Contracts vulnerabilities

Vulnerabilities list:

Involved contracts and level of the bugs

The present document aims to point out some vulnerabilities in the contracts.

Vulnerabilities

1. `Deposit` function

Severity: Medium

The following method is implemented in the liquidity_lockbox_v1 program:

```
pub fn deposit(ctx: Context<DepositPositionForLiquidity>, id: u32)
```

This method facilitates the deposit of concentrated liquidity tokens NFTs from Orca whirlpool contracts, enabling users to receive fungible token equivalents.

When depositing, the user needs to specify the identifier (id) which must align with the current total number of lockbox positions. Consequently, when attempting multiple deposits with the same identifier, the method encounters complications. Specifically, upon initiating a second deposit with an identical id and without a corresponding increase in the supply, the method fails. This problem hinders successful execution of subsequent deposits in the same slot using the same id.

2. `Withdraw` function

Severity: Medium

The following method is implemented in the liquidity_lockbox_v1 program:

```
withdraw(
    ctx: Context<WithdrawLiquidityForTokens>,
    id: u32,
    amount: u64,
    token_min_a: u64,
    token_min_b: u64
  )
```

This method facilitates a liquidation process, enabling users to exchange a designated quantity of fungible tokens, which represent liquidity NFTs within the contract, for the corresponding liquidated assets tied to those liquidity NFTs. However, the current implementation of this method presents certain issues.

1. When attempting multiple withdrawals from either the same position or with the same identifier (id), the method encounters difficulties. Specifically, starting from the second withdrawal, any subsequent attempt to withdraw a specified amount from the same position or with the same id, lacking sufficient liquidity to support multiple withdrawals, results in a failure.

2. Users retain the flexibility to choose both the position and identifier (id) from which they intend to withdraw. A crucial point to note is that a user may select a position with higher liquidity than required. This choice can force a user with a larger amount of fungible tokens to do multiple withdrawals.