ĐỖ MINH HOÀN

HỆ THỐNG THÔNG TIN

2024

BỘ CÔNG THƯƠNG

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**ĐỖ MINH HOÀN**

# PHÁT HIỆN TẤN CÔNG WEBSITE SỬ DỤNG HỌC MÁY

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ HỆ THỐNG THÔNG TIN

Hà Nội – 2024

BỘ CÔNG THƯƠNG
**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**


**ĐỖ MINH HOÀN**

# PHÁT HIỆN TẤN CÔNG WEBSITE SỬ DