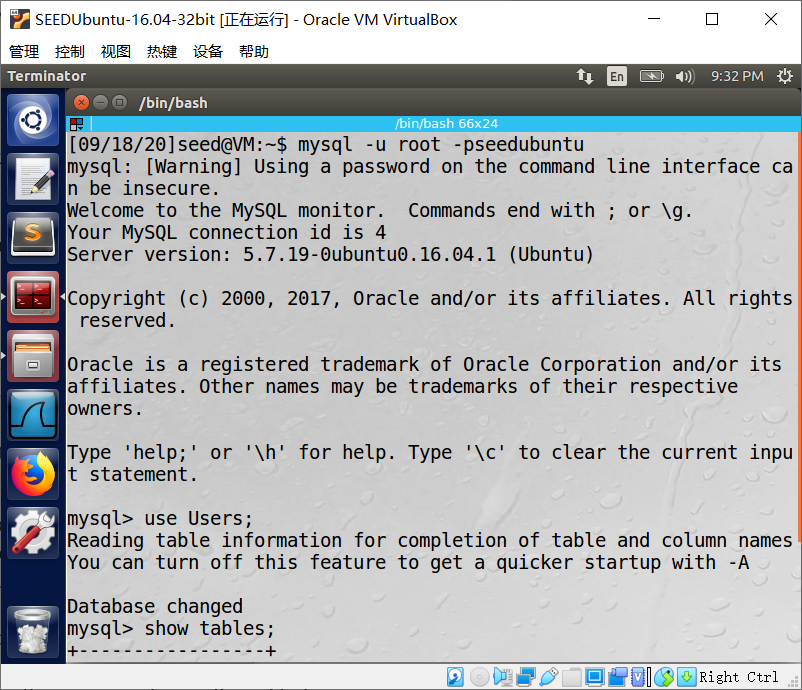
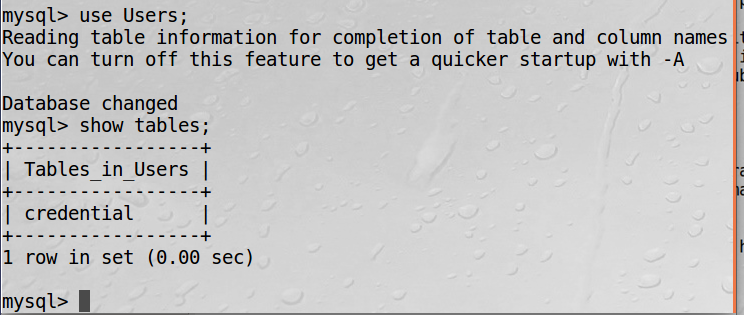
**Lab6实验报告**

57118110杨紫瑄

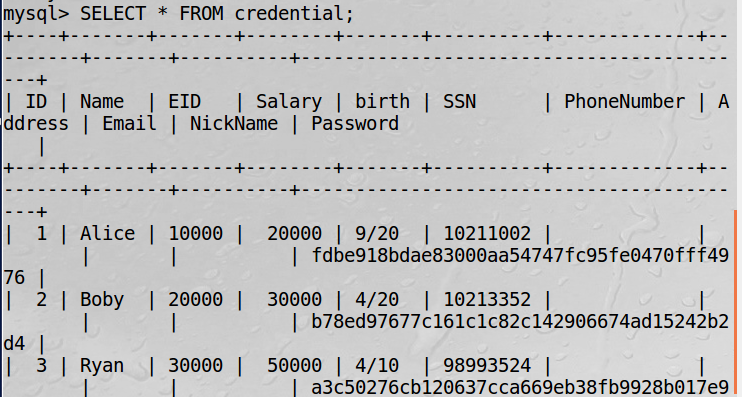
**Task1: GetFamiliarwithSQLStatements**



登录mysql



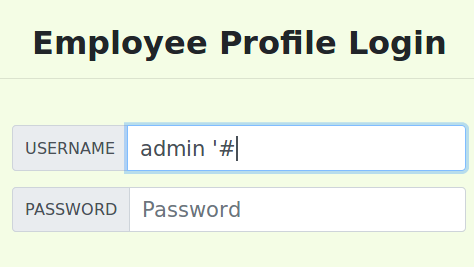
输入use Users 和show tables;



