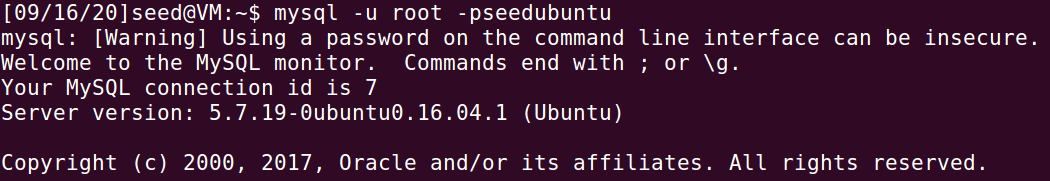
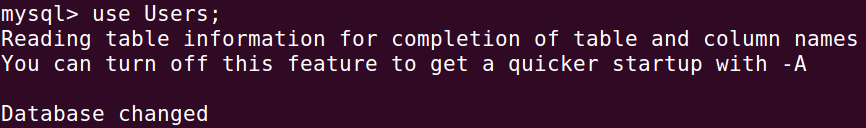
**Lab[6]-report**

57118123刘康辉

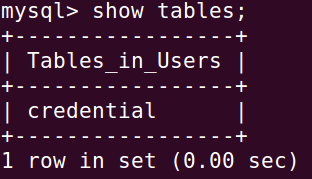
**Task 1**



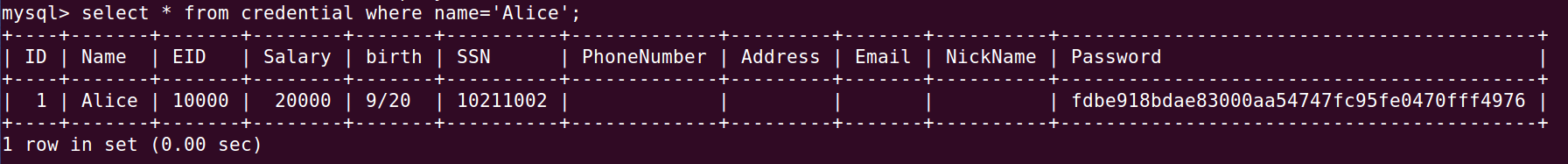
输入指令和密码进入Mysql的操作界面，如上图所示。



输入指令进入已经创建好的Users数据库，如上图所示。



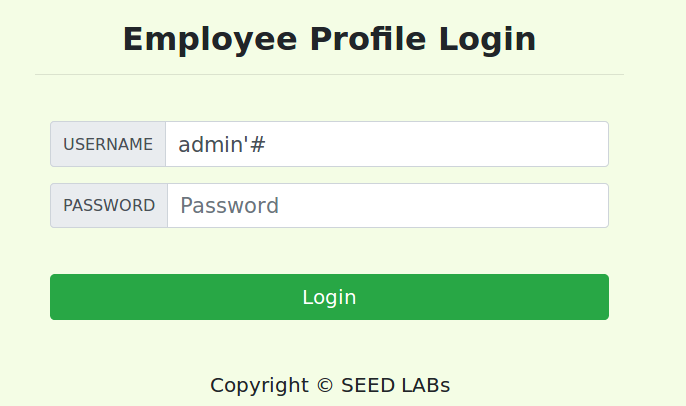
