

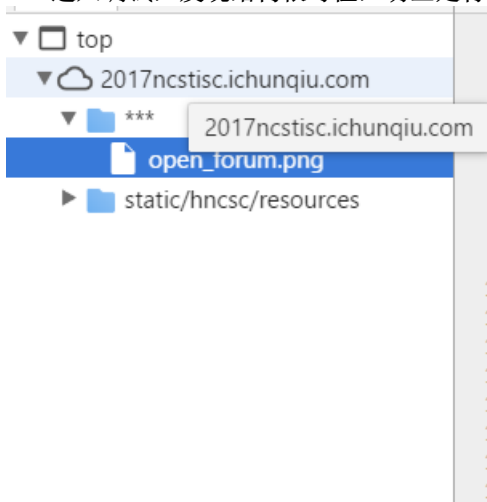
转载

[weixin\\_30332241](#) 于 2017-07-09 22:05:00 发布 130 收藏  
原文链接: <http://www.cnblogs.com/yell/p/7143370.html>  
版权

Misc 100



- 下载文件之后是一个zip压缩包。因为一开始没有给任何提示信息，题目也什么都没说，爆破了一会无果。同时不是伪加密，所以应该是明文攻击。之后官方给出提示，是一个网址。
- F12进入调试，发现结构很奇怪，明显是特意这么弄的



- 之后以为是把这个html给更改为png，然后进行压缩。但是压缩信息与原压缩包中不符，所以不对。



- 后来又经过很长时间的寻找，突然想到界面的图片是不是可以利用，果然发现了一个和压缩包中相同命名的图片

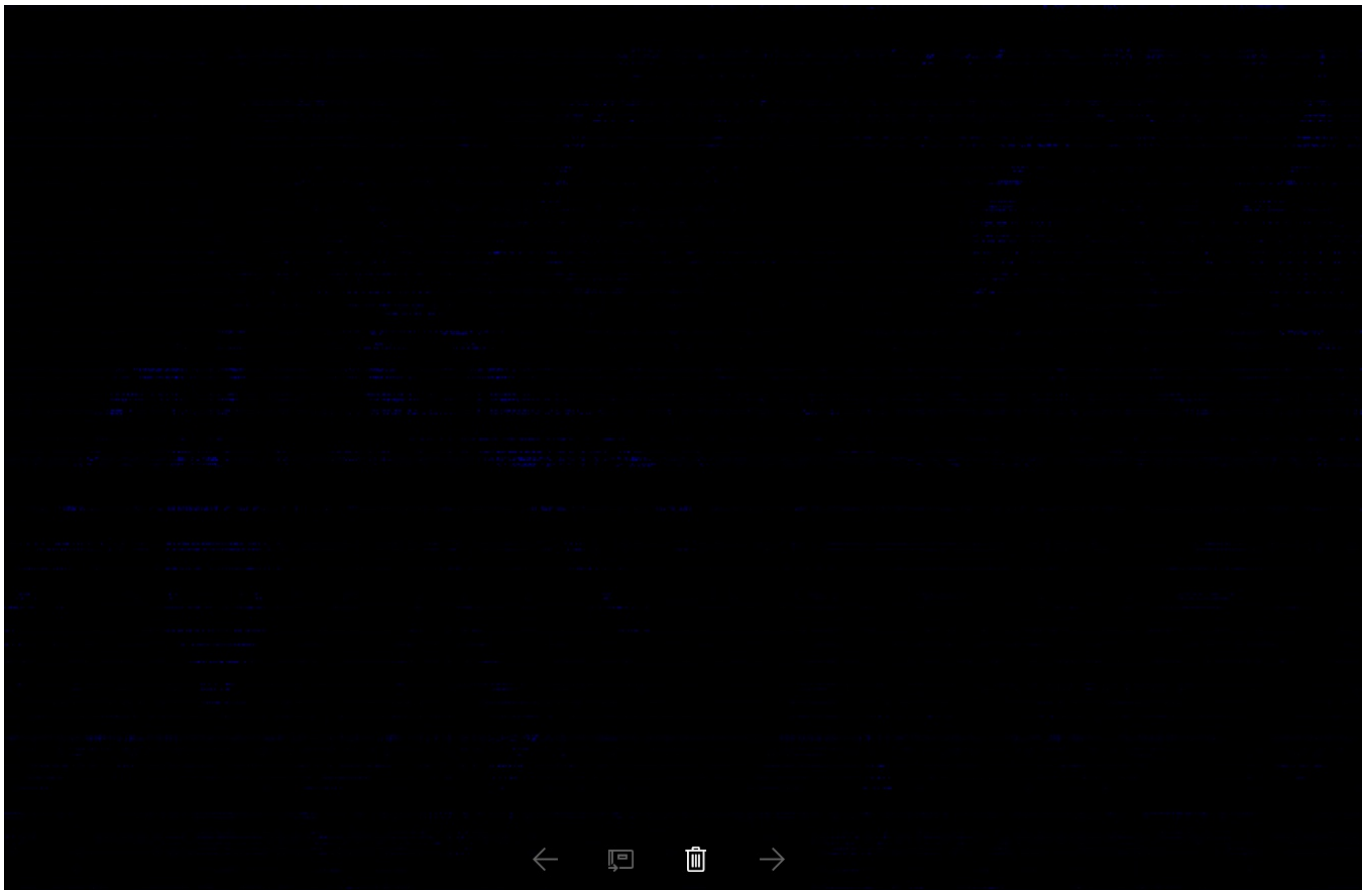
```

</div>
<a href="https://2017ncstisc.ichunqiu.com/kaiwusignup">
</div>
<div class="peaks">
<ul>