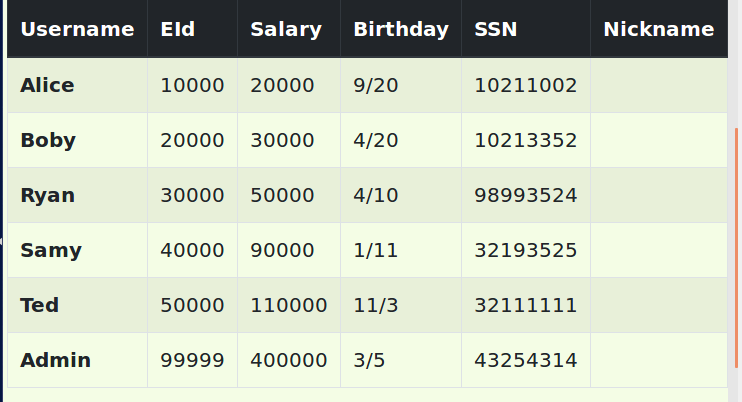
输入select \* from credential 展示每个用户信息

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

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

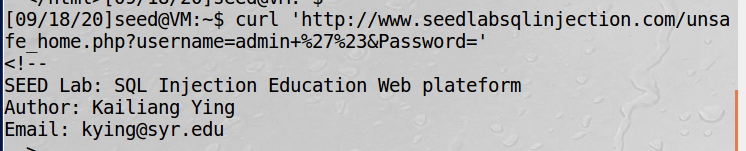


在Username 中输入 admin ‘# 可以不用密码登录，能够获得所有用户的信息





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

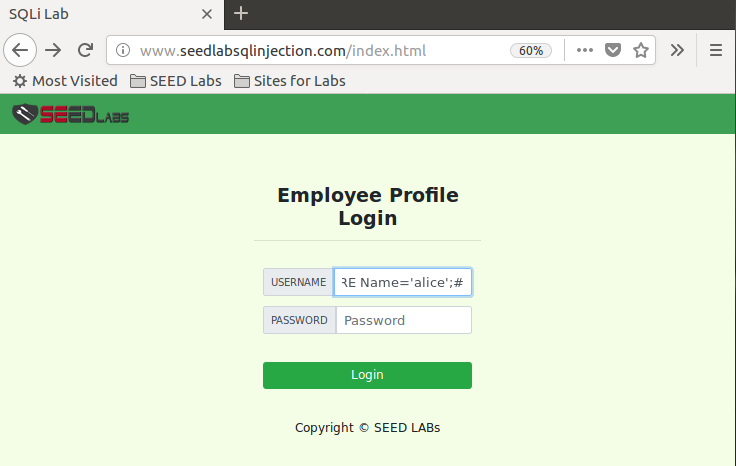


输入curl指令

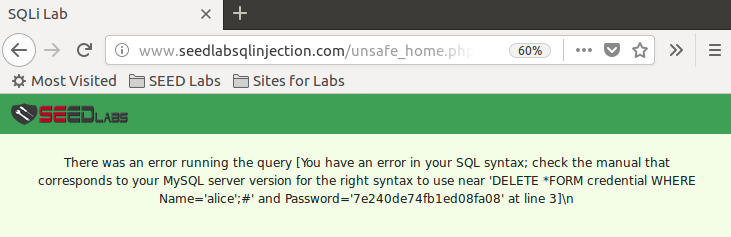


显示网页及其内容的代码，能够从中获取alice的信息

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



输入admin’,DELETE \* FROM credential WHERE Name=’alice’;#



发现攻击失败，因为以上输入试图执行两条SQL语句，但是mysql的query()函数不允许在数据库服务器上运行多条语句，所以这种攻击对mysql无效。

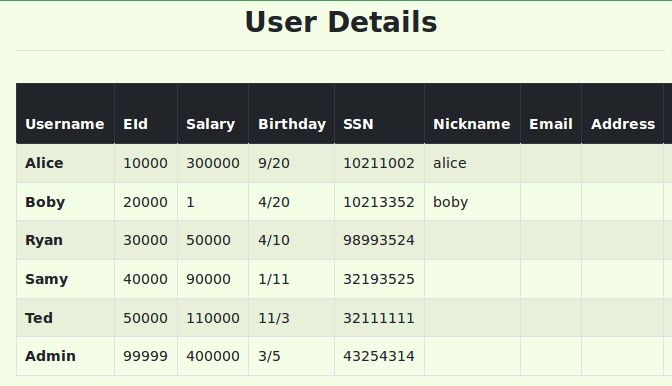
**Task 3: SQL Injection Attack on UPDATE Statement**

