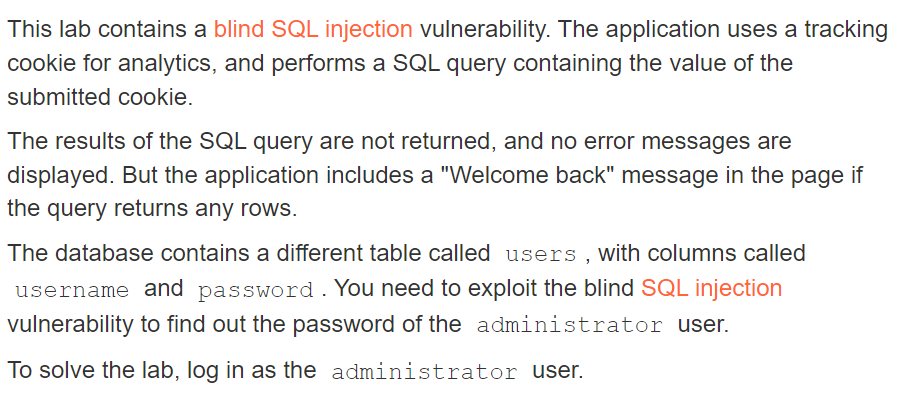
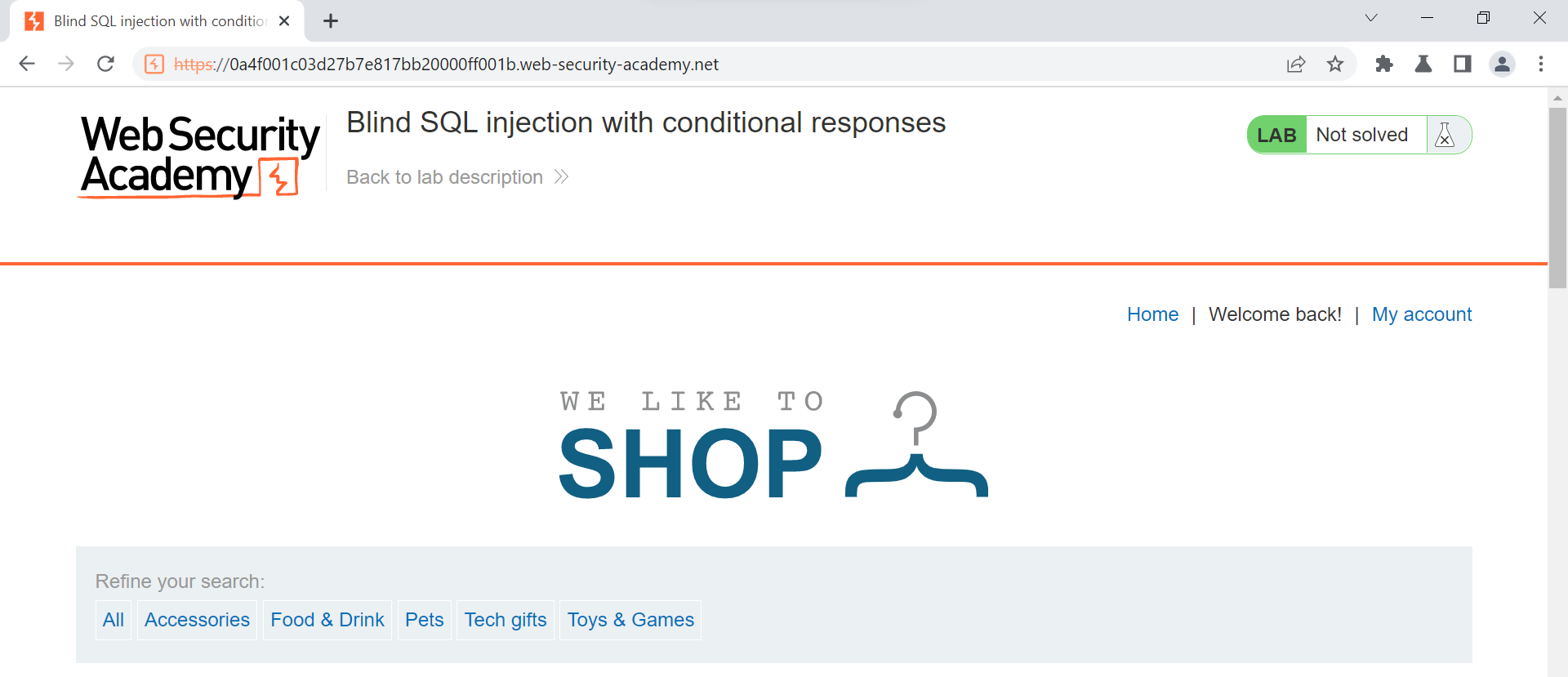
Lab: Blind SQL injection with conditional responses

