

攻防世界 逆向 open-source

原创

与日平肩以头抢地 于 2020-01-29 10:22:05 发布 2534 收藏 1

文章标签: [python c语言](#)

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/fool_best/article/details/104104789

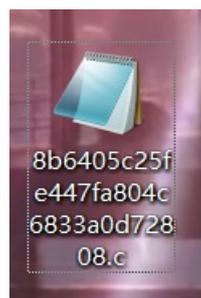
版权

攻防世界 逆向 open-source

原创

The screenshot shows a CTF challenge interface with a dark theme. At the top left, the challenge name 'open-source' is displayed. To its right, there is a '最佳Writeup' (Best Writeup) badge by 'Sec_Evil' and 'Sec_evil'. On the far right, there are buttons for 'WP' and '建议' (Suggest). Below the title, the '难度系数' (Difficulty Coefficient) is shown as '★★ 2.0'. The '题目来源' (Problem Source) is 'HackYou CTF'. The '题目描述' (Problem Description) reads: '菜鸟学逆向学得头皮发麻, 终于它拿到了一段源代码'. The '题目场景' (Problem Scenario) is '暂无'. The '题目附件' (Problem Attachments) section shows a button for '附件1'. A large input field contains the text 'flag..'. Below the input field is a '提交' (Submit) button. At the bottom right, there is a URL 'https://blog.csdn.net/fool_best' and a '查看全部评论' (View all comments) button.

首先当然是下载附件



是一个.c类型的文件, 打开codeblock将此文件拖进去查看代码

The screenshot shows a code editor with a C program and its build logs. The program code is as follows:

```
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]);
11    if (first != 0xcafe) {
12        printf("you are wrong, sorry.\n");
13        exit(2);
14    }
15
16    unsigned int second = atoi(argv[2]);
17    if (second % 5 == 3 || second % 17 != 8) {
18        printf("ha, you won't get it!\n");
19        exit(3);
20    }
21
22    if (strcmp("h4cky0u", argv[3])) {
23        printf("so close, dude!\n");
24        exit(4);
25    }
26
27    printf("Brr wrrr grr\n");
28
29    unsigned int hash = first * 31337 + (second % 17) * 11 + strlen(argv[3]) - 1615810207;
30 }
```

The build logs show the following messages:

```
File Line Message
C:\Users\del... == Build file: "no target" in "no project" (compiler: unknown) ==
C:\Users\del... In function 'main':
C:\Users\del... 7 warning: incompatible implicit declaration of built-in function 'exit'
C:\Users\del... 13 warning: incompatible implicit declaration of built-in function 'exit'
C:\Users\del... 19 warning: incompatible implicit declaration of built-in function 'exit'
```

