**Báo cáo giữa kỳ SQL INJECTION ATTACK**

1. **Lý thiết về công nghệ**

SQL Injection là một trong những kiểu [hack](https://quantrimang.com/nhung-hieu-biet-co-ban-nhat-de-tro-thanh-hacker-phan-1-4650) web bằng cách inject các mã [SQL](https://quantrimang.com/sql) query/command vào input trước khi chuyển cho ứng dụng web xử lí, bạn có thể login mà không cần username và password, remote execution (thực thi từ xa), dump data và lấy root của [SQL server](https://quantrimang.com/SQL-server). Công cụ dùng để tấn công là một trình duyệt web bất kì, chẳng hạn như Internet Explorer, Netscape, Lynx, ...

SQL injection – còn được gọi là SQLi – sử dụng những lỗ hổng trong các kênh đầu vào (input) của website để nhắm mục tiêu vào cơ sở dữ liệu nằm trong phần phụ trợ của ứng dụng web, nơi lưu giữ những thông tin nhạy cảm và có giá trị nhất. Chúng có thể được kẻ tấn công sử dụng để ăn cắp hoặc xáo trộn dữ liệu, cản trở sự hoạt động của các ứng dụng, và, trong trường hợp xấu nhất, nó có thể chiếm được quyền truy cập quản trị vào máy chủ cơ sở dữ liệu.

Để thực hiện một cuộc tấn công SQL Injection, kẻ tấn công đầu tiên phải tìm thấy đầu vào người dùng dễ bị tổn thương trong trang web hoặc ứng dụng web. Một trang web hoặc ứng dụng web có một lỗ hổng SQL Injection sử dụng đầu vào người dùng trực tiếp trong một truy vấn SQL. Kẻ tấn công có thể tạo nội dung nhập liệu. Nội dung như vậy thường được gọi là tải trọng độc hại và là phần chính của cuộc tấn công. Sau khi kẻ tấn công gửi nội dung này, độc hại SQL lệnh được thực hiện trong cơ sở dữ liệu.

## Sự nguy hiểm của SQL Injection

* Hack tài khoản của người khác
* Ăn cắp và copy dữ liệu nhạy cảm của hệ thống
* Thay đổi các dữ liệu nhạy cảm của hệ thống
* Xóa dữ liệu nhạy cảm của hệ thống
* Hacker có thể login vào ứng dụng như user khác, thậm chí là admin
* Hacker có thể xem thông tin cá nhân của user khác. Ví dụ: thông tin chi tiết profile, chi tiết giao dịch,...
* Hacker có thể thay đổi thông tin cấu hình của ứng dụng và dữ liệu của các user khác
* Hacker có thể sửa cấu trúc database, thậm chí delete bảng trong database
* Hacker có thể điều khiển server database và thực hiện câu lệnh trên nó.
* **Làm thế nào và tại sao là một cuộc tấn công SQL Injection thực hiện**

SQL là một ngôn ngữ truy vấn được thiết kế để quản lý dữ liệu được lưu trữ trong cơ sở dữ liệu quan hệ. Bạn có thể sử dụng nó để truy cập, sửa đổi và xóa dữ liệu. Nhiều ứng dụng web và trang web lưu trữ tất cả dữ liệu trong cơ sở dữ liệu SQL. Trong một số trường hợp, bạn cũng có thể sử dụng lệnh SQL để chạy lệnh hệ điều hành. Vì vậy, một cuộc tấn công SQL Injection thành công có thể có hậu quả rất nghiêm trọng.

* Kẻ tấn công có thể sử dụng SQL tiêm để tìm các thông tin đăng nhập của người dùng khác trong cơ sở dữ liệu. Sau đó, họ có thể mạo danh những người dùng này. Người dùng mạo danh có thể là người quản trị cơ sở dữ liệu với tất cả các đặc quyền cơ sở dữ liệu.
* SQL cho phép bạn chọn và xuất dữ liệu từ cơ sở dữ liệu. Một lỗ hổng SQL Injection có thể cho phép kẻ tấn công để truy cập đầy đủ vào tất cả dữ liệu trong một máy chủ cơ sở dữ liệu.
* SQL cũng cho phép bạn thay đổi dữ liệu trong cơ sở dữ liệu và thêm dữ liệu mới. Ví dụ, trong một ứng dụng tài chính, kẻ tấn công có thể sử dụng SQL Injection để thay đổi số dư, void giao dịch, hoặc chuyển tiền vào tài khoản của họ.
* Bạn có thể sử dụng SQL để xóa bản ghi khỏi cơ sở dữ liệu, thậm chí thả bảng. Ngay cả khi người quản trị làm cho cơ sở dữ liệu sao lưu, xóa dữ liệu có thể ảnh hưởng đến ứng dụng sẵn có cho đến khi cơ sở dữ liệu được khôi phục. Ngoài ra, sao lưu có thể không bao gồm các dữ liệu gần đây nhất.
* Trong một số máy chủ cơ sở dữ liệu, bạn có thể truy cập vào hệ điều hành bằng cách sử dụng máy chủ cơ sở dữ liệu. Điều này có thể là cố ý hoặc vô tình. Trong trường hợp này, kẻ tấn công có thể sử dụng một SQL Injection như vector ban đầu và sau đó tấn công mạng nội bộ đằng sau một bức tường lửa

