Họ và tên: Lương Chin Du MSSV:21110870

Môn học: An toàn thông tin Mã môn: INSE330380\_23\_1\_09

Giảng viên hướng dẫn: Huỳnh Nguyên Chính

**Lab 5: SQL Injection**

1. **Review the lab environment**

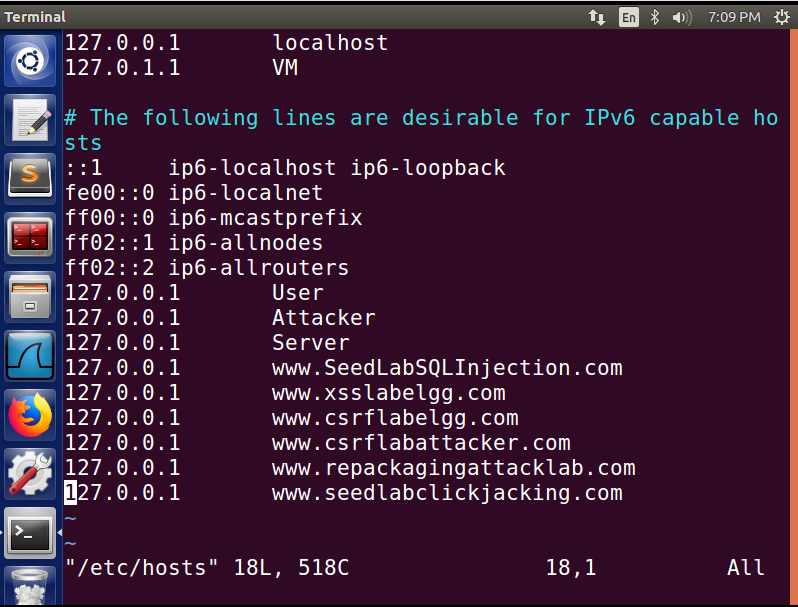
Xem xét môi trường của Lab:



**Tiến hành thực thi 2 câu lệnh:**

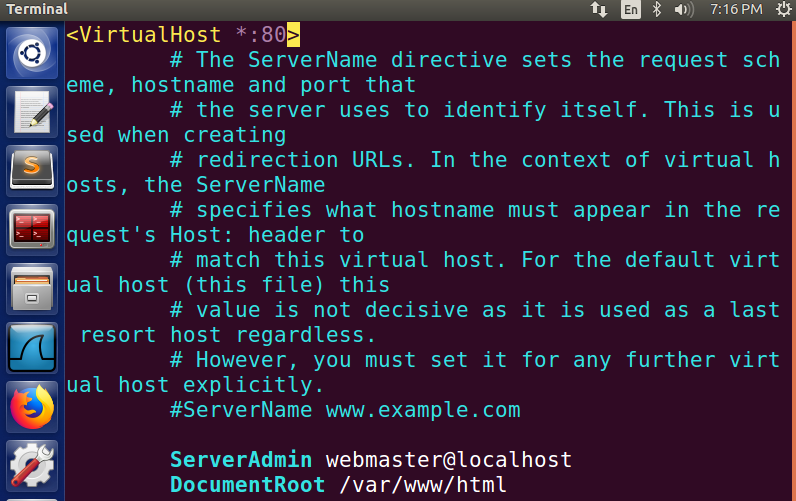
**#vi /etc/host**

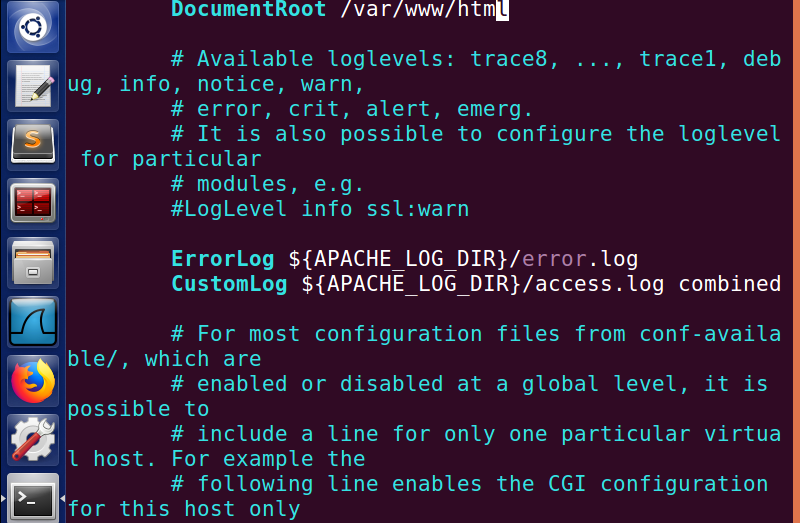


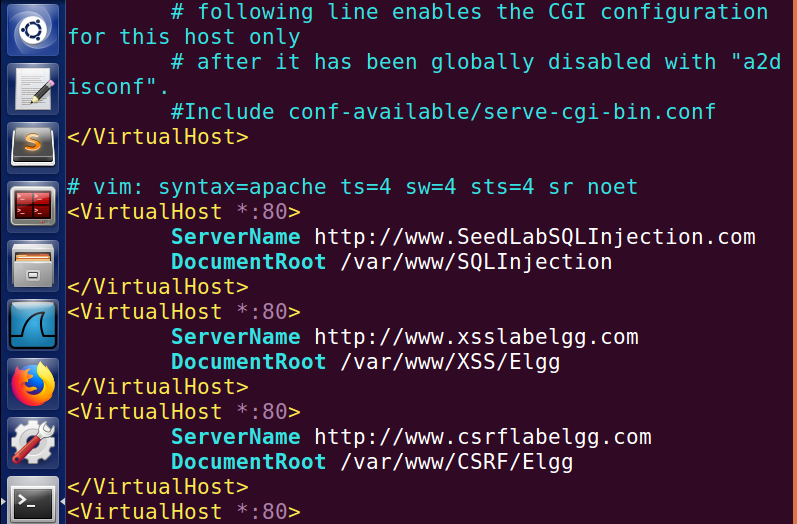


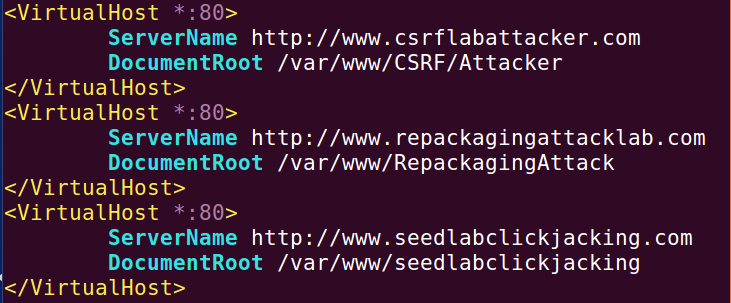
**# /etc/ apache2/sites-available/ 000-default.conf**











1. **Get Familiar with SQL Statements**

Ta có thể chuyển qua công cụ MySQL bằng lệnh:

$ mysql -uroot -pseedubuntu

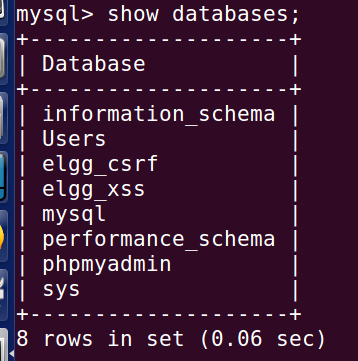
-u: user

-p: password



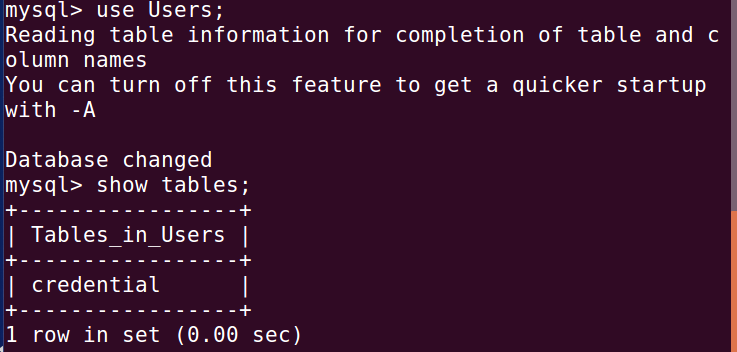
Xem các database có trong mySQL:

**mysql> show databases;**

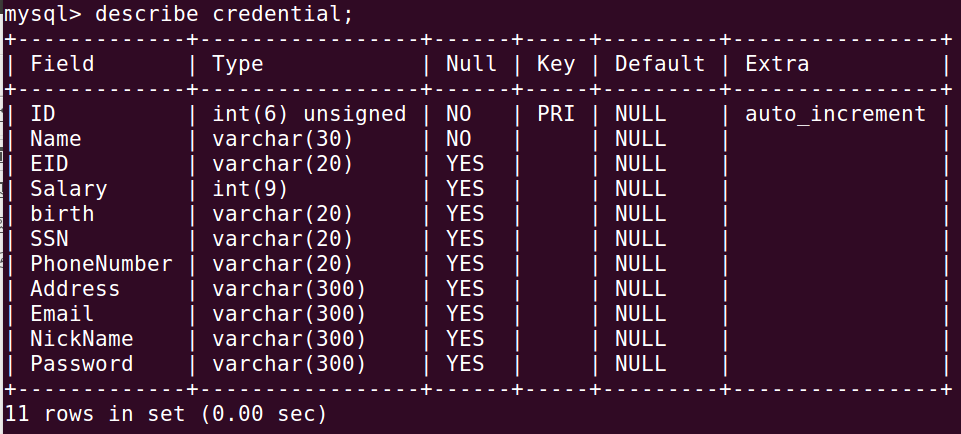


Ta sẽ sử dụng database mang tên Users: **use Users**

Và hiển thị ra các bảng của Users bằng: **show tables;**



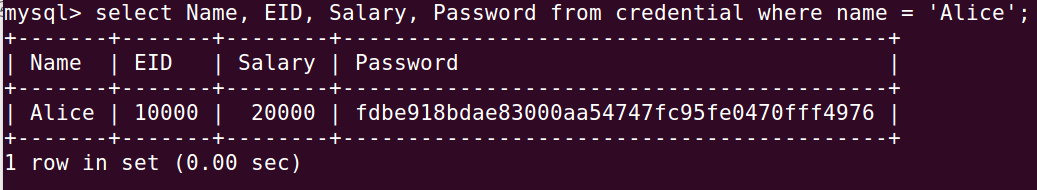
**Describe Credential:** Xem cấu trúc của table credential



* Ta thấy có 11 thuộc tính và các kiểu dữ liệu cũng như các ràng buộc... của credential.

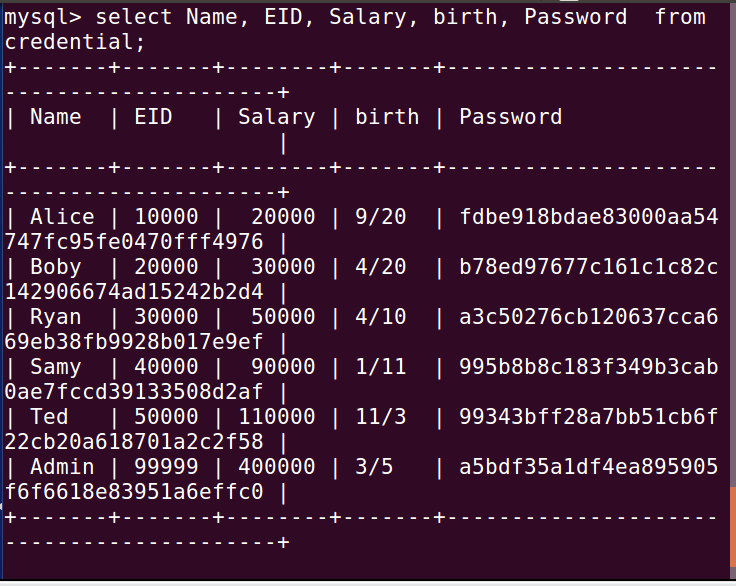
In ra một số thông tin như Tên, Mã nhân viên, Lương, Mật khẩu của nhân viên Alice

