SHERLOCK

Security Review For MetaLend

Collaborative Audit Prepared For: **MetaLend**
Lead Security Expert(s): **jokr**
oot2k

Date Audited: **June 23 - June 27, 2025**
Final Commit: **a49a7a3**

Introduction

TBA

Scope

Repository: MetaLend-DeFi/metalend-rebalancing-contracts

Audited Commit: 3d75179ed7bb6682fe530a8da30fc5fcc365fa01

Final Commit: a49a7a331551daccfb5fa6f4c0d39c4f170ce418

Files:

- contracts/manager/RebalancingManager.sol
- contracts/manager/args/EIP3009Args.sol
- contracts/manager/args/RebalancingArgs.sol
- contracts/manager/args/RebalancingConfigArgs.sol
- contracts/manager/pool/PoolIdentifier.sol
- contracts/manager/pool/PoolToken.sol
- contracts/rebalancer/Rebalancer.sol

Final Commit Hash

a49a7a331551daccfb5fa6f4c0d39c4f170ce418

Findings

Each issue has an assigned severity:

- Medium issues are security vulnerabilities that may not be directly exploitable or may require certain conditions in order to be exploited. All major issues should be addressed.
- High issues are directly exploitable security vulnerabilities that need to be fixed.
- Low/Info issues are non-exploitable, informational findings that do not pose a security risk or impact the system's integrity. These issues are typically cosmetic or related to compliance requirements, and are not considered a priority for remediation.

Issues Found

High	Medium	Low/Info
0	0	2

Issues Not Fixed and Not Acknowledged

High	Medium	Low/Info
0	0	0

Issue L-1: Missing function to claim Aave incentives

Source: https://github.com/sherlock-audit/2025-06-metalend-june-24th/issues/8

Summary

Aave offers similar incentives on pools like morpho. Currently there is no way for users to claim these extra rewards.

For more details read: https://aave.com/docs/primitives/incentives
https://aave.com/docs/developers/smart-contracts/incentives

Vulnerability Detail

Currently there is no way to claim extra rewards from aave. A function should be implemented that allows to claim these using claimRewards() function on the RewardsController.

Impact

Loss of extra rewards on aave pools.

Code Snippet

https://github.com/sherlock-audit/2025-06-metalend-june-24th/blob/main/metalend-rebalancing-contracts/contracts/manager/RebalancingManager.sol

https://github.com/sherlock-audit/2025-06-metalend-june-24th/blob/main/metalend-rebalancing-contracts/contracts/rebalancer/Rebalancer.sol

Tool Used

Manual Review

Recommendation

Add a claim function for aave rewards.

Discussion

smrza

Resolved with
https://github.com/MetaLend-DeFi/metalend-rebalancing-contracts/pull/15

Adding support to claim AAVE incentives according to docs https://aave.com/docs/developers/smart-contracts/incentives allowbreak #write-methods-claimallrewards

jokrsec

Issue Resolved : Added a function to claim Aave incentives.

Issue L-2: Pool approval verification is not required while taking funds out of a pool

Source: https://github.com/sherlock-audit/2025-06-metalend-june-24th/issues/9

Summary

Currently, the `RebalancingManager` contract enforces pool approval verification even when **withdrawing funds from a pool** during withdrawals and rebalances (both same-chain and cross-chain). However, pool approval verification is not necessary when taking funds out of a pool, and enforcing it can cause problems in some cases.

This is because users can update or revoke their configuration off-chain at any time, signing a new `RebalancingConfigArgs` that excludes previously approved protocols. Once that happens, operators may no longer possess valid signatures allowing them to move or recover funds from those previously approved protocols, as the old signature will have been deleted from the off-chain infrastructure and operators won't have access to the old signature.

Vulnerability Detail

```
function rebalanceUsdcSrc(
    address onBehalfOf,
    uint32 destinationDomain,
    RebalancingArgs calldata rebalancingArgsFrom,
    RebalancingArgs calldata rebalancingArgsTo,
    RebalancingConfigArgs calldata rebalancingConfigArgs
) external override onlyOperator {
    // 1. ensure the rebalancer exists for the owner
    address rebalancerAddress = _getRebalancerEnsureCreated(onBehalfOf);

    // 2. validation, domain must be supported, config must be valid, cooldown must
↪   be passed
    _ensureDestinationDomainSupported(destinationDomain);

    // @audit this check is not necessary
    _verifyPoolsApprovedAndSupported(rebalancingArgsFrom, rebalancingConfigArgs,
↪   thisDestinationDomain, _usdcToken);
    // this ensures at least 1 pool is approved for the destination domain
    _verifyPoolsApprovedAndSupported(rebalancingArgsTo, rebalancingConfigArgs,
↪   destinationDomain, _usdcToken);
    _verifyConfigSignature(onBehalfOf, rebalancingConfigArgs);
    _ensureCooldownPassedAndUpdate(rebalancerAddress);

    ...
```

```
}
```

Example Scenario

1. A user initially approves both the AAVE USDC pool and Morpho USDC pool.
2. The user's funds are currently in the Morpho USDC pool.
3. While the funds are still in Morpho, the user removes Morpho from their `RebalancingConfigArgs` and creates a new signature.
4. The new signature replaces the old signature in the off-chain infrastructure, and operators no longer have access to the old signature.
5. To withdraw funds from Morpho and deposit into a new pool, operators would need a `RebalancingConfigArgs` signature that includes Morpho.
6. Since operators no longer have access to such a signature, rebalancing will be blocked until the user share the old signatues again.

Impact

Rebalancing will be temporarily **DoS'd**, preventing operators from moving funds out of the old pool until the user provides an updated signature.

Code Snippet

https://github.com/sherlock-audit/2025-06-metalend-june-24th/blob/main/metalend-rebalancing-contracts/contracts/manager/RebalancingManager.sol#L151
https://github.com/sherlock-audit/2025-06-metalend-june-24th/blob/main/metalend-rebalancing-contracts/contracts/manager/RebalancingManager.sol#L213

Tool Used

Manual Review

Recommendation

Remove pool approval verification when **withdrawing funds from a pool**. Approval checks should only be done when **depositing funds into a pool**.

Discussion

smrza

Resolved with
https://github.com/MetaLend-DeFi/metalend-rebalancing-contracts/pull/14

Changing `_verifyPoolsApprovedAndSupported` for src arguments to `_ensurePoolContractSupported` for functions `rebalanceUsdcSrc` and `rebalanceUsdcThisDomain` so that only pool contract support is ensured and user configuration check is skipped while withdrawing funds.

jokrsec

Issue Resolved: Removed pool approval verification for from pool during rebalances and only doing pool contract support check now.

Disclaimers

Sherlock does not provide guarantees nor warranties relating to the security of the project.

Usage of all smart contract software is at the respective users' sole risk and is the users' responsibility.