**Appointment**

**I. GIỚI THIỆU SQL INJECTION ĐỐI VỚI ỨNG DỤNG WEB.**

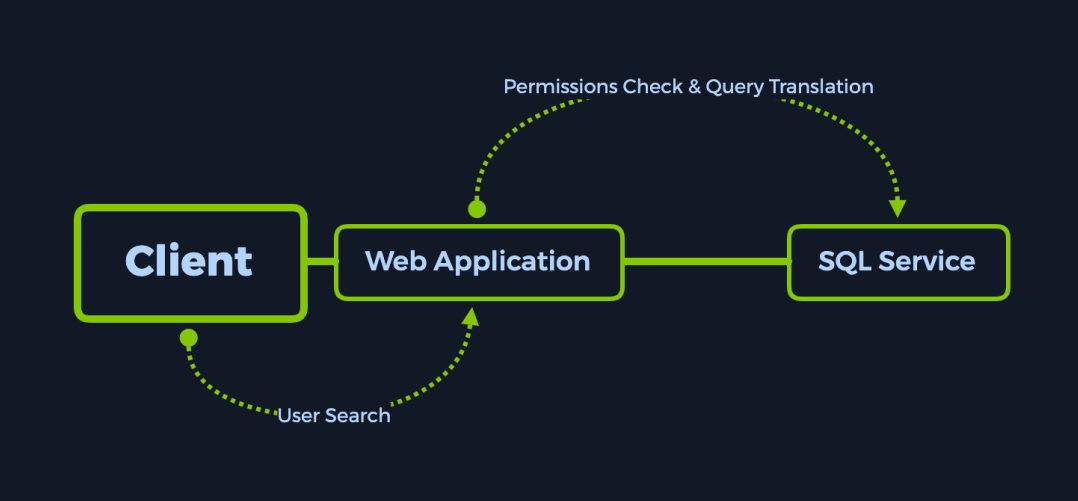
Mô tả: Chạy một trang web có chức năng tìm kiếm dựa trên cơ sở dữ liệu back-end chứa các mục có thể tìm kiếm dễ bị tấn công. Không phải người dùng nào cũng có thể nhìn thấy tất cả các mục trong cơ sở dữ liệu này, vì vậy các đặc quyền khác nhau trên trang web sẽ trả về kết quả tìm kiếm khác nhau.

Theo giả thuyết, quản trị viên của trang web tìm kiếm người dùng, email của họ, thông tin thanh toán, địa chỉ giao hàng và những thứ khác. Ngược lại, một người dùng bình thường hoặc khách truy cập chưa được xác thực có thể chỉ có quyền tìm kiếm các sản phẩm được bán. Các bảng thông tin này sẽ riêng biệt. Tuy nhiên, đối với kẻ tấn công có kiến ​​thức về lỗ hổng ứng dụng web - cụ thể là SQL Injection, trong trường hợp này - sự tách biệt giữa các bảng đó sẽ không có ý nghĩa gì, vì chúng sẽ có thể khai thác ứng dụng web để truy vấn trực tiếp bất kỳ bảng nào được tìm thấy trên Cơ sở dữ liệu SQL của máy chủ web.