Bằng lệnh: **select Name, EID, Salary, Password from credential where name=’Alice’;**



In ra các thông tin Tên, Mã nhân viên, Lương, Mật khẩu của tất cả các nhân viên trong bảng credential:

**Select Name, EID, Salary, birth, Password from credential;**



* Các password đã được lưu dưới dạng mã, được băm ra(hashing)

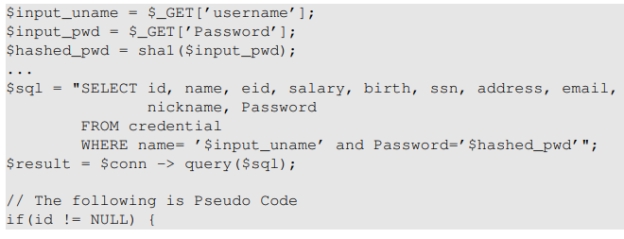
1. **SQL Injection Attack on SELECT Statement**

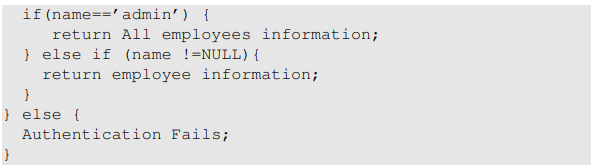
Chúng ta đăng nhập vào [www.SEEDLabSQLInjection.com](http://www.SEEDLabSQLInjection.com) qua trình duyệt trên ubuntu



The web application authenticate users based on these two pieces of data, so only employees who know their passwords are allowed to log in. Your job, as an attacker, is to log into the web application without knowing any employee’s credential.

To help you started with this task, we explain how authentication is implemented in the web application. The PHP code unsafe home.php, located in the **/var/www/SQLInjection** directory, is used to conduct user authentication. The following code snippet show how users are authenticated





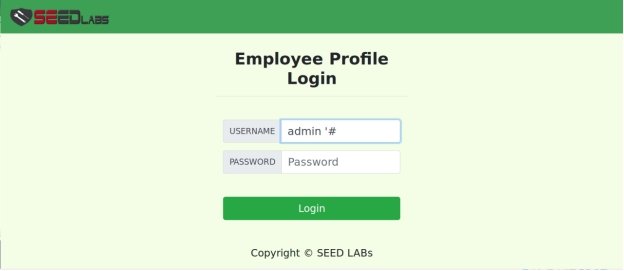
1. **SQL Injection Attack from webpage (**Các cuộc tấn công SQL Injection từ trang web)

Your task is to log into the web application as the administrator from the login page, so you can see the information of all the employees. We assume that you do know the administrator’s account name which is admin, but you do not the password. You need to decide what to type in the Username and Password fields to succeed in the attack.

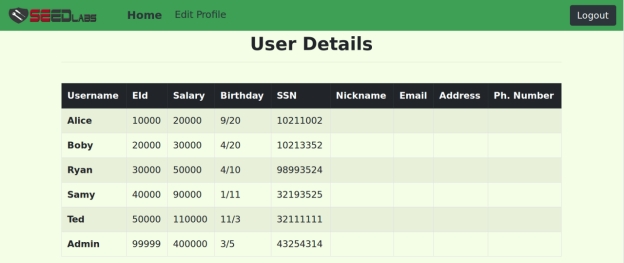
(Nhiệm vụ của bạn là đăng nhập vào ứng dụng web với tư cách là quản trị viên từ trang đăng nhập, để bạn có thể xem thông tin của tất cả các nhân viên. Chúng tôi giả định rằng bạn biết tên tài khoản của quản trị viên là quản trị viên, nhưng bạn không biết mật khẩu. Bạn cần quyết định nhập những gì vào trường Tên người dùng và Mật khẩu để tấn công thành công)

Tìm kiếm trang: <http://www.seedlabsqlinjection.com/> trên Ubuntu

→ Tiến hành đăng nhập vào với tư cách admin thông qua việc nhập username: admin ‘# (sau #các lệnh sẽ không được thực thi)



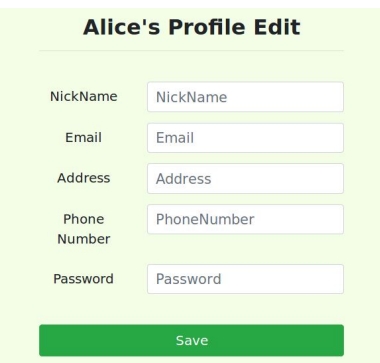
→ Vào hệ thống và xem thông tin



1. **SQL Injection Attack on UPDATE Statement (**Cuộc tấn công SQL Injection trên câu lệnh UPDATE)

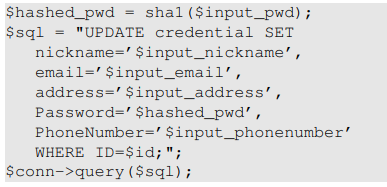
If a SQL injection vulnerability happens to an UPDATE statement, the damage will be more severe, because attackers can use the vulnerability to modify databases. In our Employee Management application, there is an Edit Profile page that allows employees to update their profile information, including nickname, email, address, phone number, and password

(Nếu một lỗ hổng SQL injection xảy ra với câu lệnh UPDATE, thiệt hại sẽ nghiêm trọng hơn, vì những kẻ tấn công có thể sử dụng lỗ hổng này để sửa đổi cơ sở dữ liệu. Trong ứng dụng Quản lý nhân viên của chúng tôi, có trang Chỉnh sửa hồ sơ cho phép nhân viên cập nhật thông tin hồ sơ của họ, bao gồm biệt hiệu, email, địa chỉ, số điện thoại và mật khẩu)



When employees update their information through the Edit Profile page, the following SQL UPDATE query will be executed. The PHP code implemented in unsafe edit backend.php file is used to update employee’s profile information. The PHP file is located in the /var/www/SQLInjection directory.