输入指令查看Users数据库中的表信息，结果如上图所示。

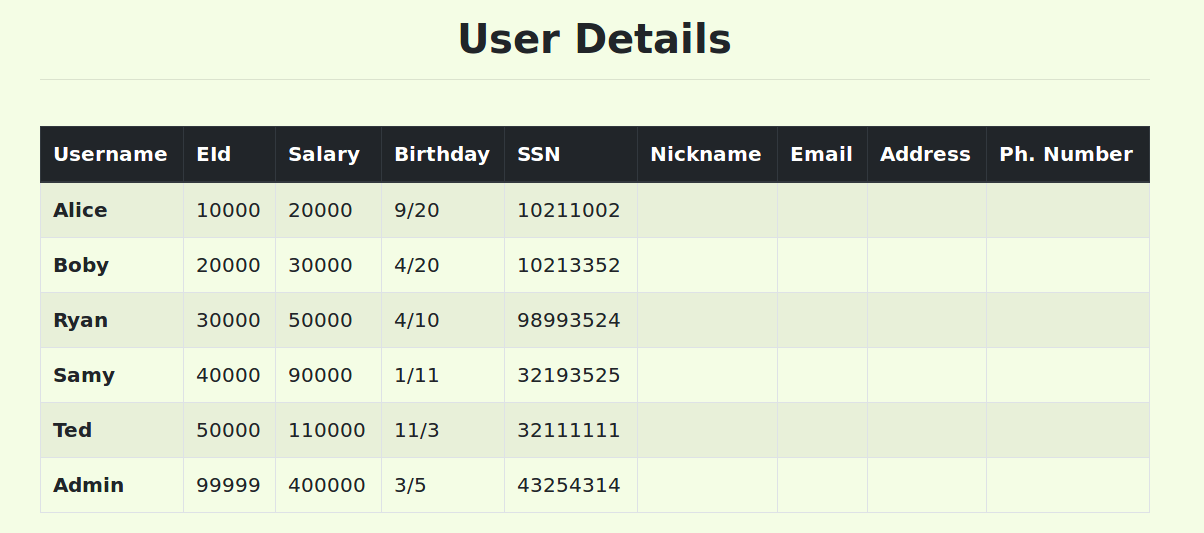


输入指令查看在Users数据库的credential表中的Alice的相关信息，结果如上图所示。

**Task 2**

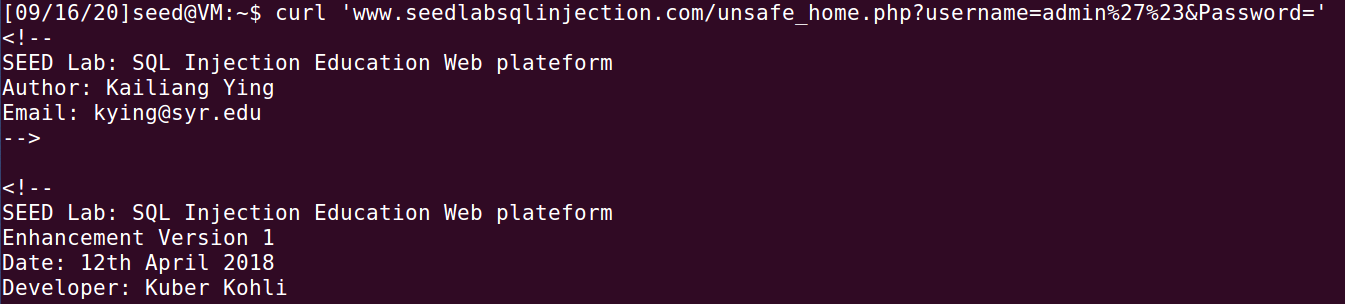
**Task 2.1**

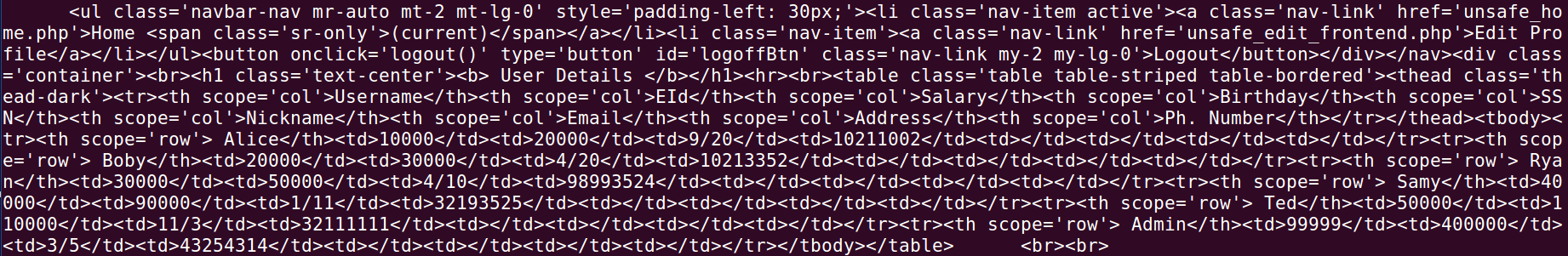




在初始页面的Username中输入**admin’#**，然后点击登录，发现能够在不输入密码的情况下通过身份验证，输出所有用户的相关信息。根据实验结果可知，通过网页的SQL注入攻击成功。

