软件安全实验报告

学号: 2310764 姓名: 王亦辉 班次: 计科一班

1 实验名称：

SQL盲注

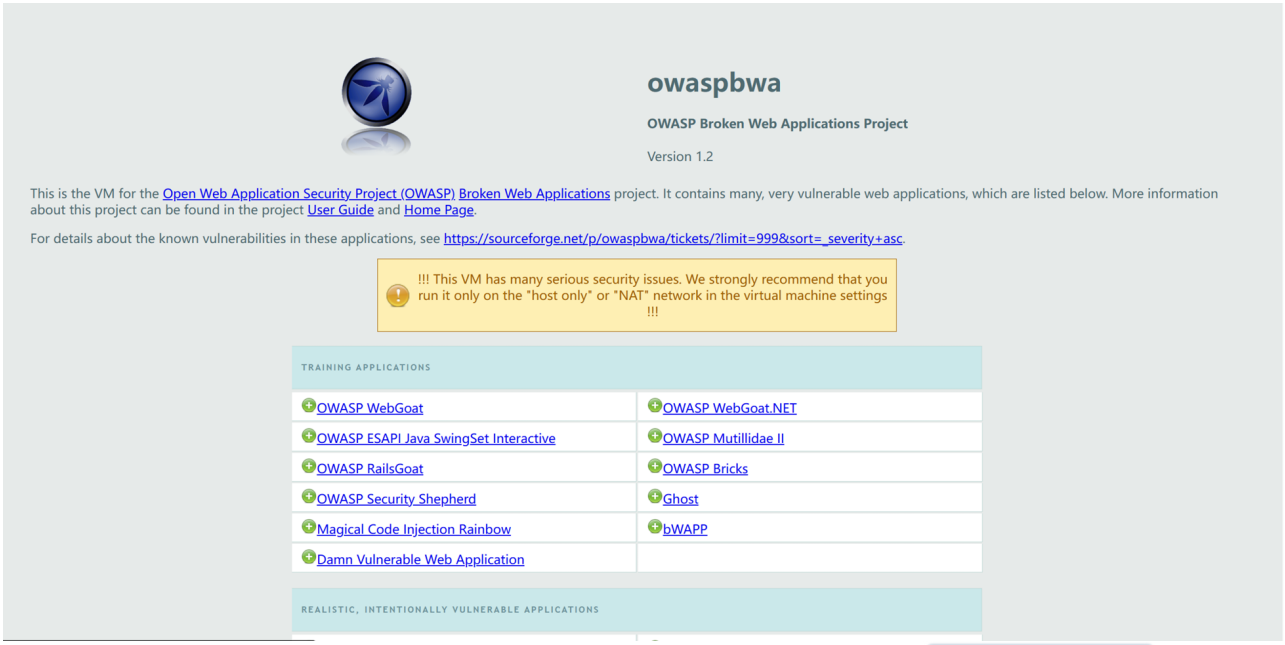
2 实验要求：

