

## MỤC LỤC

DANH MỤC CHỮ VIẾT TẮT .....	3
DANH MỤC HÌNH VẼ .....	4
MỞ ĐẦU .....	5
CHƯƠNG I. TỔNG QUAN VỀ TẤN CÔNG CHÈN MÃ SQL VÀ CÁC BIỆN PHÁP PHÒNG CHỐNG .....	7
1.1 Khái quát về tấn công chèn mã SQL .....	7
1.1.1 Giới thiệu tấn công chèn mã SQL.....	7
1.1.2 Cơ chế tấn công chèn mã SQL .....	7
1.1.3 Các dạng tấn công chèn mã SQL.....	7
1.2 Các biện pháp phòng chống tấn công chèn mã SQL .....	8
1.2.1 Các biện pháp phòng chống ở mức lập trình .....	8
1.2.2 Các biện pháp phòng chống ở mức độ nền tảng.....	8
1.3 Kết chương .....	8
CHƯƠNG II. PHÁT HIỆN TẤN CÔNG CHÈN MÃ SQL DỰA TRÊN PHÂN TÍCH CÚ PHÁP CÂU LỆNH.....	9
2.1 Khái quát về ngôn ngữ SQL và cú pháp câu lệnh SQL.....	9
2.1.1 Giới thiệu ngôn ngữ SQL .....	9
2.1.2 Cú pháp cơ bản các câu lệnh SQL .....	9
2.2 Phát hiện tấn công chèn mã SQL dựa trên phân tích cú pháp câu lệnh .....	10

2.2.1 Xây dựng các đặc tả câu lệnh SQL hợp lệ.....	10
2.2.2 Bắt câu lệnh SQL từ ứng dụng .....	11
2.2.3 Phân tích cú pháp câu lệnh .....	12
2.3 Kết chương .....	15
CHƯƠNG III. THỬ NGHIỆM VÀ ĐÁNH GIÁ .....	16
3.1 Xây dựng mô hình thử nghiệm.....	16
3.1.1 Kiến trúc bộ lọc câu lệnh SQL - Database Filter .....	16
3.1.2 Cài đặt bộ lọc cơ sở dữ liệu .....	17
3.1.3 Giới thiệu một số mô đun chương trình.....	17
3.2 Một số kết quả .....	17
3.2.1 Khởi tạo hệ thống.....	17
3.2.2 Kịch bản thử nghiệm tấn công chèn mã.....	18
3.3 Nhận xét và đề xuất .....	21
3.3.1 Nhận xét.....	21
3.3.2 Đề xuất.....	21
3.4 Kết chương .....	22
KẾT LUẬN .....	23
TÀI LIỆU THAM KHẢO .....	25

**DANH MỤC CHỮ VIẾT TẮT**

<b>STT</b>	<b>Từ viết tắt</b>	<b>Từ đầy đủ</b>
1	SQL	Structured Query Language
2	OWASP	The Open Web Application Security Project
3	OS	Operation System
4	LDAP	The Lightweight Directory Access Protocol
5	HTTP	The Hypertext Transfer Protocol
6	JDBC	Java Database Connectivity
7	PDO	PHP Data Objects
8	API	Application Program Interface
9	WAF	Web Application Firewall
10	IDS	Intrusion Detection System
11	DML	Data Manipulation Language
12	DDL	Data Definition Language

## **DANH MỤC HÌNH VẼ**

Hình 2.1: Minh họa đặc tả câu lệnh.....	10
Hình 2.2: Sơ đồ bố trí bộ lọc CSDL .....	11
Hình 2.3: Ví dụ phân tích câu truy vấn.....	12
Hình 2.4: Mô hình hoạt động phân tích cấu trúc truy vấn.....	12
Hình 2.5: Mô hình phát hiện tấn công chèn mã SQL dựa trên phân tích cú pháp câu lệnh. ....	14
Hình 3.1: Mô hình bộ lọc cơ sở dữ liệu.....	16
Hình 3.2: Dòng lệnh thực hiện chạy ứng dụng bộ lọc cơ sở dữ liệu	17
Hình 3.3: Kết quả phân tích tệp dữ liệu câu truy vấn đầu vào.....	18
Hình 3.4: Thông tin bộ lọc cơ sở dữ liệu thu được.....	18
Hình 3.5: Thông tin thu lại khi tấn công chèn mã .....	19
Hình 3.6: Thông báo lỗi cho người dùng.....	19

## MỞ ĐẦU

Tấn công chèn mã nói chung và tấn công chèn mã SQL (SQL injection attacks) nói riêng là các dạng tấn công phổ biến lên các máy chủ và các ứng dụng, gây nhiều hậu quả nghiêm trọng. Tấn công chèn mã SQL là dạng tấn công chủ yếu được thực hiện trên các website, trong đó tin tặc nhúng mã độc SQL vào dữ liệu người dùng, gửi đến máy chủ web và được thực hiện trên máy chủ cơ sở dữ liệu (CSDL) của ứng dụng web. Tùy theo mức độ tinh vi của mã độc SQL, tấn công chèn mã SQL có thể cho phép tin tặc vượt qua khâu xác thực người dùng, chèn, sửa, xóa dữ liệu trong các bảng dữ liệu của CSDL, đánh cắp thông tin trong CSDL, hoặc thậm chí có thể chiếm quyền điều khiển cả hệ thống chạy máy chủ CSDL.

