**实验六报告**

**实验名称：SQL injection attack Lab**

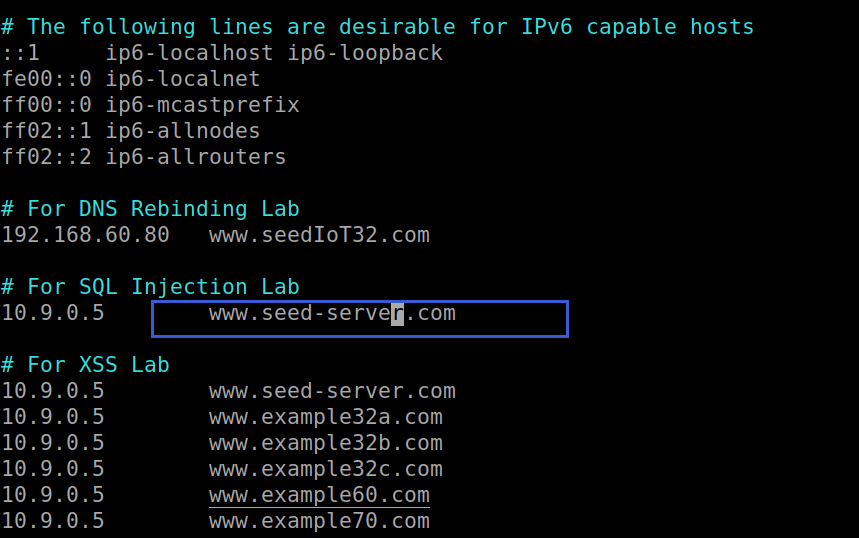
**基本信息：**

姓名：马运聪 学号：57119115 完成日期：2021.8.5

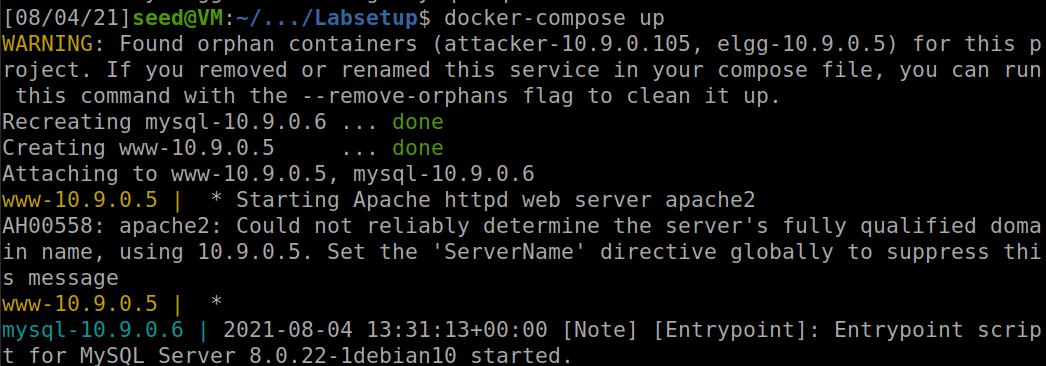
**实验内容：**

**环境配置**

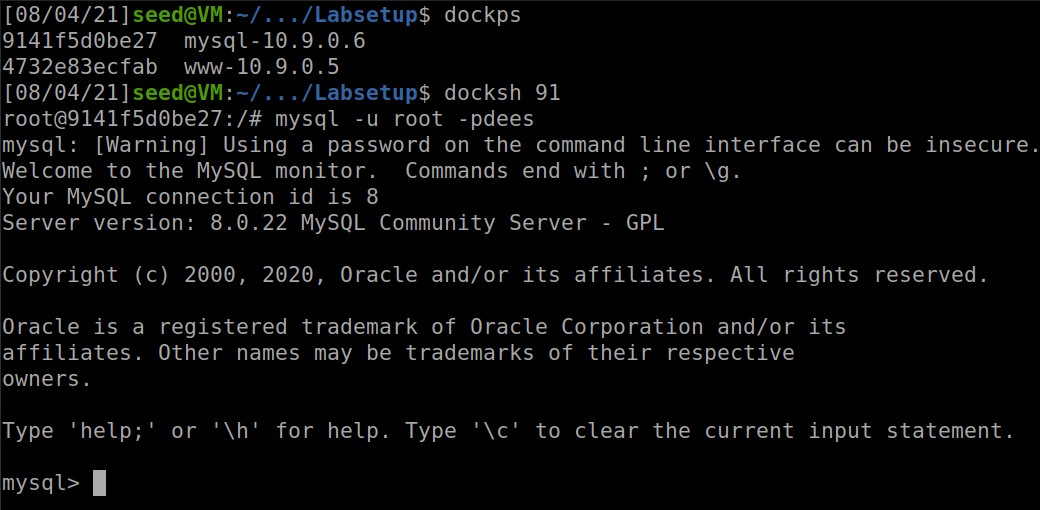
1.更改/etc/hosts文件中的地址映射，



2.启动Docker

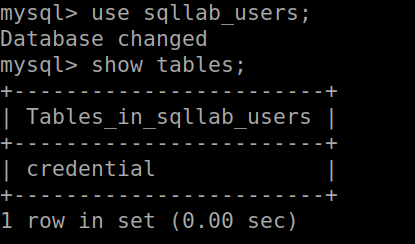


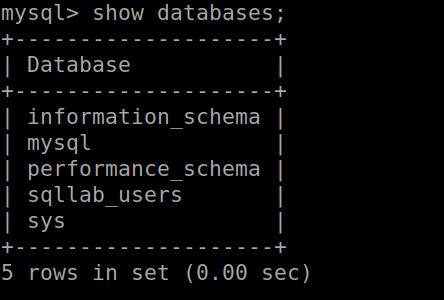
3.进入到容器中



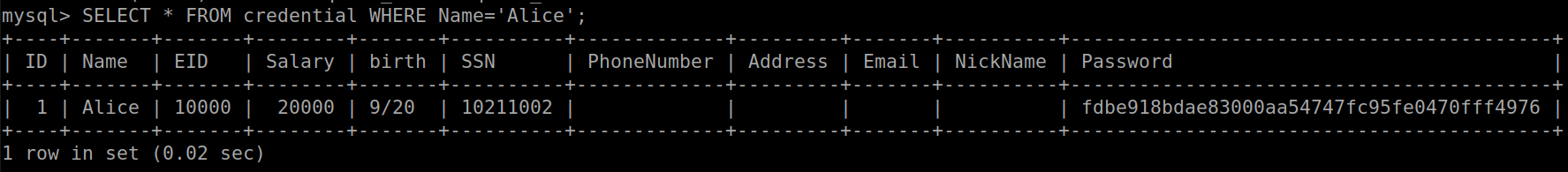
**Task 1: Get Familiar with SQL Statements**

1.熟悉mysql操作





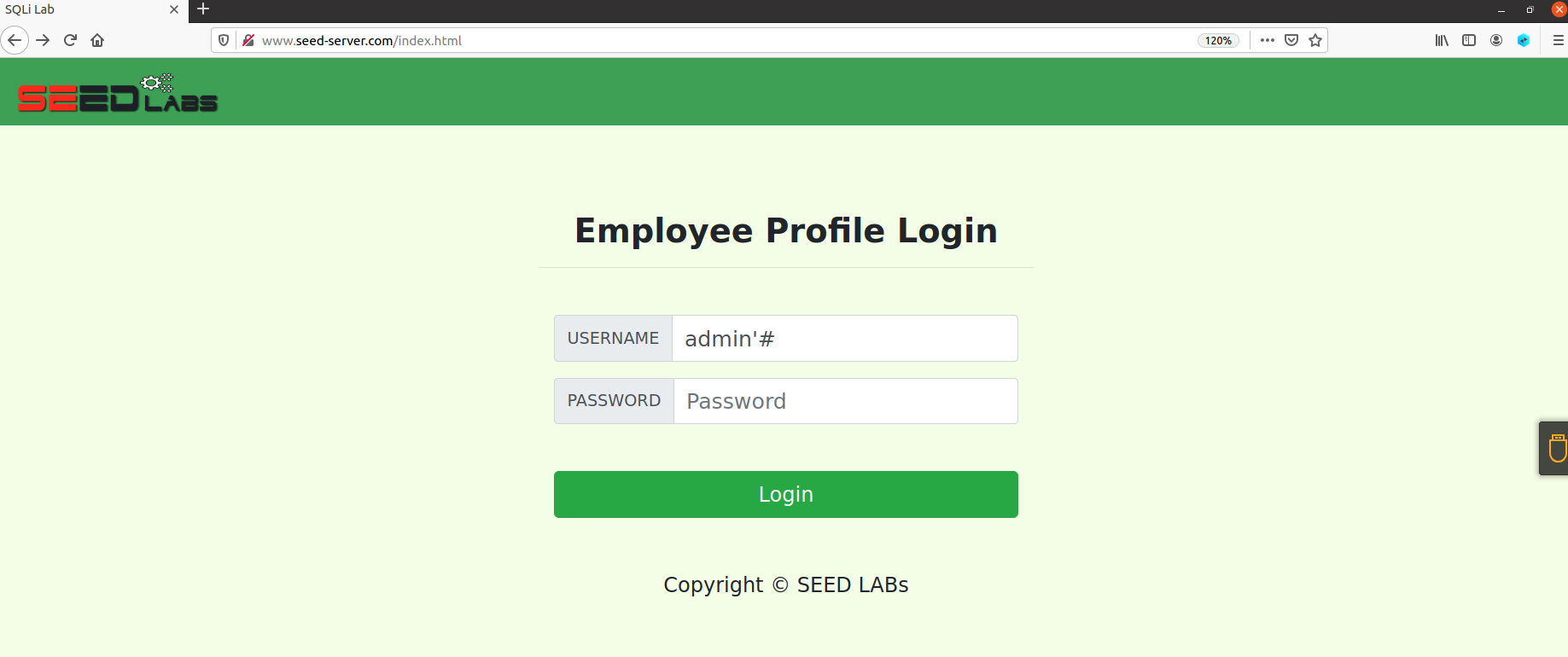
2.打印表credential中Alice的个人信息：

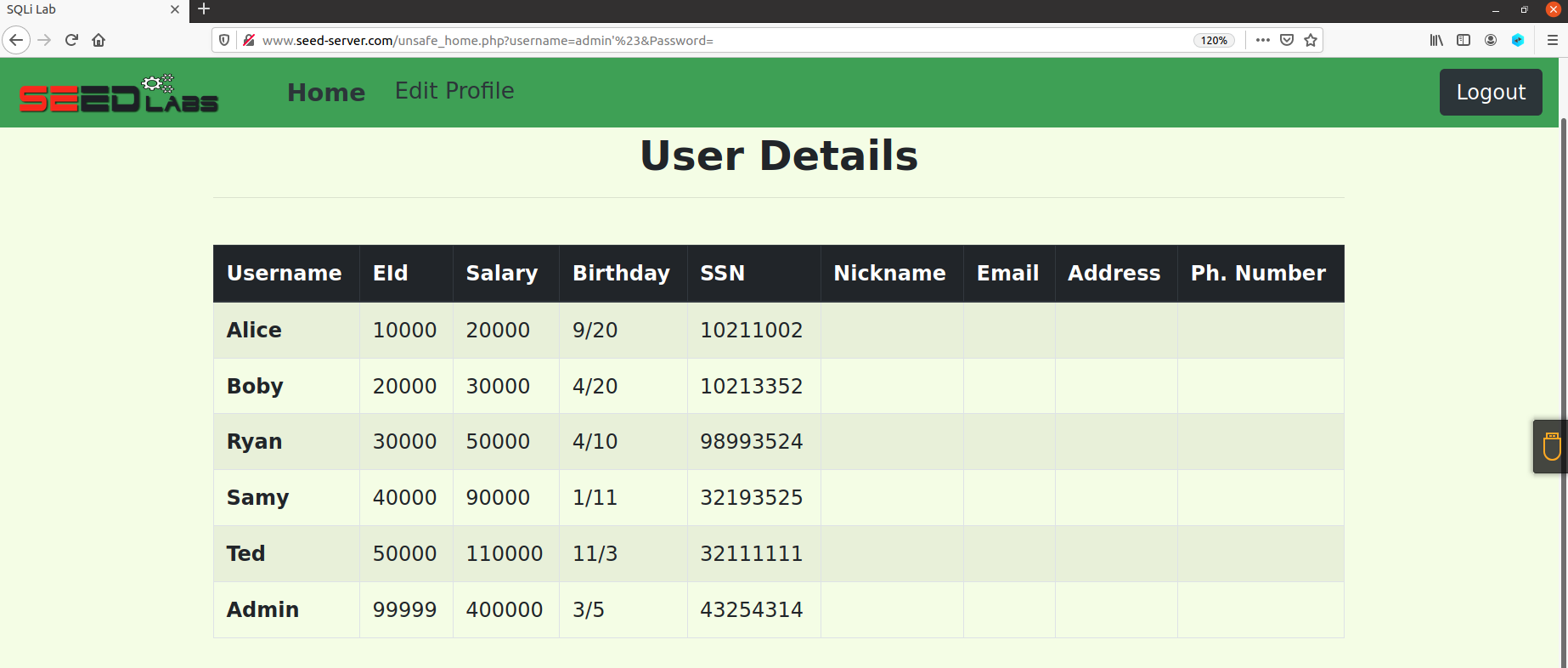


**Task 2: Posting a Malicious Message to Display Cookies**

1. Task 2.1: SQL Injection Attack from webpage

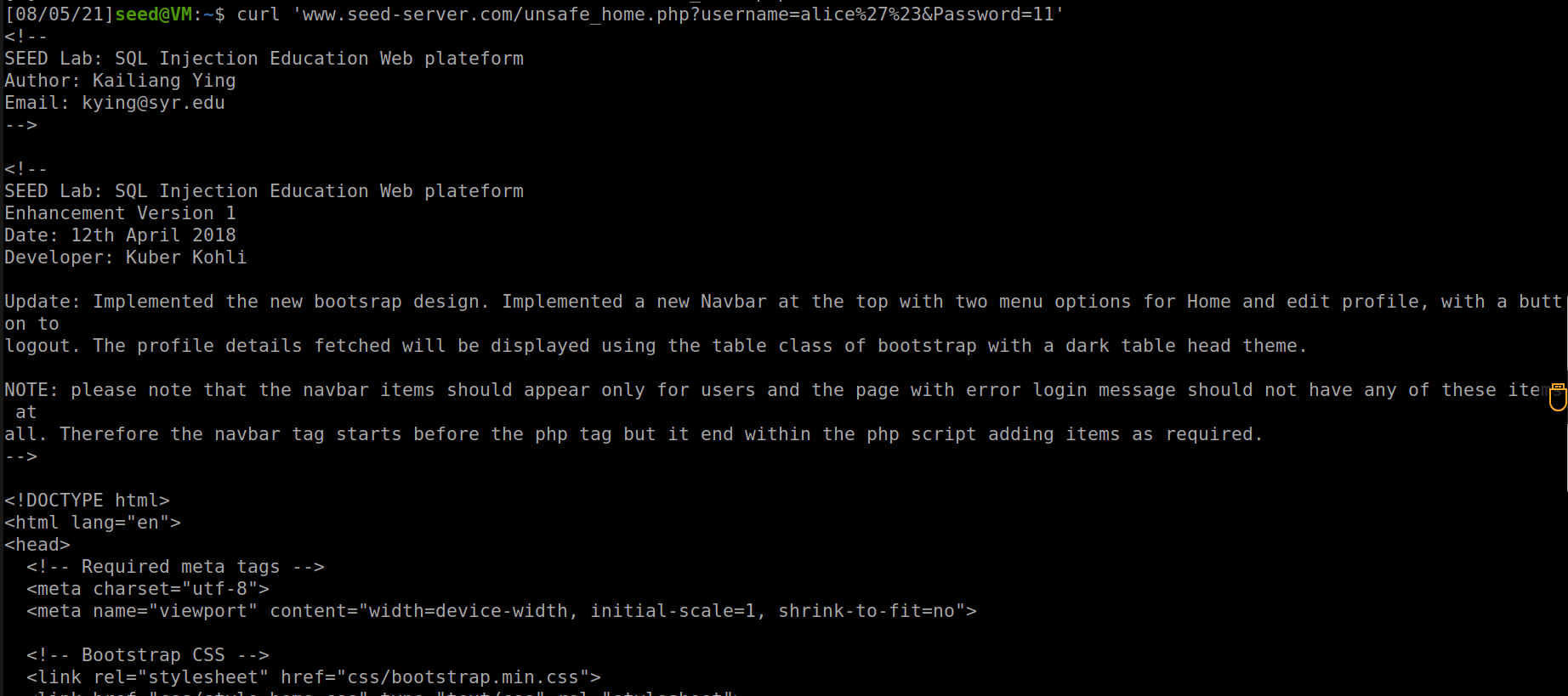
通过SQL的漏洞可知当用户名输入含撇号（‘）及井号（#）时，而密码无需输入，可直接改变SQL语句。这样当输入为“admin‘#”时，即可返回admin管理员的页面。

