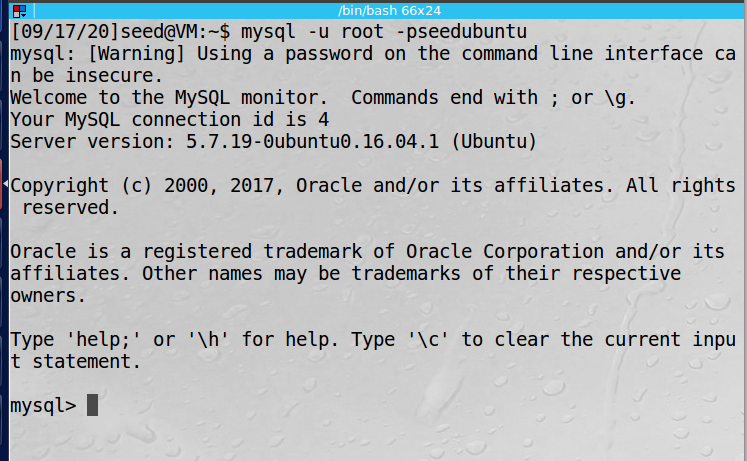
网安实训报告

Lab6

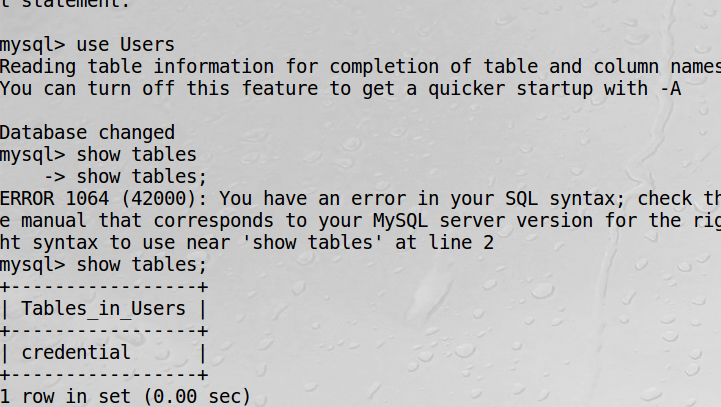
57118139 顾宸玮

**Task 1: Get Familiar with SQL Statements**

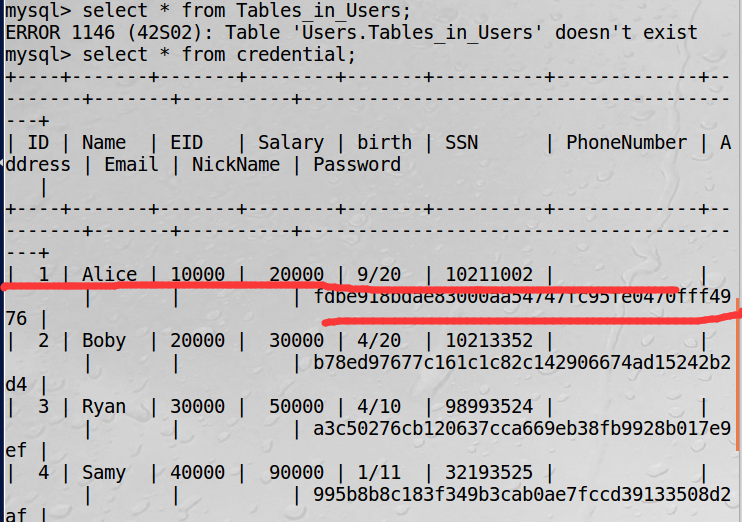
首先登录MySQL：



输入use Users；再输入show tables；（要加分号）



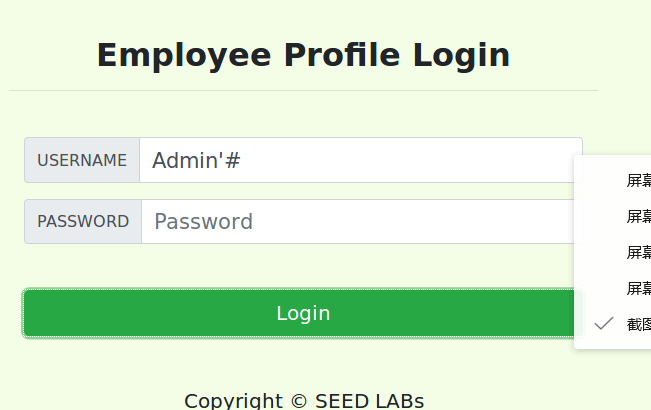
要展示Alice的信息，就输入“select \* from credential”，展示出每个用户的信息。

