Liquidation Event Handling And Collateral Reserves

Updated as of block 19340105 at 3/1/2024, 6:43:59 AM ET

- ID: 132
- Proposer: `0xC66e426404C742D81655A9D80Ce58fdbcEE468A9`
- Start Block: 15833192 (10/26/2022, 11:37:59 AM ET)
- End Block: 15852902 (10/29/2022, 5:44:47 AM ET)
- Targets: `0x316f9708bB98af7dA9c68C1C3b5e79039cD336E3` ; `0x1EC63B5883C3481134FD50D5DAebc83Ecd2E8779` ; `0x3d9819210A31b4961b30EF54bE2aeD79B9c9Cd3B`

Forum Post

Forum post is present here: Forum Post

Table of contents

Proposal Text

> # *Liquidation Event Handling And Collateral Reserves*
>
> ## *Explanation*
>
> ### *Emit `Transfer` event from `absorb`*
>
> *When a user is liquidated in Compound III, all of their collateral is absorbed into the protocol, and they are typically left with a positive balance of the base asset (USDC) and no debt (or collateral dust).*
>
> *Shortly after the first market launched, a community developer noticed that the `absorb` function was missing an event log in the case when an account is left with a positive balance.*
>
> *While this doesn't have any economic impact, adding this event log will improve the user experience on Etherscan and blockchain explorers, and make analytics easier.*
>
> ### *Implicit collateral reserves*
>
> *Without this patch, the collateral which is bought using `buyCollateral` must be part of the protocol's balance explicitly, which can happen during `absorb`. Excess collateral simply transferred to the protocol will not be available as collateral reserves to be sold by the protocol automatically.*
>
> *With this patch, all of the excess collateral asset available using the ERC20 `balanceOf` function is implicitly considered part of collateral reserves. This means that accidentally transferring the ERC20 to the protocol will automatically become reserves. It also means that interest accrued implicitly, e.g. when the collateral is the token of another Compound III market, will automatically become part of reserves, which can be sold by the protocol and bought using `buyCollateral`.*

> *This patch also formalizes the idea of collateral reserves in general, adding a* `getCollateralReserves(asset)` *function.*
>
> *The associated forum post for this proposal can be found here.*

Proposal

The proposal itself is to be made from this pull request.

The first step is to deploy a new CometFactory, using the patched version of the contract, which adds the Transfer event to `absorb` *and modifies the total collateral accounting. This is done as a 'prepare' step of the migration script.*

The first action of the proposal is to set the factory for cUSDCv3 to the newly deployed factory.

The second action is to deploy and upgrade to a new implementation of Comet, using the newly configured factory.

The third action is to transfer 10 COMP to ilemi.eth (0x2Ae8c972fB2E6c00ddED8986E2dc672ED190DA06), as a reward for identifying the issue and suggesting the `Transfer` *event improvement.*

Checks

Checks Compound Proposal Details ✅ Passed

Info:

1- 🛑 Set factory of USDC to CometFactory

2- Deploy and upgrade new implementation for USDC via Configurator.

3- 🛑 Grant **10** COMP tokens to 0x2Ae8c972fB2E6c00ddED8986E2dc672ED190DA06.