

小雪 看雪学院 1周前

有套路

今天我们不开玩笑，来认真看看KCTF 第六题《REPWN》不知道这道题有没有让你怀疑人生呢？😏

1/17

出题战队

HaCky_		
战队信息	战队成员(1)	成员动态
成员名称	职位	积分
 findreamwang	队长	200

战队成员：findreamwang

个人主页：<https://bbs.pediy.com/user-739734.htm>

个人简介：HaCky，目前就读于西安某高校网络工程专业，现在主要的研究方向是病毒分析和内核研究(即将要学)，大一开始学习安全技术。第一次参加KCTF，编码方面多有不足，请各位大佬轻打！

看雪CTF crownless 评委 点评

此题使用了pwn题和re（逆向）题相结合的方式，还涉及了DES加密的识别，考察了DES的理解。总的来说，是一道考察选手综合能力的好题目。

题目设计思路

题目信息

参赛题目：Repwn

题目答案：

20101001X1Y0uN3tG00dHaCkWel1C0me

题目设计

本题使用的Pwn题和Re题相结合的方式，考察选手的综合能力。

首先，编码不加任何混淆和反调试，题目友好，在入口处不对字符串设置加密。

```

    correctInfo[1] = 0x2010;
printf("Please Input Your Key_ Now!\n");
scanf("%s", Input);
if(CmpString(Input)==0)           //触发第一次验证函数
{
    puts(WrongInfo);
    return 0;
}

```

其次main流程简单，易于理解，设置两层check

```

if(CmpString(Input)==0)           //触发第一次验证函数
{
    puts(WrongInfo);
    return 0;
}
Jump(Input);                      //触发验证函数*/
system("pause");
return 0;

```

逆向思路简单，难度小。

```

char Cmpstr[]="X1Y0uN3tG00d";
for(i=8,j=0;i<20,j<12;i++,j++)
{
    if(Input[i]!=Cmpstr[j])
    {
        flag=1;
        break;
    }
}
if(flag==1)

```

同时也有计算题，此处不设置麻烦的运算，这样无趣。

```
Transform(Input);    //转化为int型数据
int x=1000*Input_Num[0]+100*Input_Num[1]+10*Input_Num[2]+Input_Num[3]; //2018
int y=10*Input_Num[4]+Input_Num[5];    //10
int z=10*Input_Num[6]+Input_Num[7];    //01    ---->需要添加约束条件
if((2*x+2*y+10)==4050)
{
    if((3*y/2+100*z)==115)
    {
        if((x-110*z)==1900)
            return 1;
        else
            printf("Key_Is_Wrong,Please_Input_Again!");
    }
}
```

最后需要构造溢出点，这样更加贴近实战，写exp等。

```
{
    if(Equation(Input))    //计算方程组
    {
        Input[20]+=0xA8;    // 0x48 H    0xF0
        Input[21]-=0x46;    // 0x61 a    0x1B
        Input[22]-=0x3;    // 0x43,C    0x40
        Input[23]-=0x6B;    // 0x6B k    0x00
        strcpy(Buffer,Input);    //溢出点
    }
}
```

然后是溢出的地方，这里涉及到DES的识别问题，可以人工识别，最好是软件识别。本人菜鸟就不想搞变形的DES，这样难度会挺大的，能力也有限。

```
printf("%.s\n",21,Infor);
char key[] = "XiyouNet";    //密钥
fflush(stdin);
char *str=new char[8];;    //明文 WellC0me
gets(str);    //输入明文
SetKey(key);
Des_Run(str, str);
return 0;
```

```
//S5
2,12,4,1,7,10,11,6,8,5,3,15,13,0,14,9,
14,11,2,12,4,7,13,1,5,0,15,10,3,9,8,6,
4,2,1,11,10,13,7,8,15,9,12,5,6,3,0,14,
11,8,12,7,1,14,2,13,6,15,0,9,10,4,5,3,
//S6
12,1,10,15,9,2,6,8,0,13,3,4,14,7,5,11,
10,15,4,2,7,12,9,5,6,1,13,14,0,11,3,8,
9,14,15,5,2,8,12,3,7,0,4,10,1,13,11,6,
4,3,2,12,9,5,15,10,11,14,1,7,6,0,8,13,
//S7
4,11,2,14,15,0,8,13,3,12,9,7,5,10,6,1,
13,0,11,7,4,9,1,10,14,3,5,12,2,15,8,6,
1,4,11,13,12,3,7,14,10,15,6,8,0,5,9,2,
6,11,13,8,1,4,10,7,9,5,0,15,14,2,3,12,
//S8
```