Có hai nguyên nhân chính của tấn công chèn mã SQL: (1) dữ liệu từ người dùng hoặc các nguồn khác không được người lập trình kiểm tra, hoặc kiểm tra không đầy đủ, và (2) các trang web sử dụng các câu truy vấn động, trong đó, mã lệnh SQL gốc được ghép với dữ liệu từ người dùng để tạo câu truy vấn gửi đến thực hiện trên máy chủ CSDL. Do tính chất nghiêm trọng của tấn công chèn mã SQL, nhiều biện pháp phòng chống dạng tấn công này đã được nghiên cứu, đề xuất. Có thể chia các biện pháp phòng chống tấn công chèn mã SQL thành 2 hướng chính: (1) các hướng thực hiện ở mức lập trình và (2) các hướng ở mức nền tảng. Các hướng ở mức lập trình yêu cầu trực tiếp thực hiện sửa mã của website, mã SQL của ứng dụng để lọc dữ liệu, để loại bỏ mã độc SQL. Các hướng ở mức nền tảng thường không yêu cầu trực tiếp sửa mã của website, hoặc mã SQL của ứng

dụng, mà thường sử dụng các công cụ của hệ thống, hoặc cài đặt các công cụ bổ sung để lọc, phát hiện nguy cơ tấn công chèn mã trong các câu lệnh SQL gửi từ máy chủ web đến máy chủ CSDL. Đề tài nghiên cứu thực hiện trong luận văn này theo hướng thứ 2, là hướng độc lập với ứng dụng web – không đòi hỏi sửa mã trang web và CSDL.

Luận văn gồm 3 chương chính sau:

**Chương 1 – Tổng quan về tấn công chèn mã SQL và các biện pháp phòng chống:** giới thiệu khái quát về mục đích, cơ chế và mô tả chi tiết về các kiểu tấn công chèn mã SQL. Bên cạnh đó nêu ra các nguy cơ, hậu quả cho máy chủ web khi bị tấn công chèn mã SQL. Cuối chương sẽ trình bày các biện pháp phòng chống tấn công chèn mã SQL theo hai hướng cụ thể nhằm giải quyết bài toán ngăn chặn tấn công SQL Injection.

**Chương 2 – Phát hiện tấn công chèn mã SQL dựa trên phân tích cú pháp câu lệnh:** Nội dung chương này giới thiệu khái quát về ngôn ngữ SQL và nghiên cứu chi tiết về cú pháp cơ bản của các câu lệnh SQL. Từ đó xây dựng các đặc tả câu lệnh SQL hợp lệ nhằm phục vụ việc phân tích cú pháp câu lệnh SQL. Bên cạnh đó chương 2 cũng đưa ra giải pháp xây dựng bộ lọc có nhiệm vụ bắt các câu lệnh SQL đi vào máy chủ web.

**Chương 3 – Thử nghiệm và đánh giá:** Chương này tập trung vào việc áp dụng giải pháp được đề xuất ở chương hai để xây dựng ứng dụng database filter nhằm chống lại tấn công chèn mã. Dựa trên kết quả thử nghiệm để đưa ra đánh giá, nhận xét.

# **CHƯƠNG I. TỔNG QUAN VỀ TẤN CÔNG CHÈN MÃ SQL VÀ CÁC BIỆN PHÁP PHÒNG CHỐNG**

## **1.1 Khái quát về tấn công chèn mã SQL**

### ***1.1.1 Giới thiệu tấn công chèn mã SQL***

#### ***1.1.1.1 SQL Injection là gì***

SQL injection là một kỹ thuật cho phép những kẻ tấn công lợi dụng lỗ hổng trong việc kiểm tra dữ liệu nhập trong các ứng dụng web và các thông báo lỗi của hệ quản trị cơ sở dữ liệu để "tiêm vào" (inject) và thi hành các câu lệnh SQL bất hợp pháp (không được người phát triển ứng dụng lường trước). Hậu quả của nó rất tai hại vì nó cho phép những kẻ tấn công có thể thực hiện các thao tác: vượt qua khâu xác thực, đánh cắp thông tin trong cơ sở dữ liệu, chèn, xóa, sửa đổi dữ liệu bên trong cơ sở dữ liệu, chiếm quyền điều khiển của hệ thống.

#### ***1.1.1.2 Mục đích tấn công SQL Injection***

#### ***1.1.1.3 Hậu quả của tấn công SQL Injection***

### ***1.1.2 Cơ chế tấn công chèn mã SQL***

#### ***1.1.3 Các dạng tấn công chèn mã SQL***

##### ***1.1.3.1 Tấn công vượt qua kiểm tra đăng nhập***

##### ***1.1.3.2 Tấn công sử dụng câu lệnh SELECT***

*1.1.3.3 Tấn công khai thác dữ liệu thông qua toán tử UNION*

*1.1.3.4 Tấn công truy vấn ngược*

*1.1.3.5 Tấn công suy diễn thông tin*

*1.1.3.6 Tấn công mã hóa thay thế*

## **1.2 Các biện pháp phòng chống tấn công chèn mã SQL**

### ***1.2.1 Các biện pháp phòng chống ở mức lập trình***

*1.2.1.1 Làm sạch dữ liệu đầu vào*

*1.2.1.2 Xây dựng truy vấn theo mô hình tham số hóa*

*1.2.1.3 Chuẩn hóa dữ liệu*

*1.2.1.4 Mô hình thiết kế mã nguồn tổng quát*

### ***1.2.2 Các biện pháp phòng chống ở mức độ nền tảng***

## **1.3 Kết chương**

Chương I đã giới thiệu tổng quan về mục đích, cơ chế và mô tả chi tiết về các kiểu tấn công chèn mã SQL vào máy chủ ứng dụng web. Chỉ ra được những hậu quả nghiêm trọng phải gánh chịu khi tấn công SQL thành công. Bên cạnh đó chương này đã trình bày khái quát các kỹ thuật phát hiện và ngăn chặn tấn công chèn mã SQL từ mức lập trình đến mức nền tảng làm cơ sở xây dựng phương án phù hợp để giải quyết bài toán ngăn chặn tấn công SQL Injection.



