Hackme-Misc-wp

原创

iReverse 于 2019-09-13 16:56:47 发布 917 收藏 2

分类专栏： Hackme-Misc-wp 文章标签： Hackme-Misc-wp

本文链接：https://blog.csdn.net/qq_41509200/article/details/100803074

Hackme-Misc-wp

背景：

这几天老师要求我们把hackme的题目做出来，于是便开始了新的征程，一定把全部flag找出来

fighting…

题目及解题思路

1、Flag

> All flags are in this format:
> FLAG{This is flag's format}

题目描述中就是flag

All flags are in this format:

```
FLAG{This is flag's format}
```

2、corgi can fly

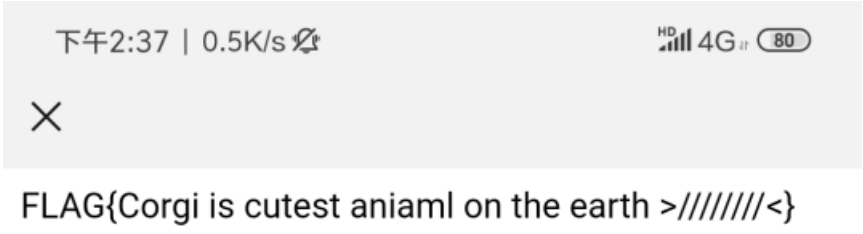
> Corgi is cute, right?
> Pillow (Python) and Bitmap (.NET) are your friends.
> (Maybe you can try stegsolve)

打开链接后是一只小柯基，而看完题目描述后有了思路，题目提示我们用stegsolve做题

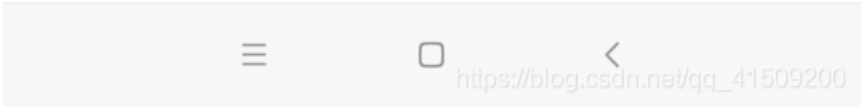
> Corgi is cute, right?
>
> Pillow (Python) and Bitmap (.NET) are your friends.
>
> (Maybe you can try `stegsolve`)

