# 跳过空字符串文本

## 1. 0.4.12之前的版本

合约地址：（也应该在源码情况下检测）

0x251848c3fc50274f5fda449919f083dc937b48b2 #0.4.11版本

0x2dd2a6b0cbd2586d3a6b975b8c35cebe7eb126e8 #只有字节码

0x7bc51b19abe2cfb15d58f845dad027feab01bfa0 #v0.4.11

0x858c9eaf3ace37d2bedb4a1eb6b8805ffe801bba #v0.4.11

0xa0174f796d3b901adaa16cfbb589330462be0329 #v0.4.11

0xa395480a4a90c7066c8ddb5db83e2718e750641c #v0.4.11

0xaa12936a79848938770bdbc5da0d49fe986678cc #v0.4.11

0xd022969da8a1ace11e2974b3e7ee476c3f9f99c6 #v0.4.11

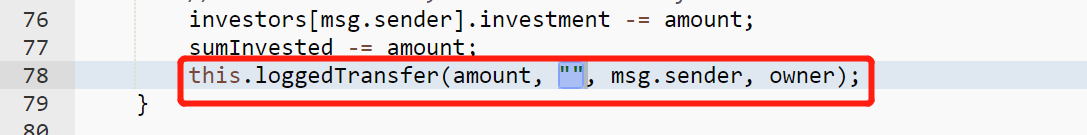
0xe63760e74ffd44ce7abdb7ca2e7fa01b357df460 #v0.4.11

（1）重要特征：

首先，版本必须是<0.4.12版本。

其次，源码情况下的，合约源码的某个参数的调用中存在空字符串（“”）。

再者，msg.sender一定不能是转账对象。



（2）只有字节码的情况下，需要校对函数的参数，对应转账的目的地址不是msg.sender，而是Owner或蜜罐创建者控制的其他的账户地址。

## 2. 我觉得~~不~~是蜜罐 --- msg.sender是转账对象，而msg.sender就必须要是Shark的创建者才能转账。

0xf4ac238121585456dee1096fed287f4d8906d519

这是一个略微复杂的蜜罐，跟“3. 格外的想法”不谋而合。

也不算与HSU的结合， 重点在managers只有一个成员, 就是Shark的创建者。

## 3. 额外的想法：

假设新手黑客有一定的蜜罐知识，我们则可以假装构造了貌似错误的蜜罐，让新手黑客再次调用陷阱。这个陷阱是在使用空字符做参数时，后面的顺序用错了，从合约取款的地址写成了当前msg.sender的地址。（如果真的是一个蜜罐，应该是Owner的地址才对）。新手黑客以为我们写错了，就可能跳入陷阱。

但是，新手黑客忽略了**隐藏状态更新**，尤其是借助modifier修饰符的隐藏状态更新，使得调用的函数无法执行，新手黑客转入的钱无法取出。