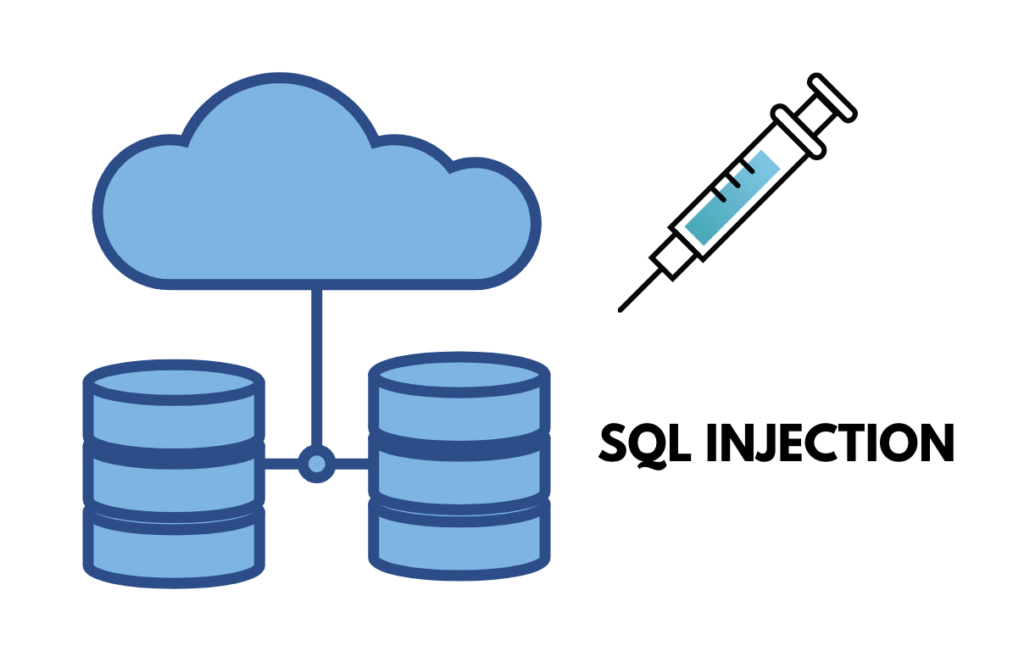
SQL Injection

# Định nghĩa

SQL Injection (SQLi) là một lỗ hổng bảo mật web cho phép kẻ tấn công can thiệp vào các truy vấn mà ứng dụng thực hiện đối với cơ sở dữ liệu. Điều này có thể cho phép kẻ tấn công xem dữ liệu mà thông thường chúng không thể truy xuất. Dữ liệu này có thể bao gồm dữ liệu thuộc về người dùng khác hoặc bất kỳ dữ liệu nào khác mà ứng dụng có thể truy cập. Trong nhiều trường hợp, kẻ tấn công có thể sửa đổi hoặc xóa dữ liệu này, gây ra những thay đổi liên tục đối với nội dung hoặc hành vi của ứng dụng.

Trong một số trường hợp, kẻ tấn công có thể leo thang tấn công SQL injection để xâm nhập máy chủ hoặc cơ sở hạ tầng back-end khác. Điều này cũng có thể cho phép chúng thực hiện các cuộc tấn công từ chối dịch vụ.



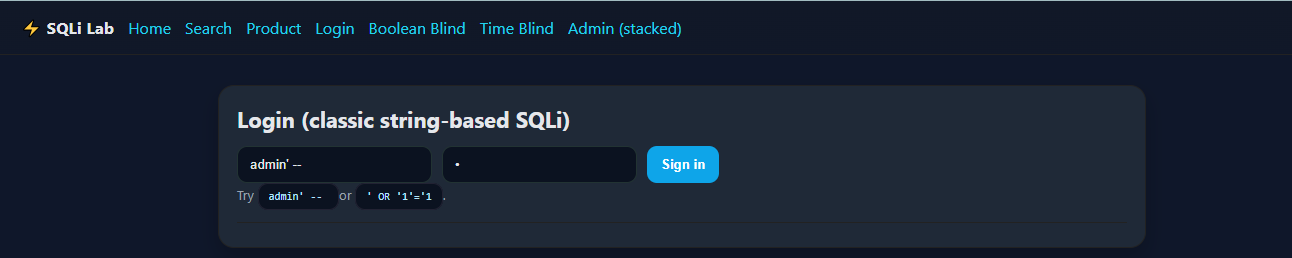
# Cách phát hiện:

* Ký tự dấu nháy đơn 'và tìm kiếm lỗi hoặc các bất thường khác.
* Một số cú pháp SQL cụ thể đánh giá theo giá trị cơ sở (gốc) của điểm vào và theo một giá trị khác, đồng thời tìm kiếm sự khác biệt có hệ thống trong phản hồi của ứng dụng.
* Các điều kiện Boolean như OR 1=1và OR 1=2, và tìm kiếm sự khác biệt trong phản hồi của ứng dụng.
* Payload được thiết kế để kích hoạt độ trễ thời gian khi thực hiện trong truy vấn SQL và tìm kiếm sự khác biệt về thời gian phản hồi.
* Payload OAST được thiết kế để kích hoạt tương tác mạng ngoài băng tần khi được thực hiện trong truy vấn SQL và giám sát mọi tương tác phát sinh.

