BÁO CÁO BÀI THỰC HÀNH SỐ 8

KIỂM THỬ LỖ HỔNG SQL INJECTION

Link Youtube: https://youtu.be/Dmi-2DTy1CM

Họ và tên: Đỗ Nhật Nam

1.4.1. Cài đặt và cấu hình Virtualbox

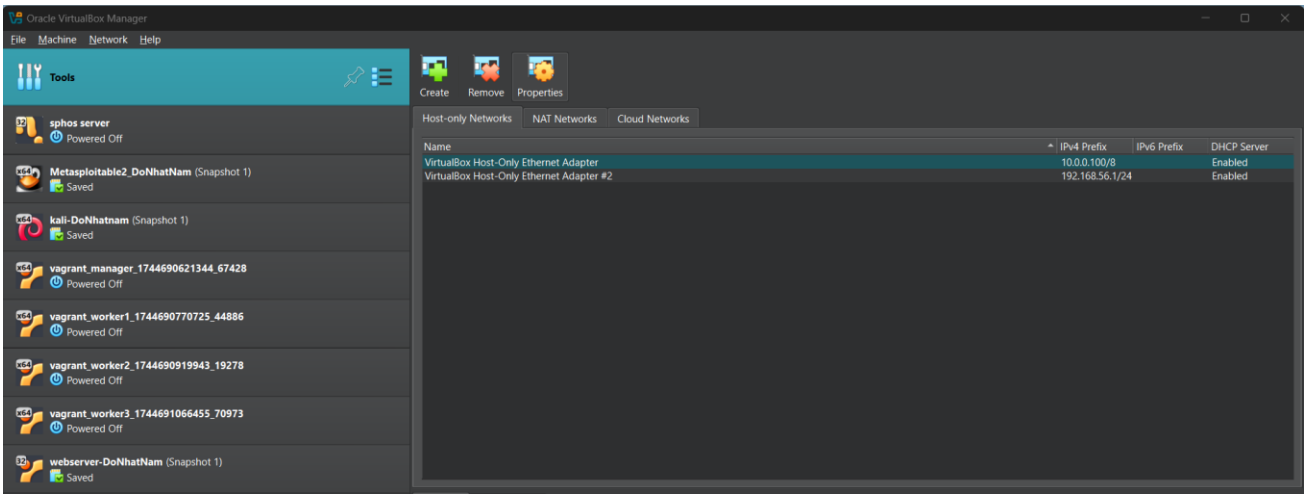
Bước 1: Download phần mềm Virtualbox tại địa chỉ sau và cài đặt như một phần mềm thông thường trên Windows:

Bước 2: Download gói mở rộng cho Virtualbox từ địa chỉ sau:

Bước 3: Khởi động phần mềm Virtualbox

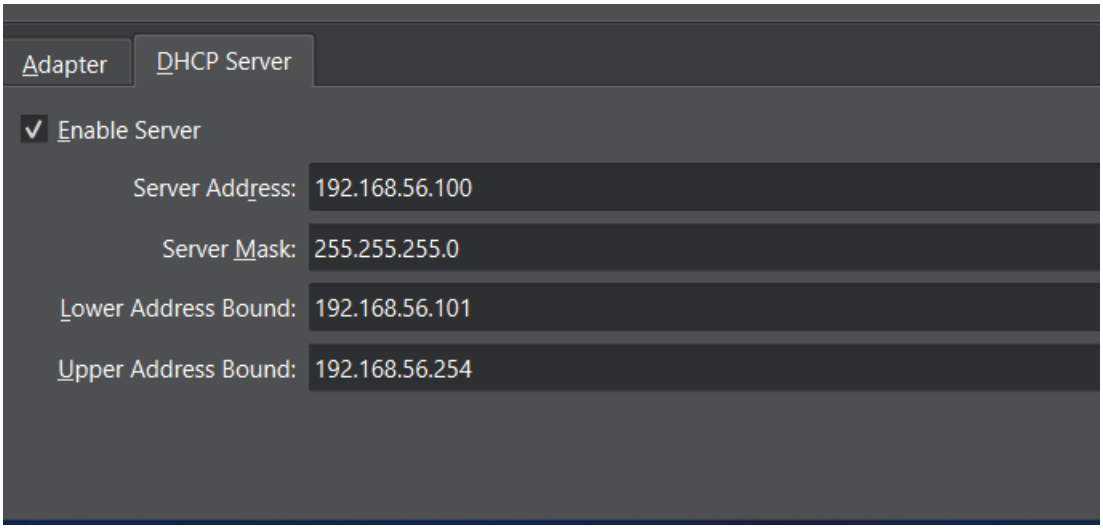
Bước 4: Trên giao diện của Virtualbox, chọn File → Host Network Manager…

Bước 5: Chọn cạc mạng ảo VirtualBox Host-Only Ethernet Adapter. Chọn



Bước 6: Chọn thẻ Adapter và lựa chọn Configure Adapter Automatically

Bước 7: Chọn thẻ DHCP Server và thiết lập các thông số như hình ảnh sau

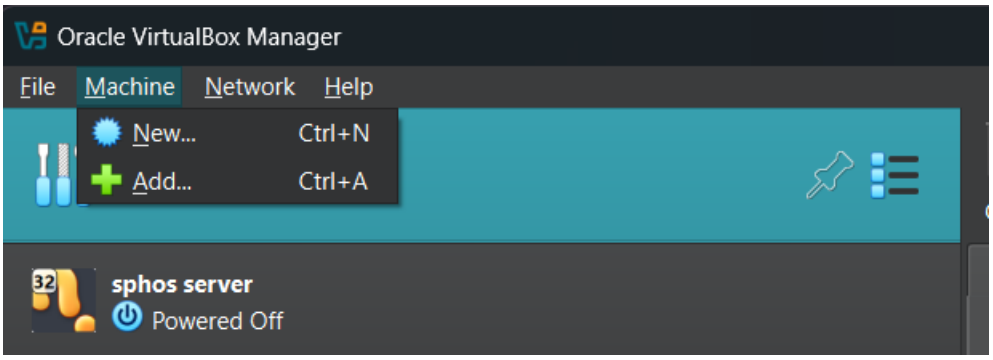


1.4.2. Triển khai máy ảo Web Server

Bước 1: Download máy ảo từ địa chỉ sau và giải nén

https://drive.google.com/file/d/1LSK_CZoha8LIKqr-8KkfyztZ6MM-eEF2/view?usp=sharing

Bước 2: Trên cửa sổ chính của Virtualbox, chọn Machine → New…



Bước 3: Trên cửa sổ tạo máy ảo, đặt các thông số như sau. Sau đó nhấn Next.

