西湖论剑2019--一道MISC题目的解题思路

转载

weixin_30687811 于 2019-04-07 21:45:00 发布 365 收藏
文章标签： python
原文链接：http://www.cnblogs.com/kevinbruce656/p/10667333.html

TTL题的writeup

第一次打西湖论剑，啥都不懂，被题目虐的很惨，一共就做出来两道题，但也算有收获。这里分享一下TTL那道misc题目的writeup,算是给自己点安慰吧。

题目描述

我们截获了一些IP数据报，发现报文头中的TTL值特别可疑，怀疑是通信方嵌入了数据到TTL，我们将这些TTL值提取了出来，你能看出什么端倪吗？

打开题目中的附件是这样的：


```
TTL=127
TTL=191
TTL=127
TTL=191
TTL=127
TTL=191
TTL=127
TTL=191
TTL=127
TTL=191
TTL=127
TTL=63
TTL=63
TTL=255
TTL=191
TTL=63
TTL=127
TTL=191
TTL=127
TTL=191
TTL=127
TTL=191
TTL=127
TTL=191
```

一开始没看出啥来，感觉像ASCII码，但是，有好多都大于127了，所以应该不是ASCII码。

后来队友跟我说，可以把每个TTL值二进制码的高两位（高两位比特的数在数据传输中不容易受影响）拿下来，每4个TTL值凑出一个字节的二进制数来。具体为啥想到这个，他跟我说是之前正好看过一篇讲这个的文章。等我找到文章在贴链接。

之后就是写脚本把想法实现了，这里用到了python。

```
with open('ttl.txt') as f:
    lines = f.readlines()
n_num = []
#分析出所有的数
for i in lines:
    if i!='\n':
        n_num.append(int(i.replace('TTL=','')))
#拿到每个TTL值的高位
rlt = ''
for i in range(0,len(lines)):
    tmp = bin(n_num[i])[2:]
    tmp = '0'*(8-len(tmp)) + tmp
    rlt += tmp[0:2]
#得到最终的结果并存到文件中
rlt2 = ''
for i in range(0,len(rlt),8):
    rlt2 += chr(int(rlt[i:i+8],2))
with open('fi.txt','w') as f:
    f.write(rlt2.rstrip())
```

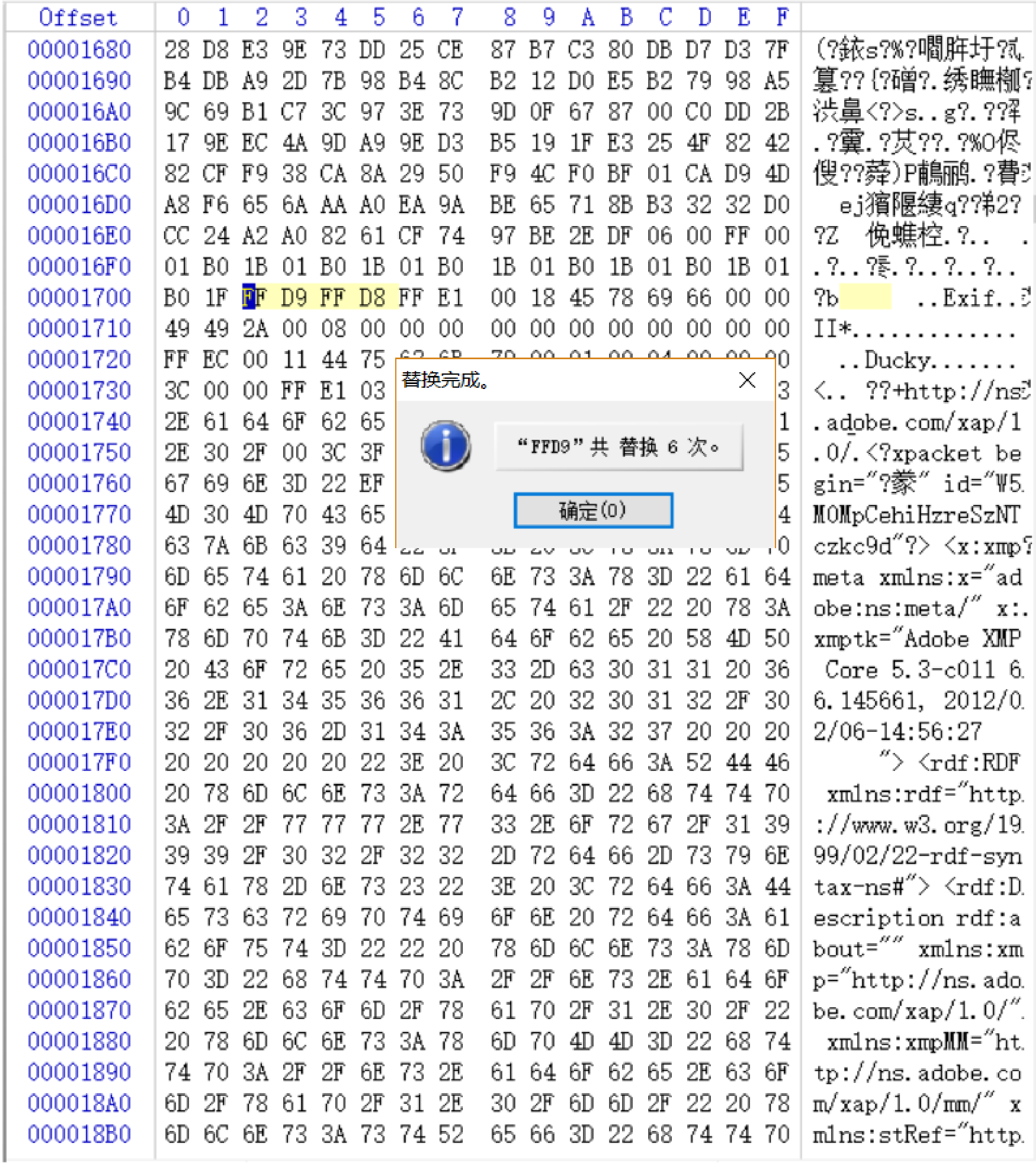
得到的文件中全部都是可打印字符，以FFD8开头，FFD9结束。所以，是jpeg格式的图片无疑了。

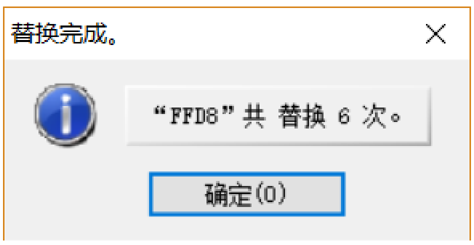
```
ffd8ffe1001845786966000049492a0008000000000000000000000ffec00114475636b
652e636f6d2f7861702f312e302f003c3f787061636b657420626567696e3d22efbbbf22
223f3e203c783a786d706d65746120786d6c6e733a783d2261646f62653a6e733a6d6574
2e332d633031312036362e3134353636312c20323031322f30322f30362d31343a35363a
64663d22687474703a2f2f7777772e77332e6f72672f313939392f30322f32322d726466
6e207264663a61626f75743d222220786d6c6e733a786d703d22687474703a2f2f6e732e
4d4d3d22687474703a2f2f6e732e61646f62652e636f6d2f7861702f312e302f6d6d2f22
2e636f6d2f7861702f312e302f73547970652f5265736f7572636552656623222078 6d70
204353362028576 96e646f7773292220786d704d4d3a496e7374616e636549443d22786d
4545414139353446322220786d704d4d3a446f63756d656e7449443d22786d702e646964
353446322223e203c786d704d4d3a4465726976656446726f6d2073745265663a696e7374
```

```
56b03cd394659f26fb338199ab0d1552a28c446ebf1afbeaecee58129ec31f8dbeebfa6580f618fc6df75fd32c07b0c7e36fbafe99603d863f1b7dd7f4cb035762
f62fd15f3bfdb7ce79ce57fbae570b2b8dfd72f5558ff92ebac0c0d5905e706959981c7cb73762e58e669c4c3cca264aba2f2554d77dd505fdfb04d5ec31f8dbee
bfa6580f618fc6df75fd32c0c0d94e9bfd196aa753de71737ccb1518e5b2796a7116495aebc75afbb06eba9edeed81d5602c0580b01602c0580b01602c0580b07f
ffd9
```

之后把这些16进制字符粘贴到winhex里并保存为jpeg格式的图片，得到了一张30KB的残缺二维码。30KB，还残缺！！！一定是夹杂私货了，于是就搜了下FFD8和FFD9的数量，正好6对。

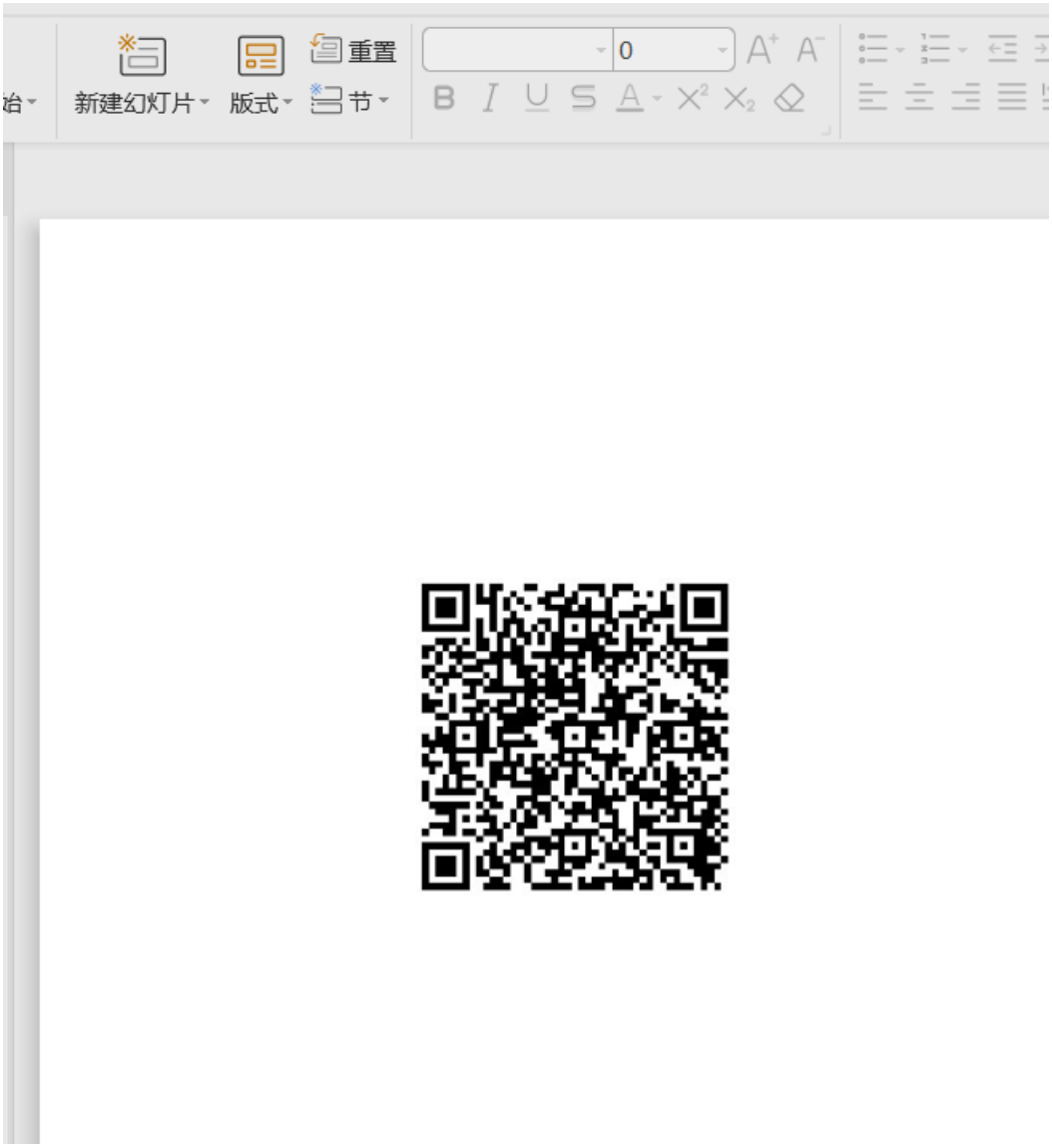
（用搜索的话只能定位，可以用替换搜索，这样可以计数）





接着就是把图片拆出来了，使用binwalk，会分出6张二维码图片，用PPT的插图和对齐的功能拼接一下，得到一

个二维码。



（PPT拼图真好用）

本以为扫完码flag就出来了，没想到还有一层加密，带秘钥的，这里我们使用的autokey解密。

之前说的维吉尼亚，不好意思，最开始我们用的维吉尼亚是个错误的被flag包裹的字符串，后来用的和他相似的autokey通过的。感谢那位发现错误我的博主。

扫码后的结果。

key:AutomaticKey cipher:fftu{2028mb39927wn1f96o6e12z03j58002p}

使用autokey解开密文：（用了个在线的，可能会慢一点）

http://www.practicalcryptography.com/ciphers/classical-era/autokey/

得到这么个东西：flagabdfdeabee，数字和括号没处理，自己手动加上就过了。
flag{2028ab39927df1d96e6a12b03e58002e}
参加完这个比赛，感觉除了隐写能写点，其他的，真的是欠缺！尤其那个PWN和reverse，得好好努力了。或许我需要一个大师傅来带带我，求带啊！

转载于:https://www.cnblogs.com/kevinbruce656/p/10667333.html