ỦY BAN NHÂN DÂN THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC SÀI GÒN
KHOA CÔNG NGHỆ THÔNG TIN

AN TOÀN VÀ BẢO MẬT DỮ LIỆU TRONG HỆ THỐNG THÔNG TIN

ĐỀ TÀI:

TÌM HIỂU CÁC PHƯƠNG PHÁP TẤN CÔNG SQL INJECTION

Nhóm 13

Sinh viên thực hiện:

3119560029	**Phùng Duy Khang**
3119560017	**Nguyễn Văn Hiền**
3119560053	**Nguyễn Thái Phương**
3119560007	**Huỳnh Lâm Khánh Duy**
3119560043	**Trần Quang Minh**

Thành phố Hồ Chí Minh, tháng 4 năm 2022

MỤC LỤC

TỔNG QUAN VỀ ĐỀ TÀI

1. Lý do chọn đề tài:

Với sự phát triển không ngừng của công nghệ thông tin và Internet, các ứng dụng web trở nên ngày càng phổ biến và quan trọng hơn đối với các tổ chức và doanh nghiệp. tấn công SQL Injection được xem là một trong những mối đe dọa lớn nhất đối với bảo mật hệ thống thông tin hiện nay. Bằng cách khai thác lỗ hổng của phần mềm hệ thống, kẻ tấn công có thể chiếm quyền kiểm soát cơ sở dữ liệu và gây thiệt hại nghiêm trọng cho các tổ chức và cá nhân sử dụng hệ thống.

Tấn công SQL Injection có thể được thực hiện bằng nhiều cách khác nhau và không ngừng tiến hóa, nên việc nghiên cứu về tấn công này là một công việc không bao giờ cũ và cần được cập nhật liên tục. Nghiên cứu về kỹ thuật tấn công SQL Injection và các giải pháp phòng chống tấn công này sẽ giúp có thêm kiến thức về bảo mật hệ thống thông tin và cải thiện khả năng phát hiện và giải quyết các vấn đề liên quan đến an ninh thông tin. Ngoài ra, đề tài còn mang tính thực tiễn cao và ứng dụng rộng rãi trong nhiều lĩnh vực.

2. Mục tiêu:

Tìm hiểu sâu về kỹ thuật tấn công SQL Injection, phân tích cách thức tấn công và tìm ra các lỗ hổng bảo mật trong các ứng dụng web. Bằng cách phân tích và đánh giá các phương pháp tấn công hiện tại, đề xuất các giải pháp bảo mật hiệu quả để ngăn chặn và giảm thiểu các tấn công SQL Injection trong các ứng dụng web.

3. Phạm vi:

Tập trung vào kỹ thuật tấn công SQL Injection trong các ứng dụng web và các phương pháp bảo mật để ngăn chặn các cuộc tấn công này, phân tích cách thức tấn công SQL Injection, những hậu quả và tổn thất mà nó gây ra, đồng thời đưa ra các giải pháp bảo mật hiệu quả nhằm giảm thiểu và ngăn chặn các cuộc tấn công SQL Injection trong các ứng dụng web.

CHƯƠNG 1: SƠ LƯỢC VỀ TẤN CÔNG SQL INJECTION

1.1. Khái niệm:

SQL Injection là một kỹ thuật tấn công phổ biến trong lĩnh vực bảo mật thông tin, mà được sử dụng để khai thác lỗ hổng bảo mật trong ứng dụng web. Khi một ứng dụng web cho phép người dùng nhập dữ liệu vào các trang web hay trang tìm kiếm, tấn công SQL Injection sẽ được sử dụng để chèn thêm câu lệnh SQL vào các dòng mã, để thực hiện các hành động không được cho phép, chẳng hạn như truy cập vào cơ sở dữ liệu, truy vấn và xóa các dữ liệu quan trọng, hay thậm chí là cài đặt chương trình độc hại trên máy chủ.

SQL Injection thường xảy ra khi ứng dụng web không xác thực hoặc kiểm tra đầu vào của người dùng đủ tốt, cho phép những câu lệnh SQL độc hại được chèn vào trong đó. Để bảo vệ ứng dụng web của mình khỏi SQL Injection, các nhà phát triển cần phải xây dựng các phương pháp xác thực đầu vào, sử dụng các thủ tục lọc dữ liệu và sử dụng các thư viện bảo mật để giảm thiểu nguy cơ tấn công.

1.2. Mục đích của SQL Injection:

Mục đích của tấn công SQL là khai thác các lỗ hổng bảo mật trong hệ thống quản trị cơ sở dữ liệu (DBMS) để có thể truy cập, sửa đổi hoặc xóa các dữ liệu trong cơ sở dữ liệu một cách trái phép. Tấn công SQL thường được thực hiện thông qua việc chèn các câu lệnh SQL độc hại vào các trang web hoặc các truy vấn cơ sở dữ liệu được thực hiện bởi ứng dụng web.

Tấn công SQL có thể gây ra những hậu quả nghiêm trọng, bao gồm lộ thông tin cá nhân, thay đổi dữ liệu, phá hoại hoặc tiêu hủy cơ sở dữ liệu và gây ảnh hưởng đến hoạt động của hệ thống.

Để bảo vệ hệ thống khỏi các cuộc tấn công SQL, các nhà phát triển và quản trị viên cần phải xây dựng các biện pháp bảo mật, bao gồm kiểm tra đầu vào, sử dụng các cấu trúc dữ liệu an toàn, giới hạn quyền truy cập và thực hiện các chính sách bảo mật nghiêm ngặt.

