基于PHP的SQL注入研究

**0x00基础**

注：PHP版本为老版本

1.数据库的连接

$link=mysql\_connect($host,$dbuser,$dbpass);

//打开一个非持久的SQL连接

mysql\_select\_db($dbname,$link);

//选择需要使用的数据库

mysql\_query($query);

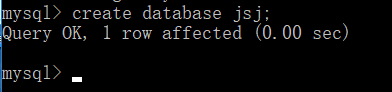
执行一条SQL查询

下方列出一些常见的sql命令 //注意语句后要使用分号”;”

//对大小写不敏感

2.对数据库的基本操作

creat database name; //创建数据库

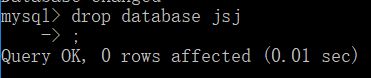


创建 jsj库 //计算机

use database name; //选择数据库



使用jsj库;



删除 jsj库;

3.对数据表的基本操作

新建数据表

create table tablename

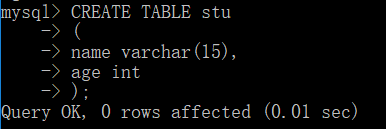
(

列名1 类型,

列名2 类型,

…

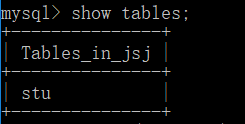
);



新建表stu;

查看本数据库的表

show tables;



向表中插入数据

insert into tablename(列名1,列名2,…) values(’值1’,’值2’);



向stu表中插入数据;

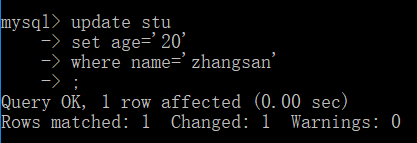
更新表中数据

update table\_name

set column\_name= new\_value

where column\_name= some\_value

;

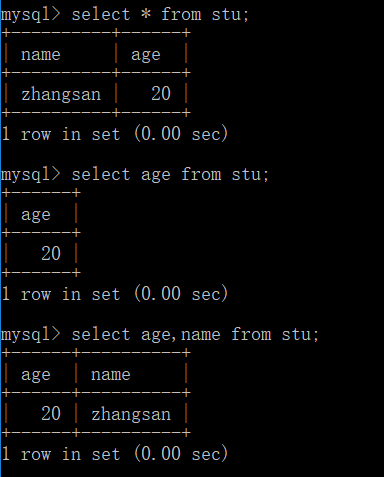


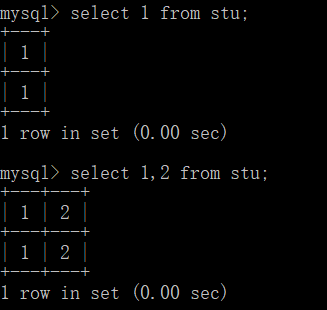
查询数据表\*

select column\_name(s) from table\_name;

select column\_name(s) from dbname.table\_name; //跨库查

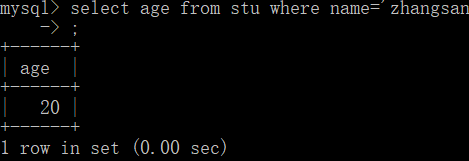
询

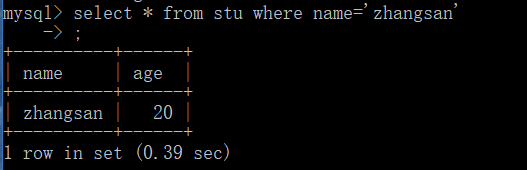




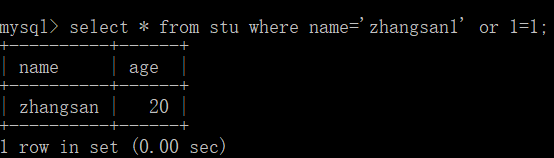
select column\_name(s) from table\_name where 加条件

注：此语句是取where后面条件为true时的数据





特殊语句常常用于注入 即where后语句恒为true



**一个php中的完整操作**

<?php

error\_reporting(0); //关闭报错

$con = mysql\_connect(’localhost’,’root’,’root’); //建立连接

if(!$con)

{

die(’连接失败’.mysql\_error()); //输出错误,退出当前脚本

}

mysql\_select\_db(’jsj’,$con);

$result = mysql\_query(’select \* from stu’);

//下方程序无实际意义

while($row =mysql\_fetch\_array($result);)

{

echo $row[’name’].’<br/>’;

echo $row[’age’].’<br/>’;

