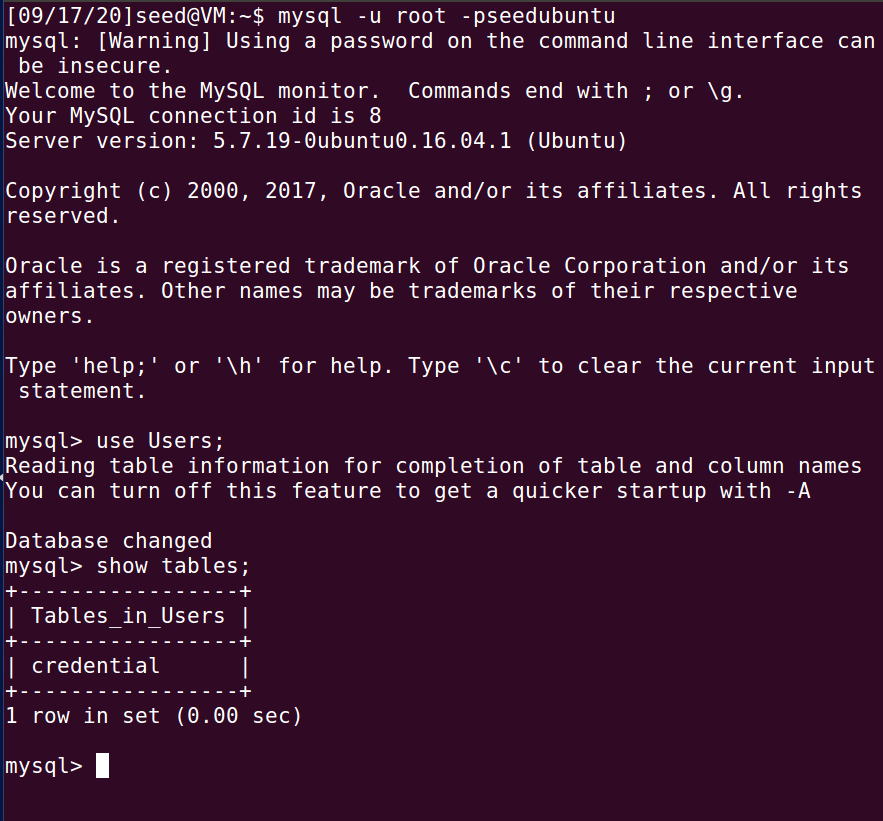
**Lab6-Report**

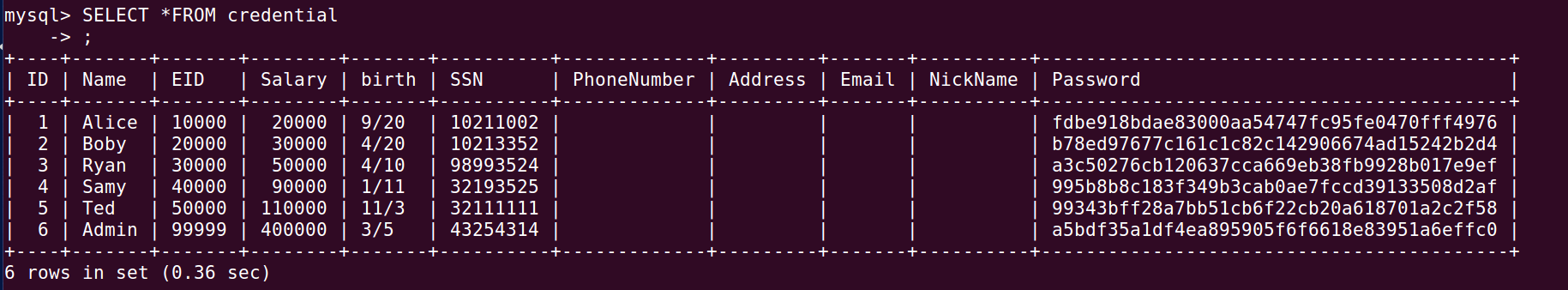
57118116 陈煜

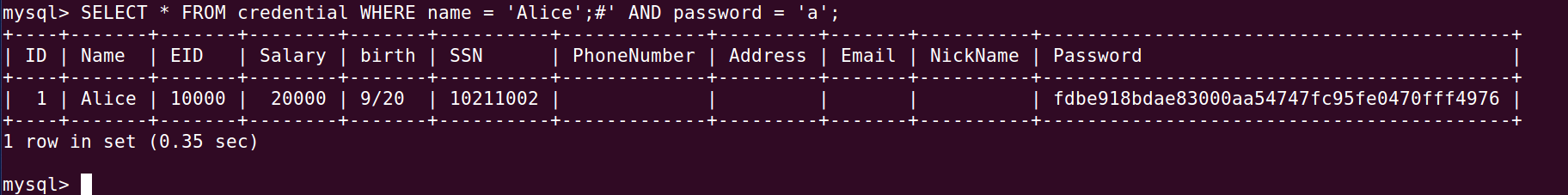
Task 1: Get Familiar with SQL Statements

在终端输入如下命令进入MySQL，载入Users数据库并查看内容：



使用如下命令打印Alice的所有信息：

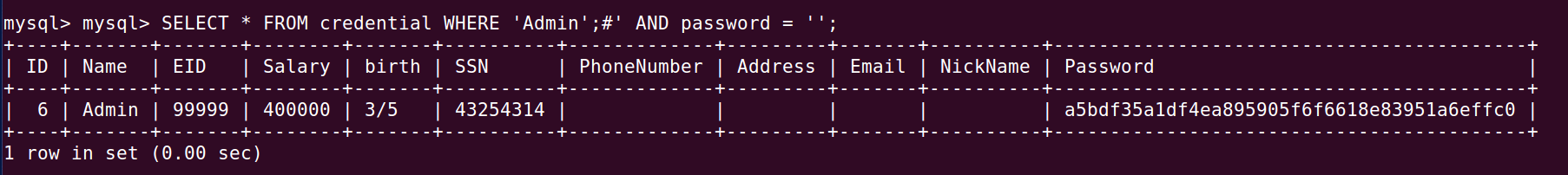




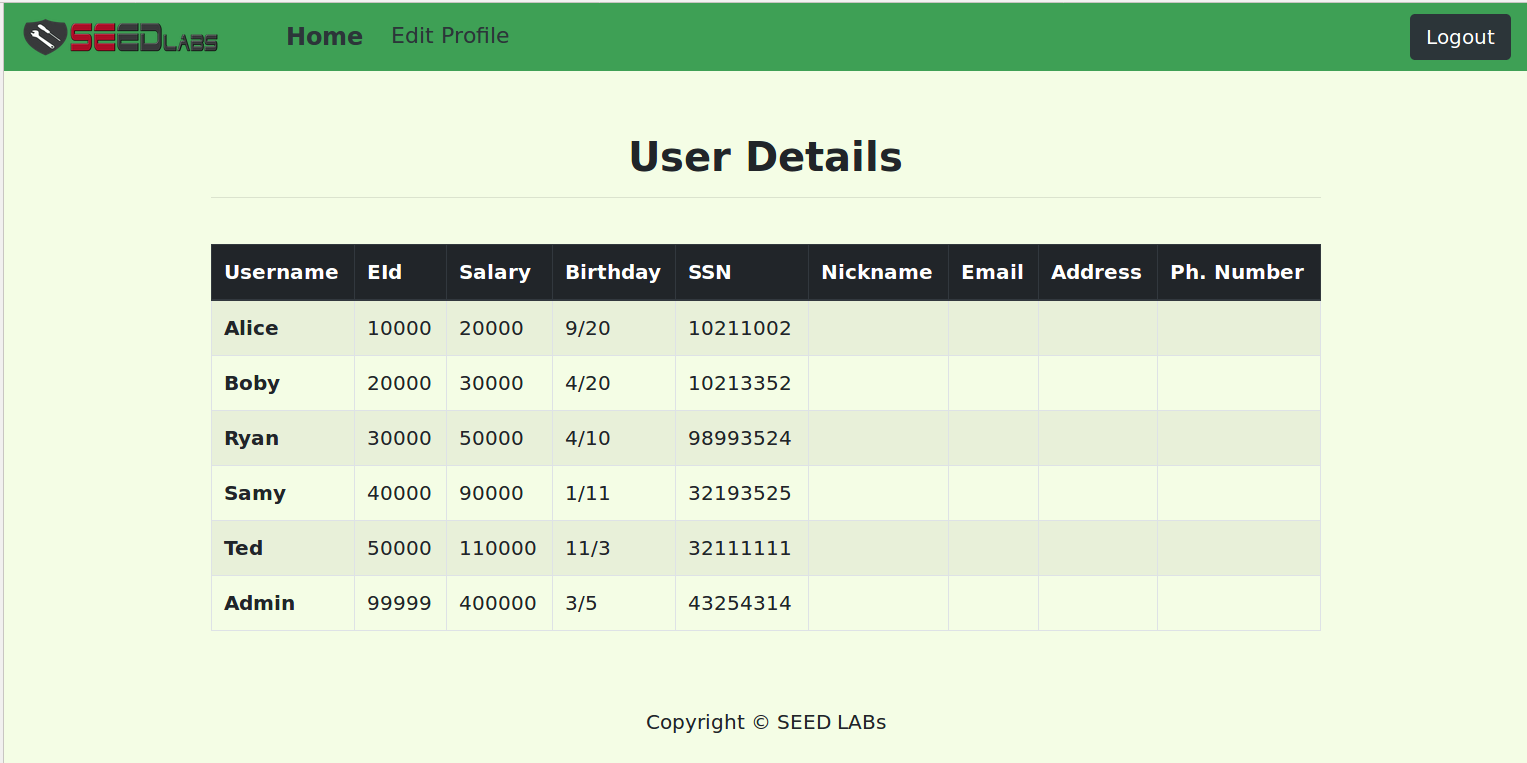
Task 2: SQL Injection Attack on SELECT Statement

• Task 2.1: SQL Injection Attack from webpage.