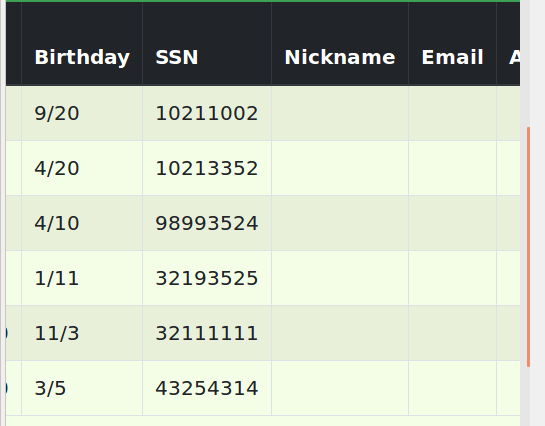


**Task 2: SQL Injection Attack on SELECT Statement**

**Task 2.1: SQL Injection Attack from webpage**.

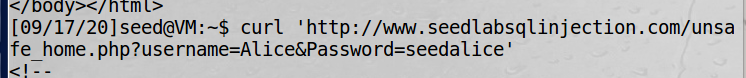
输入Admin用户名，直接在用户名输入 Admin’#,不用密码选择登陆，可以看到所有用户信息：

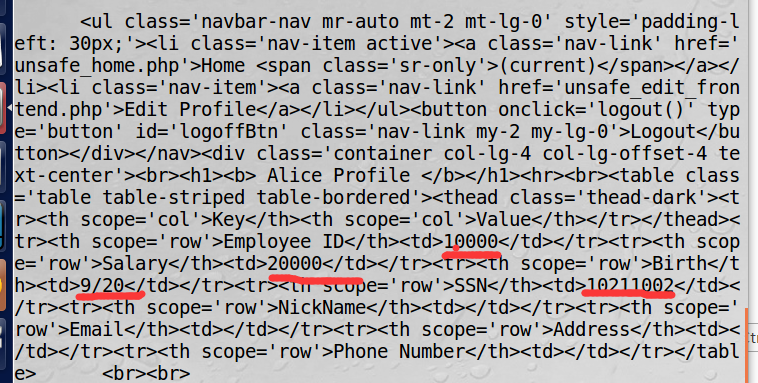




**Task 2.2: SQL Injection Attack from command line**.

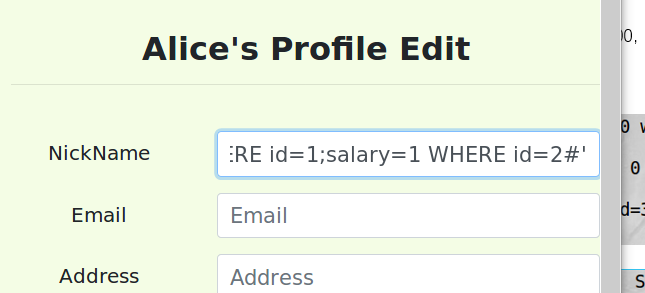
通过输入‘curl’的指令，能够出现Alice的网页代码，从而看到Alice的网页信息。

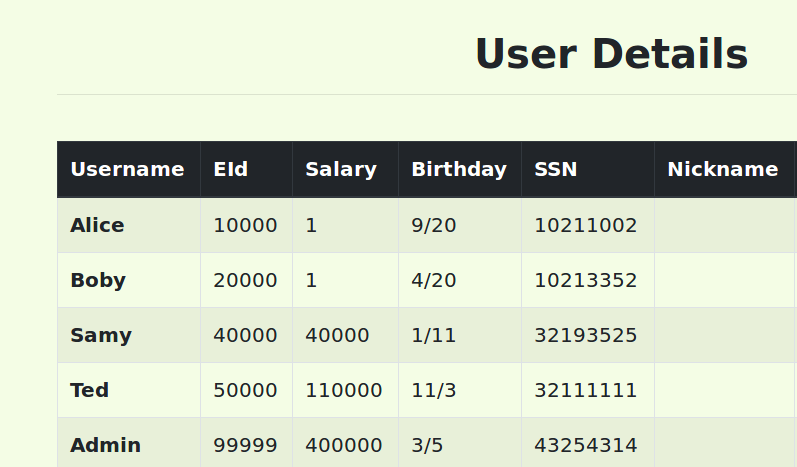




**Task 2.3: Append a new SQL statement**

用分号隔开了两个语句，首先修改Alice的薪水为1，再修改Boby薪水为1，中间冒号隔开再显示数据如下。

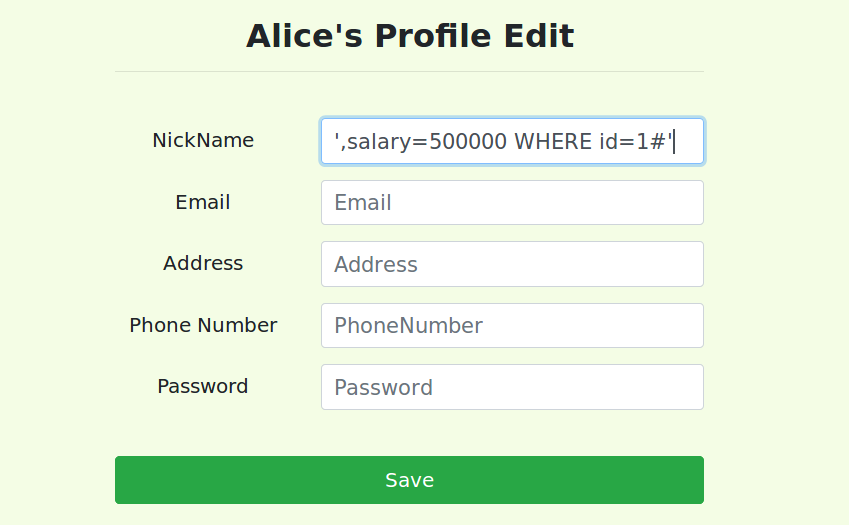


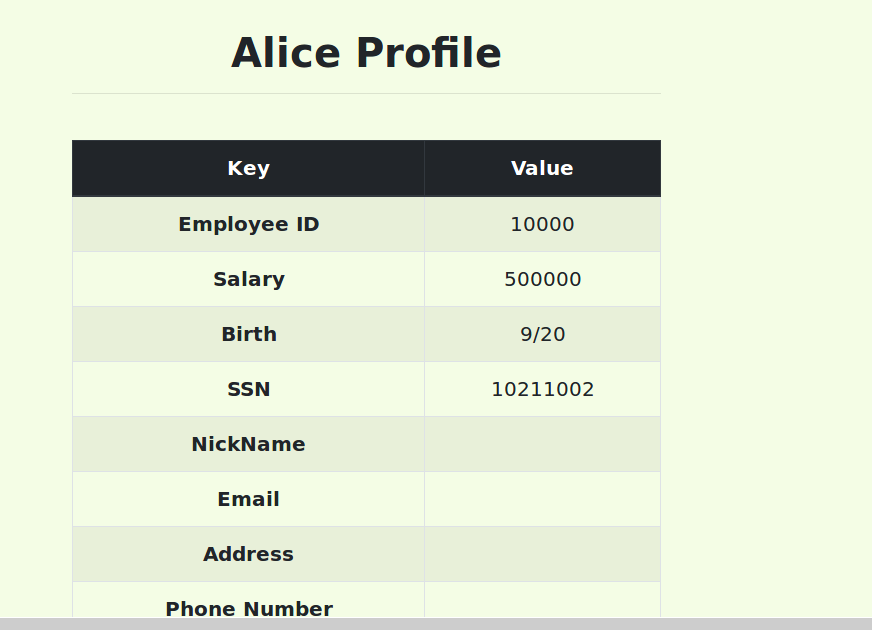