}//遍历输出

mysql\_close($con); //关闭数据库

?>

//注：mysql\_fetch\_array($result);函数会读取一行数据作为关联

组,每执行一次都会将指针置到下一行。

**0x01渗透入门**

**参考资料**

<https://blog.csdn.net/qq_35661990/article/details/80257348>

源程序片段

$sql = ”SELECT \*FROM users WHERE id=’$id’ LIMIT 0,1”;

$result = mysql\_query($sql);

$row = mysql\_fetch\_array($result);

if($row)

{

echo ………;

}

注：

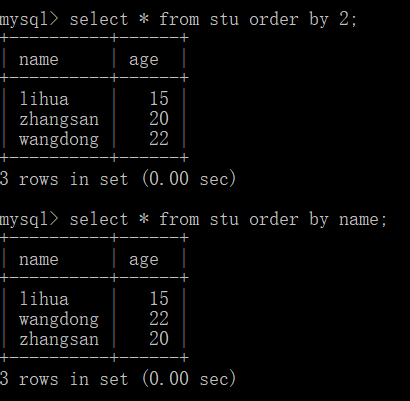
LIMIT 0,1

从0开始取,取出1个

LIMIT也可以对SELECT取出的数据内容进行限制;

order by +column\_name(列名),… //可以有多个列名作为次选条件

order by 对SELECT选取的内容进行排序



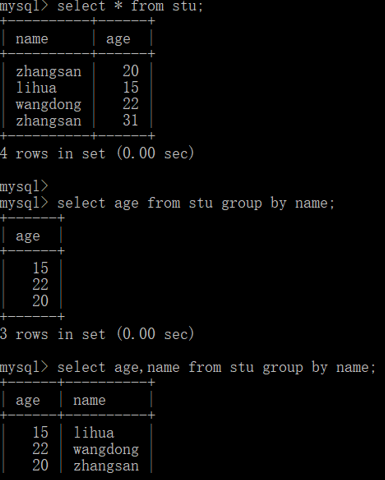
也可

order by +column\_num(列序号) //渗透时常用于判断总列数

group by + column\_name(列序号)

//按column\_num 对要取出的数据进行分类汇总 重复的排序后

第一个



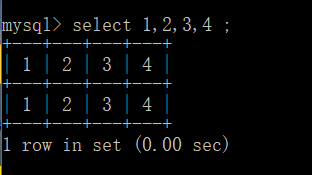
union 操作符

此操作符用于合并两个或多个SELECT语句的结果集。

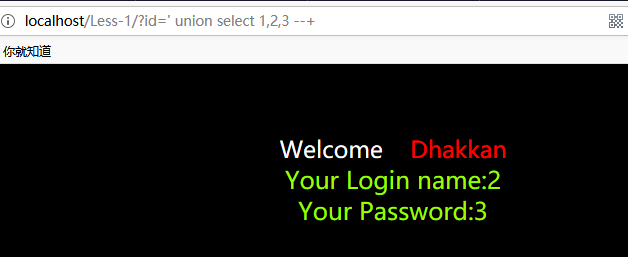
在注入时强强巧妙利用引号使正常查询的内容为空

注：注释符号 # --+ %23代替

select 1,2,3; 常用无意义语句 //用于取出个性数据



最后再根据网页中的输出,判断出可以利用的位置

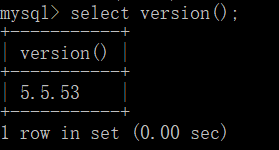


如栗子中^\_^：可利用的位置为2,3

(必要时可输入更具有的代表性的字符便于观察)

version()

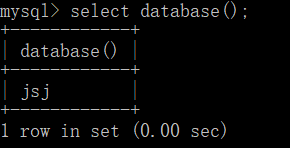
select version(); //用于查询数据库版本

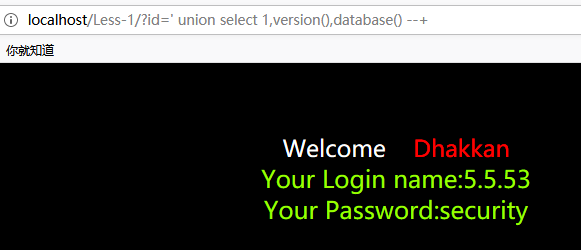


常用于判断是否为5.0以上版本,后面渗透会使用对应库

database()

