# 稻草人合约

## 1.（构造函数）Private\_Bank

合约地址：

0xd116d1349c1382b0b302086a4e4219ae4f8634ff

0xb93430ce38ac4a6bb47fb1fc085ea669353fd89e

0x8c7777c45481dba411450c228cb692ac3d550344

0xbaf51e761510c1a11bf48dd87c0307ac8a8c8a4f

0xbabfe0ae175b847543724c386700065137d30e3b

0xe610af01f92f19679327715b426c35849c47c657

0x7a8721a9d64c74da899424c1b52acbf58ddc9782 # onlyOwner权限函数引入可控

0xb4c05e6e4cdb07c15095300d96a5735046eef999

0x941d225236464a25eb18076df7da6a91d0f95e9e

0xb5e1b1ee15c6fa0e48fce100125569d430f1bd12

0xa5d6accc5695327f65cbf38da29198df53efdcf0

0x463f235748bc7862deaa04d85b4b16ac8fafef39

0x23a91059fdc9579a9fbd0edc5f2ea0bfdb70deb4

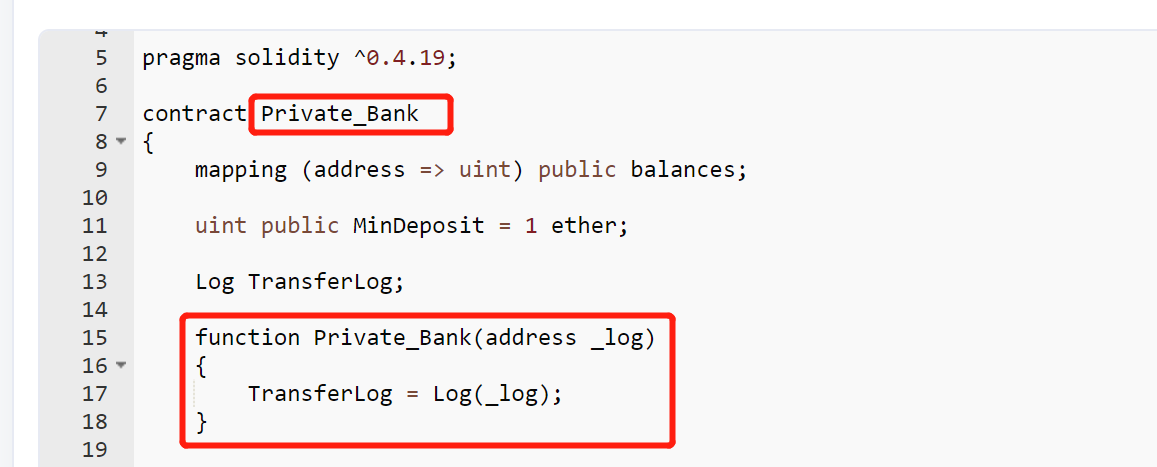
0x95d34980095380851902ccd9a1fb4c813c2cb639

重要特征：（都是分析的第一个地址）

（1）叫“稻草人”不是因为这个合约文件中存在一个相同名称的Log合约。而是合约地址的初始化是由合约创建者（蜜罐制造者）控制的。主合约初始化时，蜜罐制造者可以传入一个与Log类相似的合约地址，也可以传入一个有暗箱操作的合约地址。这里的暗箱操作包括了：a. 调用相同的接口函数时，如果不是攻击者，则抛出异常；b. 调用CashOut转账到由蜜罐制造者给出的其他地址中去。

