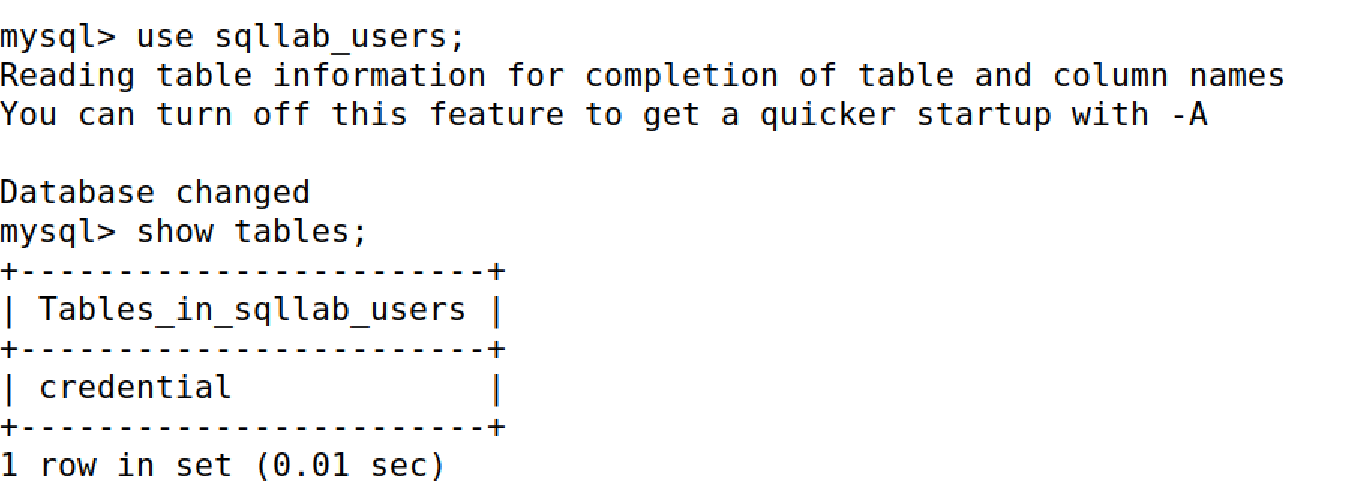
**SQL Injection Attack Lab**

作者：黄禧敏

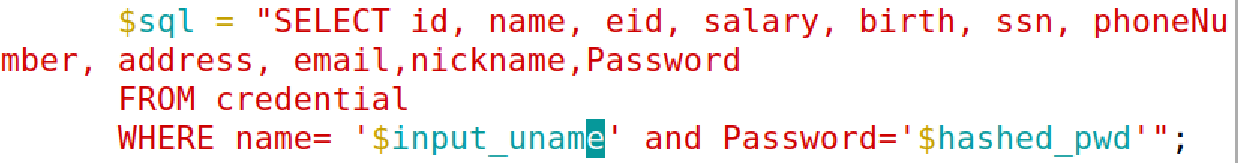
学号：57119109

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

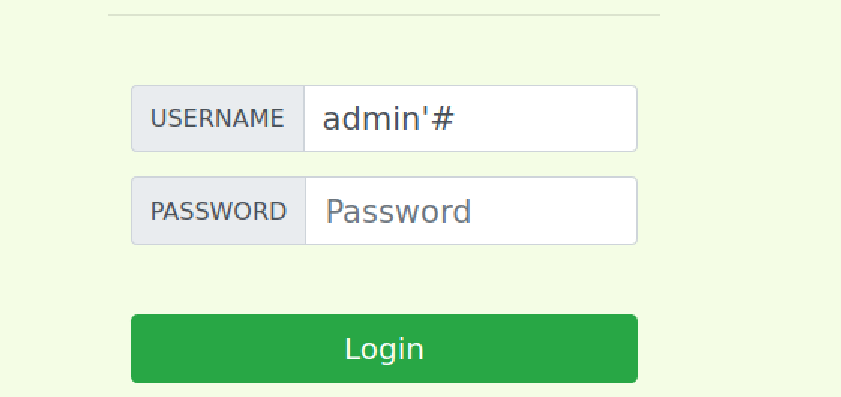


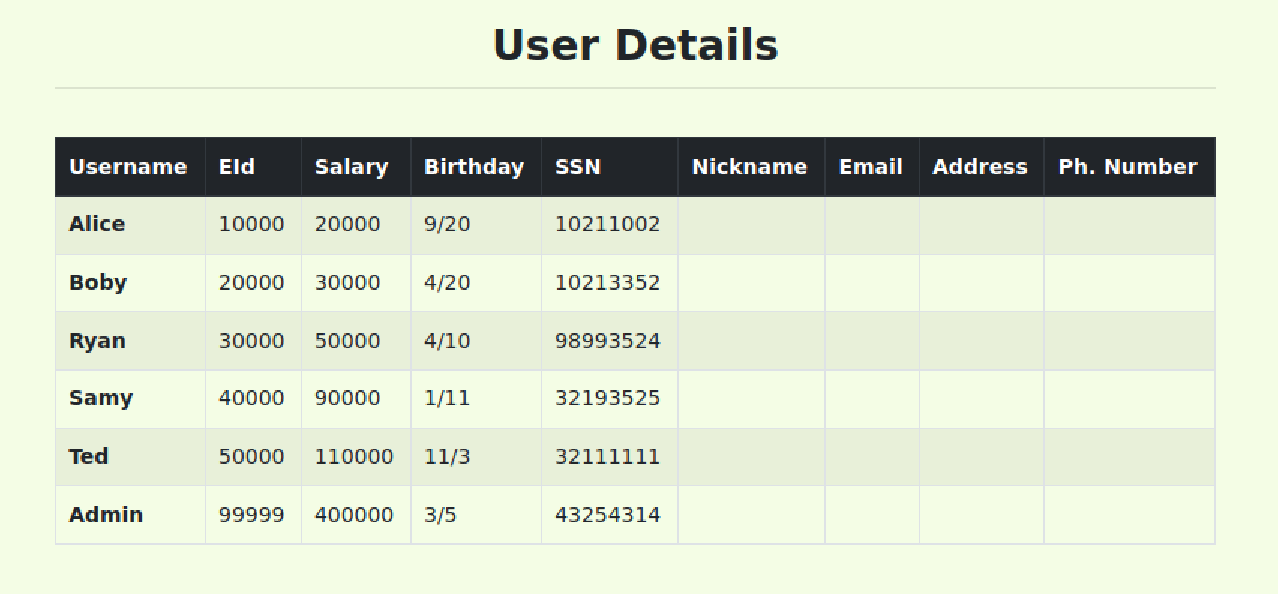
**Task 2.1 :SQL Injection Attack from webpage**

观察unsafe\_home.php:

