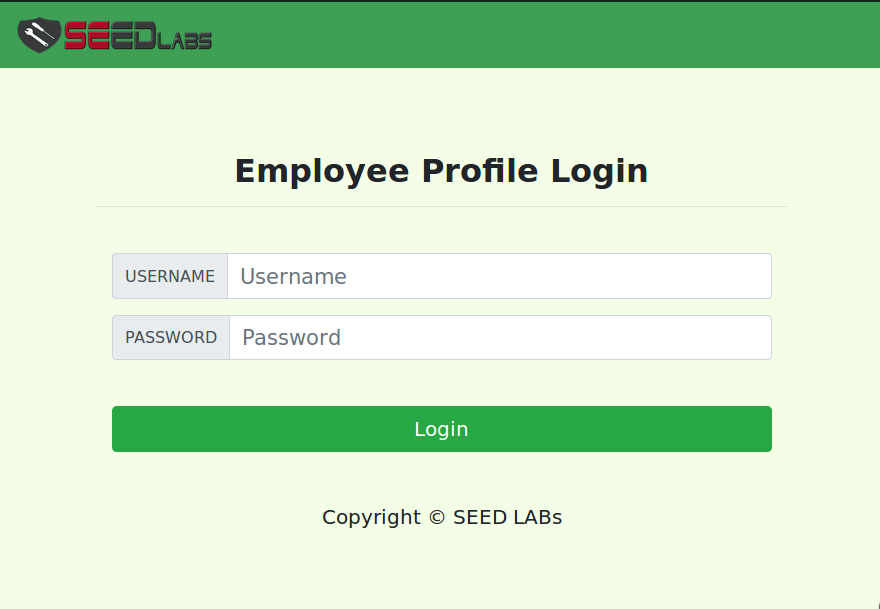
Lab6-report

57118113 蔡义涵

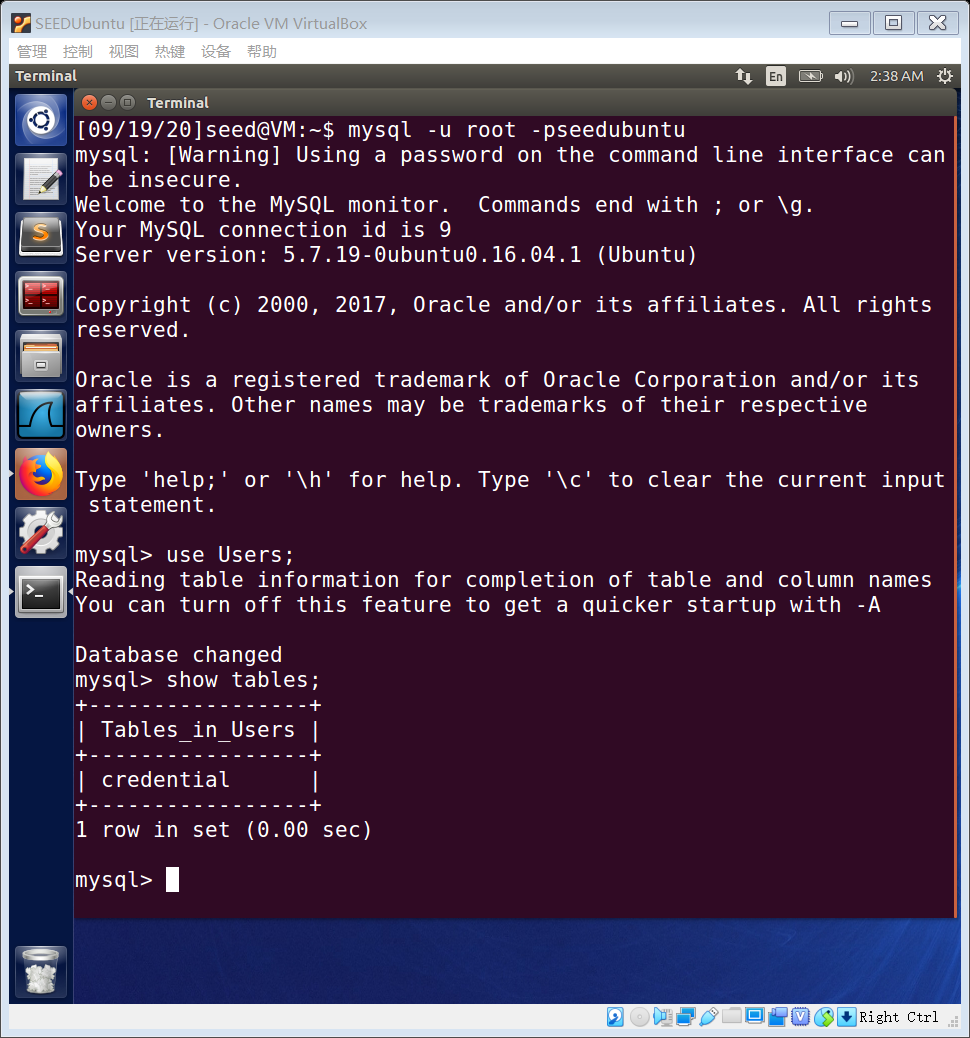
**2 Lab Environment**

在虚拟机中访问<http://www.SEEDLabSQLInjection.com>



成功进入web程序

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



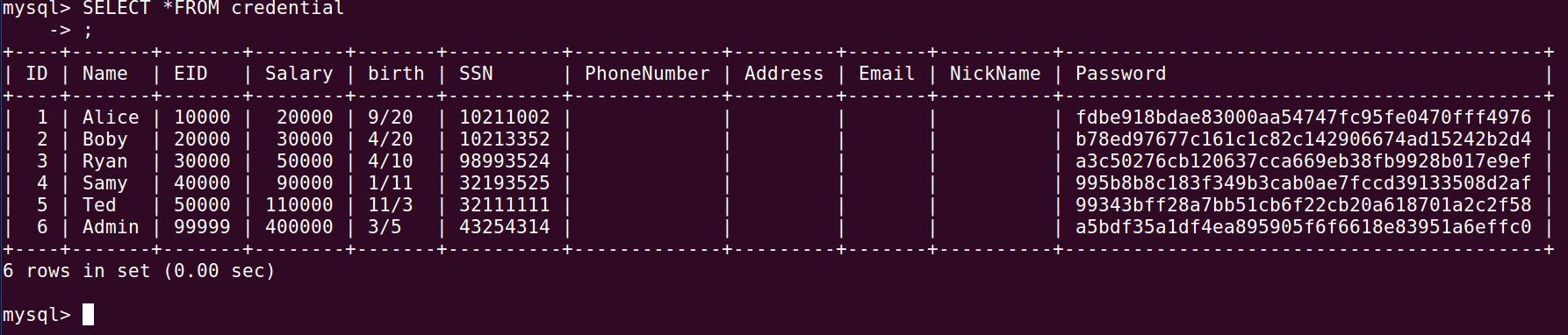
成功加载现有数据库并查看了现有的表。