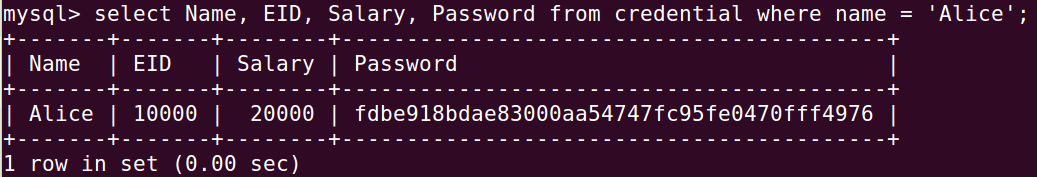
(Khi nhân viên cập nhật thông tin của họ thông qua trang Chỉnh sửa Hồ sơ, truy vấn CẬP NHẬT SQL sau sẽ được thực hiện. Mã PHP được triển khai trong tệp backend.php chỉnh sửa không an toàn được sử dụng để cập nhật thông tin hồ sơ của nhân viên. Tệp PHP nằm trong /var/www/SQLInjection directory.)

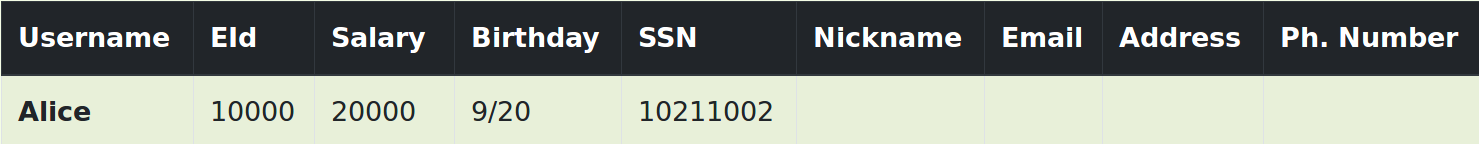


**Task 5.1: Modify your own salary. As shown in the Edit Profile page, employees can only update their nicknames, emails, addresses, phone numbers, and passwords; they are not authorized to change their salaries. Assume that you (Alice) are a disgruntled employee, and your boss Boby did not increase your salary this year. You want to increase your own salary by exploiting the SQL injection vulnerability in the Edit-Profile page. Please demonstrate how you can achieve that. We assume that you do know that salaries are stored in a column called ’salary’.**

**Bước 1: Xem “Salary” của Alice:**

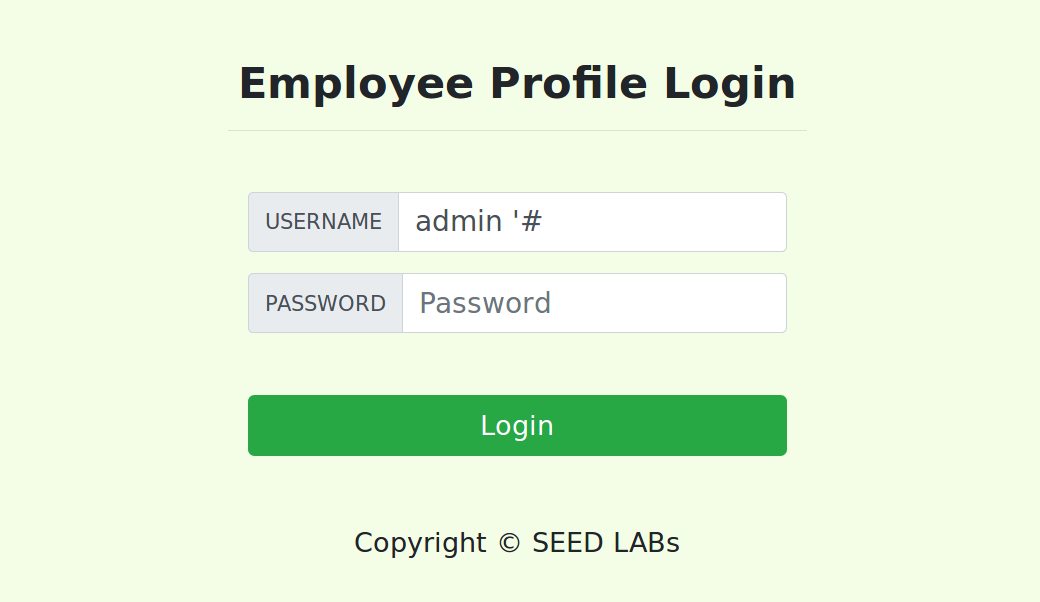
Ta thấy rằng lương ban đầu của Alice là 20000