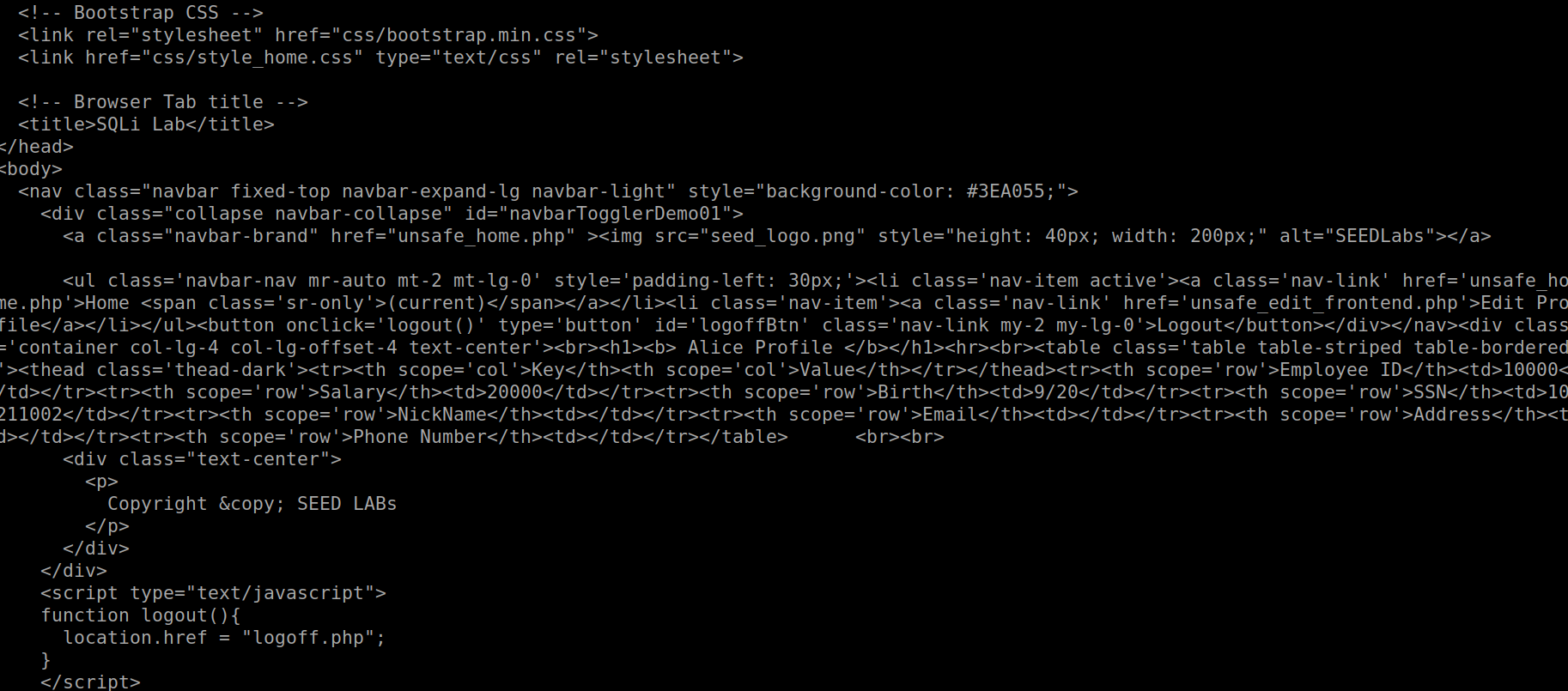




1. Task2.2: SQL Injection Attack from command line

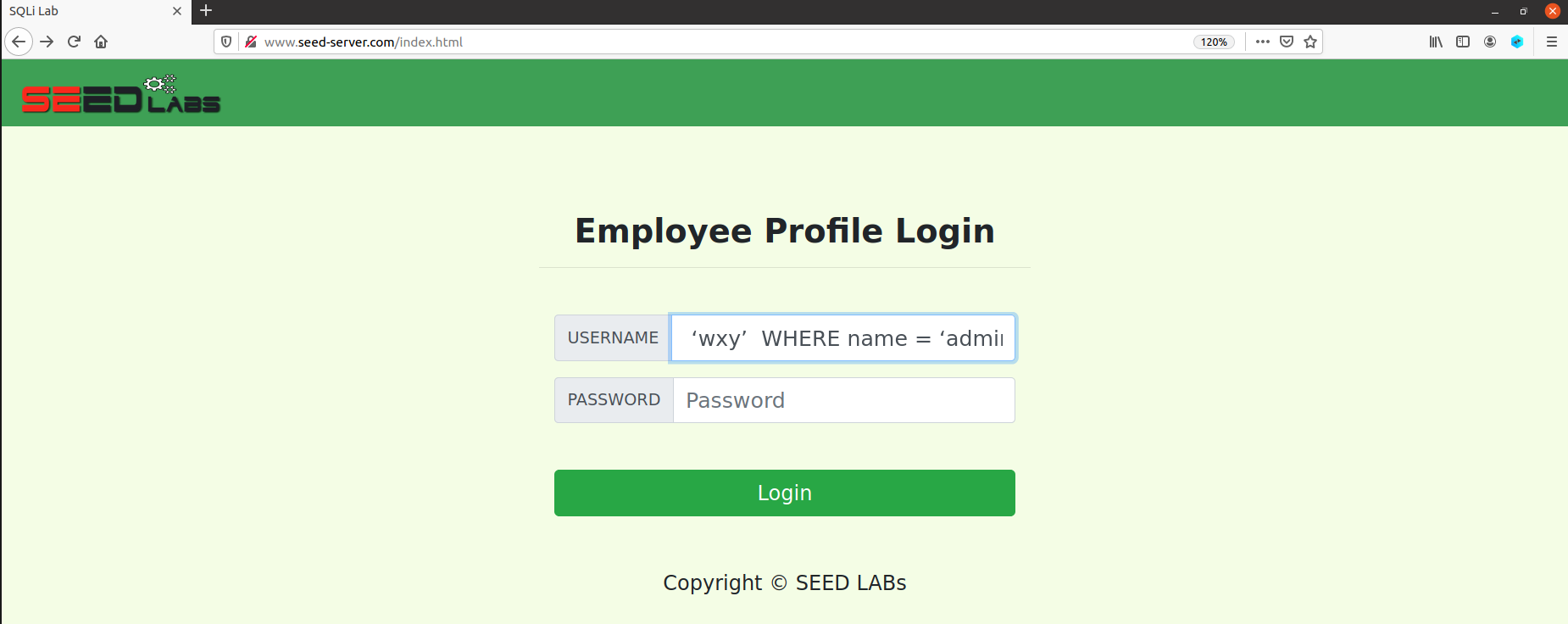
在命令行中敲入URL命令，在alice添加“%27%23”,可见直接返回查询结果

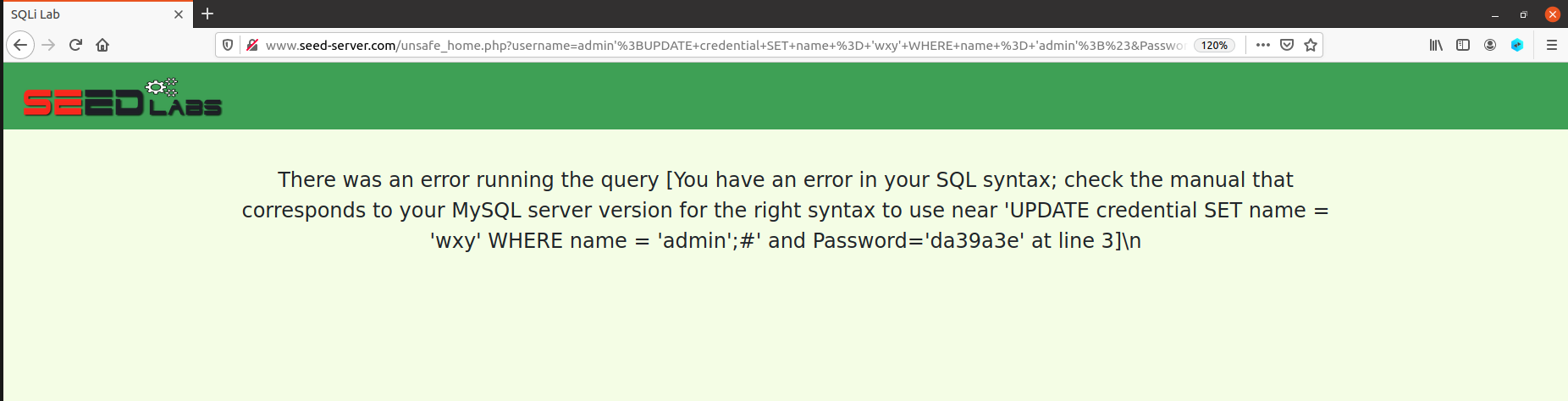




1. Task 2.3: Append a new SQL statement

利用“；”将注入攻击改为多条SQL语句，输入的内容为：admin';UPDATE credential SET name = 'wxy' WHERE name = 'admin';#



