

XCTF-Reverse-ExerciseArea-004-writeup

原创

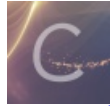
y4ung 于 2019-07-29 20:16:30 发布 3928 收藏

分类专栏: [ctf](#) 文章标签: [CTF ReverseEngineering](#)

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_35056292/article/details/97676105

版权



[ctf](#) 专栏收录该内容

35 篇文章 0 订阅

订阅专栏

0x00 介绍

本题是xctf攻防世界中Reverse的新手第四题。

这道题只给了code.c文件, 我们需要分析该文件中代码的流程, 解出flag。本题考查的主要是C语言的库函数

实验环境: Vscode

0x01 解题过程

直接打开C文件读代码即可

首先是一开始的参数数量判断, 这里需要注意的是运行的时候, `code.exe param1 param2 param3` 中 `code.exe`为参数1。因此用户输入的参数数量应该为3个

第一个参数经过 `atoi()` 函数后是否与 `0xcafe` 相等

`atoi` (表示 `ascii to integer`)是把字符串转换成整型数的一个函数, 应用在计算机程序和办公软件中。`int atoi(const char *nptr)` 函数会扫描参数 `nptr`字符串, 会跳过前面的空白字符(例如空格, `tab`缩进)等。如果 `nptr`不能转换成 `int` 或者 `nptr`为空字符串, 那么将返回 `0`。`atoi`我理解的是把数字形式的字符串转换成`int`类型, 比如"123.45" -> 123

因此, 输入的第三个参数必须为数字。把十六进制`0xcafe`转成十进制, 为`51966`, 当然也可以`51966.xxx`

```
unsigned int first = atoi(argv[1]); // 数字形式的字符串转int
if (first != 0xcafe) { // 十进制为: 51966
    printf("you are wrong, sorry.\n");
    exit(2);
}
```

测试通过:

```
1 #include <stdio.h>
2 #include <string.h>
3
4 int main(int argc, char *argv[]) {
5     // if (argc != 4) {
6     //     printf("what?\n");
7     //     exit(1);
8     // }
9
10    unsigned int first = atoi(argv[1]); // 字符串转int
11    if (first != 0xcafe) { // 51966
12        printf("you are wrong, sorry.\n");
13        exit(2);
14    }
15
16    printf("Get first!"); // 添加调试信息
17
18    // unsigned int second = atoi(argv[2]);
19    // if (second % 5 == 3 || second % 17 != 8) {
20    //     printf("ha, you won't get it!\n");
21    //     exit(3);
22    // }
23
24    // if (strcmp("h4cky0u", argv[3])) {
25    //     printf("so close, dude!\n");
26    //     exit(4);
27    // }
28
29    // printf("Brr wrrr grr\n");
```

```
cmd
振洋@LENOVO-HZY D:\ctf-learning\reverse-engineering\xctf\ExerciseArea\004
$ code.exe 51966
Get first!
```

