CBC字节反转攻击

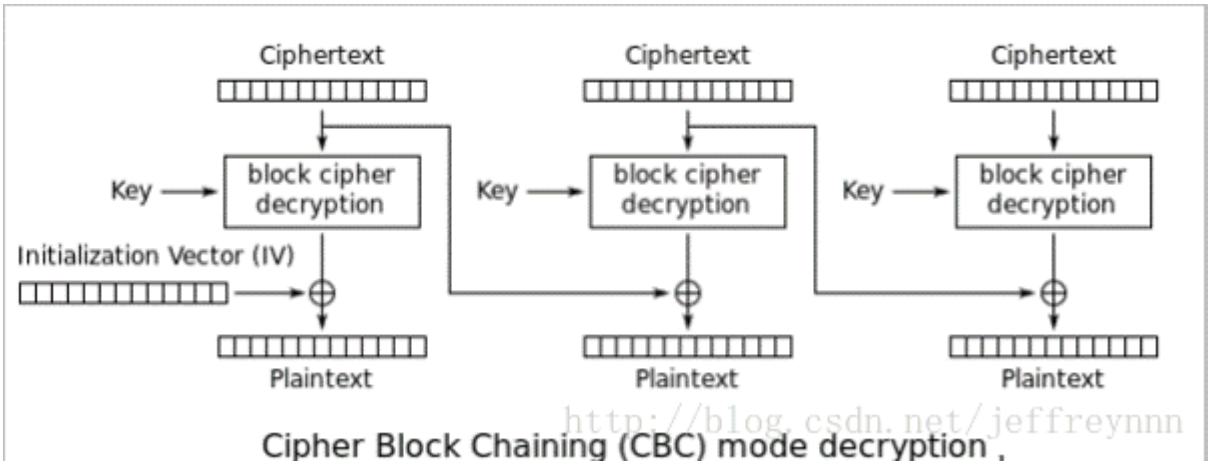
2017年08月11日 17:06:02 阅读数：1951

0x01 题目来源

https://github.com/pbiernat/BlackBox/tree/master/FL!P
下载后可以本地进行做题。

0x02 解题思路

1、首先看到 CBC的解密模式



这里可以看到在CBC解密的时候，是将密文分组为16字节的块，将前一组密文与后一组密文解密后的分组进行异或，从而得到最终的明文。

2、查看源码

发现，在判断用户输入的时候，是不能在字符串中出现“；”的，但是在验证的时候却需要在明文中出现“；”。

```
def mkprofile(self, email):
    if((";" in email)):
        return -1
```

当用于输入的时候，发现服务端会将用户的输入与prefix和suffix进行组合，将这个组合作为最终的输入进行加密。

```
def mkprofile(self, email):
    if((";" in email)):
        return -1
    prefix = "comment1=wowsuch%20CBC;userdata="
    suffix = ";coment2=%20suchsafe%20very%20encryptwowww"
    ptxt = prefix + email + suffix
    return encrypt_cbc(self.key, self.iv, ptxt)
```


CBC加密模式是16个字节进行分组的，可知我们所输入的第一个字符与前面第16字节有间接异或的关系。

comment1=wowsuch
%20CBC;userdata=“
xxxxxx;comment2=…
.....

3、数学背景

有这样一种关系：
如果： **A ^ B = ***
那么： **A ^ B ^ * ^ ; = * ^ * ^ ; = ;**
其中，**A ^ B ^ * ^ ;** 就是我要构造的。

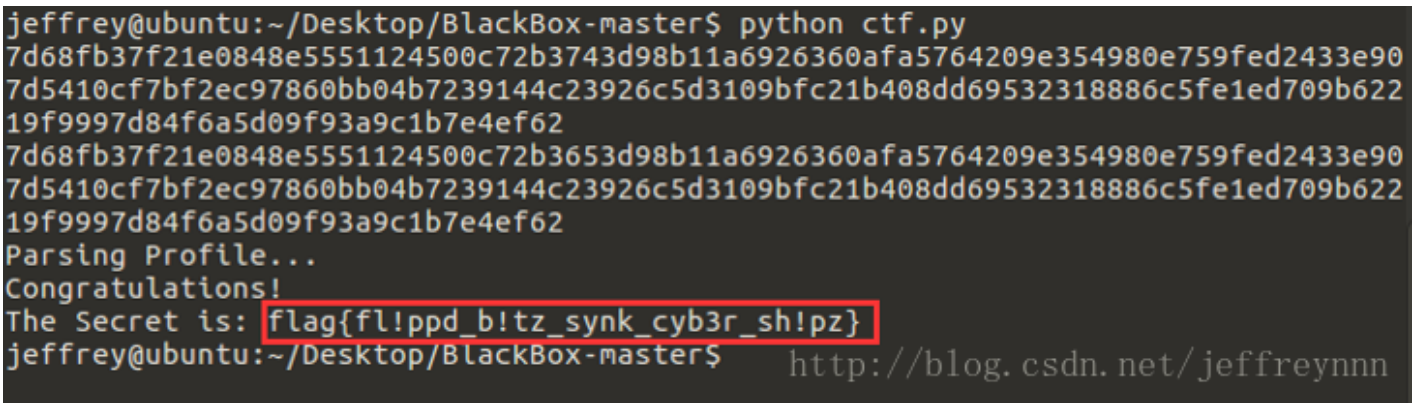
4、客户端代码

使用python编写简单的脚本与服务端进行交互

```
1  import socket
2  import sys
3
4  s = socket.socket()
5
6  host = '127.0.0.1'
7  port = 9002
8
9  s.connect((host,port))
10 s.recv(1024)
11
12 s.send("getapikey:*admin=true")
13 data = s.recv(1024)
14 print data
15 data = data.strip()
16 data = data.decode("hex")
17 data = list(data)
```

```
18  data[16] = chr(ord("*") ^ ord(";") ^ ord(data[16]))
19  data = "".join(data)
20  data = data.encode("hex")
21  print data
22  s.send("getflag:" + data)
23  print s.recv(1024)
```

当服务端运行的时候，运行编写的python脚本，得到flag。



0x03 说明

- chr()函数：参数是0 - 256 的一个整数，返回值是当前整数对应的ascii 字符 。参数可以是10进制也可以是16进制的形式
- ord()函数：参数是一个ascii字符，返回值是对应的十进制整数

0x04 遇到的问题

1、缺少库，除了要求的两个库以外，还需安装seccure，在安装seccure的时候出现错误：

```
1  Building wheels for collected packages: gmpy
2    Running setup.py bdist_wheel for gmpy ... error
3    Complete output from command /usr/bin/python -u -c "import setuptools, to
```

于是在以下的网站找到解决方法：
http://blog.csdn.net/u013687821/article/details/45113131
出错原因是系统没有安装gmp库

2、在编写python脚本的时候，涉及到网络编程，使用了socket模块，与服务端进行通信。
参考链接：
http://www.ifuryst.com/archives/CBC_bitflipping_attacks.html
http://www.cnblogs.com/LanTianYou/p/6890383.html