## Một Số Dạng Tấn Công SQL Injection thông thường

### Dạng tấn công vượt qua kiểm tra lúc đăng nhập

Với dạng tấn công này, kẻ tấn công có thể dễ dàng vượt qua các trang đăng nhập nhờ vào lỗi khi dùng các câu lệnh SQL thao tác trên cơ sở dữ liệu của ứng dụng web. Thông thường để cho phép người dùng truy cập vào các trang web được bảo mật, hệ thống thường xây dựng trang đăng nhập để yêu cầu người dùng nhập thông tin về tên đăng nhập và mật khẩu. Sau khi người dùng nhập thông tin vào, hệ thống sẽ kiểm tra tên đăng nhập và mật khẩu có hợp lệ hay không để quyết định cho phép hay từ chối thực hiện tiếp.

### Dạng tấn công sử dụng câu lệnh SELECT

Dạng tấn công này phức tạp hơn. Để thực hiện được kiểu tấn công này, kẻ tấn công phải có khả năng hiểu và lợi dụng các sơ hở trong các thông báo lỗi từ hệ thống để dò tìm các điểm yếu khởi đầu cho việc tấn công.

### Dạng tấn công sử dụng câu lệnh INSERT

Thông thường các ứng dụng web cho phép người dùng đăng kí một tài khoản để tham gia. Chức năng không thể thiếu là sau khi đăng kí thành công, người dùng có thể xem và hiệu chỉnh thông tin của mình. SQL injection có thể được dùng khi hệ thống không kiểm tra tính hợp lệ của thông tin nhập vào.

### Dạng tấn công sử dụng stored-procedures

Stored Procedure được sử dụng trong lập trình web với mục đích nhằm giảm sự phức tạp trong ứng dụng và tránh sự tấn công trong kỹ thuật SQL Injection. Tuy nhiên những kẻ tấn công vẫn có thể lợi dụng những stored procedure để tấn công vào hệ thống. Việc tấn công này sẽ gây tác hại rất lớn nếu ứng dụng được thực thi với quyền quản trị hệ thống ‘sa’ .

## Các phương pháp phòng tránh

## Kiểm soát chặt chẽ dữ liệu đầu vào

Điểm yếu của SQL Injection bắt nguồn từ việc xử lý dữ liệu của người dùng không tốt, do đó vấn đề xây dựng mã nguồn đảm bảo an ninh là cốt lõi trong việc chống SQL Injection. Để phòng tránh các nguy cơ có thể xảy ra, hãy bảo vệ các câu lệnh sql bằng cách kiểm soát chặt chẽ tất cả các dữ liệu được nhận từ đối tượng request (Request, Request. QueryString, Request.Form, Request. Cookies, và Request.Server Variables).

Trong trường hợp dữ liệu nhập vào là chuỗi, lỗi xuất phát từ việc có dấu nháy đơn trong dữ liệu. Để tránh điều này, thay thế các dấu nháy đơn bằng hàm Replace để thay thế bằng 2 dấu nháy đơn:

* p\_strUsername = Replace(Request.Form(“txtUsername”), “‘“, “‘’”)
* p\_strPassword = Replace(Request.Form(“txtPassword”), “‘“, “‘’”)
* Thiết lập độ dài tối đa cần thiết: length <= MAX\_LENGTH
* Lọc bỏ ký tự nguy hiểm (bao gồm các meta characters trong DBMS), lọc bỏ các từ khóa như union, select, order, information\_schema, insert, drop, load\_file… (sử dụng các hàm như mysql\_real\_escape\_string(), preg\_replace()…)

Trong trường hợp dữ liệu nhập vào là số, lỗi xuất phát từ việc thay thế một giá trị được tiên đoán là dữ liệu số bằng chuỗi chứa câu lệnh SQL bất hợp pháp. Để tránh điều này, đơn giản hãy kiểm tra dữ liệu có đúng kiểu hay không:

* p\_lngID = CLng(Request(“ID”)) Như vậy, nếu người dùng truyền vào một chuỗi, hàm này sẽ trả về lỗi ngay lập tức.