下面我们使用命令行查看employee Alice的信息：

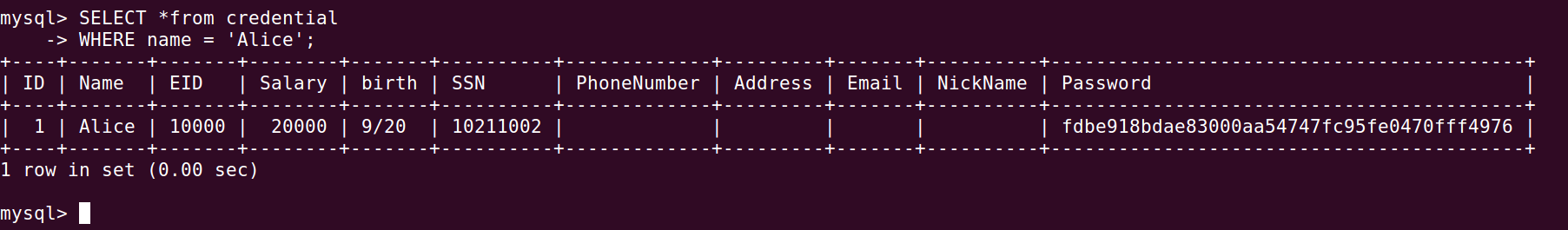
先输入命令

SELECT \*FROM credential

；



再输入SELECT \* FROM credential WHERE name = ‘Alice’;得到Alice的信息

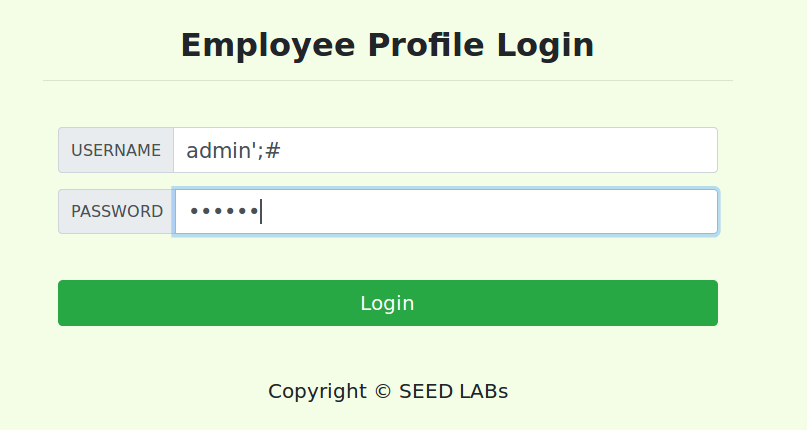


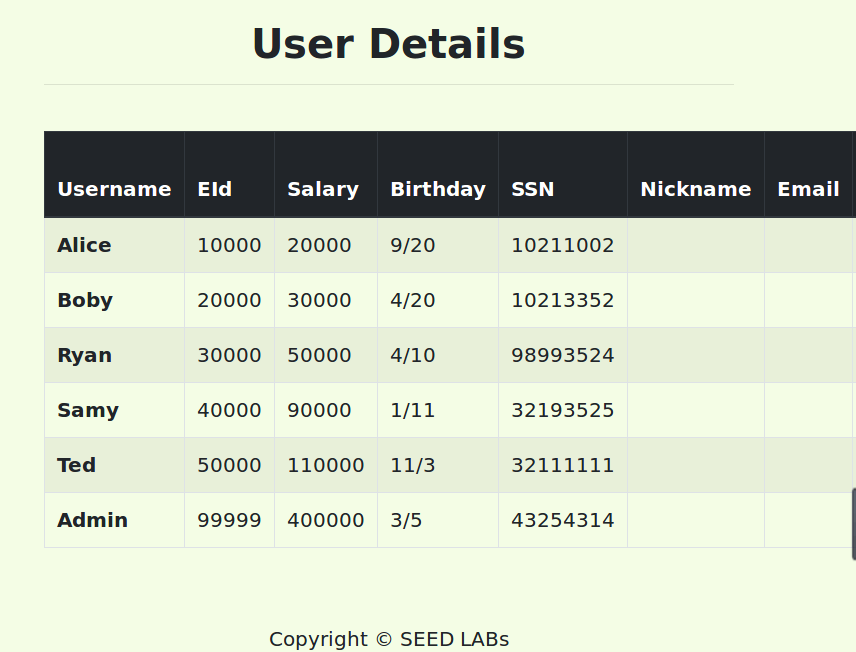
