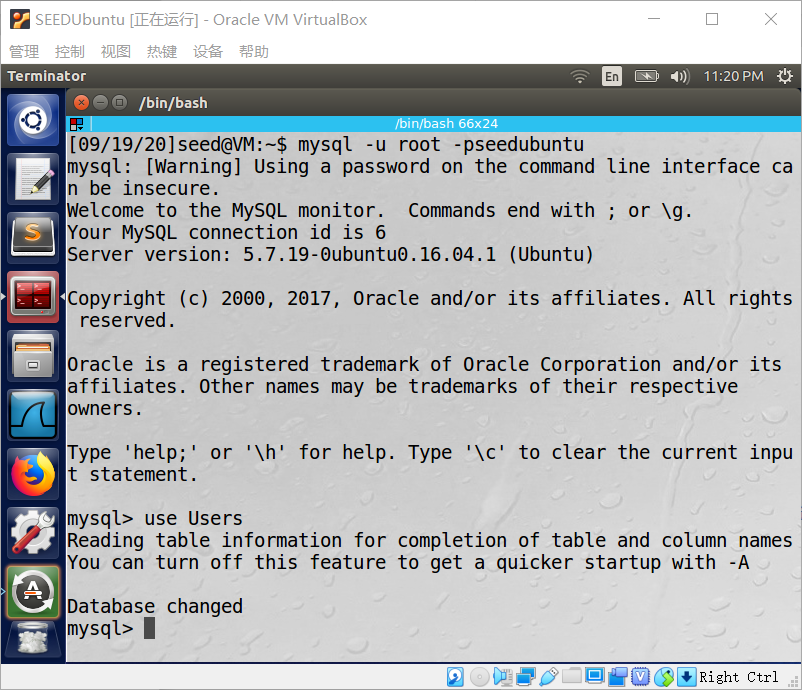
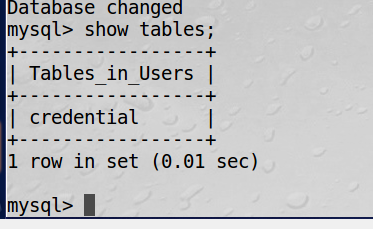
**实验六：SQL Injection Attack Lab**

57118135许皓钦

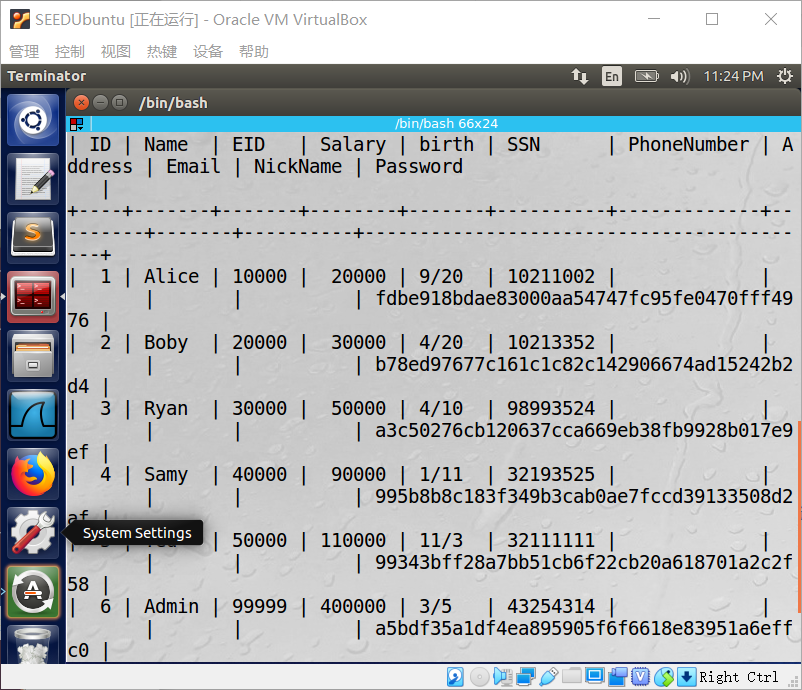
**Task 1: Get Familiar with SQL Statements**

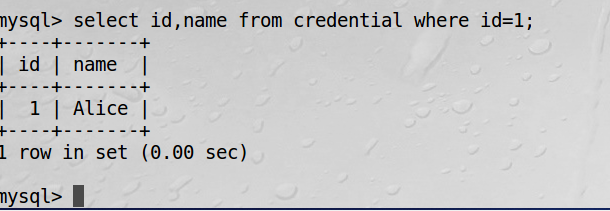
首先进入mysql，输如use Users进入users数据库，show tables显示表格





