实验吧解题笔记——编程（一）

原创

FunkyPants 于 2017-10-10 22:06:58 发布 1489 收藏

分类专栏： CTF writeup 文章标签： python 编程

本文链接：https://blog.csdn.net/FunkyPants/article/details/78198215

0.说明

每五个题目写作一篇writeup，第一行对应解题笔记（一）……

1.百米

题目描述

题目是这个样子的，很显然需要我们编程，获取网页中需要我们计算的表达式的值，然后提交上去获得flag。

分析

首先，我们使用Python中的requests这个第三方库去获取网页内容，使用以下语句获得这个网页的html文档。

```
get_url = 'http://ctf5.shiyanbar.com/jia/'
post_url = 'http://ctf5.shiyanbar.com/jia/index.php?action=check_pass'

session = requests.session()#建立一个requests的会话对象
html = session.get(get_url).content#使用上面建立的对象去打开网页，并获取html文档内容
```

接下来，我们使用Python中的BeautifulSoup库去解析获得我们想要的内容，只需短短几行语句加上一点耐心，再复杂的内容也可以分解出来。使用以下语句得到

标签后的表达式。

```
soup = BeautifulSoup(html, 'html.parser')
expr = soup.p.div.get_text())#获取到需要计算的表达式
```

这段语句这么简单是因为我取了个巧，因为这个html文档中只有一个p标签，所以直接获取p标签下div标签的内容，再取得div标签中的值，就得到了我们需要计算的表达式。运行结果如下：

```
(3087 + 568) x (5365 - 33) - (890 + 902 - 214) x 735

Process finished with exit code 0
```


需要注意的是，在这里我们获得的表达式是str格式的，我们并不能直接去计算，这里需要用到python中一个内置的函数eval（），它可以计算字符串中python表达式的值。还有一点需要注意的是这里还设了个小坑，表达式中的乘号是以英文字符“x”表示的，我们需要把它替换成“*”，所以计算语句如下。

```
num = eval(expr.replace('x', '*'))
```

接下来，我们只需把计算结果以post方式提交上去就可以了，需要使用burpsuite抓包看到浏览器是怎么把我们的计算结果提交到服务器的，再自己编写payload，并使用requests的post（）方式提交结果。最后的语句如下：

```
payload = {'pass_key':str(num)}
post = session.post(post_url, payload)
print(post.text)
```

使用print语句可以把网页的返回结果直接打印出来，最后在输出结果中可以找到KEY。

完整的程序如下：

```
from bs4 import BeautifulSoup
import requests

get_url = 'http://ctf5.shiyanbar.com/jia/'
post_url = 'http://ctf5.shiyanbar.com/jia/index.php?action=check_pass'

session = requests.session()
html = session.get(get_url).content
soup = BeautifulSoup(html, 'html.parser')
expr = soup.p.div.get_text()#获取到需要计算的表达式
num = eval(expr.replace('x', '*'))
print(num)
payload = {'pass_key':str(num)}
post = session.post(post_url, payload)
print(post.text)
```

2.迷宫大逃亡

//TODO

3.奖券

题目描述

某抽奖活动的奖券号码是6位数（100000-999999），请计算其中不含“4”的号码的奖券数量。
答案格式为：CTF{X}，X为不含“4”的号码的奖券数量

分析

使目用Python遍历字符串，判断字符串中是否含有字符‘4’，如果有不计数，没有的话总数加1。代码如下：

```
sum = 0
for i in range(100000, 999999 + 1):
    if '4' in str(i):
        continue
    else:
        sum += 1
print(sum)
```

看别人的writeup时发现了下面这个只有两行代码的解法，要读懂的话需要对reduce和map的特性有一定了解。

```
from functools import reduce
print(reduce(lambda x,y:x+y,map(lambda i:0 if '4' in str(i) else 1,range(100000,999999+1))))
```

4.三羊献瑞

//TODO

5.找素数

//TODO