**Task 3: SQL Injection Attack on UPDATE Statement**

**Task 3.1: Modify your own salary**.

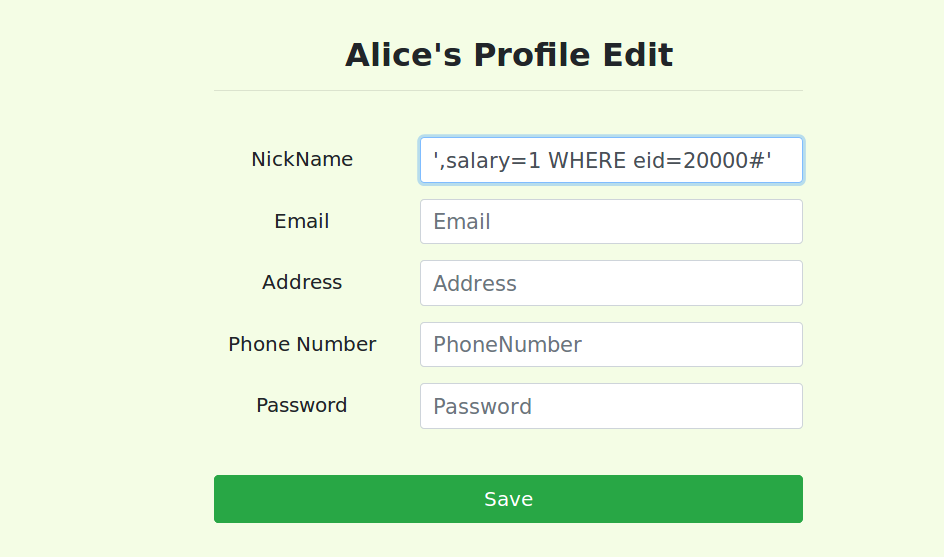
在Alice的“edit”页面中，修改Nickname内容如下，用“，”隔开语句，将薪水改成500000元，并且末尾加上“#”（Alice的id为1），保存后查看Alice信息，发现薪水成功被修改！。