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。

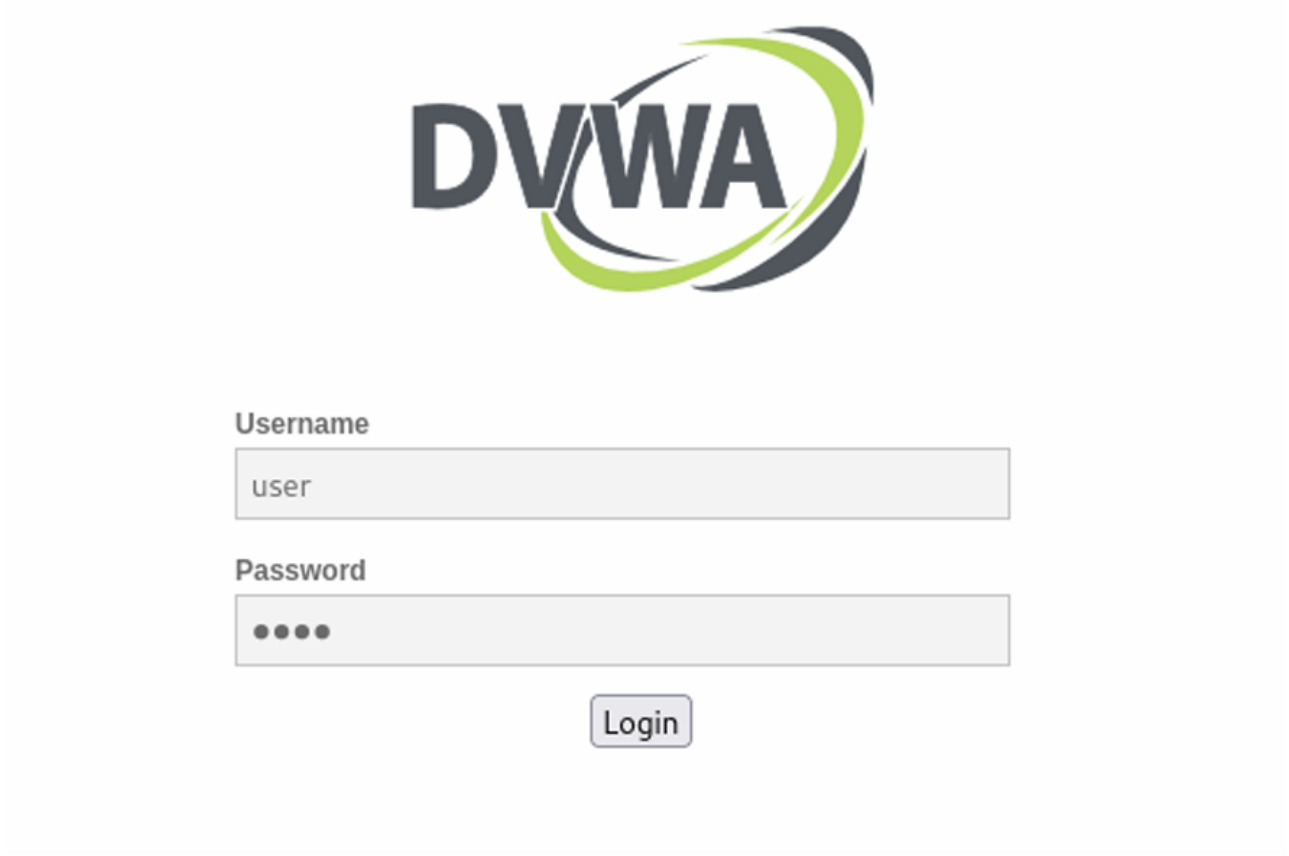
3 实验过程：

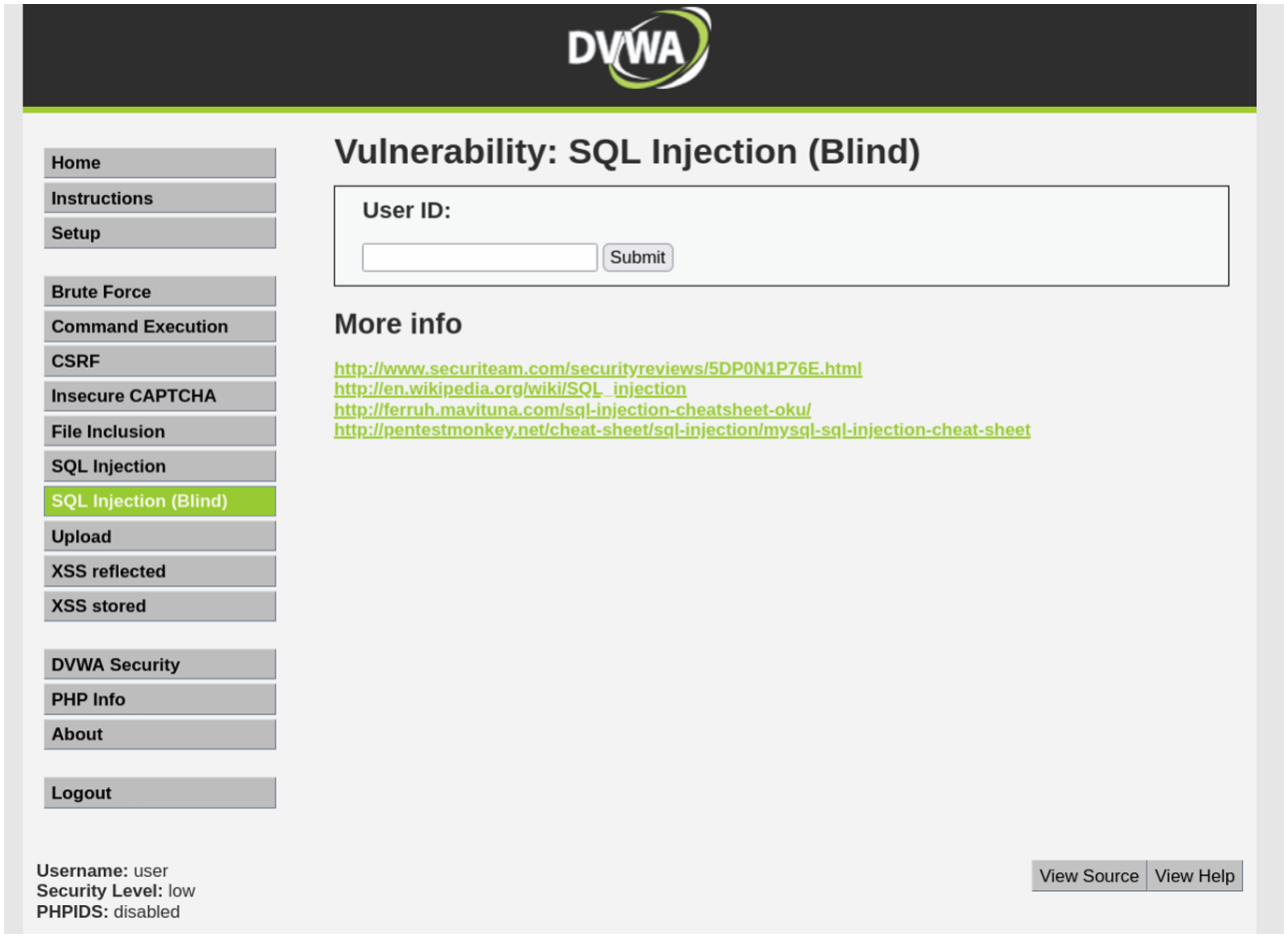
3.1 首先我们进行环境配置，

下载 OWASP_Broken_Web_Apps_VM_1.2 ，在虚拟机中打开登录，然后使用其 ip 地址 `192.168.57.124`，在 kali 的浏览器中进入。



