TRƯỜNG ĐẠI HỌC ĐÀ LẠT

**KHOA CÔNG NGHỆ THÔNG TIN**

**🙠🕮🙢**



**BÁO CÁO HỌC PHẦN AN TOÀN VÀ BẢO MẬT HỆ THỐNG**

**ĐỀ TÀI:**

**“Tấn công SQL Injection”**

**GVHD: ThS. PHAN THỊ THANH NGA**

**Sinh viên thực hiện: Mssv:**

**Nguyễn Minh Cường 1812733**

**Nguyễn Thị Hà 1812751**

**Nguyễn Trọng Hiếu 1812756**

**Nguyễn Ngọc Quang 1812832**

***Đà Lạt, tháng 03 năm 2021***

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

Đà Lạt, ngày tháng năm

Giáo viên hướng dẫn

[Ký tên và ghi rõ họ tên]

**MỤC LỤC**

[**1.** **Giới thiệu tổng quan** 4](#_Toc66623423)

[**2.** **Các kỹ thuật tấn công** 5](#_Toc66623424)

[**2.1 Tấn công trong băng** 5](#_Toc66623425)

[2.1.1 Tiêm nhiễm SQL dựa trên thông báo lỗi 5](#_Toc66623426)

[2.1.2 Tiêm nhiễm SQL dựa trên toán tử UNION 6](#_Toc66623427)

[**2.2 Tấn công suy luận** 6](#_Toc66623428)

[2.2.1 Truy vấn bất hợp pháp/không đúng logic 6](#_Toc66623429)

[2.2.2 Tiêm SQL mù 7](#_Toc66623430)

[**2.3 Tấn công ngoài băng** 7](#_Toc66623431)

[**3.** **Demo thực tế** 7](#_Toc66623432)

[**4.** **Biện pháp phòng chống** 10](#_Toc66623433)

[**4.1 Kỹ thuật mã hóa phòng vệ thủ công** 10](#_Toc66623434)

[**4.2 Sử dụng các tham số trong truy vấn** 10](#_Toc66623435)

[**5.** **So sánh tỉ lệ tấn công SQLi qua các năm** 10](#_Toc66623436)

[**6.** **Tài liệu tham khảo** 11](#_Toc66623437)

# **Giới thiệu tổng quan**

SQL viết tắt của Structural Query Language, là ngôn ngữ truy vấn có cấu trúc, được chuẩn hóa và sử dụng để định nghĩa lược đồ cơ sở dữ liệu, thao tác và truy vấn dữ liệu trong cơ sở dữ liệu quan hệ.

Các câu lệnh SQL có thể được sử dụng để tạo các bảng, chèn và xóa dữ liệu trong các bảng, tạo các khung nhìn và truy xuất dữ liệu bằng các câu lệnh truy vấn.

SQL Injection là 1 kĩ thuật tấn công phổ biến và phức tạp. Đối tượng được nhắm đến là các ứng dụng web, chương trình máy tính và cả cơ sở dữ liệu. Kĩ thuật này yêu cầu người thực hiện phải am hiểu sâu sắc về luồng xử lý của 1 ứng dụng web và các thành phần cấu thành nó, bao gồm cơ sở dữ liệu và SQL. Về cơ bản, kĩ thuật tấn công này tiêm nhiễm 1 đoạn mã hoặc 1 đoạn kịch bản nguy hiểm nhằm khai thác lỗ hổng bảo mật xảy ra trong tầng cơ sở dữ liệu của một ứng dụng (chẳng hạn như các truy vấn). Bằng cách tiêm nhiễm vào lệnh SQL, kẻ tấn công có thể trích xuất hoặc thao tác dữ liệu của ứng dụng web.

Tấn công dạng tiêm nhiễm lệnh SQL (SQL Injection – SQLi) là một trong những mối đe dọa bảo mật mạng phổ biến và nguy hiểm nhất. Một số báo cáo về phương thức tấn công này đã được tổng hợp, có thể kể đến như sau:

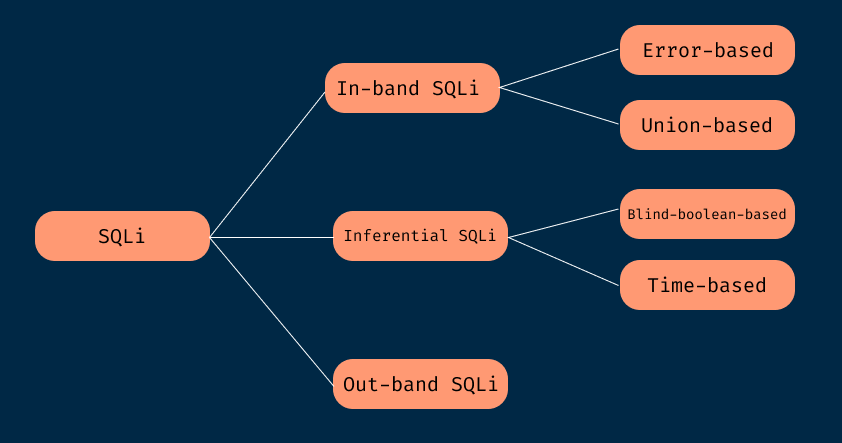
Báo cáo tấn công vào ứng dụng web của Imperva tháng 7 năm 2013 đã khảo sát một loạt các máy chủ ứng dụng Web chuyên dụng và theo dõi tám loại tấn công phổ biến khác nhau. Báo cáo cho thấy các cuộc tấn công SQLi xếp thứ nhất hoặc thứ hai theo các tiêu chí: tổng số các sự cố bị tấn công, số lượng yêu cầu tấn công với mỗi sự cố tấn công và số ngày trung bình trên mỗi tháng mà một ứng dụng trải qua ít nhất một sự cố tấn công. Imperva đã quan sát một trang web đặc biệt nhận được 94.057 yêu cầu tấn công SQL injection chỉ trong một ngày.

Báo cáo Dự án Bảo mật Ứng dụng Web Mở 2013 về 10 rủi ro bảo mật ứng dụng Web quan trọng nhất đã liệt kê các cuộc tấn công tiêm chích, đặc biệt là các cuộc tấn công SQLi, là rủi ro hàng đầu. Bảng xếp hạng này không thay đổi so với báo cáo năm 2010.

Hay, báo cáo bảo mật phần mềm Veracode 2016 cho thấy tỷ lệ phần trăm các ứng dụng bị ảnh hưởng bởi các cuộc tấn công SQLi là khoảng 35%.

# **Các kỹ thuật tấn công**

Có nhiều kỹ thuật tấn công SQL Injection, có thể phân thành 3 nhóm chính: Tấn công trong băng (In-band SQLi), Tấn công suy luận (Inferential SQLi), Tấn công ngoài băng (Out-of-band SQLi).



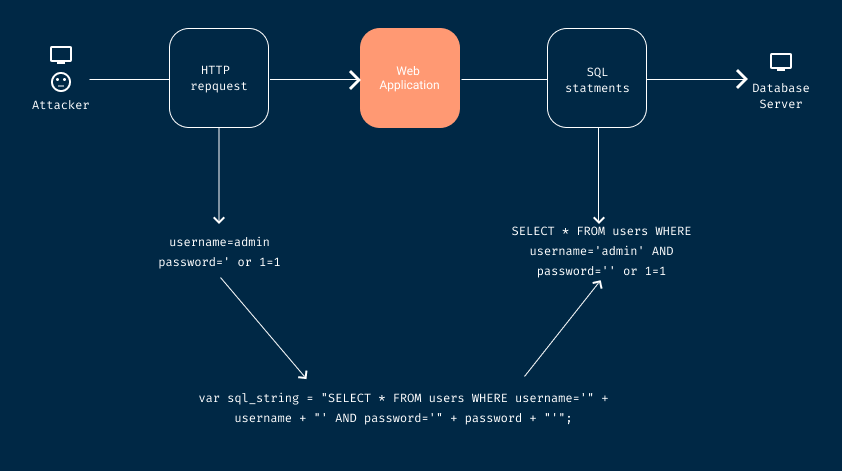
## **2.1 Tấn công trong băng**

Tiêm nhiễm SQL trong băng là 1 nhóm bao gồm nhiều phương thức tấn công. Trong đó, các kĩ thuật tiêm nhiễm SQL sử dụng cùng kênh giao tiếp để phát động tấn công và thu thập thông tin qua các phản hồi. Kĩ thuật tiêm nhiễm trong băng bao gồm các phương thức tấn công: Tiêm nhiễm SQL dựa trên thông báo lỗi và Tiêm nhiễm SQL dựa trên toán tử UNION.

### 2.1.1 Tiêm nhiễm SQL dựa trên thông báo lỗi

Kẻ tấn công dựa vào thông báo lỗi phản hồi từ máy chủ cơ sở dữ liệu, từ đó thu thập thông tin và định hình nên cấu trúc của cơ sử dữ liệu lưu trữ. Trong quá trình phát triển sản phẩm, những thông báo lỗi giúp ích rất nhiều cho các nhà phát triển trong việc tìm và xử lý lỗi. Tuy nhiên, khi sản phẩm đã được hoàn thiện và đưa vào sử dụng thực tế, những thông báo lỗi này lại tiềm ẩn nhiều nguy cơ an toàn bảo mật. Kĩ thuật tiêm nhiễm SQL dựa trên thông báo lỗi lại bao gồm nhiều kĩ thuật con, có thể kể đến như: Chú thích ở cuối dòng (End of line comment) và Hằng đúng (Tautology).