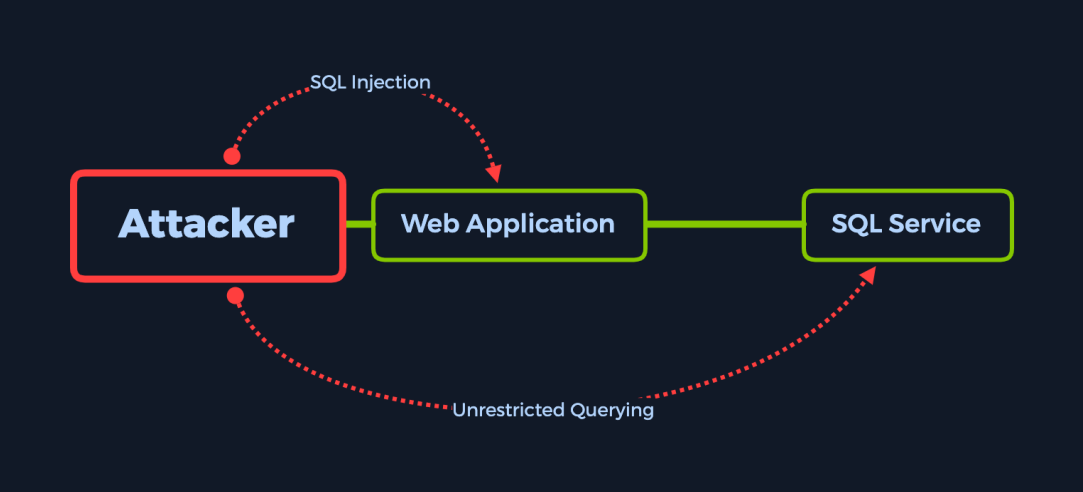
Vi dụ về Dịch vụ SQL thường hoạt động là quy trình đăng nhập được sử dụng cho bất kỳ người dùng nào. Mỗi lần người dùng muốn đăng nhập, ứng dụng web sẽ gửi đầu vào trang đăng nhập (kết hợp tên người dùng/ mật khẩu) đến Dịch vụ SQL, so sánh nó với các mục nhập cơ sở dữ liệu được lưu trữ cho người dùng cụ thể đó. Giả sử tên người dùng và mật khẩu được chỉ định khớp với bất kỳ mục nhập nào trong cơ sở dữ liệu. Trong trường hợp đó, Dịch vụ SQL sẽ báo cáo hệ thống trở lại ứng dụng web, người dùng sẽ đăng nhập, cấp cho họ quyền truy cập vào các phần bị hạn chế của trang web. Sau khi đăng nhập, ứng dụng web sẽ đặt cho người dùng một quyền đặc biệt dưới dạng cookie với sự hiện diện đã được xác thực trên trang web. Cookie này được lưu trữ cục bộ, trên bộ lưu trữ trình duyệt của người dùng và máy chủ web.

Sau đó, người dùng tìm kiếm cụ thể trong danh sách các mục được liệt kê trên trang, họ sẽ nhập tên của đối tượng vào thanh tìm kiếm, điều này sẽ kích hoạt cùng một Dịch vụ SQL chạy truy vấn SQL. Giả sử một mục nhập cho mục được tìm kiếm tồn tại trong cơ sở dữ liệu, thường nằm trong một bảng khác. Trong trường hợp đó, thông tin liên quan được truy xuất và gửi đến ứng dụng web để hiển thị cho người dùng dưới dạng hình ảnh, văn bản, liên kết và các loại khác, chẳng hạn như nhận xét và đánh giá.



Lý do các trang web sử dụng cơ sở dữ liệu như MySQL, MariaDB hoặc các loại khác là dữ liệu họ thu thập hoặc phục vụ cần được lưu trữ. Dữ liệu có thể là tên người dùng, mật khẩu, bài đăng, tin nhắn hoặc các thông tin nhạy cảm hơn như PII (Thông tin nhận dạng cá nhân).

SQL Injection là một cách phổ biến để khai thác các trang web sử dụng Câu lệnh SQL để truy xuất và lưu trữ dữ liệu đầu vào của người dùng. Nếu cấu hình sai, người ta có thể sử dụng cuộc tấn công này để khai thác lỗ hổng SQL Injection nổi tiếng, rất nguy hiểm. Có nhiều kỹ thuật khác nhau để bảo vệ khỏi việc tiêm SQL, một số trong số đó là xác thực đầu vào, truy vấn được tham số hóa, thủ tục được lưu trữ và triển khai WAF (Tường lửa Ứng dụng Web) trên mạng máy chủ. Tuy nhiên, có thể tìm thấy các trường hợp không có bản sửa lỗi nào trong số này, do đó, kiểu tấn công này lại phổ biến, theo danh sách 10 lỗ hổng web hàng đầu của OWASP.



**II. TIER1.APPOINTMENT – HACKTHEBOX**

Task 1: What does the acronym SQL stand for?

Structured Query Language

Task 2: What is one of the most common type of SQL vulnerabilities?

SQL injection

Task 3: What does PII stand for?

Personally Identifiable Information

Task 4: What does the OWASP Top 10 list name the classification for this vulnerability?

A03:2021-Injection

Task 5: What service and version are running on port 80 of the target?

Apache httpd 2.4.38 ((Debian))

Task 6: What is the standard port used for the HTTPS protocol?

443

Task 7: What is one luck-based method of exploiting login pages?

brute-forcing

Task 8: What is a folder called in web-application terminology?

directory

Task 9: What response code is given for "Not Found" errors?

404

Task 10: What switch do we use with Gobuster to specify we're looking to discover directories, and not subdomains?

dir

Task 11: What symbol do we use to comment out parts of the code?

