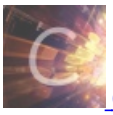
ctf-misc总结（二）

原创

博闻善行 于 2020-05-12 22:46:11 发布 1481 收藏 8

分类专栏： CTF

本文链接：https://blog.csdn.net/weixin_41038905/article/details/106086462



文章目录

常用在线网址

（1）16进制字符串文本转换https://www.bejson.com/convert/ox2str/
（2）unicode在线编解码：https://www.css-js.com/tools/unicode.html
（3）在线摩斯密码解密 https://www.bejson.com/enc/morse/
（4）MD5在线爆破 https://www.cmd5.com/
（5）维吉利亚在线解密 https://guballa.de/startseite
https://www.guballa.de/vigenere-solver
（6）在线大素数分解 http://factordb.com/
（7）在线将cap包修复为pcap包 http://f00l.de/hacking/pcapfix.php
（8）在线PS网址https://www.uupoop.com/

常用工具

：

1.uncompyle2是Python2.7的反编译工具，它可以把python生成的pyo、pyc字节码文件反编译为十分完美的源码，并可以将反编译后的源码再次生成字节码文件

安装方法：

```
git clone https://github.com/wibiti/uncompyle2
cd uncompyle2
python setup.py install
```

使用方法示例：
使用帮助：

```
uncompyle2 -h
```

例如有一个pcat.pyc，想反编译输出文件为pcat.py，你必须这样写：

```
uncompyle2 -o pcat.py pcat.pyc
```

2.git泄露：考虑 git 泄露 下载 Git_Extract 使用 python git_extract.py 加上.git 所在目录，还原出另一个 flag.txt,及 s.py, 删除原来的 flag.txt，将新抽取的 flag.txt.xxx 重命名为 flag.txt，运行 s.py 得到 flag
提取远程 git 泄露或本地 git 的工具
下载地址：https://github.com/gakki429/Git_Extract

3.压缩包解密工具除了一个archpr工具外还有一个azpr

4.winhex进行16进制搜索查找简单编辑，hdx可以大量文本编辑，下载地址l

5.要复制winhex16进制对应的文本内容，可以先吧16进制复制出来进行字符串转换即可

6.流量包分析时对请求方式筛选很有用：http.request.method==POST

7.除了使用binwalk提取文件以外，kali自带了foremost工具用来提取文件

8.存储该文件电脑的一个内存快照题目解法参考：https://www.ichunqiu.com/writeup/detail/1415
9.zsteg可以检测PNG和BMP图片里的隐写数据。

```
git clone http://www.github.com/zed-0xff/zsteg
```

安装方法：

```
git clone https://github.com/zed-0xff/zsteg
cd zsteg/
gem install zsteg
```

直接文本搜索
使用strings.exe对kill.pcapng中的可打印字符进行提取，保存到strings.txt文件中，命令【strings.exe ****.pcapng >strings.txt】
或者直接使用notepad++打开搜索

CTFcrack这个工具尝试一下

RSA题目

```
from pwn import *
from hashlib import sha256
from gmpy2 import *
```

1.已知n,c,e,求m

```
n=p*q
e=65537
p = 289540461376837531747468286266019261659
q = 306774653454153140532319815768090345109
phi = (p-1)*(q-1)
d = invert(e, phi)
m = pow(c, d, n)
```

2.已知n,c,e=3，爆破

```
i = 0
while True:
    if iroot(c + i * n, 3)[1] == True:
        m = int(iroot(c + i * n, 3)[0])
        break
    i += 1
```

3.广播攻击，已知多组n和c

```
n = [n1, n2, n3]
C = [c1, c2, c3]
N = 1
for i in n:
    N *= i
Ni = []
for i in n:
    Ni.append(N / i)
T = []
for i in xrange(3):
    T.append(long(invert(Ni[i], n[i])))
X = 0
for i in xrange(3):
    X += C[i] * Ni[i] * T[i]
m3 = X % N
m = int(iroot(m3, 3)[0])
```

4.模不互素，已知两组n,c,e

```
p = gcd(n1, n2)
q1 = n1/p
q2 = n2/p
phi1 = (p-1)*(q1-1)
phi2 = (p-1)*(q2-1)
d1 = invert(e1, phi1)
d2 = invert(e2, phi2)
m1 = pow(c1, d1, n1)
m2 = pow(c2, d2, n2)
```

5.共模攻击，已知两组n,c,e,其中n相同

```
_, s1, s2= gcdext(e1, e2)
if s1 < 0:
    s1 = -s1
    c1 = invert(c1, n)
if s2 < 0:
    s2 = -s2
    c2 = invert(c2, n)
m = (pow(c1, s1, n) * pow(c2, s2, n)) % n
```

6.wiener_attack

d = 42043

m = pow(c, d, n)

7.已知n,e,c和nextprime§*nextprime(q)

```
nn=nextprime(p)*nextprime(q)
t = nn - n
f1 = lambda x, y: pow(x * y - t, 2) - 4 * n * x * y
f2 = lambda x, y, s: (t - x * y - s) / (2 * x)
token = 0
for x in xrange(1, 3000):
    if token == 1:
        break
    for y in xrange(1, 3000):
        if f1(x, y) >= 0:
            s, b = iroot(f1(x, y), 2)
            if b:
                if is_prime(f2(x, y, int(s))):
                    p = f2(x, y, int(s))
                    token = 1
                    break
q = n/p
phi = (p-1)*(q-1)
d = invert(e, phi)
m = pow(c, d, n)
```