Với kĩ thuật hằng đúng, hình thức tấn công này tiêm mã vào một hoặc nhiều câu lệnh điều kiện để chúng luôn luôn đánh giá là đúng.



Với kĩ thuật chú thích ở cuối dòng, sau khi đưa mã vào một trường cụ thể, mã hợp pháp theo sau sẽ bị vô hiệu hóa thông qua việc sử dụng các ký tự chú thích ở cuối dòng. Một ví dụ sẽ là thêm vào -- sau các đầu vào để phần còn lại của truy vấn không được coi là mã thực thi mà là các chú thích.

### 2.1.2 Tiêm nhiễm SQL dựa trên toán tử UNION

Kỹ thuật này sử dụng toán tử UNION trong SQL để gộp kết quả chung từ 2 hoặc nhiều câu lệnh truy vấn SELECT.

SELECT <column\_name(s)> FROM <table\_1>

UNION

SELECT <column\_name(s)> FROM <table\_2>;

## **2.2 Tấn công suy luận**

Với kĩ thuật tấn công này, thực tế không có việc truyền dữ liệu từ ứng dụng web. Do đó, hacker không thể xem kết quả trực tiếp từ cuộc tấn công, chính vì thế mà nó được gọi với tên khác là "Kĩ thuật tiêm nhiễm mù". Tuy nhiên, kẻ tấn công vẫn có thể xây dựng lại thông tin bằng cách gửi các yêu cầu cụ thể và quan sát hành vi phản hồi của máy chủ trang web/cơ sở dữ liệu. Các kiểu tấn công suy diễn có thể kể đến như: Truy vấn bất hợp pháp/không đúng logic và Tiêm SQL mù.

### 2.2.1 Truy vấn bất hợp pháp/không đúng logic

Cuộc tấn công này cho phép kẻ tấn công thu thập thông tin quan trọng về loại và cấu trúc của cơ sở dữ liệu đằng sau ứng dụng Web. Cuộc tấn công được coi là bước đầu tiên, thu thập thông tin cho các cuộc tấn công khác. Lỗ hổng được khai thác bởi cuộc tấn công này là trang lỗi mặc định được trả về bởi các máy chủ ứng dụng thường được mô tả quá chi tiết. Trong thực tế, một thông báo lỗi được tạo ra thường có thể tiết lộ các thông số dễ bị tổn thương, có thể bị lợi dụng bởi kẻ tấn công.

### 2.2.2 Tiêm SQL mù

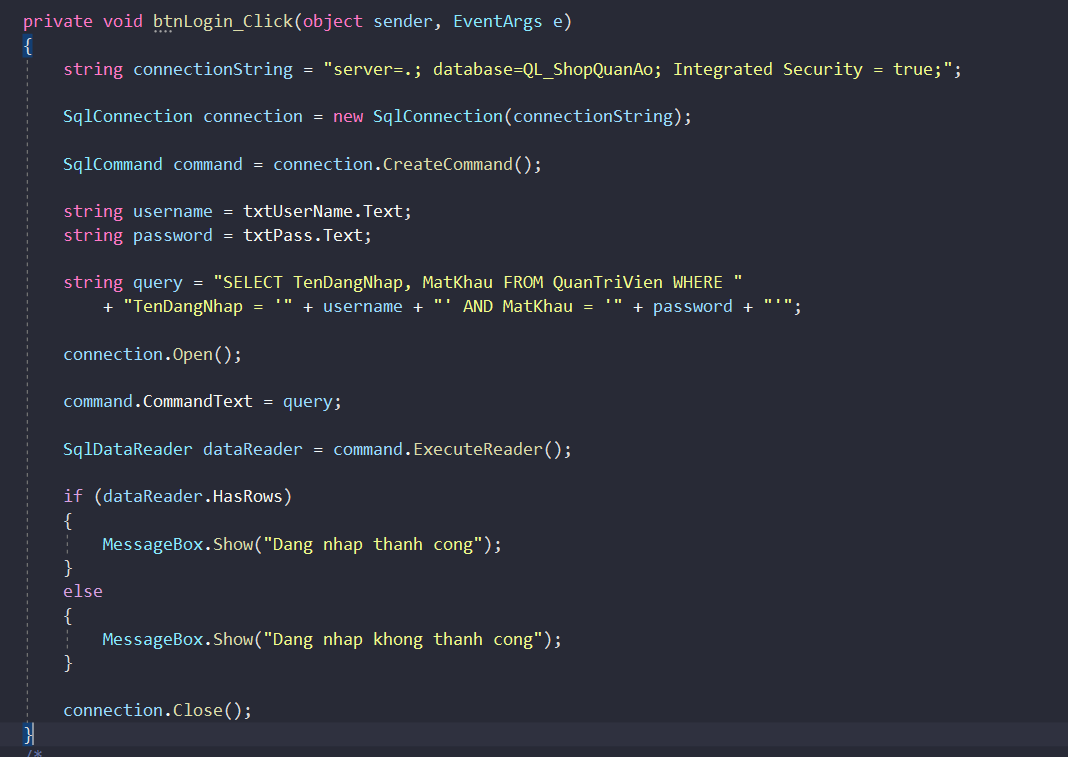
Tiêm SQL mù cho phép kẻ tấn công suy ra dữ liệu có trong hệ thống cơ sở dữ liệu ngay cả khi hệ thống đủ an toàn để không hiển thị bất kỳ thông tin sai sót nào cho kẻ tấn công. Kẻ tấn công hỏi máy chủ câu hỏi dạng đúng/sai. Nếu câu lệnh được đánh giá là đúng, trang web sẽ tiếp tục hoạt động bình thường. Nếu câu lệnh là sai, mặc dù không có thông báo lỗi mô tả, trang web sẽ khác đáng kể so với trang hoạt động bình thường.

