from accumulating in the RToken.

In practice, implementing de-peg checks is not straightforward due to the dynamic nature of MetaMorpho vaults.
The collateral can change at any time, so the collateral plugin would need to dynamically query the **supplyQueue** and **withdrawQueue**, determine for each collateral if it is pegged, and have an oracle available for each pegged collateral.

Overall, while it Is worthwhile to consider the benefits of de-peg checks, the overhead of performing the checks might not be feasible and they are not strictly needed since the **ref/tok** check exists.

Collateral plugins cannot be hot swapped

Changing the collateral plugin for a collateral that is contained in the active basket always disables the basket. This introduces overhead (switching the basket, waiting for warmup period) in the case that only a parameter needs to change like the maximum trading volume, which does not affect the basket composition.

To allow changing collateral plugins without interruption, it's possible to introduce another *unsafeSwapRegistered()* function that *Governance* can use if it's certain that the basket does not have to be disabled.

In discussion with the client, it has been suggested that hot swapping collateral plugins, and thereby not pausing issuance, can make RTokens more secure when used in external protocols, such as Lending protocols.

Systemic risks

MetaMorpho vaults involve trusted roles that can cause loss of funds

MetaMorpho has four trusted roles. These roles are not set by Morpho. Instead, Morpho only provides the smart contract infrastructure for anyone to deploy MetaMorpho vaults.

Currently, Reserve supports four MetaMorpho vaults which are governed by the following entities:

- bbUSDT: Block Analitica, B.Protocol
- steakUSDC: Steakhouse Financial
- steakPYUSD: Steakhouse Financial
- Re7WETH: RE7 Labs

Critical governance functionality is protected by a timelock delay between 1 day and 2 weeks, which is enforced by the smart contracts. However, even the timelock is insufficient to fully protect against malicious actions in the vaults, since RToken Governance is also timelocked. Actions by the RToken Governance are without effect if it takes longer to act than the MetaMorpho timelock.

In summary, integration with MetaMorpho vaults introduces trust assumptions for the roles controlling each vault, which notably are not held by Morpho itself. Users should be aware that the entities governing MetaMorpho vaults have the ability to cause loss of funds.

Beyond this worst-case scenario, it is recommended that Reserve and any RToken Governance using a MetaMorpho collateral set up monitoring for changes in the vault.