Tips: 可以先把判断参数数量和后面的参数判断的内容注释掉, 添加参数1判断成功的提示信息

3. 第二个参数经过 `atoi()` 函数后得到的整数必须同时都不满足以下两个条件

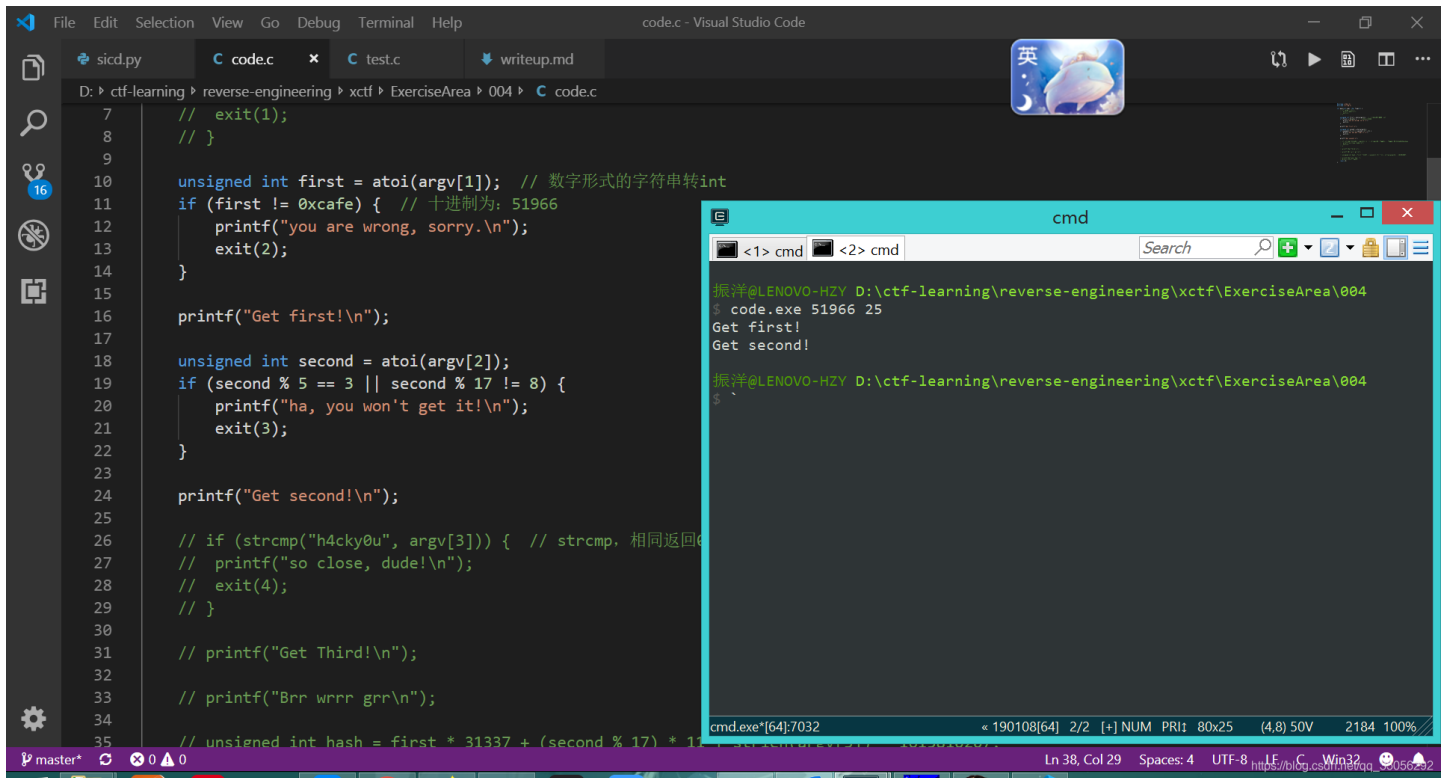
- 条件1: `second % 5 == 3`
- 条件2: `second % 17 != 8`

先从条件2开始看, $second = 17 * n + 8$ ($n=0,1,2,3\dots$), 即 `second = 8, 25, 42, ...`

结合条件1, `second`取24即可

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

测试通过:



The screenshot shows a Visual Studio Code editor with a C program named `code.c` and a terminal window. The C program is as follows:

```
7 // exit(1);
8 // }
9
10 unsigned int first = atoi(argv[1]); // 数字形式的字符串转int
11 if (first != 0xcafe) { // 十进制为: 51966
12     printf("you are wrong, sorry.\n");
13     exit(2);
14 }
15
16 printf("Get first!\n");
17
18 unsigned int second = atoi(argv[2]);
19 if (second % 5 == 3 || second % 17 != 8) {
20     printf("ha, you won't get it!\n");
21     exit(3);
22 }
23
24 printf("Get second!\n");
25
26 // if (strcmp("h4cky0u", argv[3])) { // strcmp, 相同返回0
27 //     printf("so close, dude!\n");
28 //     exit(4);
29 // }
30
31 // printf("Get Third!\n");
32
33 // printf("Brr wrrr grr\n");
34
35 // unsigned int hash = first * 31337 + (second % 17) * 17
```

The terminal window shows the execution of `code.exe` with arguments `51966 25`. The output is:

```
振洋@LENOVO-HZY D:\ctf-learning\reverse-engineering\xctf\ExerciseArea\004
$ code.exe 51966 25
Get first!
Get second!
```

4. 考察`strcmp`函数的返回值, 两个比较的字符串相同时返回0, 不同时返回正数。

if的主体中, 会调用`exit()`函数退出, 因此if的条件判断结果必须为0, 也就是参数3必须为"4cky0u"

```
if (strcmp("h4cky0u", argv[3])) { // strcmp, 相同返回0, 不同返回正数。
    printf("so close, dude!\n");
    exit(4);
}
```

5. 结合上面三个, 拿到flag: c0ffee

```
cmd
<1> cmd <2> cmd Search
Get second!
so close, dude!

振洋@LENOVO-HZY D:\ctf-learning\reverse-engineering\xctf\ExerciseArea\004
$ code.exe 51966 25 hackyous
Get first!
Get second!
so close, dude!

振洋@LENOVO-HZY D:\ctf-learning\reverse-engineering\xctf\ExerciseArea\004
$ code.exe 51966 25 hackyousadsad
Get first!
Get second!
so close, dude!

振洋@LENOVO-HZY D:\ctf-learning\reverse-engineering\xctf\ExerciseArea\004
$ code.exe 51966 25 h4cky0u
Get first!
Get second!
Get Third!
Brr wrrr grr
Get your key: c0ffee

振洋@LENOVO-HZY D:\ctf-learning\reverse-engineering\xctf\ExerciseArea\004
$ |
cmd.exe*[64]:7032 « 190108[64] 2/2 [+] NUM PRI↑ 80x25 (3.34) 50V 2184 100% http://blog.csdn.net/q_3606292
```