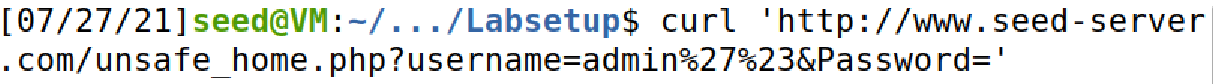


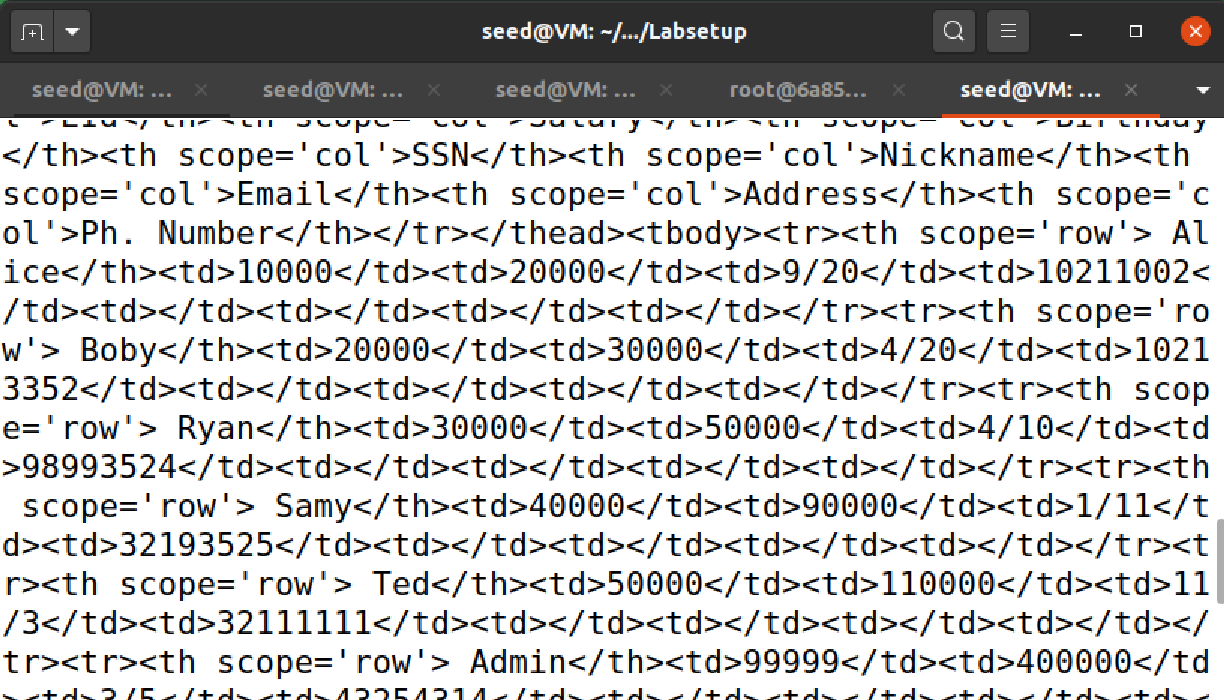
若在uname之后输入’#，则在select语句中#之后的语句将被注释





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



