Yieldoor Security Docs

Things to look out for:

- Deposit & Withdraw should always result in the same state.
    - Key moment to look at is when pool is near any position boundary
    - If a depositor has previously had a deposit, and idle funds are insufficient to cover it, they can technically deposit/ withdraw and remove all liquidity. This should not allow them to steal any funds. If anyone starts abusing this, a deposit fee would be introduced.
- Vault Withdraws should never fail
- Any issue arising if Vesting position has the same tick boundaries as any of the other two positions?
- Proper decimals scaling. Due to different needs in different parts of the contract, Oracle (USD) pricing is in 1e18, Lending Pool Indexes are in 1e27 and Vault price is in 1e30
- currBorrowedUSD for a market should be higher or equal to the sum of all positions' `initBorrowedUsd`.
- Assume reported Chainlink price might be up to 1% off. Biggest impact this should allow is to open a slightly higher leverage position than what usually should be the limit (e..g limit is 2x and user opens a 2.02x position). Look for scenarios where user can abuse this to open a position which is instantly underwater

Price Oracles:

- At least one of the vault assets must have a trustworthy price feed
    - In order to be considered trustworthy it must either have a Chainlink feed or have UniV3 pool with high enough liquidity

Loan repayments:

- Loan repayments are rounded down. Over time, a few wei could remain as permanent debt. This number should be low enough to not care about it.

Liquidations explained:

- The regular flow is that whenever the user's collateral drops below the `minCollateralPct`, they become liquidateable
- However, in scenarios where the users are not using max leverage, this could cause unfair liquidations. Assume minPctCollateral is 10%. Consider the following scenario: max Leverage is 2x, user has opened a 1.25x and has since then lost 90% of their collateral, meaning that 0.1X collateral has remained and 0.25X still borrowed. We treat the borrowed amount as if it was opened at max leverage, which would make the initial collateral `0.25 * (2 - 1) = 0.25X`. Since minimum collateral pct is 10%, this would mean that the position will become liquidateable once collateral drops below `0.25 * 0.1 = 0.025X`

- To put it into a numerical example, user opens up a $1,250 position with $1,000 of collateral (1.25x leverage). User's position would have to fall below $275 value ($250 borrowed, $25 collateral) in order to become liquidateable.

Known/ Acceptable risks:

- User can spam `collectFees` each block to lose a few wei from each withdraw from vested position. Even in scenarios where high-value low-decimal token is used (WBTC), this is an accepted risk.
- Strategy.balances does not account for uncollected fees. Any logic depending on the fees, must call collectFees before that.
- In order for a user to borrow an asset, it has to be enabled as a borrowAsset within the LendingPool, even if it is not the denomination asset.
- Lending pool reserves should have enough extra funds to cover for the `pullFunds`.
- Although the used router interfaces are for mainnet, there are no intentions to deploy on mainnet. We're aware that on other L2s, the router interface might be slightly different (swaps don't have `deadline` parameter)
- Loan repayments are rounded down. Over time, a few wei would remain as permanent debt. This number should be low enough to not care about it.
- Setting fee recipient to blacklisted address is not an issue
- Yes, if everyone withdraws while there is a vesting position going on, this is not an issue
    - Same applies if the entire Vault is made up from a leveraged position and it gets liquidated.
- frequent updates in LendingPool might result in slight differences in the actual interest rate.
- in situations where position's price has dropped significantly for a very short time and has then came back within safe levels, due to TWAP lagging behind it might still be liquidateable.
- If an asset's "true price" drops significantly (such as LST getting slashed, or erc4626-like token's share rate dropping due to bad debt/ liquidation), user is able to open significantly underwater position in the same block the slashing/ liquidation has happened. This is an accepted risk and for this reason leveraging would be limited on assets which come with such underlying risk.
- changing .minCollaterPct might result in some positions becoming instantly liquidateable. Updates to this value will only happen with prior announcing, making sure all users have enough time to react.
- `checkPoolActivity` does intentionally check 1 extra observation before `lookAgo`
- In the rare event Vault's deposit fee is set to 0.01%, Leverager will not adjust to it (will overvalue the collateral by 0.01%)