

# NSCTF-misc-writeup

原创

Brucetg 于 2017-07-22 00:28:54 发布 4110 收藏 3

版权声明：本文为博主原创文章，遵循 [CC 4.0 BY-SA](#) 版权协议，转载请附上原文出处链接和本声明。

本文链接：<https://blog.csdn.net/wanzt123/article/details/75676061>

版权

Misc:

## Kungfu

打开压缩包，里面有一张png后缀的图片，因为分辨率比较低，感觉是套路题，直接winhex打开查看，在文件末尾发现提示：

```
00022220 3F C9 D8 DB 0E 82 F1 F7 6A 0E 76 F9 0A 4D 1D 5D _E00h,ii-] vu m j
00022230 73 FD 1E B1 BC FF 96 77 CC 1C B6 56 D6 7C 5B C0 sý i4y-wi qv0|[À
00022240 FF C9 A0 3F 9F CA FF EB 1F FF 6B FC 6B FC 6B FC yË ?ÿËÿe ÿkükükü
00022250 6B FC 6B FC FF 30 FE 6F 36 5D 85 F4 B0 04 B8 6A küküÿ0p06]...ô°.j
00022260 00 00 00 00 49 45 4E 44 AE 42 60 82 6B 65 79 20 IEND0B`,key
00022270 69 73 20 56 46 39 35 63 30 73 35 58 7A 56 79 61 is VF95c0s5XzVya
00022280 47 74 66 58 33 56 47 54 58 52 39 4D 30 56 73 65 GtfX3VGTXR9M0Vse
00022290 32 35 31 51 45 55 67 20 20 251QEUg
```

<http://blog.csdn.net/wanzt123>

尝试base64解码：

```
T/3kK9_5rkd_uP[+]3E1{m0E
>>>
http://blog.csdn.net/wanzt123
```

尝试凯撒和栅栏解密都不行，还以为是思路错误，后来队友提示后面加一个空格试试，顺利得到flag：

```
'h3_kEy_Is_{Kun9Fu_M0tEr} '
```

你知道吗？这是什么

文件名为zip，binwalk跑一下，发现是zip文件，加上文件后缀.zip,解压后得到hidden.png,

再次binwalk跑一下：

```
root@kali:~/桌面# binwalk hidden.png
DECIMAL 141 20 HEXADECIMAL 0x0 DESCRIPTION
-----
019:38:52] 20 0x0 73KB PNG image, 351 x 560, 8-bit/color RGBA, non-interlaced
019:38:52] 200 73KB
41 0x29 Zlib compressed data, default compression
1975 Completed 0x7B7 Zip archive data, at least v1.0 to extract, compressed size: 291742, uncompressed size: 291742, name: hidden2.jpg
293851 0x47BDB End of Zip archive
```

发现图片后面附加了一个压缩包，使用binwalk -e hidden.png提取可得到hidden2.jpg，flag就在图片上：



So easy!!

压缩包里面有lsb.bmp，stegsolve打开调一下即可得到flag：

flag{LsB\_iS\_v3rY\_SimPl3}

<http://blog.csdn.net/wanzt123>

流量啊

下载下来发现是一个流量包，wireshark打开，导出http对象，得到一个rar格式的压缩文件，不过需要解压密码，尝试爆破，无果

序号	源地址	目标地址	内容类型	大小	备注
31	keyisnsf0cus381.nsfocus.com	text/html	616 bytes	web4.php	
83	keyisnsf0cus381.nsfocus.com	multipart/form-data	1896 bytes	web4.php	
87	keyisnsf0cus381.nsfocus.com	text/html	646 bytes	web4.php	

<http://blog.csdn.net/wanzt123>

strings 一下发现Host字段有点奇怪，像是密码提示：

```
ST:urn:dial-multiscreen-org:service:dial:1
USER-AGENT: Google Chrome/58.0.3029.110 Windows
aNOP
aNPP
3e~RB
POST /web4.php HTTP/1.1
Host: keyisnsf0cus381.nsfocus.com
Connection: keep-alive
Content-Length: 1896
Cache-Control: max-age=0
Origin: http://keyisnsf0cus381.nsfocus.com
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (
Chrome/58.0.3029.110 Safari/537.36
Content-Type: multipart/form-data; boundary=WebKitFormBou
nt
```

于是尝试用nsf0cus381作为压缩包的解压密码，成功，得到flag:

flag{dn5\_res0lv3r\_n\_wireshark}

<http://blog.csdn.net/wanzt123>

**Havefun !!**

实验吧的原题，可参考<http://blog.csdn.net/yalecaltech/article/details/64158016>求解

**N0thing**

下载下来是个流量包，导出http对象，得到以下文件：

~img27	2017/7/21 13:31	文件	1 KB
~img42	2017/7/21 13:31	文件	1 KB
~style.css	2017/7/21 13:31	层叠样式表文档	2 KB
~style.menu.css	2017/7/21 13:31	层叠样式表文档	3 KB
1.html	2017/7/21 13:31	360 se HTML Do...	5 KB
Birthday.zip	2017/7/21 13:31	好压 ZIP 压缩文件	155 KB