另外，对关键的字符串进行加密防止直接找到入口点，例如恺撒，异或。

```
0xa,0x2,0x8,0x12,0x34,0x56,0x78,0x90}; //数据
char CurrentInfo[]="AwmjGQvhmlGqkG^ty 4~ty G~wjuyLGqkG^ty cs}a3Qvhmle";
for(int i=0;i<strlen(CurrentInfo);i++)
    CurrentInfo[i]^=0x2018;
for(i=0;i<16;i++)
    0x53};
for(int j=0;j<strlen(Infor)-10;j++) //提示转化
    Infor[j]=Infor[j]+j;
system("cls");
printf("%s *s\n" 21 Infor);
```

同时也考察了DES基础理解，因为有个比较cmphex是40个，但是des只有32个字节。哈哈，小心机一把。

```
int i,flag=0;
unsigned int Hex[32]={0}; //存放十六进制数据的
unsigned int CmpHex[37]={0x0,0xc,0xf,0xd,0xc,0xc,0x0,0x4,0x4,
0x7,0xf,0xc,0xe,0x3,0x9,0x7,0x5,0x6,0x9,
0x7,0xd,0x5,0xa,0x4,0x9,0x9,0x3,0x5,0x5,
0xa,0x2,0x8,0x12,0x34,0x56,0x78,0x90};
```

格式为第一次输入+第二次输入

破解思路

0x01 查壳

不存在任何壳。

0x02

拖入IDA中看到，定位到主函数，可以看到程序经过了两层验证check1和check。

```
v0 = 0;
v3 = ' yt^';
v4 = '+pLc';
v5 = 'a+SG';
v6 = 'G-QG';
v7 = 'G1(V';
v8 = 'y}J';
v9 = 'SGA)';
v10 = 'ea+';
strcpy(Str, "Ansome_Is_Wrong");
while ( v0 < strlen((const char *)&v3) )
    *((_BYTE *)&v3 + v0++) ^= 0x18u;
puts("Please Input Your Key_ Now!");
scanf("%s", &Input);
if ( Check1(&Input) )
{
    Check2(&Input);
    system("pause");
}
else
{
    puts(Str);
}
return 0;
```

0x03 过check1

得到大概key为xxxxxxxx(8位)X1Y0uN3tG00dxxx(x为未知数据，长度未知)


```

v1 = 8;
v2 = 0;
v8 = 'ruoY';
v9 = 'pnI_';
v10 = 'I_tu';
v11 = 'rW_s';
v12 = 'gno';
v4 = '0Y1X';
v5 = 't3Nu';
v6 = 'd00G';
v7 = 0;
while ( *((_BYTE *)&v4 + v2) == *((_BYTE *)v1 + Input) )// 从第八位开始比较
{
    ++v2;
    ++v1;
    if ( v2 > 11 )
        return 1;
}
return 0;

puts( "Please input your key now: ",
scanf("%s", &Input);
if ( Ckeck1((int)&Input) )
{
    // Your_Input_Is_Wrong
    // X1Y0uN3tG00d
    // 比较11个长度，正好是X1Y0uN3tG00d
    // 得到Key:xxxxxxx(8位)X1Y0uN3tG00dxxx
}

```

0x04 过check2

首先长度为24，进入check2_1

```

char Dest; // [esp+8h] [ebp-10h]

if ( strlen(input) == 24 )
{
    if ( Check2_1((int)input) )
    {
        input[20] -= 19;
        input[21] -= 8;
        input[23] -= 47;
        strcpy(&Dest, input);
    }
}
else
{
    printf("String Length is Wrong");
}
return 0;

```

Check2_1中

首先将输入转化为Input，然后初始化数字X,Y,Z，我们可以知道x, y, z的生成关系如下：x是input的前四个构成一个4位数，y是input[4-5]构成的2位数，z是input[6-7]构成的2位数，然后解方程，很简单。

得到Input为20101001X1Y0uN3tG00dXXX

```
// x=1000*Input_Num[0]+100*Input_Num[1]+10*Input_Num[2]+Input_Num[3]
// y=10*Input_Num[4]+Input_Num[5]
// z=10*Input_Num[6]+Input_Num[7]

result = 2 * (x + y);
if ( result == 4040 )

{
    result = 3 * y / 2;
    if ( result + 100 * z == 115 )
    {
        result = 1;
        if ( x - 110 * z != 1900 )
            result = printf("Key_Is_Wrong,Please_Input_Again!");
    }
}
```

