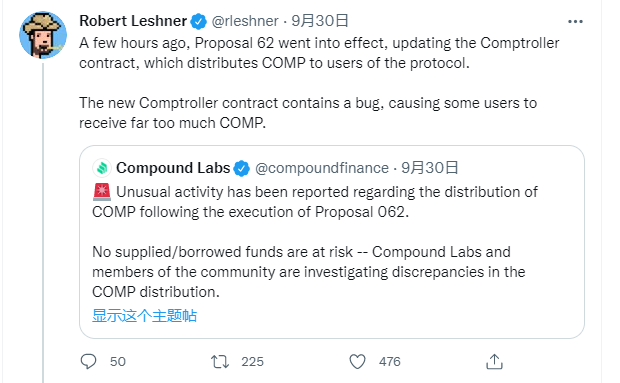
**compound第62提案升级导致奖励分发过多**





Compound创始人表示：该漏洞是由于Compound的62次提案升级所导致。

[Technical]  https://compound.finance/governance/proposals/62

* Comptroller#compSpeeds storage variable is no longer being used and has been effectively deleted. It's replaced by Comptroller#compBorrowSpeeds and Comptroller#compSupplySpeeds. This proposal copies the current rates into the new storage variables.
* Comptroller#\_setCompSpeed(CToken,uint256) function has been removed and replaced by Comptroller#\_setCompSpeeds(CToken[],uint256[],uint256[]). This new function allows COMP rewards rates for multiple markets to be set with a single call.