Name: Tên máy ảo ➢ Machine Folder: Thư mục chứa máy ảo ➢ Type: Linux ➢ Version: Ubuntu (32-bit)

Bước 4: Chọn dung lượng bộ nhớ RAM cho máy ảo là 2048 MB. Nhấn Next để tiếp tục

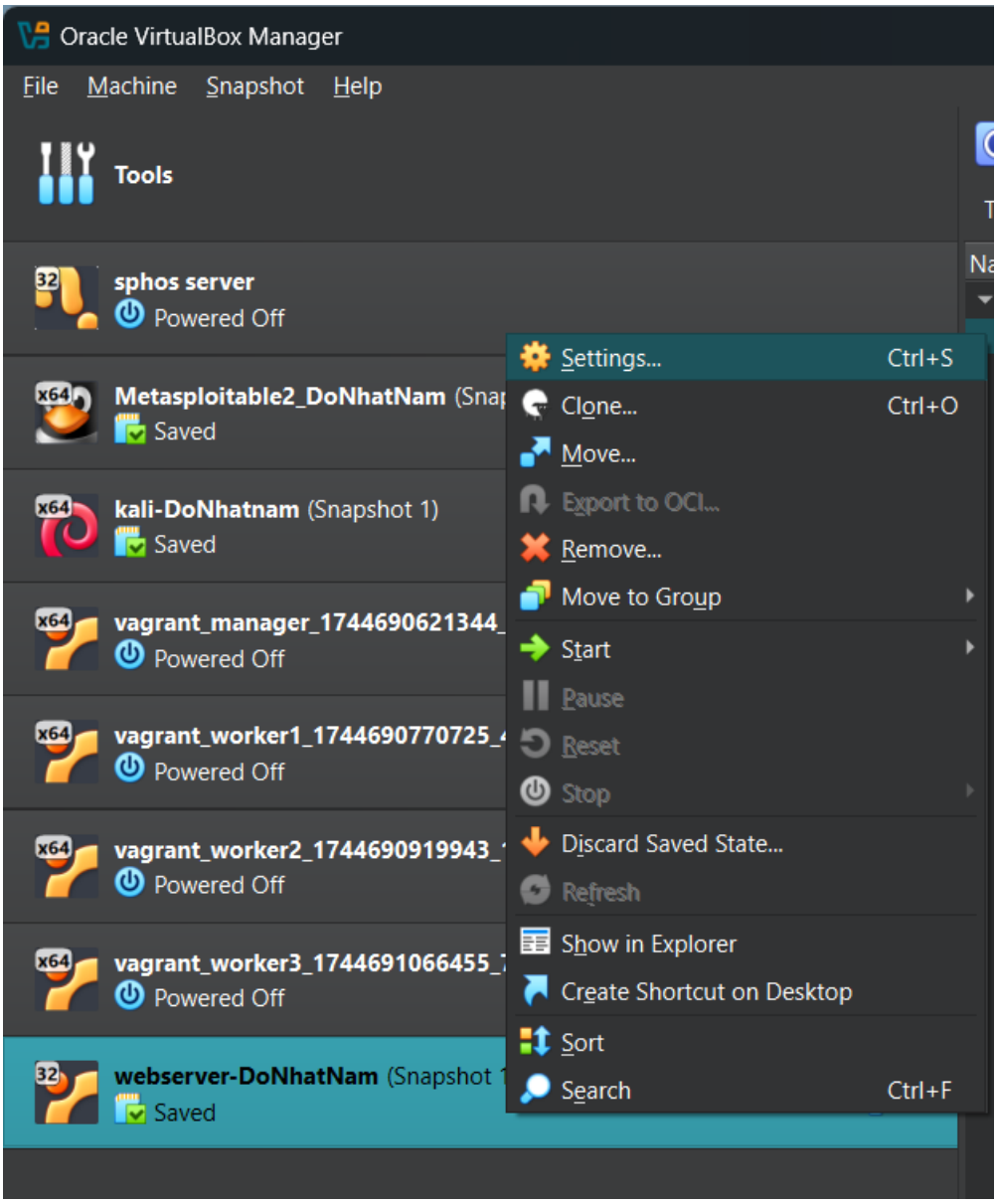
Bước 5: Trong cửa sổ Hard disk tạo ổ cứng máy ảo, chọn mục Use an existing virtual hard disk file. Sau đó bấm nút Choose a virtual hard disk file…

Bước 6: Trên cửa sổ Hard Disk Selector, nhấn Add và chọn file Server.vdi đã download ở bước 1 để thêm vào danh sách

Bước 7: Chọn file Server.vdi vừa được thêm vào trong danh sách ổ cứng ảo. Nhấn Choose để lựa chọn và đóng cửa sổ.

