攻防世界-reverse-[open-source] writeup

原创

望邹 于 2020-09-14 11:49:38 发布 145 收藏

分类专栏： CTF攻防世界逆向 文章标签： 安全 经验分享 其他

本文链接：https://blog.csdn.net/ZHiLuan1/article/details/108576351

攻防世界-reverse-open-source

攻防世界-reverse-open-source

1、查看题目

2、打开.C文件

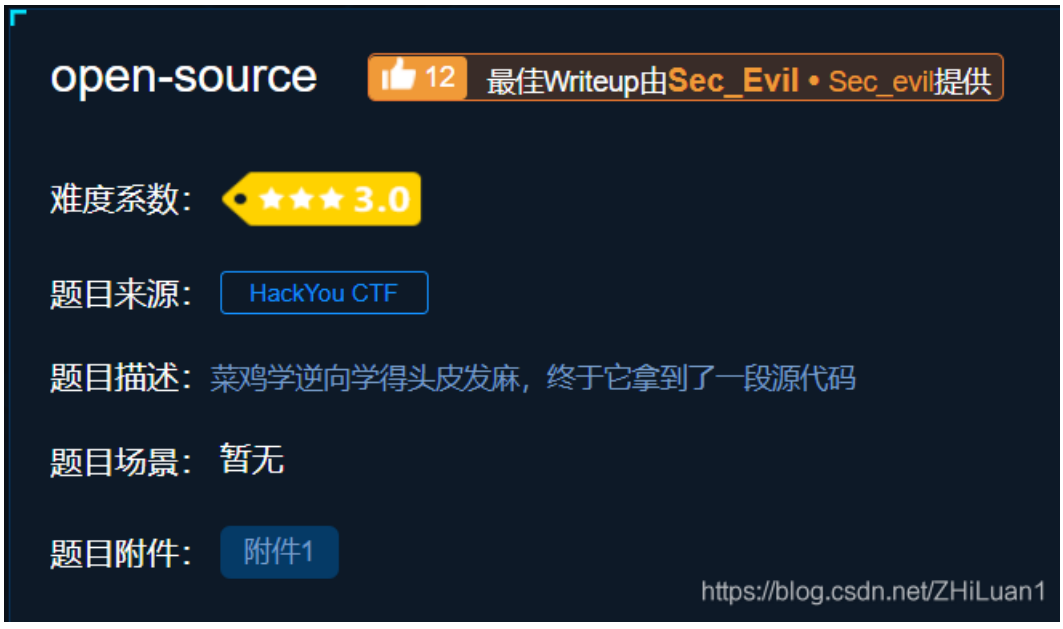
3、分析.C文件

4、计算hash

5、得出结果

攻防世界-reverse-open-source

1、查看题目



题目描述：菜鸡学逆向学得头皮发麻，终于它拿到了一段源代码。我们点击附件1下载。

2、打开.C文件

下载的附件为.c文件，可以直接打开或以.txt(记事本)打开：

8b6405c25fe447fa804c6833a0d72808.c - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

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

int main(int argc, char *argv[]) {
    if (argc != 4) {
        printf("what?\n");
        exit(1);
    }

    unsigned int first = atoi(argv[1]);
    if (first != 0xcafe) {
        printf("you are wrong, sorry.\n");
        exit(2);
    }

    unsigned int second = atoi(argv[2]);
    if (second % 5 == 3 || second % 17 != 8) {
        printf("ha, you won't get it!\n");
        exit(3);
    }

    if (strcmp("h4cky0u", argv[3])) {
        printf("so close, dude!\n");
        exit(4);
    }

    printf("Brr wrrr grr\n");

    unsigned int hash = first * 31337 + (second % 17) * 11 + strlen(argv[3]

    printf("Get your key: ");
    printf("%x\n", hash);
    return 0;
}
```


3、分析.C文件

大概浏览一遍，不难看出，flag应该就是最后printf函数中的hash。为了计算出hash的值，我们必须找出“first”、“second”、“strlen(argv[[3]])”这三个参数，由此我们继续从头开始浏览代码。

第一个参数：argc=4;由条件语句`if(argc!=4)`后`printf`的`what?`,可以猜出。
第二个参数：`first=0xcafe;`由第一个参数同理可得。
第三个参数：`second=25;`由上同理，if语句里的判断不能为真，所以second%5不等于3，且second%17等于8；综合分析，可取`second=25;`
第四个参数：

```
argv[[3]]="h4cky0u"
```

4、计算hash

参数备齐，可以开始计算hash的值了：

```python
first = 0xcafe      #参数一
second = 25         #参数二
argv = 'h4cky0u'    #参数三
# len(h4cky0u) = 7

hash = first * 31337 + (second % 17) * 11 + 7 - 1615810207
print(hex(hash))
```


5、得出结果

最后，得出结果：

```
0xc0ffee
```

原.c代码中，输出时的格式为%x,因此此处用了hex(hash)，直接输出16进制格式，即为最终结果。

flag为：flag{c0ffee}。

我会持续更新攻防世界逆向的题目，请关注我，点个收藏，谢谢！
或者我有什么不足的地方，可以留言，我会即使改正，谢谢！