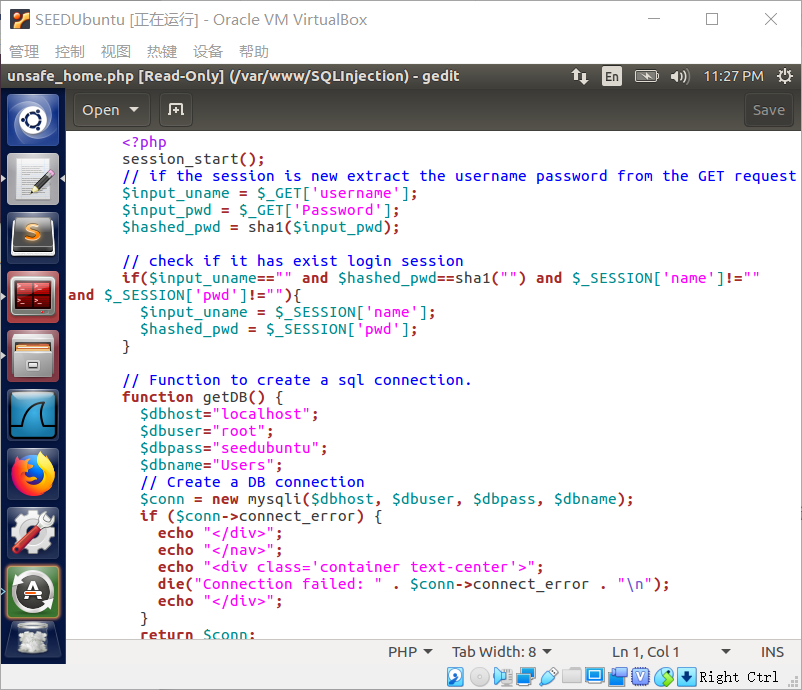
再使用select \* from credential 调出所有的数据。

使用select id,name from credential where id=1调出部分关于Alice的数据。



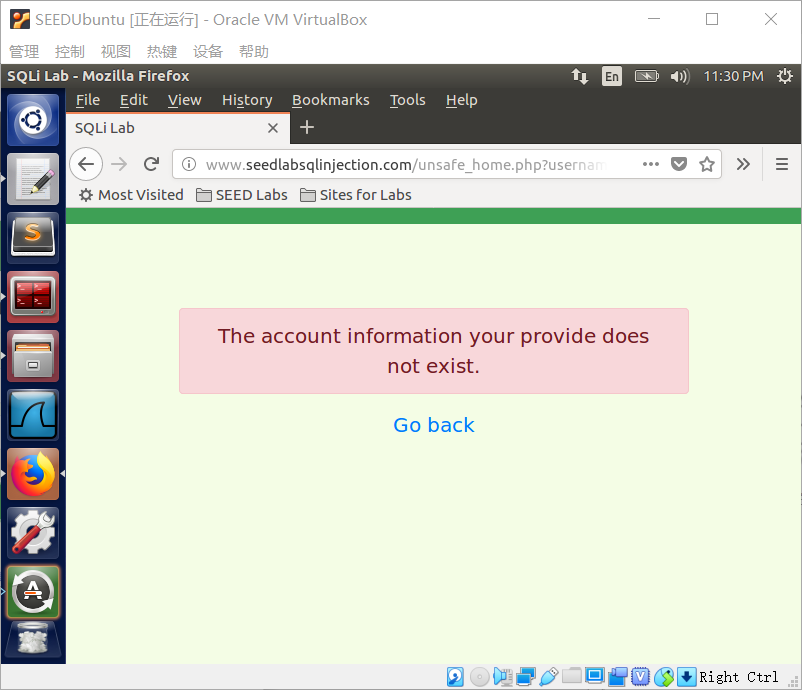
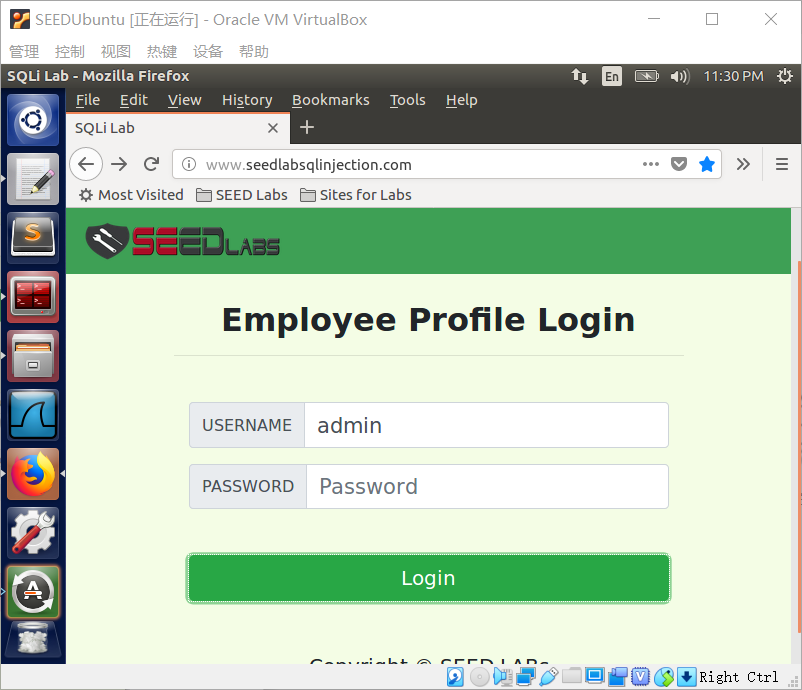
**Task 2: SQL Injection Attack on SELECT Statement**