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



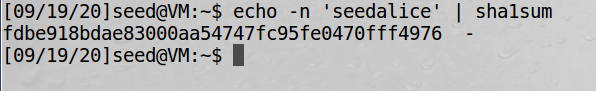
在alice’s profile edit 中的nickname里输入',salary='70000' where eid='30000'# 再重新登录发现alice的salary被更改

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

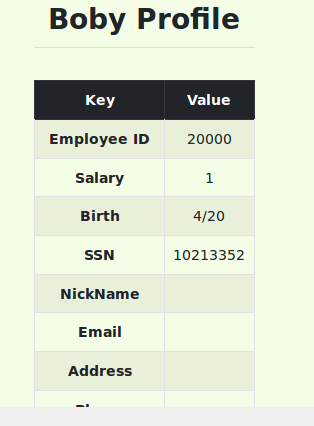


在alice’s profile edit中输入为',salary='1' where eid='20000' 再重新登录发现boby的salary被成功更改为1

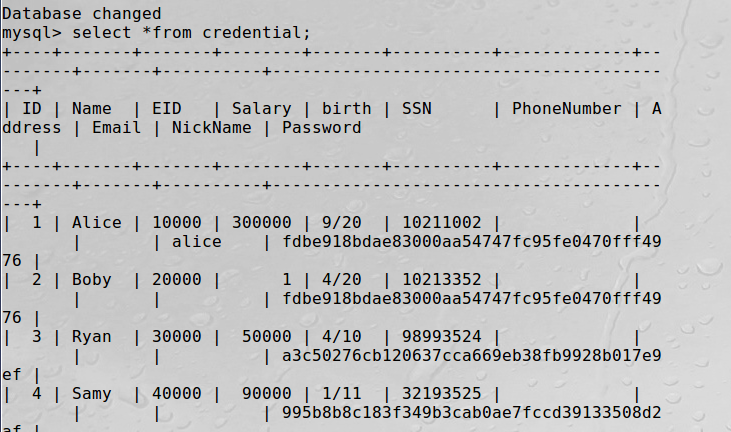
**Task 3.3: Modify other people’ password.**



使用echo -n ‘密码’ | sha1sum 命令获取密码seedalice的hash值

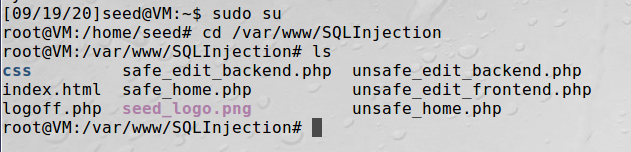


在alice’s profile edit中输入',password='fdbe918bdae83000aa54747fc95fe0470fff4976' WHERE eid=20000 #' 来更改boby的password，再用密码seedalice登录boby账号发现能够成功登录说明密码修改成功

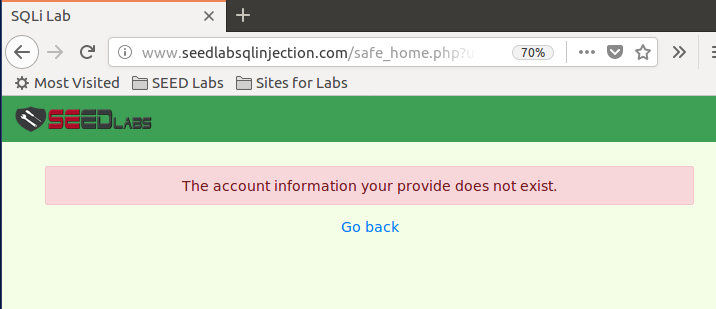


修改后的boby密码与Alice一样为seedalice

**Task 4: Countermeasure — Prepared Statement**



进入终端，进入/var/www/SQLInjection文件夹，发现有一个已经编译的版本



将先前网页链接中unsafe\_home.php修改为safe\_home.php并登录修改链接后的网页：

[www.seedlabsqlinjection.com/safe\_home.php?username=admin'#&Password=](http://www.seedlabsqlinjection.com/safe_home.php?username=admin'#&Password=)

发现失败，

实验总结：

通过此次试验我们对SQL注入攻击有了更加深刻的了解，同时我们也对SQL攻击语句的格式以及攻击的方法更加熟悉，通过对这些攻击的深入了解，也让我们对如何阻止防御此类攻击有了概念，对网络空间安全的维护有着非常重要的作用。在本次实验中，我因为攻击语句的格式问题屡屡遇到无法实现攻击的情况，好在最后及时纠正了格式使实验成功完成。