1.3. Hậu quả của tấn công SQL Injection:

Tấn công SQL Injection là một trong những kỹ thuật tấn công phổ biến và nguy hiểm nhất đối với các ứng dụng web. Nó làm cho các hacker có thể truy cập vào cơ sở dữ liệu của ứng dụng web bằng cách chèn các truy vấn độc hại vào các trang web hoặc biểu mẫu trực tuyến. Các hậu quả của tấn công SQL Injection có thể gây ra ảnh hưởng đến cả người sử dụng ứng dụng và nhà cung cấp dịch vụ :

- Một trong những hậu quả đáng lo ngại nhất của tấn công SQL Injection là việc ăn cắp dữ liệu. Với các truy vấn độc hại được chèn vào ứng dụng web, hacker có thể truy cập và lấy đi thông tin nhạy cảm từ cơ sở dữ liệu như tên đăng nhập, mật khẩu, thông tin tài khoản ngân hàng, thông tin cá nhân của khách hàng và nhiều thông tin quan trọng khác. Việc lộ thông tin cá nhân nhạy cảm này có thể gây ra những hậu quả nghiêm trọng cho người sử dụng, bao gồm mất tài sản, mất danh tính hoặc tình trạng trộm cắp danh tính.
- Thay đổi, xóa hoặc phá hủy dữ liệu. Nếu hacker có thể truy cập vào cơ sở dữ liệu của ứng dụng web, họ có thể thay đổi hoặc xóa bỏ dữ liệu của khách hàng, người dùng hoặc của tổ chức. Việc này có thể dẫn đến những rắc rối và tổn thất lớn đến việc kinh doanh của tổ chức.
- Giảm hiệu quả hoạt động của ứng dụng web. Khi hacker tiến hành tấn công SQL Injection, nó có thể gây ra một lượng lớn các truy vấn đến cơ sở dữ liệu. Điều này sẽ dẫn đến tình trạng quá tải, làm chậm hoạt động của hệ thống và làm cho ứng dụng web trở nên chậm chạp và không hoạt động đúng cách.
- Nếu người dùng truy cập website khi nó bị sập sẽ ảnh hưởng đến danh tiếng của công ty, nếu website sập trong thời gian dài thì có thể người dùng sẽ bỏ đi, lựa chọn dịch vụ khác thay thế.
- Uy tín của tổ chức bị ảnh hưởng. Khi thông tin cá nhân của khách hàng bị ăn cắp hoặc thông tin của tổ chức bị thay đổi, xóa bỏ, điều này có thể gây ra sự mất niềm tin của khách hàng đối với tổ chức. Sự mất niềm tin này có thể làm giảm doanh số bán hàng, tác động đến tài chính và ảnh hưởng đến hình ảnh của tổ chức trên thị trường.

1.4. Lịch sử của DoS và một số vụ tấn công nổi tiếng:

Lịch sử tấn công SQL injection bắt đầu từ những năm 1990 khi các ứng dụng web bắt đầu trở nên phổ biến. Các tấn công SQL injection đầu tiên chỉ đơn giản là những phép thử và sai, tìm kiếm các lỗ hổng bảo mật để truy cập cơ sở dữ liệu. Tuy nhiên, với sự phát triển của công nghệ và kỹ thuật, các cuộc tấn công này trở nên phức tạp và nguy hiểm hơn.

Dưới đây là một số vụ tấn công SQL nổi tiếng:

- Cuộc tấn công vào Sony Pictures (2014): Khoảng 100 trang web của Sony Pictures bị tấn công bằng cách sử dụng kỹ thuật SQL injection. Kết quả là hàng trăm nghìn tài liệu nhạy cảm của công ty bị rò rỉ, bao gồm thông tin nhân viên, thông tin tài khoản ngân hàng, thư điện tử, hợp đồng và bản quyền của các bộ phim.
- Cuộc tấn công vào eBay (2014): Được xem là một trong những vụ tấn công lớn nhất liên quan đến SQL injection, khoảng 145 triệu tài khoản của eBay bị đánh cắp bởi một nhóm tội phạm sử dụng kỹ thuật này.
- Cuộc tấn công vào Yahoo (2012): Khoảng 400.000 tài khoản của Yahoo bị đánh cắp thông qua một cuộc tấn công SQL injection. Các tài khoản bị rò rỉ bao gồm tên đăng nhập, mật khẩu và địa chỉ email.
- Cuộc tấn công vào Heartland Payment Systems (2008): Đây là một trong những vụ tấn công lớn nhất liên quan đến SQL injection trong lịch sử. Khoảng 130 triệu thẻ tín dụng của khách hàng của Heartland Payment Systems bị đánh cắp bởi một nhóm tội phạm sử dụng kỹ thuật này.

Các vụ tấn công này cho thấy tầm quan trọng của việc bảo vệ hệ thống khỏi các cuộc tấn công SQL injection. Những biện pháp bảo mật như kiểm tra đầu vào, sử dụng các cấu trúc dữ liệu an toàn và giới hạn quyền truy cập đều rất quan trọng để ngăn chặn các cuộc tấn công này.

CHƯƠNG 2: TẤN CÔNG SQL INJECTION

2.1. Các dạng tấn công SQL Injection:

Các dạng tấn công SQL Injection bao gồm:

- Tấn công SQL Injection cơ bản: là phương thức tấn công đơn giản nhất, khi hacker chèn các câu lệnh SQL độc hại vào các trang web hoặc biểu mẫu trực tuyến để truy cập cơ sở dữ liệu.
- Tấn công Blind SQL Injection: là phương thức tấn công khi hacker không thể xem được dữ liệu trực tiếp từ cơ sở dữ liệu, nhưng có thể xác định các giá trị đúng hoặc sai thông qua các thông báo lỗi hoặc kết quả trả về của trang web.
- Tấn công Inference SQL Injection: là phương thức tấn công tương tự như tấn công Blind SQL Injection, nhưng hacker sử dụng các truy vấn để suy đoán dữ liệu cần truy cập.
- Tấn công Error-based SQL Injection: là phương thức tấn công sử dụng các lỗi trong cú pháp SQL để truy cập cơ sở dữ liệu.
- Tấn công Union-based SQL Injection: là phương thức tấn công khi hacker sử dụng các truy vấn UNION để kết hợp các bảng và truy xuất dữ liệu từ nhiều bảng khác nhau trong cơ sở dữ liệu.
- Tấn công Time-based SQL Injection: là phương thức tấn công khi hacker sử dụng các câu lệnh SQL để trì hoãn việc trả về kết quả từ cơ sở dữ liệu để xác định các giá trị đúng hoặc sai.
- Tấn công Out-of-band SQL Injection: là phương thức tấn công khi hacker sử dụng các kênh khác nhau để gửi dữ liệu từ cơ sở dữ liệu, chẳng hạn như email hoặc DNS.
- Tấn công Second-order SQL Injection: là phương thức tấn công khi hacker sử dụng các trang web hoặc ứng dụng web để lưu trữ các truy vấn SQL độc hại, sau đó sử dụng chúng để truy cập cơ sở dữ liệu sau này.