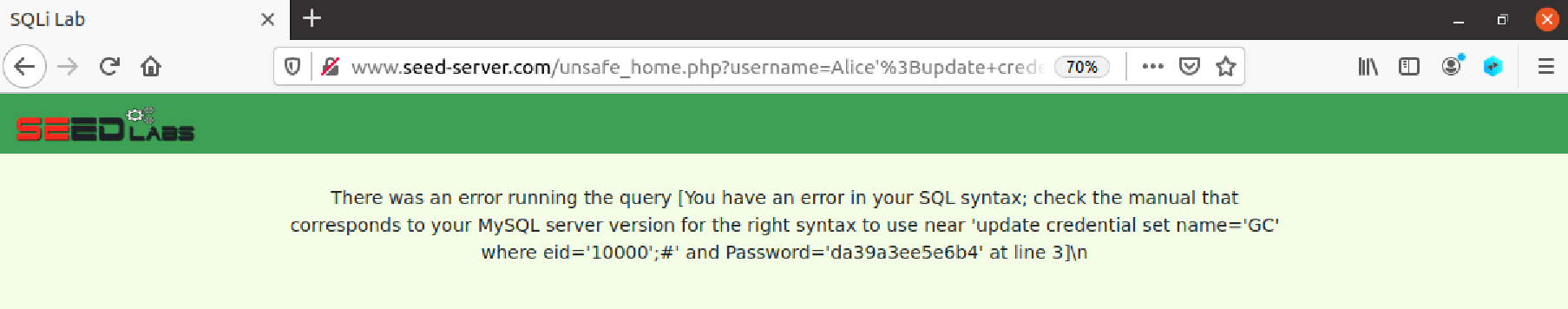


**Task 2.3 : Append a new SQL statement.**

在登陆框中输入

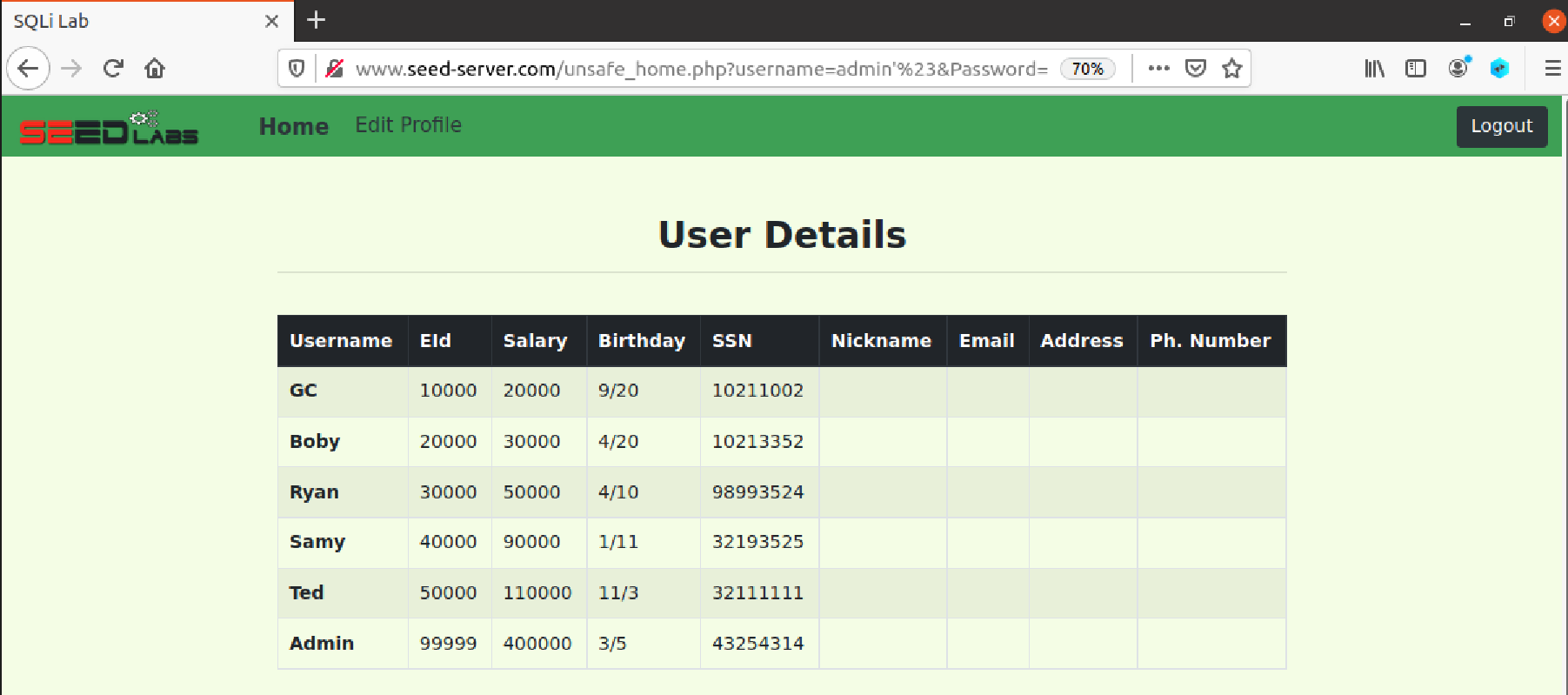
Alice'; update credential set name=’GC’ where eid=’10000’;#

