計算機安全 HW3 Writeup

- Real name: 張瀚文
- Nickname on course website: Hwww
- Student ID: b07505027.

Bet

解法

這題如題目所示，是要跟 `Bet` 合約賭博，而Bet合約產生隨機數的方法，是由一個預先存好的 `seed` 及當次交易的 `block` 中的隨機來源產生出來的，而其中， `block` 因為「相同交易的 `block` 都是相同的」，所以可以當作已知，只要在呼叫 `Bet.bet()` 的時候在自己的合約裡產生就好了。而 `seed` 則要分析一下 `BetFactory` ，發現 `BetFactory` 在執行 `create()` ，創造 `Bet` 的時候，會將當次交易的 `block.timestamp` 作為 `seed` 傳給 `Bet` ，由此可知，我們可以先在第一次交易，呼叫 `BetFactory.create()` 時，把 `seed` 也記錄下來（因為相同交易的 `block` 都是相同的）詳細步驟如下：

寫一個 `code/HackBet.sol` ，裡面有 `create()` , `run()` , `validate()` 等

第一次交易T1，呼叫 `HackBet.create()` ，紀錄 `target` (Bet的address)及此時 `seed = block.timestamp`

第二次交易T2，呼叫 `HackBet.run()` ，此時利用T1得到的 `seed` 跟T2的 `block` 來產生隨機數（可以完全利用 `Bet` 裡的 `getRandom()` ）

最後一次交易，呼叫 `HackBet.validate()` 就可以拿到了～

然後因為要收錢，所以要寫 `receive()` ， `fallback()` 也寫比較不會出問題的樣子，在試的時候沒寫 `fallback()` 好像怪怪的

另外在debug的時候為了看合約的storage的狀況，寫了 `code/bet.js` 來看合約的狀態。

心得

這題跟 [LAB 0x03] ReEntrancy滿像的，所以前面factory跟target的流程寫起來比較順，後面因為上課也有講說相同交易的block會是一樣的，所以滿快就想到解法了～