Beraborrow LP Pricing Bug

---

1. Context

- **Beraborrow** = Liquity V1 fork on Berachain.
- Lets users deposit **BEX LP tokens** (Balancer-style) as collateral to borrow NECT stablecoin.
- Needs a safe formula to price LP tokens.

---

2. Two LP Pricing Methods

A. Naive Pricing (manipulable)

```
naive pricing = (V1 + V2) / LP_totalSupply
```

- `V1`, `V2` = value of pool assets.
- Problem: Easily inflated with flashloans or reserve imbalance → overvalues LP tokens.

B. Fair Reserve Pricing (lower bound)

```
fair price = ( 2 * sqrt(r0 * r1 * p0 * p1) ) / totalSupply
```

- `r0`, `r1` = pool reserves.
- `p0`, `p1` = fair prices from oracle.
- Safe for **borrowing** (prevents manipulation).
- **Risk:** Undervalues LP tokens in **redemption** if pool is imbalanced → value leak.

---

3. The Vulnerability

Beraborrow used **Fair Reserve Pricing** for:

1. **Borrowing power** ✅ good.
2. **Redemption** ❌ risky.

Why it's a problem:

- Fair pricing = *minimum* value.
- Actual redemption returns the **real reserves**, which can be worth more than the fair price.
- The difference = **free profit** for redeemer → protocol loses value.

---

4. Small Example

Pool: USDC / USDT LP token

- Ideal balanced: `1.0 USDC + 1.0 USDT`.
- Actual reserves: `1.10 USDC + 0.91 USDT`.

```
Fair price = min(1.10, 0.91) * 2 = $2.00
Redemption output = 1.10 + 0.91 = $2.01
Leak per LP = $0.01
```

→ With large LP positions & repeated cycles = massive loss.

---

5. How Fuzzing Escalated the Bug

- **Setup:** Forked Balancer factory, deployed real pool, exposed handlers to:
  - Add/remove liquidity
  - Swap tokens
  - Withdraw LP
- **Target function:** Compare *fair price* vs. *actual redemption value* → return difference.
- **Echidna optimization mode:** Maximized the gap.
- Found extreme hypothetical case: `~1.81e21` difference (test env, unclamped inputs).
- Real-world gap smaller but still exploitable.

---

6. Impact

- Without a **high redemption fee**, attackers can continuously:
  - Supply imbalanced LPs → redeem for more than priced.
  - Repeat for arbitrage profit.

- Risk: Continuous drain of collateral value from protocol.

---

7. Fixes & Mitigation

- Stopped using **on-chain fair reserve pricing** for redemptions.
- Switched to **Chronicle oracle** for LP pricing.
- Increased **minimum redemption fee** to cover potential inaccuracy.

---

8. Key Lessons

1. **Correct formula in wrong context** can still be a bug.
2. Fair Reserve Pricing →
   - ✅ Borrowing: safe.
   - ❌ Redemption: value leak.
3. Manual review + fuzzing = best combo:
   - Manual: find novel logic flaw.
   - Fuzzing: quantify worst-case impact.