**3.2 Task 2: SQL Injection Attack on SELECT Statement**

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

在命令行窗口中输入SELECT \*FROM credential WHERE=’Admin’:#’ AND password =’’;

然后打开Web应用程序，输入USERNAME为admin’;#，密码随意。





SQL注入攻击成功

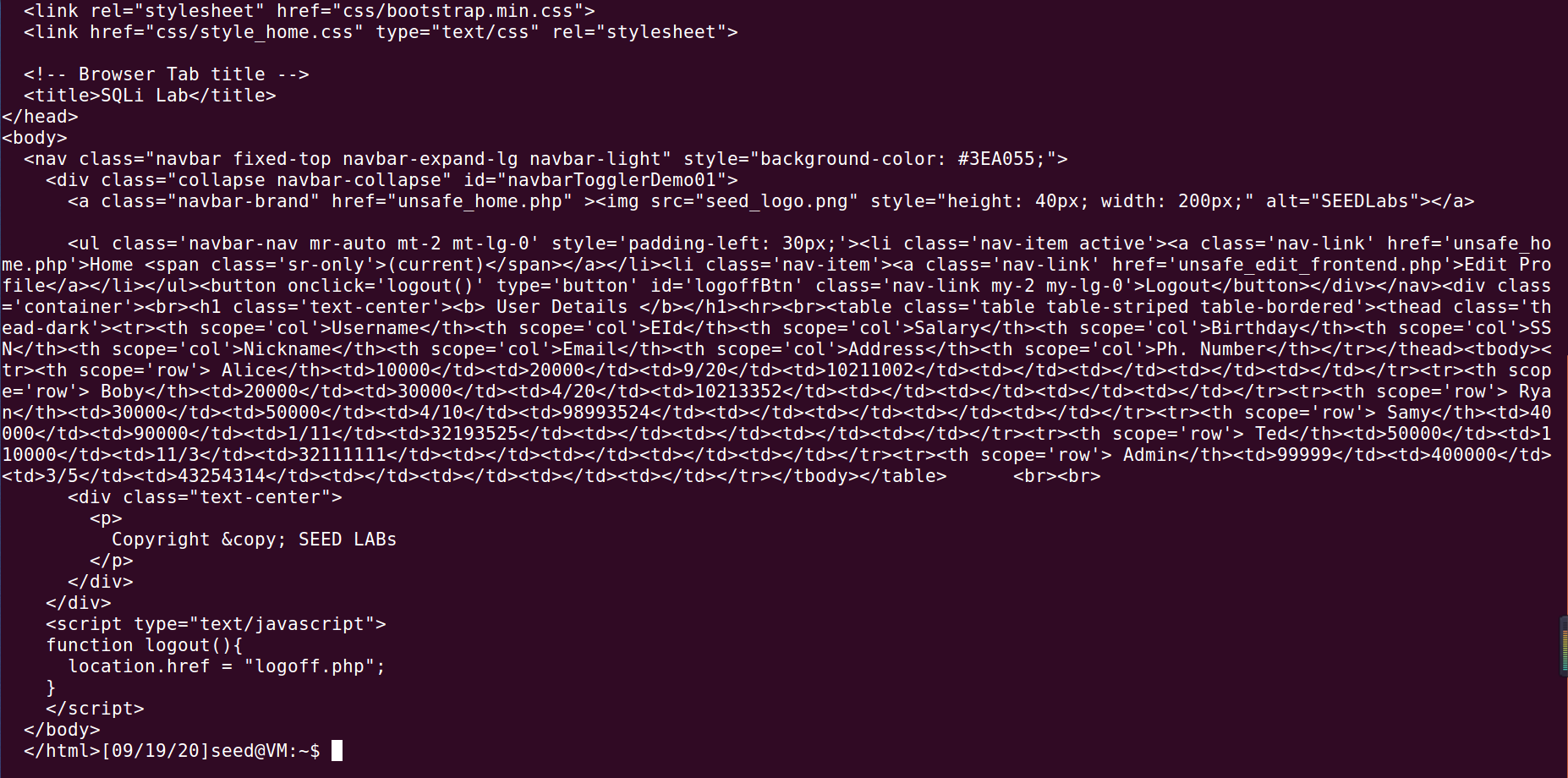
**Task 2.2: SQL Injection Attack from command line.**

首先，我们使用命令行登录网页的Admin账号

Curl 'http://www.seedlabsqlinjection.com/unsafe\_home.php?username=’Admin%27%23'&Password='a''

得到以下结果：

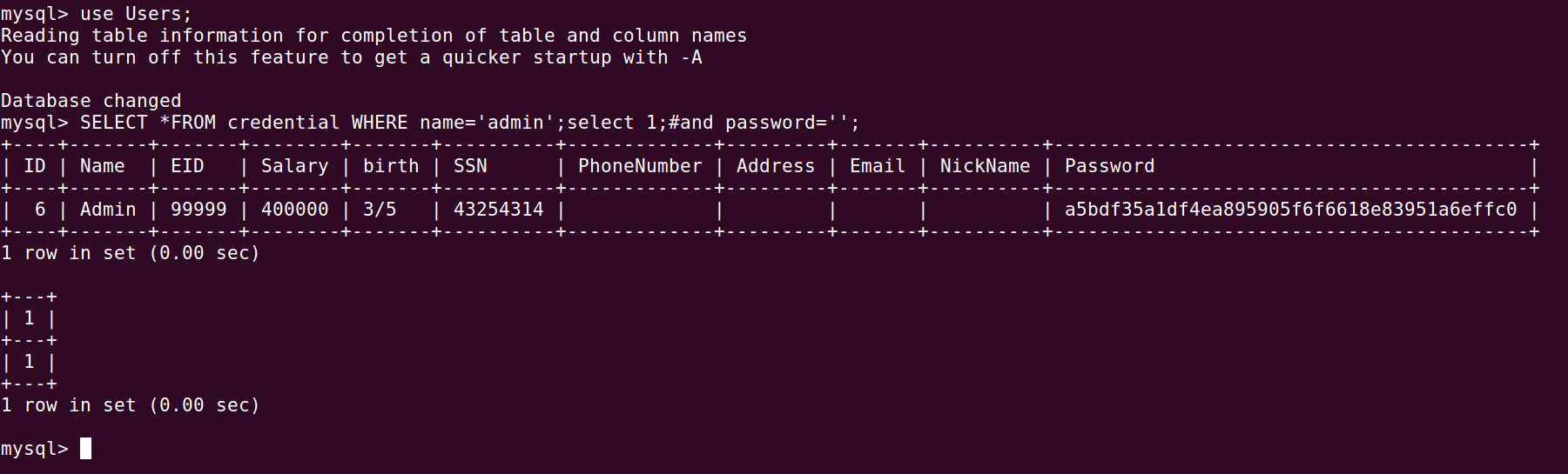




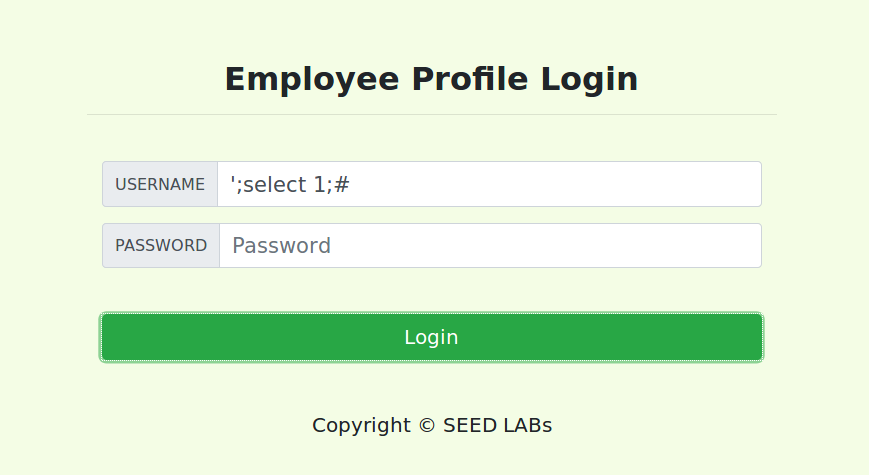
成功登录，表明攻击成功

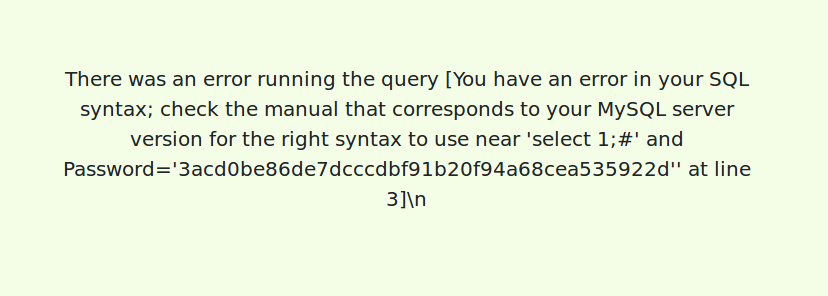
**Task 2.3: Append a new SQL statement.**

首先，我们在命令行中输入以下命令SELECT \*FROM credential WHERE name='admin';select 1;#and password='';



