FlatcoinVault::wrong accounting of net PnL

Summary

There is a wrong accounting of net PnL in updateGlobalPositionData() in FlatcoinVault.sol contract.

Vulnerability Detail

In updateGlobalPositionData() there is a internal call to _updateStableCollateralTotal() where net PnL is passed as argument. However, the net PnL is passed as negetive value:

```
        _updateStableCollateralTotal(-profitLossTotal);
```

While calculating net PnL we are substracting the previous price, i.e the price when the position was opened, from current price of asset, here:

```
        int256 priceShift = int256(price) -
int256(globalPosition.lastPrice);
```

So, if the price goes down then the priceShift will be negative, as a result the net PnL will be negative too. because net PnL is calculated like this:

```
        return (int256(globalPosition.sizeOpenedTotal) * (priceShift)) /
int256(price);
```

So, if the net PnL is negative and as it is passed with -ve sign to _updateStableCollateral() it will increase the stable collateral total amount instead of decreasing it & if positive then it will decrease stable collateral total.

Impact

As net PnL is passed to _updateStableCollateral() with -ve sign it will increase the stable collateral total instead of decreasing and decrease instead of increasing it.

PerpMath::borrowing fee is not deducted from margin collateral during margin settlement calculation

Summary

In _getPositionSummary() of PerpMath.sol library the borrowing fee is added to margin collateral instead of deducting while calculating the marginAfterSettlement of PositionSummary struct.

Vulnerability Detail

In PerpMath.sol library we have a function named _accruedFunding() which returns the borrowing fee of a particular position. The borrowing fee should be deducted from the collateral of the trader and sent to LPs. But in _getPositionSummary() while calculating the marginAfterSettlement we are adding it to margin collateral instead of deducting it. The problematic line is this:

```
                marginAfterSettlement: int256(position.marginDeposited) +
profitLoss + accruedFunding
```

Impact

As the code adding the borrowing fee instead of deducting the fee will not be deducted from the margin collateral of that trader during liquidation. Additionally the check, for whether the position is liquidateable or not, may fail for that position because in PerpMath::_canLiquidate() the marginAfterSettlement is accounted to check whether the marginAfterSettlement is greater than lMargin or not and depending on this comparison the boolean value is returned, we can see the code here, so if marginAfterSettlement is calculated wrongly, i.e if the accruedFunding is added to it instead of deducting from it it's value will increase so may the value of margineAfterSettlement be greater than lmargin so the position, which should be liquidate, will not be liquidated.