Các dạng tấn công phổ biến hiện nay:

a. Tấn công SQL Injection dựa trên lỗi cơ bản (Error-Based SQL):

Đây là dạng tấn công SQL Injection đơn giản nhất và dễ thực hiện nhất. Khi các lập trình viên không kiểm tra và xử lý đầu vào từ người dùng đúng cách, hacker có thể chèn các truy vấn độc hại vào câu lệnh SQL ban đầu của ứng dụng web. Các truy vấn độc hại này có thể được chèn vào bất cứ đâu trong trang web hoặc biểu mẫu trực tuyến.

Cách thức hoạt động:

Bước 1: Hacker tìm kiếm các trang web dễ bị tấn công SQL Injection.

Bước 2: Hacker chèn các truy vấn độc hại vào đầu vào của ứng dụng web.

Bước 3: Các truy vấn độc hại này được thực thi trên cơ sở dữ liệu của ứng dụng web, cho phép hacker truy cập và lấy thông tin từ cơ sở dữ liệu.

Ví dụ: Giả sử ta có một trang web đăng nhập cho người dùng và nó sử dụng một câu truy vấn SQL để kiểm tra thông tin đăng nhập của người dùng như sau:

```
SELECT * FROM users WHERE username='$username' AND
password='$password'
```

Trong đó, $username và $password là các biến chứa thông tin đăng nhập của người dùng. Tuy nhiên, trang web không kiểm tra hoặc xác thực các giá trị này, điều này dẫn đến lỗ hổng bảo mật và kẻ tấn công có thể tận dụng lỗi này để thực hiện tấn công SQL Injection.

Nếu kẻ tấn công muốn đăng nhập vào tài khoản người dùng mà không cần biết mật khẩu, họ có thể nhập một chuỗi ký tự đặc biệt như sau:

```
' OR 1=1 --
```

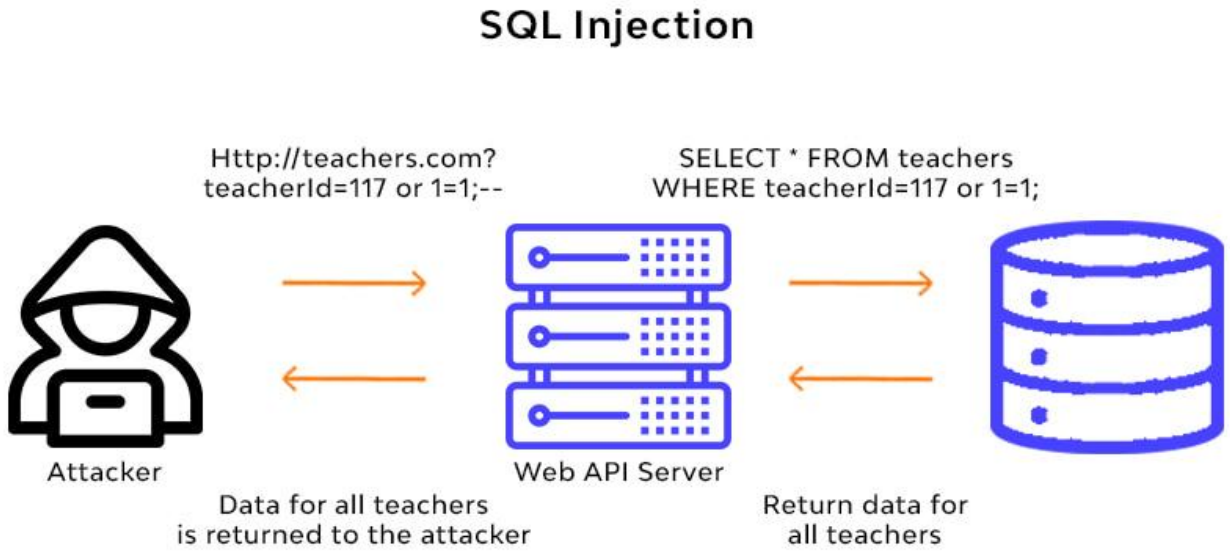
Khi đó, câu truy vấn SQL sẽ trở thành:

```
SELECT * FROM users WHERE username='' OR 1=1 -- ' AND
password='$password'
```

Trong đó, dấu hai gạch ngang (--) đánh dấu một chú thích trong SQL, điều này có nghĩa là mọi thứ sau đó sẽ được bỏ qua. Như vậy, câu truy vấn SQL trở thành một câu

truy vấn không hợp lệ, nó sẽ trả về tất cả các bản ghi trong bảng người dùng bởi vì điều kiện "1=1" là luôn đúng. Vì vậy, kẻ tấn công sẽ có quyền truy cập vào tất cả các tài khoản người dùng mà không cần mật khẩu.

Để phòng chống tấn công SQL Injection dựa trên lỗi cơ bản, các nhà phát triển phần mềm nên sử dụng các phương pháp phòng chống SQL Injection đã được đề cập ở trên, bao gồm kiểm tra và xác thực dữ liệu đầu vào của người dùng, sử dụng thư viện SQL Parameters hoặc chuyển đổi các giá trị người dùng thành các chuỗi an toàn trước khi thực thi câu truy vấn SQL.



b. Tấn công SQL Injection dựa trên Union (Union SQL Injection):

Đây là dạng tấn công SQL Injection sử dụng câu lệnh UNION để kết hợp các truy vấn SQL khác vào truy vấn SQL ban đầu của ứng dụng web. Khi các lập trình viên không sử dụng các thủ tục bảo mật hoặc kiểm tra đầu vào của người dùng, hacker có thể sử dụng Union để chèn thêm các truy vấn SQL độc hại vào truy vấn ban đầu và truy xuất các thông tin nhạy cảm từ cơ sở dữ liệu. Các thông tin này sẽ được gộp lại và hiển thị trong trang web, cho phép hacker truy cập và lấy các thông tin mật của người dùng.