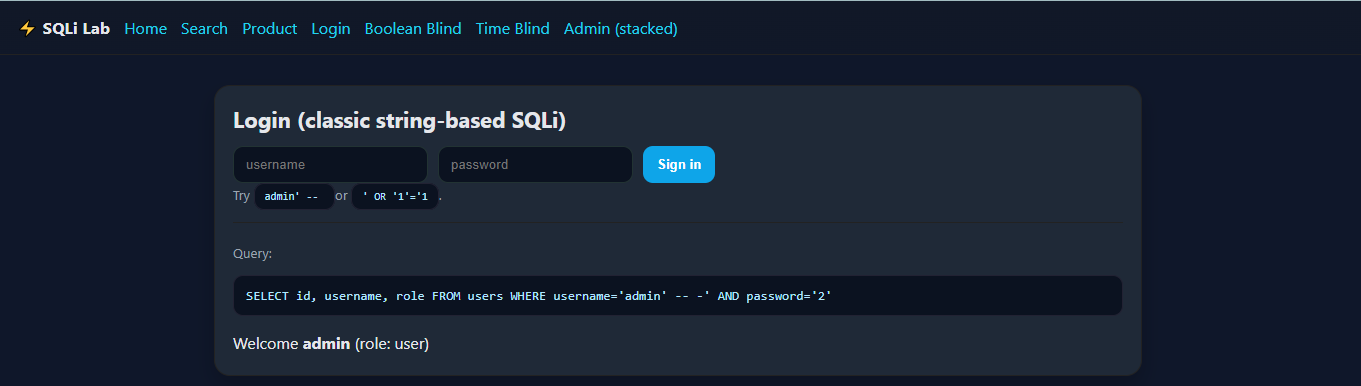
# Các case cơ bản thực hiện:

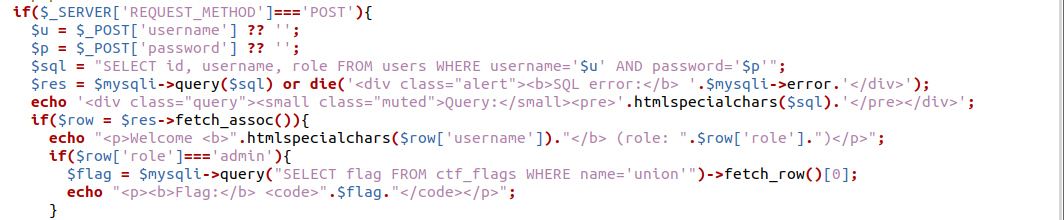
## Case 1: SQL Injection cơ bản

Thực hiện chèn thêm ‘ –- + sau admin hoặc ‘ OR 1=1



Chúng ta đã có thể login vào tài khoản admin, có thể nhìn thấy sau khi truyền payload vào thì query sẽ comment lại phần mệnh đề đằng sau dẫn đến server chỉ check user.



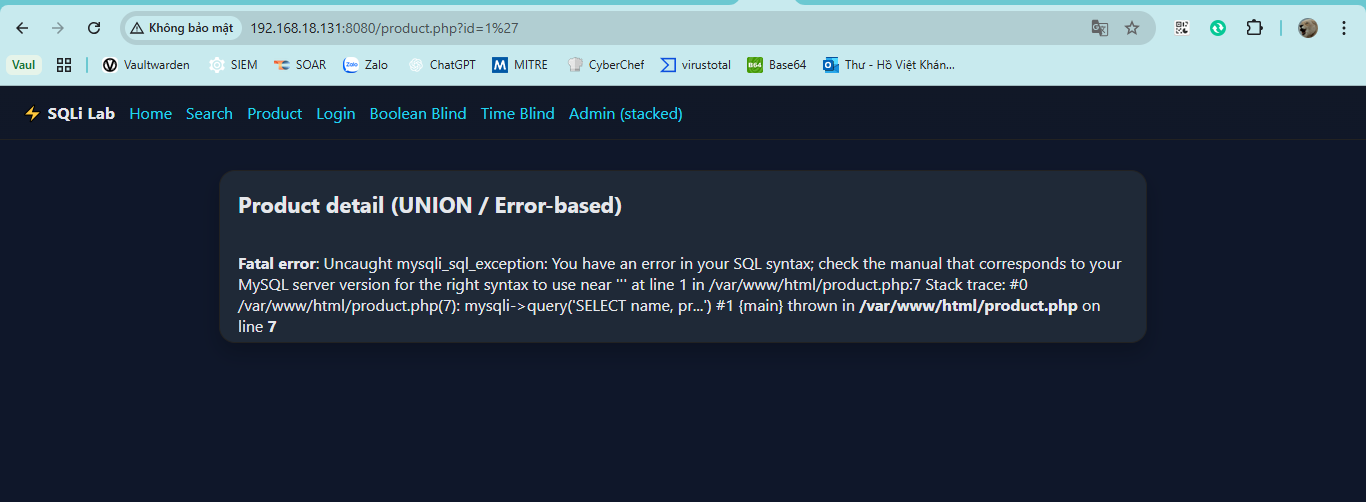


Phần lỗi nằm ở việc $u và $p được chèn thẳng vào query nên kẻ tấn công có thể làm sai lệch mệnh đề WHERE để bypass đăng nhập hoặc đọc dữ liệu.

Cách khắc phục: Dùng prepared statements + bind parameters ($stmt = $mysqli->prepare(...); $stmt->bind\_param('ss',$u,$pHash);).

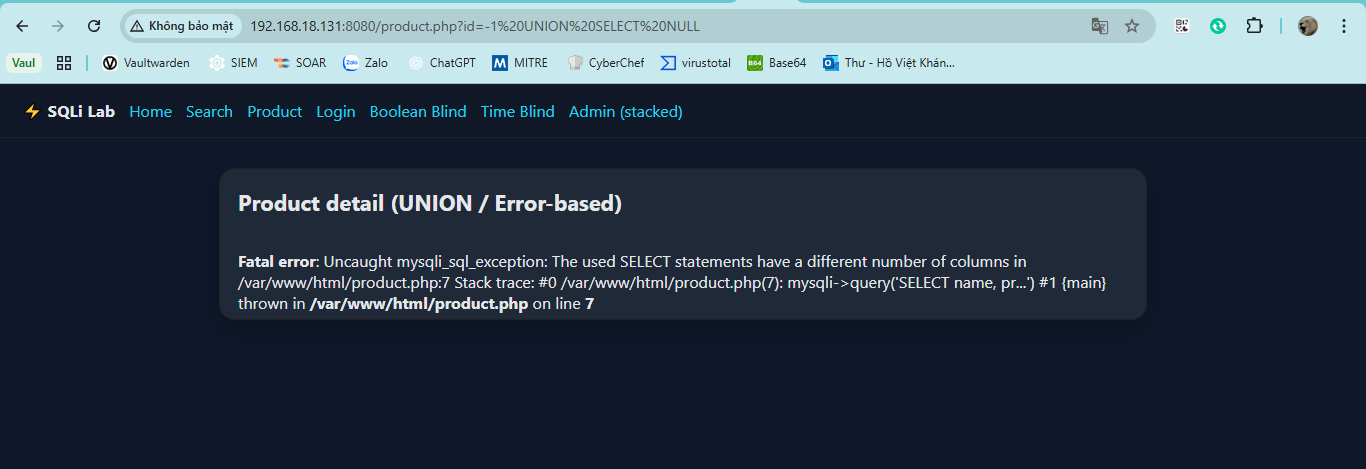
Case 2: SQL injection UNION

Thử chèn ‘ vào sau id=1 chúng ta nhận thấy server đang trả về lỗi liên quan đến syntax dẫn đến việc có thể khai thác sql injection.



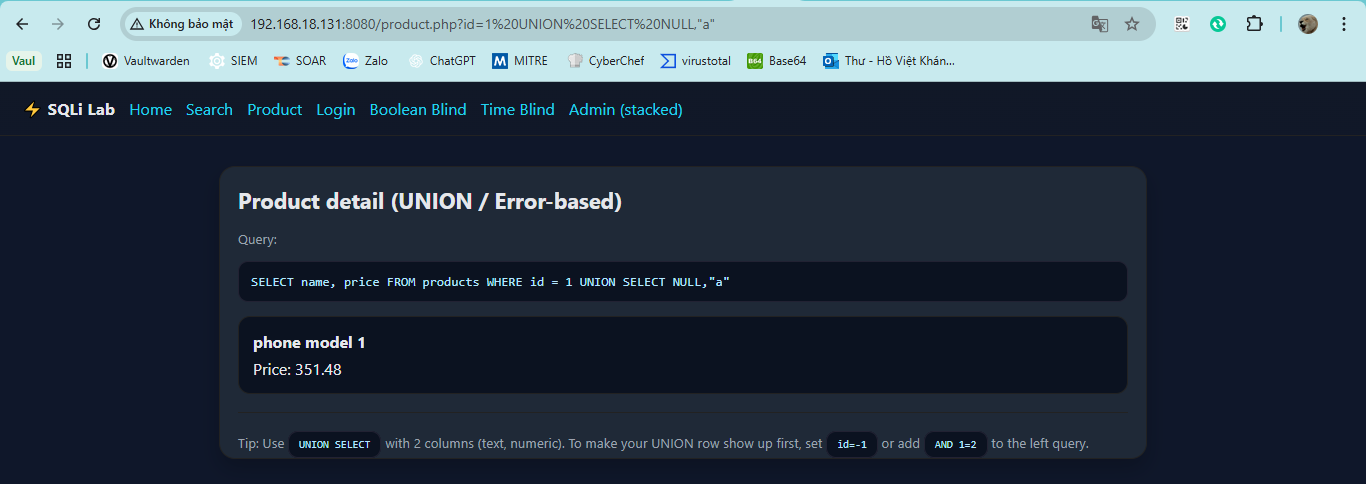
Tiếp theo chúng ta thử UNION, với UNION thì cần phải xem vế bên trái select bao nhiêu thì vế bên phải sẽ select đúng bằng vế bên trái. Để làm điều đó chúng ta sẽ thử UNION SELECT NULL đến lúc nào kết quả trả về có sự khác biệt.