在终端的SQL中输入如下命令：

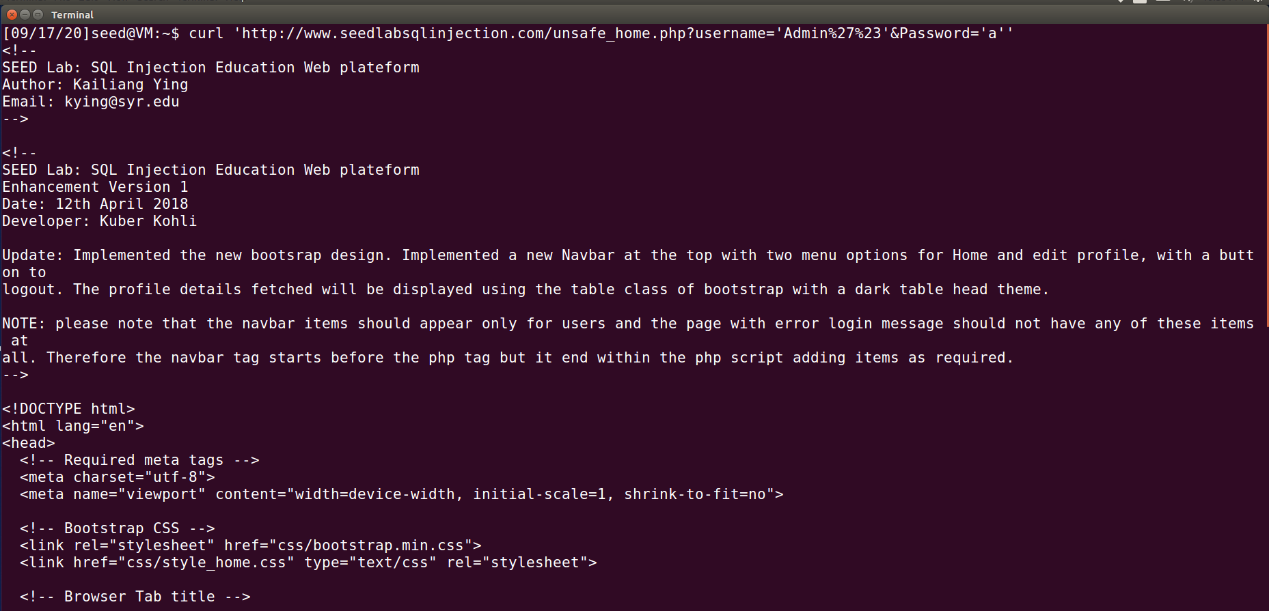


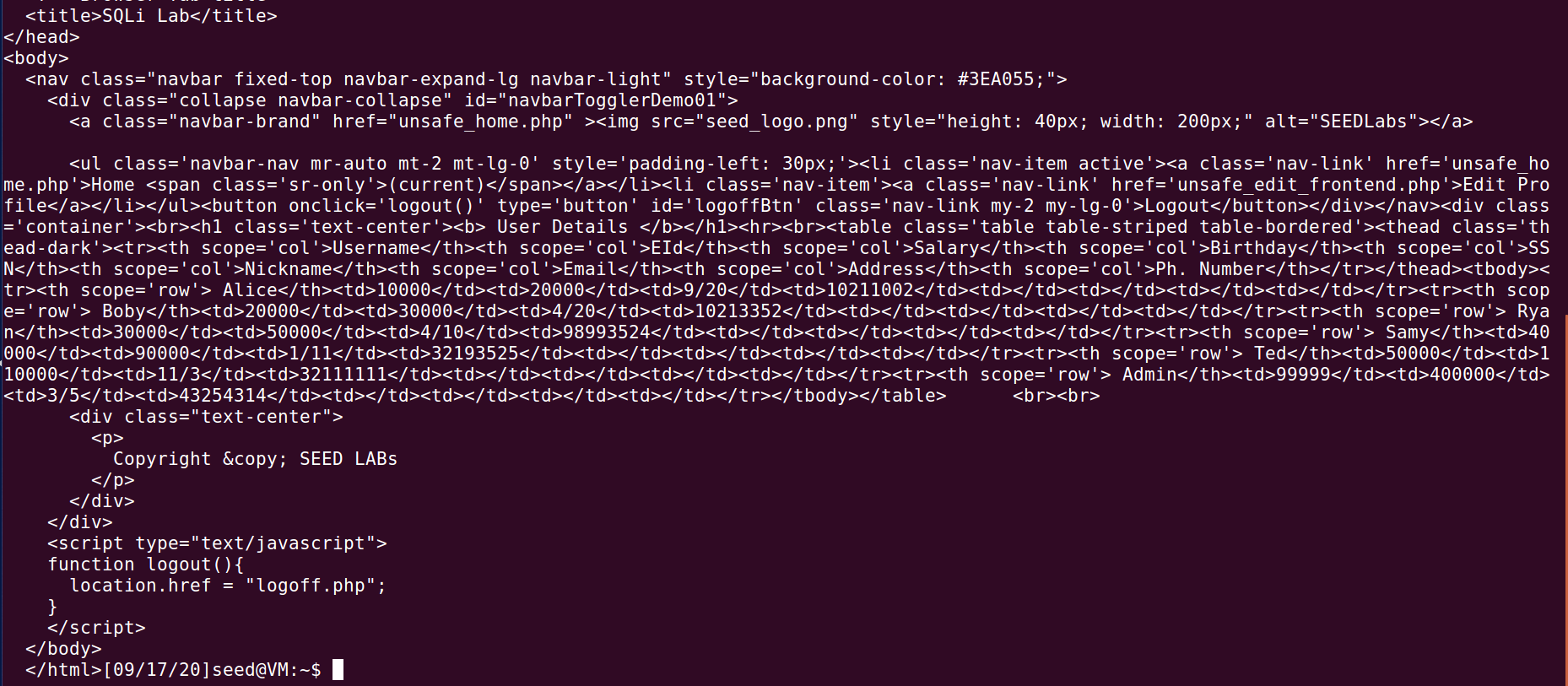
再在[www.SEEDLabSQLInjection.com](http://www.SEEDLabSQLInjection.com)中输入用户名admin’;#，任意输入密码，完成攻击：



• Task 2.2: SQL Injection Attack from command line.