用stegsolve还真的发现了一个二维码

扫完之后出现flag



FLAG{Corgi is cutest aniaml on the earth >////////<}



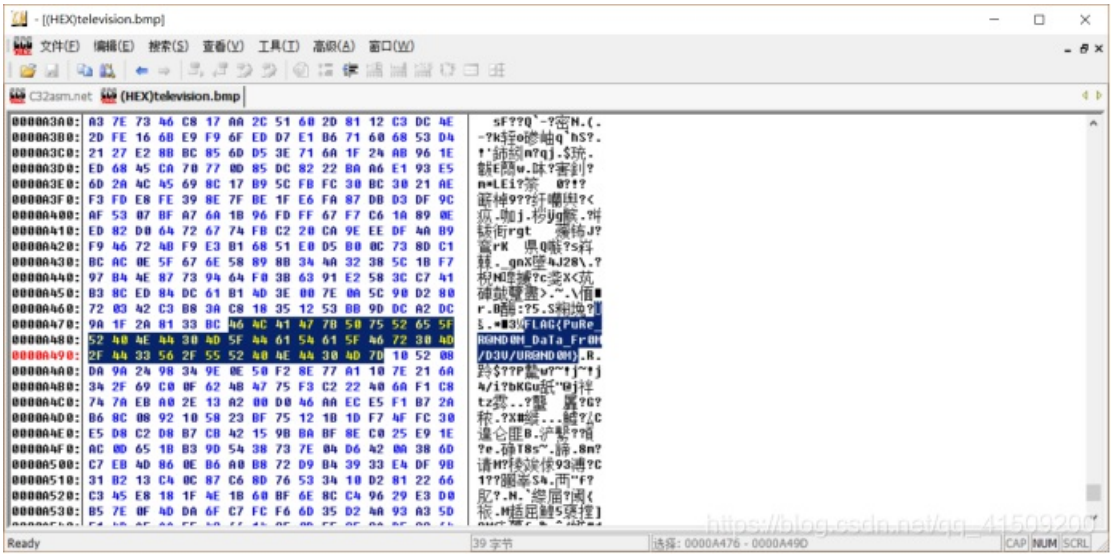
知识扩展

stegsolve的使用

3、Television

> Looks like my television was broken

打开题目链接后是一张bmp图片，估计就是图片隐写的题目，下载下来后用binwalk没发现异样，然后用C32ASM搜索flag时找到了答案



知识扩展

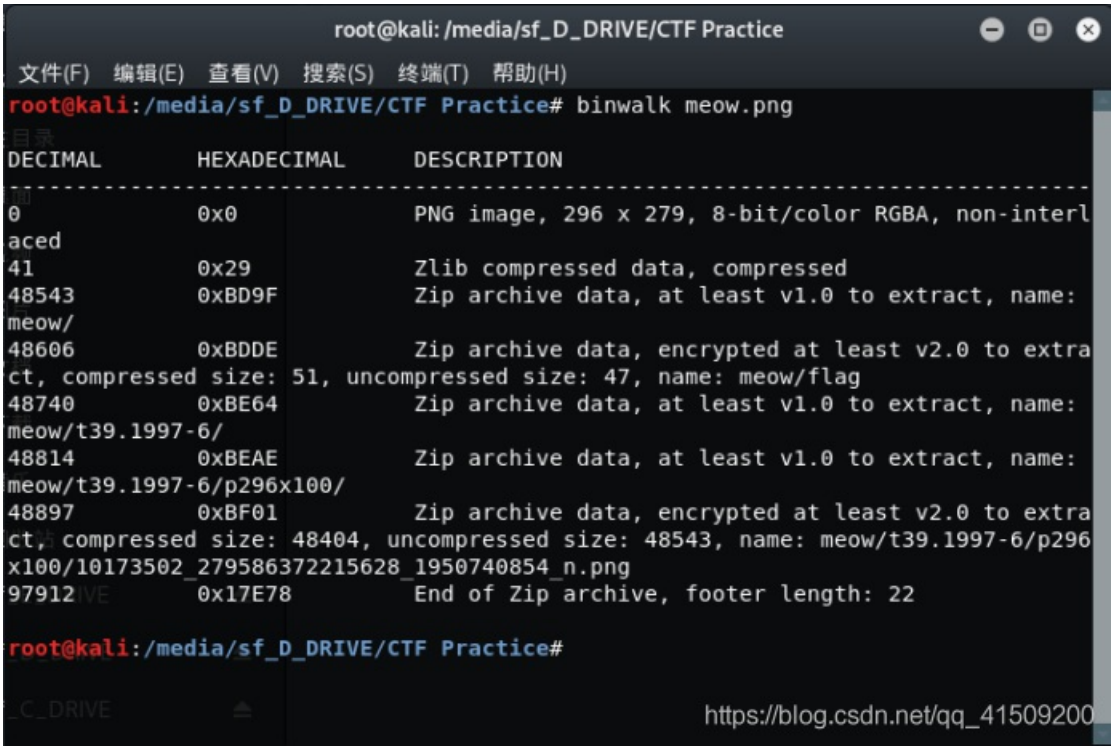
C32ASM的使用

4、meow

> Pusheen is cute!

题目给我们一个png图片，在kali中先binwalk一波，发现有好多文件

```
binwalk meow.png
```



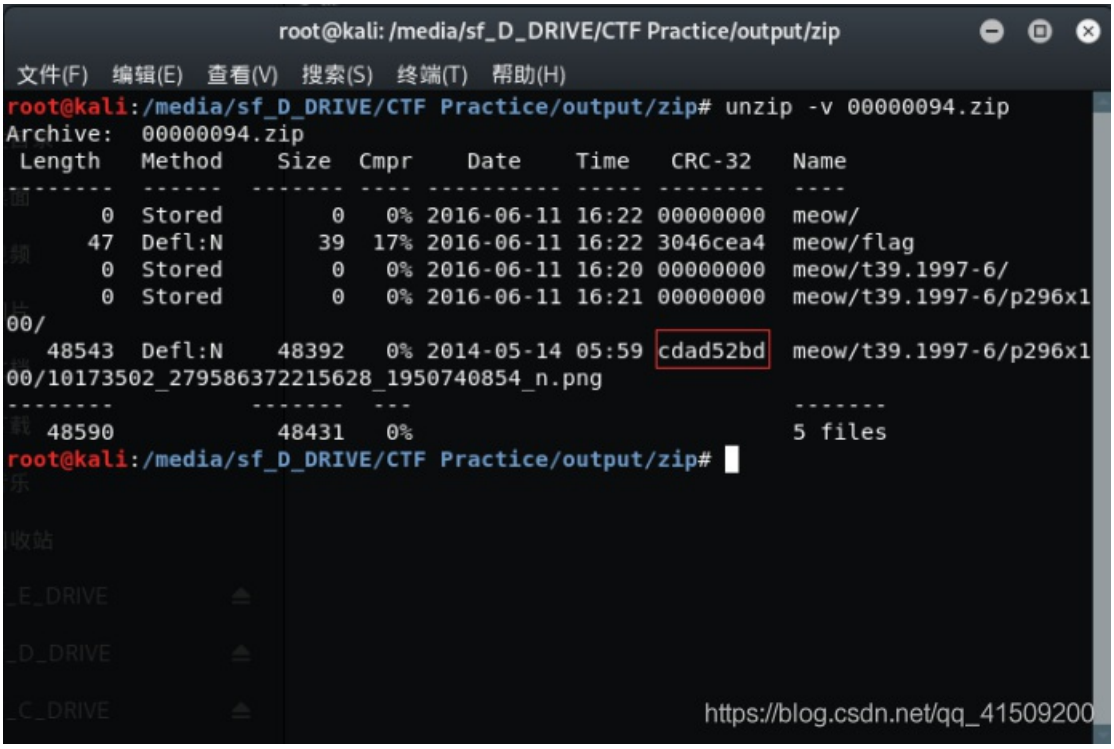
然后，就用foremost命令分离出来

```
foremost meow.png
```

发现只有png文件夹下有一个00000000.png，zip文件夹下有一个00000094.zip，然而这个压缩包需要密码，先分析一下这个zip文件，运行下面命令