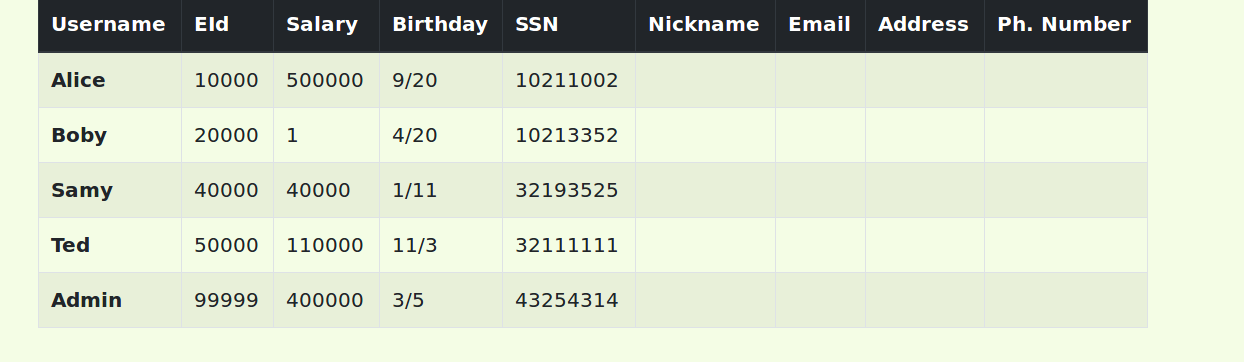


**Task 3.2: Modify other people’ salary**.

在Alice的修改页面输入以下内容，假设已知boby的工号为20000

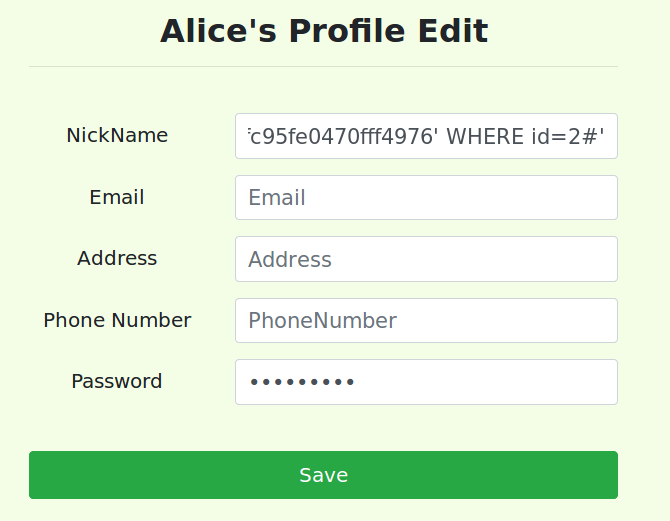


修改Boby薪水成功

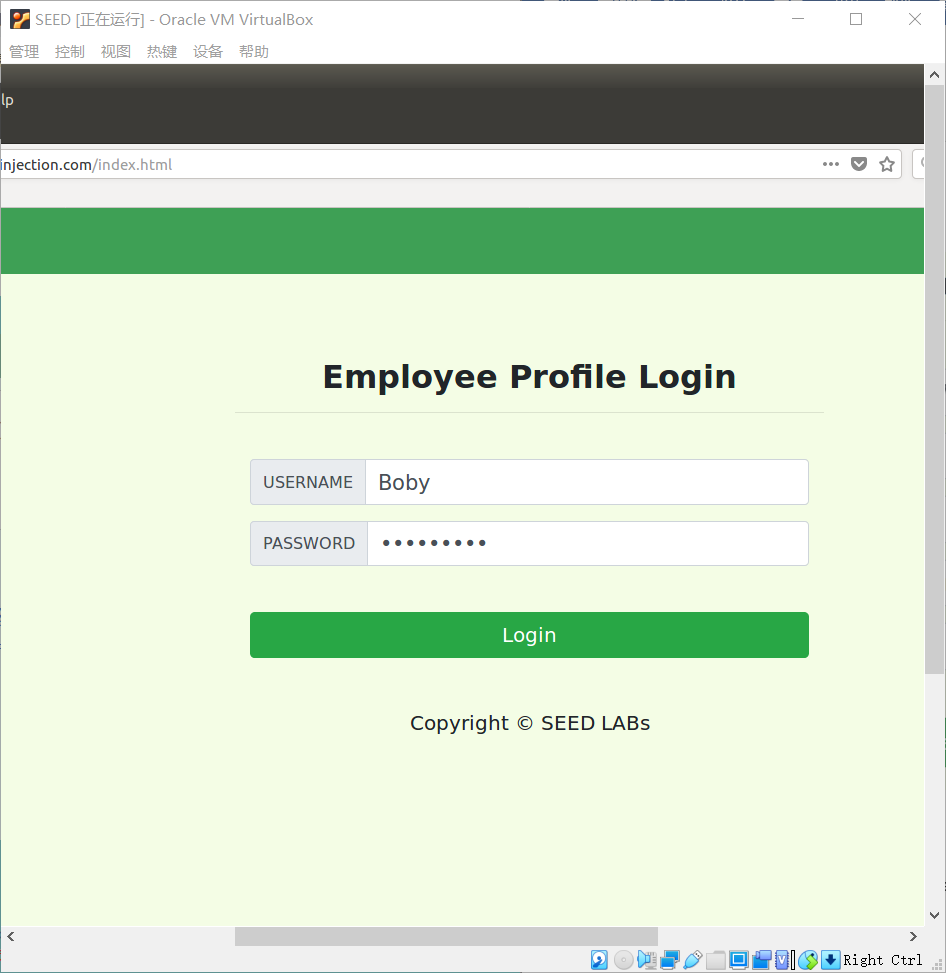


**Task 3.3: Modify other people’ password**.

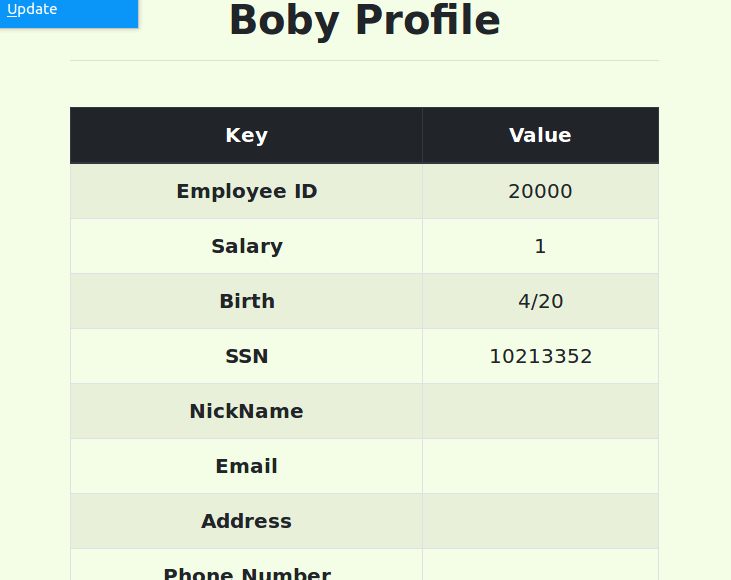