## **2.3 Tấn công ngoài băng**

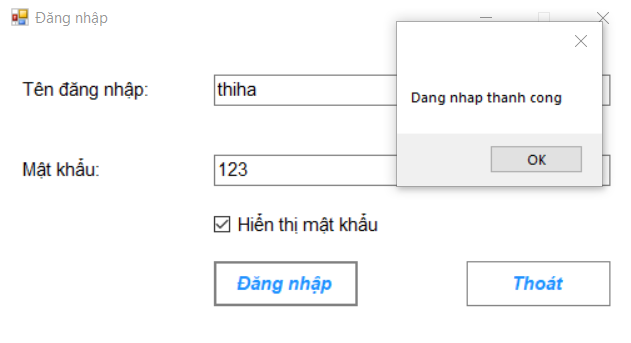
Đây là kĩ thuật sử dụng các kênh khác nhau để tiến hành cuộc tấn công và thu thập các phản hồi. Để có thể tấn công, kĩ thuật này yêu cầu 1 số tính năng phải được kích hoạt sẵn, ví dụ như DNS hoặc yêu cầu HTTP đến máy chủ cơ sở dữ liệu. Chính vì phải thõa mãn các yêu cầu trước khi tấn công nên kĩ thuật này không được phổ biến.

# **Demo thực tế**

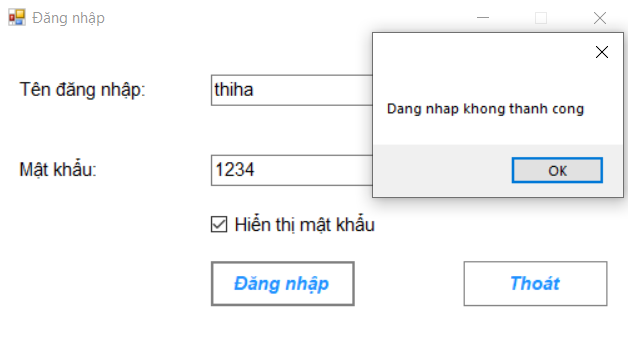
Dưới đây là đoạn chương trình mô phỏng quá trình đăng nhập người dùng vào ứng dụng. Trong hàm này, chương trình thực hiện kết nối đến cơ sở dữ liệu để tiến hành câu lệnh truy vấn tìm kiếm dữ liệu người dùng. Thông tin về tên đăng nhập và mật khẩu người dùng được lấy trực tiếp vào ứng dụng, không có bước kiểm tra dữ liệu đầu vào. Về phần truy vấn, câu lệnh được thực hiện thông qua việc nối chuỗi thủ công. Đây là những rủi ro tìm ẩn nguy cơ tấn công SQL Injection.



Trong cơ sở dữ liệu tồn tại dữ liệu người dùng có tên đăng nhập “thiha” và mật khẩu là “123”. Hình dưới đây minh họa quá trình đăng nhập thành công.