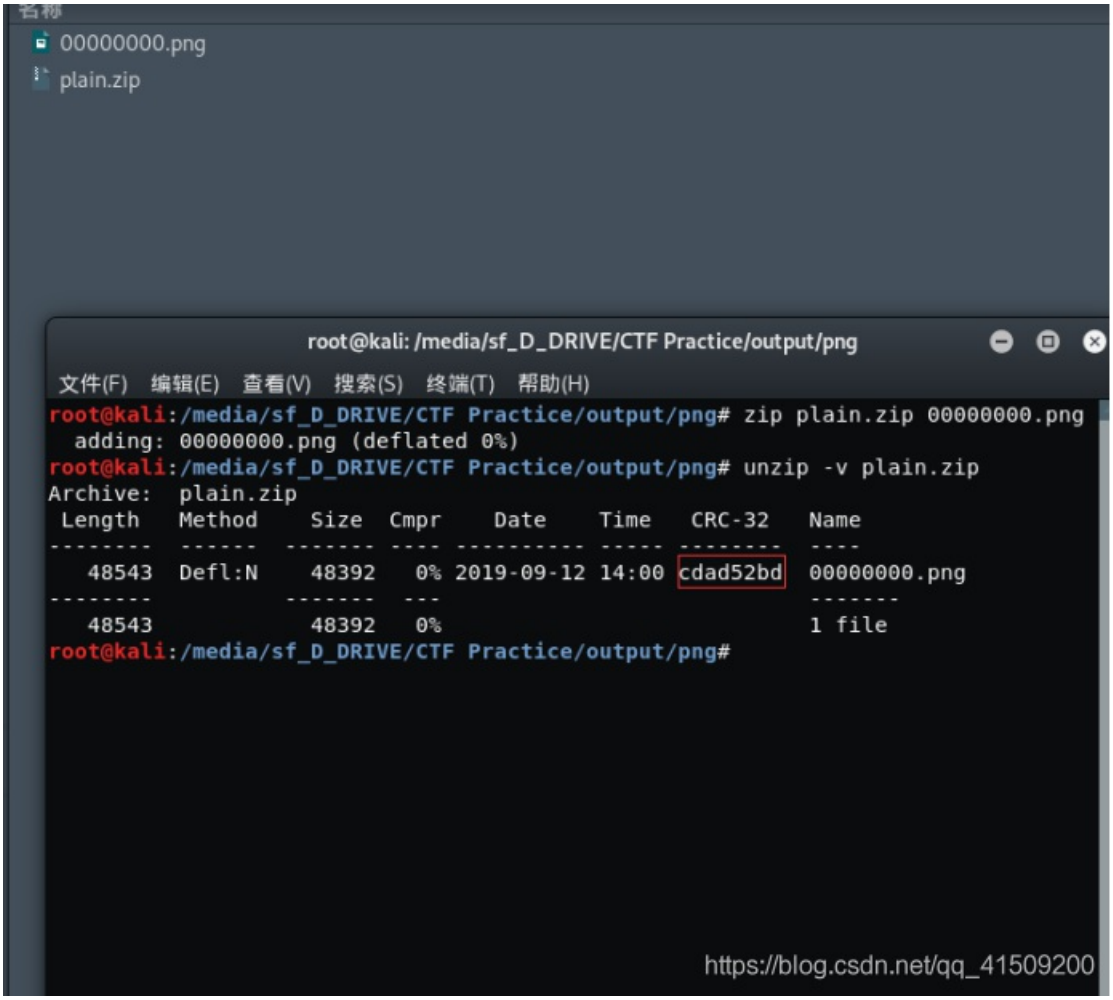
```
unzip -v 00000094.zip
```

发现压缩包中还有个png图片，其CRC32值是cdad52bd



而原来00000000.png的CRC32值也是cdad52bd
这里的00000000.png的CRC32值是先通过运行下面两行命令得到的

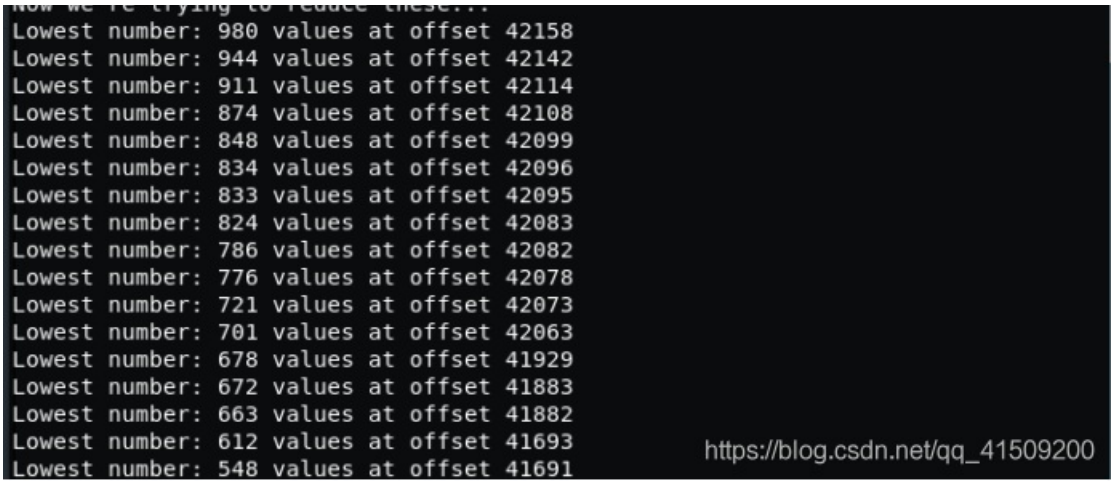
```
zip plain.zip 00000000.png
unzip -v plain.zip
```



