## 1 基础知识

EVM虚拟机在解析合约的字节码时，依赖的是ABI的定义，从而去识别各个字段位于字节码的什么地方。关于ABI，可以阅读这个文档：

https://github.com/ethereum/wiki/wiki/Ethereum-Contract-ABI

**0x01 以太坊短地址**

当调用transfer方法提币时，如果允许用户输入了一个短地址，这里通常是交易所这里没有做处理，比如没有校验用户输入的地址长度是否合法。

如果一个以太坊地址如下，注意到结尾为0：

0x1234567890123456789012345678901234567800

当我们将后面的00省略时，EVM会从下一个参数的高位拿到00来补充，这就会导致一些问题了。

这时，token数量参数其实就会少了1个字节，即token数量左移了一个字节，使得合约多发送很多代币出来。

## 2 构造短地址攻击

（1）首先生成一个ETH的靓号，这个账号末尾为2个0

使用一些跑号工具就可以做到，比如MyLinkToken工具，可以很轻易跑出末尾两个0的。



（2）找一个交易所钱包，该钱包里token数量为256000

（3）往这个钱包发送1000个币

（4）然后再从这个钱包中提出1000个币，当然这时候写地址的时候把最后两个0去掉

如果交易所并没有校验用户填入的以太坊地址，则EVM会把所有函数的参数一起打包，会把amount参数的高位1个字节吃掉。

（5）这三个参数会被传入到msg.data中，然后调用合约的transfer方法，此时，amount由于高位的1个字节被吃掉了，因此amount = amount << 8，即扩大了256倍，这样就把25600个币全部提出来了。

## 3 总结

针对这个漏洞，说实话以太坊有不可推卸的责任，因为EVM并没有严格校验地址的位数，并且还擅自自动补充消失的位数。此外，交易所在提币的时候，需要严格校验用户输入的地址，这样可以尽早在前端就禁止掉恶意的短地址。