Cách thức hoạt động của tấn công Union SQL Injection thường được thực hiện bằng cách chèn một câu lệnh SELECT vào trang web đích và chọn các cột nhất định để lấy thông tin từ cơ sở dữ liệu. Sau đó, hacker sẽ chèn vào câu lệnh UNION và truy vấn SQL độc hại của mình, kết hợp với câu lệnh SELECT ban đầu để hiển thị thông tin từ cả hai truy vấn.

Ví dụ, giả sử trang web đích hiển thị thông tin sản phẩm từ cơ sở dữ liệu. Câu lệnh ban đầu của truy vấn SQL có thể là:

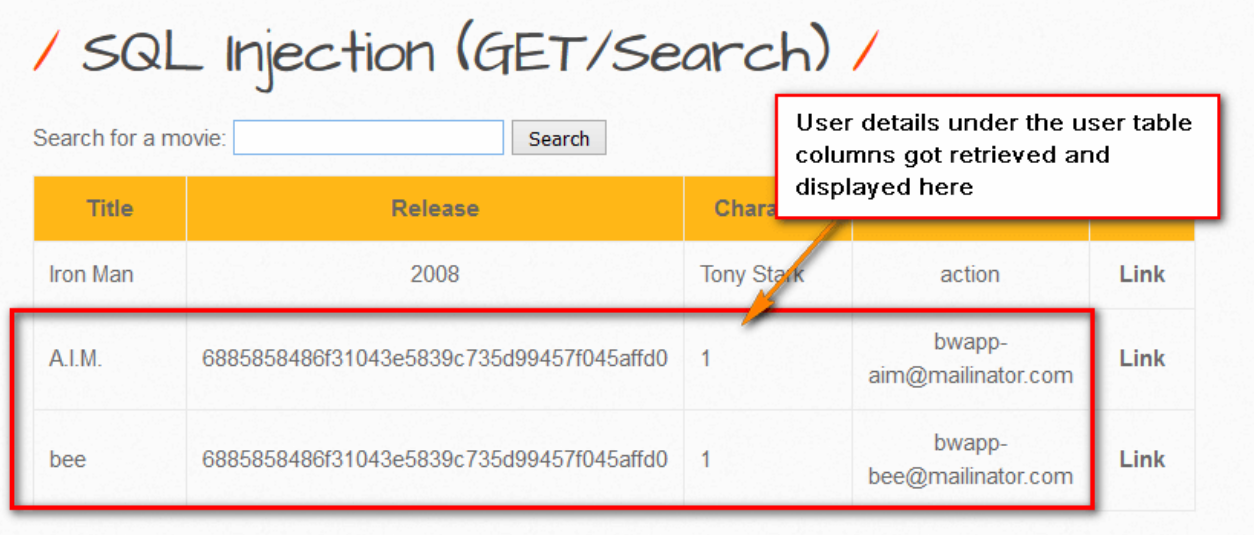
```
SELECT product_name, product_description FROM products WHERE
product_id = 1;
```

Hacker sẽ chèn vào đó một câu lệnh UNION và truy vấn SQL độc hại của mình để truy xuất các thông tin khác từ cơ sở dữ liệu:

```
SELECT username, password FROM users WHERE id = 1 UNION SELECT
product_name, product_description FROM products WHERE product_id
= 1;
```

Kết quả của truy vấn này sẽ hiển thị thông tin sản phẩm nhưng cũng bao gồm các thông tin tài khoản người dùng như tên đăng nhập và mật khẩu.

Để phòng chống tấn công Union SQL Injection, các lập trình viên cần phải thực hiện kiểm tra đầu vào của người dùng và sử dụng các cấu trúc Prepared Statement để thực hiện truy vấn SQL. Các hệ thống bảo mật cũng có thể sử dụng các công cụ như Web Application Firewall (WAF) để chặn các truy vấn độc hại và bảo vệ ứng dụng web khỏi tấn công Union SQL Injection.



c. Tấn công SQL Injection dựa trên Blind-Based:

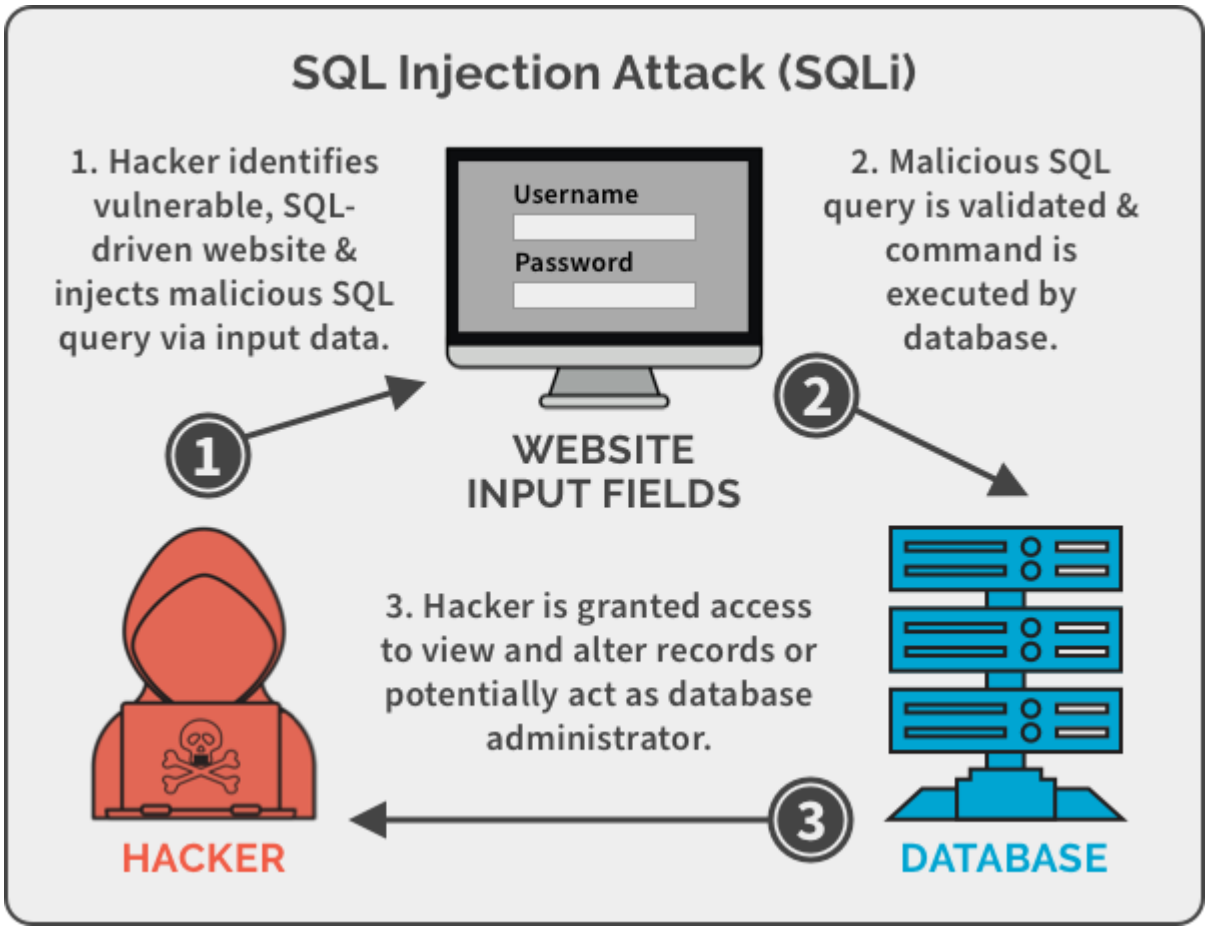
Là một dạng tấn công tập trung vào việc khai thác lỗ hổng của ứng dụng web mà không có bất kỳ phản hồi nào từ phía ứng dụng. Điều này có nghĩa là hacker sẽ không thấy được kết quả của các truy vấn SQL mà họ đưa vào, thay vào đó, họ phải dựa vào các phản hồi bên ngoài để xác định xem liệu một truy vấn SQL được thực thi thành công hay không.