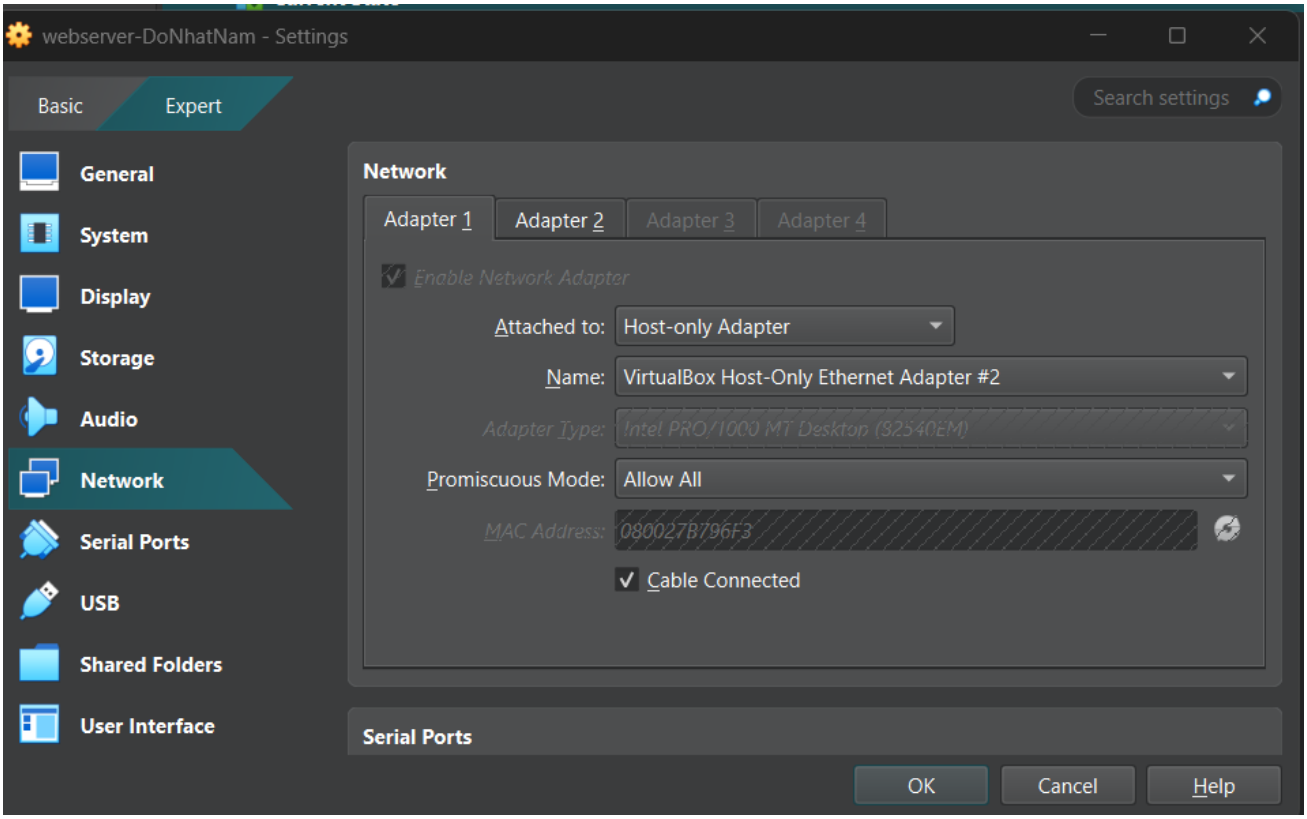
Bước 8: Trên cửa sổ Hard disk, nhấn Create để tạo ổ cứng ảo.

Bước 9: Trên cửa sổ chính của Virtualbox, chọn máy ảo vừa tạo và nhấp chuột phải. Chọn Settings…



Bước 10: Chọn Network → Adapter 1.

Thiết lập các thông số như sau: ➢ Attached to: Host-only Adapter ➢ Name: VirtualBox Host-Only Ethernet Adapter (hoặc còn gọi là VirtualBox Host-Only Network)

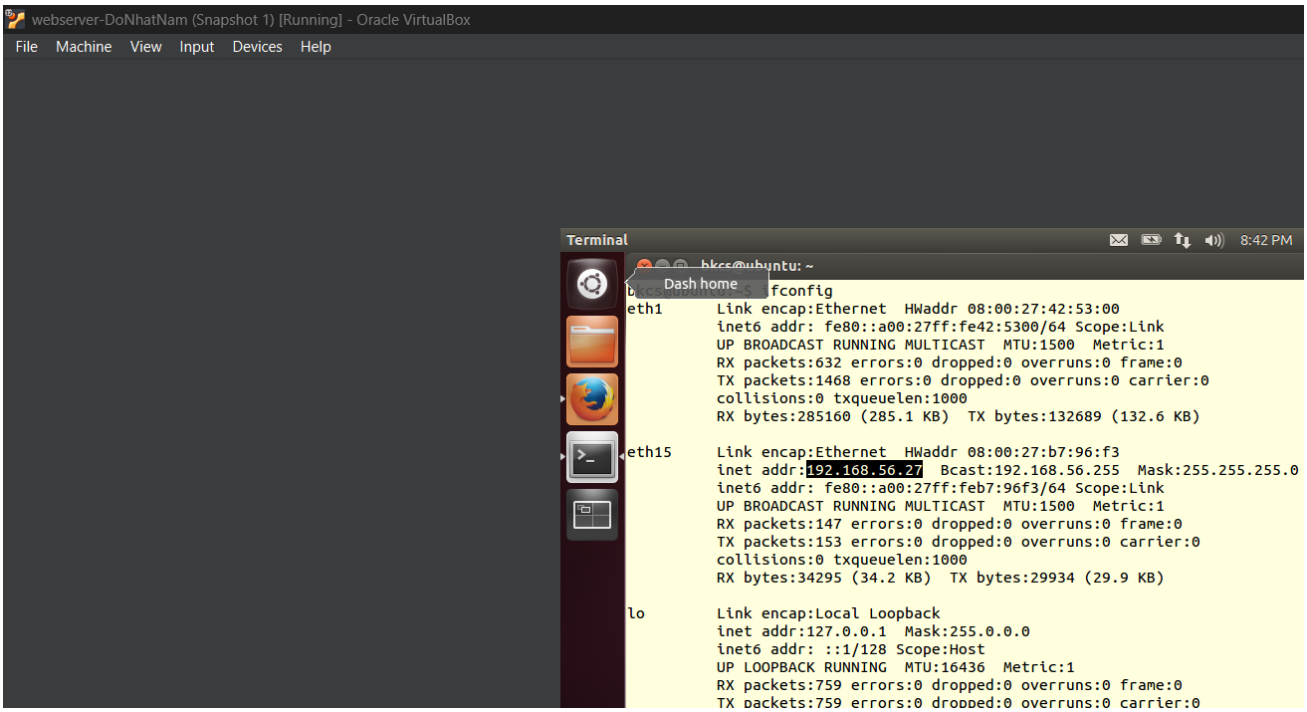


Sau khi máy ảo khởi động xong, đăng nhập bằng tài khoản sau:

➢ Username: bkcs

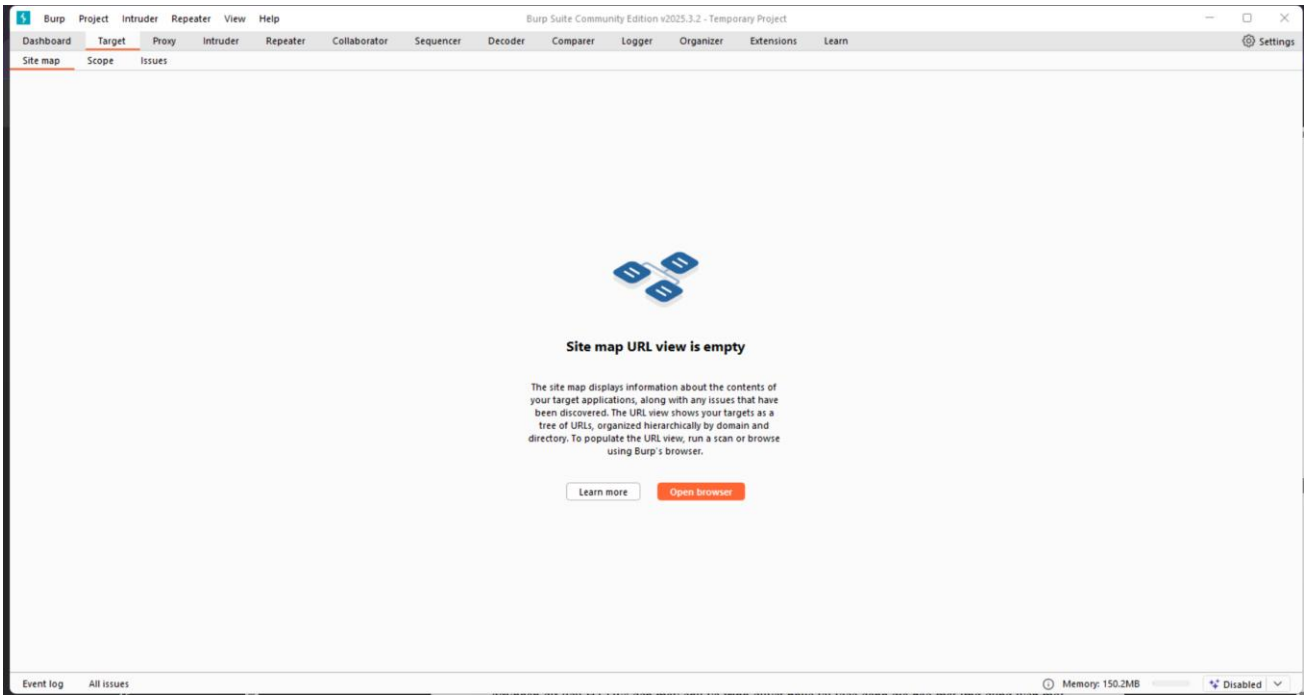
➢ Password: bkcs

Khi thực hiện các nội dung luyện tập trong tài liệu hướng dẫn bài thực hành số 5 và số 6, địa chỉ localhost khi truy cập vào các trang Web được thay thế bằng địa chỉ IP của Web Server. Trên Web Server mở cửa sổ Terminal và thực hiện lệnh ifconfig. Trong hình ảnh minh họa sau, địa chỉ của Web Server là 192.168.56.27



4.1. Hướng dẫn cấu hình cơ bản

Người dùng Windows có thể thực thi chương trình bằng cách mở tập tin burpsuite_free_vxx.jar sau khi đã download và bảo đảm rằng Java Runtime đã được cài đặt trên máy tính.

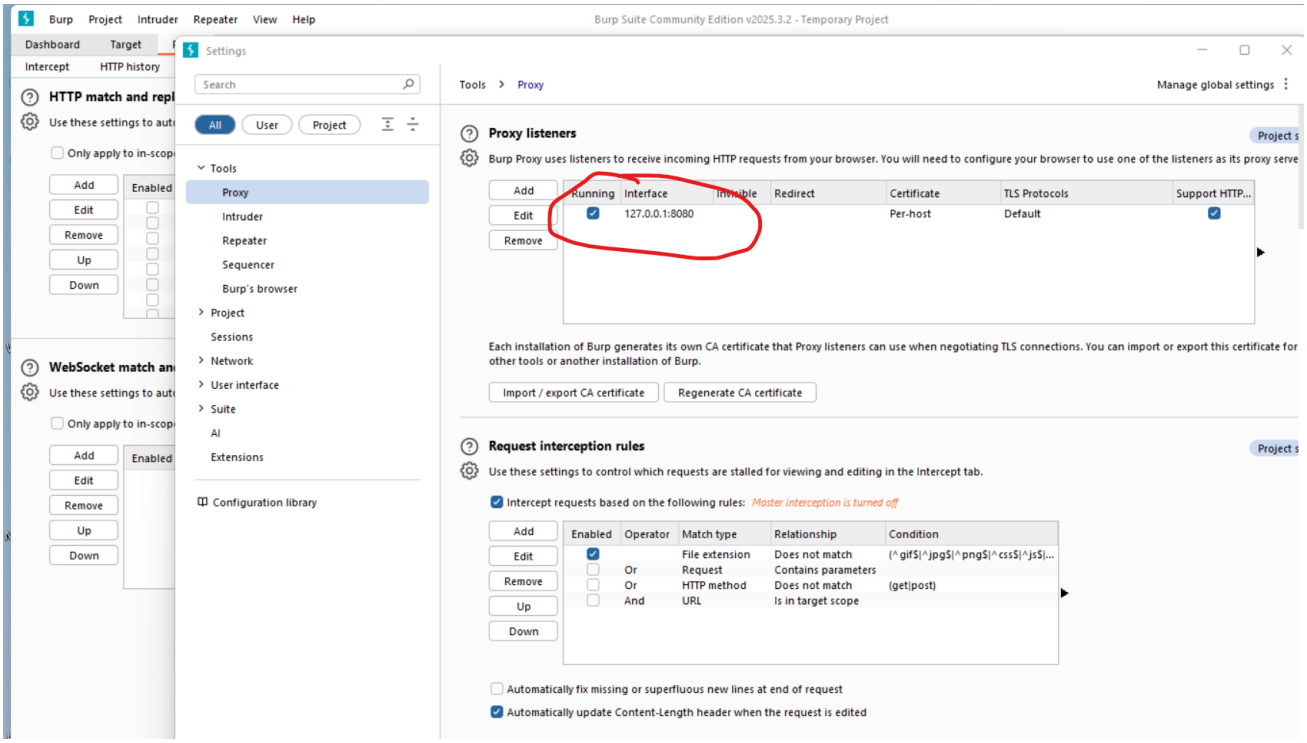


Kiểm tra hoạt động Burp

Burp Proxy đóng vai trò là chương trình trung chuyển các HTTP Request/Respone giữa trình duyệt và ứng dụng web, gọi là Intercepting Proxy. Burp cho phép người dùng toàn quyền điều khiển việc gởi/nhận dữ liệu HTTP/s đến máy chủ và trình duyệt phục vụ việc đánh giá bảo mật ứng dụng web một cách cụ thể cho từng lỗ hổng bảo mật.