Có thể thấy với lần NULL đầu tiên, kết quả trả về là lỗi syntax

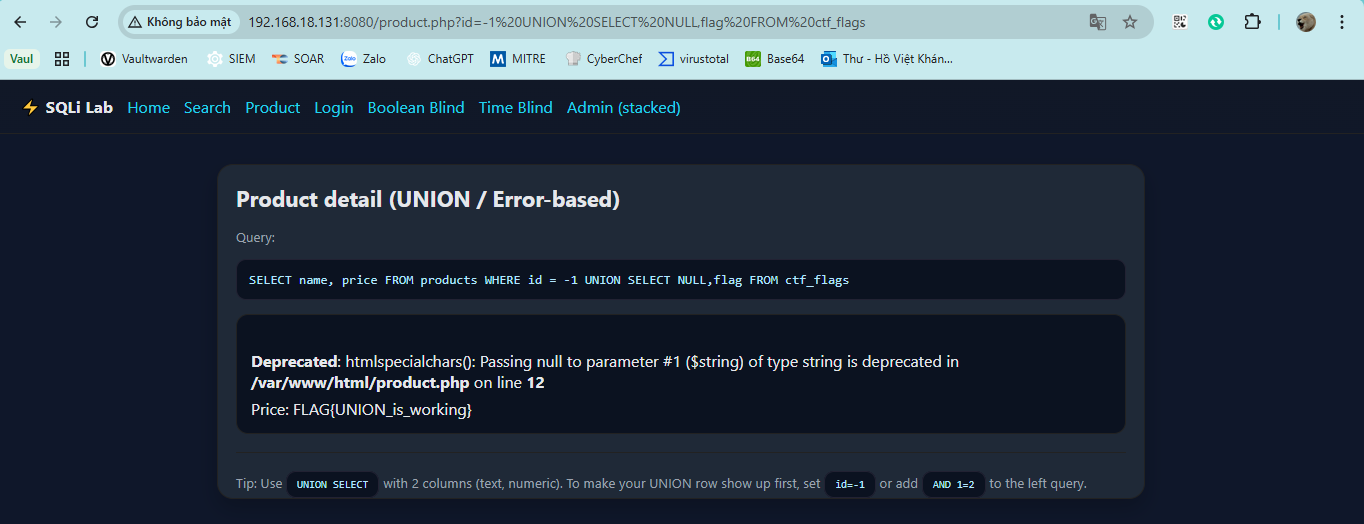


Sau khi thử với lần NULL thứ 2, website trả về lỗi: Deprecated: htmlspecialchars(): Passing null to parameter #1 ($string) of type string is deprecated in /var/www/html/product.php on line 12. Điều này có thể đoán được truy vấn vế đều tiên có SELECT 2 cột

Tiếp theo, chúng ta cần tìm ở trong cột nào sẽ chứa dữ liệu dạng string để trích xuất flag. Ta thử đặt “a” ở phần NULL

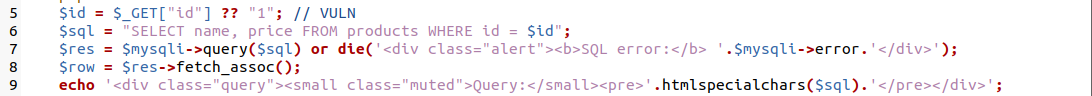


Được rồi, nó đã không bị lỗi nên ở đó có thể chèn string vào. Tiếp theo chúng ta cần vô hiệu hóa kết quả trả về của query ban đầu và trích xuất flag



Thành công.

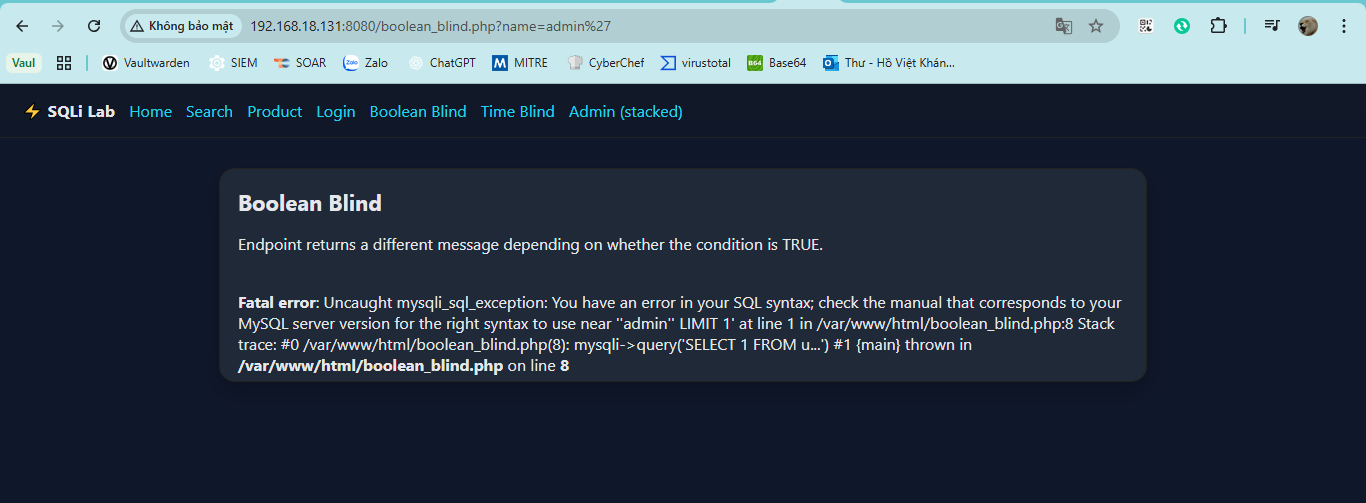
Lỗi ở đây nằm ở việc $id lấy trực tiếp từ $\_GET["id"] rồi chèn thẳng vào SQL



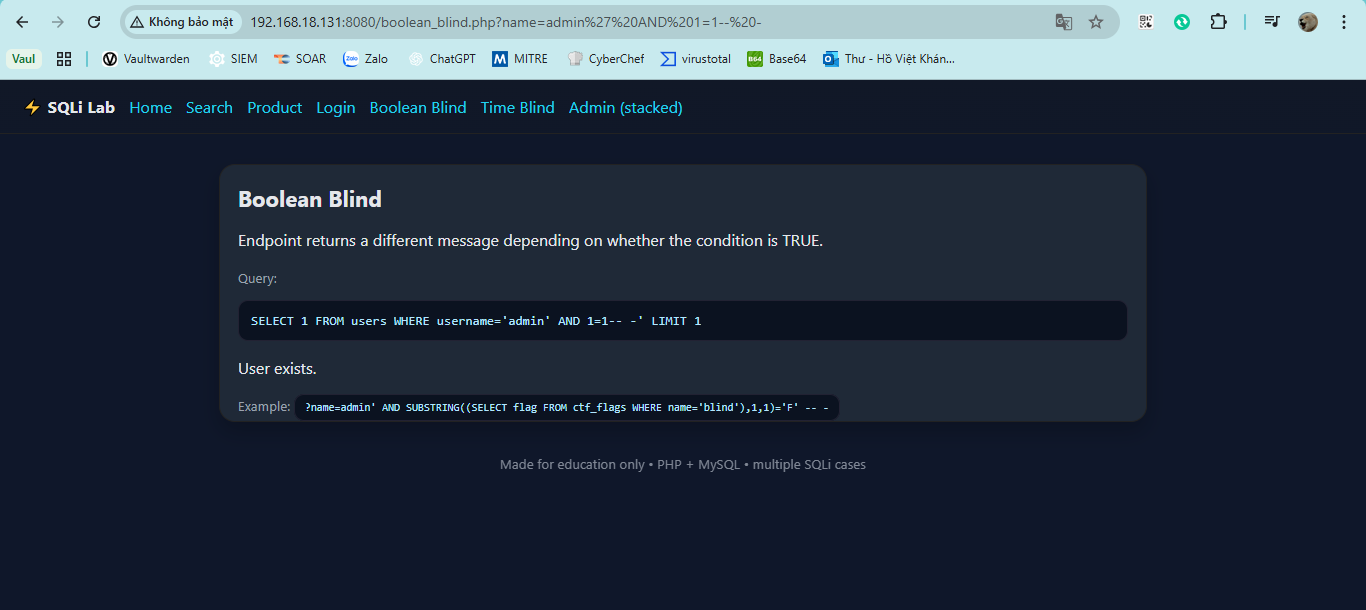
Cách khắc phục: : Dùng prepared statements + bind parameters