总之，关键点在于，有一处输入是蜜罐制造者可控的。

蜜罐制造者可以在构造函数中制造可控，也可以通过delegatecall制造可控。



## 2.（delegatecall制造可控）AdminInterface

0xa91a453abde404a303fb118c46e00c8f630216a9

0x7a7d08bcb2faf27414e86ecf9a0351d928054b6b

0x62d5c4a317b93085697cfb1c775be4398df0678c

0xdad02644b70cbb20dec56d25282ddc65bb7805a1

~~0xe4c94d45f7aef7018a5d66f44af780ec6023378e~~ #honey\_badger认为不是蜜罐

0xfa8bb2a68c67e39409cd336d1a8024a2ad9a62ff

0xff5a11c0442028ee2a60d31e6ebb3cbac121ffe5

~~0x60bf91ac87fee5a78c28f7b67701fbcfa79c18ec~~ #honey\_badger认为不是蜜罐

0x23f890c412f3f06784a30ba40ed4832a0fca55f0

0x477d1ee2f953a2f85dbecbcb371c2613809ea452

~~0xc86e7c69a53582537cc52c4dfdeaeae27a84d7c5~~ #honey\_badger认为不是蜜罐

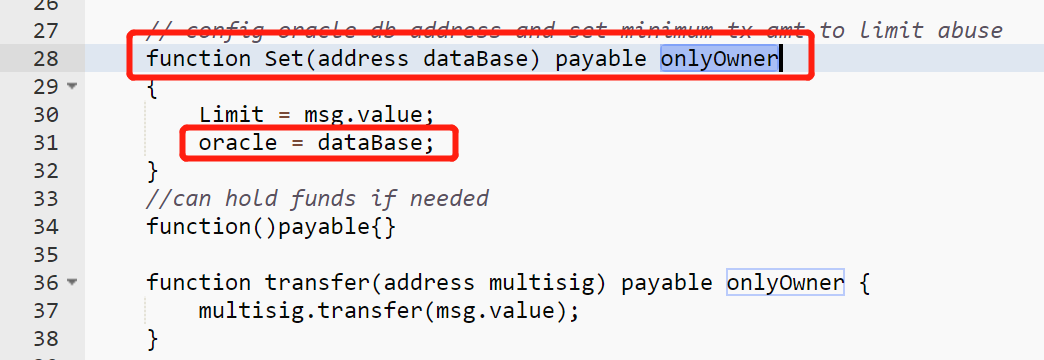
重要特征：（都是分析的第一个地址）

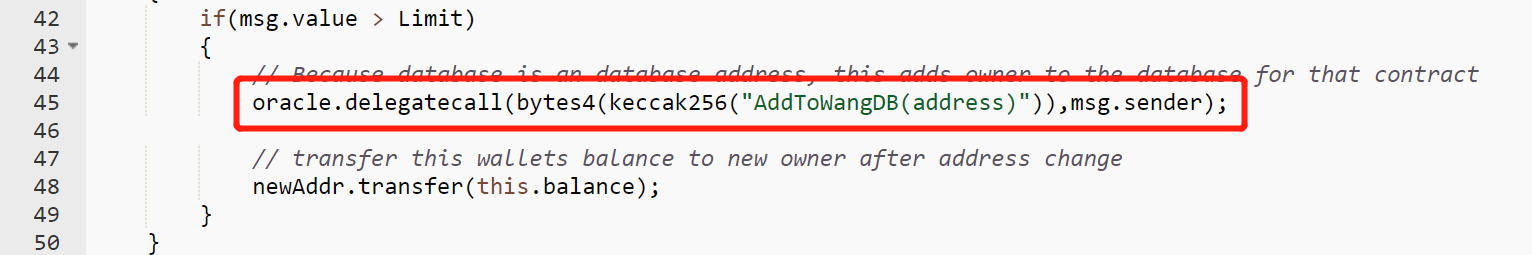
（1）

我理解：当前合约是调用者，oracle是被调用者，但都由蜜罐制造者控制。

当前合约调用oracle中的AddToWangDB(msg.sender)，msg.sender是发起调用addOwner这个函数的发起者的地址，可以是EOA，也可也可以SC. 因为这个合约是个蜜罐，所以一般可以认为发起者是受害者(新手hacker)。

oracle合约是蜜罐创建者控制的合约，因此蜜罐创建者可以通过delegatecall实现对当前合约堆栈内容的修改。





在转移余额之前使用委托代理（delegatecall）。委托代理允许被调用方合约修改调用方合约的堆栈。因此攻击者只需将堆栈中包含的用户地址与自己的地址交换，当从委托代理返回时，余额将转移到攻击者（蜜罐制造者）而不是用户的地址。

PS：在本合约中，尽管onlyOwner中对oracle的设置是一个前提，但delegatecall()的使用与onlyOwner无必然联系，只要合约中存在delegatecall()，则可以默认该合约中由蜜罐制造者可以控制的输入。（因为delegatecall是会执行函数调用。）

## 3.（intitalized – 隐藏状态更新）存在小小的逻辑陷阱，成为蜜罐的前提是要有一个行为，即：合约创建后，首先进行初始化，使得貌似可控的输入却无法被（新手）黑客控制

0xd518db222f37f9109db8e86e2789186c7e340f12

0xbe4041d55db380c5ae9d4a9b9703f1ed4e7e3888

0x01f8c4e3fa3edeb29e514cba738d87ce8c091d3f

0x4320e6f8c05b27ab4707cd1f6d5ce6f3e4b3a5a1

0x561eac93c92360949ab1f1403323e6db345cbf31

0xdd17afae8a3dd1936d1113998900447ab9aa9bc0

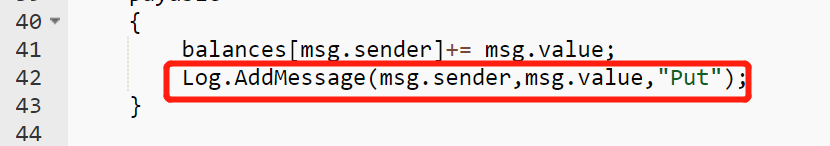
0x4e73b32ed6c35f570686b89848e5f39f20ecc106

0xaae1f51cf3339f18b6d3f3bdc75a5facd744b0b8

0x96edbe868531bd23a6c05e9d0c424ea64fb1b78b

0x85179ac15aa94e3ca32dd1cc04664e9bb0062115





## 4. honey\_badger认为不是蜜罐（？？？没看懂）

0xbf64a825e602a4f1c31480a470e99e1d896c88a7

## 5.？？也没太看懂，感觉像是一个初始化（构造）函数输入可控时的调用，但是好像没有定义让转账给受害者的函数？

0xa6c76471cc89cff4e65cc1fc36613f3c31e4d0d1

0xd754ee6a9e8964602f48e11971e79d0b2f6452d5

0x2f846758e479ee7e0bd87cea5b9f8f3e314c6bd9

0xe35a91f2acceccf1ce6bae792274da6100b639af