Cấu hình tại Burp Proxy

Theo mặc định, Burp Proxy được cấu hình lắng nghe trên cổng 8080/TCP. Để kiểm tra chắc chắn rằng không có chương trình hoặc dịch vụ nào khác đang lắng nghe trên cùng cổng 8080/TCP, bạn thực hiện kiểm tra tại thẻ Proxy | Options



4.2. Sử dụng Burpsuite hỗ trợ kiểm thử SQL Injection

Phần này sẽ minh họa việc sử dụng Burpsuite để tương tác với website khi kiểm thử SQL Injection

Bước 1: Cấu hình và khởi động Burpsuite như hướng dẫn ở phần trước

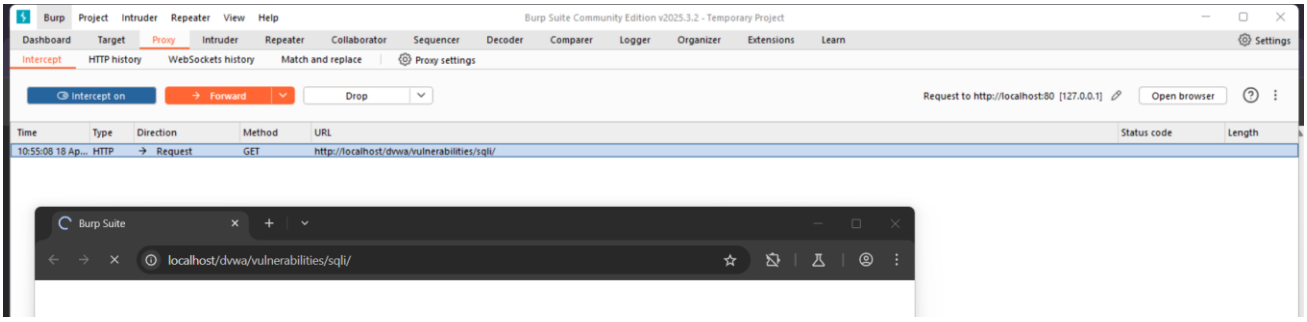
Bước 2: Trên cửa sổ công cụ Burpsuite, mở thẻ Proxy → Intercept và chắc chắn tính năng Intercept is on đã được bật

Bước 3: Mở địa chỉ trang Web cần kiểm thử. Ví dụ dưới đây là giao diện kiểm tra lỗ hổng SQL Injecion trên website DVWA.

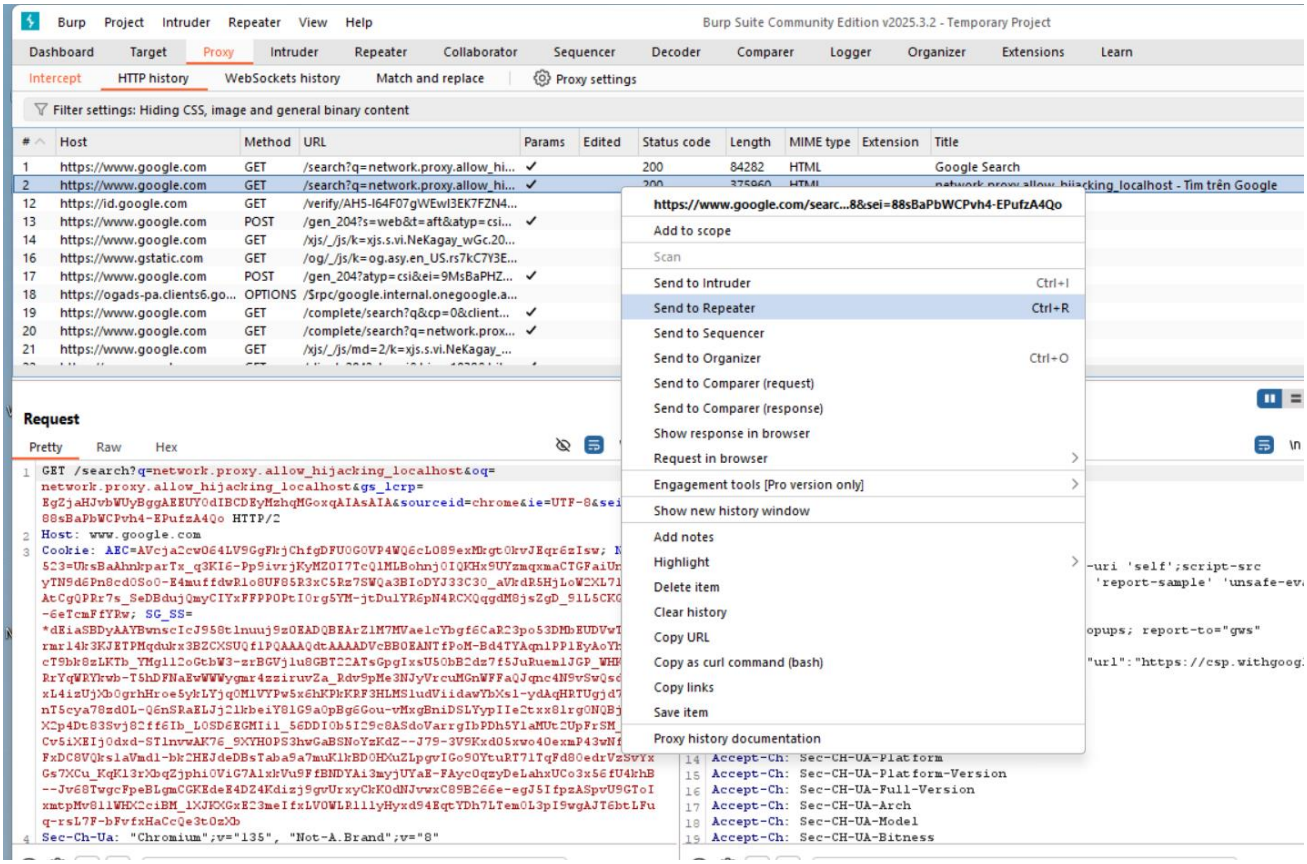
http://localhost/dvwa/vulnerabilities/sqli/