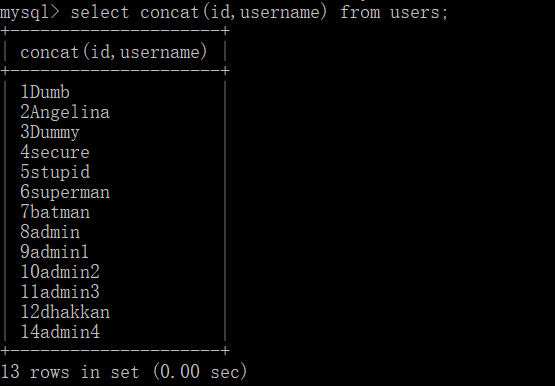
Select database(); //查询正在使用的数据库





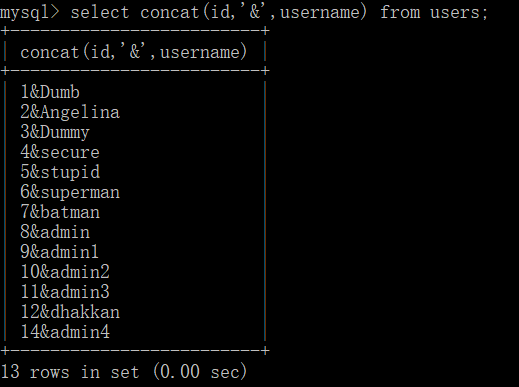
concat(列名1,列名2) -----了解(渗透用不上)