## **CHƯƠNG II. PHÁT HIỆN TẤN CÔNG CHÈN MÃ SQL DỰA TRÊN PHÂN TÍCH CÚ PHÁP CÂU LỆNH**

### **2.1 Khái quát về ngôn ngữ SQL và cú pháp câu lệnh SQL**

#### **2.1.1 Giới thiệu ngôn ngữ SQL**

SQL (Structured Query Language) là ngôn ngữ sử dụng để tổ chức, quản lý và truy xuất dữ liệu được lưu trữ trong các cơ sở dữ liệu. SQL là một ngôn ngữ lập trình bao gồm tập các câu lệnh sử dụng để tương tác với cơ sở dữ liệu quan hệ. Các đơn vị điển hình thực hiện của SQL là 'truy vấn', câu lệnh SQL có thể sửa đổi cấu trúc của cơ sở dữ liệu và thao tác các nội dung của cơ sở dữ liệu bằng cách sử dụng DDL (Data Manipulation Language) khác nhau, DML (Data Definition Language) lệnh tương ứng.

#### **2.1.2 Cú pháp cơ bản các câu lệnh SQL**

##### *2.1.2.1 Cú pháp câu lệnh SELECT*

##### *2.1.2.2 Cú pháp câu lệnh INSERT*

##### *2.1.2.3 Cú pháp câu lệnh UPDATE*

##### *2.1.2.4 Cú pháp câu lệnh DELETE*

##### *2.1.2.5 Cú pháp câu lệnh CREATE*

##### *2.1.2.6 Cú pháp câu lệnh ALTER*

##### *2.1.2.7 Cú pháp câu lệnh DROP*

##### *2.1.2.8 Khung nhìn VIEW*

## 2.2 Phát hiện tấn công chèn mã SQL dựa trên phân tích cú pháp câu lệnh

### 2.2.1 Xây dựng các đặc tả câu lệnh SQL hợp lệ

Đặc tả các câu lệnh SQL hợp lệ là một trong các thành phần quan trọng trong việc đánh giá một câu lệnh gửi từ ứng dụng web đến cơ sở dữ liệu là hợp lệ hay không. Trên cơ sở các đặc tả, các luật được định nghĩa để mô tả cấu trúc cú pháp mà một câu lệnh SQL hợp lệ phải tuân theo. Một trong các yêu cầu chính của các đặc tả là tính hoàn thiện, tức là chúng phải bao gồm các luật cho từng lệnh gửi bởi ứng dụng web. Nếu các đặc tả không bao phủ hết các câu lệnh có thể có của ứng dụng web, các lỗi phát hiện (cảnh báo sai dương, hoặc sai âm) có thể xảy ra. Các đặc tả được xây dựng dựa trên ký hiệu Extended Backus-Naur Form (EBNF). Xem xét một ví dụ xây dựng đặc tả một câu lệnh:

```
SELECT user_id, full_name, email FROM tbl_users
WHERE username = 'cuong' AND password = 'abc12345'
```

Hình 2.1 minh họa đặc tả tạo cho câu lệnh trên. Các luật của đặc tả này chỉ ra các trình tự hợp lệ của các thành phần của câu lệnh không có tấn công.

```
<Query specification> :=
  SELECT <Select List> <From Clause> <Where Clause>

<Select List> :=
  <Table Column> (<COMMA> <Table Column>)*

<From Clause> :=
  FROM <Table reference>

<Where Clause> :=
  WHERE <search condition> AND <search condition>

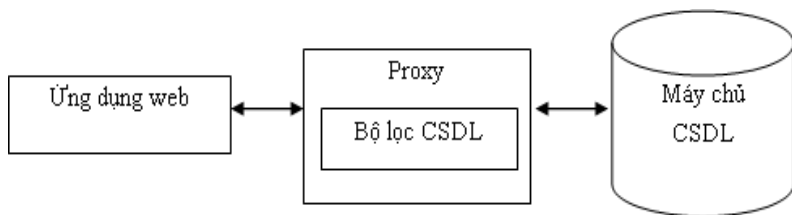
<search condition> :=
  <Table Column> "=" <STRING LITERAL>
```

**Hình 2.1: Minh họa đặc tả câu lệnh**

### 2.2.2 Bắt câu lệnh SQL từ ứng dụng

Thông thường, ứng dụng web được cấu hình để gửi thẳng các câu lệnh SQL sang máy chủ cơ sở dữ liệu để thực hiện.

Với máy chủ cơ sở dữ liệu Microsoft SQL Server, ứng dụng web được cấu hình để kết nối với máy chủ cơ sở dữ liệu sử dụng một chuỗi kết nối (connection string) thông qua cổng TCP 1433. Cổng TCP 1433 là cổng chuẩn của máy chủ cơ sở dữ liệu Microsoft SQL Server. Để bộ lọc cơ sở dữ liệu có thể hoạt động, ứng dụng web cần được cấu hình để kết nối và gửi các câu lệnh SQL đến bộ lọc cơ sở dữ liệu, bộ lọc tiến hành đánh giá câu lệnh nhận được. Nếu câu lệnh SQL được đánh giá là hợp lệ thì nó được chuyển tiếp đến máy chủ cơ sở dữ liệu để thực hiện. Ngược lại, nếu câu lệnh SQL được đánh giá là không hợp lệ thì nó được ghi log và một thông báo lỗi thực hiện được gửi lại cho ứng dụng web. Hình 2.2 dưới đây minh họa giao tiếp giữa Ứng dụng web, Proxy và Máy chủ CSDL, trong đó Proxy là thành phần trung gian có nhiệm vụ bắt các gói tin TCP gửi từ máy chủ web, chuyển thành các câu lệnh SQL, sau đó chuyển cho Bộ lọc CSDL kiểm tra, đánh giá các câu lệnh SQL.