Bước 4: Điền các giá trị bất kỳ nào đó và gửi yêu cầu từ trình duyệt

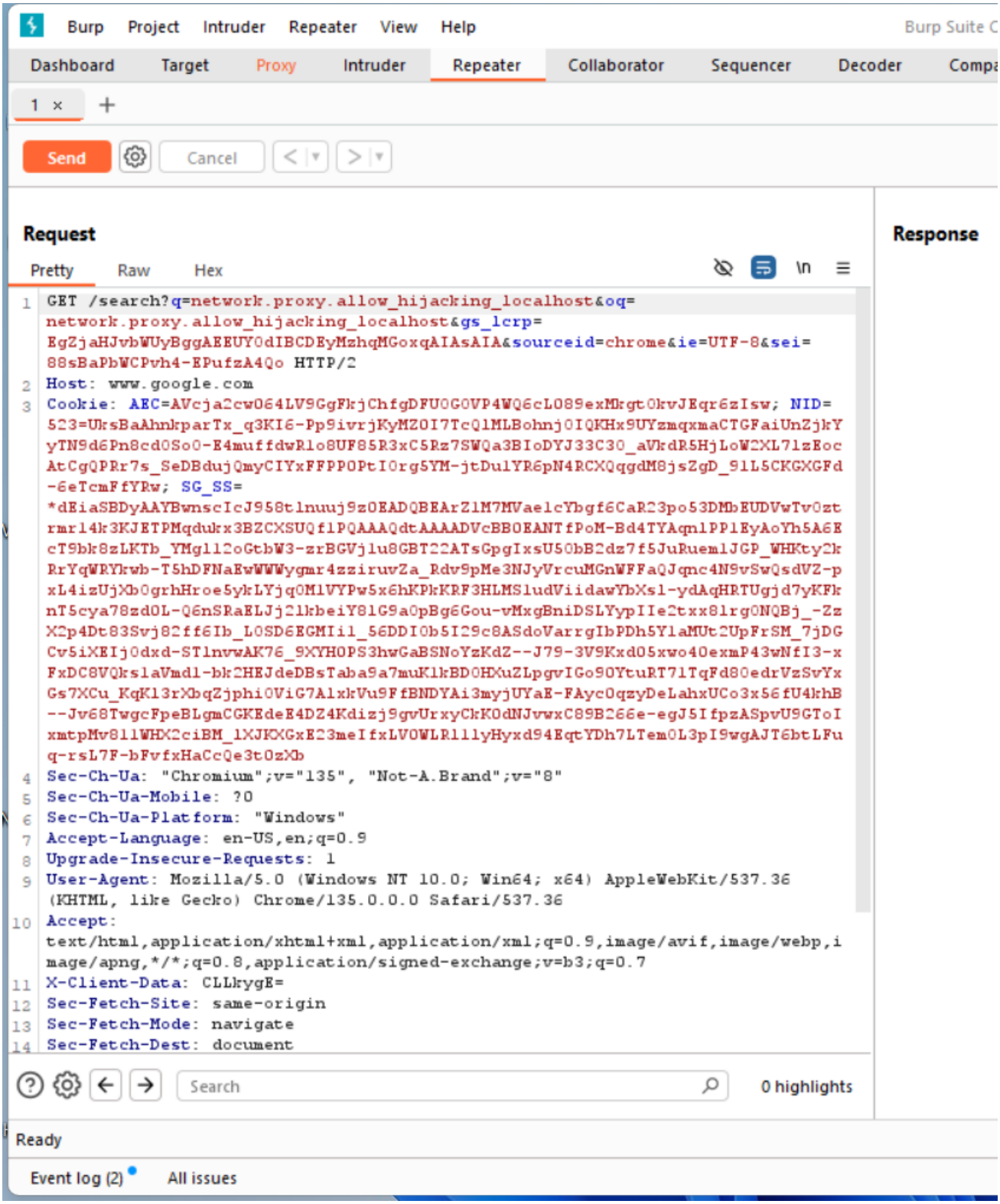
Bước 5: Trên thẻ Proxy → Intercept, nhấn nút Forward để chuyển tiếp yêu cầu



Bước 6: Mở thẻ Proxy → HTTP history, chúng ta sẽ thấy danh sách các thông điệp HTTP mà Burpsuite đã bắt được. Chọn thông điệp HTTP Request tương ứng ở bước 4, nhấn chuột phải và chọn Send to Repeater



Bước 7: Chọn thẻ Repeater. Thẻ này cho phép chúng ta thay đổi nội dung của HTTP Request và phát lại tới máy chủ. Thẻ con Params liệt kê danh sách các giá trị trên HTTP Header có thể là tham số đầu vào.



(Giá trị các tham số cũng có thể sửa trực tiếp từ thẻ Params mà không cần qua encode)

Bước 8: Để thực hiện kiểm thử cho tham số đầu vào id chúng ta sẽ sửa trực tiếp trên thẻ con Raw. Chọn thẻ Decoder, điền chuỗi 1' or 1;# và chọn Encode as…→ URL. Kết quả encode cho chúng ta xâu %31%27%20%6f%72%20%31%3b%23. Chọn lại thẻ Repeater, thay xâu giá trị này vào cho tham số id và nhấn nút Go. Thông điệp HTTP Response trả về từ server được hiển thị theo nhiều dạng khác nhau trong phần Response.