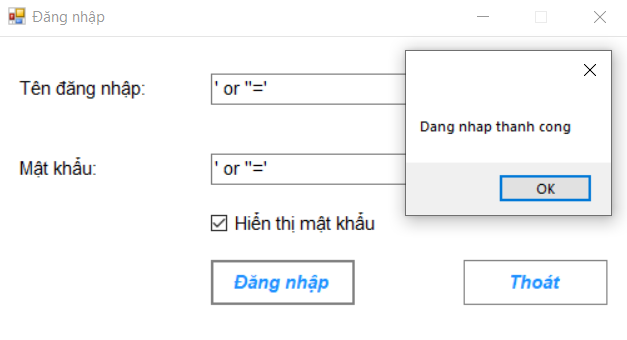
Hình dưới đây minh họa quy trình đăng nhập không thành công, do người dùng nhập không đúng mật khẩu.



Hình dưới đây minh họa quá trình tấn công SQL Injection, cụ thể dùng kỹ thuật hằng đúng (*tautology*). Người tấn công biết tên đăng nhập người dùng là “thiha”, nhưng không biết mật khẩu. Vì vậy, anh ta thử thay thế mật khẩu bằng 1 đoạn mã SQL để thực hiện tấn công SQL Injection. Chương trình không được lập trình để phòng chống kỹ thuật tấn công này. Chính vì vậy, người tấn công dễ dàng vượt qua bước xác thực người dùng.



Hình dưới đây minh họa 1 cách tấn công khác. Người tấn công không cần biết cả tên đăng nhập và mật khẩu người dùng. Anh ta cố ý chèn 1 đoạn mã SQL vào khung tên đăng nhập và mật khẩu. Cuộc tấn công SQL Injection thành công và anh ta dễ dàng vượt qua bước xác thực người dùng. Trong trường hợp này, chương trình sẽ trả về mọi hàng trong bảng.



# **Biện pháp phòng chống**

Bởi vì các cuộc tấn công SQLi rất phổ biến, gây tổn hại và thường xuyên thay đổi cả về phương thức lẫn kiểu tấn công, một biện pháp đối phó là không đủ. Thay vào đó, cần có một bộ kỹ thuật tích hợp. Một số biện pháp phòng chống có thể kể đến như: Kỹ thuật mã hóa phòng vệ thủ công, Sử dụng các tham số trong truy vấn và SQL DOM.

## **4.1 Kỹ thuật mã hóa phòng vệ thủ công**

Một lỗ hổng phổ biến được khai thác bởi các cuộc tấn công SQLi là không xác thực đầu vào. Giải pháp đơn giản để loại bỏ các lỗ hổng này là áp dụng các kỹ thuật mã hóa phòng thủ phù hợp. Một ví dụ là kiểm tra kiểu đầu vào có phải là chuỗi chỉ chứa các ký số hay không. Loại kỹ thuật này có thể tránh các cuộc tấn công dựa trên các lỗi ràng buộc trong hệ quản trị cơ sở dữ liệu. Một kỹ thuật khác là thực hiện so khớp mẫu để cố gắng phân biệt đầu vào bình thường với đầu vào bất thường.

## **4.2 Sử dụng các tham số trong truy vấn**

Cách tiếp cận này cố gắng ngăn SQLi bằng cách cho phép nhà phát triển ứng dụng xác định chính xác hơn cấu trúc của truy vấn SQL và truyền các tham số giá trị cho nó một cách riêng biệt sao cho mọi đầu vào của người dùng không được phép sửa đổi cấu trúc truy vấn.

# **So sánh tỉ lệ tấn công SQLi qua các năm**

Theo số liệu tham khảo từ trang *“hackmageddon”*, chúng tôi thực hiện vẽ biểu đồ so sánh tỉ lệ các cuộc tấn công SQLi so với các cuộc tấn công khác qua từng năm. Số liệu chi tiết bao gồm: tháng 8 năm 2012: 32.4%, tháng 8 năm 2013: 15.9%, tháng 8 năm 2014: 9.9% và gần nhất, tháng 8 năm 2020 là 0.5%.



Biểu đồ này thể hiện rõ xu hướng giảm dần các cuộc tấn công SQLi, cho thấy phương pháp tấn công này gần như đã được khắc phục triệt để. Bên cạnh đó, cho thấy sự phát triển của các công nghệ tấn công và xâm nhập hệ thống. Vì vậy, chúng ta luôn luôn phải cảnh giác trước các mối nguy đe dọa an toàn bảo mật hệ thống.

# **Tài liệu tham khảo**

[1] Sách *“CEH v10 Complete Training Guide With Labs”*

[2] Sách *“Computer Security Principles and Practice (Fourth edition)”*