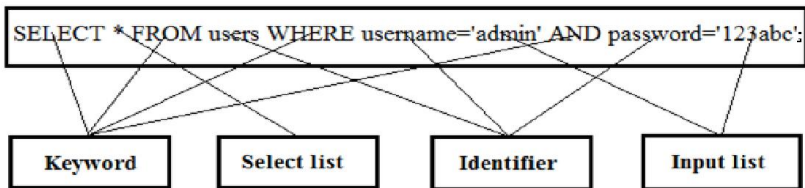


**Hình 2.2: Sơ đồ bố trí bộ lọc CSDL**

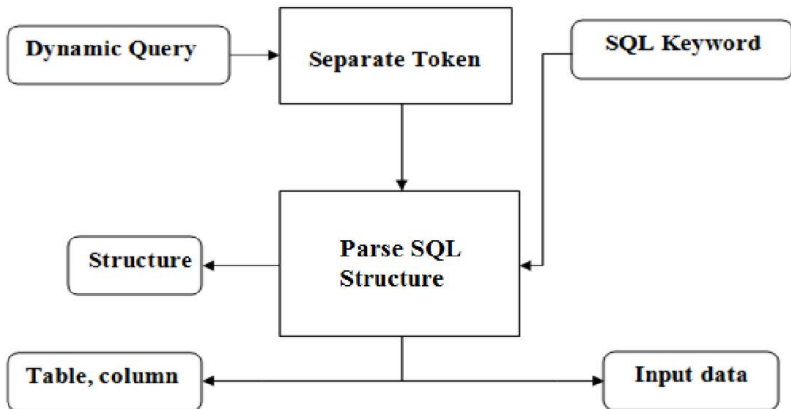
### 2.2.3 Phân tích cú pháp câu lệnh

#### 2.2.3.1 Phân tích cú pháp câu truy vấn

Phân tích cú pháp câu truy vấn là việc phân tích các thành phần đại diện của ngôn ngữ SQL được sử dụng trong câu truy vấn như cấu trúc ngữ pháp từ vựng của ngôn ngữ ( ví dụ như: MS-SQL, MySQL.. ).



Hình 2.3: Ví dụ phân tích câu truy vấn



Hình 2.4: Mô hình hoạt động phân tích cấu trúc truy vấn

#### 2.2.3.2 Các bước tiến hành phân tích cấu trúc cú pháp câu lệnh

Câu truy vấn đầu vào:

```
SELECT * FROM users WHERE username='admin' AND  
password='123abc'
```

### **Bước 1: Tách câu truy vấn**

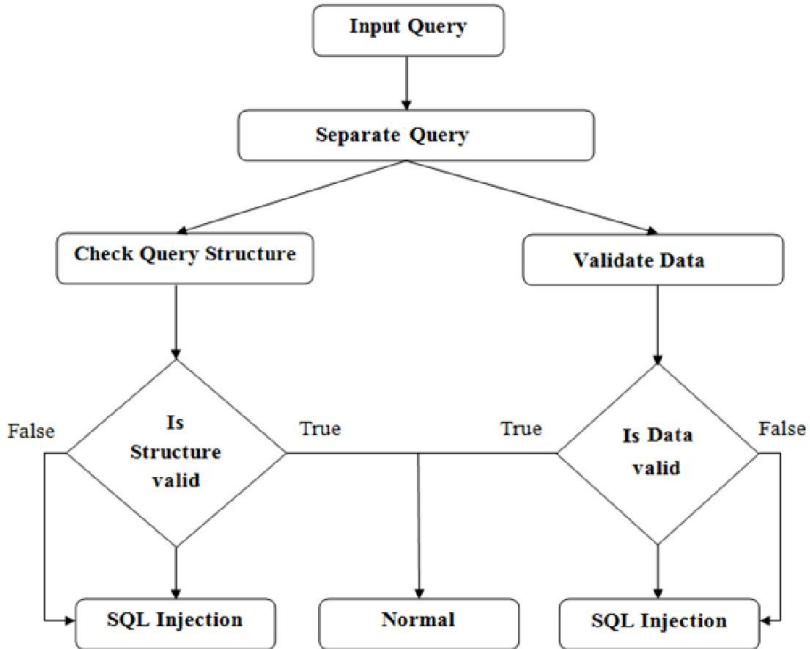
Tách câu truy vấn thành 3 phần: cấu trúc, danh sách các thành phần truy vấn, dữ liệu đầu vào từ người dùng. Dữ liệu bao gồm cấu trúc của câu truy vấn và dữ liệu nhạy cảm trong câu truy vấn. Dữ liệu từ bước này là thành phần đầu vào của bước kiểm tra phát hiện tấn công chèn mã SQL.

### **Bước 2: Kiểm tra cấu trúc và dữ liệu**

#### **1. Kiểm tra cấu trúc:** Hỗ trợ 2 lựa chọn thực hiện

- Lựa chọn 1: Danh sách cấu trúc câu truy vấn SQL sử dụng trong quá trình so sánh cấu trúc câu truy vấn được tạo chủ quan theo hiểu biết và tham khảo về các câu truy vấn thông dụng trong các ứng dụng web. Với sự lựa chọn này quá trình kiểm tra cấu trúc chỉ chuẩn xác ở mức tương đối do tính chất đa dạng của các câu truy vấn.