00007BE Check2_1:24 (4013BE)

然后到达溢出点，目标是寻找溢出点，这个有点坑，方法不唯一，我的想法是这样的：首先查壳的时候发现有DES加密函数，一直交叉引用到溢出点：401C1



现在的问题是如何达到溢出点。00401C10，又由于小端序的问题，顺序应该是101C4000，也就是说input[23]-47=00，也就是说input[23]为47的ASCII，同理，溢出点设计为#\$/，Key为20101001X1Y0uN3tG00d#\$/

```

{
    input[20] -= 19;
    input[21] -= 8;
    input[23] -= 47;
    strcpy(&Dest, input);
}
// 构造溢出
// INput[20],[21]和[23]为溢出点

```

0x5 溢出后

之前，我们知道了DES加密。得到加密过程如下

```

printf("%.6s\n", 21, &v6);
v3 = 'oyiX';
v4 = 'teNu';
v5 = 0;
fflush(iob);
Input2 = (char *)sub_401D50(8u);
gets(Input2);
sub_401730((int)&v3);
sub_4018B0((int)Input2, (int)Input2);
return 0;
}
{
dword_4085F0 = *((_DWORD *)off_40600C);
dword_4085F4 = *((_DWORD *)off_40600C + 1);
dword_4085F8 = *((_DWORD *)off_40600C + 2);
dword_4085FC = *((_DWORD *)off_40600C + 3);
dword_408600 = *((_DWORD *)off_40600C + 4);
dword_408604 = *((_DWORD *)off_40600C + 5);
dword_408608 = *((_DWORD *)off_40600C + 6);
dword_40860C = *((_DWORD *)off_40600C + 7);
v4 = (int)v2;
v2 += 48;
sub_4017D0((int)off_40600C, v4);
sub_401690(32, off_406008);
--v3;
v5 = off_406008;
*(_DWORD *)off_406008 = dword_4085F0;
*(_DWORD *)v5 + 1 = dword_4085F4;
*(_DWORD *)v5 + 2 = dword_4085F8;
*(_DWORD *)v5 + 3 = dword_4085FC;
*(_DWORD *)v5 + 4 = dword_408600;
*(_DWORD *)v5 + 5 = dword_408604;
*(_DWORD *)v5 + 6 = dword_408608;
*(_DWORD *)v5 + 7 = dword_40860C;
}
while ( v3 >= 0 );

```

末尾有个比较

```

v30 = 8;
do
{
    if ( Dst[v8] != CmpHex[v8] )
        flag = 1;
    ++v8;
}

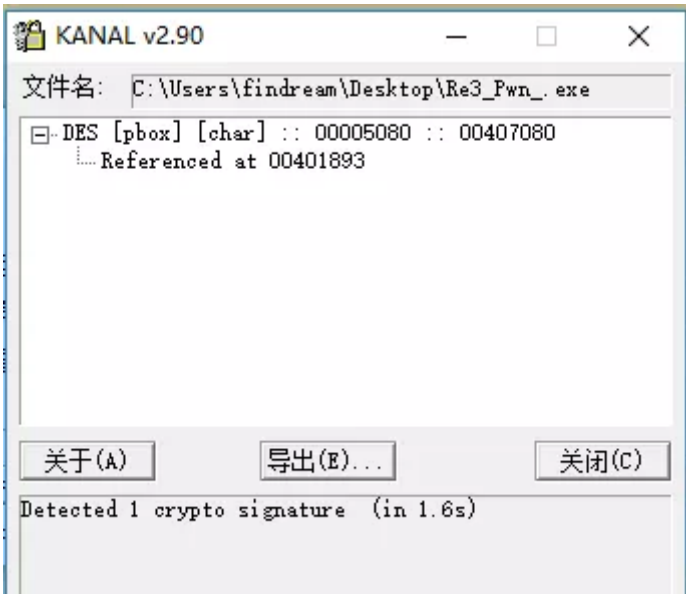
```

引用查看一下：unk_406010，整理出来是40位：但是我们知道DES加密可能会生成32位，不可能生成40位的，所以我们去其32位。密钥为XiyouNet，DES解密得到：Wel1C0m