Sử dụng hàm sprintf và mysql\_real\_escape\_string để xác định kiểu dữ liệu cho câu truy vấn.

* Hàm sprintf gồm có hai tham số trở lên, tham số thứ nhất là chuỗi và trong đó có chứa một đoạn Regex để thay thế, tham số thứ 2 trở đi là các giá trị sẽ được thay thế tương ứng. Giá trị ráp vào sẽ được convert phù hợp rồi mới rap vào.
* Hàm mysql\_real\_escape\_string có nhiệm vụ sẽ chuyển một chuỗi thành chuỗi query an toàn, nên ta sẽ kết hợp nó để gán vào câu truy vấn.

### **Xử lý thông báo lỗi trên hệ thống**

Ngoài ra để tránh các nguy cơ từ tấn công SQL Injection, nên chú ý loại bỏ bất kì thông tin kĩ thuật nào chứa trong thông điệp chuyển tới cho người dùng khi ứng dụng có lỗi. Các thông báo lỗi thông thường tiết lộ các chi tiết kĩ thuật có thể cho phép kẻ tấn công biết được điểm yếu của hệ thống.

1. **Backup dữ liệu**

Phòng trừ trường hợp bị tấn công và mất hoàn toàn CSDL chúng ta nên thường xuyên backup dữ liệu.

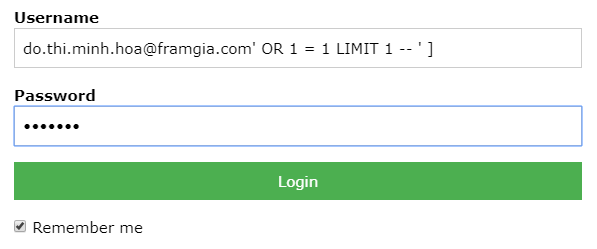
### **Sử dụng các tool có tỉ lệ thành công cao để tấn công**

Hiện nay thì vấn đề app bị tấn công bằng lỗi SQL Injection không còn nhiều như trước đây nữa, tuy nhiên để phòng chống thì chúng ta cũng nên kiểm tra trước khi đưa vào sử dụng bằng cách tấn công bằng các tool có sẵn như: Havij, Kali, SQLmap… Ngoài ra thì việc thường xuyên cập nhật các phương thức tấn công mới cũng sẽ giúp ứng dụng của mình có tính bảo mật cao. Tuy nhiên, kỹ thuật và công nghệ thay đổi từng ngày và thế giới hacker có nhiều cao thủ cao siêu thì rất khó đối phó, nên việc tốt nhất là không nên thách thức hoặc gây thù với bất cứ ai.

1. **Các ví dụ minh họa tình huống sử dụng**

* **Tấn công vượt qua kiểm tra lúc đăng nhập**

Ví dụ như với một form login như dưới đây. Thử submit các field username, password hoặc field id, .. bằng hi' or 1=1. Nếu thành công, thì có thể login vào mà không cần phải biết username và password.



* Nhập vào ô email với nội dụng:do.thi.minh.hoa@framgia.com' OR 1 = 1 LIMIT 1 -- ' ]
* Nhập vào ô password với nội dụng bất kỳ, sau đó click vào nút Login.

Chúng ta có thể đăng nhập thành công. Nguyên nhân chúng ta có thể dễ dàng đăng nhập vào hệ thống mà thậm chí chúng ta không có thông tin đăng nhập là vì khi đăng nhập hệ thống, chương trình sẽ tự động tạo ra cậu lệnh truy vấn dữ liệu với tham số được truyền vào từ form login

SELECT \* FROM users WHERE email = $\_POST['email'] AND password = md5($\_POST['password']);

Với dữ liệu chúng ta vừa nhập, câu lệnh truy vấn sẽ là:

SELECT \* FROM users WHERE email = 'do.thi.minh.hoa@framgia.com' OR 1 = 1 LIMIT 1 -- ' ]' AND password = '123lala';

Sau kí tự “–“, hệ thống sẽ hiểu đó là comment, chương trình sẽ không thực thi những gì phía sau nó. Câu truy vấn có thể rút gọn lại:

SELECT \* FROM leo.users WHERE email = 'do.thi.minh.hoa@framgia.com' OR 1 = 1 LIMIT 1

Điều kiện của câu lệnh truy vấn: email = 'do.thi.minh.hoa@framgia.com' OR 1 = 1, nó luôn là true, đó là lý do vì sao chúng ta có thể vượt qua màn hình đăng nhập một cách dễ dàng.

