


title: 问鼎杯决赛
date: 2017-09-24 09:00:08

tags:

决赛去水了一把

3-1 合格铲屎官

下载下来一张图片，用神奇stegsolve打开,随便按发现通道最低位有点奇怪，先用lsb提取一下。发现熟悉的pk，save bin为一个zip文件，打开后发现是一串base64加密后的字符串，先解码看一下是什么东西.发现是png文件头，直接写脚本提取一下

```
import base64
f=open('flag.png','wb')
a=''
iVBORw0KGgoAAAANSUHEUgAAAPoAAD6CAYAAACI7Fo9AAAAAXNSR0IArs4c6QAAAAARnQU1BAACxjwv8YQUAAAJcEhZcwAAEnQAABJ0Ad5mH3gAAAVqSURBVHhe7d1bTu
...
f.write(base64.b64decode(a))
```

打开即是flag

3-2 easy_py

下载下来一个压缩包,先试试伪加密，用zipCenOp打开之后发现果然加密标志没了。然后把flag.pyc反编译一下,一个加密函数(队友说是rc4)，不过需要个key。于是又打开了key文件，发现是一串熟悉的东西,懒得写脚本，直接用编辑器的替换功能，从9开始替换，把这么一串东西变成一个表达式

得到key之后替换掉加密算法中的key，然后根据加密算法写个解密算法

```
# uncompyle6 version 2.9.10
# Python bytecode 2.7 (62211)
# Decompiled from: Python 2.7.11 (v2.7.11:6d1b6a68f775, Dec 5 2015, 20:40:30) [
# Embedded file name: /home/ctf/WDCTF2017/test.py
# d: 2017-09-08 19:54:01
import random
import base64
from hashlib import sha1
strCipher = 'Xw6aM5fbiQ0kkezmbdLC7Gbnj5siJJc5DpzKvjtdKPKT3A=='
key = 'I_4m-k3y'

def crypt(data, key):
    x = 0
    box = range(256)
    for i in range(256):
        x = (x + box[i] + ord(key[i % len(key)])) % 256
        box[i], box[x] = box[x], box[i]

    x = y = 0
    out = []
    for char in data:
        x = (x + 1) % 256
        y = (y + box[x]) % 256
        box[x], box[y] = box[y], box[x]
        out.append(chr(ord(char) ^ box[(box[x] + box[y]) % 256]))

    return ''.join(out)

def decrypt(data, key):
    x = 0
    box = range(256)
```

```

for i in range(256):
    x = (x + box[i] + ord(key[i % len(key)])) % 256
    box[i], box[x] = box[x], box[i]
x = y = 0
data1=[]
for char in data:
    x=(x+1)%256
    y=(y+box[x])%256
    box[x],box[y] = box[y], box[x]
    data1.append(chr(ord(char) ^ box[(box[x] + box[y]) % 256]))
return ''.join(data1)

def encode(data, key, encode=base64.b64encode, salt_length=16):
    salt = ''
    for n in range(salt_length):
        salt += chr(random.randrange(256))
    #salt='11'
    data = salt + crypt(data, sha1(key + salt).digest())
    if encode:
        data = encode(data)
    return data

def decode(data, key, decode=base64.b64decode, salt_length=16):
    salt = ''
    if decode:
        data=decode(data)
    for n in range(salt_length):
        salt += chr(random.randrange(256))
    #salt='11'
    salt=data[:16]
    out=data[16:]
    return decrypt(out,sha1(key + salt).digest())

print decode(strCipher,key)

```

得到flag

4-1 简单加密

py文件还是个加密函数

```

from hashlib import sha256

def xor(a,b):
    return ''.join([chr(ord(i)^ord(j)) for i,j in zip(a,b)])

def HASH(data):
    return sha256(data).digest()[:8]

def bes_encrypt(subkeys, data):
    i = 0
    d1 = data[:8]
    d2 = data[8:]
    for i in subkeys:
        d1 = xor(xor(HASH(d2),i),d1)
        d1,d2 = d2,d1
        print (d2+d1).encode('hex')
    return d2 + d1

def key_schedule(key):
    subKeys = []
    subKey = key
    for i in xrange(16):
        subKey = HASH(subKey)

```

```

        subKeys.append(subKey)
    return subKeys

def bes(key,data):
    subKeys = key_schedule(key)
    return bes_encrypt(subKeys, data).encode('hex')

if __name__ == "__main__":
    print bes('wdctfhhh','This_is_the_flag')
    # 19714d622d75f32fd9bd98feaa93df0d

```

因为没有随机数什么的，根据加密函数稍微改改写个解密函数就好了

```

from hashlib import sha256

def xor(a,b):
    return ''.join([chr(ord(i)^ord(j)) for i,j in zip(a,b)])

def HASH(data):
    return sha256(data).digest()[:8]

def bes_encrypt(subkeys, data):
    i = 0
    d1 = data[:8]
    d2 = data[8:]

    print d2.encode('hex')
    for i in subkeys:
        d1 = xor(xor(HASH(d2),i),d1)
        d1,d2 = d2,d1

    return d2 + d1

def bes_decrypt(subkeys,data):
    i=0

    d2= data[:16]
    d2=d2.decode('hex')
    d1= data[16:]
    d1=d1.decode('hex')

    subkeys=subkeys[::-1]
    for i in subkeys:
        d1,d2=d2,d1
        d1 = xor(xor(HASH(d2),i),d1)

    return d1+d2

def key_schedule(key):
    subKeys = []
    subKey = key
    for i in xrange(16):
        subKey = HASH(subKey)
        subKeys.append(subKey)
    return subKeys

def bes(key,data):
    subKeys = key_schedule(key)
    return bes_encrypt(subKeys, data).encode('hex')

def besdd(key,data):
    subKeys = key_schedule(key)
    return bes_decrypt(subKeys, data)

```

```
if __name__ == "__main__":  
  
    print besdd('wdctfhhh','19714d622d75f32fd9bd98feaa93df0d')  
  
# 19714d622d75f32fd9bd98feaa93df0d
```

□

附加题:万里挑一

下载下来一个压缩包,里面**1024**个文件,随便点个进去都是一堆十六进制,想想万里挑一,感觉像是在里面找一个正常的东西,就随便点点。发现有点不正常的地方,📄,这个文件和前面的文件有很明显的时间差,像是前面是用什么脚本生成的,而从这里开始是加进去的东西。

那就点开这个文件,发现很标准的**flag**形式,WDFLAG{},那就是它了。

不过一开始并不知道什么加密方法,用**ascii**试了试发现不对,然后仔细观察发现每一位都小于**10**,而且都只有两位,第二位都小于等于**4**,再想到提示提到短信,那应该就是手机键盘加密了,解开之后再用凯撒加密解开就得到了**flag**。