代码不多，运行之后程序打印“what?”，这。。。并没有什么大用处
通过观察代码，可以发现最后一部分才是解题的关键。

```
    exit(4);
}

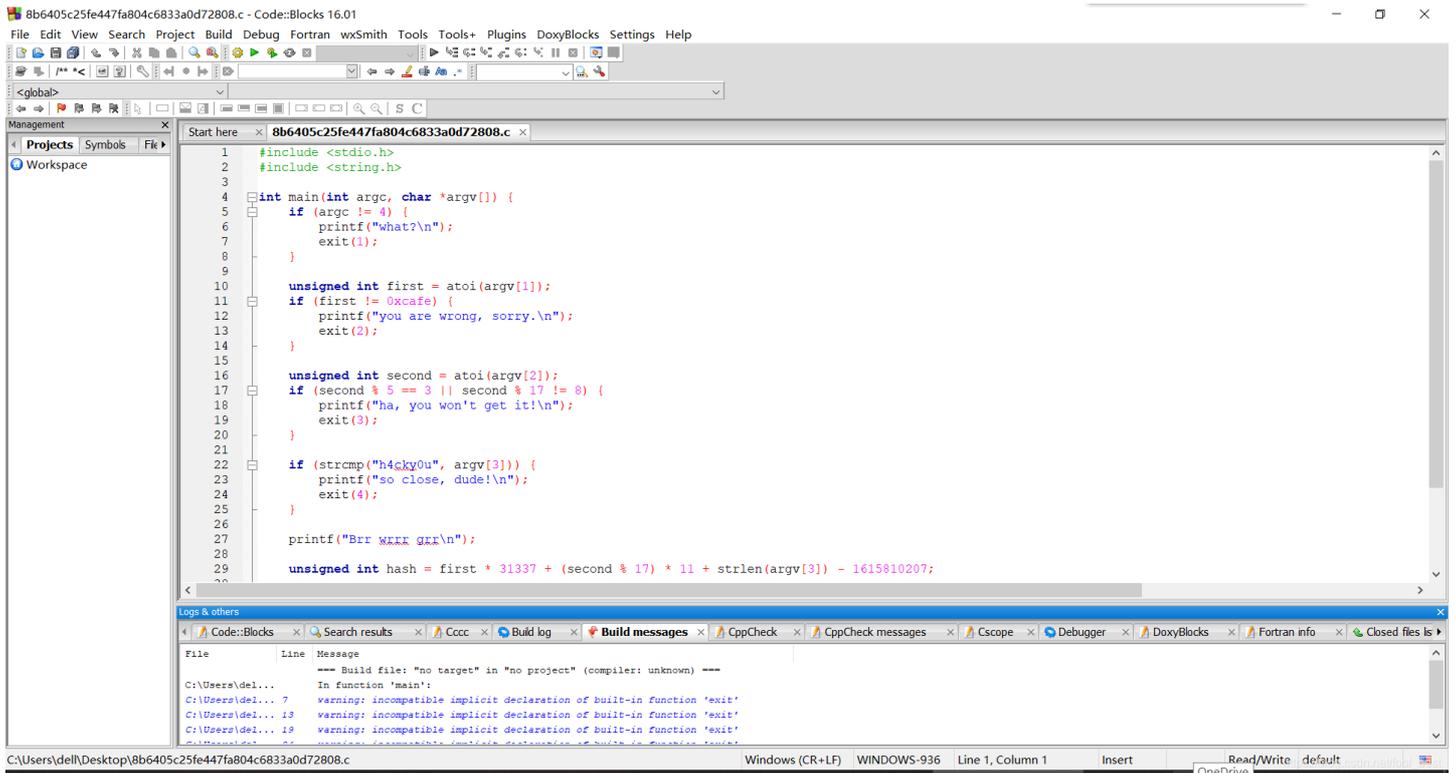
printf("Brr wrrr grr\n");

unsigned int hash = first * 31337 + (second % 17) * 11 + strlen(argv[3]) - 1615810207;

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

找到目标了，就是要求hash，这就好办多了。
现在需要求到first, second和strlen(argv[3])的值。

看上面的代码中有提到那三个变量。



The screenshot shows a code editor window with a C program. The code includes `<stdio.h>` and `<string.h>`. The `main` function takes three arguments. It checks if `argc` is 4, then checks if `first` is `0xcafe`, then checks if `second` is divisible by 5 and 17, and finally checks if `argv[3]` is `"h4cky0u"`. The final line calculates a hash value.

```
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]);
11    if (first != 0xcafe) {
12        printf("you are wrong, sorry.\n");
13        exit(2);
14    }
15
16    unsigned int second = atoi(argv[2]);
17    if (second % 5 == 3 || second % 17 != 8) {
18        printf("ha, you won't get it!\n");
19        exit(3);
20    }
21
22    if (strcmp("h4cky0u", argv[3])) {
23        printf("so close, dude!\n");
24        exit(4);
25    }
26
27    printf("Brr wxxx grr\n");
28
29    unsigned int hash = first * 31337 + (second % 17) * 11 + strlen(argv[3]) - 1615810207;
30 }
```

The build log at the bottom shows warnings about incompatible implicit declarations of `exit` in the `main` function.

first求法:

源代码如下

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

通过if的条件可以知道first就是0xcafe，一个16进制的数字，也就是first的值。

second求法:

源代码如下

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

通过if的条件可以知道，`(second % 5 == 3 || second % 17 != 8)`应该为false，也就是说`(second % 5 == 3)`和`(second % 17 != 8)`两者都是错的，也就可以得到`second%7==8`成立，hash中的一部分`second%17`也是求出来了，就是8，此处暂时将`second`取一个正确的值为25。

strlen(argv[3])求法:

源代码如下

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

直接可以知道`argv[3]=="h4cky0u"`，`strlen(argv[3])=7`。