<https://portswigger.net/web-security/sql-injection/blind/lab-conditional-responses>

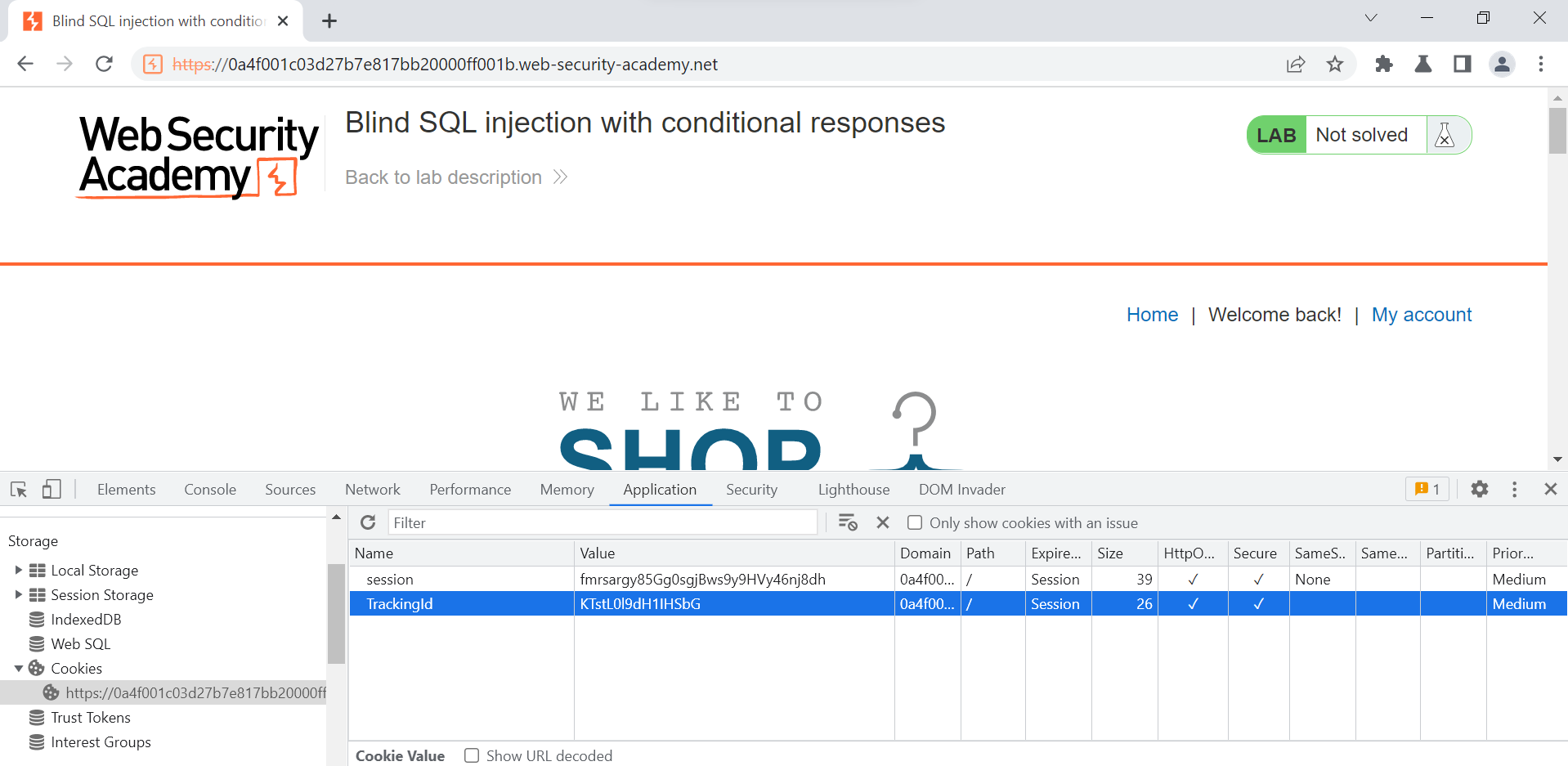


Đọc phần mô tả thì mình thấy là trang web dính lỗi **blind SQL**, sử dụng **tracking cookie** để phân tích, thực hiện truy vấn SQL và web sẽ hiển thị chuỗi “**Welcome back**” nếu có truy vấn được trả về. Cuối cùng là ta cần tìm ra **password** của **administrator** để solve được lab. Như vậy, ta cần tập trung khai thác vào **cookie**.