选择 DVWA（Damn Vulnerable Web Application），并登录。将 `DVWA Security` 设置为 `low`

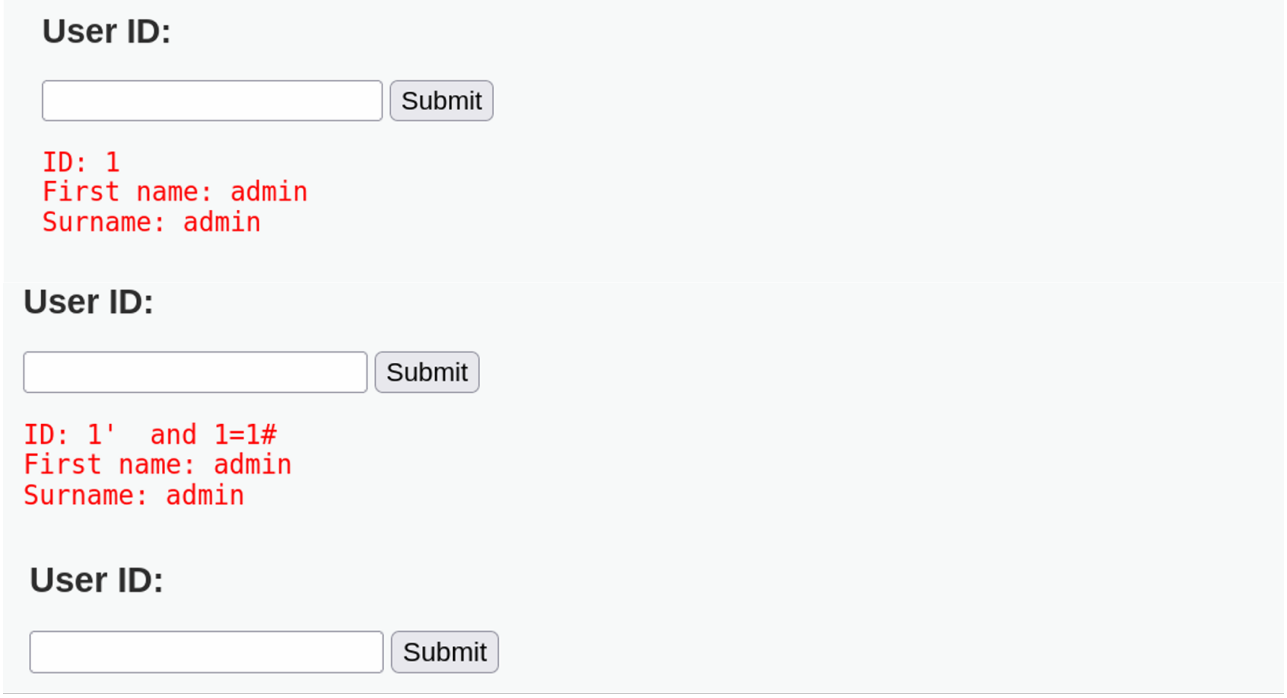

DVWA
Username
user
Password
Login


DVWA
Home
Instructions
Setup
Brute Force
Command Execution
CSRF
Insecure CAPTCHA
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
PHP Info
About
Logout
Vulnerability: SQL Injection (Blind)
User ID:
Submit
More info
http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/
http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet
Username: user
Security Level: low
PHPIDS: disabled
View Source
View Help