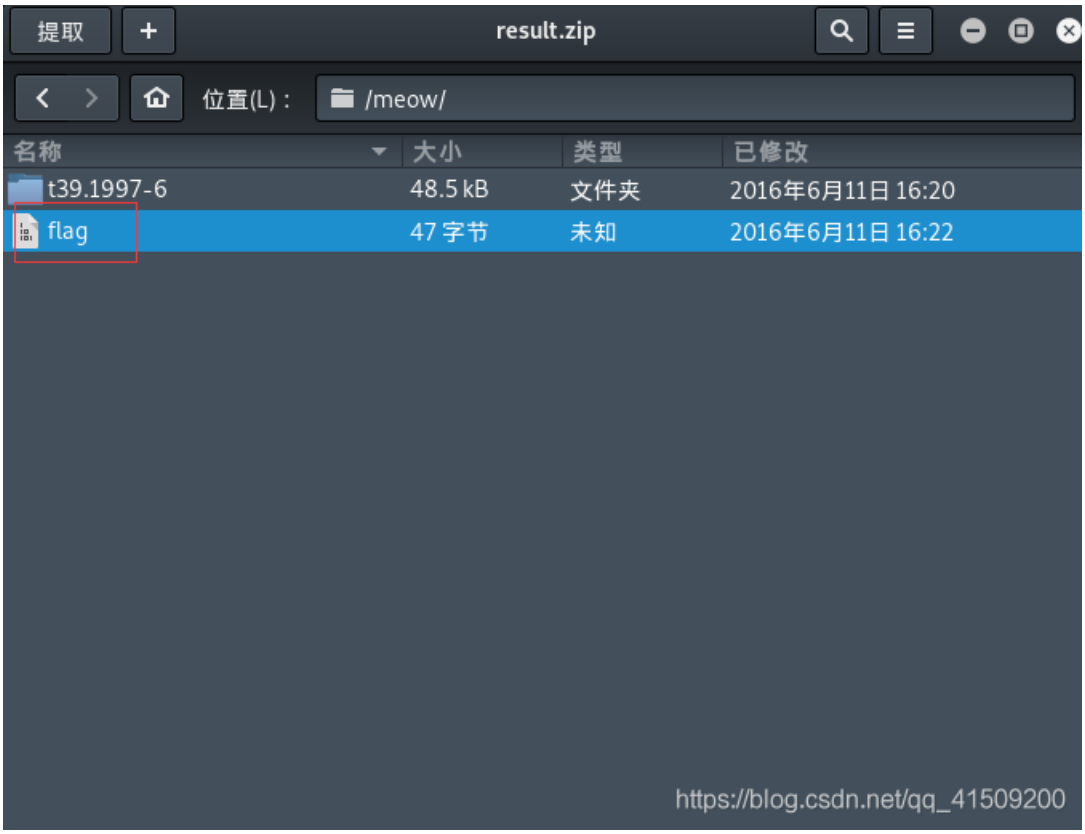
于是这个zip存在明文攻击，于是把压缩00000000.png得到的plain.zip和00000094.zip放到pkcrack的文件夹下，利用命令

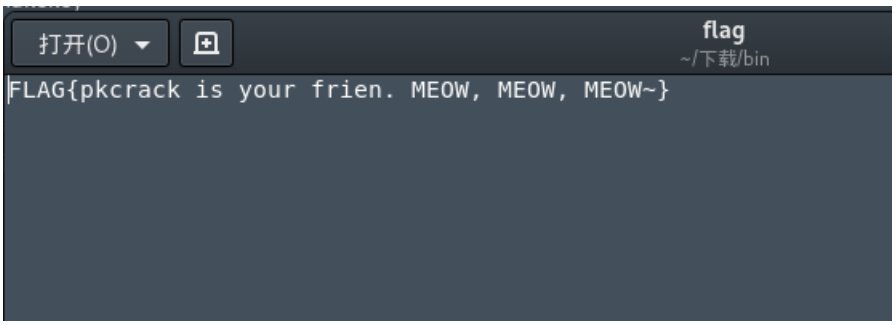
```
./pkcrack -C 00000094.zip -c meow/t39.1997-6/p296x100/10173502_279586372215628_1950740854_n.png -P plain.zip  -p
00000000.png -d result.zip -a
```

