# 2020高校战疫CTF wp

## WEB(想复现,web环境关了)

## MISC

### 2019-nCoV

flag{shijiejiayou}

### 简单MISC

一个压缩包和一张图片

1.图片用winhex分析,发现尾部有PK头

```
            ÿÙPK
         ÊµaP▌O´
.   m        ctf.
txtM▌±  0 Å. ÿÿ®
 ▌  JôYÅ|B¸▌{Û C
JHO0ð îA▌Ý÷▌ nJñ
 PK  ?        Êµa
P▌O´ .   m     $
                c
tf.txt
   ´▌*0iÕ Àø<'0ï
Õ  æÌ 0iÕ PK
      Y   S
```

foremost分离一下得到压缩包

```
PS G:\Damya\bugku\foremost-master\foremost-master\binary> .\foremost.exe .\photo.jpg
Processing: .\photo.jpg
|foundat=ctf.txtM婁□
*|
PS G:\Damya\bugku\foremost-master\foremost-master\binary>
```

里面的txt里有摩斯密码

```
./.--./../-.././--/../-.-./.../../-/..-/.-/-/../---/-./--
-/..-./..-/-./../...-/./.-./.../../-/-.--/.--/.-/.-.
转换后
EPIDEMICSITUATIONOFUNIVERSITYWAR
```

2.用密码解密,得到txt里有base64密码

```
VGgxc19pc19GbGFHX3lvdV9hUkVfcmlnSFQ=
转换后
Th1s_is_FlaG_you_aRE_rigHT
```

flag(Th1s_is_FlaG_you_aRE_rigHT)

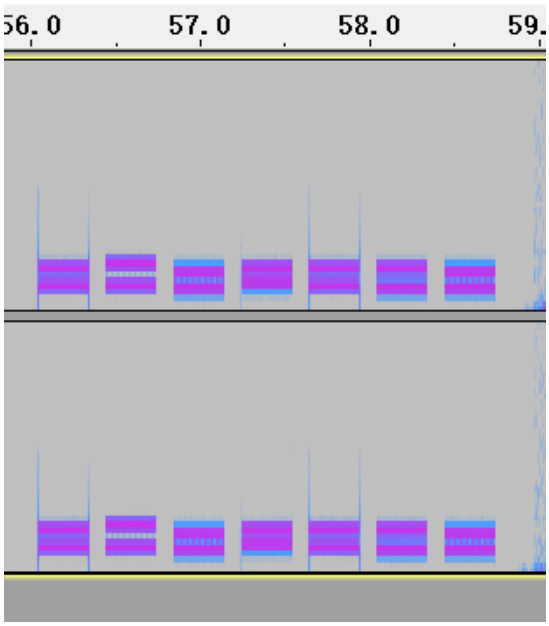
## 隐藏的信息

一个zip和一张残缺二维码

1.zip伪加密破解

```
FA E8 D9 CF A7 BF 5D 40  27 CD BF 00 50 4B 01 02   úèÙÏ§¿]@'Í¿ PK
1F 00 14 00 00 00 08 00  03 71 54 50 5B 5A 0F E0         qTP[Z à
94 01 92 00 AE 36 B4 00  0E 00 40 00 00 00 00 00   ' ®6´   @
00 00 20 08 00 00 00 00  00 00 D2 FE B2 D8 B5 C4         Òþ²Ø
D0 C5 CF A2 2E 77 61 76  0A 00 20 00 00 00 00 00   ÐÅÏ¢.wav
01 00 18 00 7B AA 52 1E  B4 E7 D5 01 02 A8 60 0D       {ªR ´çÕ  ¨`
D0 EB D5 01 AC C7 44 0D  D0 EB D5 01 75 70 18 00   ÐëÕ ¬ÇD ÐëÕ up
01 1A 9E DA 72 E9 9A 90  E8 97 8F E7 9A 84 E4 BF     Úré è  ç  ä¿
A1 E6 81 AF 2E 77 61 76  50 4B 05 06 00 00 00 00   ¡æ ¯.wavPK
01 00 01 00 7C 00 00 00  DC 01 92 00 00 00         |   Ü '
```

得到一个wav文件,用audacity打开,调成频谱图形式,音频尾部有类似拨打电话的按键音




经队友提醒是DTMF Tones密码

参考博客:https://blog.csdn.net/X_s_yu/article/details/103649922

跑matlab脚本算出音频数据表

696 1207
855 1334
855 1207
771 1207
855 1334
771 1334
771 1476
696 1207
855 1334
771 1334
696 1334
696 1207

得到电话号码

187485618521

2.将二维码反色补充后扫出假flag

已扫描到以下内容

flag{this_is_also_not_flag}
解压密码不在这里0.0!