首先找到/var/www/SQLInjection 目录下的unsafe\_home.php，观察其代码的组成。发现可以利用的漏洞，原理在后。

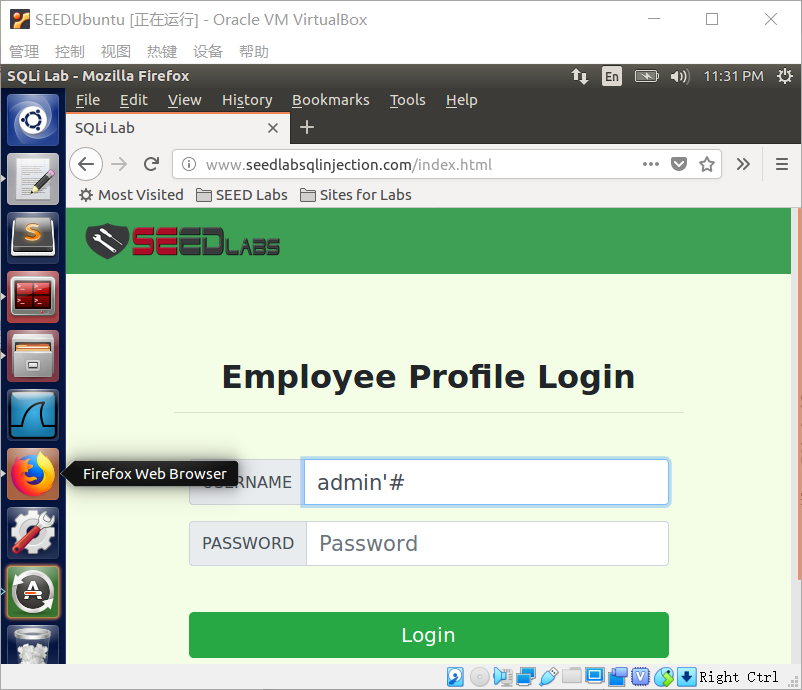


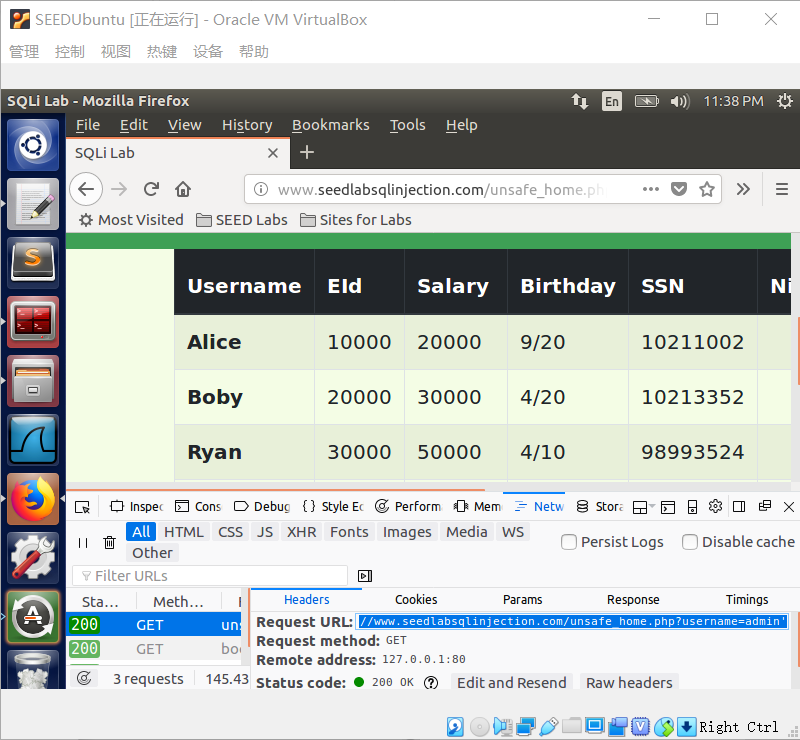
**Task 2.1: SQL Injection Attack from webpage**.

首先使用admin无密码登录，发现无法登录成功。



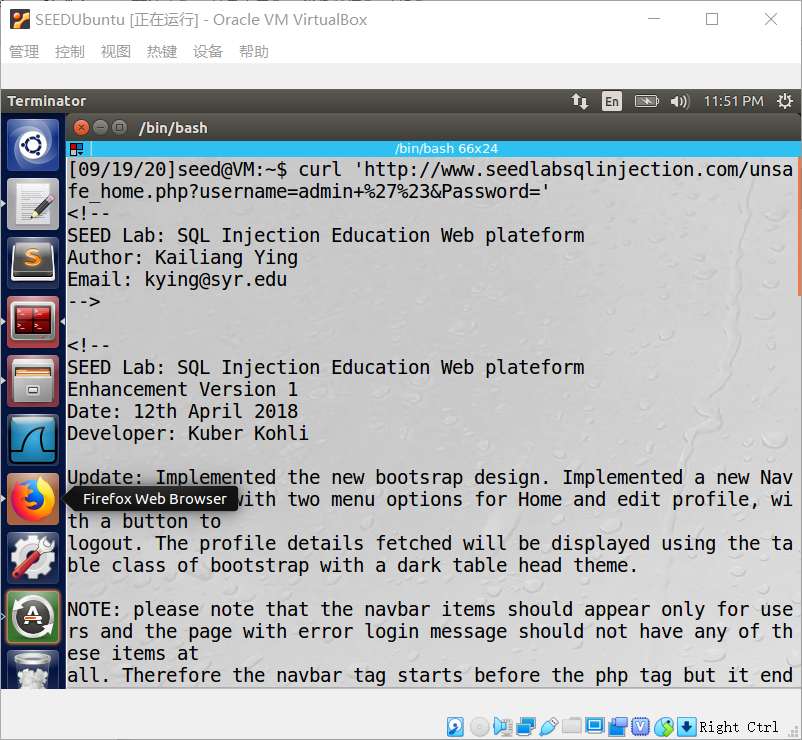
将用户名输入为admin’#，点击login，发现能够进入系统，在network tool中发现用户名输入的是admin’#，没有输入密码。