3.2 接下来我们进行基于布尔的 SQL 盲注

首先我们需要判断是否存在注入，以及注入是字符型还是数字型。

输入 `1` 以及 `1' and 1=1#` ，显示存在，输入 `1' and 1=2#`，显示不存在。

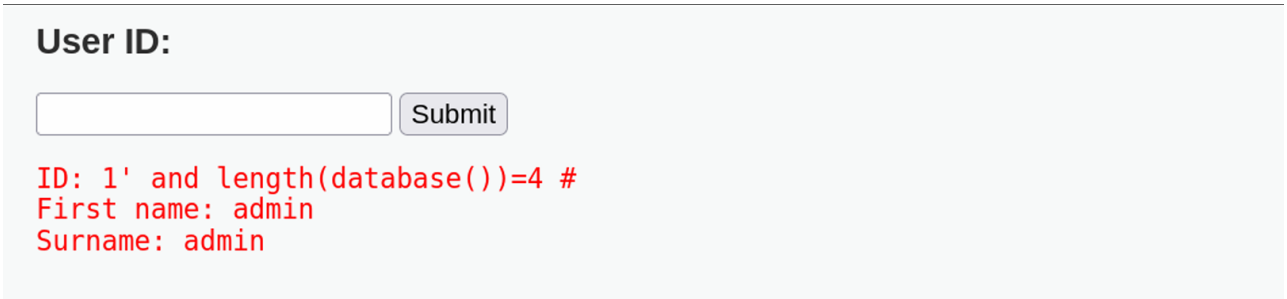


说明存在字符型的 SQL 盲注

3.3 接下来破解数据库名

首先猜测数据库名的长度。

`1' and length(database())=1 #`，一直试到 `=4`，成功显示，说明数据库名长度为 4。

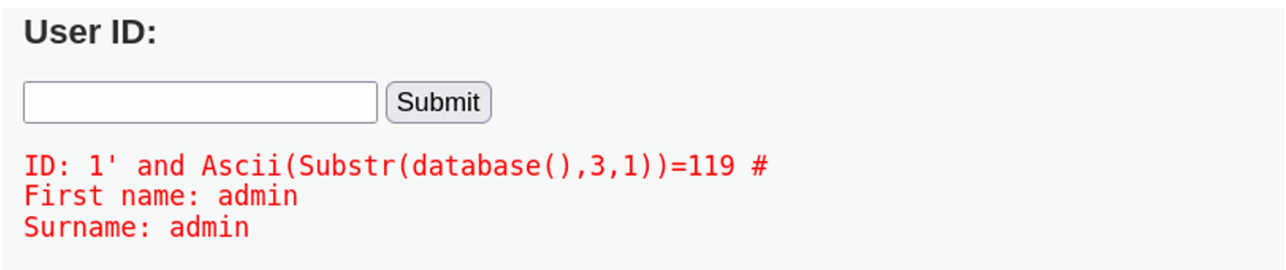


然后使用二分法一个个猜每个字符是什么。

输入 `1' and Ascii(Substr(database(),1,1))>97 #`，`1' and Ascii(Substr(database(),1,1))<122` 等，猜出第一个字符。

直到发现 `1' and Ascii(Substr(database(),1,1))=100 #` 有显示， 成功了。

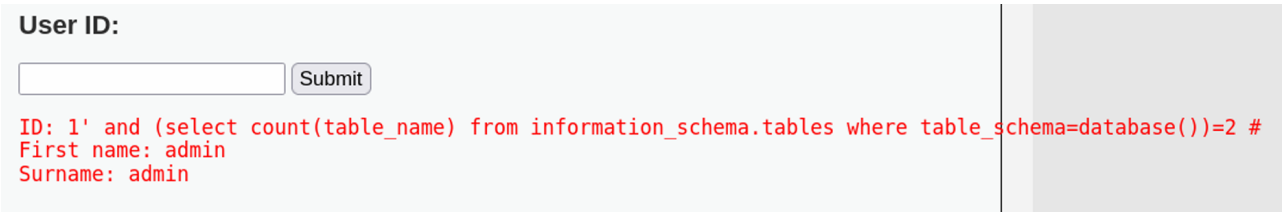
同理猜出其他字符，ascii 码分别是 118，119，97。



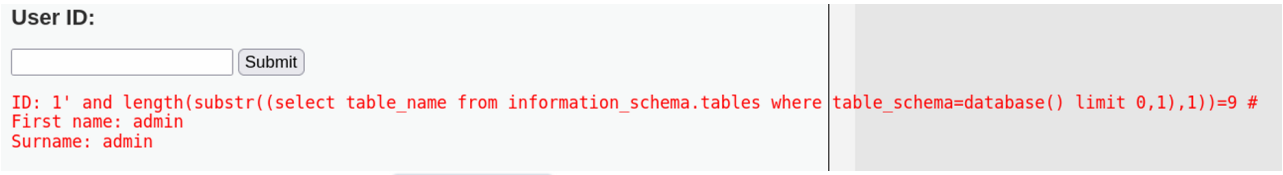
因此我们知道了完整数据库名为 dvwa

3.4 猜测数据库中的表名

首先从 `=1` 开始，往上猜，猜出数据库有 2 张表。



第一张表，类似猜数据库名，采用先猜出表名长度为 9，再逐个字符使用二分法猜出的方法。



继续用二分法，得到表名为 `guestbook`。

第二张表，同理，表名长度为 5，表名为 `users`

接下来，对于表中的字段名，数据等等，仍然可以用一样的方法，逐步破解。

4 心得体会：

本次实验让我系统性地了解了SQL 盲注的原理、分类及其具体攻击流程。通过手动构造注入语句，我切身体会到了在安全防护缺失的情况下，攻击者是如何一步步获取数据库信息的。相比普通SQL注入，盲注无法直接看到返回的SQL查询结果，因此攻击者只能通过页面的响应差异来推测数据库内部信息，这要求攻击者具备更强的逻辑推理和耐心。

通过实践，我深刻体会到输入验证和权限控制在 Web 开发中的重要性。若缺乏这些防护措施，哪怕只是一个普通的表单输入，也可能成为攻击入口，造成严重的数据泄露问题。