WEB-CTF 条件竞争

原创

iamsongyu 于 2018-10-24 16:36:20 发布
分类专栏： CTF 理论知识 文章标签： CTF 网络安全


CTF 同时被 2 个专栏收录
35 篇文章 12 订阅
订阅专栏


理论知识
109 篇文章 7 订阅
订阅专栏

如何来形容这个类型的问题呢，就像是幼儿园发糖吃，每个人只能拿一块，但是如果你跟别人一起再拿一次老师也没办法分辨，毕竟一群小朋友老师也没办法分辨手和对应的人。

关于部分的Burpsuite使用的知识，在前面的https://blog.csdn.net/iamsongyu/article/details/82989478中已经讲了，就不在赘述

这是一个好文章，从上边摘抄和借鉴了不少

http://wiki.secbug.net/web_race-condtion.html

条件竞争漏洞是一种服务器端的漏洞，由于服务器端在处理不同用户的请求时是并发进行的，因此，如果并发处理不当或相关操作逻辑顺序设计的不合理时，将会导致此类问题的发生。

这是一个小例子，很多web程序都会有上传文件的功能，头像和图像等，服务器肯定会检查文件是否满足条件，不满足的要被删除，大概的框架是：

```
<?php
  if(isset($_GET['src'])){
    saveimg($_GET['src']);
    //得到保存路径filename
    //检查文件
    check(filename);
    if(不符合规范)
        deldte(filename);
    else
        pass;
    //...
 }
?>
```

那么问题就在于，如果我们采用大量的并发请求，就传递一个生成恶意webshell的图像，访问它就可以生成webshell。在上传完成和安全检查完成并删除它的间隙，攻击者通过不断地发起访问请求的方法访问了该文件，该文件就会被执行，并且在服务器上生成一个恶意shell的文件。至此，该文件的任务就已全部完成，至于后面发现它是一个不安全的文件并把它删除的问题都已经不重要了，因为攻击者已经成功的在服务器中植入了一个shell文件，后续的一切就都不是问题了。

预防方法

注意并发操作及相关操作逻辑是否得当，如上述获取远端文件时，尽量在将文件保存在本地前就进行相应的安全检查。其他建议待补充。

这里有两个例题，来自以下作者

作者：0verWatch

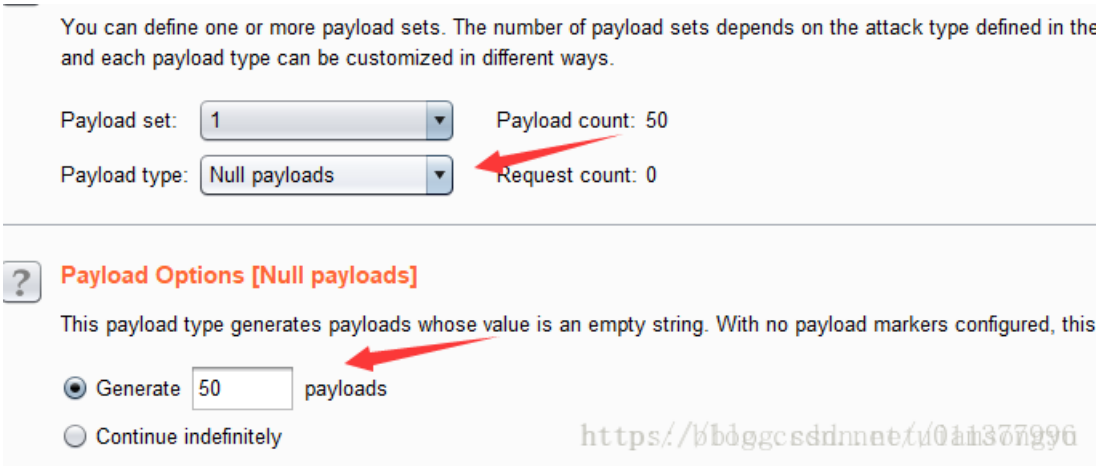
来源：CSDN

原文：https://blog.csdn.net/u011377996/article/details/79511160

cumt平台上的 上传三 http://202.119.201.199/challenge/web/uploadfile/

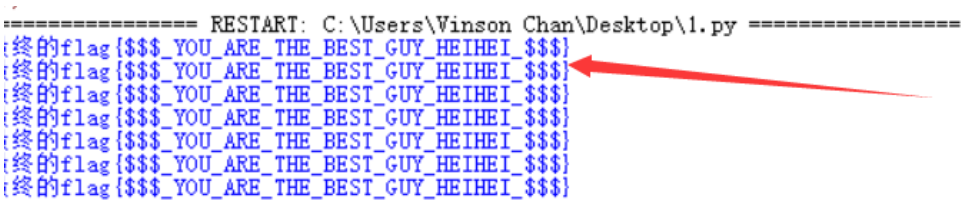
这是一个图像上传，在服务器端使用黑名单的方式过滤，不符合的文件会被删除

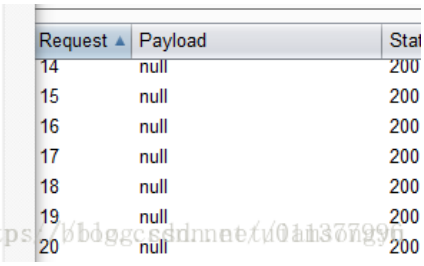
使用条件竞争的方法就是，一边不断地去POST我们的恶意shellcode,另一方面就是不断地去访问服务器创建的那个文件。不断地提交文件可以使用Bp来完成，使用intruder爆破模块，负载设置为没有或者是随意找一个无用的设置一下。



另一方面，使用python不断的访问我们提交的文件，从条件竞争的方向来看，会使得我们有机会访问到我们的文件，最后得到flag

```
import requests
url = '你的文件路径'
while 1:
    r = requests.get(url)
    if 'moctf' in  r.text:
        print r.text
```





moctf上的 没时间解释 http://119.23.73.3:5006/web2/index2.php

题目首先是一个302跳转，我们可以抓包获取跳转前的信息

提示：我们需要上传一下东西.php

访问对应的网页，是一个上传的界面，名字和内容

我们就上传一句话木马，服务器给出保存路径

我们访问一下，发现提示太慢了，那么原理是一样的，他被删除了。我们可以采取跟上一题目相同的方法，也可以都使用Bp。