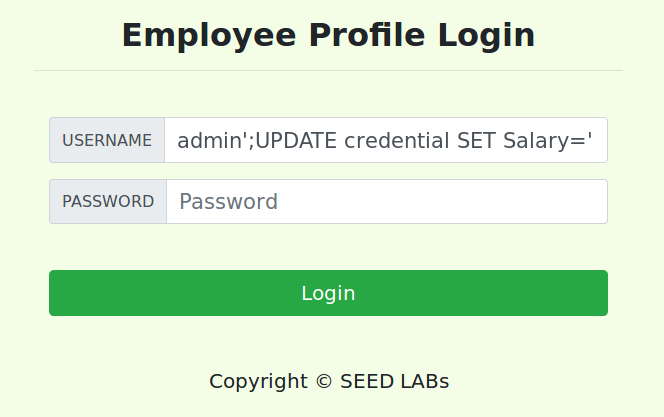
**Task 2.2**

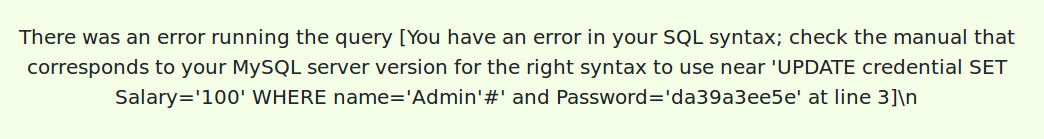




在终端的命令行中使用curl命令，而指定的url为**'www.seedlabsqlinjection.com/unsafe\_home.php?username=admin%27%23&Password='**，运行后输出的信息中包含所有用户的相关信息。根据实验结果可知，通过命令行的SQL注入攻击成功。

**Task 2.3**

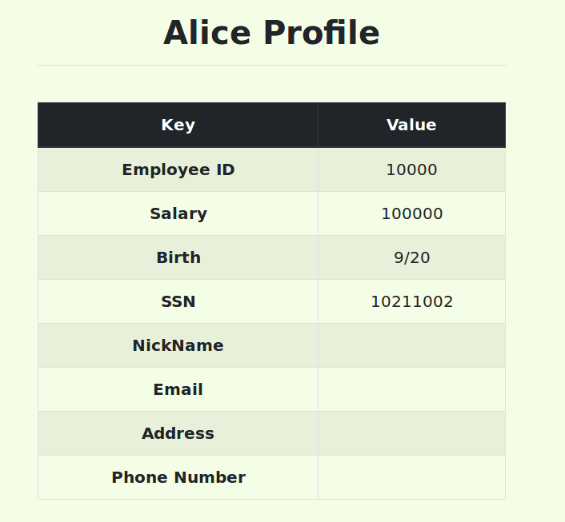
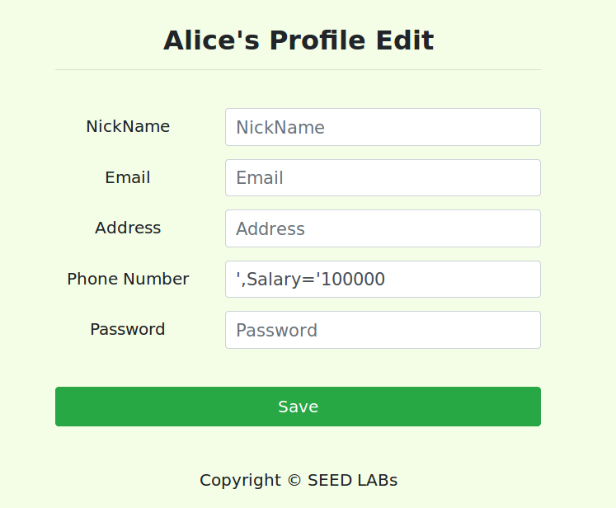




在初始页面的Username中输入**admin’;UPDATE credential SET Salary='100' WHERE name='Admin'#**，然后点击登录，发现会出现错误页面。根据实验结果可知，追加新的SQL语句攻击失败，原因是Mysql中的query()函数正常情况下不支持执行多条语句。