<http://blog.csdn.net/wanzt123>

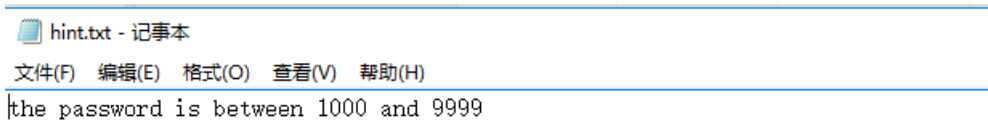
其中只有Birthday.zip文件有用，不过需要解压密码，根据文件名猜测密码应该是生日密码，爆破得到解压密码为：19970818

解压得到Birthday.jpg，strings一下即可得到flag:

```
root@kali:~/桌面# strings Birthday.jpg
JFIF
Adobe
zExif
flag{H4PPY_81RTHD4Y}
CTF{H4PPY_81RTHD4Y}
http://ns.adobe.com/xap/1.0/
<?xpacket begin=
' id='W5M0MpCehiHzreSzNTczkc9d'?>
<x:xmpmeta xmlns:x="adobe:ns:meta/"><rdf:RDF xmlns:rdf
2/22-rdf-syntax-ns#"><rdf:Description rdf:about="uuid:
3d75182f1b" xmlns:dc="http://purl.org/dc/elements/1.1/
rdf:about="http://www.flickr.com/photos/wanzt123/12345678901234567890/">
```

### 白房子

下载后得到压缩包，解压后得到whitehouse.jpg，binwalk一下发现jpg文件后有个rar，提取后解压得到hint:

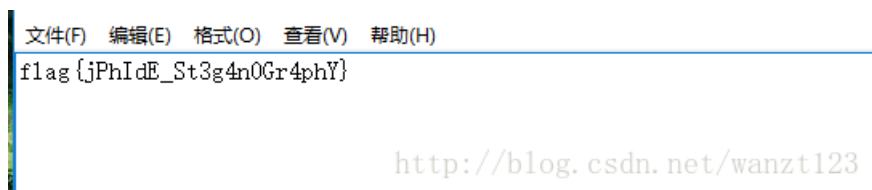
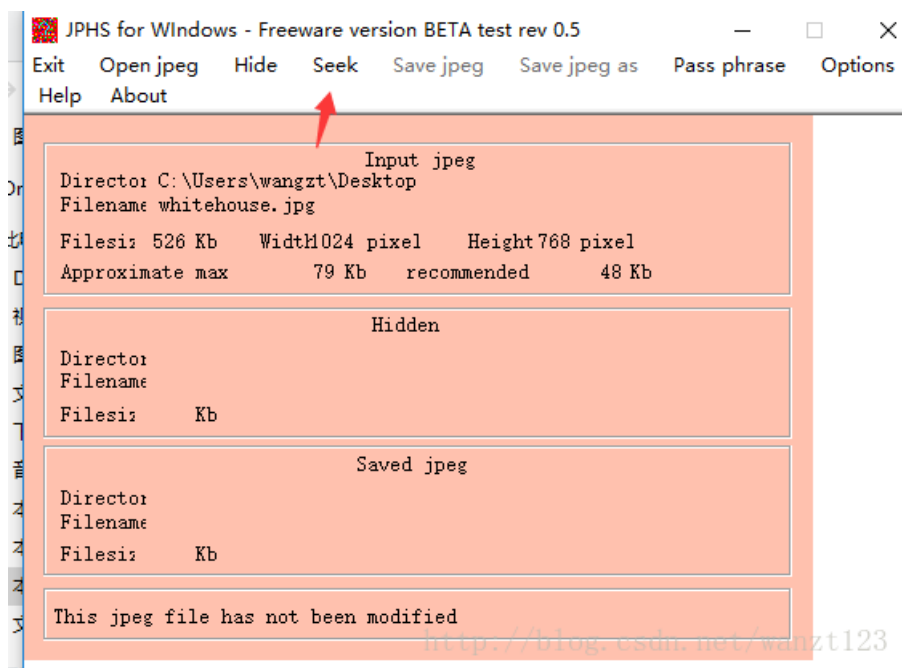


<http://blog.csdn.net/wanzt123>

提示密码在1000-9999之间，用stegbreak爆破可得到密码为2324

```
Corrupt JPEG data: 112 extraneous bytes before marker 0xd9
Loaded 1 files...
whitehouse.jpg : jphide[v5](2324)
Processed 1 files, found 1 embeddings.
Time: 0 seconds: Cracks: 1214, Inf c/s
```

