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

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

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

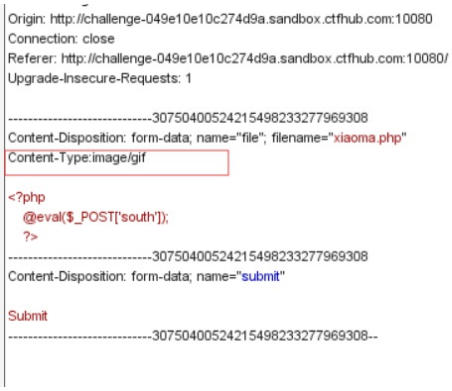
上传文件相对路径
upload/xiaoma.jpg

CTFHub 文件上传 - htaccess

Filename: 浏览... 未选择文件。
Submit

利用蚁剑之间连接即可

2.MIME绕过



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

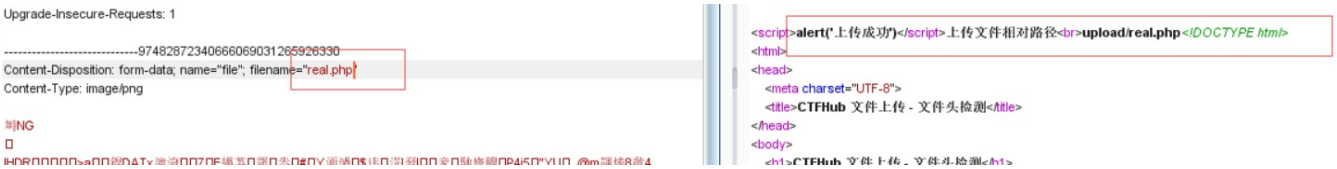
```
<script>alert('文件类型不正确，只允许上传 jpeg jpg png gif 类型的文件')</script><!DOCTYPE html>
<html>
```

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：

```php
if (!empty($_POST['submit'])) {
    $name = basename($_FILES['file']['name']);
    $info = pathinfo($name);
    $ext = $info['extension'];
    $whitelist = array("jpg", "png", "gif");
    if (in_array($ext, $whitelist)) {
        $des = $_GET['road'] . "/" . rand(10, 99) . date("YmdHis") . "." . $ext;
        if (move_uploaded_file($_FILES['file']['tmp_name'], $des)) {
            echo "<script>alert('上传成功')</script>";
        } else {
            echo "<script>alert('上传失败')</script>";
        }
    } else {
        echo "文件类型不匹配";
    }
}
```

前面是利用文件名后缀对jpg,png,gif进行筛选的白名单。我一直以为是对这里进行利用，用 .php.jpg化.为00的方式绕过前缀名检查，结果发现不行，应该是我对这个知识的原理理解有误
然后就到了这个题最精髓的地方了，下面这一段

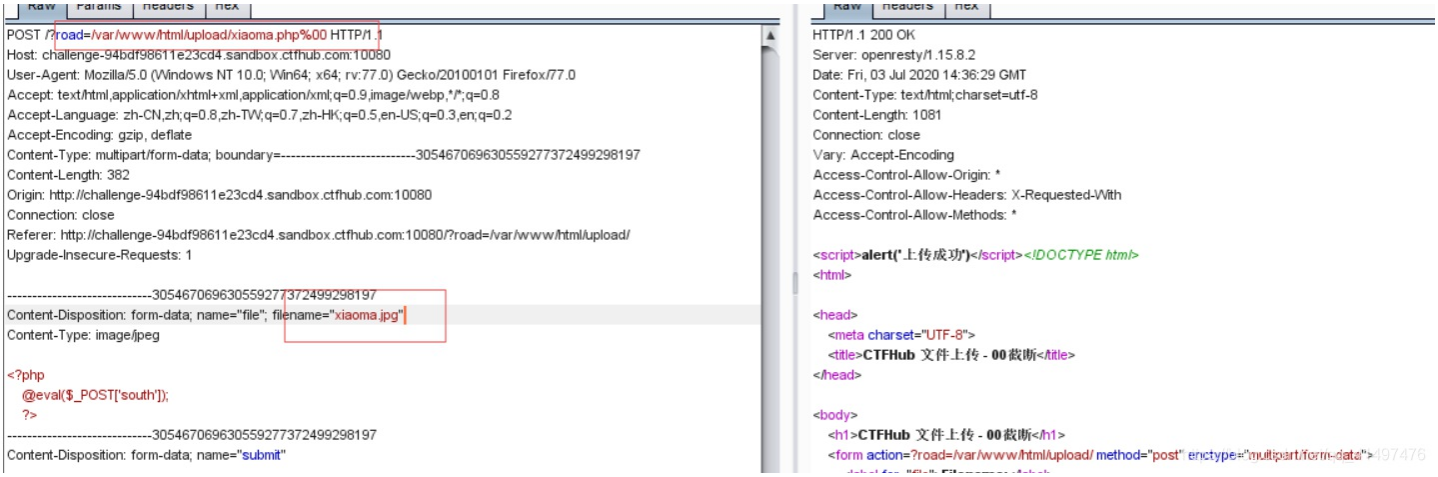
```php
 $des = $_GET['road'] . "/" . rand(10, 99) . date("YmdHis") . "." . $ext;
        if (move_uploaded_file($_FILES['file']['tmp_name'], $des))
```

是将我们上传的文件重命名并存储在某个路径下，
完整的路径是 GET['road']+随机数+日期加前面获得的后缀名
正常来说，这里的随机数和时间我们无从知晓，唯一能控制的后缀名还被限制了。 这样的情况下唯一的解决思路其实就在于 `$_GET['road']` 这里

可以发现在上传图片的时候
还会同时用GET方式上传一个GET变量用于生成文件路径

```php
 $des = $_GET['road'] . "/" . rand(10, 99) . date("YmdHis") . "." . $ext;
```

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



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。

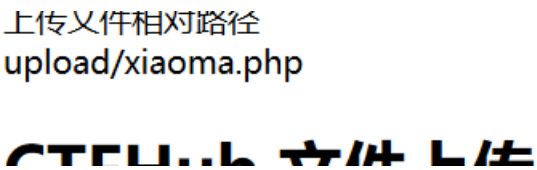
5.双写后缀

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", "jspa", "jspx", "jsw", "jsv", "jspf", "
jtml", "asp", "aspx", "asa", "asax", "ascx", "ashx", "asmx", "cer", "swf", "htaccess", "ini");
$name = str_ireplace($blacklist, "", $name);
```

这个是直接用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.pphphp`即可，检测到php删掉后，剩下的p和hp重新组成php



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