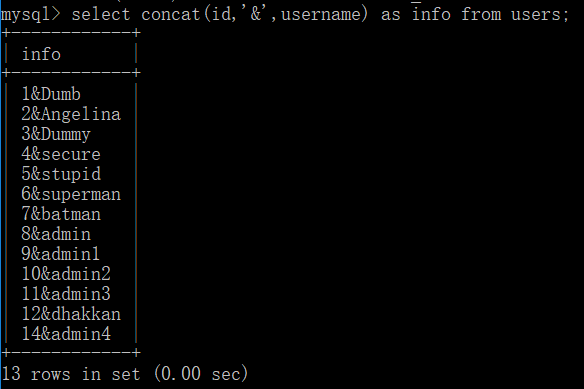
可以将多列信息视为一列,占一个位置输出



上图为将 id,username合并输出

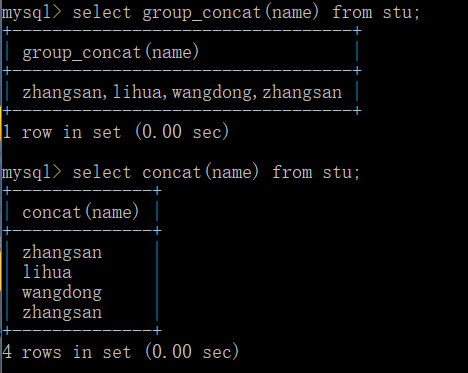
也可用来在数据间增加分隔符





**\*\*group\_concat(conlumn\_name1，conlumn\_name2..)**

同concat() 但可以将数据合并为一行(一项)进行输出。



上图明显看出区别

可以利用group\_concat()我们可以利用行查询单个表中的任意长度数据 ---------重点

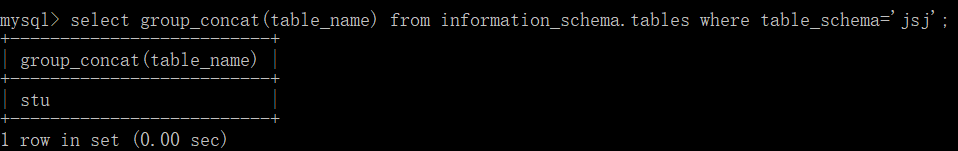
Mysql 特殊数据库 ：information\_schema

information\_schema.tables 有所有的数据库的表信息

列信息：

table\_name 所有表名

table\_schema 表名对应的库名

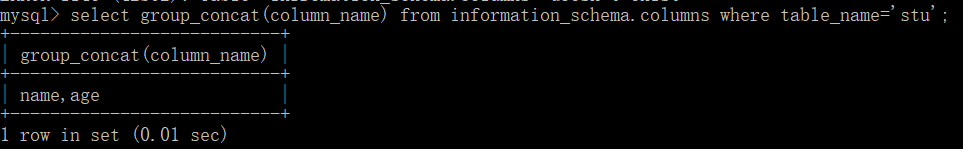


information\_schema.columns 有所有的数据库的表信息

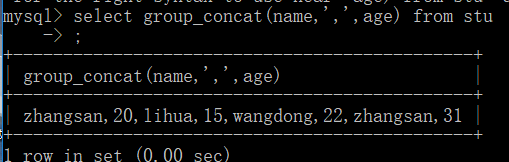
列信息：

colsumn\_name 所有列名

table\_name 列名对应的表名



最后



恭喜你入门惹（￣▽￣）～■□～（￣▽￣）----完！

**注入顺序 尝试在正常内容后面增加 单引号 双引号 括号**

**0x02利用报错渗透**

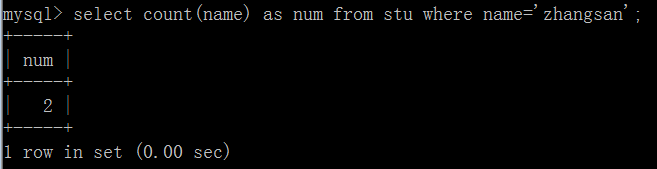
**参考资料：**

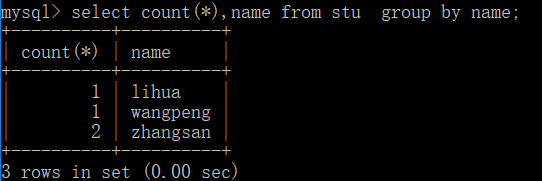
<https://blog.csdn.net/x728999452/article/details/52091723>

<https://blog.csdn.net/Butterfly_Erhai_Lake/article/details/78425600>

<https://www.jianshu.com/p/50c112c78569>

count(column\_name): 返回匹配指定条件的行数





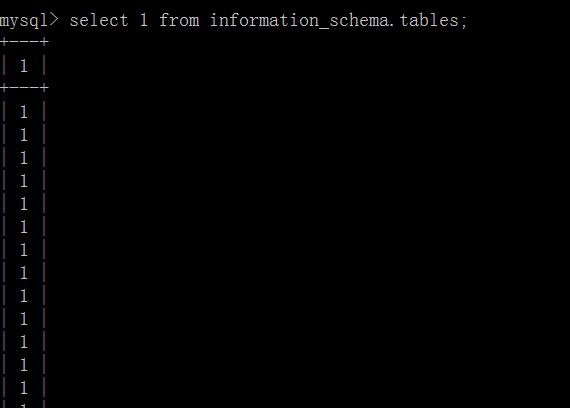
count(\*) 返回被选行数

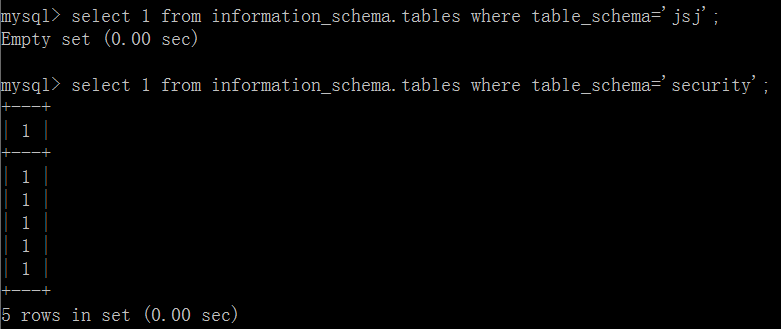
当使用 group by 时 count(\*) 会显示分类的数量

rand(): 返回一个0~1的float伪随机数

floor(): 向下取整

发现from会规定选出的列数





所以where实际是限制了读取哪几列

核心语句：

?id=1' and (select 1 from (select count(\*),concat(database(), '~' , floor (rand()\*2))as a from information\_schema.tables group by a) as b limit 0,1) --+;

得到数据库名称后可将

database()更改为

(select table\_name from information\_schema.tables where table\_schema=database() limit 0,1)

由于只能取出一个 所以之只能通过更改 上面的 limit

例如更改为 limit 1,1实现读取第二个.

(select password from security.users limit 0,1)

最后一条只能一项一项的进行查询 例如上面只能输入一个password

不能输入 group\_concat(username,password) 这种

**0x03报错不提示的渗透**

**核心语句**

select \* from stu into outfile ’D:\\1.txt’;

方法很有局限性 首先需要知道

secure\_file\_priv

//Mysql允许导出的路径 如不存在则无法继续进行

可以通过命令行 show variables like ’%secure%’ 进行模糊查询

或打开配置文件 my.ini 查看 secure\_file\_priv

**0x04无任何回显的渗透 --考的可能性很小(了解)**

需要进行盲注,手工注入太慢 需要变成

<https://blog.csdn.net/ski_12/article/details/70168346>

核心代码:

payload:

?id=1' and if(ascii(substr((select database()),1,1))>115,0,sleep(5)) --+

注:

ascii() 返回参数对应的ASCII编码

“0” → 48

“A” → 65

“a” → 97

substr(database(),1,1)

返回 database() 字符串的第一位开始的一位

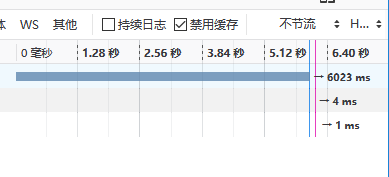
if(expr1,expr2,expr3)

expr1值为TRUE,则返回值为expr2

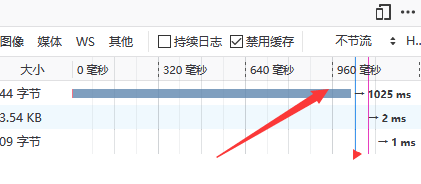
expr2值为FALSE,则返回值为expr3

(工具火狐浏览器)

条件不成立时的响应时间 (大于5秒)



条件成立时 时间明显正常一些



原理结束!

**0x05 宽字节注入**

**参考资料**

[**https://www.cnblogs.com/jinqi520/p/9581510.html**](https://www.cnblogs.com/jinqi520/p/9581510.html)

原理:mysql在使用汉字编码的时候,会认为两个字符为一个汉字，例如%df%5c就是一个汉字(前面的ascii码大于118才能够到汉字的范围)。我们在程序中过滤 ’ (单引号)的时候,往往利用的思路时 将 ’ (单引号)的前面增加 \ 进行转义

注入过程:

我们大致确定后台代码的执行结构后输入%df%27

// %df → 乱码字符 %27 → 单引号

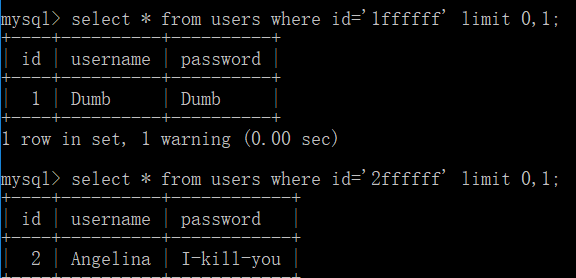
在后台 %27会被转义为 \’ 也就是%5c%27

但是后台执行时会将 %df%5c 结合到一起

这样我们能的单引号就起作用了



下图为执行过程中偶然发现数据库对条件具有略智能的判断(也可能只是转换了类型)



get基础注入 end!!

POST注入

首先我们要知道第一个不同点 这里我们的注释符号变成了 ’#’

Less-17

<https://blog.csdn.net/qq_30464257/article/details/80632323>

报错注入(二)

updateXml()

核心语句:

id=1’ and updatexml(1,concat(0x7e,(select version()),0x7e),1) #

也只能一个一个查询

admin' or updateXML(1,concat(0x7e,(select table\_name from information\_schema.tables where table\_schema=database() limit 0,1),0x7e),1)#

Less-18,19为头注入 需要使用工具 例如火狐修改头插件

<https://blog.csdn.net/fly_hps/article/details/80281913>

Less-20 正常登陆后 后台信任cookie 利用这点注入

<https://blog.csdn.net/fly_hps/article/details/80284993>

//注:很多程序会对用户名进行过滤,这时利用密码参数进行渗透

//注入基础20关 (完) 熟练掌握上面的资料可以轻松过20关

**0x06 防注入(修复方法)**

htmlspecialchars()函数和addslashes()函数

两个函数都是对特殊字符进行转义

addslashes()

通常用于防止sql注入,它可将单引号和双引号进行转义

如 bb’ or 1=’1 通过addslashes() 过滤后会变成

bb\’ or 1=\’1

htmlspecialchars()

也是用于字符转义 但是将特殊字符进行实体转换

<script>alert(’xss’)</script>通过htmlspecialchars()过滤后为&lt;script&gt;alert(&#039;xss&#039;)&lt;/script&gt

不同点

addslasher()通常用于防止sql语句注入,在转义后sql执行完成后,插入数据库中的数据时不带有转义字符\的。所以防止XSS时依然需要htmlspecialchars()。

$keyWord = str\_replace(’%’,’\%’,$keyWord);

直接进行指定字符替换