CTF 杂项之隐写术、Crypto

1、图片隐写：

1.1、将图片里的数据转换成二维码：

用 linux 下的信息提取工具 Binwalk 看一下：

root@kali:~/Desktop# binwalk 图片名

DECIMAL HEXADECIMAL DESCRIPTION

---

0 0x0 PNG image, 1000 x 562, 8-bit/color RGBA, non-interlaced

91 0x5B Zlib compressed data, compressed

3526 0xDC6 Zlib compressed data, best compression

1421307 0x15AFFB Zlib compressed data, default

compression 后面是 Zlib 压缩的数据，写个脚本解压一下：

python 提取脚本：

```
from PIL import Image
from zlib import *
data = open('图片名','rb').read()[0x15AFFB:]
data = decompress(data)
img = Image.new('1', (25,25))
d = img.load()
```

for n,i in enumerate(data):

d[(n%25,n/25)] = int(i)*255

f = open('flag.png','wb')

img.save(f)

2、音频、视频隐写术

2.1、音频加密

（1）在 mp3 中插入密文：

用 MP3Stego 进行加密解密：

加密：encode -E 加密文本 -P 密码 mp3 文件

解密：decode -X -P 密码 mp3 文件

3、Crypto 及解密脚本

3.1、培根加密

一段字符串用两种字体书写，然后按字体来决定其代表“A”还是“B”。

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
#__author__ = 'tyomcat'
# 培根解密代码，两种加密方式
import sys
def peig1(m):
basic1 = {
'AAAAA' : 'A',
'AAAAB' : 'B',
'AAABA' : 'C',
```

```
'AAABB' : 'D',
'AABAA' : 'E',
'AABAB' : 'F',
'AABBA' : 'G',
'AABBB' : 'H',
'ABAAA' : 'I',
'ABAAB' : 'J',
'ABABA' : 'K',
'ABABB' : 'L',
'ABBAB' : 'N',
'ABBBA' : 'O',
'ABBBB' : 'P',
'BAAAA' : 'Q',
'BAAAB' : 'R',
'BAABA' : 'S',
'BAABB' : 'T',
'BABAA' : 'U',
'BABAB' : 'V',
'BABBA' : 'W',
'BABBB' : 'X',
'BBAAA' : 'Y',
'BBAAB' : 'Z'
}
output = ''
for i in range(0, len(m) - 4, 5):
temp = m[i: i + 5]
output += basic1[temp]
return output
```

```
def peig2(m):
    basic2 = {
    'AAAAA' : 'A',
    'AAAAB' : 'B',
    'AAABA' : 'C',
    'AAABB' : 'D',
    'AABAA' : 'E',
    'AABAB' : 'F',
    'AABBA' : 'G',
    'AABBB' : 'H',
    'ABAAA' : 'I',
    'ABAAA' : 'J',
    'ABAAB' : 'K',
    'ABABA' : 'L',
    'ABABB' : 'M',
    'ABBAA' : 'N',
    'ABBAB' : 'O',
    'ABBBA' : 'P',
    'ABBBB' : 'Q',
    'BAAAA' : 'R',
    'BAAAB' : 'S',
    'BAABA' : 'T',
    'BAABB' : 'U',
    'BAABB' : 'V',
    'BABAA' : 'W',
    'BABAB' : 'X',
    'BABBA' : 'Y',
    'BABBB' : 'Z'
```

```
    }
    output = ''
    for i in range(0, len(m) - 4, 5):
       temp = m[i: i + 5]
       output += basic2[temp]
    return output
if __name__ == '__main__':
    m = raw_input("请输入密文:")
    mode = input("选择密文对应的方式 1 or 2: ")
    if len(m)%5 == 0:
    l = []
        k = []
        for i in xrange(len(m)/5):
            l.append(m[:5])
            m = m[5:]
        if mode == 1:
            for i in l:
               if i.isupper():
                   k.append(peig1(i))
               else:
                   i = i.upper()
                   k.append(peig1(i))
        elif mode == 2:
            for i in l:
                if i.isupper():
                    k.append(peig2(i))
                else:
                    i = i.upper()
```

```
                k.append(peig2(i))
    flag = ''
    for i in k:
    flag+=i[0]
    print flag
```

加密的解密脚本：

```
#! /usr/bin/env python
# coding=utf-8
#__author__ = 'tyomcat'
def convert(c, key, start = 'a', n = 26):
    a = ord(start)
    offset = ((ord(c) - a + key)%n)
    return chr(a + offset)
def caesarEncode(s, key):
    o = ""
    for c in s:
        if c.islower():
           o+= convert(c, key, 'a')
        elif c.isupper():
            o+= convert(c, key, 'A')
        else:
            o+= c
    return o
def caesarDecode(s, key):
    return caesarEncode(s, -key)
if __name__ == '__main__':
```

```
    for key in range(27):
      e='Jr1p0zr2VfPp'     #写这里
      d = caesarDecode(e, key)
      print d
      print '\n'
```

3.2、词频分析

```
#! /usr/bin/env python
# -*- coding:utf-8 -*-
# __author__ == "tyomcat"
import operator
str=''  #词频分析的字符串
payloads = 'abcdefghijklmnopqrstuvwxyz'
payloads = payloads.upper()
# print payloads
dists = {}
for x in payloads:
    dists[x] = 0
# print x,dists[x]
for s in str:
    dists[s] += 1
    ans = ''
    res = sorted(dists.iteritems(), key=operator.itemgette
r(1), reverse=True)
for r in res:
    ans += r[0]
    print r
```

```
    print ans
```

word 文件的宽窄字距加密，先将其整理成 xml 文件：

```
# /usr/bin/env python

#coding: utf-8

#__author__ == "tyomcat"

import  xml.dom.minidom

import sys

reload(sys)

sys.setdefaultencoding('utf-8')

dom = xml.dom.minidom.parse('1.xml')  #xml 文件

print dom

root = dom.documentElement

str1 = ''

bb = root.getElementsByTagName('w:spacing')

b= bb[10]

print b

b1 = b.getAttribute("w:val")

for k in range(10,len(bb)):

    if bb[k].getAttribute("w:val") == "2" :

        str1 += '1' * len(str(bb[k].parentNode.parentNode.g
etElementsByTagName('w:t')[0].childNodes[0].data).decode
('utf-8'))

    elif bb[k].getAttribute("w:val") == "-2" :

        str1 += '0' * len(str(bb[k].parentNode.parentNode.g
etElementsByTagName('w:t')[0].childNodes[0].data).decode
('utf-8'))

    print str1

    print len(str1)
```