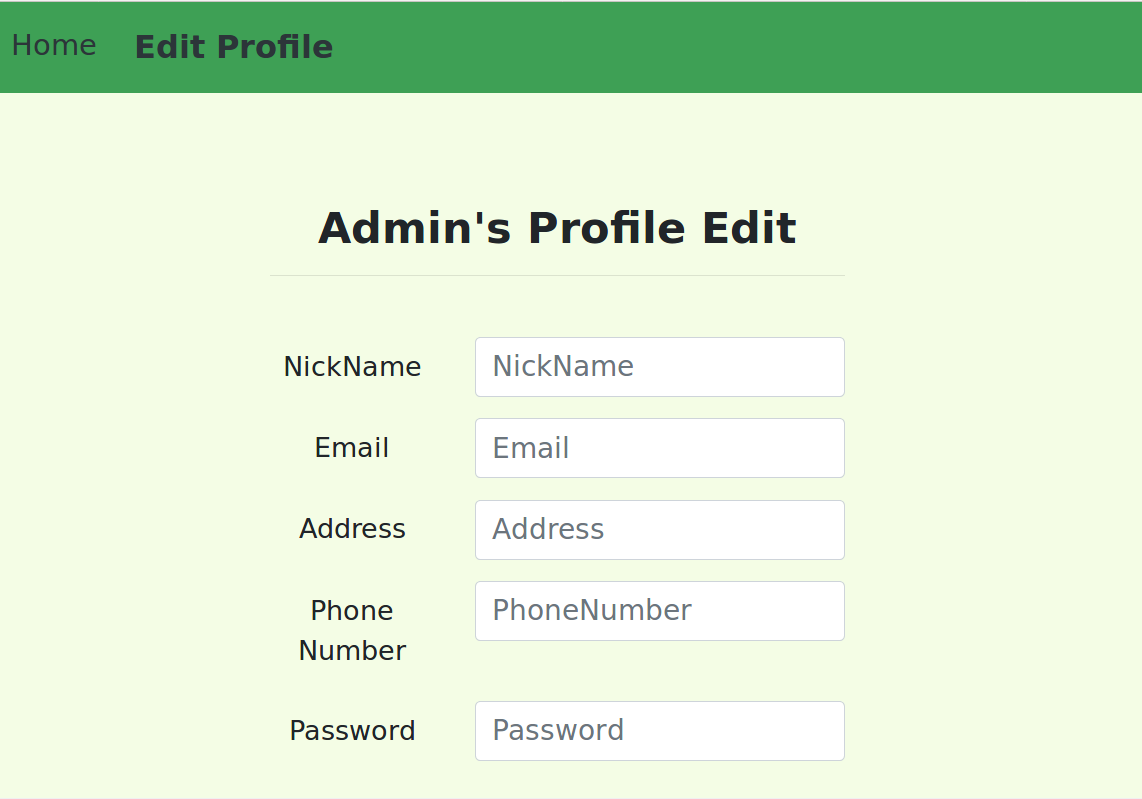


**Bước 2: Tiến hành tìm kiếm và đăng nhập:**

* Vào trang: <http://www.seedlabsqlinjection.com/>
* Đăng nhập: Ta có thể dùng SQL injection để đăng nhập vào hệ thống với quyền admin bằng username là: admin’# (vì sau dấu # là các lệnh không thực thi), không cần biết mật khẩu vẫn có thể đăng nhập được.



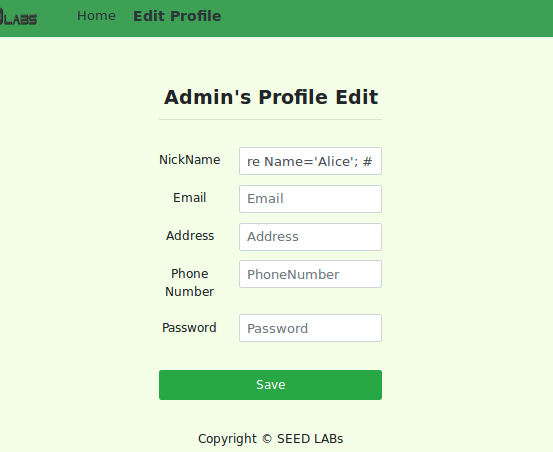
* Sau khi đã đăng nhập thành công với quyền admin, ta vào **Edit Profile**



**Bước 3: Tiến hành để thay đổi “Salary” của Alice**

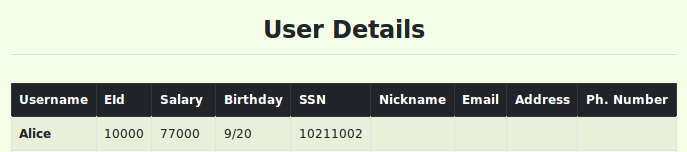
Click vào khung nickname và nhập lệnh

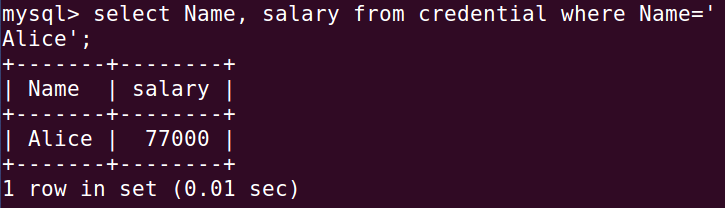
**‘, salary = 77000 where Name = ‘Alice’; #**



* Nhấn Save để hoàn tất chỉnh sửa

Tiếp đến ta check các thuộc tính đã chỉnh sửa, xem có thay đổi hay chưa?



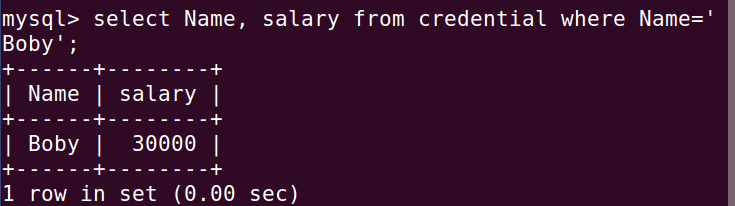


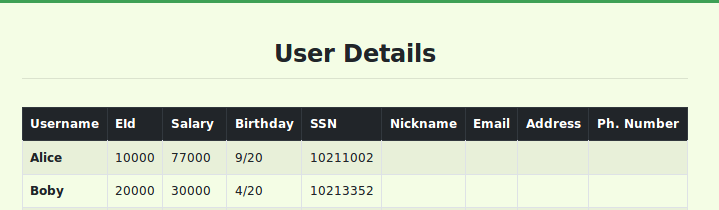


🡪 Như 2 ảnh trên, kết quả của việc chỉnh sửa lương Alice đã thành công.