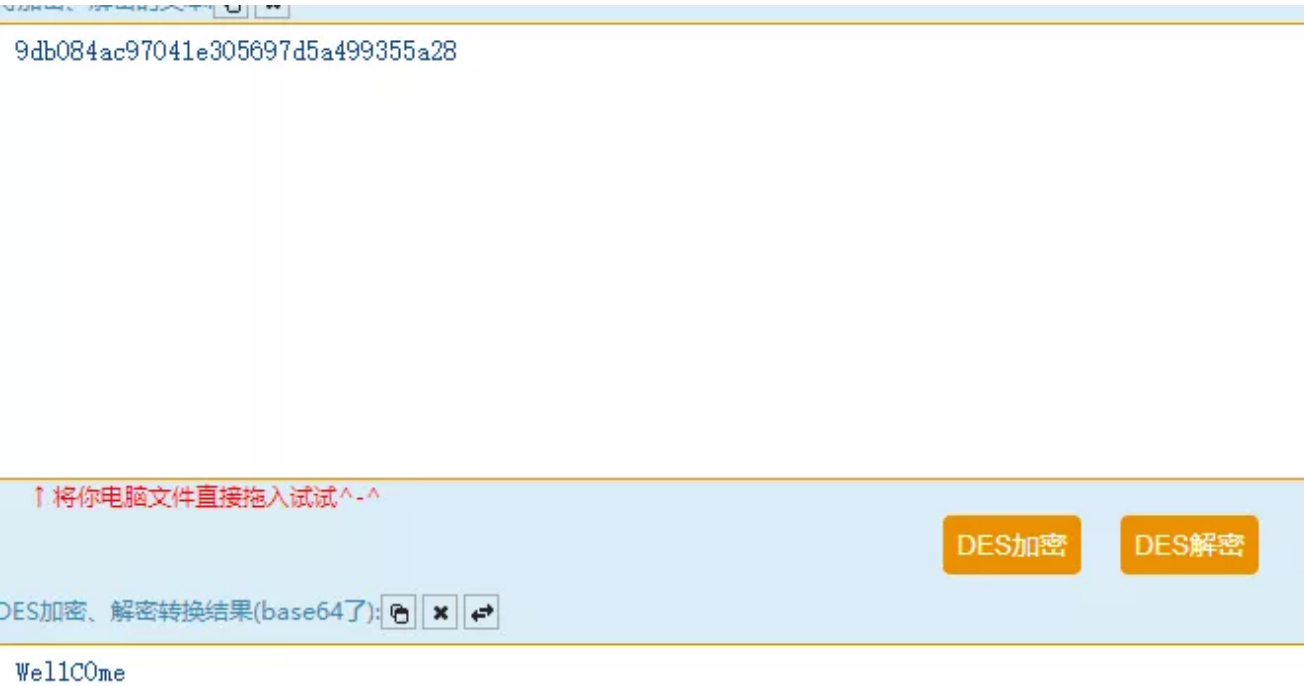
```

flag = 0;
memset(Dst, 0, 0x80u);
memcpy(CmpHex, &unk_406010, 0x94u);
qmemcpy(Format, &unk_40740C, sizeof(Format));
v6 = 0;
v12 = 101;
while ( v6 < strlen(Format) )
{

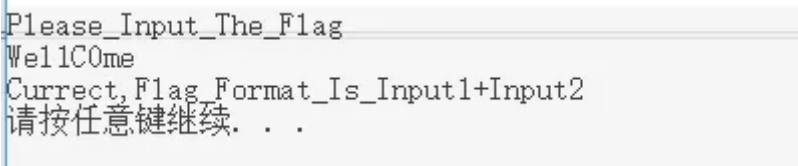
```



解密



最后也提示了flag格式，所以flag为20101001X1Y0uN3tG00dHaCkWel1C0m



解题思路

本题解题思路由看雪论坛 **scpczc** 提供



发消息

scpczc

中级 ★★★

精华数: 1

RANK: 80

雪币: 2116 商城

浏览人数: 88

在线时长: 🕒🕒🕒

注册时间: 2008-08-28

最近活跃: 42分钟前

1、运行

Please Input Your Key_ Now!