在终端输入如下命令，完成攻击：

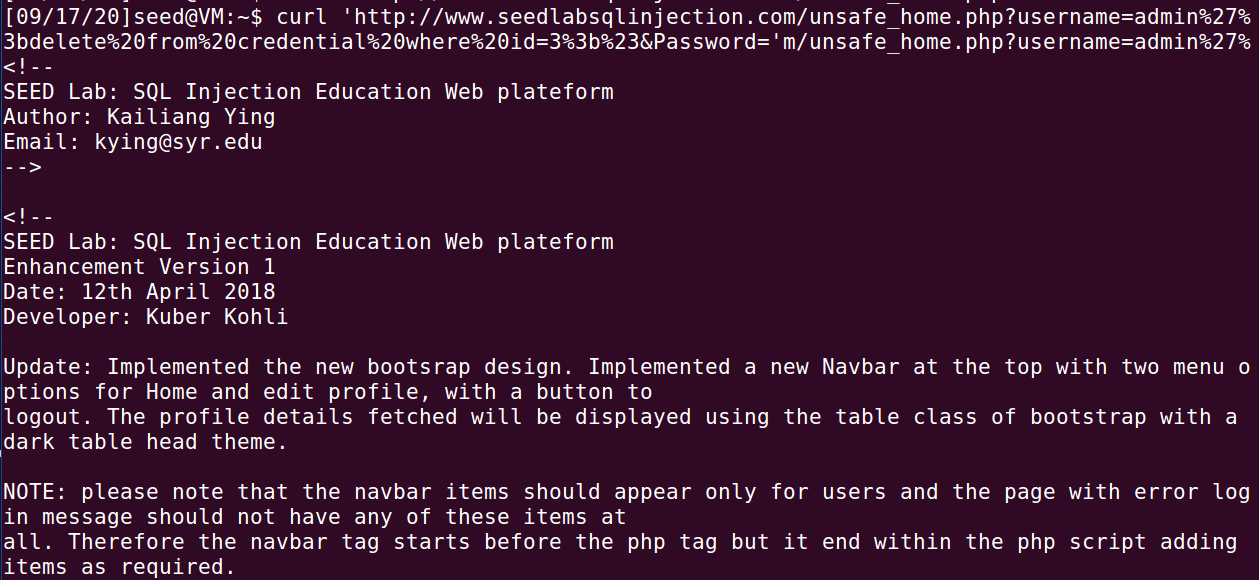




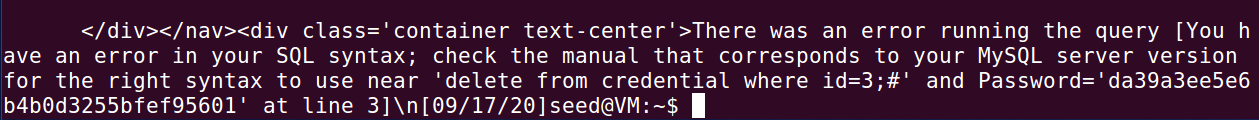
• Task 2.3: Append a new SQL statement.

在终端中写入url，在url中插入多条sql语句：

curl 'http://www.seedlabsqlinjection.com/unsafe\_home.php?username=admin%27%3bdelete%20from%20credential%20where%20id=3%3b%23&Password='