可以看到攻击不成功；

由于$mysqli->query() API的限制，同时只能输入一条sql语句；

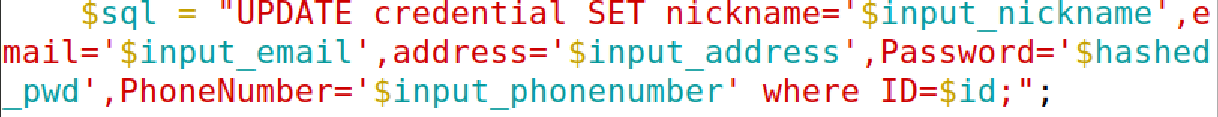
将unsafe\_home.php中的query()语句全部改为multi\_query();

输入攻击语句，页面跳转为空白，再次查询信息列表，可以发现Alice用户名已改为GC



**Task 3.1: Modify your own salary**

观察unsafe\_edit\_backend.php;

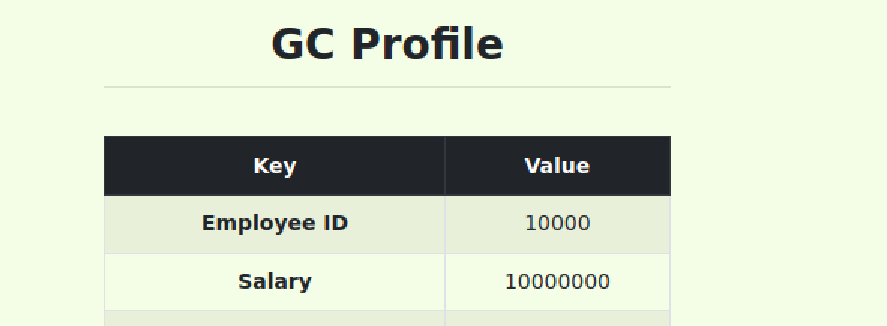


在profile edit中修改Nickname



‘, salary='10000000' where eid=’10000’;#

攻击成功，薪水被成功修改为设定值；



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

与task3.1相似，仅将eid改为其他人的eid即可完成攻击;

如下，将Boby薪水改为1；



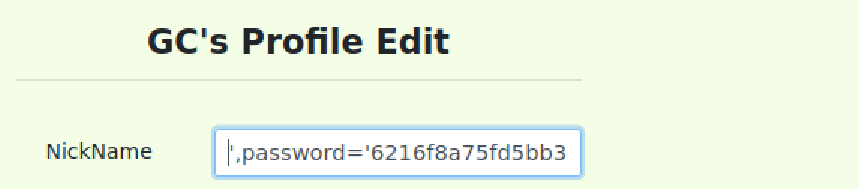
**Task 3.3: Modify other people’password**