Các hacker thường sử dụng các kỹ thuật dựa trên các phản hồi khác nhau từ ứng dụng web để xác định xem liệu một truy vấn SQL đã được thực thi thành công hay không. Các phản hồi này có thể bao gồm:

- Thông báo lỗi: Khi ứng dụng web phát hiện một lỗi SQL Injection, nó sẽ hiển thị thông báo lỗi cụ thể. Hacker có thể sử dụng thông báo lỗi này để xác định xem liệu một truy vấn SQL đã được thực thi thành công hay không.
- Thời gian phản hồi: Các hacker cũng có thể sử dụng thời gian phản hồi của ứng dụng web để xác định xem liệu một truy vấn SQL đã được thực thi thành công hay không. Nếu thời gian phản hồi của ứng dụng web tăng lên, điều này có thể cho thấy rằng một truy vấn SQL đã được thực thi.
- Phản hồi dựa trên URL: Hacker có thể thực hiện các truy vấn SQL theo cách nào đó để biến đổi các thông tin trong URL của trang web và xác định xem liệu một truy vấn SQL đã được thực thi thành công hay không. Ví dụ,

nếu hacker sử dụng một truy vấn SQL để truy xuất các thông tin từ cơ sở dữ liệu, họ có thể thay đổi các thông tin trong URL và xem xét phản hồi của ứng dụng web để xác định xem liệu một truy vấn SQL đã được thực thi thành công hay không.

Tấn công SQL Injection dựa trên Blind-Based là một dạng tấn công khá phức tạp và yêu cầu các kỹ năng kỹ thuật cao của các hacker. Các ứng dụng web phải đảm bảo rằng họ đã thực hiện các biện pháp bảo mật chặt chẽ để phòng ngừa các cuộc tấn công này.



2.2. Phương pháp phát hiện tấn công SQL Injection

Phát hiện tấn công SQL injection là một quá trình quan trọng trong việc bảo vệ các ứng dụng web trước các cuộc tấn công độc hại. Dưới đây là một số phương pháp phát hiện tấn công SQL injection:

- Kiểm tra đầu vào: Một cách đơn giản nhưng hiệu quả để phát hiện tấn công SQL injection là kiểm tra các đầu vào từ người dùng để đảm bảo rằng chúng không chứa các câu lệnh SQL độc hại.
- Sử dụng bộ lọc đầu vào: Một cách khác để phát hiện tấn công SQL injection là sử dụng các bộ lọc đầu vào để loại bỏ các ký tự đặc biệt và các câu lệnh SQL độc hại.
- Sử dụng công cụ phát hiện tấn công SQL injection: Có nhiều công cụ phát hiện tấn công SQL injection được phát triển để phát hiện các lỗ hổng bảo mật trong các ứng dụng web, dưới đây là một số công cụ:
    - SQLMap
    - Netsparker
    - Acunetix
    - AppScan
    - OpenVAS
- Kiểm tra lỗi cơ sở dữ liệu: Việc kiểm tra lỗi cơ sở dữ liệu có thể giúp phát hiện các tấn công SQL injection, bằng cách kiểm tra các lỗi trong các bảng cơ sở dữ liệu và xác định xem liệu chúng có bị lỗ hổng bảo mật hay không.
- Sử dụng các giải pháp bảo mật phân tích mã nguồn: Các giải pháp phân tích mã nguồn có thể giúp phát hiện tấn công SQL injection bằng cách tìm kiếm các câu lệnh SQL độc hại trong mã nguồn của ứng dụng web.

Tuy nhiên, không có một phương pháp nào là tuyệt đối an toàn để phát hiện tấn công SQL injection. Vì vậy, việc kết hợp các phương pháp phát hiện và bảo vệ là cần thiết để đảm bảo an toàn cho các ứng dụng web

2.3. Phương pháp phòng chống tấn công SQL Injection

Để phòng chống tấn công SQL injection, có thể áp dụng các phương pháp sau:

- Sử dụng các thư viện SQL parameterized queries: Đây là một phương pháp phòng chống tấn công SQL injection hiệu quả. Thay vì tạo câu truy vấn SQL trực tiếp bằng các giá trị đầu vào từ người dùng, thư viện parameterized queries sẽ sử dụng

các tham số để truyền các giá trị đầu vào. Điều này sẽ giúp tránh được việc kết hợp các đoạn mã SQL bất hợp lệ vào trong câu truy vấn.