```



- 把图片下载后再压缩，信息与压缩包中一致。使用archpr明文攻击，果然跑出来了。
- 又得到两张相同图片，因为图片大小不一致，想到以前做的题，有没有可能是异或后改二维码。就又把图片做对比。
- 但是区别太大，没法继续进行下去



## Misc 100

**300pt** **传感器1**

分值：100分 未解答 第一名：61d 第二名：Mirage 第三名：Nebula

题目名称：apk crack

已知ID为0x8893CA58的温度传感器的未解码报文为：3EAAAAA56A69AA55A95995A569AA95565556

第一名：S3c N0t B4d  
第二名：WHU\_DAWN 第三名：61d

此时有另一个相同型号的传感器，其未解码报文为：3EAAAAA56A69AA556A965A5999596AA95656

请解出其ID，提交flag(不含0x的hex值)

**250pt**

题目名称：Flag :

题目类型：Web

第一名：Conquer 第二名：372

- 一开始是以为和以前的简单解密题一样，通过异或操作得到结果，但是很多次都不对，后来经过很多方法，考虑到是不是编码。找到以前的一些题的writeup

```

# -*- coding: utf-8 -*-
# coding:utf8

hexstr = "3EAAAAA56A69AA55A95995A569AA95565556"
id = "8893CA58"
# 30
# hexstr = "3EAAAAA56A69AA55A95995A569AA95565556"
# 转二进制字符串
s = ''
t = ''
for i in xrange(len(hexstr) / 2):
    ch = hexstr[i * 2: i * 2 + 2]
    b = bin(int(ch, 16))[2:]
    b = '0' * (8 - len(b)) + b
    s += b
    r = ''
print 's- >' + s
print s.__len__()

for i in xrange(len(id) / 2):
    ch = id[i * 2: i * 2 + 2]
    a = bin(int(ch, 16))[2:]
    a = '0' * (8 - len(a)) + a
    t += a
    r = ''
print 'id- >' + t
print t.__len__()

# 曼彻斯特解码, 01对应数据1, 10对应0
for i in xrange(len(s) / 2):
    c = s[i * 2: i * 2 + 2]
    if c == '01':
        r += '1'
    else:
        r += '0'
print 'r- >' + r
ret = ''
for i in xrange(len(r) / 8):
    c = r[i * 8: i * 8 + 8][::-1] # 调整字节序
    print str(r[i * 8: i * 8 + 8]) + '- >' + c
    ret += hex(int(c, 2))[2:].upper()
print ret
def main():
    pass
if __name__ == '__main__':main();

```

```
G:\Python27\python.exe G:/Python27/Lib/site-packages/PIL/2.py
s- > 001111101010101010101010101010101001010110101001101001101010010101011001100101011010010101100110010101101001010110011
144
id- > 10001000100100111100101001011000
32
r- > 000000000000001110001001000011110001110101110011100100000111111011111110
00000000- > 00000000
00000011- > 11000000
10001001- > 10010001
00001111- > 11110000
00011101- > 10111000
01110011- > 11001110
10010000- > 00001001
01111110- > 01111110
11111110- > 01111111
0C091F0B8CE97E7F

Process finished with exit code 0
```

和题目所给ID不太一样，不管是长度还是其他什么。又试了好几种编码的排列，都没能得到那个ID值

我觉得大方向应该是没错的，通过题目给的ID和报文信息，找出对应的编码方案，然后用这个方案直接跑出另外那个报文的ID值。同理，Misc 250也是一样的，但是那题的编码方案应该是要在这基础上更进一步的。

转载于:<https://www.cnblogs.com/ycll/p/7143370.html>