Case 3: Blind Boolean

Khi thêm ‘ vào sau admin, website trả về lỗi về systax nên có thể dự đoán nó dính sql injection

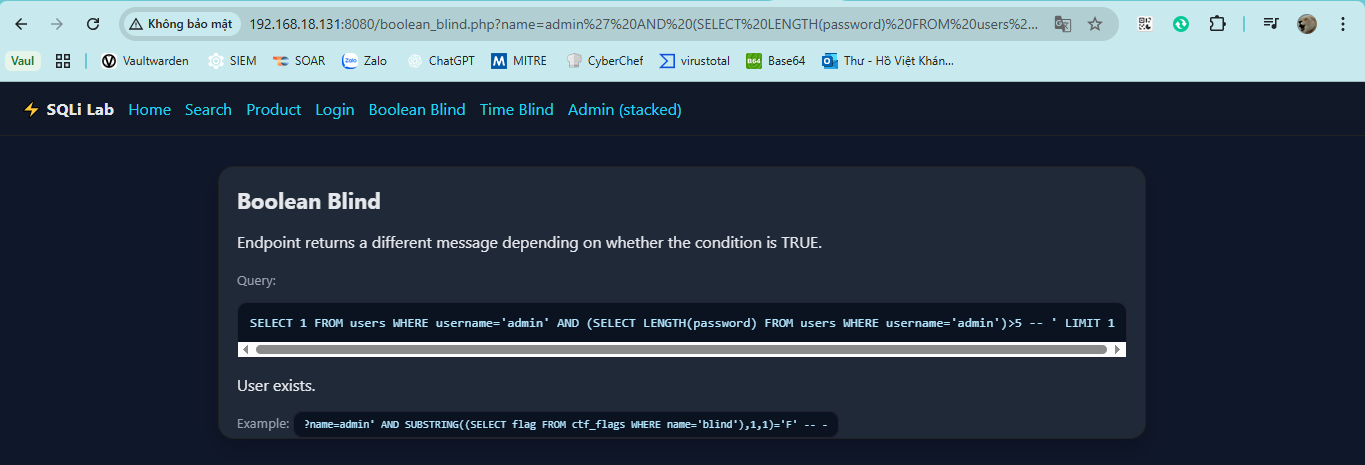


Thử truyền vào %27%20AND%201=1--%20- và %27%20AND%201=2--%20-, ta thấy website phản hồi khác nhau chứng tỏ đã dính lỗi Blind Boolean





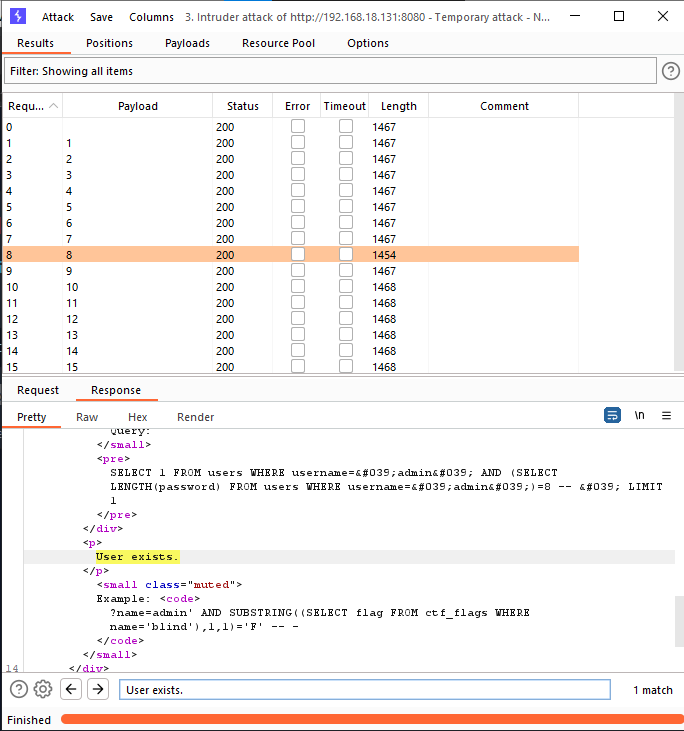
Nếu user tên là admin, chúng ta thử dò độ dài mk admin bằng substr: %27%20AND%20(SELECT%20LENGTH(password)%20FROM%20users%20WHERE%20username=%27admin%27)>5%20--+ mục đích là tìm ra độ dài rồi sau đó dò từng kí tự



Bạn cũng có thể dùng cách tính toán nhị phân để dò nhanh hơn hoặc sử dụng intruder của burp