可以看到密码加密方式为sha1, 将密码111在线加密

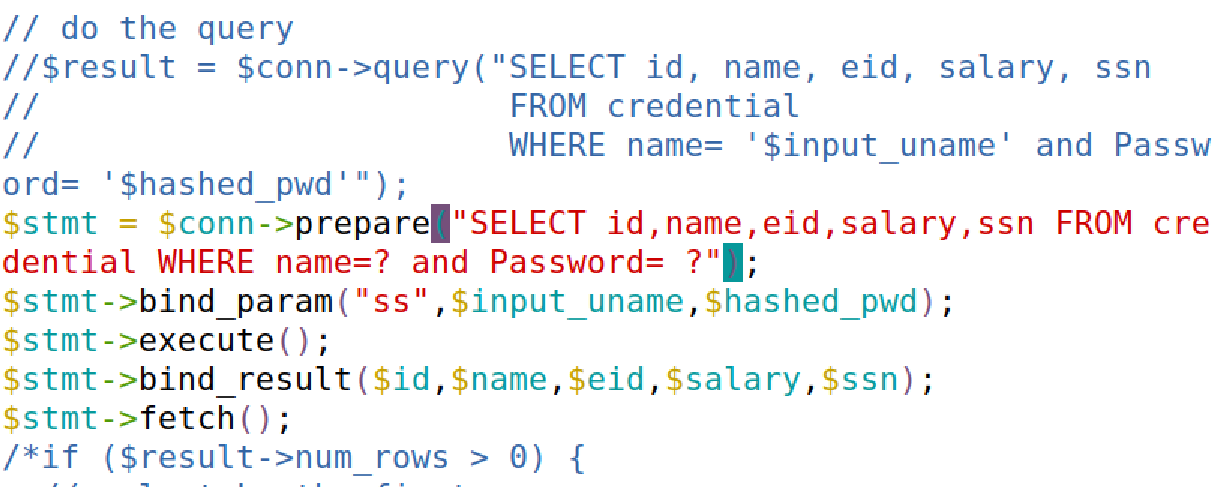


在GC账号将密文注入，使用Boby用户名输入111即可登录；

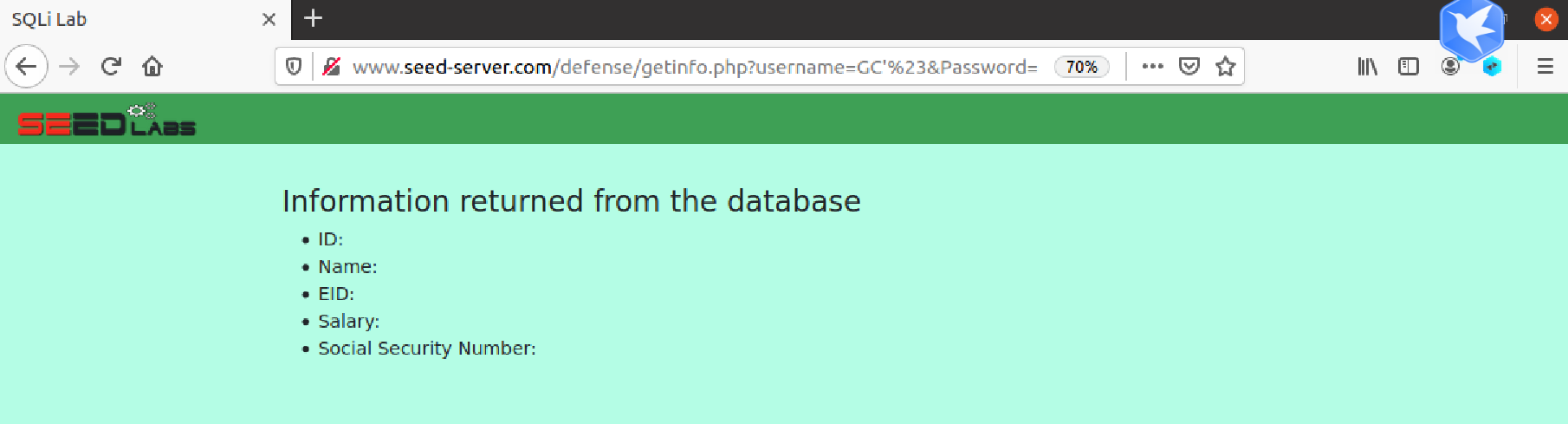


**Task 4: Countermeasure — Prepared Statement**

修改/defense文件夹下的unsafe.php



再次注入攻击



攻击失败，无法看到个人信息；