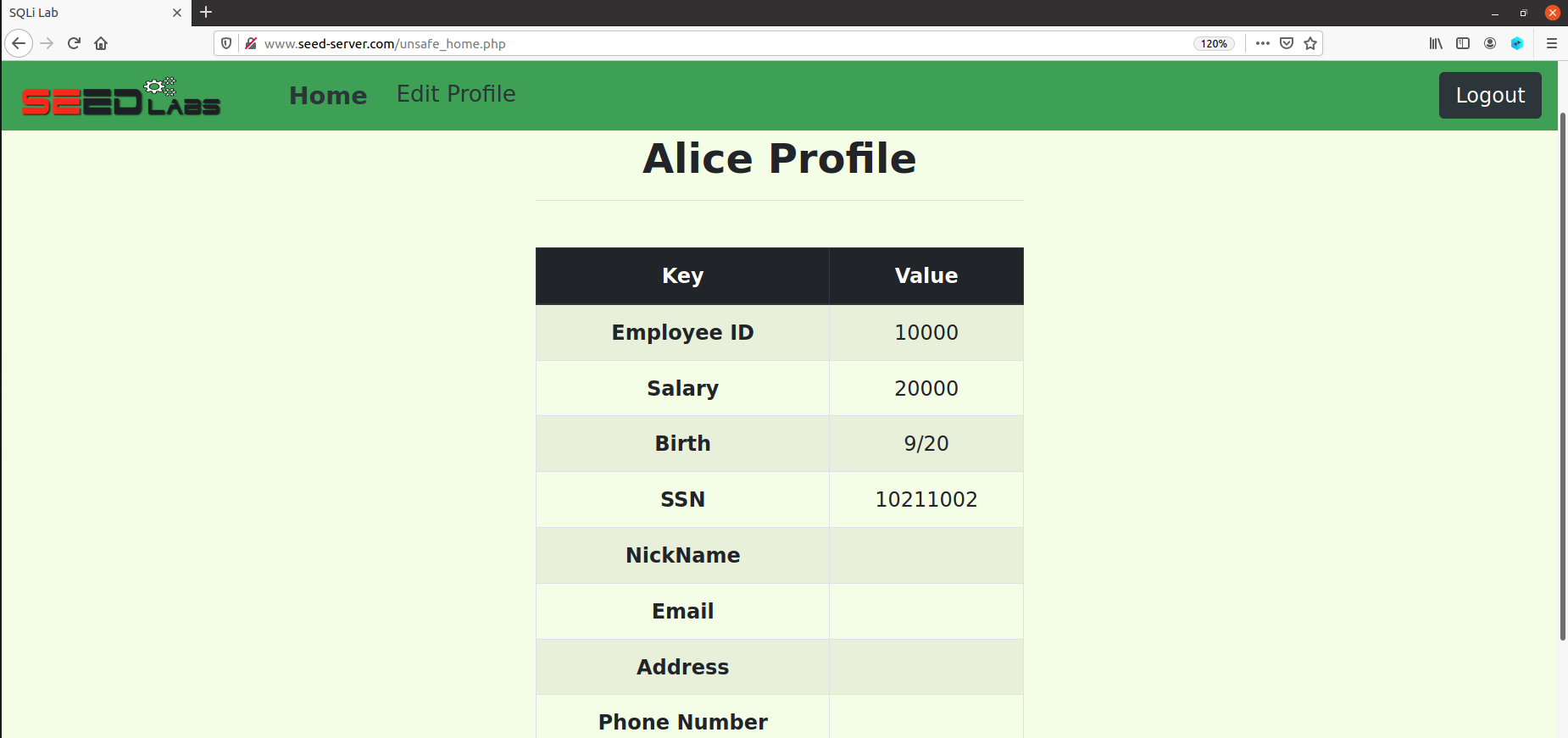


不过从结果可知攻击无效，查明是因为PHP中mysqli扩展的query()函数不允许在数据库服务器中运行多条语句，是对SQL注入攻击的一种防护措施。

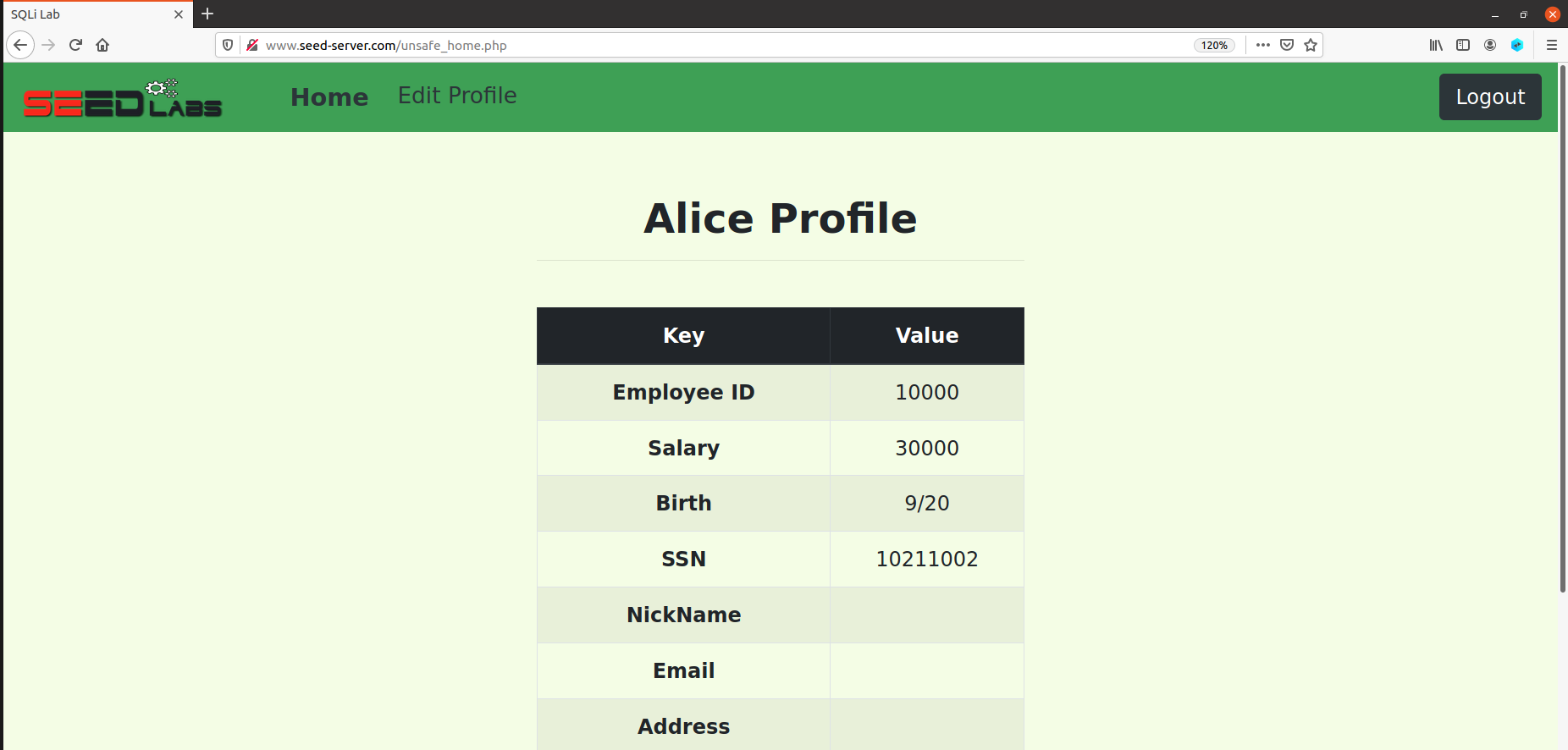
**Task 3: Stealing Cookies from the Victim’s