Scenario 4: underwater debt + attacker drops TCR to CCR with huge trove + redemption triggers RM

1. Attacker waits for the system to reach a state with underwater Troves (with CRs < MCR) present, and the underwater debt greater than the LUSD in the SP
2. Attacker brings TCR = MCR with huge Trove
3. Attacker performs redemption to drop TCR below CCR
4. Attacker liquidates q remaining debt fraction with CR < CCR, get SP gain + coll surplus

Terms

D: initial total system debt
D': total system debt after attacker opens huge trove T
S: total LUSD in SP before liquidation
d_A: total underwater debt before attack i,e. with CR_A <= MCR
d_T: debt of attacker's huge trove T
b: huge trove size relative to initial total system debt
d_B: total redeemed amount
k: Initial ratio of SP to total debt
s: Attacker's share of SP before liquidation
f: Initial ratio of underwater debt to total debt, i.e. 0<f<=1
p: Fraction of initial total debt redeemed by attacker
q: Fraction of remaining debt after redemption that can be liquidated in RM

In this attack the attacker can engineer the underwater debt, the huge Trove and the TCR drop with a redemption. They don't have to wait for an ETH price drop or low/empty SP.

However, we still assume a worst-case distribution of Troves (a highly unlikely state) in order to obtain a lower bound on the redemption size.

In order to drop TCR to CCR, we assume the attacker creates a huge Trove T with collateral ratio CR_T such that MCR <= CR_T <= CCR.

We know from Scenario 2, equation 9 - that it is necessary (but not sufficient) for this condition on the aggregate redemption CR_B to hold: CR_B > CCR.

Given that CR_T <= CCR, then the attacker must necessarily fully redeem their own Trove as part of the redemption.

This gives us a lower bound on the redemption size. That is:

1) $d_B >= d_T$

i.e.

2) $p >= b$

And we assume the minimum i.e.

3) $p = b$.

Scenario 3, equation 32 gives b as a function of initial system TCR, and we assume the TCR falls to the CCR, that is the TCR buffer m = 0:

4) $b = (TCR - CCR)/(CCR - MCR)$

i.e.

5) $p = (TCR - CCR)/(CCR - MCR)$

This is also a lower bound in another way - it assumes the attacker trove T has $CR_T = MCR$. If their CR is higher, then the Trove size and min redemption size would be higher.

Profit calculation - pre-existing underwater debt

The attacker's profit is given by:

6) profit = liq_SP_gain + liq_gas compensation - attacker_borrow_cost - redemption_cost

Let's look at the quantities involved:

Liquidated debt

The debt liquidated is the underwater debt $d_{A,}$, plus a fraction q of the remaining debt in the system after redemption which gets dragged down below the CCR. The remaining debt thus excludes both the underwater debt d_A and the redeemed debt d_B.

(We can ignore the debt of the huge Trove d_T, since this is already included in the redeemed debt d_B)

So:

7) $liquidated\_debt = d_A+q(D - d_A - d_B)$

8) $liquidated\_debt = fD +q(D - fD - pD))$

9) $liquidated\_debt = D(f + q(1 - f - p))$

Amount liquidated against SP

The amount liquidated against the SP is the lesser of the amount in the SP and the total liquidated debt, that is:

10) liq_SP = D*min(k, (f +q(1-f-p))

Attacker's SP liquidation gain

The attacker owns a fraction s of the SP, and receives that share of the collateral surplus. That is:

11) liq_SP_gain = 0.1D*s*min(k, (f +q(1-f-p))

Liquidation gas compensation

The underwater debt has CR_A = 1.1, and the new liquidateable fraction q has aggregate collateral ratio CR_q with MCR <= CR_q < CCR. This aggregate ratio depends on the distribution of the Troves corresponding to q. So we parameterize it.

The total gas compensation is given by:

12) liq_gas_compensation = D(1.1f +CR_qq(1-f-p))/200

Attacker Costs

Huge trove borrow cost

13) attacker_borrow_cost = bD/200

And so by 4):

14) attacker_borrow_cost = D(TCR - CCR)/(200(CCR - MCR))

(Here 'TCR' is the initial TCR before trove T is opened).

Redemption cost

The redemption fee formula is:

15) redemption_cost = d_B(1/200 + d_B/2D')

In this scenario the redemption occurs *after* the attacker has opened the huge trove and increased the system debt from D to D'. So we use D' in the redemption denominator.

We have:

16) D’ = D(1+b)

And by 3):

17) D’ = D(1+p)

18) redemption_cost = pD(1/200 + pD/2D(1+p))

19) redemption_cost = pD/200 + p^2D^2/2D(1+p)

20) redemption_cost = pD/200 + p^2D/2(1+p)

Profit equation

21) profit = liq_SP_gain + liq_gas compensation - attacker_borrow_cost - redemption_cost

22) profit = D[0.1smin(k, (f +q(1-f-p)) + (1.1f +CR_qq(1-f-p))/200 - (TCR - CCR - m)/(200(CCR + m - MCR)) - p/200 + p^2/2(1+p)]

Plotting it:
https://www.desmos.com/calculator/bbvrpoltyj

Analysis

The main factor in attack profitability is again q, the fraction of remaining debt that becomes liquidateable in Recovery Mode. For conservative q=10% and underwater debt fraction f=20%, the attack is only profitable up to TCR=160%, i.e. a fairly low TCR.

This assumes the attacker owns the entire SP (s=1) and also assumed a worst-case distribution of Troves in order to get a lower bound on redemption size (and cost). In reality, the redemption cost will be much higher and the profitable range much smaller or non-existent.

Variant: attacker creates underwater debt

Here the attacker loses the coll surplus on the underwater debt (0.1fD) and regains part of it through their SP gain and gas compensation. They also pay the borrow fee on the underwater debt, i,e. fD/200.

Therefore, we have the adjusted profit equation:

profit_attacker_creates_underwater_debt = liq_SP_gain + liq_gas compensation - attacker_borrow_cost - redemption_cost - underwater_coll_surplus - underwater_borrow_fee

That is:

profit_attacker_creates_underwater_debt = D[0.1smin(k, (f +q(1-f-p)) + (1.1f +CR_qq(1-f-p))/200 - (TCR - CCR)/(200(CCR - MCR)) - p/200 + p^2/2(1+b) - 0.1f - f/200]

Plotting it:
https://www.desmos.com/calculator/9kjdfi6uoh

Analysis

For q =10%, profitability is lower with attacker-created underwater debt - and the greater the underwater debt, the lower the profitable range.

Notably it's still profitable for f=0 (i.e. infinitesimal) underwater debt, since we've assumed the TCR has already been dragged to CCR by the huge Trove. And so for the right Trove distribution, a drag-down below CCR can occur for tiny (~zero) f.

With attacker-created underwater debt f and q=10%, the maximum TCR at which this is profitable is ~155%.

Conclusion

Even under worst-case assumptions (attacker owning all of the SP and the ideal distribution of Troves which minimizes their redemption cost) the attack is only profitable at low TCR ranges, regardless of whether the underwater debt pre-exists or is engineered by the attacker.