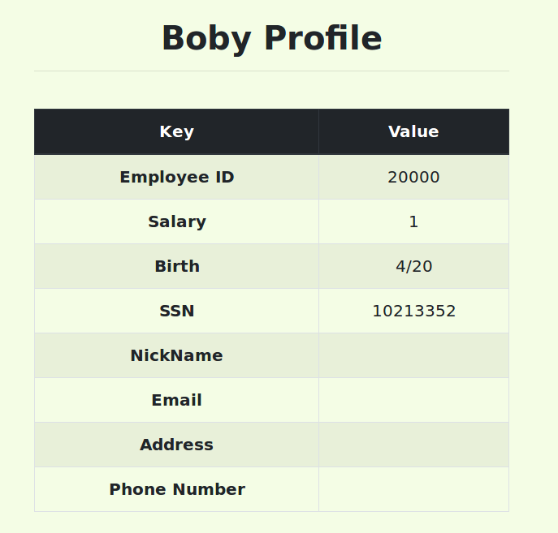
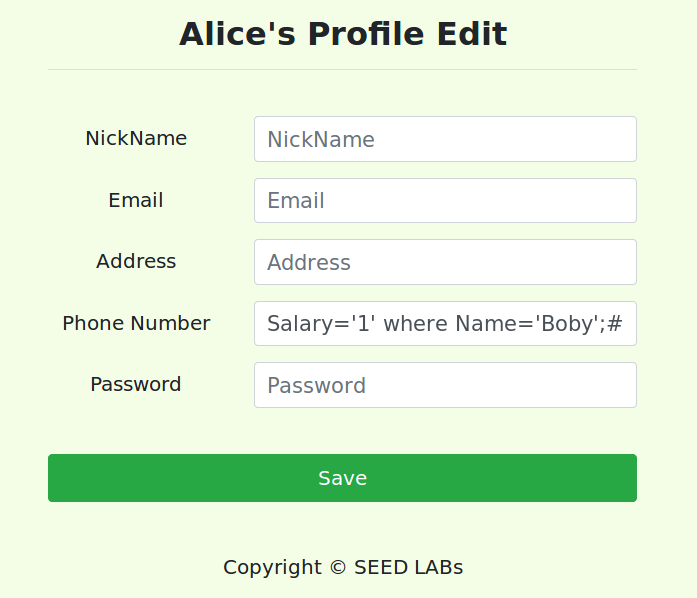
**Task 3**

**Task 3.1**



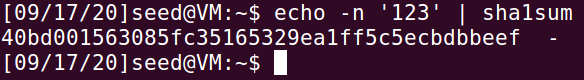
首先登录Alice的账户，然后进入Edit Profile界面，在Phone Number中输入**’，Salary=’100000**，点击保存，发现Alice的工资已经被修改为100000。根据实验结果可知，修改自己的工资成功。

**Task 3.2**

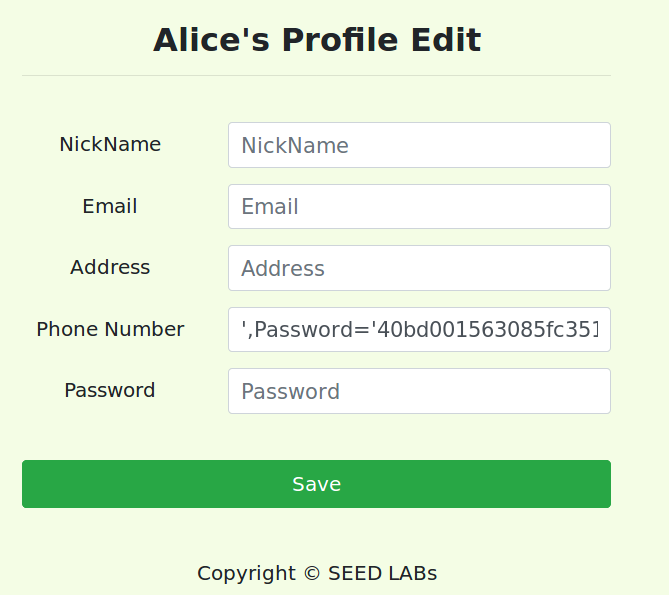


首先登录Alice的账户，然后进入Edit Profile界面，在Phone Number中输入**’，Salary=’1’ where Name=’Boby’;#**，点击保存。登录Boby的账户，发现Boby的工资已经被修改为1。根据实验结果可知，修改其他人的工资成功。

