Compound Verification Update

Sekura Team

August 23, 2018

Disclaimer

New Issues

Issue 1 A failed assertion in `PriceOracle`, method `setPriceInternal`. The call to `calculateSwing` in `setPriceInternal` is expected to never fail. However, given an asset A, for which a pending price set by Chad (the account authorized to set the anchor price) to some value C, we can generate an input of `setPriceInternal` that triggers an assertion failure. The input is to set the `asset` argument to A, and `requestedPriceMantissa` to any positive value equal at least `MAX_UINT256`$/10^{18} + C$, denoted P.

The `calculateSwing` method multiplies the absolute difference of C and P by the scale factor 10^{18}. Since $P \geq$`MAX_UINT256`$/10^{18} + C$, the absolute difference is $D = |C - P| \geq$`MAX_UINT256`$/10^{18}$. When D is multiplied by 10^{18} (in (getExp), called from `divExp` at the end of `calculateSwing`), the result is a value equal at least `MAX_UINT256`, generating an overflow that is propagated from `divExp` to `calculateSwing`, resulting in failure of the assertion immediately after `calculateSwing` returns. This is a counterexample to the comment just before that assertion that the only possible error is if anchorPrice is 0.

An overflow that should be recorded by the error codes manifests as an assertion failure. The possible damage due to Ethereum semantics of assertions is the loss of all gas given to the transaction.

Issue 2 When the price and anchor price of an asset is in the range $[1, 4]$, the price cannot be updated by the poster, because the minimum change of 1 is greater than the maximum allowed swing of 10%. For an anchor $a \in [1, 4]$, the maximal price and the minimal price are equal a. For example, for $a = 4$ the minimal price is

$$\lfloor \frac{4 * (10^{18} - 10^{17}) + 10^{18}/2}{10^{18}} \rfloor = \lfloor \frac{41}{10} \rfloor = 4$$

and the maximal price is

$$\lfloor \frac{4 * (10^{18} + 10^{17}) + 10^{18}/2}{10^{18}} \rfloor = \lfloor \frac{49}{10} \rfloor = 4$$

Therefore, if the anchor price of an asset A is set to a small value a in the range $[1, 4]$, the only way new prices can be set is for Chad to set the anchor price manually until the update values become large enough to work reasonably.

Issue 3 Chad can freeze the protocol by setting an extreme anchor price. When Chad sets a pending anchor, new prices are not capped. Rather, new prices must be within 10% of the pending anchor price set by Chad. Therefore, if Chad sets a pending anchor for an asset A which is very different from the current trend for prices for A, prices will not be updated at all. Chad may also set a very high pending anchor that will trigger overflow errors in `calculateSwing`.