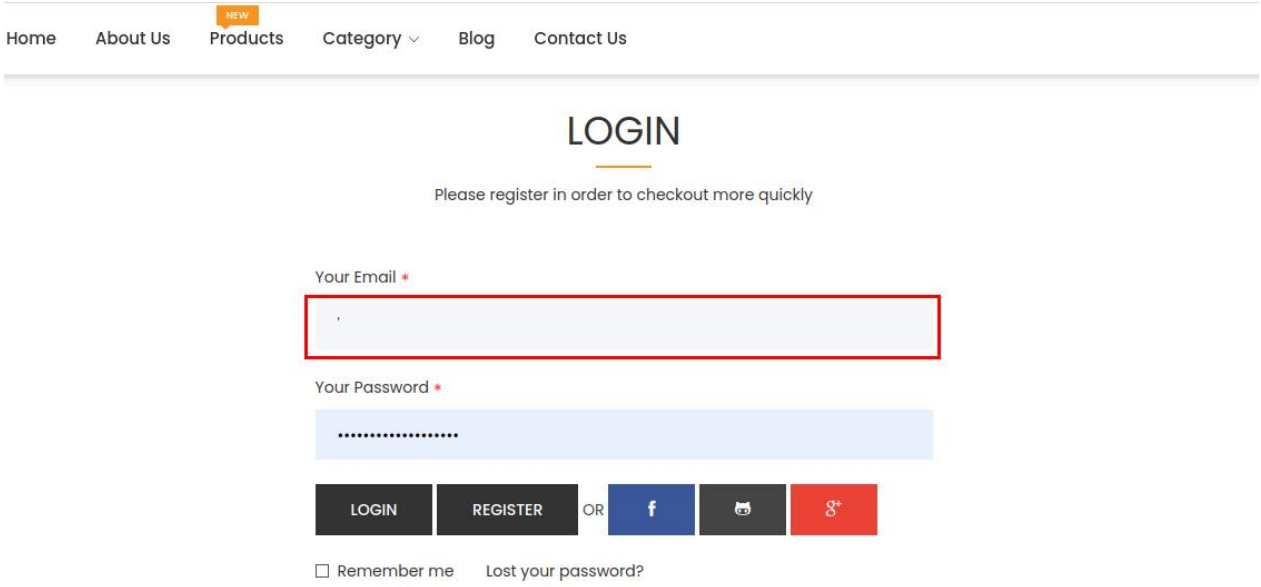
- Kiểm tra và xử lý đầu vào từ người dùng: Khi người dùng nhập dữ liệu vào các trang web, chúng ta nên kiểm tra dữ liệu đầu vào để đảm bảo rằng nó không chứa các ký tự đặc biệt hoặc các đoạn mã SQL bất hợp lệ. Nếu dữ liệu đầu vào chứa các ký tự đặc biệt, chúng ta nên xử lý hoặc loại bỏ chúng trước khi sử dụng.
- Cập nhật và bảo mật hệ thống: Các lỗ hổng bảo mật trong hệ thống cơ sở dữ liệu có thể trở thành điểm yếu cho tấn công SQL injection. Để giảm thiểu rủi ro này, chúng ta nên cập nhật hệ thống và bảo mật chúng một cách thường xuyên.
- Sử dụng các giải pháp bảo mật web: Các giải pháp bảo mật web như firewall ứng dụng hoặc các giải pháp bảo mật WAF (Web Application Firewall) có thể giúp ngăn chặn các cuộc tấn công SQL injection trước khi chúng đến được hệ thống cơ sở dữ liệu.
- Đào tạo nhân viên về an ninh thông tin: Đào tạo nhân viên về an ninh thông tin và các kỹ năng bảo mật web sẽ giúp tăng cường nhận thức về tấn công SQL injection và giúp họ đưa ra các giải pháp bảo mật phù hợp.
- Tuy nhiên, không có phương pháp nào đảm bảo 100% phòng chống tấn công SQL injection. Chính vì vậy, việc sử dụng một số phương pháp trên kết hợp với việc sử dụng các công cụ phát hiện lỗ hổng bảo mật có thể giúp giảm thiểu rủi ro của tấn công SQL injection đối với các

CHƯƠNG 3: THỰC HIỆN TẤN CÔNG SQL INJECTION

3.1. Tấn công SQL Injection dựa trên lỗi cơ bản:

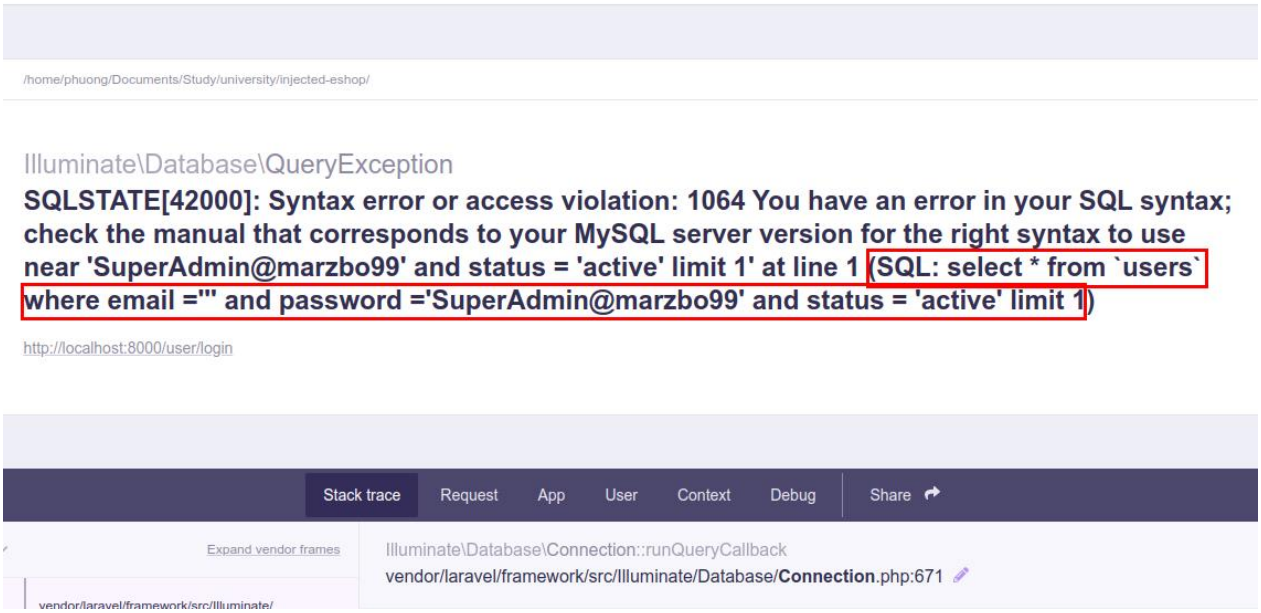
Thực hiện tấn công trên trang web mô phỏng ở localhost:

B1: Truy cập vào trang đăng nhập.