得到一个result.zip



result.zip里面就有我们要找到flag





知识扩展：

kali命令的使用binwalk、foremost、zip
举个栗子：

zip plain.zip 00000000.png

上述命令的意思是把00000000.png这个图片压缩成一个名为plain.zip的压缩包 unzip -v plain.zip的意思是分析这个plain.zip压缩包中有什么文件

明文攻击
这是一种较为高效的攻击手段，大致原理是当你不知道一个zip的密码，但是你有zip中的一个已知文件（文件大小要大于12Byte）或者已经通过其他手段知道zip加密文件中的某些内容时，因为同一个zip压缩包里的所有文件都是使用同一个加密密钥来加密的，所以可以用已知文件来找加密密钥，利用密钥来解锁其他加密文件，此时我们可以尝试用ARCHPR或者pkcrack进行明文攻击~

pkcrack的安装

```
#!/bin/bash -ex
wget https://www.unix-ag.uni-kl.de/~conrad/krypto/pkcrack/pkcrack-1.2.2.tar.gz
tar xzf pkcrack-1.2.2.tar.gz
cd pkcrack-1.2.2/src
make
mkdir -p ../../bin
cp extract findkey makekey pkcrack zipdecrypt ../../bin
cd ../../
```

上面为安装pkcrack的shell脚本，我们先运行上面的脚本后

```
chmod u+x install.sh
```

再运行上述命令给予其适当的权限即可使用pkcrack

pkcrack的使用

-C:要破解的目标文件(含路径)
-c:破解文件中的明文文件的名字(其路径不包括系统路径,从zip文件一层开始)
-P:压缩后的明文文件
-p:压缩的明文文件中明文文件的名字(也就是readme.txt在readme.zip中的位置)

5、where is flag

> Do you know regular expression?

打开题目链接后会下载一个flag.xz的压缩包，在kali中解压得到一个名为flag的文件，估计flag就藏在里面，于是运行下面的命令得到了好多flag

```
cat flag|grep -oP "FLAG{[^{}]+?}"
```

个人比较笨，将这些flag一个一个试，也算是找到了flag
参考了一下大佬们的博客，都是说用下面的命令也可以得到flag，命令的大概意思是匹配出所有FLAG{}的内容，其中花括号中的内容可以是a-z、A-Z和0-9的任何字符

```
cat flag | grep -oP FLAG{[a-zA-Z0-9]*}
```

知识扩展：

- 正则表达式
  正则表达式使用单个字符串来描述、匹配一系列符合某个句法规则的字符串。在很多文本编辑器里，正则表达式通常被用来检索、替换那些符合某个模式的文本。
  具体语法参见下面这位大佬的博客
  http://idc.wanyunshuju.com/cym/30.html

6、encoder

> Can you decode this?

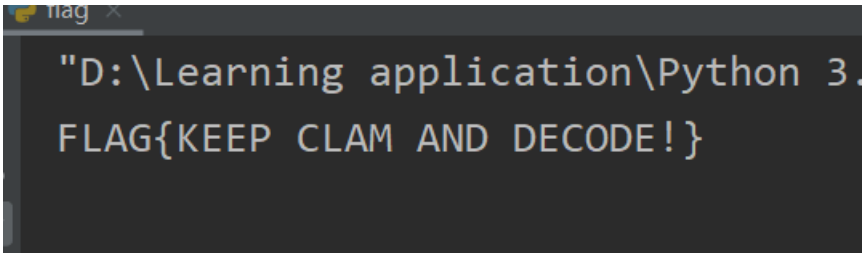
打开题目链接后又是一个压缩包，在kali中解压得到encoder文件夹，里面包含着encoder.py和flag.enc文件，用pycharm打开encoder.py后，发现是好几个加密的函数

```
#!/usr/bin/env python2

import random
import string

def rot13(s):
    return s.translate(string.maketrans(string.uppercase + string.lowercase,
        string.uppercase[13:] + string.uppercase[:13] +
        string.lowercase[13:] + string.lowercase[:13]))

def base64(s):
    return ''.join(s.encode('base64').split())

def hex(s):
    return s.encode('hex')

def upsidedown(s):
    return s.translate(string.maketrans(string.uppercase + string.lowercase,
        string.lowercase + string.uppercase))

flag = 'FLAG{.....................}'  # try to recover flag

E = (rot13, base64, hex, upsidedown)

for i in range(random.randint(30, 50)):
    print i
    c = random.randint(0, len(E) - 1)
    flag = '%d%s' % (c, E[c](flag))

open('flag.enc', 'w').write(flag)
```

