CTF考核writeup（2）

原创

离离原上草_# 于 2016-11-23 15:41:07 发布

分类专栏： 安全 文章标签： xss csrf 绕过 上传

没有什么，保存下来。

0x01 Russian

利用burpsuite抓包，其他内容都很正常，发现cookie中info项比较可疑，大写小写数字都有，猜测其为base64编码过的字符串。通过Base64在线工具解码字符串，即得到了第一题flag。

0x02 United States

首先，在输入框尝试输入各种内容，系统提示“unless you are root”。于是输入root，弹窗提示“输入无效”。查看源码，发现前端对输入做了限制，而且仅限制了root。

burp绕过前端验证，将参数username的值修改为root。

成功跳转到了新的页面。新页面上提示“localhost（127.0.0.1）only”。

首先考虑到题意可能是限制ip来源，尝试后发现并不是。所以又尝试了修改Refer、Host。最终发现在Host被修改时拿到flag。

0x03 Canada

打开题目，提示“The form below is protect by Django CSRF MIDDLEWARE”。先尝试点击getflag按钮，提示“Your CSRF token is not what I want. ”。

没有其他线索，先抓包。可以看到提交的参数是csrfmiddlewaretoken，另外cookie里也有一项csrftoken。没有及时截图，实际上第一次抓包的时候这两个值都是一串很长的编码字符串，它们值相等。如果修改其中一个，会报错“CSRF verification failed.”。

为了方便测试，看这两个值被修改的各种效果，我把这个请求放到了repeater模块。找到响应里居然有注释。前面两句代码大概意思是说，如果csrfmiddlewaretoken的值为bypass，那么则返回flag。另外还提到了一个Django的CVE。

看到提示，首先当然会把csrfmiddlewaretoken的值改成bypass，但仅仅这样肯定是无法通过保护机制验证的。实际上到了这里，基本上都能想到把cookie也改了。不过还是先看看cve-2016-7401。

这一漏洞造成的原因是，在某些版本的django中，cookie的解析代码是有问题的。如果这个网站同时采用了google analytics，此时攻击者就可以通过强制设置cookie来绕过csrf保护机制。

这道题的场景里，既然已知‘bypass’，已知漏洞，那就可以通过这种简单方法绕过，从而得到flag。

0x04 China

随意上传一张图片。从提示看出，上传点对上传的文件类型应该是有一定要求的，而且很有可能要求的是txt、doc之类的。

第一次上传doc文档时未成功，又尝试了其他一些不同类型的文件，仍未上传成功。此时，将exe文件的Content-Type修改为doc文件的application/msword类型时，上传成功了。于是发现，只要Content-Type的值为application/msword，不论什么文件都可以上传成功，对实际类型和后缀都是没有过滤的。而第一次上传doc之所以失败是因为文件太大。

题目已提示考察XSS，首先想到上传文件名是否存在XSS。尝试`<script>`标签，根据结果可以看出'/'及其之前的字符都被过滤了。

故想到利用img标签。在filename属性中构造语句：

```
"><image src=1 onerror=alert('xss')>"
```

由于'>'将之前的标签闭合，而src不存在，onerror弹窗，成功发起xss攻击。