* An upgrade hook initializes all non-initialized market state indices (initial index is 1e36).
* When new markets are added, their state indices are now properly initialized (used to use lazy initialization which caused many problems).

#### **升级主要做了以下改变：**

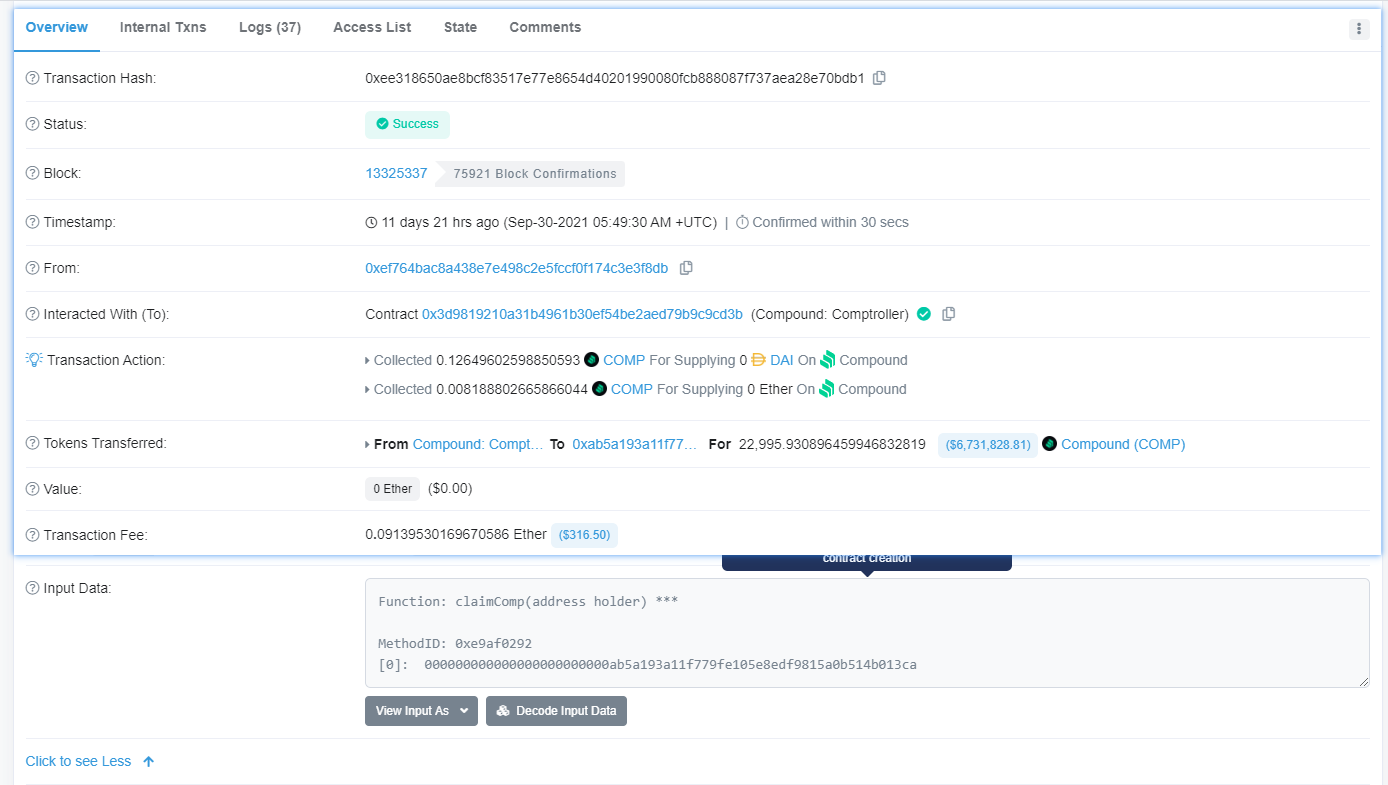
* 1.使用 Comptroller#compBorrowSpeeds 和Comptroller#compSupplySpeeds来代替Comptroller#compSpeedsstorage类型变量；
* 2.Comptroller#\_setCompSpeeds(CToken[],uint256[],uint256[])来对Comptroller#\_setCompSpeed(CToken,uint256) 进行升级；
* 3.***设定所有新增的初始化的市场状态指数（initial index is 1e36）***；