**5.2** **Modify other people’ salary. After increasing your own salary, you decide to punish your boss Boby. You want to reduce his salary to 1 dollar. Please demonstrate how you can achieve that.**

**Bước 1: Xem “Salary” của Boby:**

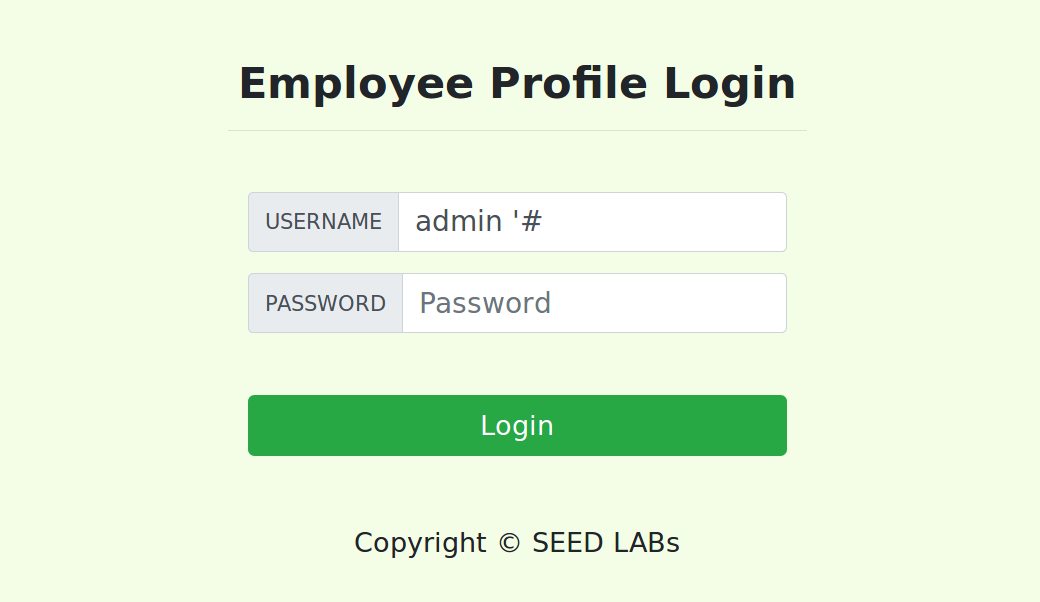




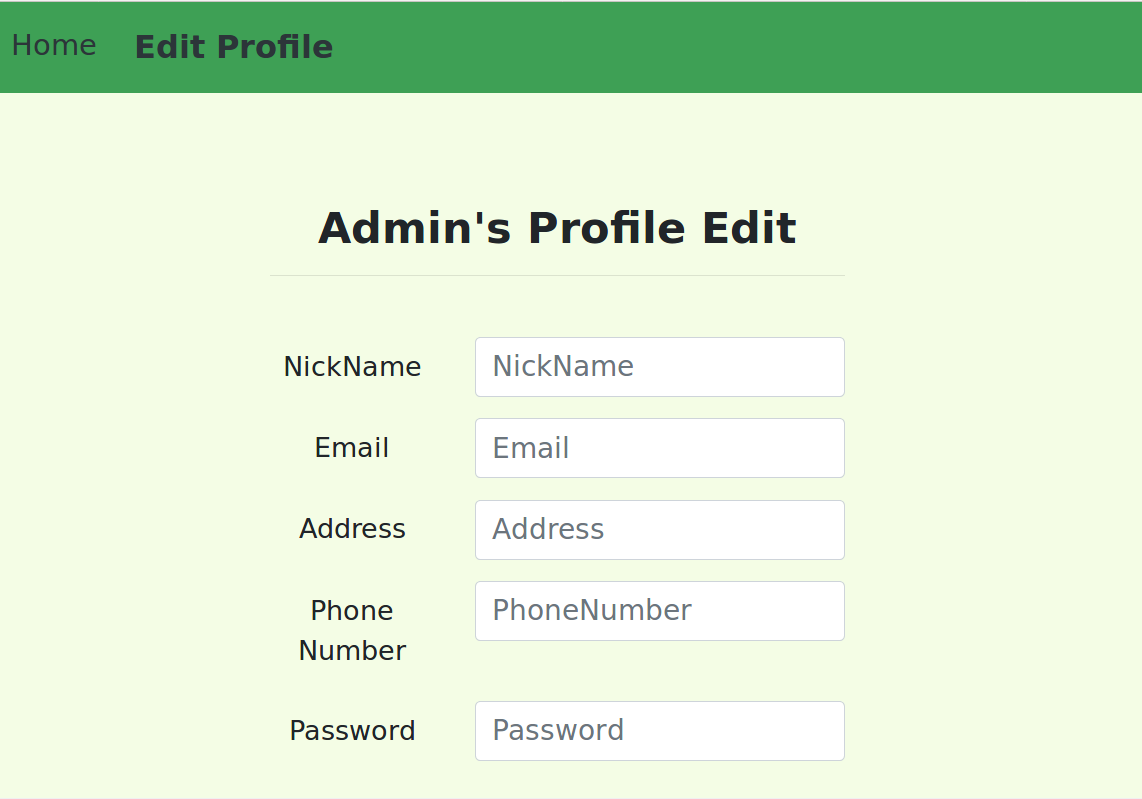
**Bước 2: Tiến hành tìm kiếm và đăng nhập:**

- Vào trang: <http://www.seedlabsqlinjection.com/>

- Đăng nhập: Ta có thể dùng SQL injection để đăng nhập vào hệ thống với quyền admin bằng username là: admin’# (vì sau dấu # là các lệnh không thực thi), không cần biết mật khẩu vẫn có thể đăng nhập được.