Machine**

**1． Task 3.1: Modify your own salary**

①登录Alice账户查看工资salary为20000



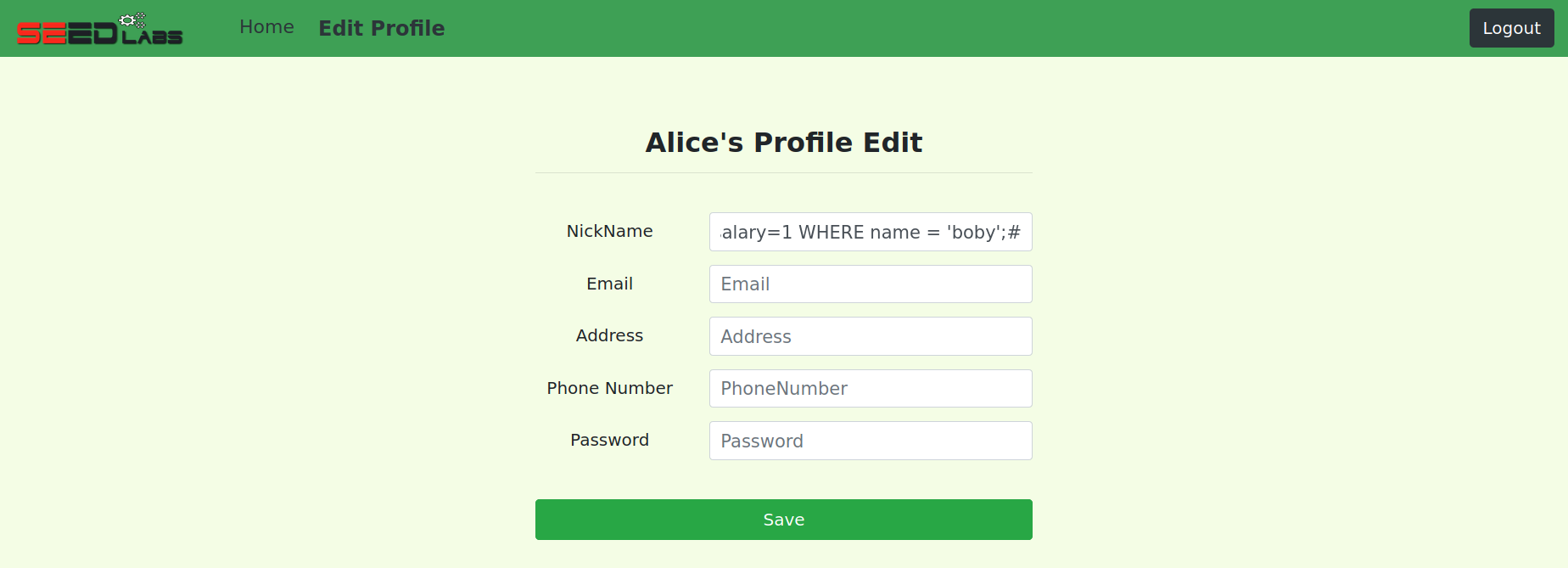
②打开Edit Profile 页面，输入命令 ',salary=999999 WHERE name = 'alice';# 使工资salary增加为30000