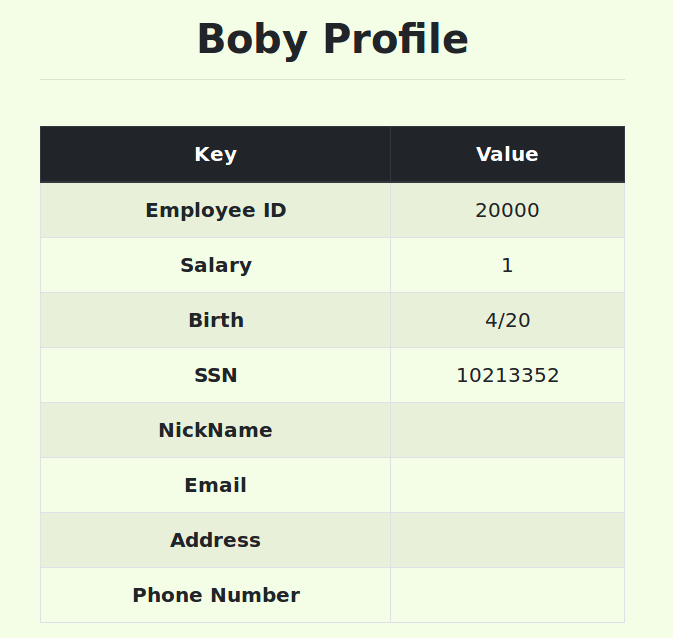
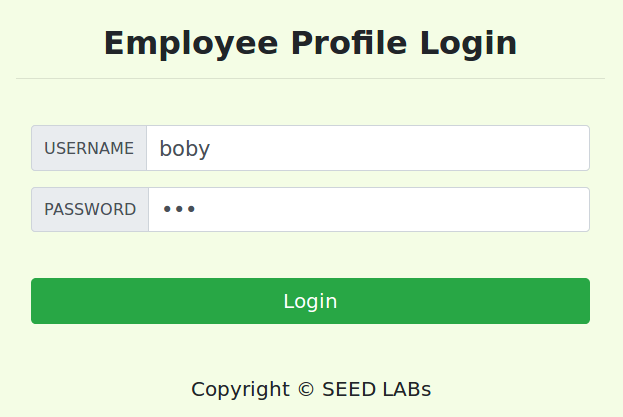
**Task 3.3**



在终端的命令行中输入指令计算**123**的SHA1加密后的值，如上图所示。

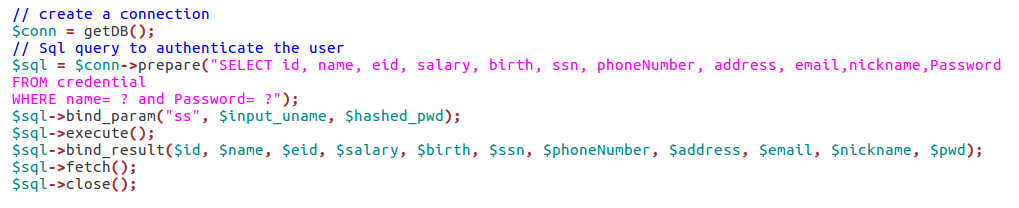


首先登录Alice的账户，然后进入Edit Profile界面，在Phone Number中输入**',Password='fce06cc54b414aecb0a418eb3a8b759a0eeccbd5' where Name='Boby'#**，点击保存。

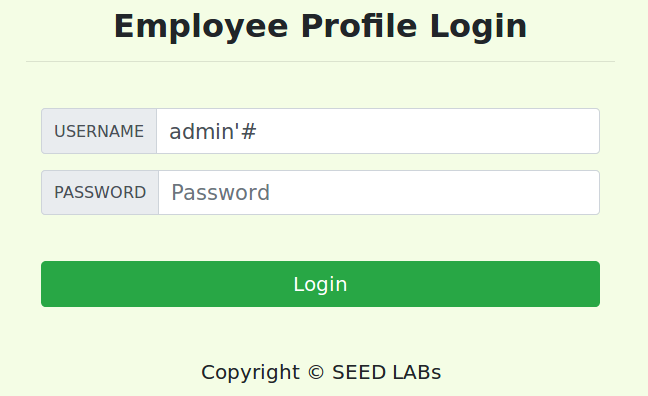


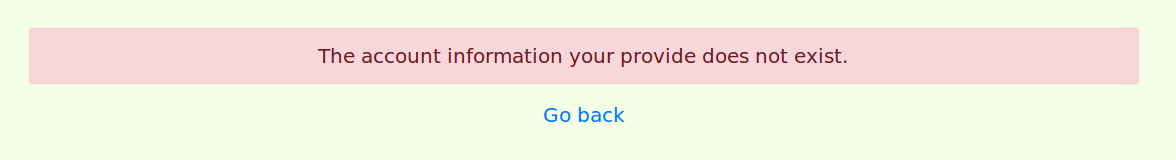
在初始页面的Username中输入boby，Password中输入123，然后点击登录，发现能够通过身份验证。根据实验结果可知，修改其他人的密码攻击成功。

**Task 4**



修改/var/www/SQLInjection中unsafe\_home.php文件，如上图所示。





在初始页面的Username中输入**admin’#**，然后点击登录，发现会出现错误页面。根据实验结果可知，SQL注入漏洞已经被修复，无法成功进行SQL注入攻击。