Trước tiên, mở Burp Suite lên rồi mở web lên dạo một vòng, kích vào các chức năng xem thử thì khi vô tình mình thấy chuỗi “Welcome back” khi kích vào nút Home.

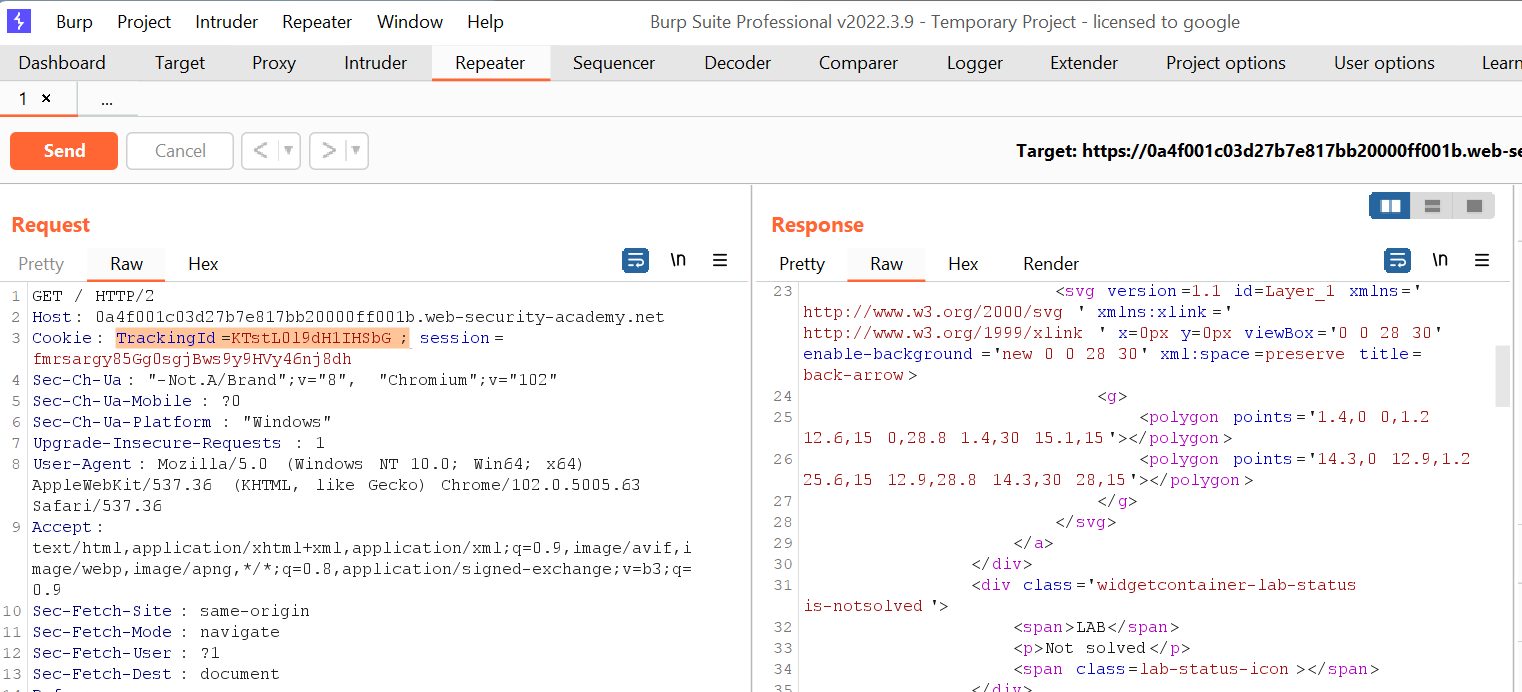


Chứng tỏ có gì đó được truy vấn và trả về nên có chuỗi “Welcome back” xuất hiện. Vì bài lab gợi ý liên quan đến cookie nên mình mở cookie lên xem thì thấy có một trường **TrackingId** trong cookie