- Lựa chọn 2: Danh sách cấu trúc câu truy vấn sử dụng trong ứng dụng được nhà phát triển ứng dụng web cung cấp và mô đun phân tích cấu trúc ở bước 1 phân tích thành cấu trúc chuẩn vào lưu vào file sử dụng trong quá trình kiểm tra dưới dạng các đặc tả như trình bày ở mục 2.2.1. Ở lựa chọn này việc kiểm tra cấu trúc được phát huy hiệu quả, đảm bảo tính an toàn gần như hoàn toàn cho cơ sở dữ liệu.



**Hình 2.5: Mô hình phát hiện tấn công chèn mã SQL dựa trên phân tích cú pháp câu lệnh.**

## **2. Kiểm tra thành phần câu truy vấn:**

Danh sách các cột, bảng trong câu truy vấn gửi đến máy chủ cơ sở dữ liệu tách ra ở bước 1 được so sánh với bộ danh sách các cột, bảng dữ liệu nhạy cảm không được phép truy cập.

## **3. Kiểm tra dữ liệu đầu vào:**

Dữ liệu đầu vào của người dùng bình thường có thể chứa các đoạn mã nguy hiểm cần kiểm tra, lọc dựa trên các thư viện lọc dữ liệu chuẩn cung cấp bởi dự án OWASP.

## **Bước 3: Kiểm tra kết thúc**

Nếu trong quá trình kiểm tra phát hiện ra tấn công chèn mã SQL thì mọi thông tin về câu truy vấn, thời gian thực hiện, địa chỉ IP của ứng dụng web đều được lưu trữ lại để phục vụ quá trình điều tra, nghiên cứu về sau. Đồng thời câu truy vấn bị loại bỏ. Ngược lại nếu câu truy vấn là hợp lệ nó được chuyển đến máy chủ cơ sở dữ liệu để thực hiện.

### **2.3 Kết chương**

Trong Chương II, luận văn đã trình bày tổng quan về ngôn ngữ SQL với cấu trúc, các lệnh được sử dụng trên hệ quản trị cơ sở dữ liệu và phương pháp phân tích cấu trúc của chúng; các đặc tả của một câu lệnh SQL chuẩn và phương pháp để bắt một câu truy vấn từ ứng dụng web. Trên cơ sở đó luận văn giới thiệu mô hình phân tích cấu trúc của câu truy vấn SQL và phương pháp phát hiện tấn công chèn mã SQL dựa trên phân tích cấu trúc câu truy vấn. Trên cơ sở này, Chương III tập trung xây dựng bộ lọc giúp phát hiện tấn công chèn mã SQL dựa trên phân tích cấu trúc câu truy vấn.

## CHƯƠNG III. THỬ NGHIỆM VÀ ĐÁNH GIÁ

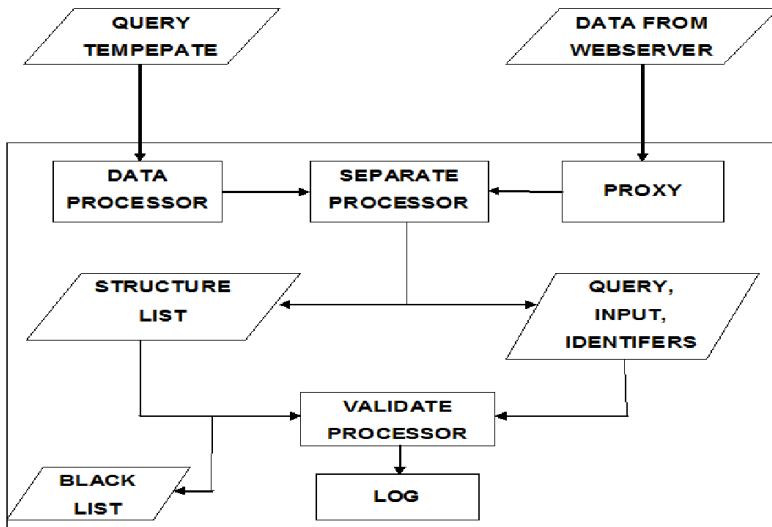
### 3.1 Xây dựng mô hình thử nghiệm

#### 3.1.1 Kiến trúc bộ lọc câu lệnh SQL - Database Filter

Mô hình bộ lọc cơ sở dữ liệu được thực hiện theo 3 giai đoạn xử lý

**Giai đoạn 1:** Xử lý dữ liệu đầu vào. Dữ liệu đầu vào ở đây là bộ danh sách các câu truy vấn chuẩn (Query Template) và các câu truy vấn ứng dụng web cần bảo vệ đang sử dụng.

**Giai đoạn 2:** Lọc câu truy vấn từ máy chủ ứng dụng web gửi tới máy chủ cơ sở dữ liệu. Mỗi gói tin được gửi đi giữa các máy tính khác nhau trong mạng đều được gán địa chỉ IP và kết nối đến một dịch vụ nhất định. Các dịch vụ này thường được định nghĩa theo port đích được gán vào trường header của gói tin.



Hình 3.1: Mô hình bộ lọc cơ sở dữ liệu



**Giai đoạn 3:** Đây là quá trình kiểm tra tính đúng đắn của câu truy vấn được gửi đến máy chủ cơ sở dữ liệu. Nếu câu truy vấn là đúng đắn nó sẽ được gửi đến máy chủ cơ sở dữ liệu để thực hiện ngược lại nó sẽ bị loại bỏ và gây ra lỗi máy chủ không hồi đáp phía máy chủ ứng dụng web.