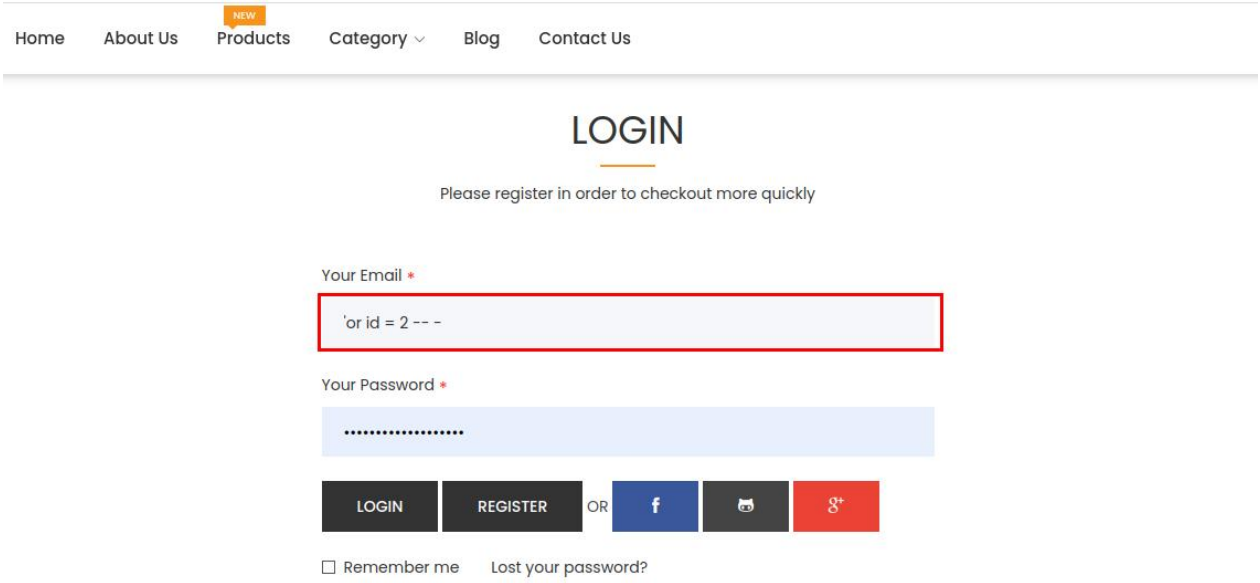


nhập vào dấu ‘ vào trường email để kiểm tra trang web có thể tấn công SQL injection được hay không

B2: Nếu hệ hiển thị như ảnh thì hệ có thể tấn công sql injection được



B3: nhập câu SQL Injection để truy cập trang web với tài khoản có id = 2;