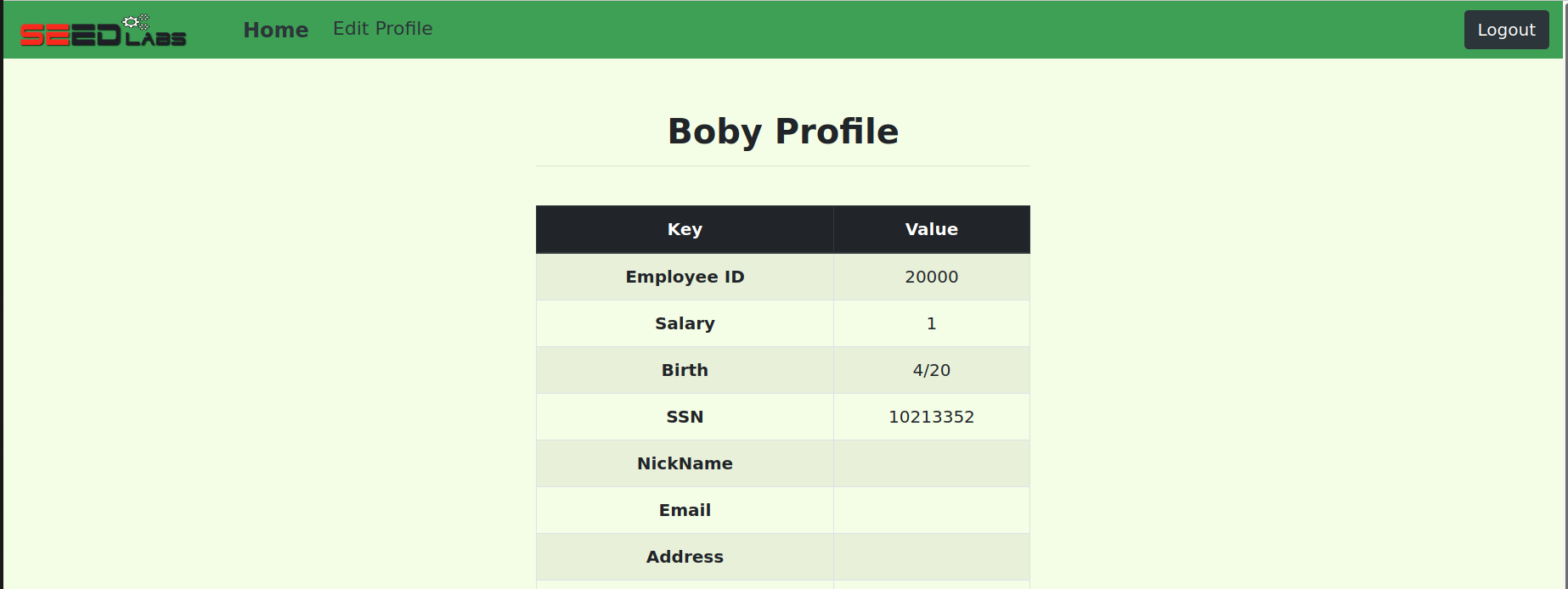
工资已改变，攻击生效！

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

①再次打开Edit Profile 页面，输入命令 ',salary=1 WHERE name = 'boby';# ，试图修改老板boby的工资

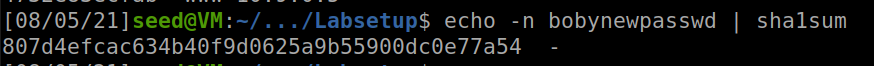


1. 在登陆老板boby的账户查看工资，确实改变为1

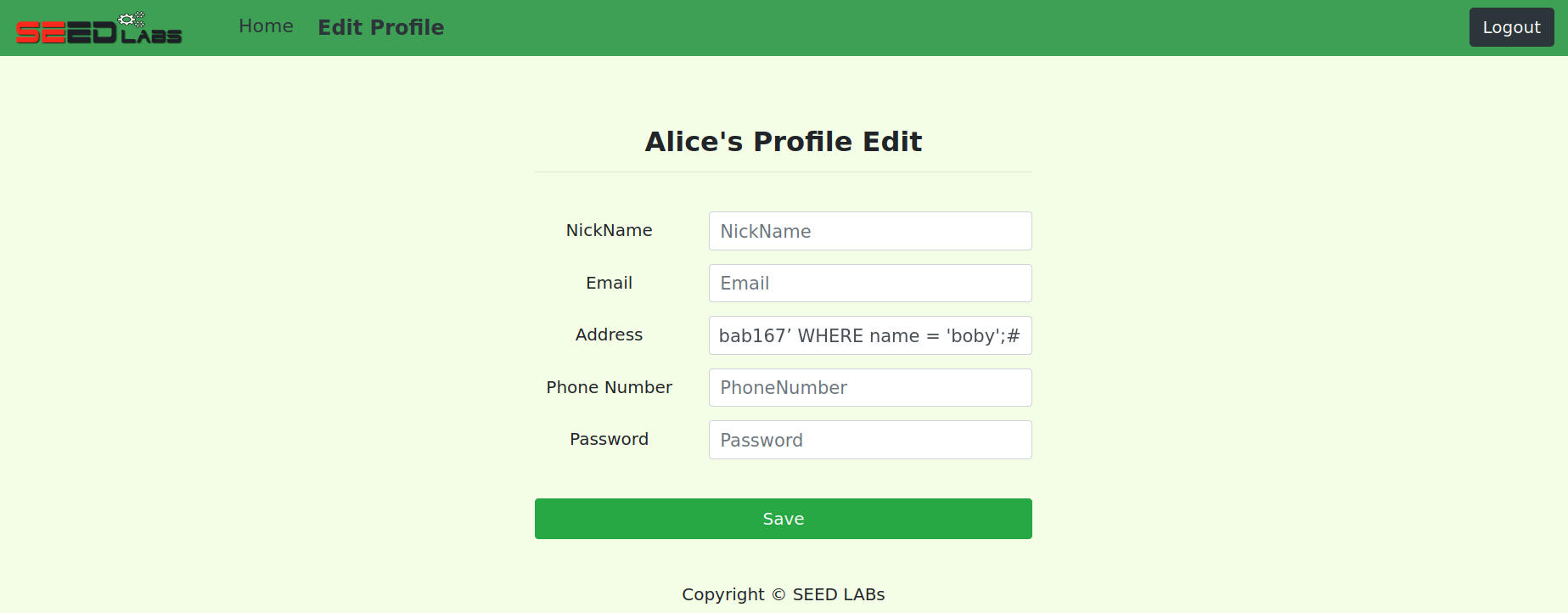


1. **Task 3.3: Modify other people’ password**

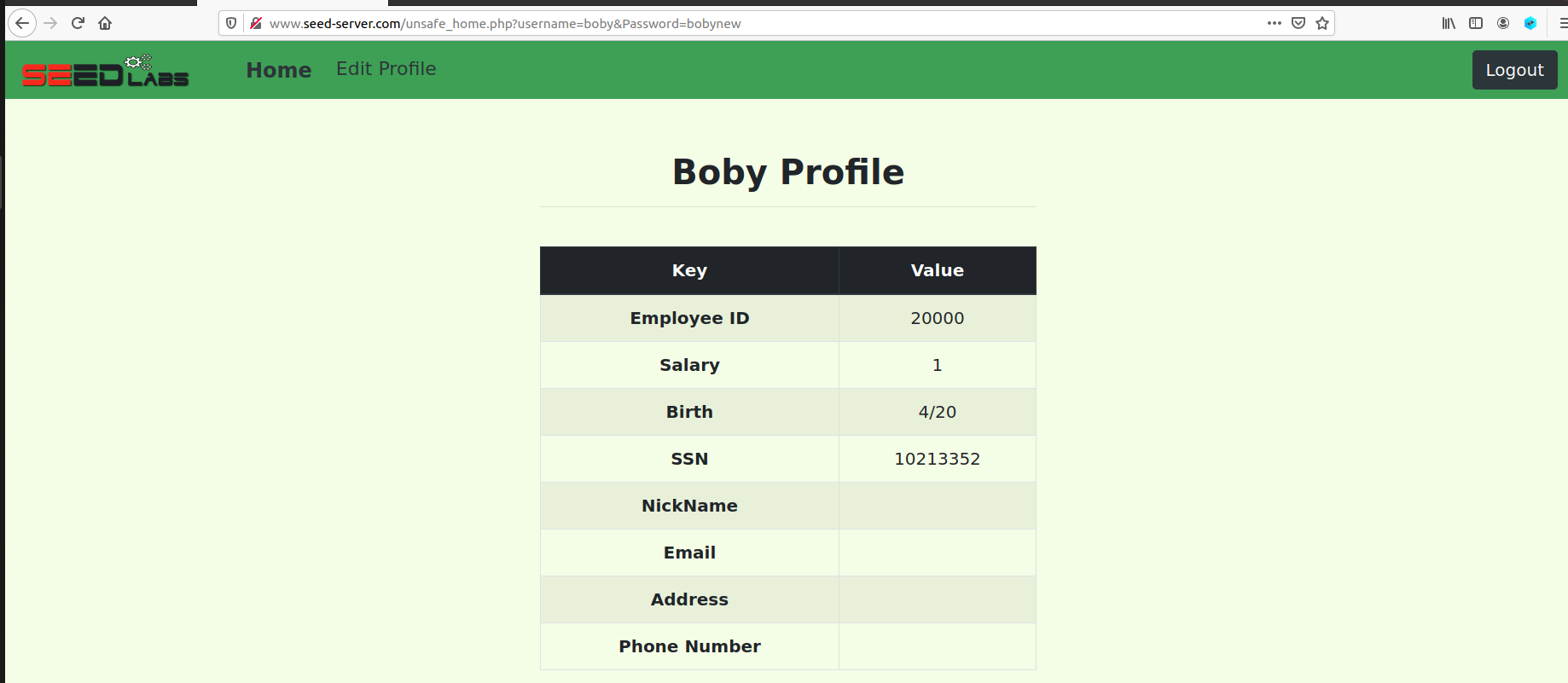
**①**获取新密码bobynewpasswd 的哈希值



②通过SQL注入攻击将boby的密码进行更改（命令为：',Password=‘0f080987ef5313fd24efe534c409292997bab167’ WHERE name = 'boby';#）