### ***3.1.2 Cài đặt bộ lọc cơ sở dữ liệu***

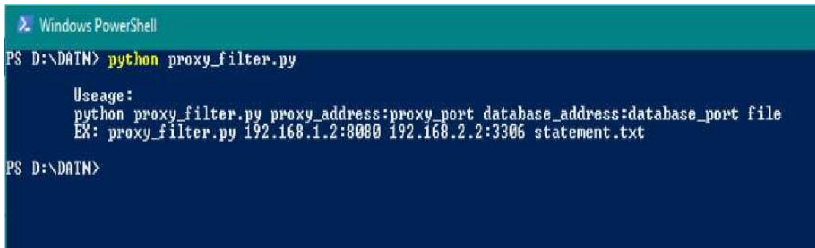
### ***3.1.3 Giới thiệu một số mô đun chương trình***

#### ***3.1.3.1 Mô đun proxy***

#### ***3.1.3.2 Mô đun phân tích cú pháp***

## **3.2 Một số kết quả**

### ***3.2.1 Khởi tạo hệ thống***



```

Windows PowerShell
PS D:\DATN> python proxy_filter.py

Usage:
python proxy_filter.py proxy_address:proxy_port database_address:database_port file
EX: proxy_filter.py 192.168.1.2:8080 192.168.2.2:3306 statement.txt

PS D:\DATN>
  
```

**Hình 3.2: Dòng lệnh thực hiện chạy ứng dụng bộ lọc cơ sở dữ liệu**

```

Windows PowerShell
PS D:\DATN> python proxy_filter.py

Usage:
python proxy_filter.py proxy_address:proxy_port database_address:database_port file
EX: proxy_filter.py 192.168.1.2:8080 192.168.2.2:3306 statement.txt

PS D:\DATN> python proxy_filter.py 127.0.0.1:8080 127.0.0.1:3306 statement.txt
SELECT * FROM Users;
SELECT First_Name, Last_Name Users;
SELECT All FROM Users;
SELECT id FROM users WHERE email = '' OR username = '';
INSERT INTO users (username, password, email) VALUES(,,);
SELECT * FROM users WHERE username = '{username}' OR email = '{username}' AND password = '{password}'
SELECT userid FROM users WHERE email = '{email}' OR username = '{username}';
INSERT INTO users (username, password, email) VALUES('{username}', '{password}', '{email}' );
SET NAMES utf8

-----Parser Structure-----
SELECT * FROM Users;
-----Parser Structure-----
SELECT First_Name, Last_Name Users;
-----Parser Structure-----
SELECT All FROM Users;
-----Parser Structure-----
SELECT id FROM users WHERE email= OR username=;
-----Parser Structure-----
INSERT INTO users(username,password,email) VALUES (,,);
-----Parser Structure-----
SELECT * FROM users WHERE username= OR email= AND password=SELECT userid FROM users WHERE email= OR username=;
-----Parser Structure-----
INSERT INTO users(username,password,email) VALUES (,,);
-----Parser Structure-----
SET NAMES utf8
Process done!

```

**Hình 3.3: Kết quả phân tích tập dữ liệu câu truy vấn đầu vào**

```

2015-12-03 09:45:10,851 - Database filter - INFO - opened connection from ('127.0.0.1', 13992), connection count now 1
SET NAMES utf8

----- Analysis -----
*SET NAMES utf8*
----- Check query -----
*****validate*****
dataSET NAMES utf8:)
SET NAMES utf8
----- Validate Structure -----
* SET NAMES utf8*
f6bbdac839ffb7150f76630c63c2bb7c
step 2
SELECT * FROM users WHERE username = 'admin' OR email = 'admin' AND password = '123456';
----- Analysis -----
*SELECT * FROM users WHERE username = 'admin' OR email = 'admin' AND password = '123456';*
----- Check query -----
*****validate*****
dataSELECT * FROM users WHERE username = 'admin' OR email = 'admin' AND password = '123456';;)
SELECT * FROM users WHERE username= OR email= AND password=;
----- Validate Structure -----
*SELECT * FROM users WHERE username= OR email= AND password=;*
657b4a6ff6e21408ba458ece2a977687
step 2

```

**Hình 3.4: Thông tin bộ lọc cơ sở dữ liệu thu được**

### 3.2.2 Kịch bản thử nghiệm tấn công chèn mã

#### 3.2.2.1 Công thức hằng đúng

Nhập giá trị vào trường đầu vào:

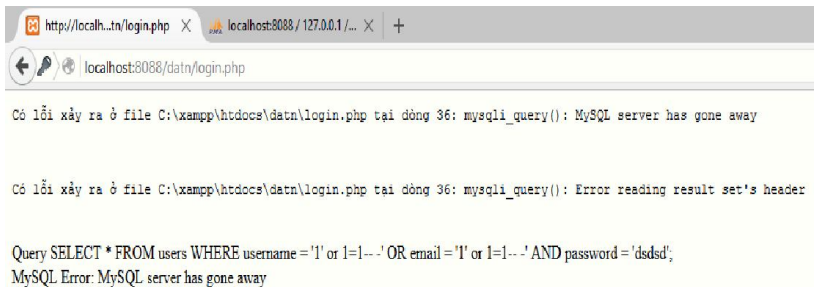
Kết quả trên bộ lọc cơ sở dữ liệu:

```

SELECT * FROM users WHERE username = '1' or 1=1-- -' OR email = '1' or 1=1-- -' AND password = 'dsdsd';
----- Analysis -----
*SELECT * FROM users WHERE username = '1' or 1=1-- -' OR email = '1' or 1=1-- -' AND password = 'dsdsd';*
----- Check query -----
*****validate*****
dataSELECT * FROM users WHERE username = '1' or 1=1-- -' OR email = '1' or 1=1-- -' AND password = 'dsdsd';;)
SELECT * FROM users WHERE username= or = -- -' OR email = '1' or 1=1-- -' AND password = 'dsdsd';
----- Validate Structure -----
*SELECT * FROM users WHERE username= or = -- -' OR email = '1' or 1=1-- -' AND password = 'dsdsd';*
7f88a85844c4a4ab11fbdhc0dec528f3
Structure Detection
Detected:Thu, 03 Dec 2015 03:03:23 Query: SELECT * FROM users WHERE username = '1' or 1=1-- -' OR email = '1' or 1=1-- -