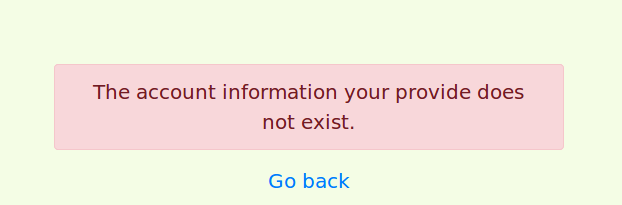
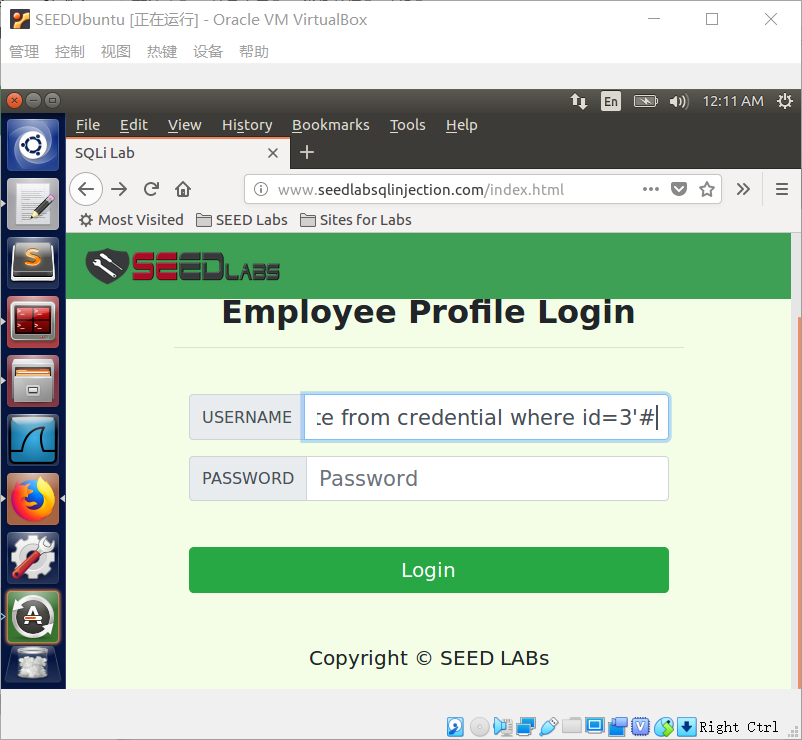