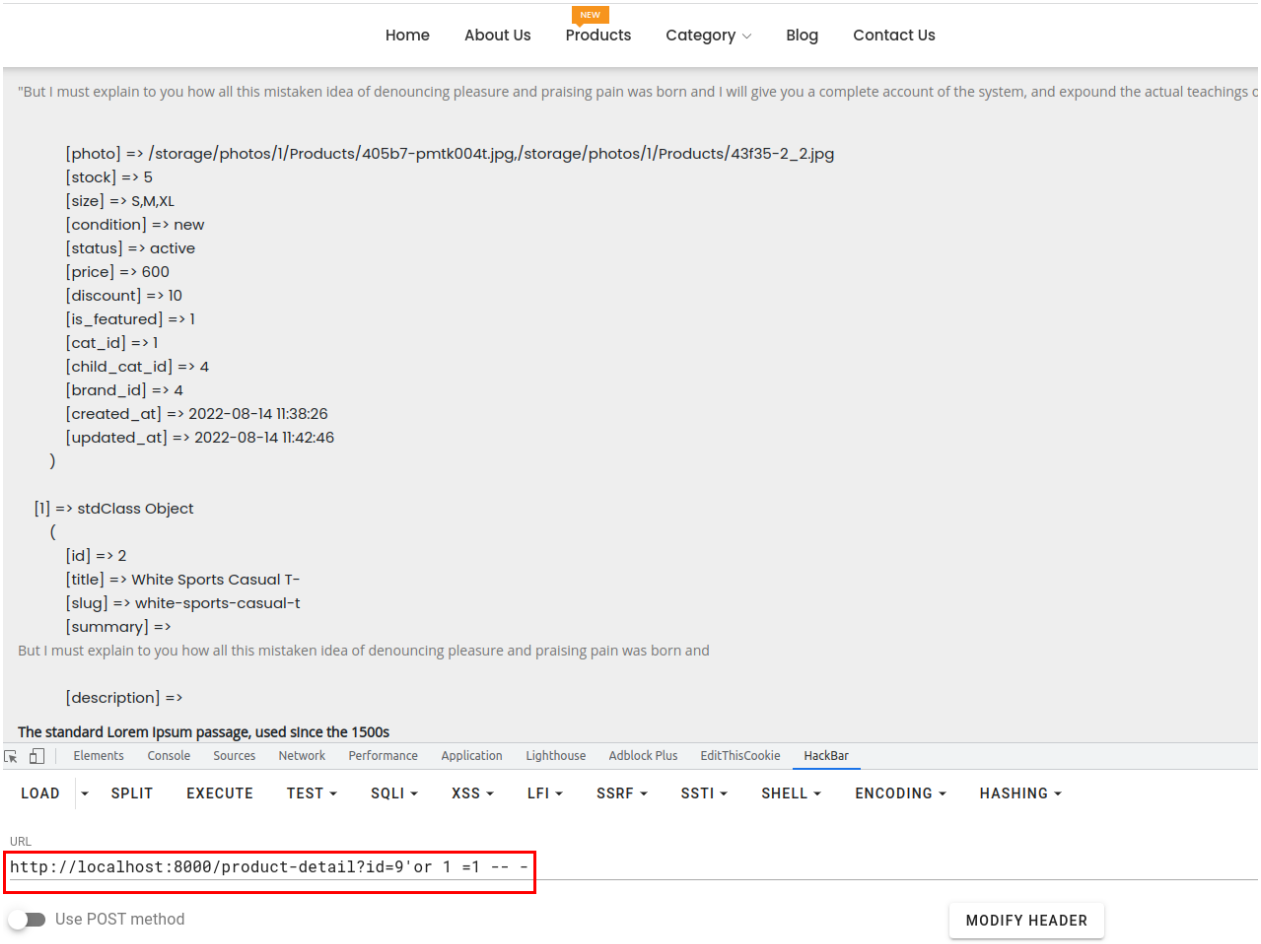


3.2. Tấn công SQL Injection dựa trên Blind-Based:

B1: tải extension:

https://chrome.google.com/webstore/detail/hackbar/djmoeoifnlhjolebkehmpaocfnipknbh?hl=vi

B2: nhập url trang web kèo theo câu lệnh tấn công SQL injection



B3: như vậy bạn đã thành công truy vấn được sản phẩm ẩn của trang web.

3.3. Phương pháp phòng chống

Để phòng chống tấn công SQL Injection, có thể áp dụng một số phương pháp sau đây:

Sử dụng câu lệnh Prepared Statements: Prepared Statements được sử dụng để tách biệt các câu lệnh SQL và các đối số của chúng. Điều này giúp ngăn chặn hacker chèn các mã độc vào trong câu lệnh SQL.

Sử dụng Stored Procedures: Stored Procedures là các chương trình được lưu trữ trong cơ sở dữ liệu và thực thi bởi cơ sở dữ liệu. Chúng giúp ngăn chặn hacker truy cập trực tiếp vào cơ sở dữ liệu.

Kiểm tra và xử lý đầu vào của người dùng: Đây là một phương pháp quan trọng để đảm bảo rằng đầu vào của người dùng là hợp lệ. Các lập trình viên có thể sử dụng các hàm mã hóa đầu vào hoặc các thư viện mã hóa để bảo vệ đầu vào của người dùng.

Sử dụng bộ lọc web: Bộ lọc web được sử dụng để chặn các yêu cầu độc hại trước khi chúng đến tới máy chủ web. Nó giúp bảo vệ ứng dụng web khỏi các cuộc tấn công SQL Injection.

Cập nhật thường xuyên: Các nhà phát triển phải thường xuyên cập nhật phần mềm và bảo mật của ứng dụng web để ngăn chặn các lỗ hổng bảo mật.

Sử dụng firewall: Firewall có thể giúp ngăn chặn các cuộc tấn công SQL Injection bằng cách chặn các yêu cầu đến cơ sở dữ liệu từ các địa chỉ IP không hợp lệ hoặc từ các loại yêu cầu đáng ngờ.

Các giải pháp khác: Ngoài các phương pháp phòng chống SQL Injection trên, còn có thể sử dụng các giải pháp khác như mã hóa dữ liệu, cấu hình bảo mật, sử dụng các chứng chỉ SSL/TLS, kiểm tra mã nguồn ứng dụng web trước khi triển khai để tìm kiếm các lỗ hổng bảo mật, v.v.

KẾT LUẬN

SQL Injection là một trong những phương thức tấn công phổ biến nhất đối với các ứng dụng web, và có thể gây ra những thiệt hại nghiêm trọng cho các hệ thống và dữ liệu của họ. Việc khai thác lỗ hổng SQL Injection có thể dẫn đến mất cắp thông tin quan trọng, thay đổi hoặc xóa dữ liệu, gây tổn hại nghiêm trọng đến hệ thống và danh tiếng của các tổ chức.

Các phương pháp tấn công SQL Injection đa dạng và phức tạp, bao gồm tấn công dựa trên lỗi cơ bản, tấn công dựa trên Union và tấn công dựa trên Blind-based. Để bảo vệ ứng dụng web của mình khỏi tấn công SQL Injection, các nhà phát triển cần phải thực hiện các biện pháp bảo mật như kiểm tra đầu vào người dùng, sử dụng thủ tục lưu trữ tham số hóa, mã hóa dữ liệu, cập nhật phần mềm định kỳ và tạo các quyền hạn cho người dùng.

Tuy nhiên, người dùng cũng có trách nhiệm chung trong việc bảo vệ hệ thống của mình bằng cách sử dụng các mật khẩu mạnh, tránh sử dụng các ký tự đặc biệt và thường xuyên thay đổi mật khẩu. Đồng thời, họ cũng nên sử dụng phần mềm diệt virus và bảo mật, cập nhật hệ thống định kỳ và tránh truy cập các trang web đáng ngờ hoặc lạ.

Vì vậy, chúng ta có thể kết luận rằng việc phòng chống SQL Injection là rất quan trọng và đóng vai trò quan trọng trong bảo vệ các ứng dụng web và dữ liệu của chúng ta. Chúng ta cần áp dụng các biện pháp phòng chống tấn công này để giảm thiểu nguy cơ bị tấn công và bảo vệ tài sản trực tuyến của mình.

Trong tổng thể, việc phòng chống SQL Injection là một quá trình liên tục và đòi hỏi sự chú ý và nỗ lực từ cả nhà phát triển lẫn người sử dụng. Bằng cách tìm hiểu về các mối đe dọa này và thực hiện các biện pháp bảo mật, chúng ta có thể đảm bảo an toàn cho dữ liệu và hệ thống của mình trong môi trường kỹ thuật số ngày càng phát triển và phức tạp.

TÀI LIỆU THAM KHẢO

[1] Wikipedia, "SQL injection" [Online]. https://vi.wikipedia.org/wiki/SQL_injection#:~:text=SQL%20injection%20l%C3%A0%20m%E1%BB%99t%20k%E1%BB%B9,l%E1%BB%87nh%20SQL%20b%E1%BA%A5t%20h%E1%BB%A3p%20ph%C3%A1p.

[2] "SQL Injection" - Microsoft - https://docs.microsoft.com/en-us/sql/relational-databases/security/sql-injection?view=sql-server-ver15