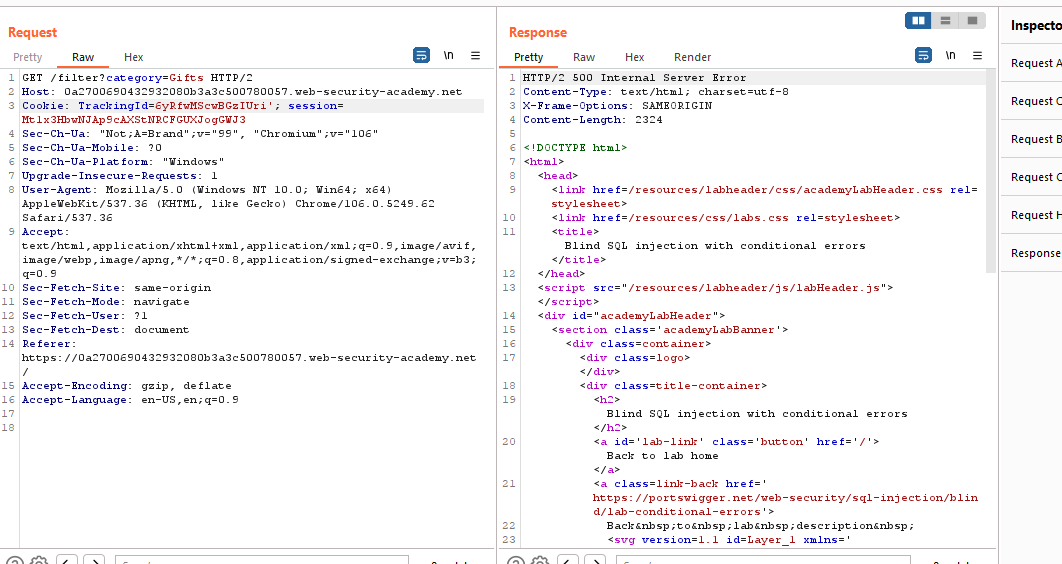
Có thể dò được password chứa 8 kí tự

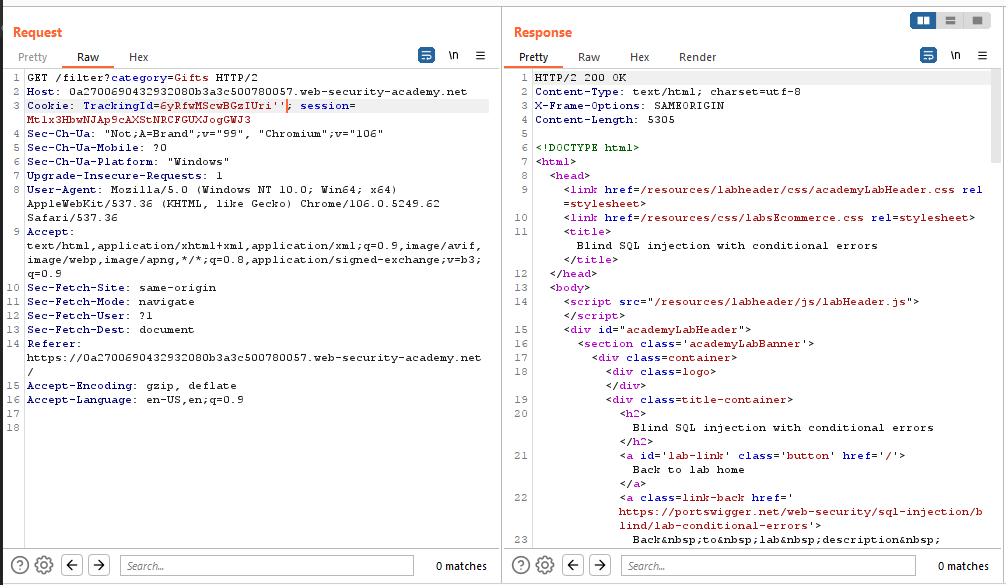


Tương tự có thể sử dụng để dò password

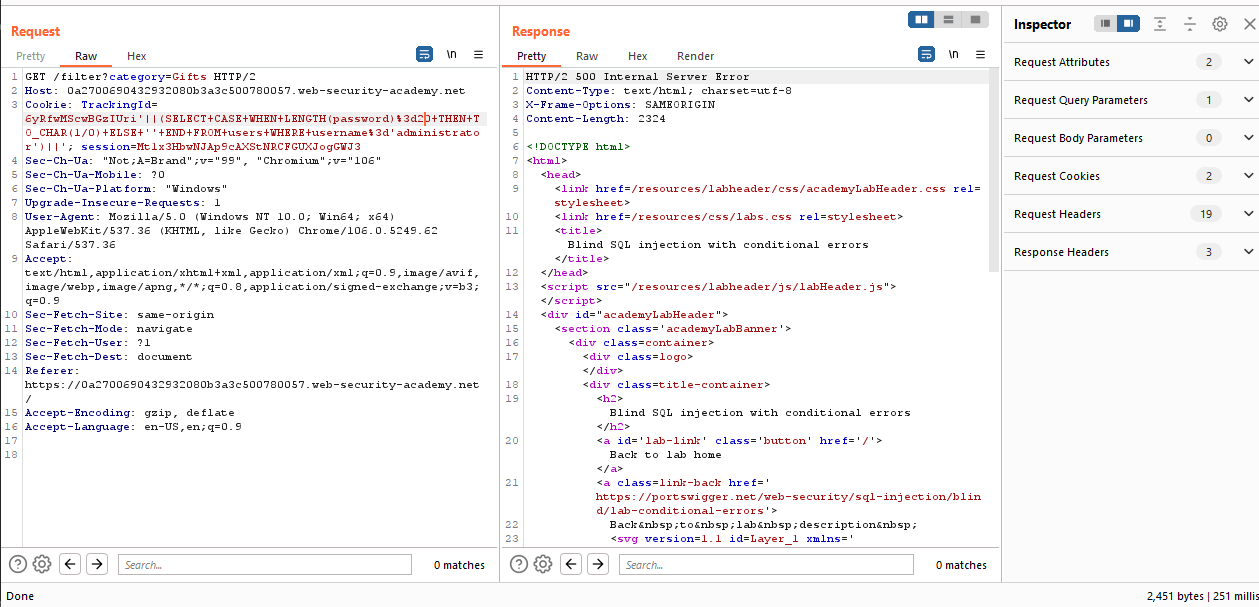
Case 4: Blind SQL injection with conditional errors