在终端输出信息的最后，发现出现了错误：

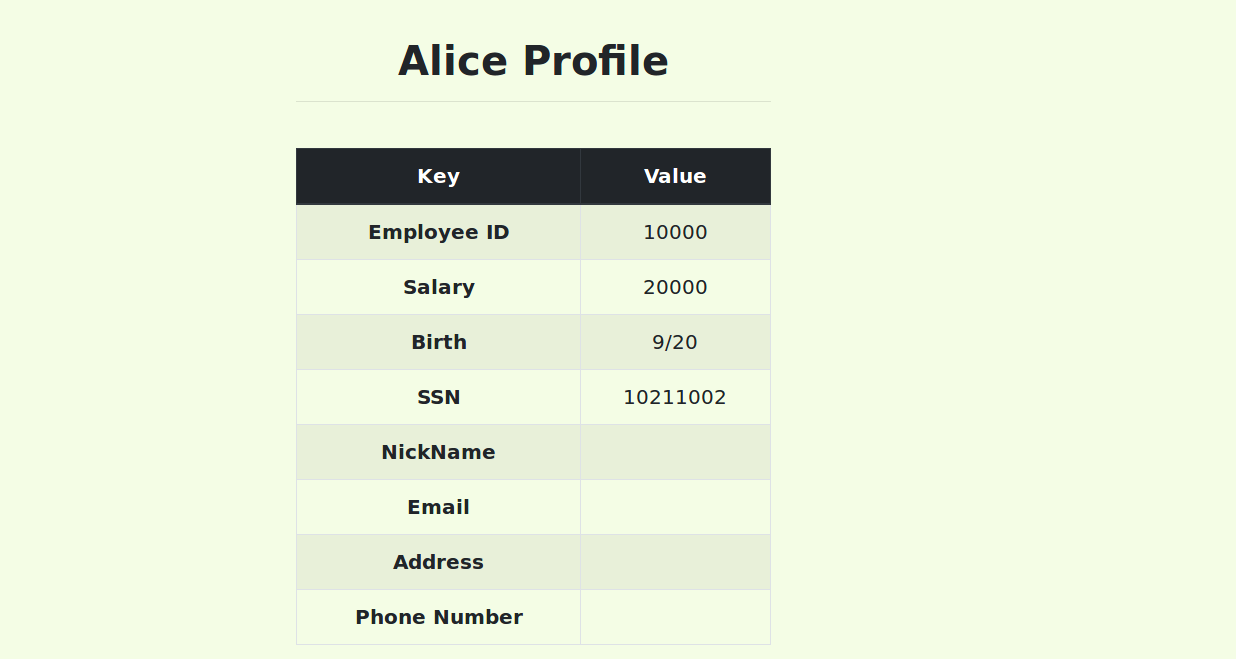


因此，插入多条sql语句的攻击不成功。PHP的MySQL query函数不支持多个语句。

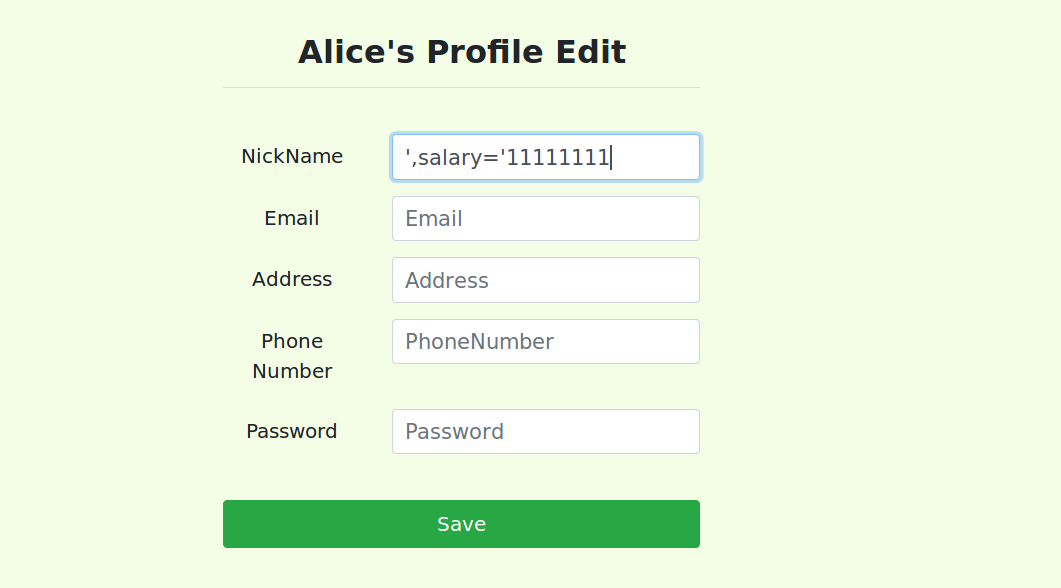
Task 3: SQL Injection Attack on UPDATE Statement

• Task 3.1: Modify your own salary.

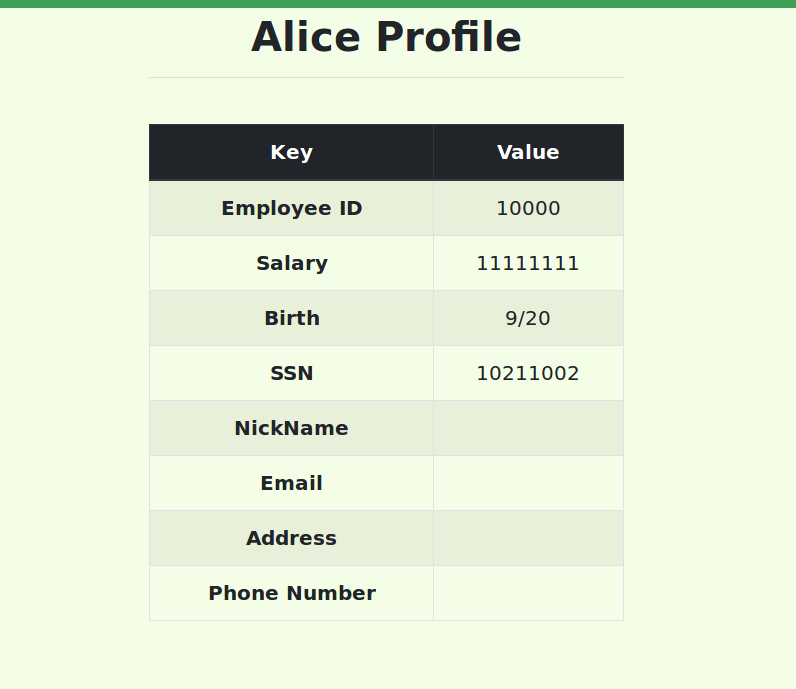
先查看修改前Alice的salary：



在edit profile的nickname中输入以下内容：

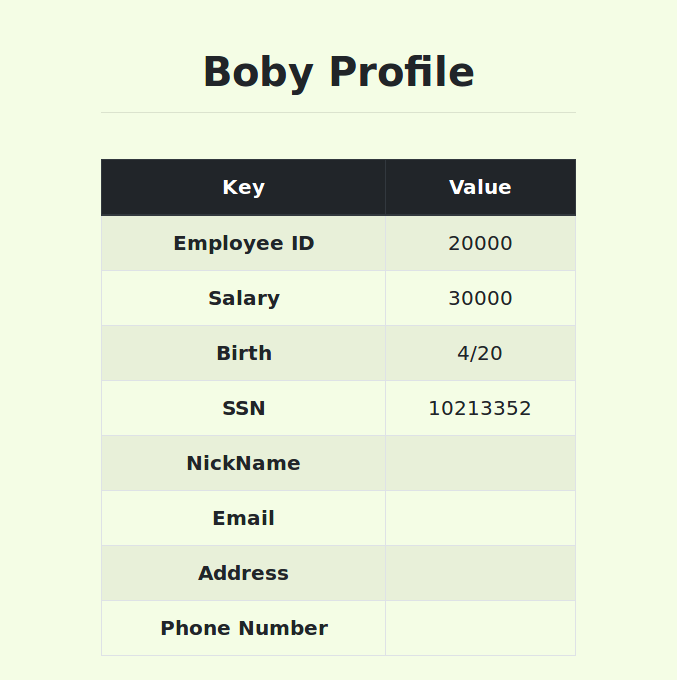


保存后返回主页，发现salary被修改：

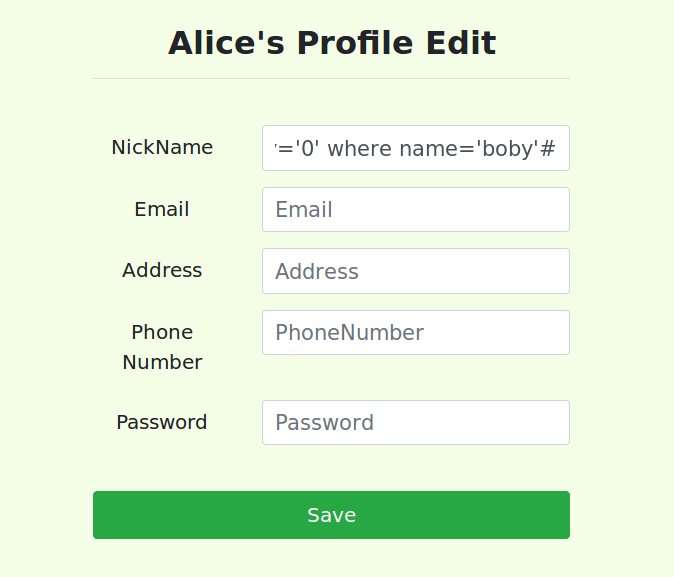


Task 3.2: Modify other people’ salary.