5. Minh họa cách thức kiểm thử hộp đen

5.1. Cài đặt môi trường

- Bước 1: Download mã nguồn và cơ sở dữ liệu từ địa chỉ sau

https://users.soict.hust.edu.vn/tungbt/it4263/lab05_tut.zip

Bước 2: Giải nén file download. Sử dụng WinSCP hoặc công cụ tương tự để upload thư mục lab05_tut vừa giải nén được vào thư mục /home/bkcs của máy ảo

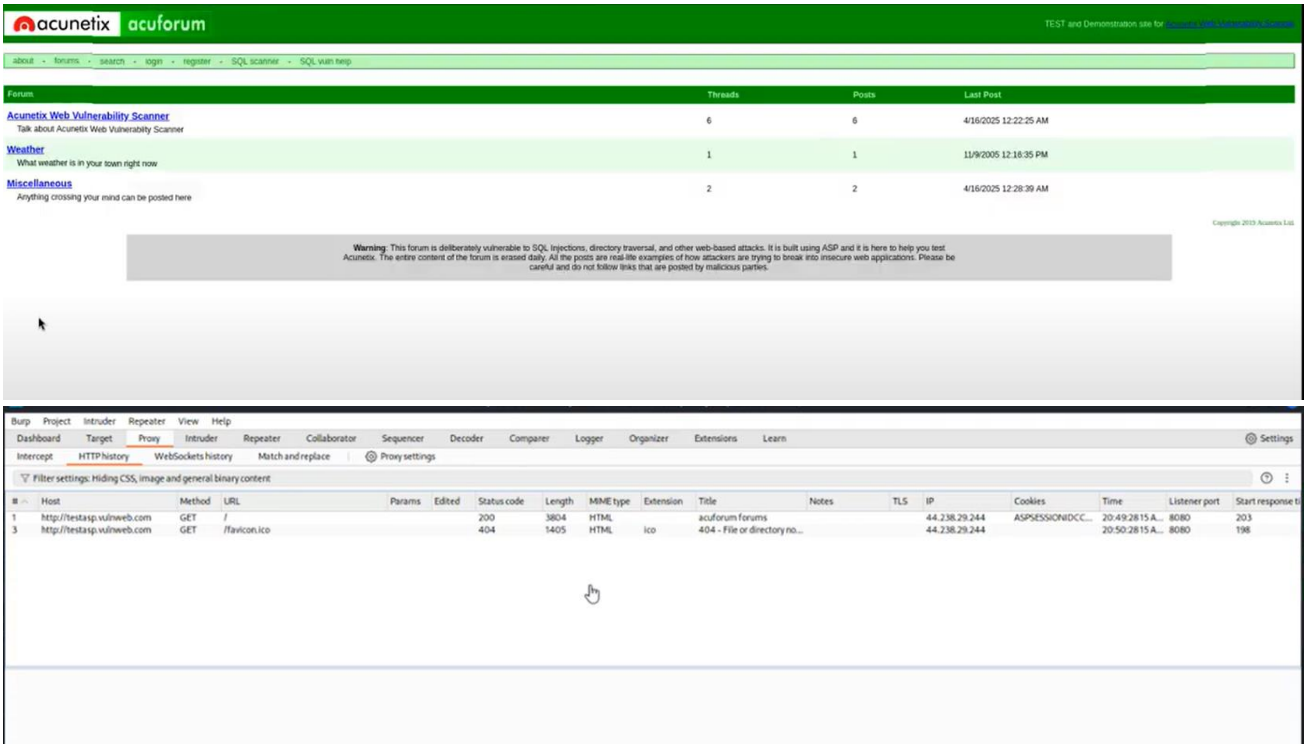
Bước 3: Khởi động và truy cập vào máy ảo.

Bước 4: Trên cửa sổ dòng lệnh, chuyển thư mục làm việc

5.2 Luyện tập

Bước 1: Truy cập địa chỉ http:///sqli/sqli1.html

Bước 2: Lựa chọn một liên kết bất kỳ để truy cập



- Bước 3: Trang kết quả hiển thị cho thấy thông tin của một người dùng. Từ thanh địa chỉ có thể quan sát thấy có một tham số truyền vào theo phương thức GET là cat với giá trị được truyền là 1.

Bước 4: Sử dụng Burp Suite để chắc chắn rằng chỉ có tham số cat ở trên là duy nhất.

Bước 5: Thực hiện thử truy cập bằng một số đường dẫn khác từ trang ban đầu, ta có thể đoán nhận kiểu truy vấn là SELECT

Bước 6: Thử giá trị tham số là laptop’ ta thấy có thông báo lỗi

Từ thông báo này cho thấy có khả năng website đã có lỗ hổng SQL Injection. Hơn nữa, từ thông báo ta có thể thấy phần mềm CSDL là MySQL. Để khẳng định rõ hơn mức độ ảnh hưởng của lỗ hổng này, chúng ta có thể thực hiện thêm một số bước kiểm thử khác

Bước 7: Thử giá trị tham số là laptop';%23, trong đó %23 là mã URL Encode của ký tự chú thích #. Ta thấy giá trị này được chấp nhận và có trang kết quả trả về. Từ đó ta đoán nhận được câu truy vấn có thể có dạng:

> Tham số đầu vào cần được đặt trong cặp dấu vì đây là phép so sánh xâu. Lưu ý rằng, SQL còn sử dụng toán tử LIKE để so sánh xâu. Khi đó, xâu đối sánh có thể chứa thêm các ký tự đại diện là ? và %. Các bạn tự thực hiện kiểm thử để cho thấy, nếu toán tử LIKE được sử dụng thì các ký tự đại diện này không có mặt.

- Bước 8: Sử dụng giá trị đầu vào là any' or 1;%23. Kết quả cho thấy có danh sách sản phẩm hiển thị. Ở bước này, ta có thể khẳng định website có lỗ hổng SQL Injection. Lỗ hổng này có thể bị khai thác bởi kỹ thuật Tautology-based SQL Injection.

Bước 9: Sử dụng toán tử ORDER BY để xác định số cột được truy vấn. Truyền giá trị tham số đầu vào là laptop' ORDER BY 2;%23. Trang kết quả trả về thành công cho thấy truy vấn có ít nhất 2 cột trong mệnh đề SELECT. Các bạn tự thực hiện với các giá trị khác để xác định số cột chính xác của truy vấn. Kết quả này cho thấy lỗ hổng có thể bị khai thác bởi mệnh đề ORDER BY.

Bước 10: Sử dụng toán tử UNION để khai thác thông tin khác trong CSDL. Trong bước trên, khi thực hiện khai thác bằng mệnh đề ORDER BY 2 chúng ta thấy các sản phẩm được sắp xếp theo thông tin Vendor. Do đó, có thể phán đoán rằng cột thứ 2 chứa thông tin Vendor sẽ được hiển thị. Sử dụng giá trị đầu vào laptop' UNION SELECT 1, database(), 3, 4, 5;%23, chúng ta thấy trong danh sách sản phẩm cuối cùng có tên của CSDL mà website đang sử dụng:

Như vậy, có thể thấy lỗ hổng này có thể bị khai thác bởi kỹ thuật dùng mệnh đề UNION.

