Custodia Security

Size Market Maker Review

Conducted By: Ali Kalout, Ali Shehab

Contents

1. Disclaimer

A smart contract security review cannot ensure the absolute absence of vulnerabilities. This process is limited by time, resources, and expertise and aims to identify as many vulnerabilities as possible. We cannot guarantee complete security after the review, nor can we assure that the review will detect every issue in your smart contracts. We strongly recommend follow-up security reviews, bug bounty programs, and on-chain monitoring.

2. Introduction

Custodia conducted a security assessment of Size’s smart contract following the implementation of the Market maker, ensuring the proper implementation of it.

3. About Size

Size is a lending marketplace with unified liquidity across maturities.

Size is built on an order book model where offers are expressed as yield curves, allowing efficient and continuous pricing of fixed-rate products while maintaining unified liquidity.

4. Risk Classification

Severity	**Impact: High**	**Impact: Medium**	**Impact: Low**
Likelihood: High	Critical	High	Medium
Likelihood: Medium	High	Medium	Low
Likelihood: Low	Medium	Low	Low

4.1. Impact

- High: Results in a substantial loss of assets within the protocol or significantly impacts a group of users.
- Medium: Causes a minor loss of funds (such as value leakage) or affects a core functionality of the protocol.
- Low: Leads to any unexpected behavior in some of the protocol's functionalities, but is not critical.

4.2. Likelihood

- High: The attack path is feasible with reasonable assumptions that replicate on-chain conditions, and the cost of the attack is relatively low compared to the potential funds that can be stolen or lost.
- Medium: The attack vector is conditionally incentivized but still relatively likely.
- Low: The attack requires too many or highly unlikely assumptions, or it demands a significant stake by the attacker with little or no incentive.

4.3. Action required for severity levels

- Critical: Must fix as soon as possible
- High: Must fix
- Medium: Should fix
- Low: Could fix

5. Security Assessment Summary

Duration: 25/01/2025
Repository: SizeCredit/size-periphery
Commit: ff1bcb20022075be2c9d7d1acdb7728413b99f81

- src/MarketMakerManager.sol
- src/MarketMakerManagerFactory.sol
- src/libraries/YieldCurvesValidationLibrary.sol

6. Executive Summary

Throughout the security review, Ali Kalout and Ali Shehab engaged with Size's team to review Size. During this review, one issue was uncovered.

Findings Count

Severity	**Amount**
Critical	N/A
High	N/A
Medium	1
Low	N/A
Total Finding	**1**

Summary of Findings

ID	Title	Severity	Status
M-01	Underlying borrow tokens could end up stuck in the MarketMaker contract	Medium	Resolved

7. Findings

7.1. Medium Findings

[M-01] Underlying borrow tokens could end up stuck in the MarketMaker contract

Severity:
Medium

Description:
Market Maker allows the emergency withdrawer or the owner to “force” withdraw a specific underlying borrow token from all available markets. However, `_emergencyWithdrawToken` wraps the withdrawal call in a try/catch block:

```
try markets[i].withdraw(
    WithdrawParams({token: address(underlyingBorrowToken), amount: borrowATokenBalance, to: owner()})
) {} catch {
    continue;
}
```

The “continue” in the catch block could lead to tokens not being transferred, as the transfer call is at the end of the markets loop.

Let's take the following edge case, assuming we have a market that has a unique underlying borrow asset, i.e. different from all other markets in the factory:

1. Some deposits are made through the manager, i.e. the contract now holds some aTokens.
2. Some underlying borrow tokens are sent to the contract, let's assume in preparation to call `depositDirect`.
3. For some reason, the corresponding market is paused.
4. An emergency withdrawal is initiated, but the tokens deposited in Step 2 won't be withdrawn

This is because the contract will attempt to withdraw => reverts, jumps into the catch block, where "continue" is fired, where the loop iteration is exited, so the transfer call won't be made.

Recommendations:
Remove the “continue” from the catch block.

7.2. Informational Findings

1. Deposit functions don't allow ETH deposits, while it's supported in the markets.
2. Emergency withdrawals can only withdraw the underlying borrow tokens, which makes sense as it's for LPs, however, the deposit functions don't block collateral deposits.
3. MarketMaker contract doesn’t have a way to recover ERC20s.