```

    exit(4);
}

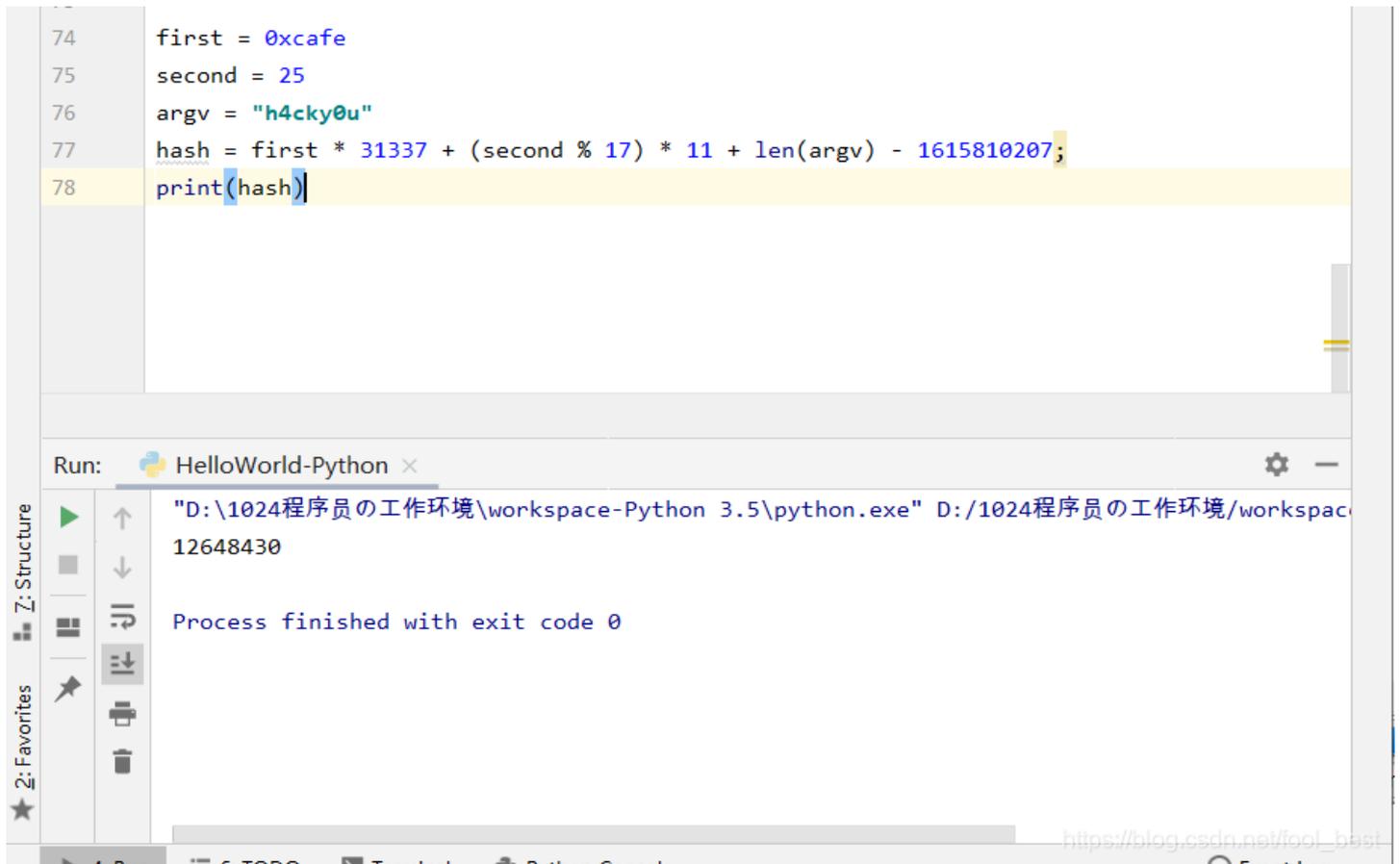
printf("Brr wrrr grr\n");

unsigned int hash = first * 31337 + (second % 17) * 11 + strlen(argv[3]) - 1615810207;

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

```

计算过程中可以有比较大的数字出现，使用C语言可能导致溢出得到错误答案，我们在这里使用python脚本进行运算。



得到结果为12648430，但是这不是flag

接着看下面的源代码

```

unsigned int hash = first * 31337 + (second % 17) * 11 + strlen(argv[3]) - 1615810207;

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

```

%x, 要求以16进制的方式输出。转为16进制，在python中使用 print(hex(hash)), 得到结果0xc0ffee。

(ps: 结果中的0xc0ffee的0x说明这个数是16进制的数字，并不是数字部分)



[创作打卡挑战赛](#) >

[赢取流量/现金/CSDN周边激励大奖](#)