8.dp泄露（ dp = d%(p-1)）

为快速实现RSA，会使用 dp = d%(p-1) 来进行计算，若该参数泄露，私钥d可被求出。

```
for i in range(1,65538):
    if (dp*e-1) % i == 0:
        if n%(((dp*e-1)/i)+1) == 0:
            p = ((dp*e-1)/i)+1
            break
q = n/(((dp*e-1)/i)+1)
phi = (p-1)*(q-1)
d = invert(e, phi)
m = pow(c, d, n)
```

9.私钥d低一半比特泄露攻击half_d

对于一个较小的e来讲（例如e≤65537），d的上半部分可以被有效的估计出来，根据RSA定义我们有：

ed ≡ 1 (mod phi)

即：

ed ≡ k * phi + 1

由于：

phi = (p-1) * (q-1)

= p * q - p - q + 1

= n - p - q + 1

所以我们有：

ed = k * (n+1) - k * (p+q) + 1

由RSA定义可知d<φ(n)，而我们知道ed-k*φ(n)=1>0,因此可知k<e。因此，当e较小时，k就落在了一个较小的搜索空间，我们就可以通过穷举k来估计d。

```
    d = 45159787940421567053389692873525016894044126603328403245044194862092560129767800975750759211073400677059
43166959977421216972923946428438688480550087568522919481229961914648178786968576686996405971916213105804989849441
49740950972453366494422535945732839868669098606348675115592285929727382430314107812389594 67
    m = pow(c, d, n)
```

.10 AES解密（需要key）

```
#coding=utf-8
from Crypto.PublicKey import RSA
from Crypto.Cipher import AES
key="copy__white__key"
obj=AES.new(key,AES.MODE_ECB)
s=open("AES.encryt","rb").read()
str=obj.decrypt(s)
with open(r'next.zip','wb') as f:
    f.write(str)#解密后得到文件
```

2.工具winhex 和010Editor都需要安装

3.AES秘钥解密

```
#coding=utf-8
from Crypto.Cipher import AES
key="copy__white__key"
obj=AES.new(key,AES.MODE_ECB)
path="/home/adworld/MISC/i_chunqiu/CryMisc_E1C844B98C4CAC14060994BD1933AF9F/gogogo/AES.encryt"
s=open(path,"rb").read()
str=obj.decrypt(s)
with open(r'next.zip','wb') as f:
  f.write(str)#解密后得到文件
```

另附上青龙组misc图片相关的解密程序供参考：
文件是PNG头，修改文件后缀为png，该文件像素为12*36

```
#对该文件的RGB进行识别
#255 记为1 0记为0

from PIL import Image
#import sys
#im = Image.open(sys.argv[1])
im = Image.open('file.png')
width = im.size[0]
height = im.size[1]

temp = ''
#竖着识别
for w in range(width):
    for h in range(height):
        pixel = im.getpixel((w, h))
        temp += '1' if pixel[0] == 255 else '0'
        temp += '1' if pixel[1] == 255 else '0'
        temp += '1' if pixel[2] == 255 else '0'
print(temp)
```

运行代码得到

```
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111000000011110010000000011000000011111011111111001100001001111111110111110100000111111100100100110000010
01111101011111110101011000000111110101111101010001110101001001110100010101111101010001101101010101111100010
10111110101000101100101110111110001010111111111111101010101000011011111111111111111111111101010110011000100
01111111100100111001010011111110110110111111100000101011110101010001111100111110101011110100100000011000010
11111111101110010110011001100110000001111111111111111111110011101111100001111111000000000000000000001101101
11010111110001011011011101001111000010111111000101011110100100011001010010111110101010101100100011000110
11100011111011110100111011101000011110000111110010100111100000001000101101000111100100100001101100001000010
10011101111100010110001101011110100110111001111111001000111011001000101111111111111111111110100010100000010
11111111111110101000111000011111111110001010111110101000110111111111111100010101111101010001010100111111111
11000101011111010111101111111111111111111010111110100000110100111111111100000101111101111111111111111111111
11111111101111100000001100111111111111000000011111111111111111111111111111111111111111111111111111111111111
111111111111
```

弄成36*36的汉信码，并旋转，反色：

```
#incoding:utf-8
from PIL import Image

str = "11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111110010110110110110110111111111101101011011111001001001011011001001001111111011011011011011011001111111
11111111000111000011010010010111111010000010110111000010010010111101100010001111010010010011000111000111111111111
11110011110011110010010011111111011010111010111000110101101101010001100110000111001001001111001111001111111111111111
11111011111011010101101011111001011011101110001101110110111111001001111010110101010011101111100111111111111110010
11111101101101001101101101111001011111000111010001110000101101011001000111100111001111111111111111111100000000
10110010100110100100010100000010010001010100001110001000110010101111111111111111111111111111111111111100001000100
00110110100011011110100010011000100100111001001000111000000011111111111111111111111111111111111100101100010001101
11111110011111100110111011101111100001000001000100101010111111111111111111111111111111111111100111100111100100100
01110000110110000001101010001011011110110101101111111001001001111001111001111111111111111000111000110010010010111
10001111111100100011001000111010001001111111111110100100101100001110001111111111111110011011011011011011011111110
11001111111111110111101100100100110100111111110110110110110110100111111"
length_str = len(str)
print (length_str)
MAX = int(len(str)**0.5) #根据01的个数生成黑白二维码的 size
print (MAX)

pic = Image.new("RGB",(MAX,MAX))

i=0
for y in range (0,MAX):
    for x in range (0,MAX):
        if(str[i] == '1'):
            pic.putpixel([x,y],(0,0,0))
        else:
            pic.putpixel([x,y],(255,255,255))
        i = i+1

pic.show()
pic.save("flag.png")
```

得到汉信码的图片之后根据汉信码的图片格式修复图片在专门解密汉信码的网站上解密得到flag