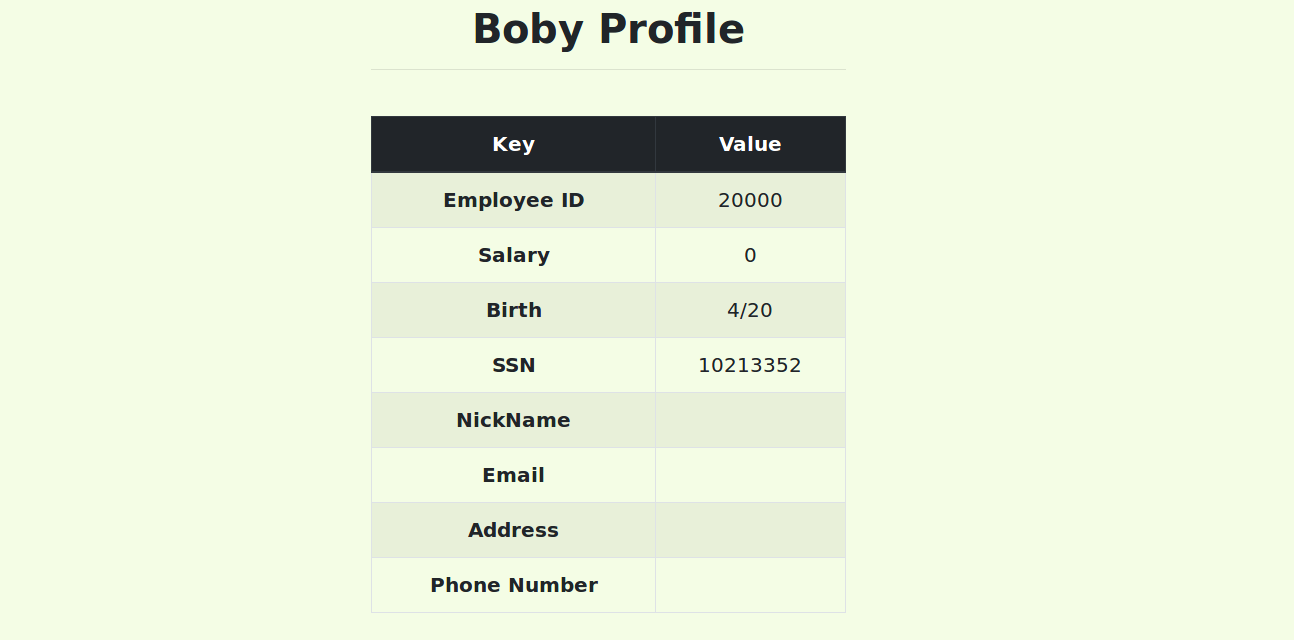
先登录Boby的账号，查看Boby的salary：



进入Alice账号的edit profile界面，在NickName中输入',salary='0' where name='boby'#：



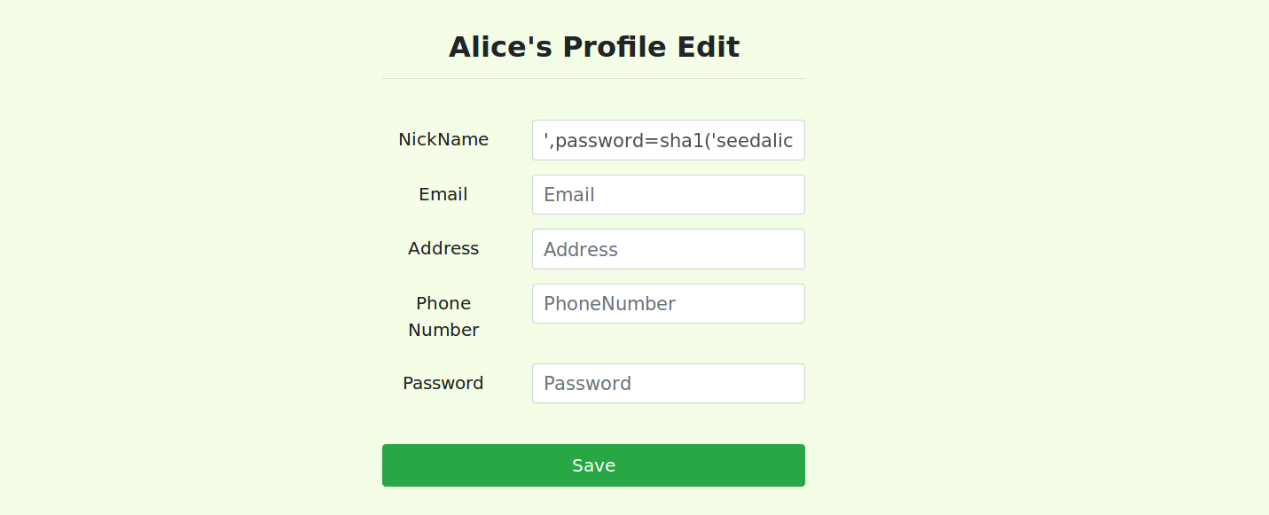
进入Boby的账号，发现salary被改变：



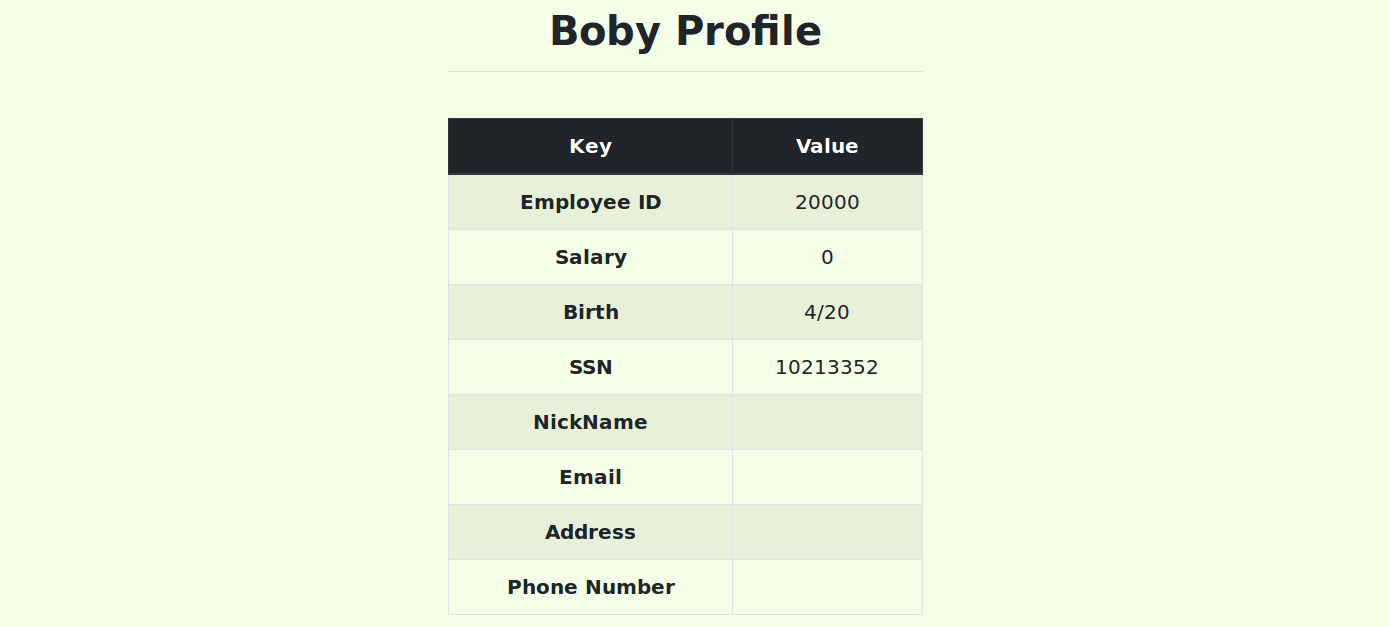
攻击成功。

Task 3.3: Modify other people’ password.