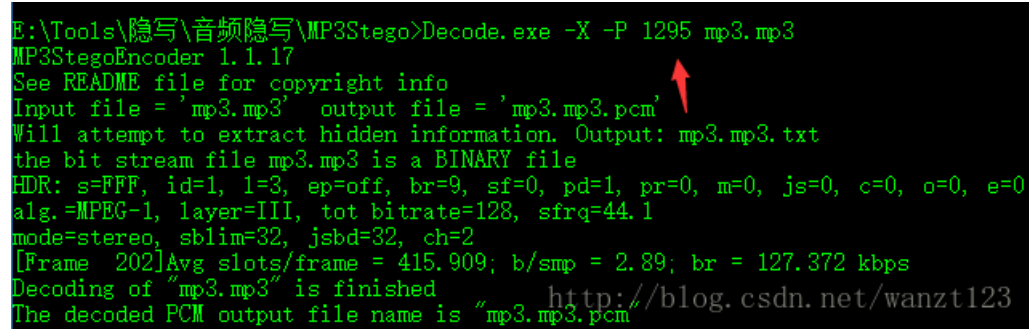
然后使用JPHS，点击Seek，输入密码后选择文件提取路径即可得到flag:



**MP3is so good**



根据hint.txt中的数据生成二维码，扫描得到提示，密码在1000-1300之间，队友写了个批处理来爆破密码，爆破完之后，查找正确的密码，因为是从上往下找的，找到的最后一个能decode音频文件的密码是1295



但是最终提取出来的是乱码，这个问题困扰了差不多半个小时，后来往上面翻了下了，还有好几个能decode mp3文件的密码，第一个能decode的密码是1067

```
MP3StegoEncoder 1.1.17
See README file for copyright info
Input file = 'mp3.mp3' output file = 'mp3.mp3.pcm'
Will attempt to extract hidden information. Output: mp3.mp3.txt
the bit stream file mp3.mp3 is a BINARY file
HDR: s=FFF, id=1, l=3, ep=off, br=9, sf=0, pd=1, pr=0, m=0, js=0, c=0, o=0, e=0
alg.=MPEG-1, layer=III, tot bitrate=128, sfrq=44.1
mode=stereo, sblim=32, jsbd=32, ch=2
[Frame 202]Avg slots/frame = 415.909; b/smp = 2.89; br = 127.372 kbps
[ERROR]Encrypt: unexpected end of cipher message.

E:\Tools\隐写\音频隐写\MP3Stego>Decode.exe -X -P 1067 mp3.mp3
MP3StegoEncoder 1.1.17
See README file for copyright info
Input file = 'mp3.mp3' output file = 'mp3.mp3.pcm'
Will attempt to extract hidden information. Output: mp3.mp3.txt
the bit stream file mp3.mp3 is a BINARY file
HDR: s=FFF, id=1, l=3, ep=off, br=9, sf=0, pd=1, pr=0, m=0, js=0, c=0, o=0, e=0
alg.=MPEG-1, layer=III, tot bitrate=128, sfrq=44.1
mode=stereo, sblim=32, jsbd=32, ch=2
[Frame 202]Avg slots/frame = 415.909; b/smp = 2.89; br = 127.372 kbps
Decoding of "mp3.mp3" is finished
The decoded PCM output file name is "mp3.mp3.pcm"

E:\Tools\隐写\音频隐写\MP3Stego>Decode.exe -X -P 1068 mp3.mp3
MP3StegoEncoder 1.1.17
See README file for copyright info
Input file = 'mp3.mp3' output file = 'mp3.mp3.pcm'
Will attempt to extract hidden information. Output: mp3.mp3.txt
```

<http://blog.csdn.net/wanzt123>

尝试用密码1067提取，成功得到flag

```
E:\Tools\隐写\音频隐写\MP3Stego>Decode.exe -X -P 1067 mp3.mp3
MP3StegoEncoder 1.1.17
See README file for copyright info
Input file = 'mp3.mp3' output file = 'mp3.mp3.pcm'
Will attempt to extract hidden information. Output: mp3.mp3.txt
the bit stream file mp3.mp3 is a BINARY file
HDR: s=FFF, id=1, l=3, ep=off, br=9, sf=0, pd=1, pr=0, m=0, js=0, c=0, o=0, e=0
alg.=MPEG-1, layer=III, tot bitrate=128, sfrq=44.1
mode=stereo, sblim=32, jsbd=32, ch=2
[Frame 202]Avg slots/frame = 415.909; b/smp = 2.89; br = 127.372 kbps
Decoding of "mp3.mp3" is finished
The decoded PCM output file name is "mp3.mp3.pcm" http://blog.csdn.net/wanzt123
```

```
mp3.mp3.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
flag {Brute_f0Rce_iS_W0nderful}

http://blog.csdn.net/wanzt123
```