SHERLOCK

Security Review For Arcadia Finance

Collaborative Audit Prepared For: **Arcadia Finance**
Lead Security Expert(s): **ge6a**
thekmj
Date Audited: **September 26 - September 29, 2025**

Introduction

Arcadia is a next-generation liquidity management protocol for CLAMM DEX. Through user-owned credit accounts, it provides: Simple one-click transactions to enter sophisticated high yield strategies, normally reserved for professionals. Automation of asset and risk management without giving up on self custody. Built-in margin (credit). An interface for third parties and AI agents to manage assets within onchain enforceable boundaries.

As the main liquidity infrastructure on Base, Arcadia is expanding to other L2s to bring the simplified LP experience to more users.

Scope

Repository: arcadia-finance/accounts-v2

Audited Commit: 611b57db1cacc4ba55ab2e857173bae8c101297e

Final Commit: 611b57db1cacc4ba55ab2e857173bae8c101297e

Files:

- src/accounts/AccountV3.sol
- src/accounts/AccountV4.sol
- src/accounts/helpers/AccountsGuard.sol
- src/asset-modules/Aerodrome-Finance/StakedAerodromeAM.sol
- src/asset-modules/Slipstream/StakedSlipstreamAM.sol
- src/asset-modules/Stargate-Finance/StakedStargateAM.sol
- src/Factory.sol
- src/libraries/CreateProxyLib.sol
- src/registries/RegistryL1.sol

Repository: arcadia-finance/asset-managers

Audited Commit: d079e18142a758f23caeda735cc3d7df630ec91c

Final Commit: 7aea6c12b63ebfc0ce22959222ab8f57e36d89fb

Files:

- src/cl-managers/compounders/Compounder.sol
- src/cl-managers/rebalancers/Rebalancer.sol
- src/cl-managers/yield-claimers/YieldClaimer.sol
- src/merkl-operator/MerklOperator.sol

Repository: arcadia-finance/lending-v2

Audited Commit: 635bc0b55de80de60c43396839c2391eb2a537b7

Final Commit: 9a5db6c74e31aaaa05a3d4d42bae51115cb2dea8

Files:

- src/LendingPool.sol
- src/liquidators/LiquidatorL1.sol

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	0

Issues Not Fixed and Not Acknowledged

High	Medium	Low/Info
0	0	0

We found no method to bypass the cross-account re-entrancy guard. In order to prematurely unlock the guard, the initiating account must directly call `AccountsGuard.unLock()` (other than from the modifiers). No instances of such were found. Note that account versions 1 and 2 do not trigger the cross-account guard. However the new Asset Managers no longer support these older account versions.

We found no method to reproduce the exploit post-fix. In order for the exploit to be possible, the account owner must have whitelisted the Asset Manager. There are currently three Asset Managers developed by Arcadia team (Compounders, Rebalancers, Yield Claimers): For all of the Asset Managers, `executeAction()` is an account-only entrypoint. This is blocked by reentry guard For Compounder, `compound()` is an initiator entrypoint For Rebalancer, `rebalance()` is an initiator entrypoint For Yield Claimer, `claim()` is an initiator entrypoint In all of the initiator entrypoints, `flashAction()` is always called and will lock the Asset Manager behind the re-entrancy guard as a result, ensuring that only the initiating account can ever execute action. There's also no controllable external calls made before `flashAction()`.

In addition, swaps are now routed through the Router Trampoline, which is not an asset manager and has no ability to control any account's assets. This was applied to both Compounder and Rebalancer.

Flash actions now have an additional restriction of not being able to call into any accounts. That means Asset Managers must adhere to this restriction. This may block some obscure use cases where an asset manager can manage multiple accounts at once (within one action). - This is relevant only to third-party Asset Managers. All Arcadia supported Asset Managers currently make no such action.

Regarding deployment AMs on different chains, CREATE opcode is dependent on the deployer address, and the deployer nonce, independent of contract code. Hence deployment order should just be the same as on the original chain.

The other items we have found no issues on.

All `onSetAssetManager()` only calls into `_setAccountInfo()`, which is the same as if the account owner calls `setAccountInfo()` themselves.

Asset manager whitelisting is owner-dependent, so someone calling remove asset manager from their whitelist does not affect whitelistings of other users.

Merkl-claiming with duplicate tokens still works correctly, as second instance onwards of the same token will just short-circuit and no-op (the whole claimed token balance would've been processed before).

There's one noteworthy item regarding MerklOperator: For a given reward token, the supported Merkl Operator will only work for an account if (1) it is the only Merkl operator of said account (of that token), and (2) itself is set as the token's receiver on the Merkl Distributor.

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.