**Task 2.2: SQL Injection Attack from command line**.

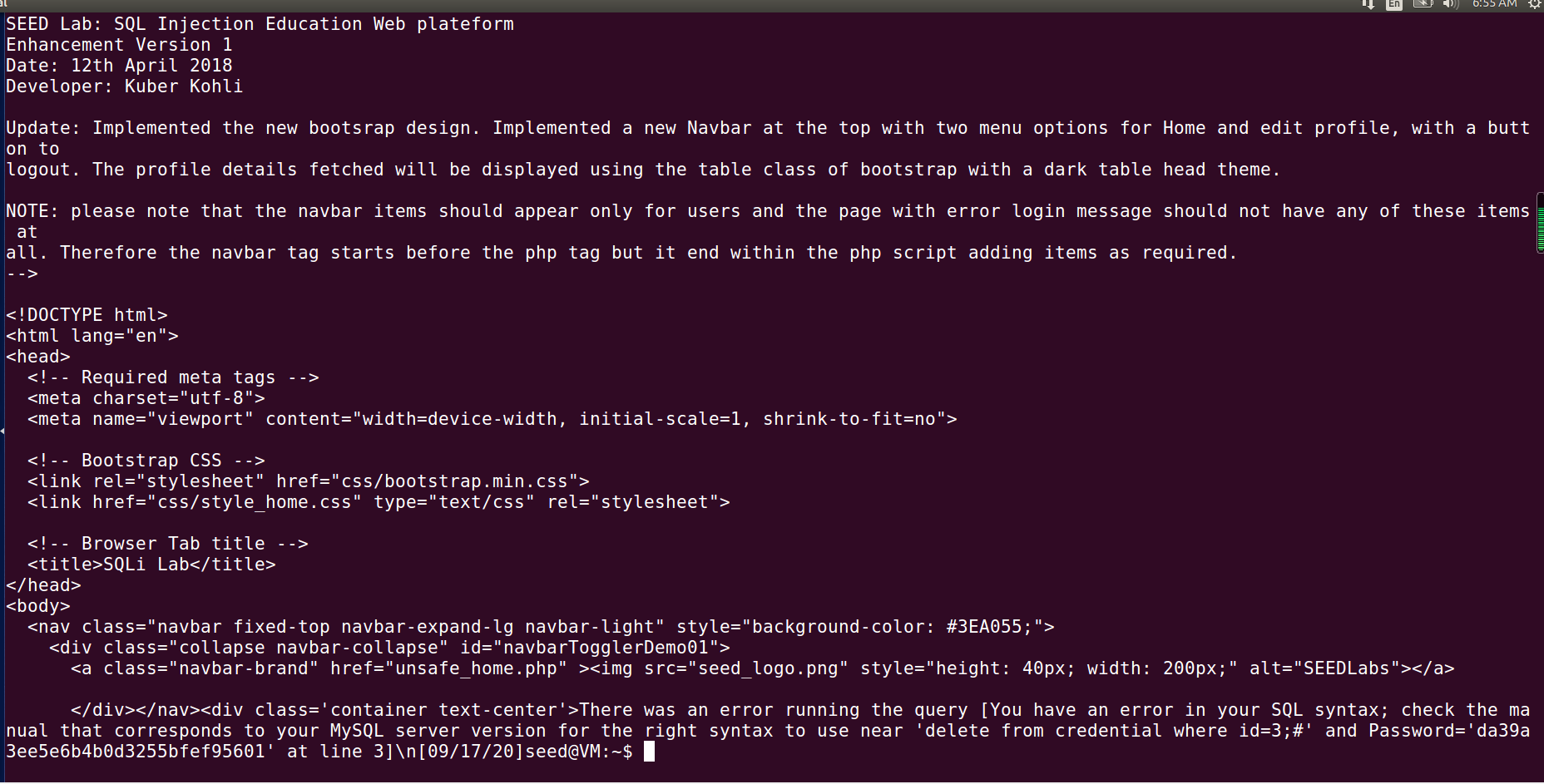
直接在终端输入curl+以上network tool捕获到的URL即可。



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

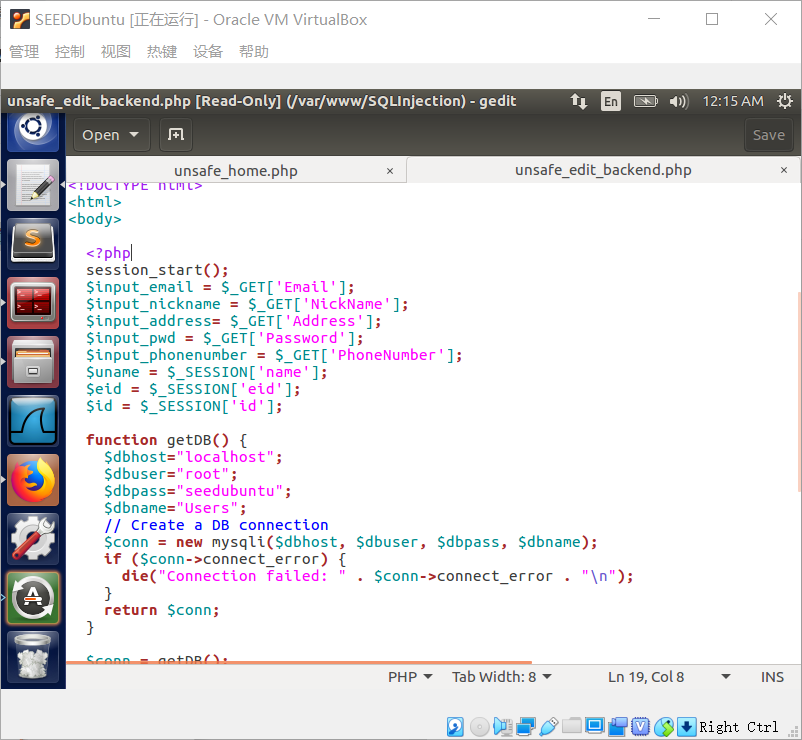
如下，使用了两种办法，一种是直接输入，一种是在终端使用curl注入，发现均不能成功，说明PHP的mysqli query函数不支持多语句。

以下是第二种办法。



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

首先还是观察/var/www/SQLInjection下unsafe edit backend.php的内部代码，发现也有可以利用的漏洞。



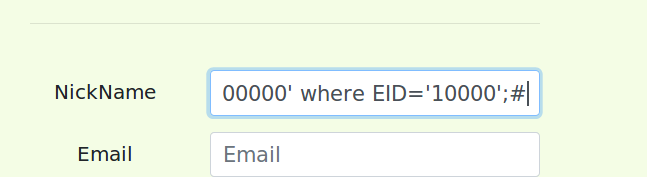
登录Alice。

用以下网址进入进入profile edit页面：

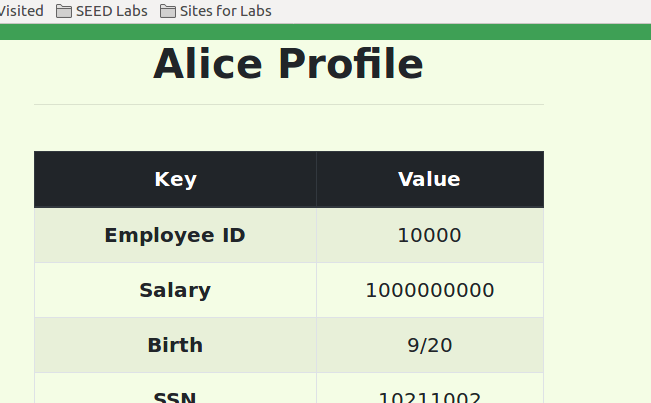
<http://www.seedlabsqlinjection.com/unsafe_edit_frontend.php>

在nickname一框中输入',salary='1000000000' where EID='10000';#：

然后点击提交

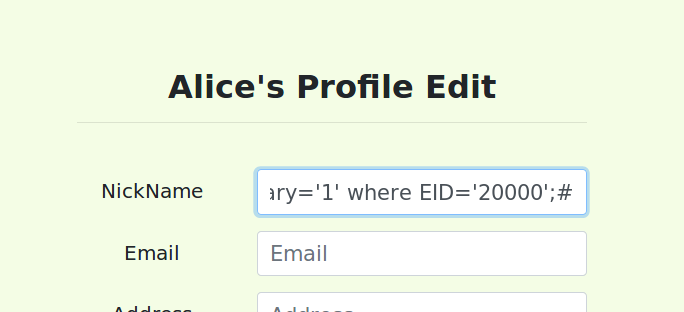


保存后跳回到Alice的Profile界面，发现工资已经被修改

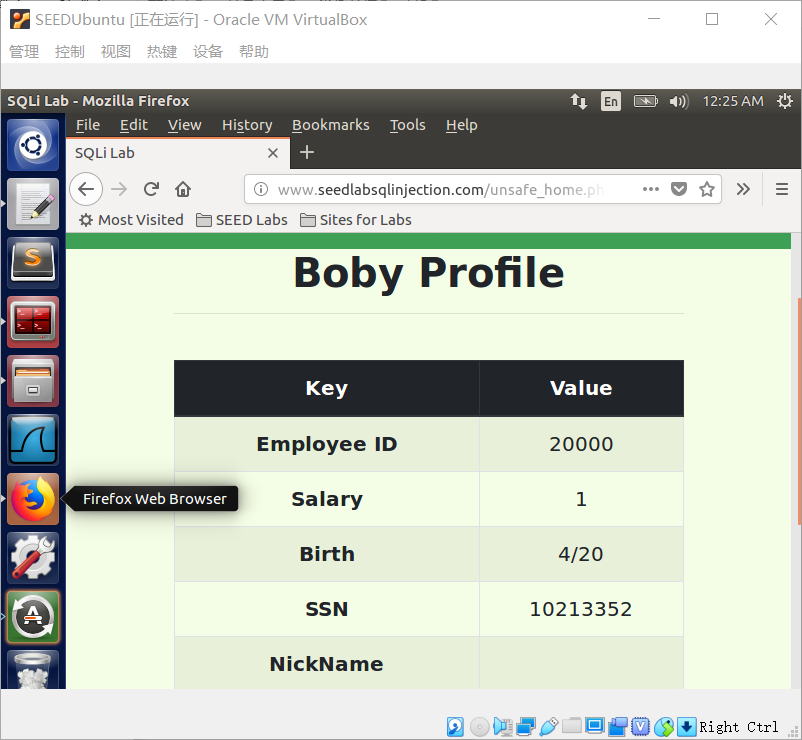


**Task 3.2: Modify other people’ salary**.

同理，我们在nickname里加入', salary='1' where EID='20000';#

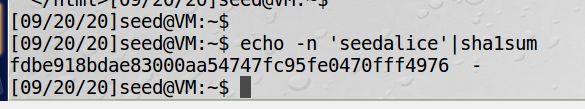


登录boby的账号查看一下，发现信息修改成功。

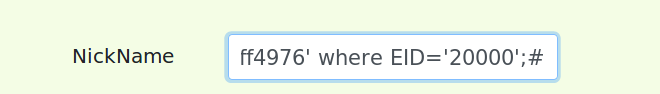


**Task 3.3: Modify other people’ password**.

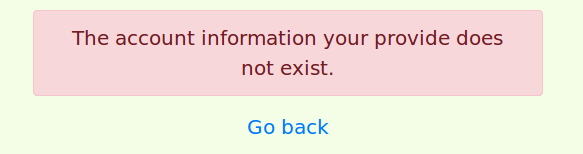
根据任务描述我们知道密码是以sha1 hash存储的，所以我们要改密码的时候，首先要知道我们修改密码的sha1 hash值，比如我们要把boby的密码改成Alice自己的密码，所以我们去终端输入以下命令：