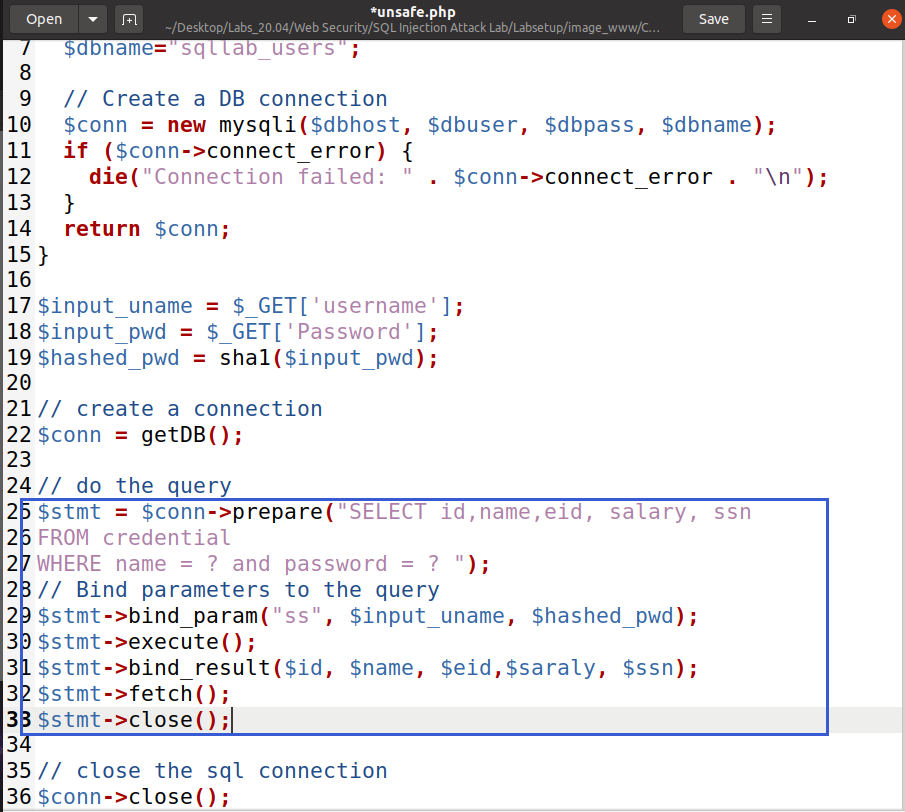


1. 使用新密码尝试登录，成功

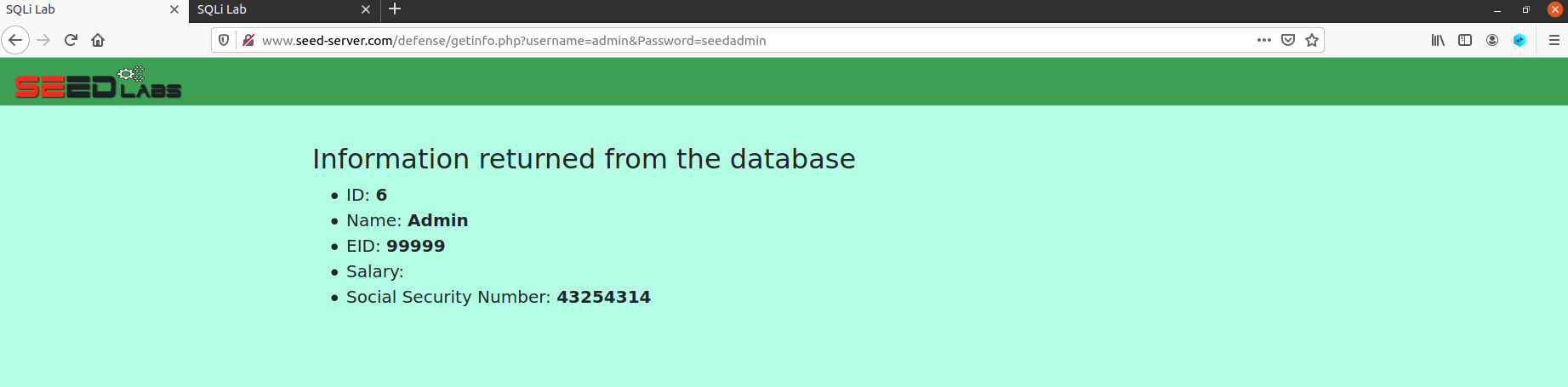


**Task 4: Countermeasure — Prepared Statement**

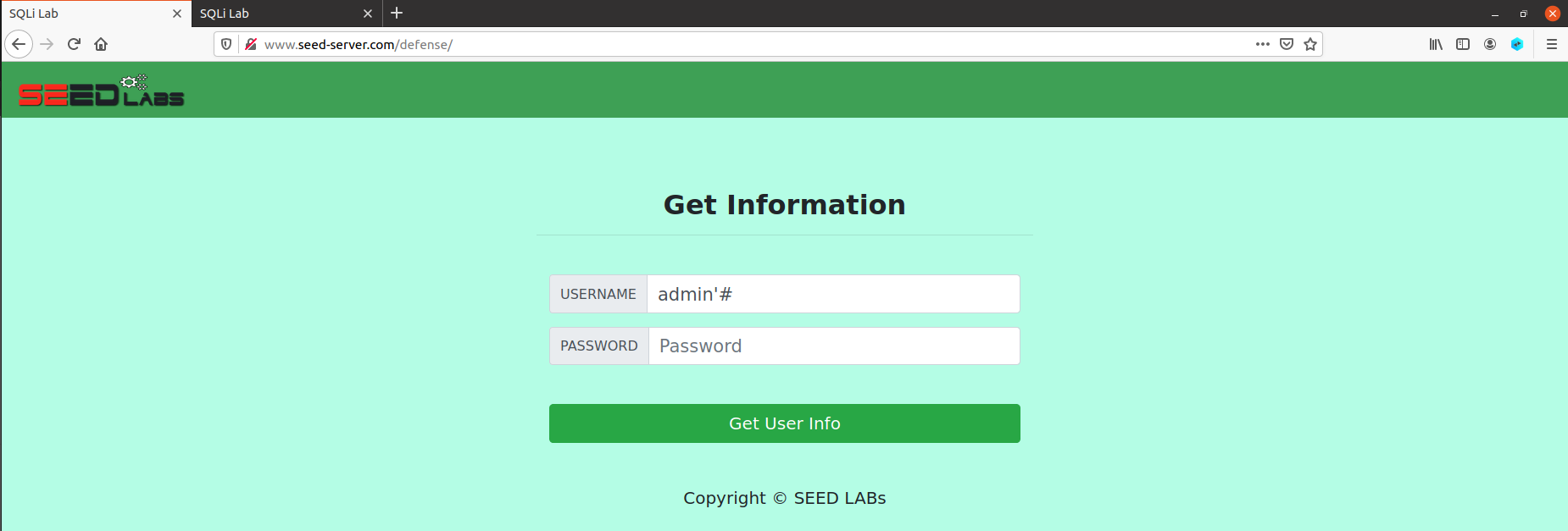
1.打开image\_www/Code/defense下的unsafe.php文件，依据示例增加修改过的预处理SQL语句，然后重新打开容器；

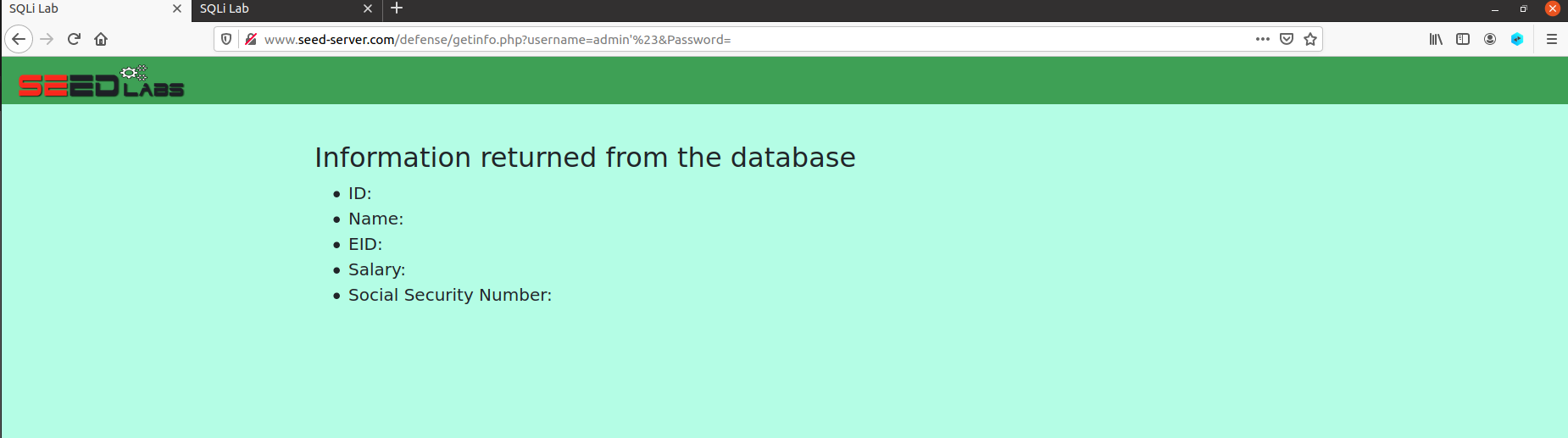


2.打开<http://www.seed-server.com/defense/>，登录admin管理者账号



3.尝试进行SQL注入攻击





但未能像task1中返回所有数据库中的用户信息，说明攻击失败，预处理语句的防御成功！

**实验总结：**

本次是第六次实验，但还是在实验中犯了几个小问题导致实验一直难以进行下去。

1. 在mysql输入命令行时忘记在末尾添加“；”；
2. 忘记先进入mysql的容器再登录数据库系统；
3. 在task3.2中更改boby的工资，以为使用SQL注入攻击后会直接跳转到boby的信息页，导致一致误认为实验失败。。。

经过实验，我也对本次SQL注入攻击进行了学习，总结如下；

1. SQL注入攻击是指构造错误格式的SQL语句（如添加符号‘、#实现）来注入恶意代码，并让数据库执行这种恶意行为的攻击；
2. SQL注入攻击可以通过网页应用程序和cURL两种方式实现攻击；多条语句的注入攻击通过分号；连接起来；
3. 防御SQL注入攻击的思想主要是过滤掉代码或者将代码和数据分开。预处理语句是采用将数据和代码分离开这种方法。