

2021强网杯青少赛（qwtac）楼上大佬ddw WriteUp

原创

末小心 于 2021-10-12 20:40:17 发布 104 收藏

分类专栏: [CTFwp](#) [安全工具](#) 文章标签: [算法](#) [网络安全](#)

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/Moxin1044/article/details/120731533>

版权



[CTFwp](#) 同时被 2 个专栏收录

7 篇文章 0 订阅

订阅专栏



[安全工具](#)

4 篇文章 0 订阅

订阅专栏

楼上大佬ddw战队WRITEUP

- 战队信息

战队名称: 楼上大佬ddw

战队排名: 24

- 解题情况



- 解题过程

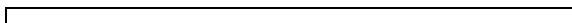
01 签到

操作内容:

下载附件, 打开运行拿到flag



如该题使用自己编写的脚本代码请详细写出, 不允许截图



值: **flag{鲸鱼带你进入鲸奇世界}**

02 Lihua's for

操作内容:

RE 扔进IDA

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
    char flag[42]; // [rsp+20h] [rbp-60h] BYREF
    int a[42]; // [rsp+50h] [rbp-30h] BYREF
    int b[42]; // [rsp+100h] [rbp-30h]
    int i_0; // [rsp+1B4h] [rbp+34h]
    int i; // [rsp+1B8h] [rbp+38h]
    int good; // [rsp+1BCh] [rbp+3Ch]

    _main();
    qmemcpy(a, &unk_103040, sizeof(a));
    puts("input flag");
    scanf("%s", flag);
    puts(flag);
    for (i = 0; i <= 41; ++i)
        b[i] = i * flag[i];
    for (i_0 = 0; i_0 <= 41; ++i_0)
    {
        if (a[i_0] != b[i_0])
        {
            good = 0;
            break;
        }
        good = 1;
    }
    if (good == 1)
        printf("good");
    else
        printf("error!");
    return 0;
}
```

对输入进行异或操作

字符串比较

CSDN @末小心

```
qmemcpy(a, &unk_103040, sizeof(a));
puts("input flag");
scanf("%s", flag);
puts(flag);
for (i = 0; i <= 41; ++i)
    b[i] = i * flag[i];
for (i_0 = 0; i_0 <= 41; ++i_0)
{
    if (a[i_0] != b[i_0])
    {
        good = 0;
        break;
    }
    good = 1;
}
if (good == 1)
    printf("good");
else
    printf("error!");
return 0;
```

CSDN @末小心

下断点，动态调试，然后取a数组的数据

```
Stack[00002AB4]:000000000062FCA9 db 0
Stack[00002AB4]:000000000062FCA8 db 0
Stack[00002AB4]:000000000062FCA0 db 66h
Stack[00002AB4]:000000000062FCA1 db 0
Stack[00002AB4]:000000000062FCA2 db 0
Stack[00002AB4]:000000000062FCA3 db 0
Stack[00002AB4]:000000000062FCA4 db 60h
Stack[00002AB4]:000000000062FCA5 db 0
Stack[00002AB4]:000000000062FCA6 db 0
Stack[00002AB4]:000000000062FCA7 db 0
Stack[00002AB4]:000000000062FCA8 db 63h
Stack[00002AB4]:000000000062FCA9 db 0
Stack[00002AB4]:000000000062FCAa db 0
Stack[00002AB4]:000000000062FCAB db 0
Stack[00002AB4]:000000000062FCAC db 64h
Stack[00002AB4]:000000000062FCAD db 0
Stack[00002AB4]:000000000062FCAE db 0
Stack[00002AB4]:000000000062FCAF db 0
```

CSDN @末小心

写解密代码

```
C:\Users\Administrator\Desktop>python 1.py
42
0x
f1
flag
flag{
flag a
flag a4
flag a41
flag a41b
flag a41be
flag a41bed
flag a41bed6
flag a41bed65
flag a41bed65-
flag a41bed65-a
flag a41bed65-a5
flag a41bed65-a50
flag a41bed65-a50f
flag a41bed65-a50f-
flag a41bed65-a50f-4
flag a41bed65-a50f-41
flag a41bed65-a50f-412
flag a41bed65-a50f-4124
flag a41bed65-a50f-4124-
flag a41bed65-a50f-4124-b
flag a41bed65-a50f-4124-b7
flag a41bed65-a50f-4124-b7b
flag a41bed65-a50f-4124-b7ba-
flag a41bed65-a50f-4124-b7ba-2
flag a41bed65-a50f-4124-b7ba-27
flag a41bed65-a50f-4124-b7ba-276
flag a41bed65-a50f-4124-b7ba-2766
flag a41bed65-a50f-4124-b7ba-2766a
flag a41bed65-a50f-4124-b7ba-2766af
flag a41bed65-a50f-4124-b7ba-2766aff
flag a41bed65-a50f-4124-b7ba-2766aff6
flag a41bed65-a50f-4124-b7ba-2766aff6b
flag a41bed65-a50f-4124-b7ba-2766aff6ba
flag a41bed65-a50f-4124-b7ba-2766aff6baf
flag a41bed65-a50f-4124-b7ba-2766aff6baf2
flag a41bed65-a50f-4124-b7ba-2766aff6baf2
```

C:\Users\Administrator\Desktop>

CSDN @末小心

如该题使用自己编写的脚本代码请详细写出，不允许截图