根据上述代码，可将其改为解密的函数

```
from binascii import unhexlify
from base64 import b64decode
import string

def unrot13(s):
     return s.translate(str.maketrans(string.ascii_uppercase[13:] + string.ascii_uppercase[:13] +
                                      string.ascii_lowercase[13:] + string.ascii_lowercase[:13],
                                      string.ascii_uppercase + string.ascii_lowercase))

def unupsidedown(s):
     return s.translate(str.maketrans(string.ascii_lowercase + string.ascii_uppercase,
                                      string.ascii_uppercase + string.ascii_lowercase))

def unbase64(s):
    return b64decode(s).decode()

def unhex(s):
    return unhexlify(s).decode()

with open('D:/CTF Practice/encoder/flag.enc') as f:
    data = f.read()

E = (unrot13, unbase64, unhex, unupsidedown)

for i in range(50):
    c, data = int(data[0]), data[1:]
    data = E[c](data)
    if data.startswith('FLAG'):
        print(data)
        break
```

运行之后即可得到flag



7、slow

> nc hackme.inndy.tw 7708
> OMG, It's slow.

这道题到现在都不懂，先把别人的wp拿过来，日后在好好研究吧，朋友们说是时序攻击的题目，需要跑特别半个多小时，我在自己电脑上连这些代码都不能完全调试正确，的确是太菜了，所以这一题就不给大家提供flag了
代码如下

```
#!/usr/bin/env python
from pwn import *
import time
#context.log_level = 'debug'
string = 'ABCDEFGHIIJKLMNOPQRSTUVWXYZ0123456789_'
flag1 = ['F','L','A','G','{']
flag2 = '}'
start_delay = 6

while True:
    for i in string:
        p = remote('hackme.inndy.tw', 7708)
        temp = p.recv()
        #print temp
        a = time.time()
        payload = ''.join(flag1)+i+flag2
        print i
        p.sendline(payload)
        s = p.recvline()
        #print s
        b = time.time()
        print "[+] Delay:", int(b-a), 's'
        p.close()
        if (start_delay) != len(flag1)+1:
            #p.close()
            print 'ooooooooooooo_has_some_bug_repeating...'
            time.sleep(0.8)
            p = remote('hackme.inndy.tw', 7708)
            temp = p.recv()
            a = time.time()
            p.sendline(payload)
            s = p.recvline()
            b = time.time()
            print "[+] Delay:", int(b-a), 's'
            p.close()

        if (start_delay+1) == int(b-a):
            print payload
            flag1.append(i)
            start_delay += 1
            break
        time.sleep(0.2)
    if start_delay == 30:
        break
```

知识扩展：

时序攻击：

> 在密码学中,时序攻击是一种侧信道攻击,攻击者试图通过分析加密算法的时间执行来推导出密码。每一个逻辑运算在计算机需要时间来执行,根据输入不同,精确测量执行时间,根据执行时间反推出密码。

8、pusheen.txt

> Do you think pusheen is cute?

题目txt文件中只有两种格式

```
grep -oP "(▄▀▄▀▄▀▄)|(▄▀▄ ▀▄▀ ▀▄▀ )" pusheen.txt|xargs echo -n|sed 's/▄▀▄▀▄▀▄/1/g;s/▄▀▄ ▀▄▀ ▀▄▀/0/g;s/ //g'|rax
2 -bt
```

于是将其替换成1和0，然后转换成ASCII，出现了flag

本文在参考下面各位大佬的博客的得到的部分wp，谢谢各位朋友

https://www.aloxaf.com/2018/07/hackme_inndy/
https://sp4rta.github.io/2018/12/01/inndy-writeup/
https://sp4rta.github.io/2018/12/01/inndy-writeup/