Ở lab này, lỗi nằm ở phần TrackingId, thử ‘ thấy server trả về 500



Khi tiếp tục thử ‘’, server lại trả về 200, chứng tỏ nó đã dính lỗi Blind SQL injection with conditional errors

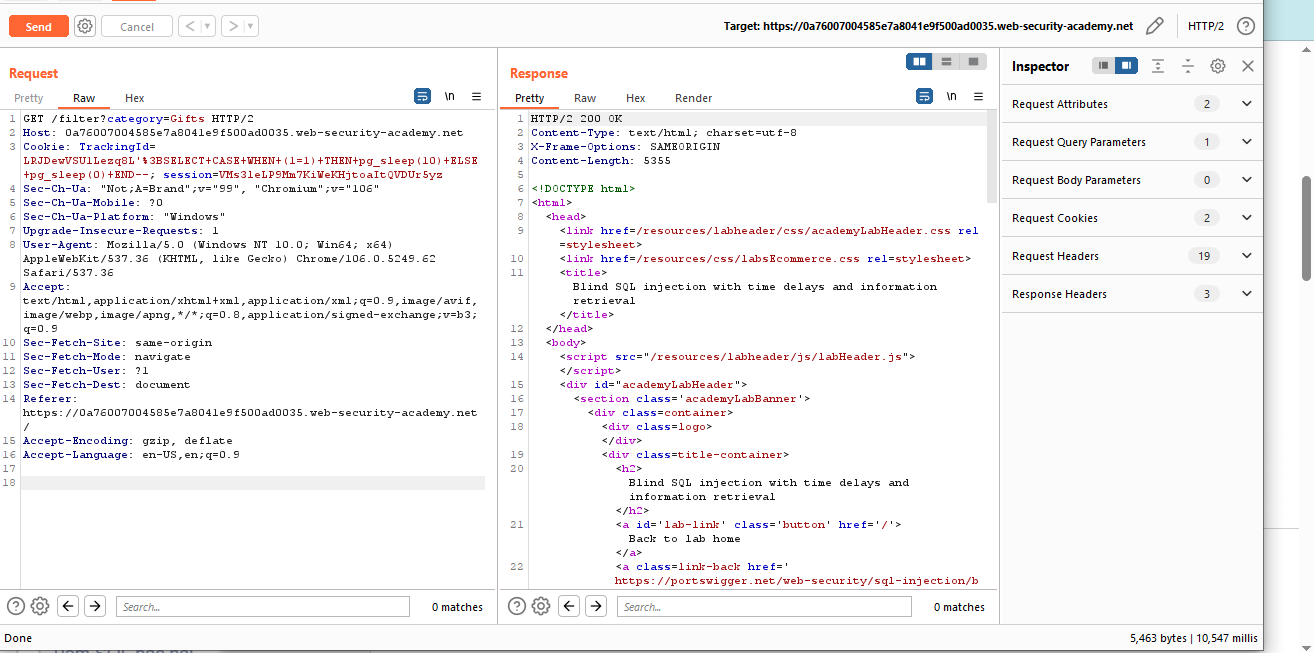
Tiếp theo chúng ta sẽ thử dò độ dài password administrator bằng '||(SELECT CASE WHEN LENGTH(password)>3 THEN TO\_CHAR(1/0) ELSE '' END FROM users WHERE username='administrator')||' ( dành cho oracle). Ở đây password có độ dài là 20 kí tự.



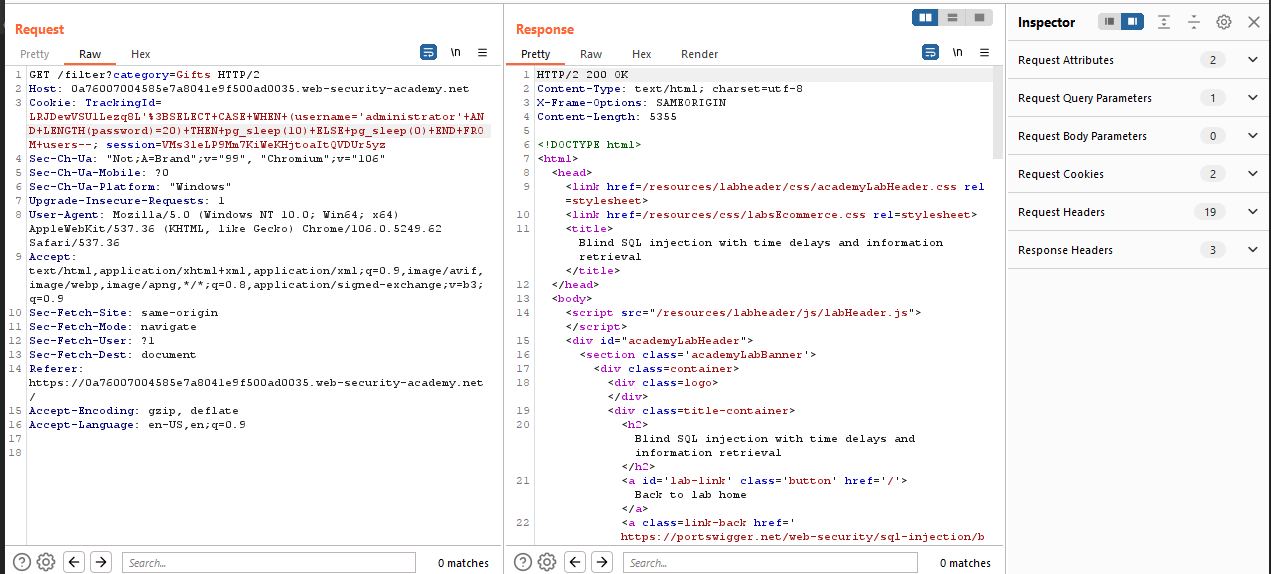
Tương tự với dò giá trị password.