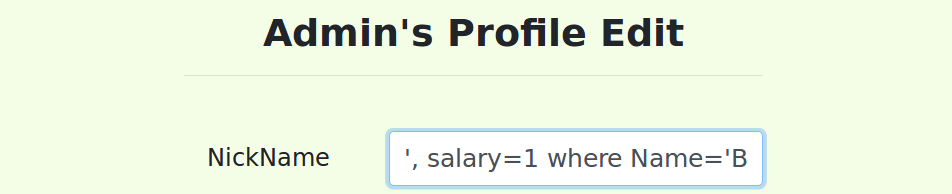
* Sau khi đã đăng nhập thành công với quyền admin, ta vào **Edit Profile**

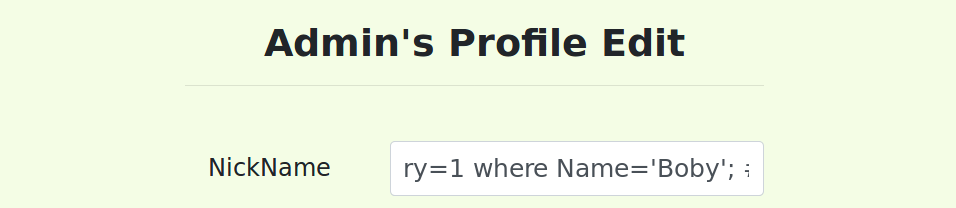


**Bước 3: Tiến hành để thay đổi “Salary” của Alice**

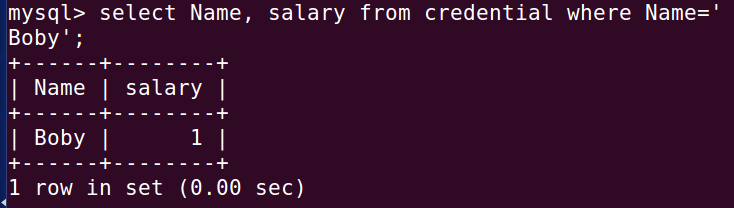
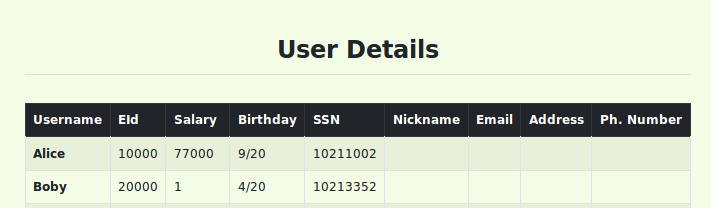
Click vào khung nickname và nhập lệnh

**‘, salary = 1 where Name = ‘Boby’; #**





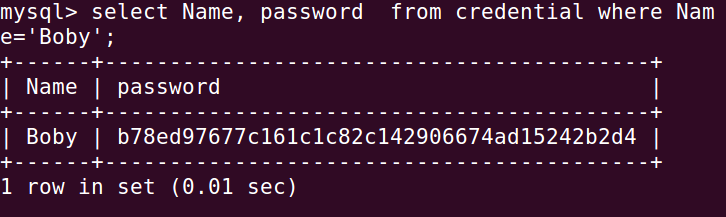
* Nhấn Save để hoàn tất chỉnh sửa



* **Kết quả:** đã đổi “Boby” có lương ban đầu là 3000 thành 1.

**5.3** **Modify other people’ password. After changing Boby’s salary, you are still disgruntled, so you want to change Boby’s password to something that you know, and then you can log into his account and do further damage. Please demonstrate how you can achieve that. You need to demonstrate that you can successfully log into Boby’s account using the new password. One thing worth mentioning here is that the database stores the hash value of passwords instead of the plaintext password string. You can again look at the unsafe edit backend.php code to see how password is being stored. It uses SHA1 hash function to generate the hash value of password**

Trước hết ta phải xem mật khẩu của Boby**:**

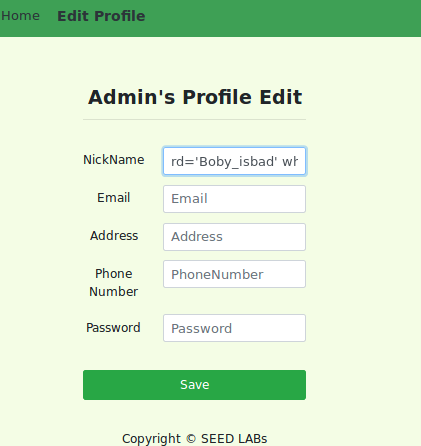


* Mật khẩu ban đầu ở 1 dạng mã đã được băm ra(hashing).

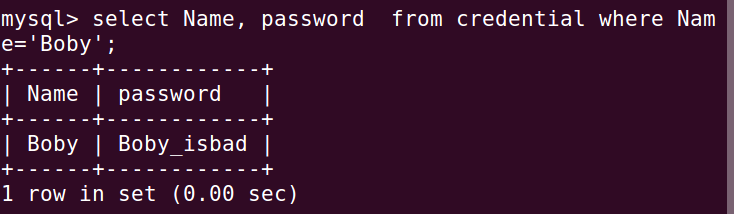
Chúng ta sẽ sửa đổi lại mật khẩu của Boby:

**Bước 1: Vào Edit Profile tiến hành đổi password, giả sử ta đặt password mới cho Boby là “Boby\_isbad” thông qua lệnh:**