在Alice的edit profile中将NickName修改为',password=sha1('seedalice') where name = 'boby'#：



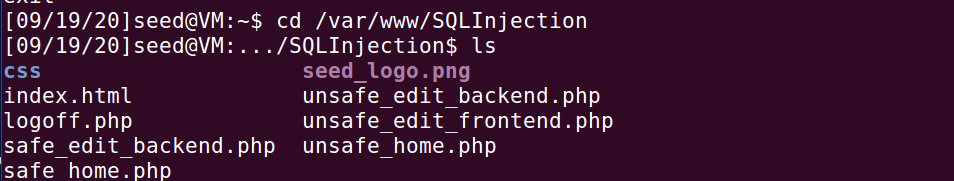
保存后，用修改后的密码seedalice能进入Boby的账户：



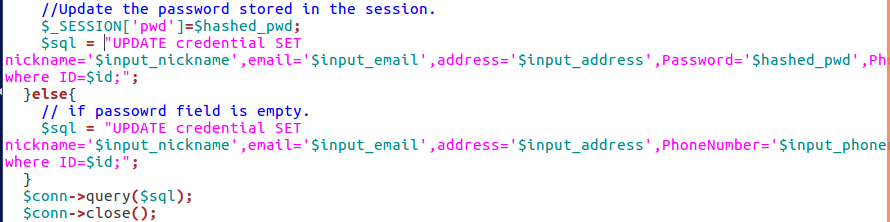
攻击成功。

Task 4: Countermeasure — Prepared Statement

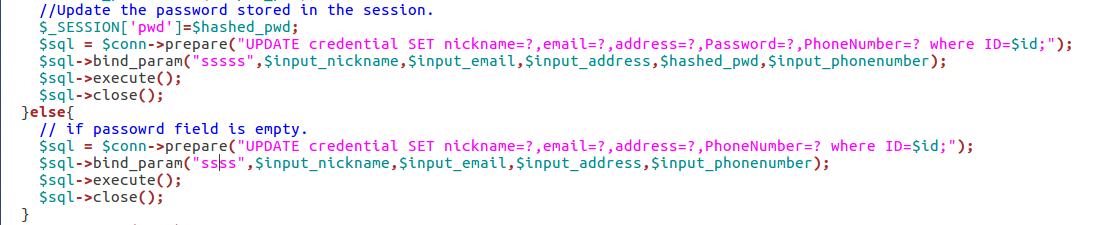
进入/var/www/SQLInjection目录，通过ls命令查看相应的php文件：



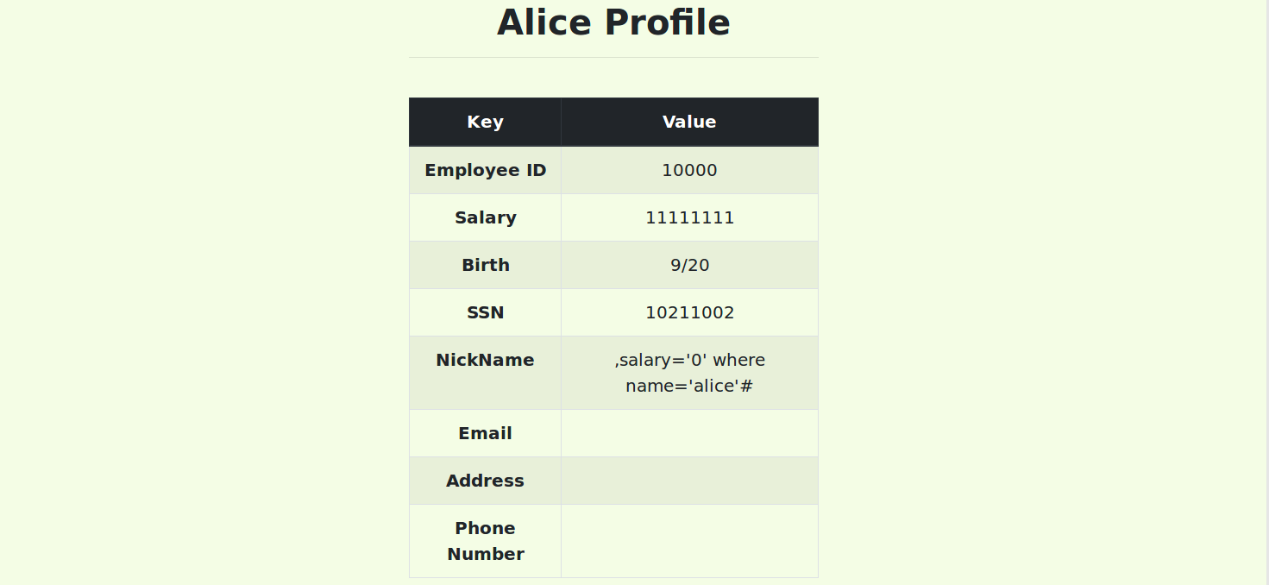
进入unsafe\_edit\_backend.php，修改前代码如下：



代码和数据区分不明显，导致攻击能成功。将其代码修改如下：



进入浏览器，执行task3.2中的攻击：进入Alice的edit profile，修改nickname为,salary='0' where name='alice'#，保存后回到主页，发现没有修改Alice的salary：



攻击失败。

由此可见，合理区分代码和数据，能有效防止攻击。