```
less=
[0x66,0x6D,0x63,0x64,0x7f,0x64,0x32,0x36,0x6a,0x6c,0x3e,0x3d,0x39,0x20,0x6f,0x3a,0x20,0x77,0x3f,0x27,0x25,0x27,0x22,0x3a,0x7a,0x2e,0x78,0x7a,0x31,0x2f,0x29,0x16,0x40,0x44,0x45,0x12,0x47,0x47,0x41,0x1a,0x54]
print(len(less))
sas=""
for i in range(len(less)):
    sas+=chr([i]*4)
print(sas)
```

值: **flag{a41be465-a50f-4124-b7ba-2766aff6baf2}**

02 Crypto2

操作内容:

两组数中e相同, n, c不同, 求出n1与n2的最大公因数即为p, 之后就可以得到q和d, 从而求解m, 该题中的flag为两部分, 要依次求出再拼接。

如该题使用自己编写的脚本代码请详细写出, 不允许截图

```
import gmpy2
import binascii
e = 65537
n1 =
20663949646446771694737024742706480203229077367457341749115493465796673487424103630763356769517513101484061720805193175347622314965242713348516077106899407356643165296924396221
c1 =
205227722495914368659057961032325424942116953769733777228756066789998996904504808092316713464898218780503543805919999359607958884836644739522072985041962038305432084772291621776
n2 =
232608340243766400925368889220411471683877020148149105494697303546888487603792742030887166496096754499362347325287785570417015249812003689963100645844796570420984261643662866701
c2 =
187150099447668151494925606450516263292041140499277072923064810187243234337019702535414950902447873788265695498854804917640575268285314290333781434262722489402564324239399778052
p = gmpy2.gcd(n1,n2)
q1 = n1//p
q2 = n2//p
phi1 = (p-1)*(q1-1)
phi2 = (p-1)*(q2-1)
d1 = gmpy2.invert(e,phi1)
d2 = gmpy2.invert(e,phi2)
m1 = gmpy2.powmod(c1,d1,n1)
m2 = gmpy2.powmod(c2,d2,n2)
print(binascii.unhexlify(hex(m1))[2:])
print(binascii.unhexlify(hex(m2))[2:])
```

值: **flag{afb1e6f2-9acb-efde-ad7c-246a99d8f1fd}**

04 Crypto1

操作内容:

e1=49

e2=35

首先想到了共模攻击, 但是不满足gcd条件(e1, e2互质)想到可以将49,35同时除以7, 然后进行低加密指数攻击即可

如该题使用自己编写的脚本代码请详细写出, 不允许截图