随后我们再使用’;select 1;#账户登录网站



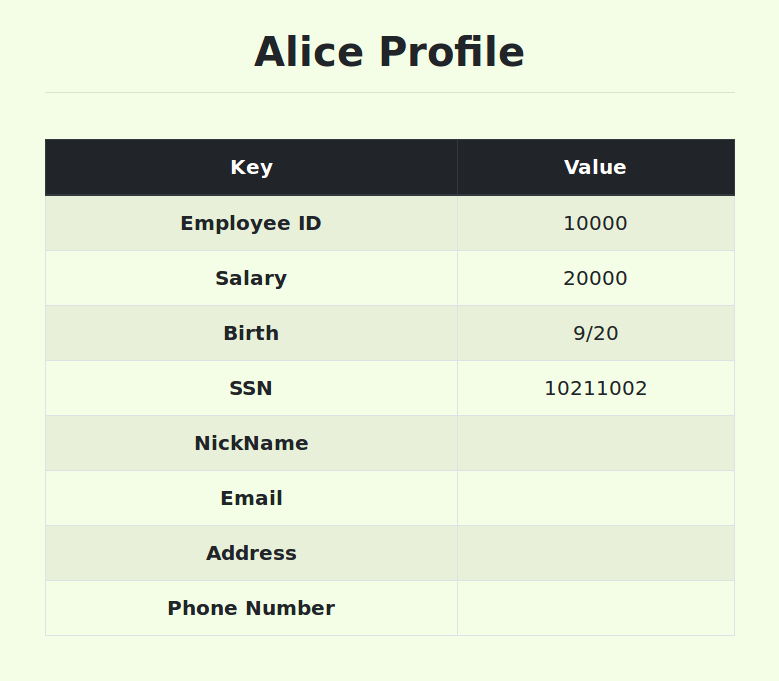


发现注入攻击失败，原因是两条语句会被数据库识别，导致攻击失败。

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

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

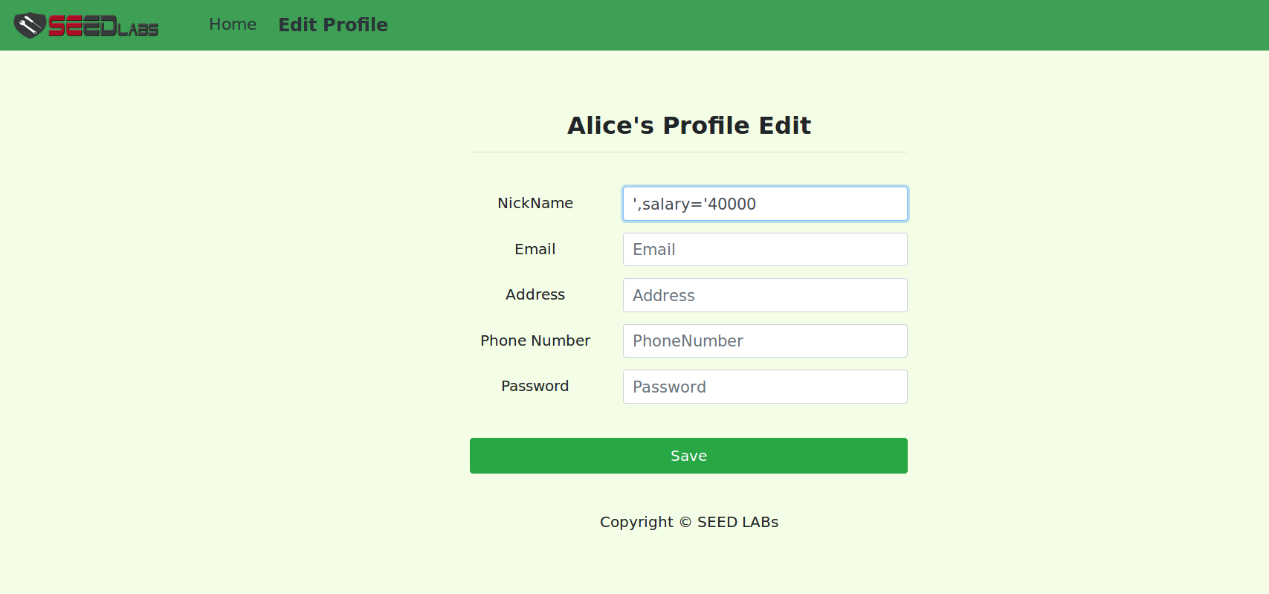
我们先登录Alice账号，查看当前信息



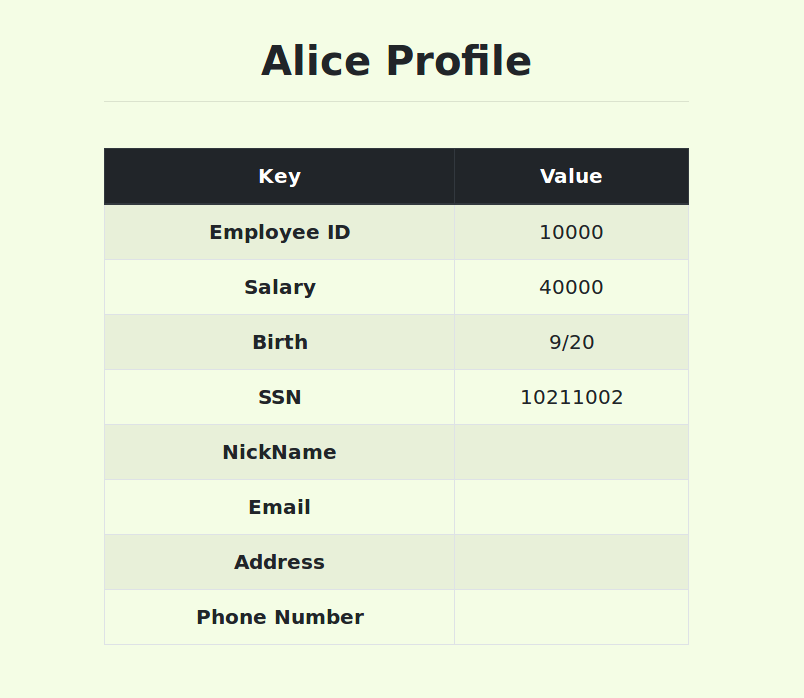