```

**Hình 3.5: Thông tin thu lại khi tấn công chèn mã**  
 Trả thông tin về cho người dùng:



**Hình 3.6: Thông báo lỗi cho người dùng**  
 3.2.2.2 Truy vấn không hợp lệ

Nhập giá trị đầu vào:

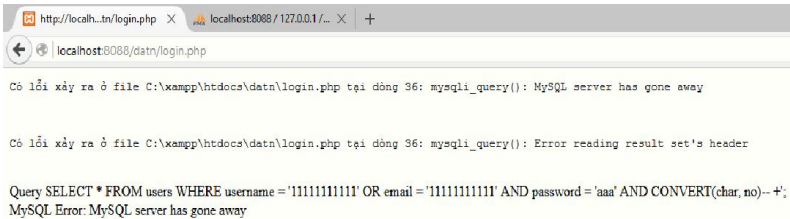
Kết quả trên bộ lọc cơ sở dữ liệu:

```

----- Analysis -----
*SELECT * FROM users WHERE username = '1111111111' OR email = '1111111111' AND password = 'aaa' AND CONVERT(char, no)-- '+';
----- Check query -----
*****validate*****
dataSELECT * FROM users WHERE username = '1111111111' OR email = '1111111111' AND password = 'aaa' AND CONVERT(char, no)-- '+';
SELECT * FROM users WHERE username= OR email= AND password= AND CONVERT(char, no ) -- '+';
----- Validate Structure -----
*SELECT * FROM users WHERE username= OR email= AND password= AND CONVERT(char, no ) -- '+';
331f83f7c7f31499e74fc852f2eb22f5
Structure Detection
Detected:Thu, 03 Dec 2015 03:15:23 Query: SELECT * FROM users WHERE username = '1111111111' OR email = '1111111111' AND password =

```

Trả thông tin về cho người dùng:



The screenshot shows a web browser window with the address bar displaying 'http://localhost:8088/login.php'. The page content shows an error message: 'Có lỗi xảy ra ở file C:\xampp\htdocs\data\login.php tại dòng 36: mysqli\_query(): MySQL server has gone away'. Below this, it says 'Query SELECT \* FROM users WHERE username = '1111111111' OR email = '1111111111' AND password = 'aaa' AND CONVERT(char, no)-- '+'; MySQL Error: MySQL server has gone away'.

### 3.2.2.3 Truy vấn phép hợp

Nhập giá trị đầu vào:

Thông tin trả về phía bộ lọc cơ sở dữ liệu:

```

----- Analysis -----
*SELECT * FROM users WHERE username = 'a' UNION SELECT * FROM (tablename) WHERE userId= admin--+ '+' OR email = 'a' UNION SELECT * FROM (tablename) WHERE userId= admin--+ '+';
----- Check query -----
*****validate*****
dataSELECT * FROM users WHERE username = 'a' UNION SELECT * FROM (tablename) WHERE userId= admin--+ '+' OR email = 'a' UNION SELECT * FROM (tablename) WHERE userId= admin--+ '+';
SELECT * FROM users WHERE username= UNION SELECT * FROM (tablename) WHERE userId= admin--+ '+' OR email = 'a' UNION SELECT * FROM (tablename) WHERE userId= admin--+ '+';
----- Validate Structure -----
*SELECT * FROM users WHERE username= UNION SELECT * FROM (tablename) WHERE userId= admin--+ '+' OR email = 'a' UNION SELECT * FROM (tablename) WHERE userId= admin--+ '+';
4bea56f4e0700a3958c955b27a35da84
Structure Detection
Detected:Thu, 03 Dec 2015 03:22:27 Query: SELECT * FROM users WHERE username = 'a' UNION SELECT * FROM (tablename) WHERE userId= admin--+ '+' OR email = 'a' UNION SELECT * FROM (tablename) WHERE userId= admin--+ '+' AND password = '11111';

```

### **3.3 Nhận xét và đề xuất**

#### **3.3.1 Nhận xét**

Với danh sách câu truy vấn được cung cấp trước bởi nhà phát triển ứng dụng web, bộ lọc hoạt động tốt và hiệu quả do đó có thể đảm bảo các câu truy vấn khi được gửi đến đều phù hợp với cấu trúc ngữ pháp được xây dựng. Kết quả kiểm tra dữ liệu nhạy cảm phụ thuộc vào định nghĩa về dữ liệu nhạy cảm. Do tính chất phụ thuộc vào đầu vào cung cấp bởi nhà phát triển ứng dụng web nên tính phụ thuộc của bộ lọc còn cao. Bù lại, kết quả phát hiện và ngăn chặn ở hình thức này đạt độ an toàn mức cao nhất.

Trong trường hợp danh sách câu truy vấn không được cung cấp trước ứng dụng hoạt động kém hiệu quả hơn. Mô hình phát hiện chủ yếu dựa vào việc kiểm tra tính đúng sai của cấu trúc ngữ pháp và dựa vào danh sách các dữ liệu nhạy cảm và các từ khóa nguy hiểm.

Vấn đề xử lý các câu truy vấn có độ trễ còn cao.