##### 原本设定所有未初始化的市场状态指数是为了解决惰性初始化导致的许多其他问题，来解决之前存在的整数溢出漏洞，但是并没有全面考虑到如此的升级对原本的功能的影响！

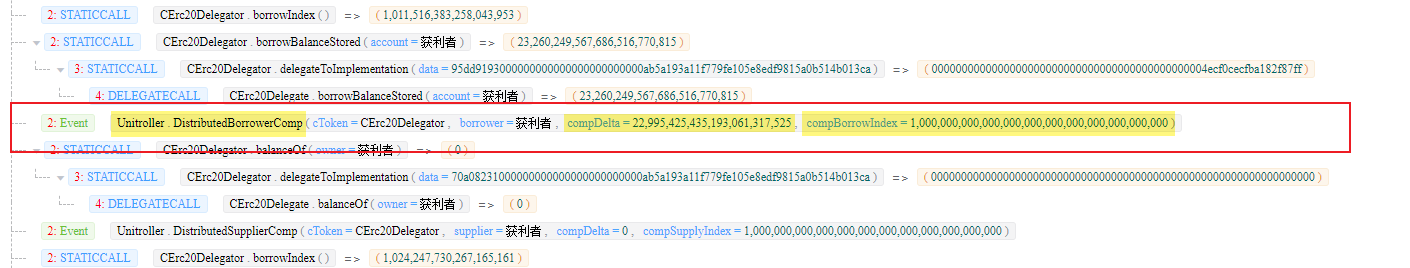
* 

##### 该漏洞产生的条件：需要在62次提案更新之前与某些特定代币（cSAI、cTUSD、cMKR、cSUSHI、cAAVE、cYFI）进行存款或借款，并留有一定的余额，就可以获得大量的comp收益奖励。

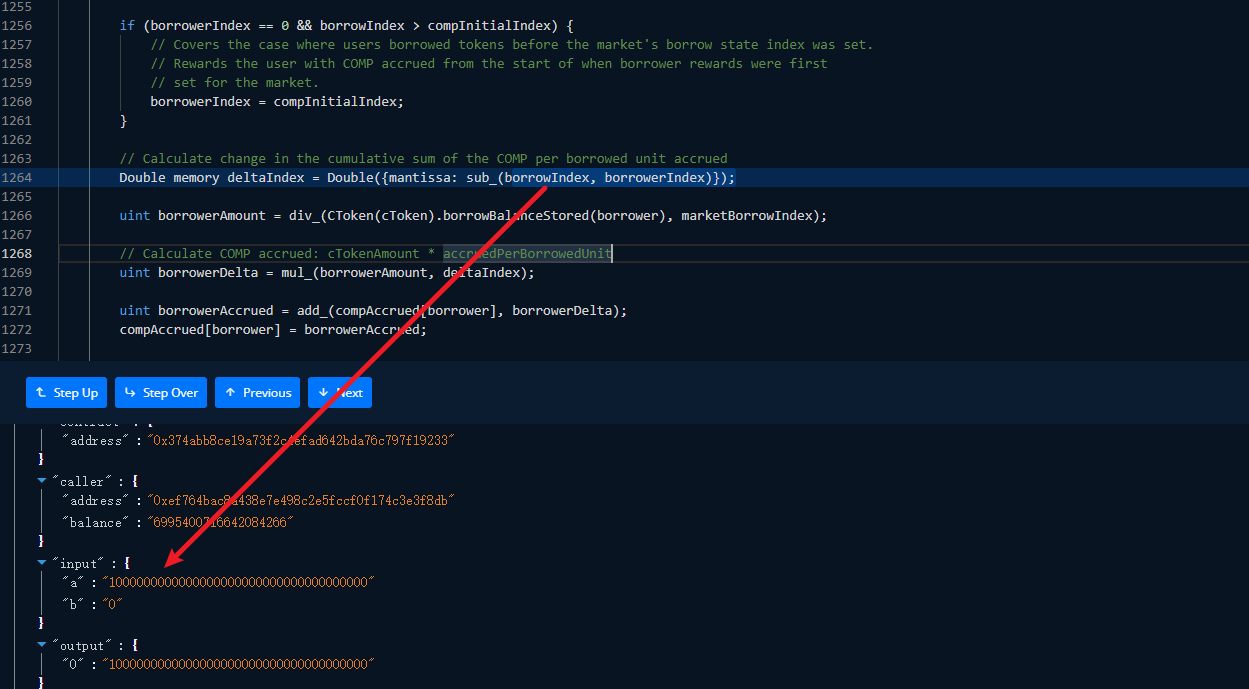
##### 我们以其中一个交易为例进行分析：<https://etherscan.io/tx/0xee318650ae8bcf83517e77e8654d40201990080fcb888087f737aea28e70bdb1>

