Custodia Security

Size v1.5 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 v1.5 and the migration for it, ensuring the proper implementation of both.

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

Repository: SizeCredit/size-solidity-private
Commits:

- 0e6f32a270b2c2ca940f35b90ce8ce65a172de23 (PR 5)
- 4d8fbfc488e79002e2514c0087c3dcc6346c146b (PR 7)

6. Executive Summary

Throughout the security review, Ali Kalout and Ali Shehab engaged with Size's team to review Size. The review was held from Nov 12 to 16. During this period, two issues were uncovered.

Findings Count

Severity	**Amount**
Critical	N/A
High	N/A
Medium	1
Low	1
Total Finding	**2**

Summary of Findings

ID	Title	Severity	Status
M-01	`Initialize::executeReinitializeV1_5` could be DOSed by donating ATokens to the contract	Medium	Resolved
L-01	`GetV1_5ReinitializeDataScript` is not batching users, forcing it sometimes to revert	Low	Resolved

7. Findings

7.1. Medium Findings

[M-01] `Initialize::executeReinitializeV1_5` could be DOSed by donating ATokens to the contract

Severity:
Medium

Description:
For the v1.5 migration, the protocol checks that the scaled supply of szTokens equals the scaled balance of ATokens for the contract, this is using the following condition: `(newScaledTotalSupplyAfter - newScaledTotalSupplyBefore) != aToken.scaledBalanceOf(address(this))`
This could be DOSed by sending any amount of ATokens to the contract, blocking the migration.

Proof of Concept:
```
function testFork_ForkReinitializeV1_5Audit_ATokenDonationReinitializeDOS() public {
    sizeFactory = _deploySizeFactory(owner);
    _deployNewBorrowAToken();

    USDC = sizeWethUsdc.data().underlyingBorrowToken;
    IAToken aUSDC =
IAToken(sizeWethUsdc.data().variablePool.getReserveData(address(USDC)).aTokenAddress);

    address aUSDC_holder = 0x4E32E08f3d8d1cAb474A489B4E31f8D3FD627abf;

    vm.prank(aUSDC_holder);
    aUSDC.transfer(address(sizeWethUsdc), 1e6);

    Supply memory old;
    _testFork_ForkReinitializeV1_5_migrate(
        "base-production-weth-usdc", sizeWethUsdc, priceFeedWethUsdc, addressesWethUsdc, old,
true
    );
}
```

Recommendations:
Have a more graceful condition, i.e. use `<` instead of `!=`.

7.2. Low Findings

[L-01] `GetV1_5ReinitializeDataScript` is not batching users, forcing it sometimes to revert

---

Severity:
Low

Description:
The protocol uses `GetV1_5ReinitializeDataScript` to fetch the holders of the Size v1.2 tokens, which uses `vm.eth_getLogs`, which fetches the holders from the deployment block to the current block. However, this has a limit of 10,000 and will sometimes reverts with the following error:
`query exceeds max block range`

Recommendations:
Batch the fetching process of holders to only do X blocks at a time, where X <= 10,000.