当前工资为20000

接下来我们进行SQL攻击

点击Edit Profile栏目，修改个人信息

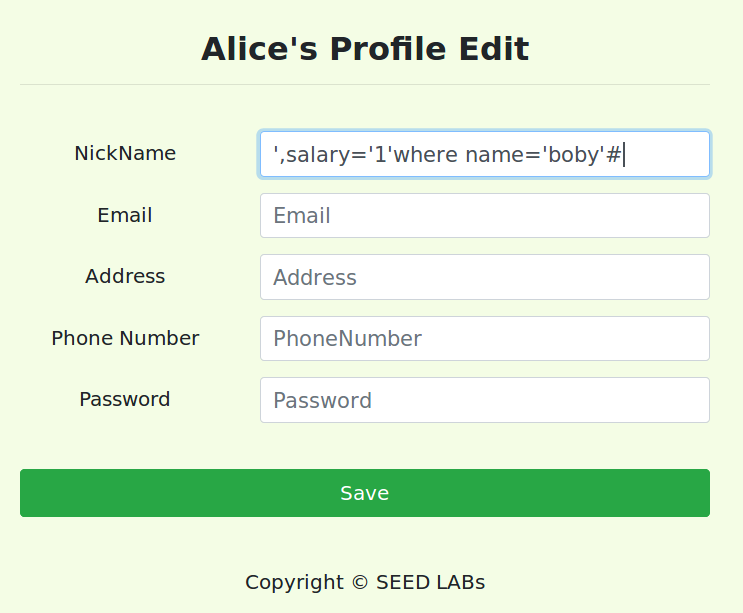


保存后发现，salary已被修改

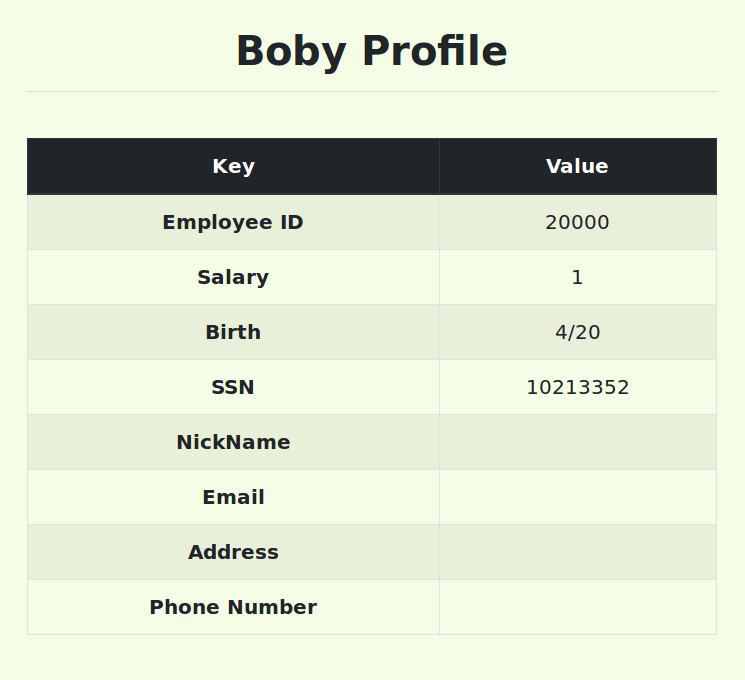


**Task 3.2: Modify other people’ salary.**

我们进入Alice账号，修改个人信息NickName为‘，salary=’1’ where name=’boby’#



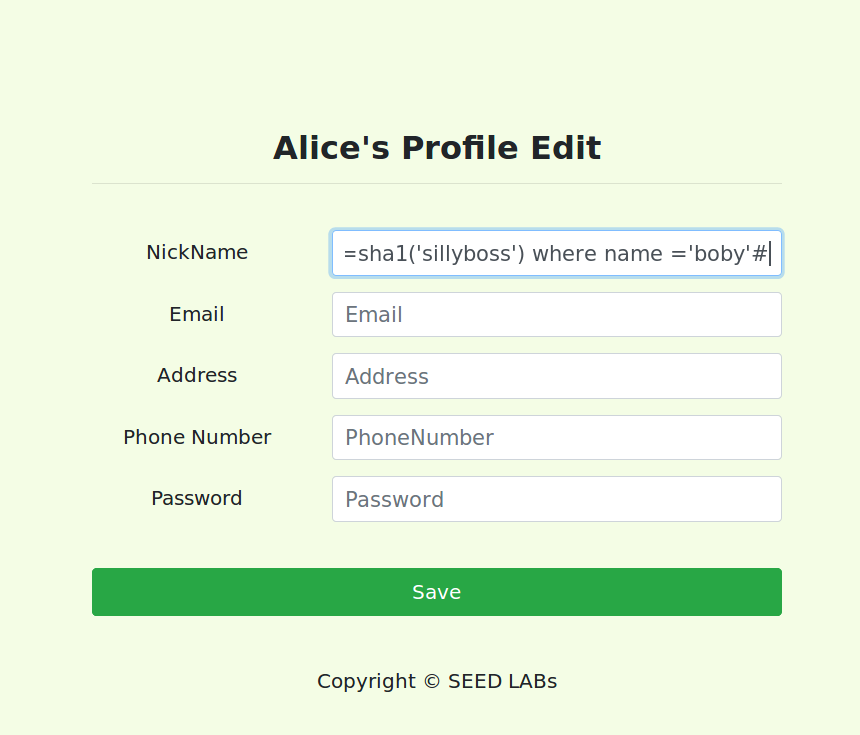
然后我们登录Boby账号检验结果



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

进入Alice账号Edit Profile栏目，修改个人信息Nickname为

’,password=sha1(‘sillyboss’) where name=’boby’#

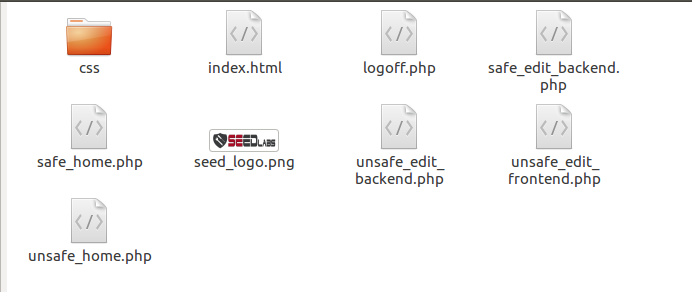


修改后，Alice成功登录

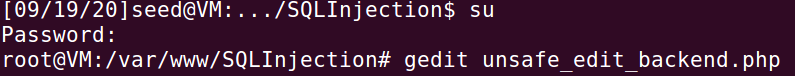
**3.4 Task 4: Countermeasure — Prepared Statement**