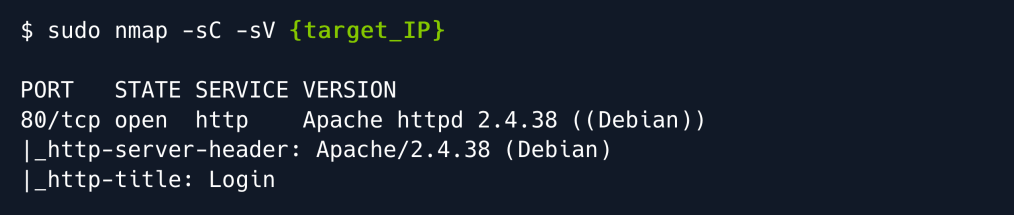
#

Submit root flag: e3d0796d002a446c0e622226f42e9672

**III. HƯỚNG DẪN**

**1. Sử dụng nmap quét máy chủ mục tiêu với tham số -sV**

Thực hiện quét nmap để tìm các cổng đang mở và có sẵn cũng như các dịch vụ. Nếu không có cờ được chỉ định, nmap sẽ quét 1000 cổng TCP phổ biến nhất cho các dịch vụ đang hoạt động.



Cú pháp: nmap –sC –sV <target>

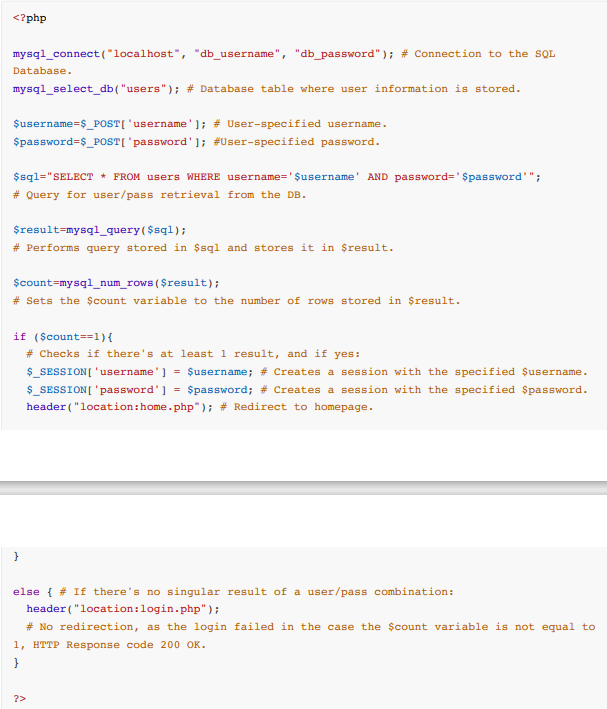
-sC: Thực hiện quét tập lệnh bằng cách sử dụng bộ tập lệnh mặc định. Nó tương đương với --script = default.

-sV: Cho phép phát hiện phiên bản dịch vụ đang chạy trên các cổng.

**2. Khai thác lỗ hổng SQL**

- Kiểm tra biểu mẫu đăng nhập để tìm lỗ hổng SQL Injection có thể xảy ra

Dưới đây là một ví dụ về cách xác thực hoạt động bằng PHP & SQL:



Để ý sau ký hiệu #, mọi thứ chuyển thành chú thích. Đoạn mã trên dễ bị tấn công SQL Injection, có thể sửa đổi truy vấn (biến $sql) thông qua biểu mẫu đăng nhập trên trang web để khiến truy vấn thực hiện bỏ qua đăng nhập hoàn toàn do không có biểu thức hoặc hàm nào lọc các ký tự đặc biệt như một dấu ngoặc kép hoặc thẻ bắt đầu bằng #. Trong khi ký hiệu thẻ bắt đầu bằng # được sử dụng để đưa ra nhận xét.

- Câu truy vấn kiểm tra xác thực người dùng có thể là:

SELECT \* FROM users WHERE username = ‘user’ AND password= ‘pass’

+ Trong đó: user, pass là giá trị người dùng nhập thông qua biểu mẫu đăng nhập

- Kết hợp # để trở thành câu truy vấn với điều kiện luôn đúng bằng cách thay giá trị username nhập vào bằng: 1’ OR 1=1 #

SELECT \* FROM users WHERE username = ‘1’ OR 1=1 #’ AND password= ‘admin\_password’

=> Các điều kiện sau ký tự # trở thành chú thích câu lệnh truy vấn.

=> Câu lệnh kiểm tra tài khoản tồn tại hay không luôn trả về kết quả đúng.