Case 5: Blind SQL injection with time delays

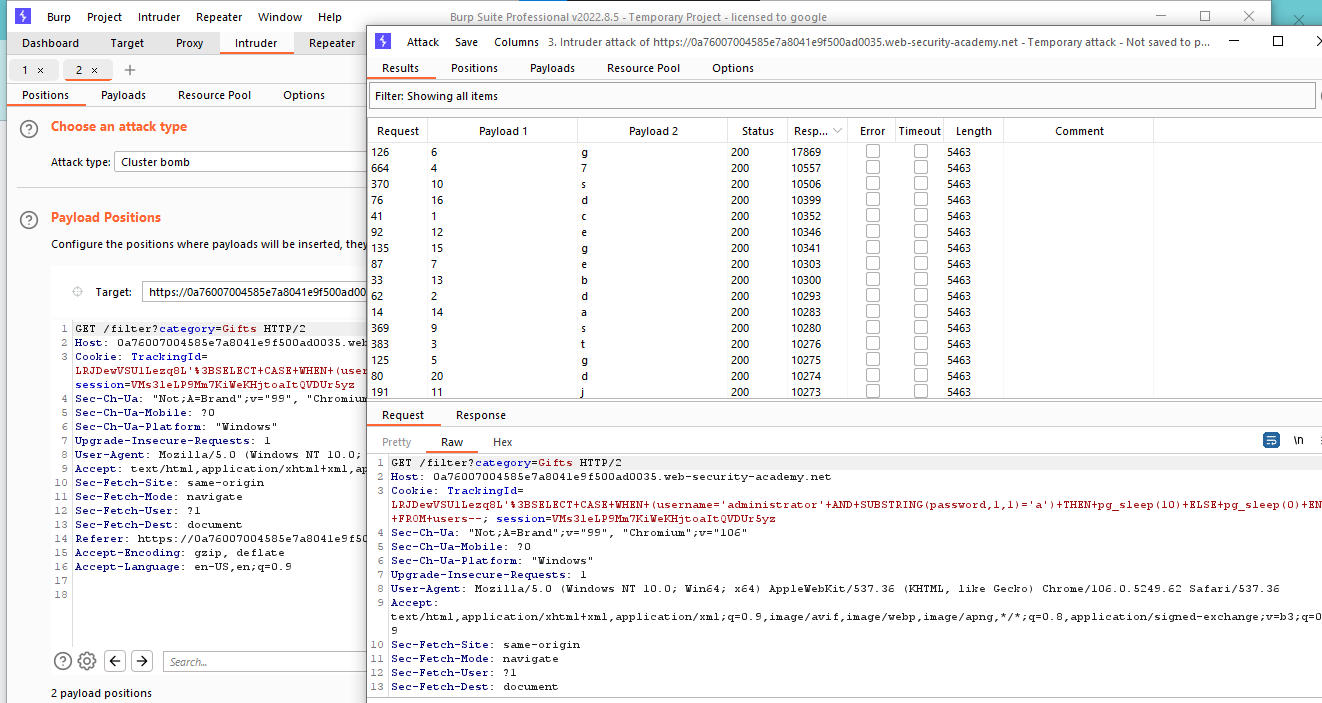
Thử '%3BSELECT+CASE+WHEN+(1=1)+THEN+pg\_sleep(10)+ELSE+pg\_sleep(0)+END— thấy ứng dụng mất 10s để phản hồi, chứng tỏ nó đã dính SQL injection



Tiếp tục theo luồng sử dụng Intruder để scan password



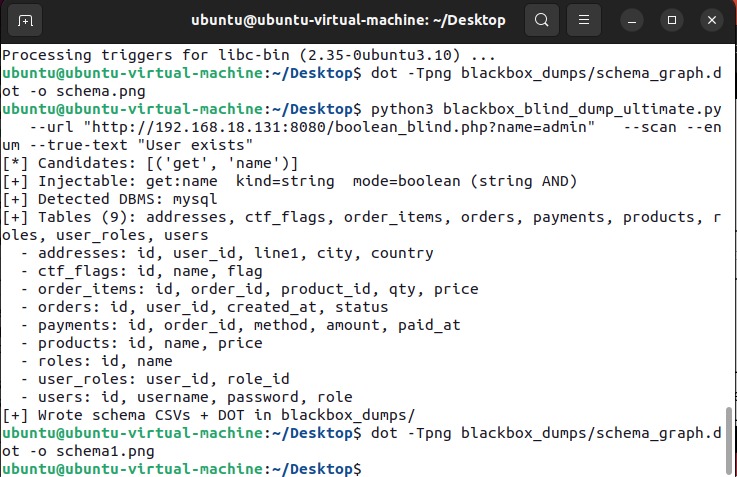
Xác minh rằng password có 20 ký tự, tiếp tục scan password bằng Intruder



Dễ dàng thu được password

# Dump sll record và vẽ lại sơ đồ quan hệ DB

Ở đây e đã xây dựng lại 1 trang web dính lỗ hổng sql injection và script để dump các bảng của trang web, sau đó lưu lại vào file csv



Ngoài ra, nó còn có thể xây dựng lại lược đồ quan hệ của các bảng để mình có thể nắm được CSDL của khách hang

