Audit Report – Panoptic

Repository: [https://github.com/code-423n4/2025-06-panoptic]

Date: [12.01.2026]

Auditor: [Harisuthan]

Summary:
Total issues found: [1(high and medium included)]
Severity breakdown: [1 high]

Issue M-1 : `PoolExposure` Calculation Bug in computeNAV causes NAV miscalculation

Summary:

The computeNAV function in PanopticVaultAccountant.sol is intended to calculate the correct net asset value (NAV) of a vault by properly accounting for assets and liabilities. However, due to a reversed calculation for poolExposure1, the NAV is miscalculated, causing users to potentially receive incorrect shares or assets.

Vulnerability Detail:

Context:

The vault tracks option positions via getAccumulatedFeesAndPositionsData, which returns:

shortPremium → asset for the vault (money the vault earned)

longPremium → liability for the vault (money the vault may owe)

How the bug occurs:

For token0, exposure is correctly calculated as:

poolExposure0 = int256(uint256(shortPremium.rightSlot())) - int256(uint256(longPremium.rightSlot()));

✓ asset – liability

For token1, the calculation is reversed:

poolExposure1 = int256(uint256(longPremium.leftSlot())) - int256(uint256(shortPremium.leftSlot()));

❌ liability – asset

Why it's wrong/risky:

Pool exposure for token1 becomes negative when it should be positive (or vice versa)

NAV calculation sums poolExposure0 + poolExposure1 → resulting NAV is wrong

Deposits and withdrawals rely on NAV → users may get too many or too few shares or assets

Impact:

As a result of this bug, vault users and the protocol may experience:

Incorrect share allocations for deposits (sharesReceived)

Incorrect asset amounts for withdrawals (assetsReceived)

Economic imbalance between vault participants
This can lead to financial loss or unfair value distribution.

Proof of code

```
int256 poolExposure0;
int256 poolExposure1;

{
    LeftRightUnsigned shortPremium;
    LeftRightUnsigned longPremium;
```

```
        (shortPremium, longPremium, positionBalanceArray) = pools[i]
            .pool
            .getAccumulatedFeesAndPositionsData(_vault, true, tokenIds[i]);

        // Correct for token0
        poolExposure0 = int256(uint256(shortPremium.rightSlot())) - int256(ui

        // Incorrect for token1
        poolExposure1 = int256(uint256(longPremium.leftSlot())) - int256(uint
    }
```

Recommendation:

Calculate poolExposure1 the same way as poolExposure0: