CTF做题总结(一)

原创

Qwzf 于 2020-01-14 19:57:51 发布 1448 收藏 2

分类专栏： CTF 文章标签： CTF

本文链接：https://blog.csdn.net/qq_43625917/article/details/96152962

---

ctf做题总结一

上个周做了一道MISC题和两道Crypto题。感觉还是总结一下比较好，毕竟做题时查了很多大佬的博客和一些知识点。

1、Crypto-哈夫曼树和哈夫曼编码

题目下载

刚看到这道题时我毫无头绪。毕竟在写这篇博客时，还没有学过数据结构。而哈夫曼树是数据结构里的一个重要部分。于是我便在百度上搜索有关哈夫曼树和哈夫曼编码的知识。。。。。。。。

哈夫曼树

哈夫曼最大的目的是为了解决当你远距离通信(电报)的数据传输的最优化问题

路径：树中一个结点到另一个结点之间的分支序列构成两个结点间的路径
路径长度：路径上的分支数目
树的路径长度：树根到每个结点的路径长度的和
结点带权路径长度：结点到树根的路径长度与结点的权的乘积
树的带权路径长度：树中所以叶子结点的带权路径长度之和（WPL）

哈夫曼编码

学习完这些知识点后，我对哈夫曼树和哈夫曼编码有了大致的了解，就是找最优二叉树，然后哈夫曼编码就是根据每个字母的出现频率不同，按照它们的权值进行构造哈夫曼树。将所有权值左分支改为0，右分支改为1，得到相应字符的的传输数据。

然后，下载解压下载的题目文件，发现一个txt文件，打开后

果然，是哈夫曼编码，由于我现在的编程能力，写不出有关的编码脚本。所以我画了一个哈夫曼树

根据txt文件，我猜想前五位编码应该是**flag{**，最后一位是**}**，**然后根据这六位编码，调整最底下的**g、l、{、}**的位置。然后得到每个字符的分配权值：

a：000

d：10(或01)

g：00101

f：110

l：00111

0：111

5：01(或10)

{：00100

}：00110

然后把txt文件里的0和1按照上述权值进行分隔

然后比对每个字符的权值，对分隔好的0和1，进行编码，最终得到两个结果

然后最终答案就是其中的一个啦！！！！

2、Crypto-滴答滴答

题目下载

这道题题目没有给任何提示，下载文件，发现并不能打开和用winrar解压文件。于是我便选择用notepad++打开

很显然这是摩斯电码，进行解码得到

发现第一个斜杠前有一串特别的字母MORSEISCOOLBUTBACONISCOOLER仿佛看不太懂，写成小写字母morseiscoolbutbaconiscooler很明显，这句话的意思是“摩斯是酷的，但是培根是更酷的”，那么接下来就应该进行培根解密，直接解密，发现不行。查了查培根加密的格式发现，并没有斜杠**“/”**，于是我用notepad++的替换功能把所有的斜杠去掉，然后培根解密得到

"DO YOU KNOW THE FOUR FENCE ZGIAHYANAUOZNXWI"用百度翻译的意思是“你知道四道栅栏吗？”，可以想到接下来是栅栏密码解密，且每组字数为4。我把所有字符进行解密，发现并没有解出有意义的字符。所以我把“ZGIAHYANAUOZNXWI”进行栅栏解密，最后得到

很显然最终结果出来了，下面进行提交，发现错误。这就应该是格式的问题了，调过之后成功提交了。。

3、MISC-TTL字段

题目下载

从题目中，我们可以知道这道题是TTL字段。然而对此我有点懵，从来没有接触过它。于是我便查有关资料和一些大佬的博客

下载解压题目文件，发现ttl.txt文件，打开发现ttl.txt中的ttl只有4个值**63,127,191,255**

写出他们的二进制表示后发现只有最高两位不同，**63-00111111、127-01111111、191-10111111、255-11111111**。于是把4个值替换成**00、01、10、11**如果传输4个就是一字节，取前面的2位组成8位，对照二进制字母表，可以发现前面是ffd8，jpg图片标志

因为这个脚本我写不出来，于是用了大佬的脚本

```
fp = open('ttl.txt','r')
a = fp.readlines()
p = []
for i in a:
    p.append(int(i[4:]))
s = ''
for i in p:
    if i == 63:
        a = '00'
    elif i == 127:
        a = '01'
    elif i == 191:
        a = '10'
    elif i == 255:
        a = '11'
    s += a
# print(s)

import binascii
flag = ''
for i in range(0,len(s),8):
    flag += chr(int(s[i:i+8],2))
flag = binascii.unhexlify(flag)
wp = open('res.jpg','wb')
wp.write(flag)
wp.close()
#00111111 63
#01111111 127
#10111111 191
#11111111 255
```

然后把ttl.txt和这个脚本(脚本文件名我写成了1.py)放在同一路径，在cmd命令行输入命令`python 1.py`生成了一个fi.txt文件，打开

这是16进制编码，所以把这些编码粘贴在winhex里生成一个jpg文件，打开

发现只有二维码的一部分，在最后转换出的结果中，发现了六个jpg的文件头（ffd8），说明这就是六张图片，用foremost直接分开(或用strgsolve分离图片)就好了，之后用ps(或ppt)拼在一块

扫描结果如下所示：

应该就是AutoKey(自动密钥密码)那个加密，找了个在线网站解密得到

得到最终flag了！！！！！

这便是我上周做题的总结，小白进阶ing，欢迎大佬批评指正！