

BUUCTF Reverse [GXYCTF2019]luck_guy WriteUp

原创

PlumpBoy 于 2021-09-11 16:13:32 发布 48 收藏

分类专栏: [BUUCTF 逆向题解](#) 文章标签: [安全](#) [系统安全](#)

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/weixin_45723661/article/details/120227985

版权



[BUUCTF 逆向题解 专栏收录该内容](#)

18 篇文章 1 订阅

订阅专栏

luck_guy-WP

首先用hxd打开看一下, 发现是elf文件, 直接拖入ida64中F5反编译。

```
IDA View-A Pseudocode-A Hex View-1
1 int __cdecl main(int argc, const char **argv, const char
2 {
3     int v4; // [rsp+14h] [rbp-Ch] BYREF
4     unsigned __int64 v5; // [rsp+18h] [rbp-8h]
5
6     v5 = __readfsqword(0x28u);
7     welcome(argc, argv, envp);
8     puts("_____");
9     puts("try to patch me and find flag");
10    v4 = 0;
11    puts("please input a lucky number");
12    __isoc99_scanf("%d", &v4);
13    patch_me(v4);
14    puts("OK,see you again");
15    return 0;
16 }
```

CSDN @PlumpBoy

进入patch_me

```
IDA View-A Pseudocode-A Hex View-1
1 int __fastcall patch_me(int a1)
2 {
3     int result; // eax
4
5     if ( a1 % 2 == 1 )
6         result = puts("just finished");
7     else
8         result = get_flag();
9     return result;
10 }
```

CSDN @PlumpBoy

进入get_flag

```

unsigned __int64 get_flag()
{
    unsigned int v0; // eax
    int i; // [rsp+4h] [rbp-3Ch]
    int j; // [rsp+8h] [rbp-38h]
    __int64 s; // [rsp+10h] [rbp-30h] BYREF
    char v5; // [rsp+18h] [rbp-28h]
    unsigned __int64 v6; // [rsp+38h] [rbp-8h]

    v6 = __readfsqword(0x28u);
    v0 = time((time_t *)'\0');
    srand(v0);
    for ( i = 0; i <= 4; ++i )
    {
        switch ( rand() % 200 )
        {
            case 1:
                puts("OK, it's flag:");
                memset(&s, 0, 0x28uLL);
                strcat((char *)&s, f1); //f1=GXY{do_not_
                strcat((char *)&s, &f2);
                printf("%s", (const char *)&s);
                break;
            case 2:
                printf("Solar not like you");
                break;
            case 3:
                printf("Solar want a girlfriend");
                break;
            case 4:
                s = 0x7F666F6067756369LL; //注意，在脚本中要反过来，因为是小端存储
                v5 = 0;
                strcat(&f2, (const char *)&s);
                break;
            case 5:
                for ( j = 0; j <= 7; ++j )
                {
                    if ( j % 2 == 1 )
                        *(&f2 + j) -= 2;
                    else
                        --*(&f2 + j);
                }
                break;
            default:
                puts("emmm,you can't find flag 23333");
                break;
        }
    }
    return __readfsqword(0x28u) ^ v6;
}

```

查看后发现，有用的case只有1，4，5，而且其中1要使用f2，4初始化了f2，5修改了f2，故正确的调用顺序应该是4，5，1。

写一个逆向脚本即可

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

int main()
{
    char s2[] = { 0x69,0x63,0x75,0x67,0x60,0x6f,0x66,0x7f,0 };//0x7F 66 6F 60 67 75 63 69 LL;//注意，在脚本中要反过来，因为是小端存储
    char flag[20];
    char s1[] = "GXY{do_not_";

    for (int j = 0; j <= 7; ++j)
    {
        if (j % 2 == 1)
            s2[j] -= 2;
        else
            s2[j]-=1;
    }

    memset(flag, 0, 20);
    strcat_s(flag, s1);
    strcat_s(flag, s2);
    puts(flag);
    return 0;
}
```

结果为 `GXY{do_not_hate_me}`，但是，提交到平台上的正确内容是 `flag{do_not_hate_me}`，应该是答案没设置好的原因。