```
import gmpy2
from Crypto.Util.number import long_to_bytes
n=96722669749951212913756678234358651184134068407812470434435916603156818917545841439779031943800634250032106764154804309935557678512858630048212204696471487762160744924838010741
c1=667381132234472214300097399149483032610028115530643075329267880246943198469093408069827083479046884206716564105548523407323958180070636484785930716659362778369880505261880641
c2=883309491466510425173376537408103851873616890125017927999008732799787360357906592110010479373372151219485270170229676429066327321363137502723776191071091545973355142165325991
e1=7
e2=5
def ext_euclid(a,b):
    if b == 0:
        return 1,0,a
    else:
        x,y,q = ext_euclid(b, a % b) #gcd(a,b)=gcd(b,a%b)
        x,y = y, (x-(a//b)*y)
        return x,y,q
def same_mod(n,e1,e2,c1,c2):
    s,t,q = ext_euclid(e1,e2)
    m=(gmpy2.powmod(c1,s,n)*gmpy2.powmod(c2,t,n)) % n # 大数运算
    flag = m
    # 爆破
    i = 0
    while 1:
        if gmpy2.iroot(m + i * n,7)[1]==True:
            print(long_to_bytes(gmpy2.iroot(m+i*n,7)[0]))
            break
        i += 1
    return flag
if __name__ == '__main__':
    same_mod(n,e1,e2,c1,c2)
```

值:

flag{8ac9f9e3-82ba-ff7e-ac7b-235a02d891ef}

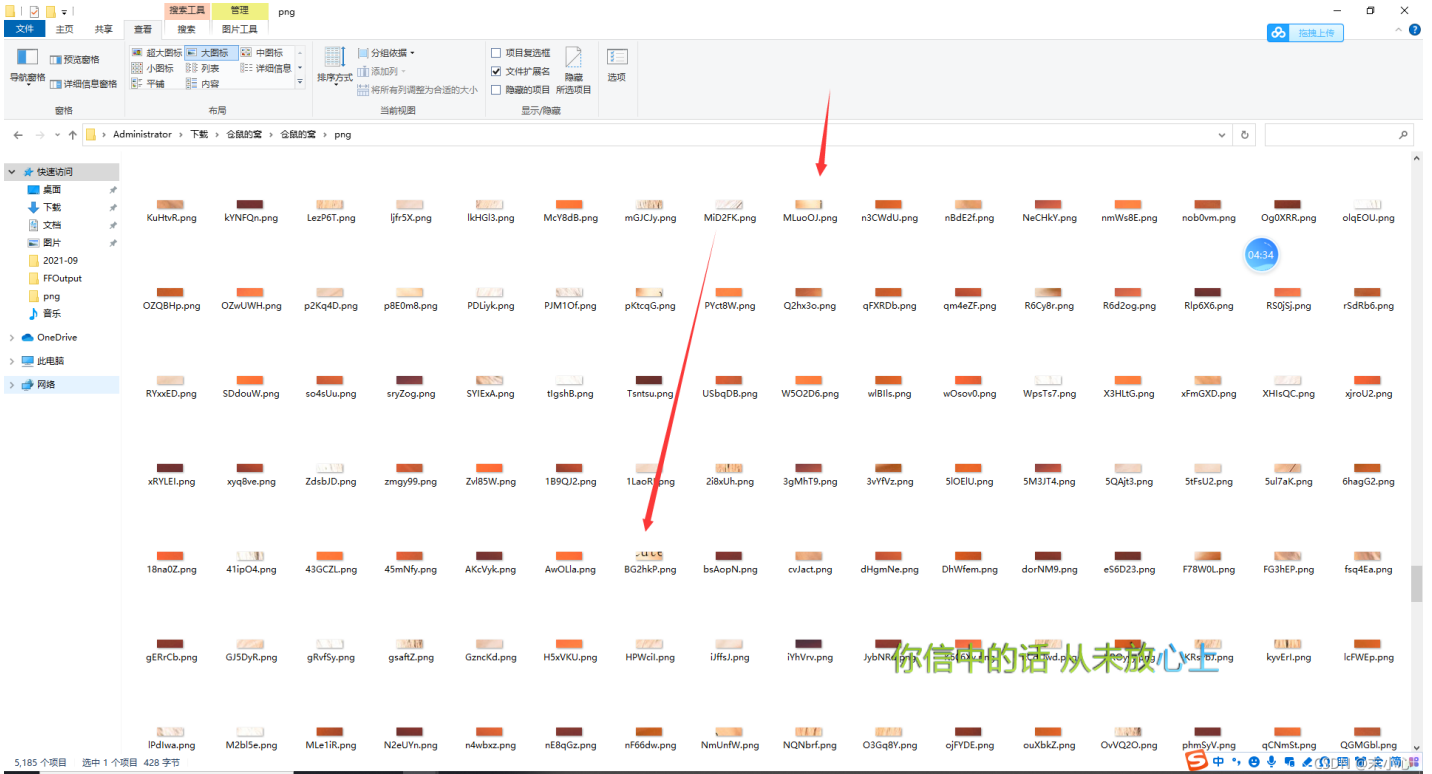
06 拼图

操作内容:

拿到附件先解压, 原图先不用看, 直接看分割后的。

因为分割图片, 所以可以通过时间排列 (分割文件的保存的时间排序。)

可以看到分割的顺序是从左到右依次分割。整理含字符串的图片。



整理出来有如下片段:



能拼成这样子，提交该flag: `flag{Hamsters_are_so_cute}` 都提示不对。



发现了这个东西，可能是个感叹号什么的？还是i呢？我想了一下，按照英语语法，i有点不合适。

可能是感叹号，那就交了试试`flag{Hamsters_are_so_cute!}`发现不对

但是我发现文件中还有很多没有拼接，比如



试试多加几个!吧，提交发现3个!就对了得到flag

flag值:`flag{Hamsters_are_so_cute!!!}`