3.二维码图片用winhex分析,尾部有 USE BASE64 TO GET YOUR FLAG 信息

```
00009200  0F C4 0F FC 1E 55 53 45  42 41 53 45 36 34 FC 6F  Ä ü USEBASE64üo
00009470  CF A7 FF 00 67 45 14 50  07 FF D9 54 4F 47 45 54  Ï§ÿ gE P ÿÙTOGET
00009480  59 4F 55 52 46 4C 41 47                          YOURFLAG
```

```
187485618521
转换后
MTg3NDg1NjE4NTIx
```

flag{MTg3NDg1NjE4NTIx}

## ez_mem&usb

一个流量包

1.binwalk captured.pcap –e 提取得到data.vmem

```
root@iZ2ze5rmf8lyj1geahh44hZ:~/_captured.pcap.extracted# ls
28C632D  28C832B.xml  28C8CD2.xml  28CAE63.xml  28CB888.xml  519BE5E  data.vmem
28C6958  28C8860.xml  28C92BF.xml  28CB416.xml  28CC188.zip  519C487  EE8.zip
```

2.volatility –f data.vmem ––profile=WinXPSP2x86 consoles 分析,得到 passwd:weak_auth_top100

```
Cmd #0 @ 0x3609ea0: passwd:weak_auth_top100
Cmd #1 @ 0x5576d0: start wireshark
```

3. volatility –f data.vmem ––profile=WinXPSP2x86 filescan | grep flag 提取得到zip包,解密得到usbdata.txt

4.脚本进行usb键盘解密

```
00:00:09:00:00:00:00:00 00:00:0F:00:00:00:00:00 00:00:04:00:00:00:00:00
00:00:0A:00:00:00:00:00 00:00:2F:00:00:00:00:00 00:00:23:00:00:00:00:00
00:00:26:00:00:00:00:00 00:00:1F:00:00:00:00:00 00:00:27:00:00:00:00:00
00:00:27:00:00:00:00:00 00:00:25:00:00:00:00:00 00:00:20:00:00:00:00:00
00:00:22:00:00:00:00:00 00:00:24:00:00:00:00:00 00:00:25:00:00:00:00:00
00:00:21:00:00:00:00:00 00:00:08:00:00:00:00:00 00:00:06:00:00:00:00:00
00:00:20:00:00:00:00:00 00:00:08:00:00:00:00:00 00:00:07:00:00:00:00:00
00:00:25:00:00:00:00:00 00:00:07:00:00:00:00:00 00:00:1F:00:00:00:00:00
00:00:04:00:00:00:00:00 00:00:23:00:00:00:00:00 00:00:21:00:00:00:00:00
00:00:08:00:00:00:00:00 00:00:24:00:00:00:00:00 00:00:20:00:00:00:00:00
00:00:09:00:00:00:00:00 00:00:08:00:00:00:00:00 00:00:26:00:00:00:00:00
00:00:1E:00:00:00:00:00 00:00:20:00:00:00:00:00 00:00:06:00:00:00:00:00
00:00:27:00:00:00:00:00 00:00:30:00:00:00:00:00
转换后(提交flag形式)
flag{69200835784EC3ED8D2A64E73FE913C0}
```

# CRYPTO(这里贴脚本+解释,另一题还要继续学习)

## lancet

这题和之前做的pico里的rsa-pop-quiz很像,都是答题形式

```
from pwn import *
import gmpy2, base64 from Crypto.Util.number
import bytes_to_long, long_to_bytes
```

```
p = remote('121.37.174.33', 9999) //链接,解决第一题
p.recvuntil('Welcome to RSA WORLD !!!')  //读取信息
p.recvuntil('n:') //读取n:后的数据
n = int(p.recvline().strip())
p.recvuntil('e:') //读取e:后的数据
e = int(p.recvline().strip())
p.recvuntil('flag:') //读取flag:后的数据
flag = int(p.recvline().strip())
log.info(hex(n)) //传入参数
log.info(hex(e))
log.info(hex(flag))
def encrypt(m):  //第二题,选择是,直接用上题数据解密
    p.recvuntil('you can choose what you want here\n')
    p.sendline('1')
    p.recvuntil('send how long you want to encrypt\n')
    p.sendline(str(len(base64.b64encode(m))))
    p.recvuntil('send the message in base64 encode\n')
    p.sendline(base64.b64encode(m))
    p.recvuntil('res:')
    res = int(p.recvline().strip().decode('base64'))
    return res
def decrypt(c):  //第三题,选择否,有个长度判断
    p.recvuntil('you can choose what you want here\n')
    p.sendline('2')
    p.recvuntil('send how long you want to decrypt\n')
    print len(c), len(base64.b64encode(c))
    if (len(base64.b64encode(c)) >= 100):
        p.send(str(len(base64.b64encode(c))))
    else:
        p.sendline(str(len(base64.b64encode(c))))
    p.recvuntil('send the message in base64 encode\n')
    p.sendline(base64.b64encode(c))
    p.recvuntil('res:')
    res = int(p.recvline().strip())
    #res = int(p.recvline().strip().decode('base64'))
    return res
upper_limit = n / (2 ** 1024) //限制最长数据和最短数据
lower_limit = 0
i = 1025
# for 1024 bit n
while i <= 2048:  //以下计算rsa值,用于上面题的解密
    chosen_ct = long_to_bytes(flag*pow(2**i, e, n) % n)
    output = decrypt(chosen_ct)
    if output == 0:
        upper_limit = (upper_limit + lower_limit)/2
    elif output == 1:
        lower_limit = (lower_limit + upper_limit)/2
    else:
        raise Exception
```

```
    i += 1
    print lower_limit, upper_limit
    # Decrypted ciphertext
    print long_to_bytes(upper_limit)
```