SHERLOCK

Security Review For Sentiment

Collaborative Audit Prepared For: **Sentiment**
Lead Security Expert(s): **0x52**
hash

Date Audited: **March 8 - March 10, 2025**
Final Commit: **63019a5**

Introduction

Sentiment is a credit protocol focused on leveraged lending, and onchain yield strategies. This review is focused on the oracles needed for the upcoming launch on HyperLiquid.

Scope

Repository: sentimentxyz/protocol-v2

Audited Commit: 05f0dd4a25923c16e90028ea56cc36776095741e

Final Commit: 63019a5f7e1d032ddf2145ee74417c9e64d30ce5

Files:

- src/oracle/AggV3Oracle.sol
- src/oracle/HyperliquidOracle.sol
- src/oracle/HyperliquidUsdcOracle.sol

Final Commit Hash

63019a5f7e1d032ddf2145ee74417c9e64d30ce5

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	1

Issues Not Fixed and Not Acknowledged

High	Medium	Low/Info
0	0	0

Issue L-1: `SuperPoolReallocated` emits skipped deposits as well

Source: https://github.com/sherlock-audit/2025-03-sentiment-v2-hyperliquid-oracle-update/issues/4

Summary

`SuperPoolReallocated` always emits the deposits array as is without taking into consideration that some deposits might be skipped

Vulnerability Detail

Deposits passed in the input array can get skipped in case the deposit will surpass the poolCap. But the `SuperPoolReallocated` event emits the deposits array as-is without considering this and hence will contain deposits they weren't actually executed

```
function reallocate(ReallocateParams[] calldata withdraws, ReallocateParams[]
↪  calldata deposits) external {

    .......

    for (uint256 i; i < depositsLength; ++i) {
        uint256 poolCap = poolCapFor[deposits[i].poolId];

        .......

        uint256 assetsInPool = POOL.getAssetsOf(deposits[i].poolId, address(this));
        if (assetsInPool + deposits[i].assets <= poolCap) {
            ASSET.forceApprove(address(POOL), deposits[i].assets);
            POOL.deposit(deposits[i].poolId, deposits[i].assets, address(this));
            idleAssets -= deposits[i].assets;
        }
    }
    emit SuperPoolReallocated(withdraws, deposits);
}
```

Impact

Event emission incorrectness and possible incorrect offchain interpretations

Code Snippet

https://github.com/sherlock-audit/2025-03-sentiment-v2-hyperliquid-oracle-update/blob/b3a8627c9802d7ba288627d7487021f4af821e43/protocol-v2/src/SuperPool.sol#L456-L462

Tool Used

Manual Review

Recommendation

Emit only the deposits that were actually executed

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.