* 

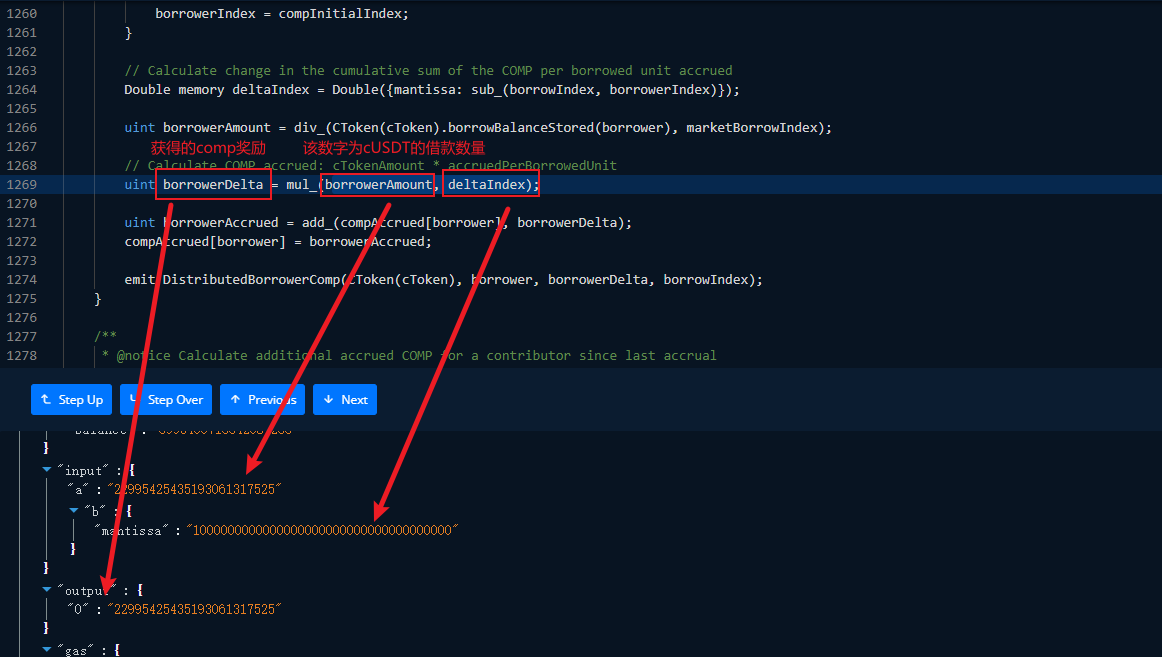
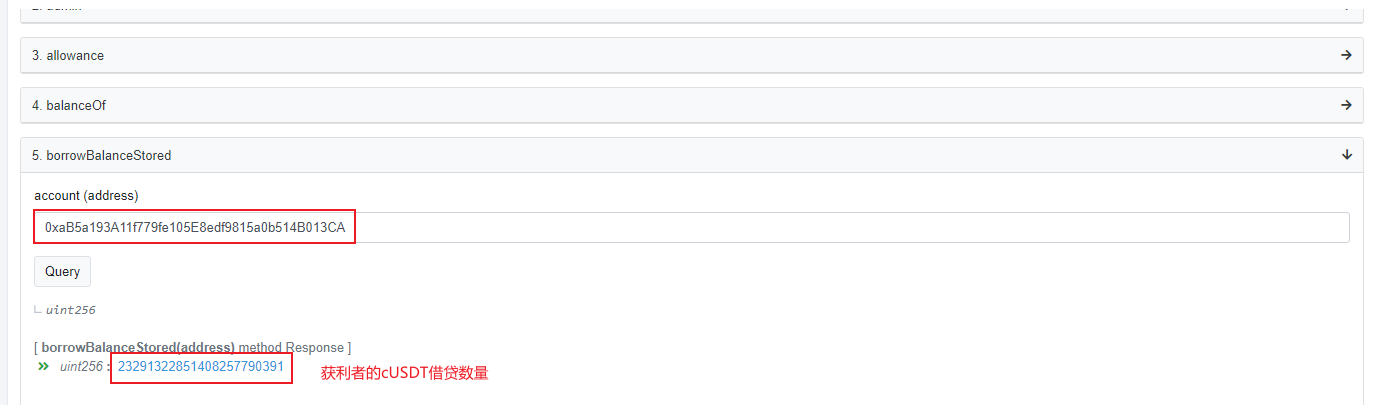
获利者地址为（0xab5a193a11f779fe105e8edf9815a0b514b013ca），通过claimCmop函数来提取收益，但是却获得了22,995.930896459946832819 枚comp的收益，这个数字远远大于应获得的收益。

