solidity面试题(四)

1. 重入攻击

攻击合约实现fallback方法，在fallback内调用被攻击合约的取eth方法，被攻击合约通过call发送eth，会触发攻击合约的fallback方法，fallback又调用被攻击合约的发送eth方法。如此反复，直到被攻击合约eth被耗尽。

解决：用transfer或者send发送eth,先做余额判断 - 修改用户余额 - 发送eth,这样多次重入余额会一直减少，不满条件不会到发送eth流程。

2. 拒绝服务攻击（DOS）

被攻击合约的功能被破坏，拒绝正常服务使用。原理跟重入攻击类似。当被攻击合约通过call给黑客合约发送eth时，触发了fallback,黑客合约的fallback直接就说revert回滚。使得流程无法正常往下走，这就是拒绝服务攻击。例如：权限转让，如果黑客是管理员，转让管理权限要给黑客返还eth，如果黑客合约的fallback里实现revert（false）,eth永远不能返还成功，黑客一直霸占管理员位置。

解决：改成用户自己提取eth

3. 带有tx.origin的网络钓鱼

如果被攻击合约中利用tx.origin校验所有者，然后通过call发送eth。黑客可通过钓鱼让用户调用黑客合约，黑客合约再调用被攻击合约，被攻击合约判断tx.origin是用户地址，call发送eth到msg.sender中，这时eth就发送给黑客合约了。

解决：用msg.sender判断真正用户。

4. 算术溢出

在solidity0.8以前，数字溢出不会抛出错误。例如，黑客利用被攻击合约的增加时间方法，使得数字溢出，让质押时间从一周变成0，提前取出质押token。原理： 2^256最大加1会变成0。如果传入的时间+原有锁仓时间，发生了溢出，结果会变成0，可到期取出质押。

解决：0.8以前用openzepplin的safemath。0.8之后数字溢出会报错自动回滚。

5. 访问私有数据

如果在合约中保存敏感数据，即使是private修饰也会被读取。web3.eth.getStorageAt可以读取合约某个交易时期的存储状态，根据数据类型，推算在哪个存储插槽中。

解决：不要在合约中存储敏感信息，密码之类的。

6. 签名重播

合约中的方法可以多次使用相同的签名来执行一个函数。

解决：获取hash签名时，加入nonce值。