在Alice修改界面中输入 ‘,password= ‘fdbe918bdae83000aa54747fc95fe0470fff4976’ WHERE id=2#’ 并且输入新密码，否则新密码位置为空，程序直接被截断，难以实现更新。其中fdbe918bdae83000aa54747fc95fe0470fff4976表示“seedalice”的明文经过SH1变换的hash值



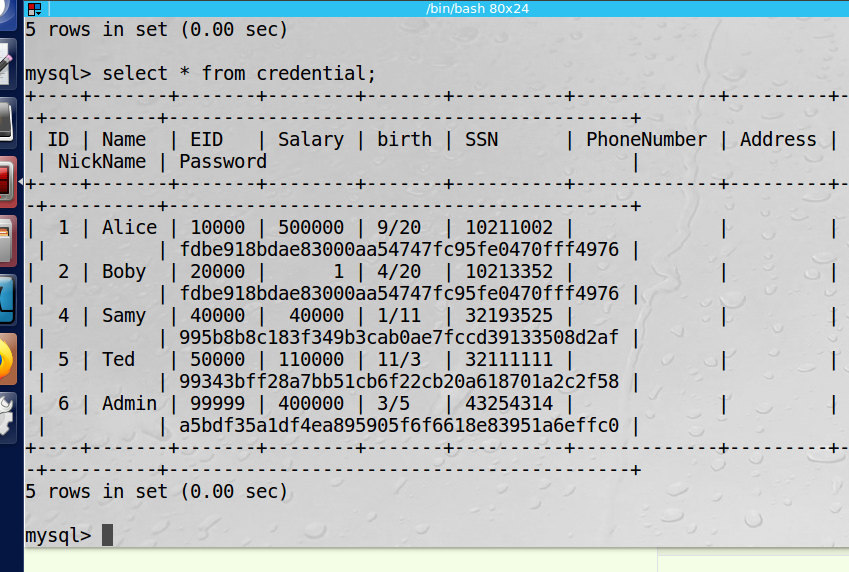
重新登录Boby账号，使用新的密码“seedalice”