下面，我们修改这个Web程序使得代码和数据有清晰的区分。

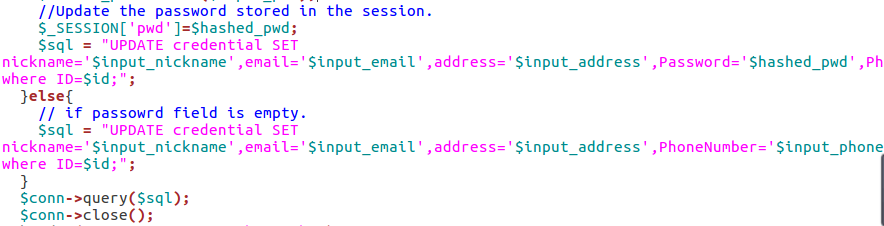
首先，我们进入/var/www/SQLInjection目录，查看文件夹里的文件。



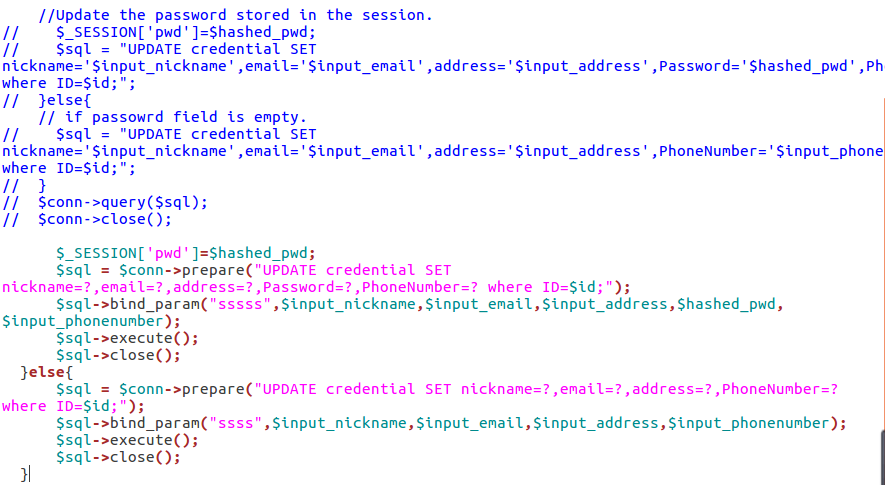
下面，我们先进入root模式，然后修改unsafe\_edit\_backend.php文件：



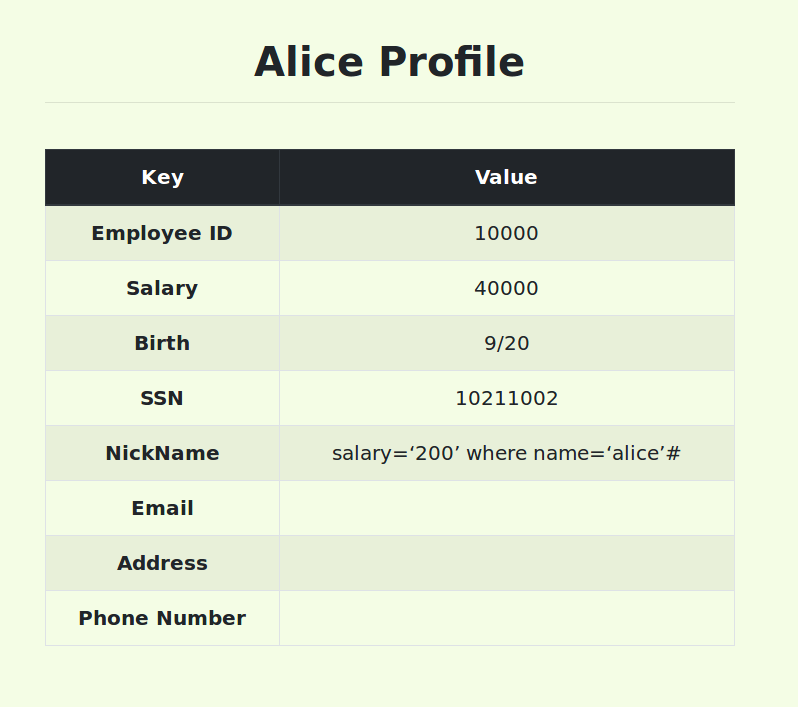
修改前



修改后



回到网页中重复Task 3.2，实验结果如下：



SQL注入攻击失败，防御成功。