BeraBorrow Rounding Issue

---

1. Liquity Recap (Core Mechanics)

Liquity is an ETH-backed borrowing protocol:

- **Open Trove** → Lock ETH, borrow LUSD.
- **Overcollateralization** → CR (Collateralization Ratio) = Collateral Value ÷ Debt Value.
  - Must stay ≥ **110%** in normal mode.
- **Liquidations** → If CR < 110%, trove is liquidated; debt absorbed by Stability Pool.
- **TCR** (Total Collateral Ratio) = Total Collateral Value ÷ Total Debt Value.
  - Measures *system-wide* health.

---

2. Recovery Mode

- Trigger: **TCR < CCR (150%)**.
- Changes rules instantly:
  - Liquidation threshold jumps from 110% → 150%.
  - Even “safe” troves can be liquidated.
- Purpose: Quickly remove systemic risk.
- Protections:
  - Users can’t maliciously self-liquidate profitably.
  - No operation should directly *trigger* Recovery Mode unless the system is already close to CCR.
  - Cannot add more risky debt while in Recovery Mode.

Example:

- Normal mode: Bob at CR = 140% is safe.
- Recovery mode: Bob at CR = 140% is liquidatable.

---

3. Beraborrow’s Vault Accounting

Beraborrow integrated vault-like (ERC4626-ish) share accounting in its *Den Manager*, which manages user-specific vaults holding collateral.

- **Shares** represent proportional claim on assets.
- **PPFS** (Price Per Full Share) = `totalAssets / totalShares`.
- Should only change with deposits/withdrawals in a predictable way.

---

4. The Fuzzer's Finding

They ran a **global invariant**:

```
PPFS after any user action ≥ PPFS before (except for legitimate withdrawals)
```

The fuzzer found a **PPFS drop** after a redemption:

- totalAssets ↓ by 2
- totalShares ↓ by only 1
  This meant **more shares were in circulation than assets backing them**.

---

5. Root Cause

- Fees were collected **in shares**, not assets.
- Fee share amount was **rounded up**.
- When redeeming, instead of burning `X` shares, it burned `(X - fee)` shares —
  rounding caused **1 wei of shares not to be burned**.
- Result:
  Assets drop more than shares → PPFS decreases slightly.

Example:

```
Before: totalAssets = 100, totalShares = 100 → PPFS = 1.0
After:  totalAssets = 98, totalShares = 99 → PPFS ≈ 0.9899
```

This tiny PPFS drop affects **all depositors** system-wide.

---

6. Why It's Dangerous

PPFS is used to value collateral in the system.
If PPFS drops, **every trove's collateral is valued less**, lowering TCR.

If TCR was near CCR (150%), a small PPFS drop could:

1. Push TCR < 150%.
2. Instantly trigger Recovery Mode.
3. Mass-liquidate troves between 110%–150% CR.

---

7. The Attack Chain

Step 1 – Setup:
Attacker opens a large trove, borrowing enough to push TCR to just above 150%.
Protocol check prevents going below CCR directly.

Step 2 – Trigger the Drop:
Attacker redeems in a way that exploits the rounding bug, causing a **1 wei share leftover** → PPFS drops → TCR falls below 150%.

Step 3 – Recovery Mode On:
Instant state change: liquidation threshold = 150% CR.
Many "safe" troves now liquidatable.

Step 4 – Liquidate Others:
Liquidate from healthiest (just above 150%) to unhealthiest to maximize profits.

Step 5 – Exit:
Close attacker's trove after collecting rewards.

---

8. Profits vs. Costs

- **Costs:**
  - Gas
  - Trove opening fee
- **Gains:**
  - 0.5% liquidation bonuses on all liquidated collateral.
  - Additional profit if attacker is in Stability Pool (gets seized ETH from liquidations).
  - Flashloan → Temporarily inflate Stability Pool position to grab more liquidated collateral.

9. Why Manual Reviews Missed It

- 1 wei rounding difference is hard to spot in a manual review.
- Looks like a small accounting quirk, not an exploit path.
- Without fuzzing, it’s unlikely an auditor would:
  - Notice the PPFS drop
  - Connect it to TCR mechanics
  - Chain it into Recovery Mode mass-liquidations

---

10. Key Takeaways

1. **Tiny math errors can have protocol-wide consequences** if they impact global metrics.
2. **Fuzzing excels at catching edge cases** that are near-impossible to see via manual reasoning.
3. **Low severity → high severity**: A small bug can be chained with protocol rules for large-scale profit.
4. Always check:
   - Price per share invariants
   - How global thresholds (like CCR) are computed
   - Whether attacker-controlled state changes can trigger systemic mode shifts

---