**‘, password=’Boby\_isbad’ where Name=’Boby’; #**

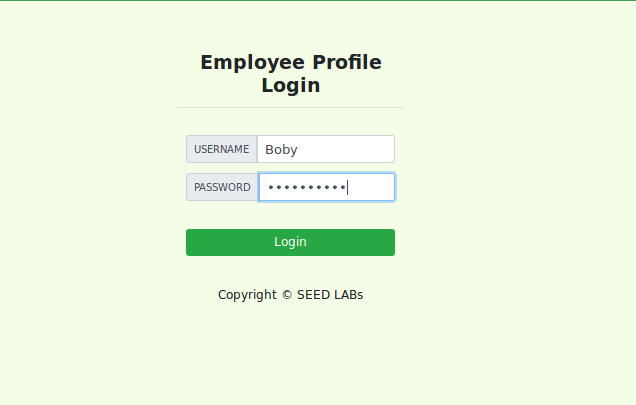


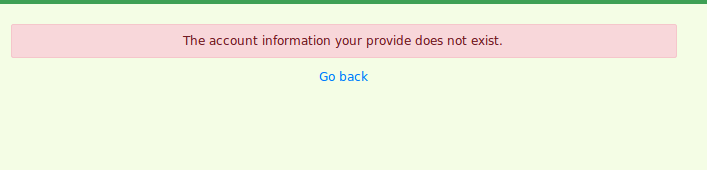
Tiếp đến, ta check xem mật khẩu sau khi đã đổi



* Mật khẩu mới đã được cập nhật thành công

Bắt đầu đăng nhập vào hệ thống với username là Boby và mật khẩu mới vừa cập nhật

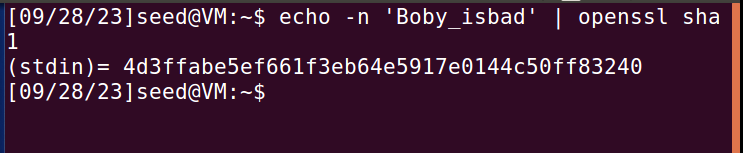




* Ta thấy không thể đăng nhập vào được khi ta nhập mật khẩu “Boby\_isbad” bởi vì nó đã được băm ra thành chuỗi kí tự khác thông qua quá trình hashing.

**Bước 2: ta vào Ubuntu mở Terminater chuyển mật khẩu “Boby\_isbad” thành giá trị hash thông qua SHA1 hash function.**

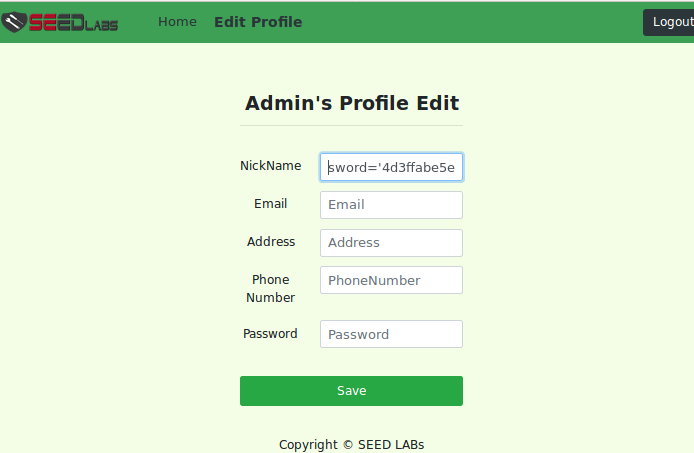
**echo -n ‘<Mat Khau>’ | openssl sha1**

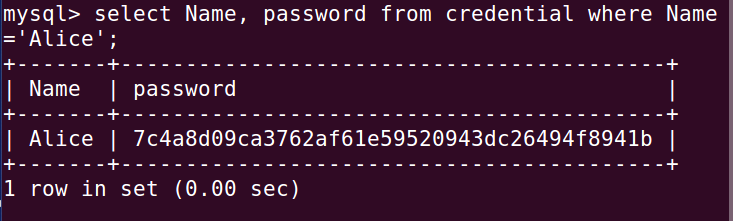


Sau khi chuyển đổi ta nhận được 1 chuỗi 4d3ffabe5ef661f3eb64e5917e0144c50ff83240 từ “Boby\_isbad”

* Thực hiện edit profile đặt mật khẩu là chuỗi kí tự này

**‘, Password=’4d3ffabe5ef661f3eb64e5917e0144c50ff83240’ where Name=’Boby’; #**

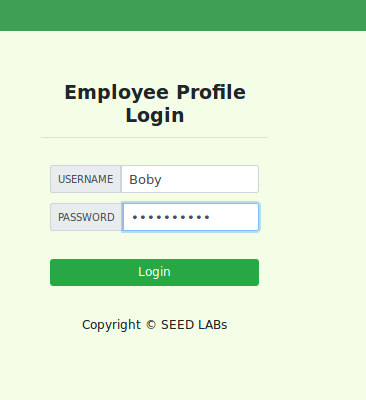


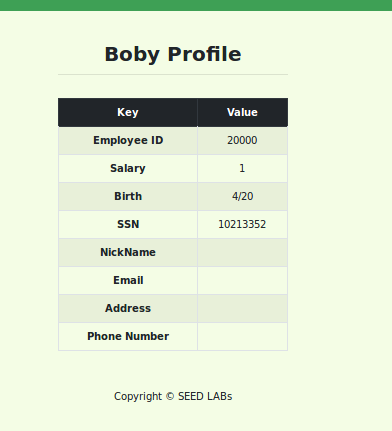


* Mật khẩu đã được cập nhật

**Bước 3: Sau đó tiến hành đăng nhập lại trên trang**

Đăng nhập lại vào hệ thống với username “Boby” và password là “Boby\_isbad”





* Đã đăng nhập vào được vào tài khoản của Boby.
* Việc sử dụng hashing mật khẩu là một phần quan trọng của bảo mật hệ thống và ngăn chặn việc truy cập trái phép vào thông tin cá nhân của người dùng. Khi ta cập nhật mật khẩu và lưu trữ chúng dưới dạng chuỗi băm, sẽ giúp tăng cường bảo mật và đảm bảo rằng người quản trị hệ thống không thể dễ dàng biết được mật khẩu ban đầu của người dùng.