Contracts vulnerabilities

Vulnerabilities list:

Contracts vulnerabilities **1**
Vulnerabilities list: 1
Involved contracts and level of the bugs 1
Vulnerabilities 1
1. Withdraw function 1

Involved contracts and level of the bugs

The present document aims to point out some vulnerabilities in the autonolas-tokenomics-solana contracts.

Vulnerabilities

1. `Withdraw` function

Severity: Informative

The following function is implemented in the liquidity_lockbox.sol contract:

```
function withdraw(uint64 amount) external
```

This method facilitates a liquidation process, allowing a user to exchange a specified quantity, X, of fungible tokens representing liquidity NFTs in the contract, for the liquidated assets making up the liquidity NFTs.

With current implementation, only if the user liquidates 100% of one position, e.g. withdraw(X) and least recent NFT liquidity L in the contract is equal to X, they are able to accrue fees for such a liquidity. So if one user withdraws an amount X corresponding to 99% of the liquidity of the least recent NFT, they won't receive fees, which will be accrued by the user withdrawing an amount X corresponding to the remaining 1% of the liquidity of the least recent. Since with Orca rewards and fees are harvestable at any time (cf. https://docs.orca.so/reference/trading-fees) there is a possibility to improve such behavior providing the proportional share of the fees. However, given that the primary mean of these fungible token is to participate in bonding, it is likely that the 100% of

fungible tokens will owned by the protocol contract, in which case, it is likely to be able to call withaws with an amount equal to the full amount of the last wrapped nft token.