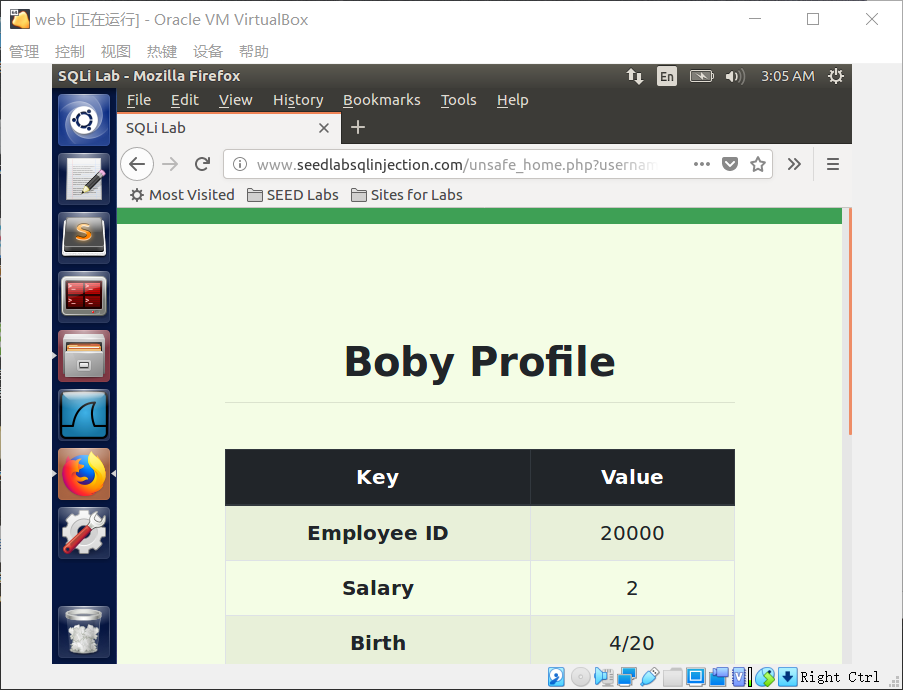
得到密码是fdbe918bdae83000aa54747fc95fe0470fff4976，接着我们去nickname里面修改boby的密码，输入以下命令：', Password=' fdbe918bdae83000aa54747fc95fe0470fff4976' where EID='20000';#



这下我们用boby输入自己原来的密码登录，出现



而用seedalice登录时，会发现可以正常登录，可知攻击成功。



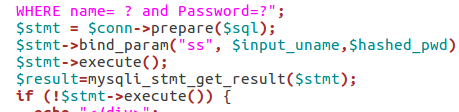
**Task 4: Countermeasure — Prepared Statement**