Bước 11: Sử dụng kỹ thuật khai thác Boolean-based Blind SQL Injection

➢ Sử dụng giá trị laptop' and substring(database(),1,1)='a';%23 ta thấy không có danh sách sản phẩm trong trang kết quả trả về. Như vậy, tên CSDL mà website đang sử dụng không bắt đầu bằng chữ cái 'a'.

➢ Thực hiện tiếp tục cho tới giá trị laptop' and substring(database(),1,1)='w';%23, ta thấy có danh sách sản phẩm trong trang kết quả trả về. Như vậy tên CSDL mà website đang sử dụng bắt đầu bằng chữ cái 'w'

Như vậy, có thể thấy lỗ hổng này có thể bị khai thác bởi kỹ thuật Boolean-based Blind SQL Injection.

Bước 11: Sử dụng kỹ thuật khai thác Timed-based Blind SQL Injection.

➢ Sử dụng giá trị laptop' UNION SELECT 1, IF(SUBSTRING(database(),1,1) = 'a',BENCHMARK(500000,ENCODE('a','b')),null), 3, 4, 5;%23 ta thấy danh sách sản phẩm hiển thị ngay. Như vậy, hàm BENCHMARK() không thực hiện, tức là biểu thức trong lệnh IF là sai. Do đó, tên CSDL mà website đang sử dụng không bắt đầu bằng chữ cái 'a'.

➢ Sử dụng giá trị laptop' UNION SELECT 1, IF(SUBSTRING(database(),1,1) = 'w',BENCHMARK(500000,ENCODE('a','b')),null), 3, 4, 5;%23 ta thấy mất một khoảng thời giang,

danh sách sản phẩm hiển thị. Như vậy, hàm BENCHMARK() được thực hiện, tức là biểu thức trong lệnh IF là đúng. Do đó, tên CSDL mà website đang sử dụng không bắt đầu bằng chữ cái 'w'.

Như vậy, có thể thấy lỗ hổng này có thể bị khai thác bởi kỹ thuật Time-based Blind SQL Injection.

Ví dụ trên đã minh họa các bước cơ bản đề kiểm thử lỗ hổng SQL Injection trên website với kiểu truy vấn SELECT. Có thể thấy rằng, các giá trị đầu vào hoàn toàn không được kiểm soát. Trong các trường hợp khác, chúng ta cần phải thử thêm các giá trị đầu vào với các kỹ thuật vòng tránh.

5.2.2. Ví dụ 2

Bước 1: Truy cập địa chỉ http:192.168.52.1///sqli/sqli1.html

Bước 2: Điền các giá trị để thử chức năng. Từ kết quả, ta có thể phán đoán kiểu truy vấn như sau

Bước 3: Sử dụng Burp Suite ta có thể xác định được có 4 tham số đầu vào từ form nhập dữ liệu truyền tới server bằng phương thức POST.

Bước 4: Quan sát từ trang kết quả, ta thấy cả 4 giá trị của tham số đầu vào đều được sử dụng trong câu truy vấn INSERT INTO

Sau đây, ta sẽ kiểm tra xem có lỗ hổng SQL Injection trên tham số vendor không.

Bước 5: Nhập giá trị anyVendor' cho mục Vendor, các mục khác sử dụng lại các giá trị như cũ. Kết quả nhận được là một thông báo lỗi "Error! Cannot add product". Thông báo này chưa khẳng định được có lỗ hổng SQL Injection không.

Bước 6: Ta thực hiện phán đoán vị trí của giá trị tham số vendor trong danh sách giá trị. Nhập giá trị anyVendor');# cho mục Vendor, các mục khác sử dụng lại các giá trị như cũ. Kết quả nhận được là thông báo lỗi.

Bước 7: Thử lần lượt các giá trị anyVendor', '1');#, anyVendor', '1', '2');# ta đều nhận được thông báo lỗi.

Bước 8: Thử giá trị anyVendor', '1', '2', '3');# ta nhận được trang kết quả cho thấy một sản phẩm mới được thêm vào. Như vậy, có thể kết luận website có lỗ hổng SQL Injection ở vị trí tham số vendor.

Ngoài ra với kết quả như trên, mặc dù chưa chắc chắn nhưng dựa trên ngữ nghĩa, ta cũng xác định được thứ tự lần lượt giá trị của các tham số trong mệnh đề VALUES lần lượt là vendor, model, price, cat. Ta có thể thực hiện thêm một số kiểm thử để xác định mức độ ảnh hưởng của lỗ hổng này.

Bước 9: Với vị trí của các tham số đã được xác định ở bước trên, ta có thể phán đoán câu truy vấn có thể như sau:

Do đó, ta sử dụng giá trị sau để kiểm thử anyVendor', database(), '2', '3');#. Trên trang kết quả, ta có thể thấy tên cơ sở dữ liệu là webvul đã được hiển thị.

Có thể sử dụng các hàm version(), user(),… để thấy lỗ hổng này có thể bị khai thác để do thám thông tin về cơ sở dữ liệu

Bước 10: Tiếp tục sử dụng giá trị sau anyVendor', (select 'anyModel'), '2', '3');#. Trang kết quả có một sản phẩm mới được thêm vào. Như vậy có thể thấy lỗ hổng còn có thể bị khai thác bằng truy vấn

SELECT. Điều này cho phép kẻ tấn công khai thác để lấy ra dữ liệu tùy ý tùy thuộc quyền truy cập của tài khoản trên phần mềm quản trị cơ sở dữ liệu

Bước 9: Thực hiện tương tự, ta cũng có thể xác định được ở vị trí các tham số còn lại cũng gặp lỗ hổng tương tự.

Ví dụ trên đã minh họa các bước cơ bản đề kiểm thử lỗ hổng SQL Injection trên website với kiểu truy vấn INSERT. Có thể thấy rằng, các giá trị đầu vào hoàn toàn không được kiểm soát. Trong các trường hợp khác, chúng ta cần phải thử thêm các giá trị đầu vào với các kỹ thuật vòng tránh.