登陆成功！



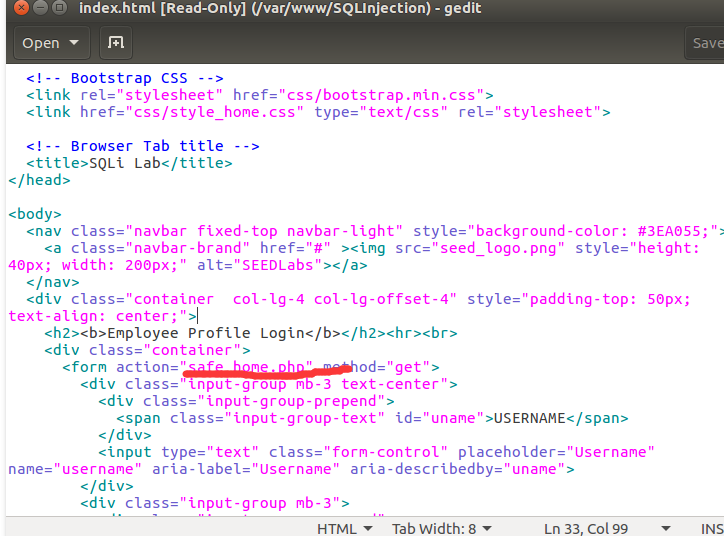
密码修改成功



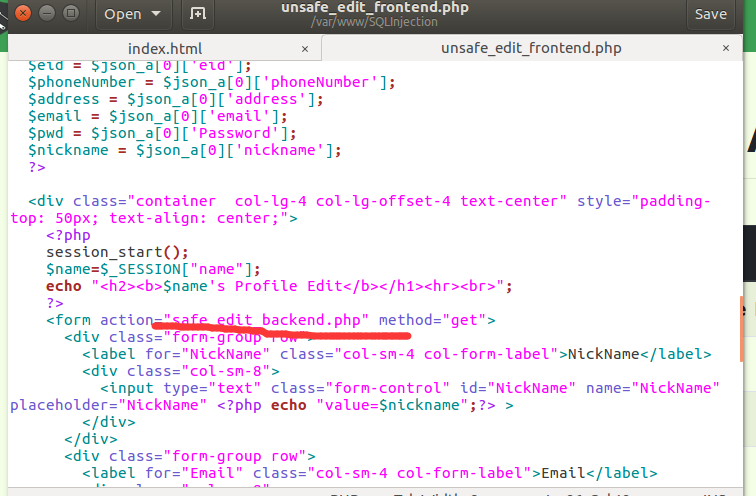
最后重新登录Boby账户

**Task 4: Countermeasure — Prepared Statement**

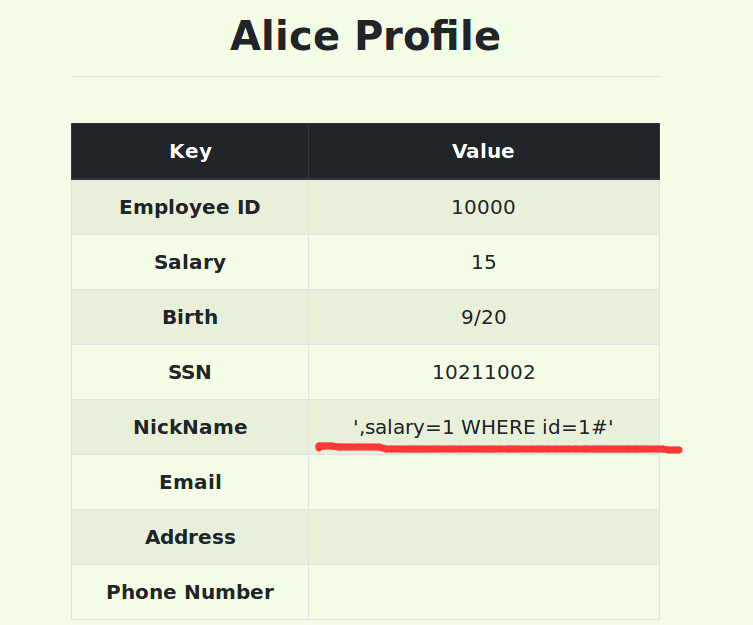
将网页的链接改为safe\_home.php

****

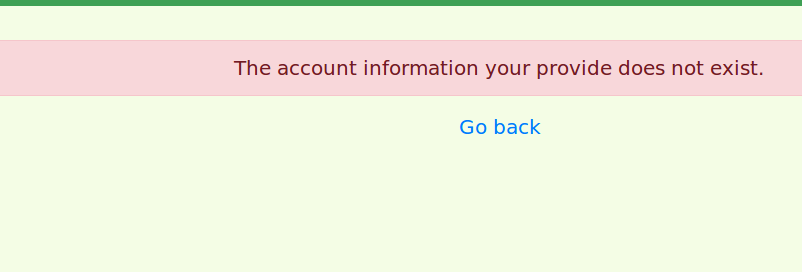
将unsafe\_edit\_frontend.php链接也改为safe的前端



进行修改年薪测试：发现输入修改语句，但是并没有成功



重新修改boby密码也没有成功：



**实验感想**

网络应用程序一般将数据存储在数据库中。当它们需要从数据库访问数据时，需要构造SQL语句并将语句发送给数据库执行。通常这些SQL语句包含不可信用户提供的数据。网络应用程序需要确保来自用户的数据不被当成代码，否则数据库可能会执行用户注入的指令。这次实验也让我深切体会到了这些。实验过程中也遇到了不少困难，特别是boby密码始终修改不正确是因为没有输入新的密码。但是最终完成了实验。