

XCTF_BABYRE

原创

永远是深夜有多好。 于 2022-01-17 21:46:14 发布 178 收藏

分类专栏: [XCTF](#) 文章标签: [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/qq_37370714/article/details/122547694

版权



[XCTF 专栏收录该内容](#)

17 篇文章 0 订阅

订阅专栏

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
    char s[24]; // [rsp+0h] [rbp-20h] BYREF
    int v5; // [rsp+18h] [rbp-8h]
    int i; // [rsp+1Ch] [rbp-4h]

    for ( i = 0; i <= 181; ++i )
    {
        judge[i] ^= 0xCu;
        printf("Please input flag:");
        __isoc99_scanf("%20s", s);
        v5 = strlen(s);
        if ( v5 == 14 && (*(unsigned int (__fastcall **)(char *))judge)(s) )
            puts("Right!");
        else
            puts("Wrong!");
    }
    return 0;
}
```

打开发现看不懂也不知道 `judge` 是个啥, 只能我自己会的方法来做试试。

在循环后面下个断点

```
7   for ( i = 0; i <= 181; ++i )
8       judge[i] ^= 0xCu;
9       printf("Please input flag:");
```

动态调试后发现 `judge` 值改变了

```
.data:0000000000600B00 ; DATA XREF: main+16↑r ...
.data:0000000000600B01 db 48h ; H
.data:0000000000600B02 db 89h
.data:0000000000600B03 db 0E5h
.data:0000000000600B04 db 48h ; H
.data:0000000000600B05 db 89h
.data:0000000000600B06 db 7Dh ; }
```

```

.data:0000000000600B07 db 0D8h
.data:0000000000600B08 db 0C6h
.data:0000000000600B09 db 45h ; E
.data:0000000000600B0A db 0E0h
.data:0000000000600B0B db 66h ; f
.data:0000000000600B0C db 0C6h
.data:0000000000600B0D db 45h ; E
.data:0000000000600B0E db 0E1h
.data:0000000000600B0F db 6Dh ; m

```

00000B04|0000000000600B04: .data:0000000000600B04 (Synchronized with RIP) CSDN @永远是深夜有多好。

现在需要弄清楚这个 `judge` 是什么
 想了一会估计是个函数被怎么处理过就成了数据
 重新生成汇编代码看看(按c)

```

.data:0000000000600B00
.data:0000000000600B00
.data:0000000000600B00 public judge
.data:0000000000600B00 judge: ; CODE XREF: main+80↑p
.data:0000000000600B00 ; DATA XREF: main+16↑r ...
.data:0000000000600B00 push rbp
.data:0000000000600B01 mov rbp, rsp
.data:0000000000600B04 mov [rbp-28h], rdi
.data:0000000000600B08 mov byte ptr [rbp-20h], 66h ; 'f'
.data:0000000000600B0C mov byte ptr [rbp-1Fh], 6Dh ; 'm'
.data:0000000000600B10 mov byte ptr [rbp-1Eh], 63h ; 'c'
.data:0000000000600B14 mov byte ptr [rbp-1Dh], 64h ; 'd'
.data:0000000000600B18 mov byte ptr [rbp-1Ch], 7Fh
.data:0000000000600B1C mov byte ptr [rbp-1Bh], 68h ; 'k'
.data:0000000000600B20 mov byte ptr [rbp-1Ah], 37h ; '7'
.data:0000000000600B24 mov byte ptr [rbp-19h], 64h ; 'd'

```

00000B18|0000000000600B18: .data:0000000000600B18 (Synchronized with RIP) CSDN @永远是深夜有多好。

哎有东西的样子
 鼠标放到 `public judge` 在这里插入代码片 e 后面按 p 重新生成函数

```

a:0000000000600AFE db 0
a:0000000000600AFF db 0
a:0000000000600B00 ; -----
a:0000000000600B00
a:0000000000600B00 public judge
a:0000000000600B00 judge:
a:0000000000600B00

```

重新生成c代码发现这应该是

```

2 {
3 char v2[5]; // [rsp+8h] [rbp-20h] BYREF
4 char v3[9]; // [rsp+Dh] [rbp-1Bh] BYREF
5 int i; // [rsp+24h] [rbp-4h]
6
7 memcpy(v2, "fmcd", 4);
8 v2[4] = 127;
9 memcpy(v3, "k7d;V`;np", sizeof(v3));
10 for ( i = 0; i <= 13; ++i )
11     *(_BYTE*)(i + a1) ^= i;

```

```

12 for ( i = 0; i <= 13; ++i )
13 {
14     if ( *(_BYTE *)(i + a1) != v2[i] )
15         return 0LL;
16 }
17 return 1LL;

```

CSDN @永远是深夜有多好。

```

#include <stdio.h>
#include <string.h>
#pragma warning(disable:4996)
int main()
{
    char v2[15]= { 0 };
    char flag[15] = { 0 };
    strcat(v2, "fmcD");
    v2[4] = 127;
    strcat(v2, "k7d;V` ;np");
    //printf("%d\n", sizeof(v2)); v2大小为15
    for (int i = 0; i <= 13; ++i)
        flag[i]=v2[i]^i;
    printf("%s", flag);
}
flag{n1c3_j0b}

```

Microsoft Visual Studio 调试控制台

CSDN @永远是深夜有多好。

后面百度和看别人的wp发现这道题相关知识点是SMC(Self-Modifying-Code)代码自修改，是一种反调试手法。

新知识get



[创作打卡挑战赛](#) >

[赢取流量/现金/CSDN周边激励大奖](#)