Thử xóa đi rồi tải lại trang web thì chữ “Welcome back” không xuất hiện nữa. Nhấn lại một nút khác thì “Welcome back” lại xuất hiện 🡺 điều này chứng tỏ là lúc vừa mở trang web lên mình sẽ có 1 **TrackingId**, mỗi lần load lại trang thì sẽ có truy vấn kiểm tra xem cookie có tồn tại hay không? Nếu tồn tại thì sẽ hiển thị “Welcome back” và nếu không thì không hiển thị 🡺 như vậy mình thấy kết quả trả về có sự khác nhau giữa True và False nên mình sẽ dựa vào đây để khai thác tiếp.

Bây giờ chuyển gói tin sang Repeater để dễ thao tác, mình có biến TrackingId như hình:



Có thể tưởng tượng câu truy vấn SQL như sau: **SELECT <cái gì đó> FROM <bảng nào đó> WHERE TrackingId = ‘KTstL0l9dH1IHSbG’**

Bây giờ để đăng nhập bằng user là **administrator** thì mình cần phải có **password**.

Ở phần mô tả mình đã biết có bảng **users** với 2 cột là **username** và **password** rồi nên bây giờ chỉ việc dò xem **password** có bao nhiêu ký tự rồi sau đó sẽ dò từng ký tự là gì (vì đây là **blind SQL** nên không thể thấy được kết quả trả về mà chỉ có thể dò đoán bằng **True or False**). Ok bắt tay vào thực hành nào!

Ta có một câu truy vấn luôn đúng là: **(SELECT username FROM users WHERE username = 'administrator') = ‘administrator’**

Để dò xem password có bao nhiêu ký tự mình sẽ dùng hàm **LENGTH ()** kết hợp với **AND** để tạo câu truy vấn trả về **True or False** thì mình sẽ có câu truy vấn hoàn chỉnh như sau:

**' AND (SELECT username FROM users WHERE username = 'administrator' AND LENGTH(password) > 1) = 'administrator**

Copy câu này Reepeater test thử đã (nhớ bôi đen ctrl+U để mã hóa nha)

Text

Description automatically generated

Ok vậy là payload hoạt động tốt, có dòng Welcome back nên mình biết kết quả là True. Giờ để nhanh hơn thì mình chuyển gói tin qua Intruder, sửa điều kiện so sánh thành biến bằng cách **Add $** rồi thêm payload là các số từ 1 đến 100 như hình (có chạy trong khoảng bao nhiêu tùy thích, miễn sao thấy được sự khác biệt khi nó True or False là được ^^):

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Sau khi Start Attack thì mình thấy từ số 20 trở đi Length thay đổi, mình vào xem Response thì thấy mất dòng “Welcome back” rồi.

Graphical user interface, text, application

Description automatically generated

Vậy password của mình có độ dài là 20.

Tương tự vậy mình sẽ thay đổi câu truy vấn và dùng **SUBSTRING()** để tìm từng kí tự của password: **' AND (SELECT SUBSTRING(password,1,1) FROM users WHERE username = 'administrator') = 'a**

Mình sẽ lần lượt thay vị trí từ 1 đến 20 và dùng payload từ a đến z, từ 0 đến 9 tìm ra từng kí tự của password. Mình sử dụng Intruder để bruteforce như hình:

Graphical user interface, application

Description automatically generated

Vì sử dụng 2 payload khác nhau nên mình sử dụng kiểu attack là **Cluster bomb**



Graphical user interface, text, email

Description automatically generatedGraphical user interface

Description automatically generated

Sau khi nhận được kết quả attack, mình biết nếu câu truy vấn đúng thì “Welcome back” sẽ xuất hiện nên mình dùng Filter để lọc ra các gói tin có trả về dòng chữ này.

Graphical user interface, table

Description automatically generated

Sắp xếp lại các payload 1 theo thứ tự từ 1 đến 20 và ghép các kí tự ở payload 2 lại thì mình được password có nội dung: 7a0w6vp16qlqex2qisz9

Bây giờ chỉ việc mang password đi đăng nhập thôi.

Graphical user interface, application

Description automatically generated

Yep, đăng nhập thành công!!!