#### **3.3.2 Đề xuất**

Đưa ra hệ thống lọc câu truy vấn từ ứng dụng web bằng cách thực hiện quá trình học truy vấn. Quá trình này ứng dụng web sẽ được nhà phát triển ứng dụng web chạy trong môi trường tách biệt (không có tấn công), mọi chức năng của ứng dụng web sẽ được kiểm tra hoạt động. Trong quá trình kiểm tra các câu truy vấn SQL từ ứng dụng web gửi sang máy chủ cơ sở dữ liệu sẽ được thu thập lại tạo thành bộ mẫu đầu vào cho bộ lọc cơ sở dữ liệu. Điều này đảm bảo tính tiện lợi cho nhà phát triển ứng dụng web.

Nên cài đặt ứng dụng bộ lọc cơ sở dữ liệu như một phần mở rộng của hệ thống phát hiện và ngăn chặn xâm nhập hoặc tường lửa.

Bổ sung hỗ trợ xử lý đa luồng cho hệ thống nhằm giảm độ trễ lọc.

### **3.4 Kết chương**

Chương III đã trình bày chi tiết quá trình cài đặt và thử nghiệm bộ lọc cơ sở dữ liệu dựa trên phân tích cấu trúc cú pháp câu truy vấn để phát hiện tấn công chèn mã SQL. Một số kịch bản tấn công chèn mã đơn giản đã được thử nghiệm.

Từ kết quả thực nghiệm, ta có thể thấy đây là một phương pháp có triển vọng, có hiệu quả để phát hiện ra các cuộc tấn công chèn mã SQL. Mô hình này có thể tiếp tục nghiên cứu, cải thiện, tối ưu và ứng dụng hiệu quả hơn trong thực tế.

## KẾT LUẬN

### Kết quả đạt được

- Luận văn trình bày khái quát về khái quát về tấn công chèn mã SQL, chi tiết về cơ chế và các dạng tấn công chèn mã SQL. Trong đó đã phân tích chi tiết về mục đích, phương thức và mức độ nghiêm trọng của một số kiểu tấn công chèn mã SQL gây ra cho ứng dụng web

- Luận văn đưa ra các biện pháp phát hiện và ngăn chặn tấn công chèn mã SQL từ mức lập trình đến mức nền tảng. Đặc biệt, luận văn tập trung nghiên cứu sâu phương pháp và mô hình phát hiện tấn công chèn mã SQL dựa trên cấu trúc ngữ pháp câu truy vấn.

- Xây dựng và thử nghiệm thành công bộ lọc cơ sở dữ liệu sử dụng trong phát hiện tấn công chèn mã. Từ kết quả thử nghiệm thu được, đề án đã rút ra được ưu điểm, nhược điểm của mô hình và có được những hướng phát triển, cải thiện bộ lọc trong tương lai.

### Hướng phát triển

Đề tài của luận văn có thể được phát triển theo các hướng:

- Tiếp tục thử nghiệm bộ lọc trên với nhiều ứng dụng web cùng kết nối đến một máy chủ cơ sở dữ liệu. Từ đó, ta có thể tinh chỉnh, cài đặt lại ứng dụng sao cho đạt hiệu năng cao nhất. Chương trình cần được tối ưu hóa để tận dụng tối đa hiệu năng của hệ thống phần cứng máy tính và mạng. Điều này có ý nghĩa rất quan trọng trong việc đảm bảo tốc độ truy vấn dữ liệu của ứng dụng web và trải nghiệm của người dùng truy cập vào ứng dụng web.

- Sử dụng phương pháp học truy vấn cho ứng dụng để khả năng phát hiện tấn công chèn mã của ứng dụng đạt hiệu quả cao nhất.

- Nghiên cứu kết hợp ứng dụng với các phương pháp phát hiện tấn công khác nhằm nâng cao khả năng phòng thủ cho ứng dụng web.



## TÀI LIỆU THAM KHẢO

- [1] Kuldeep Kumar, Debasish Jena, and Ravi Kumar, *A Novel Approach to detect SQL injection in web applications*, International Journal of Application or Innovation in Engineering & Management (IJAIEM), ISSN 2319-4847, Volume 2, Issue 6, June 2013.
- [2] V. Nithya, R. Regan and J. Vijayaraghavan, *A Survey on SQL Injection attacks, their Detection and Prevention Techniques*, International Journal Of Engineering And Computer Science, ISSN:2319-7242, Volume 2, Issue 4, April 2013.
- [3] IndraniBalasundaram and E. Ramaraj, *An Approach to Detect and Prevent SQL Injection Attacks in Database Using Web Service*, International Journal of Computer Science and Network Security, VOL.11 No.1, January 2011.
- [4] Konstantinos Kemalis and Theodoros Tzouramanis, *SQL-IDS: A Specification-based Approach for SQL-Injection Detection*, ACM SAC 08, March 16-20, 2008, Fortaleza, Ceara, Brazil.
- [5] Atefeh Tajpour, Suhaimi Ibrahim and Maslin Masrom, *SQL Injection Detection and Prevention Techniques*, International Journal of Advancements in Computing Technology Volume 3, Number 7, August 2011.
- [6] M. Howard và D. LeBlanc, *Writing Secure Code*, tập 2, Microsoft Press, Redmond, Washington, 2003.
- [7] M. F. Estiban, “Advanced SQL Injection in Oracle Databases,” *Black Hat Briefings*, pp. 1-30, 2005.
- [8] C. Anley, *Advanced SQL Injection In SQL Server Applications*, Next Generation Security Software, 2002.
- [9] J. Clarke, *SQL attack and Defense*, Syngress Publishing, Inc., 2009.