* 

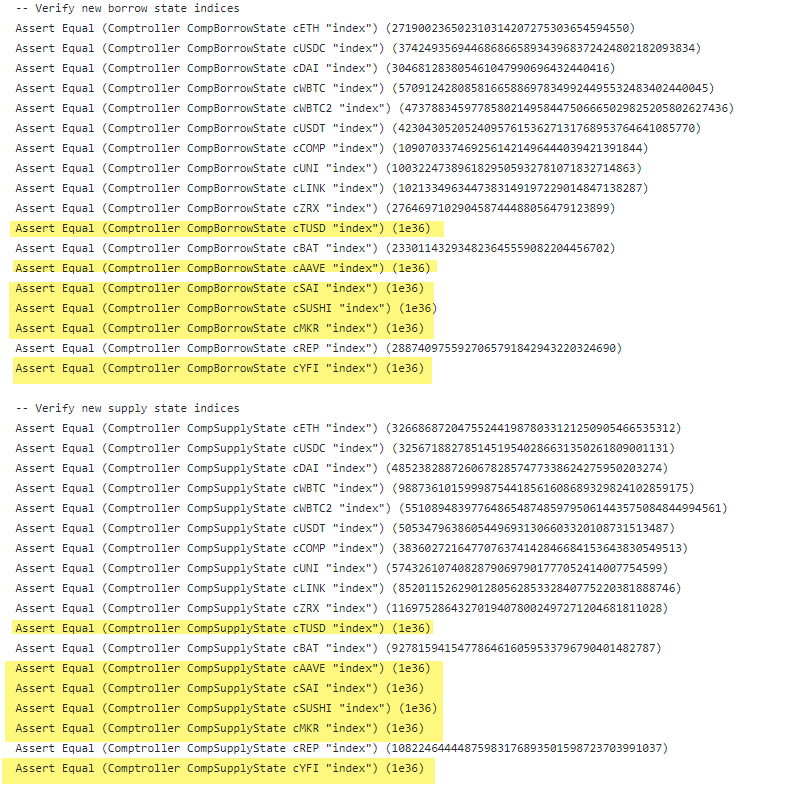
通过交易追踪发现，该获利者在领取借贷代币（cUSDT）的奖励时，获得了大量的comp代币（ 22,995.425435193061317525枚）。

* 

用户借贷comp奖励的公式为：borrowerAmount \* deltaIndex（即：借款数量\*奖励因子）。deltaIndex = borrowIndex - borrowerIndex（奖励因子 = 池借贷指数 - 用户借贷指数），由于第62次提案的升级，cUSDT池借贷指数被初始化为 compInitialIndex （1e36），而此时用户的借贷指数为0，导致算出的奖励因子deltaIndex非常大为1e36。合约正常逻辑应该先通过if (borrowerIndex == 0 && borrowIndex > = compInitialIndex) 该分支的判断，使用户的借贷指数进行更新为compInitialIndex（1e36），才可以进行奖励的计算！此处由于 borrowIndex > compInitialIndex（缺少等号） 导致绕过该判断，直接进行奖励计算，导致奖励因子计算出的值非常的大。

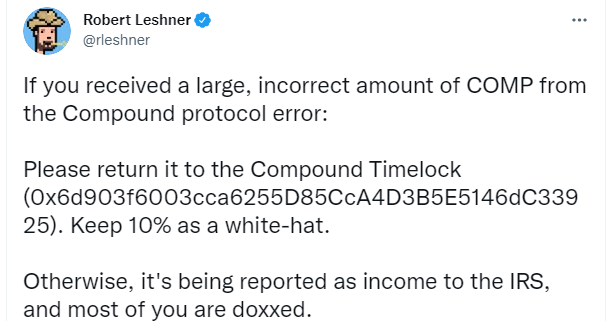
* 
* 

该用户在62次更新之前的一笔cUSDT的借款，为其带来了22,995枚comp的”收益“。因此该漏洞并不是任何人都可以获利，需要在62次提案更新之前与某些特定代币进行存款或借款，并留有一定的余额，就可以获得大量的comp收益奖励。

* 

通过更新提案可知，这些特定代币为cTUSD、cAAVE、cSAI、cSUSHI、cMKR、cYFI这6种代币，其池借贷指数和存款指数均被初始化为1e36！

##### **后续：**

* 

Compound创始人表示拿出多出来收益的10%作为白帽奖励，希望获利的用户可以归还多出来的奖励，并通过提案63，64对问题进行修复。

##### **启示：**

Compound虽然通过DAO的管理模式进行自治，但是在每次进行协议提案升级之前是否进行充分的安全性验证？当问题突发时，需要一定的提案过程与时间才可以对问题进行修复，充分暴露了对突发问题机动能力不足等问题！