在/var/www/SQLInjection目录下找到unsafe\_home.php文件，打开修改

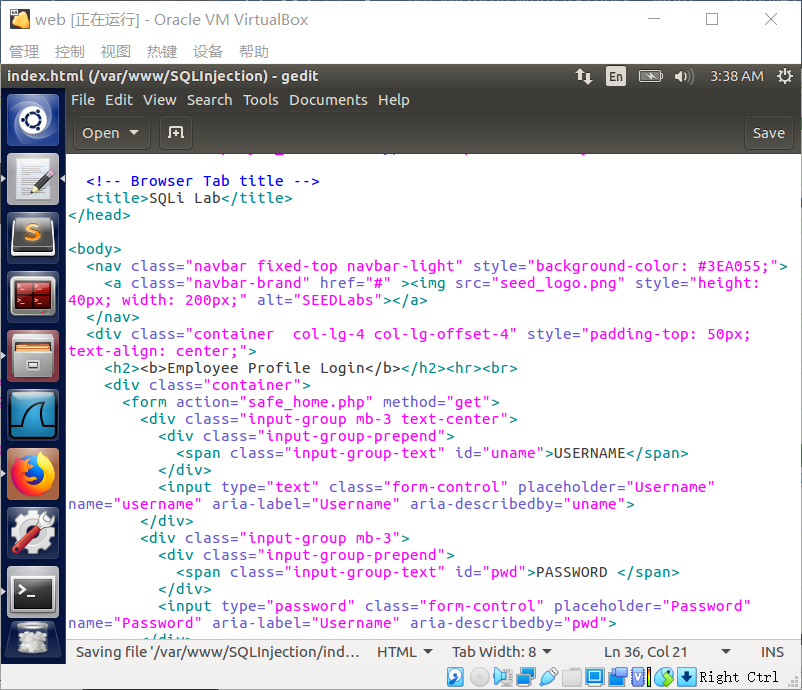
修改之前：



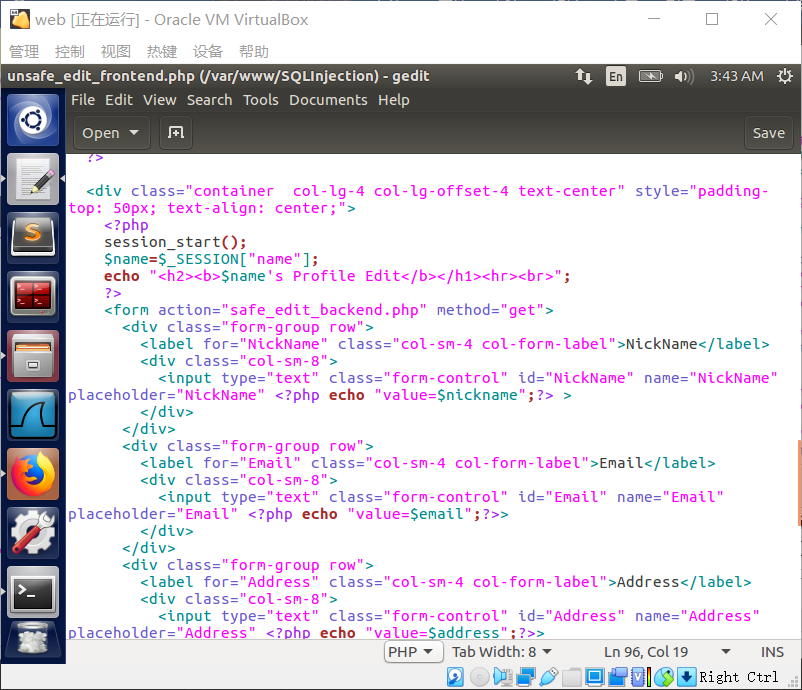
修改之后：



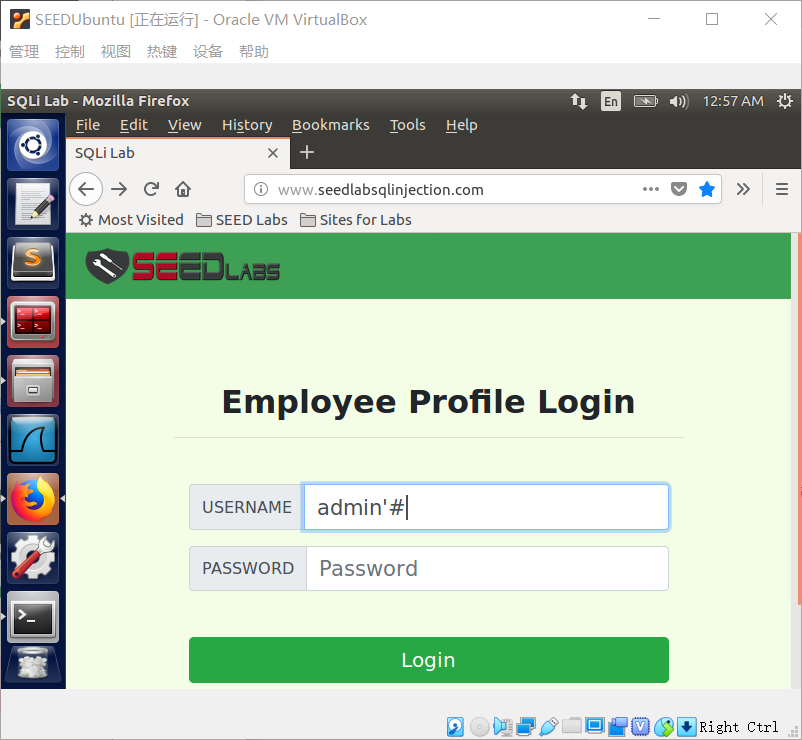
再修改index.html里的代码，将所示处的unsafe改成safe：



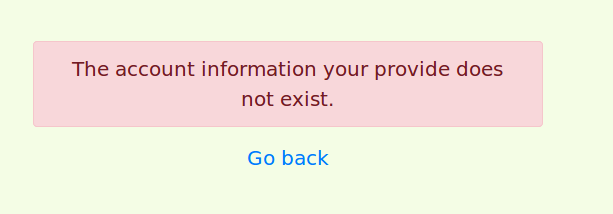
修改unsafe\_edit\_frontend.php，同样将所示处的unsafe改成safe：



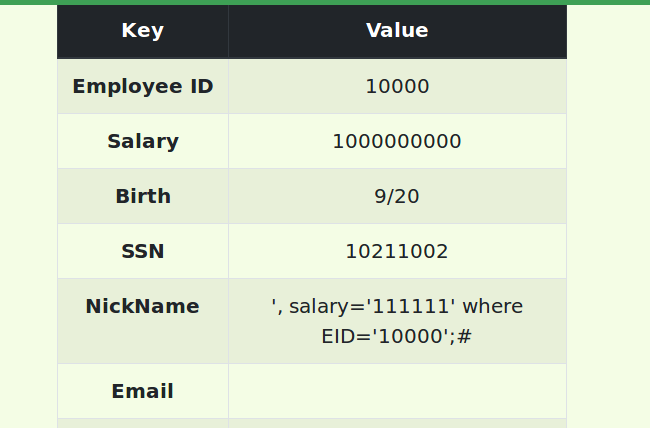
然后我们随便选择上述任意一个攻击，这里我们选择task2的攻击



出现下面情况



我们再来修改薪水：



却发现只改了nickname

尝试修改密码，发现也未成功。