123456789

Ansome_Is_Wrong

2、IDA载入，快速定位

可以看到获取输入，对输入判断后，错误输出错误信息，否则进入另外的流程。

```
int sub_4014C0()
{
    unsigned int v0; // ebx
    char Str[4]; // [esp+10h] [ebp-58h]
    CHAR v3[32]; // [esp+20h] [ebp-48h]
    char a1; // [esp+40h] [ebp-28h]

    sub_404930();
    sub_4044B0();
    v0 = 0;
    strcpy(v3, "^ty\x7FcLp+GS+aGQ-GV(1GJ}y))AGS+ae");
    strcpy(Str, "Ansome_Is_Wrong");
    while ( v0 < strlen(v3) )
    {
        v3[v0] ^= 0x18u;
        ++v0;
    }
}
```

```
puts("Please Input Your Key_ Now!");
scanf("%s", &a1);
if ( sub_4012F0(&a1) )
{
sub_401460(&a1);
system("pause");
}
else
{
puts(Str);
}
return 0;
}
```

3、先用X64DBG修改跳转爆破试试。

输出长度错误，说明sub_401460内部还有判断。

004015B1 74 14 je repwn.4015C7 75->74

4、进入第一个判断函数

发现第9位开始后的12位为X1Y0uN3tG00d。第21位为H。

```
signed int __cdecl sub_4012F0(char *a1)
{
signed int v1; // ecx
signed int v2; // edx
signed int result; // eax
char v4[12]; // [esp+0h] [ebp-38h]
CHAR v5[20]; // [esp+10h] [ebp-28h]

v1 = 8;
v2 = 0;
strcpy(v5, "Your_Input_Is_Wrong");
strcpy(v4, "X1Y0uN3tG00d");
while ( v4[v2] == a1[v1] )
{
++v2;
++v1;
}
```

```
if ( v2 > 11 )
{
result = 1;
if ( a1[20] == 'H' )
return result;
return 0;
}
}
return 0;
}
```

5、进入另一个判断函数

发现长度必须为24.后四位分别减去 88, 70, 3, 'k', 另有玄机, 暂不表。

```
int __cdecl sub_401460(char *Str)
{
char Dest; // [esp+8h] [ebp-10h]

if ( strlen(Str) == 24 )
{
if ( sub_4013B0((int)Str) )
{
Str[20] -= 88;
Str[21] -= 70;
Str[22] -= 3;
Str[23] -= 'k';
strcpy(&Dest, Str);
}
}
else
{
printf("String Length is Wrong");
}
return 0;
}
```

6、进入sub_4013B0

前8位需要解方程。这是为了保护小学生不搞破解？

$2(a+b) = 4040$
 $3b/2 + 100c = 115$
 $a - 110c = 1900$
 $a = 2010$
 $b = 10$
 $c = 1$

所以前8位为20101001。

```

signed int __cdecl sub_4013B0(int a1)
{
    int v1; // ebx
    int v2; // ecx
    int v3; // esi
    signed int result; // eax

    sub_401380(a1);
    v1 = dword_40802C + 1000 * dword_408020 + 100 * dword_408024 + 10 * dword_408028;
    v2 = dword_408034 + 10 * dword_408030;
    v3 = dword_40803C + 10 * dword_408038;
    if ( 2 * (v1 + v2) != 4040 || 3 * v2 / 2 + 100 * v3 != 115 )
        goto LABEL_2;
    result = 1;
    if ( v1 - 110 * v3 != 1900 )
    {
        printf("Key_Is_Wrong, Please_Input_Again!");
    LABEL_2:
        result = 0;
    }
    return result;
}

```

20101001X1Y0uN3tG00dH? ? ?

随机输入一个，发现程序异常了，多次调试发现在sub_401460返回时错误。Strcpy会溢出，覆盖掉原来的地址。

要想不异常，返回的地址必须在程序空间内，那么大致为004XXXXX，第一个X最大可能为0，而这个值加上 B6 03 46 58，就是最后4位。那么结果就H?Ck。‘H’-58为F0，就是说返回地址为0040?? F0.检测测试发现输入HaCk正确。

所以结果为：20101001X1Y0uN3tG00dHaCk

结果显示：Success_Please_Input_The_Flag

请按任意键继续...

看雪CTF晋级赛Q1 题解列表

- 1、2019KCTF 晋级赛Q1 | 第一题点评及解题思路
- 2、2019KCTF 晋级赛Q1 | 第二题点评及解题思路
- 3、2019 KCTF 晋级赛Q1 | 第三题点评及解题思路
- 4、2019KCTF 晋级赛Q1 | 第四题点评及解题思路
- 5、2019 KCTF 晋级赛Q1 | 第五题点评及解题思路




- End -



公众号ID: ikanxue

官方微博: 看雪安全

 戳原文，查看更多精彩writeup!

阅读原文