* **Dạng tấn công sử dụng cân lệnh select**

Chúng ta có tể gặp các website tin tức có các tham số được truyền vào như: index.php?id=21 , index.asp?id=6,.. thì chúng ta chỉ cần thêm các kí tự dấu nháy đơn ('), dấu nháy kép ("), phần trăm 27 (%27), 00 phần trăm (00%) vào sau id để kiểm tra nếu nó hiện ra lỗi hoặc không hiện ra 1 cái gì (khác hoàn toàn với lúc ban đầu thì 90% có khả năng khai thác được lỗi này)

Ví dụ <http://www.ampak.com.tw/product.php?id=22> là một website bị lỗi SQL Injection thì chúng ta sẽ thử với <http://www.ampak.com.tw/product.php?id=22>' hoặc [http://www.ampak.com.tw/product.php?id=22'](http://www.ampak.com.tw/product.php?id=22%27) thì đều trả về trang lỗi Trong các tình huống thông thường, URL hiển thị nội dung của tin có ID trùng với ID đã chỉ định và hầu như không thấy có lỗi. Tuy nhiên, giống như ví dụ đăng nhập ở trước, đoạn mã này để lộ sơ hở cho một lỗi SQL injection khác. Kẻ tấn công có thể thay thế một ID hợp lệ bằng cách gán ID cho một giá trị khác, và từ đó bắt đầu cho một cuộc tấn công bất hợp pháp, ví dụ như: 0 OR 1=1 (nghĩa là <http://www.ampak.com.tw/product.php?id=0> or 1=1) Câu truy vấn SQL lúc này sẽ không trả về trang lỗi mặc dù chúng ta không có sản phẩm nào có ID = 0, vì nó sẽ thực hiện câu lệnh:

SELECT \* FROM Products WHERE Product\_ID=0 or 1=1

Tương tự như trên, kẻ tấn công có thể lợi dụng sơ hở trong câu truy vấn SQL để nhập thêm lệnh tiếp theo: '; DROP TABLE users -- Lúc này, chương trình sẽ chạy câu truy vấn thực hiện việc xóa bảng users.

* **Dạng tấn công sử dụng câu lệnh insert**

 Một câu lệnh INSERT có thể có cú pháp dạng: INSERT INTO TableName VALUES('1st Value', '2nd Value', '3rd Value') Thì chắc chắn sẽ bị lỗi Sql Injection, bởi vì nếu ta nhập vào trường thứ nhất ví dụ như: ' + (SELECT TOP 1 FieldName FROM TableName) + ' Lúc này câu truy vấn sẽ là:

INSERT INTO TableName VALUES(' ' + (SELECT TOP 1 FieldName FROM TableName) + ' ', 'abc', 'def')

Khi đó, lúc thực hiện lệnh xem thông tin, xem như chúng ta đã yêu cầu thực hiện thêm một lệnh nữa đó là:

SELECT TOP 1 FieldName FROM TableName

## Một số tool để khai thác

* + Havij: Havij là tool giúp tester thâm nhập để tìm kiếm và khai thác lỗ hổng SQL Injection trên một trang web. Người dùng có thể thực hiện các phương pháp khai thác lỗ hổng SQL Injection một cách dễ dàng qua giao diện. Havij có thể lấy tên đăng nhập CSDL, mật khẩu mã Hash, dump bảng và cột, trích xuất dữ liệu từ cơ sở dữ liệu, thực hiện các câu lệnh SQL tấn công máy chủ của nạn nhân, thậm chí truy cập vào các tập tin quan trọng của hệ thống và thực hiện các lệnh tương tác với hệ điều hành (shell-chạy command line). Sức mạnh đặc biệt của Havij để phân biệt nó với các công cụ tương tự là nhờ phương thức Injection độc đáo. Tỉ lệ thành công của các cuộc tấn công nhờ Havij lên tới 95%. Với giao diện thân thiện và thiết đặt hệ thống phát hiện lỗ hổng tự động khiến nó trở nên dễ dàng cho tất cả người dùng.
  + Sqlmap: SQLMAP là công cụ khai thác những lổ hỏng của cơ sở dữ liệu SQL. Công cụ này được xem là công cụ khai thác SQL tốt nhất hiện nay. Được giới bảo mật và giới hacker sử dụng thường xuyên. Với người dùng Kali hoặc Back Track 5 thì SQLMAP đã được tích hợp sẵn vào hệ điều hành. Riêng Windows thì chúng ta phải cài đặt thêm python và SQLMAP để sử dụng. Đây là công cụ mã nguồn mở, tự động hóa quá trình phát hiện và khai thác lỗ hổng SQL. Nó đi kèm với một công cụ phát hiện mạnh mẽ, nhiều tính năng thích hợp cho trình kiểm tra thâm nhập cuối cùng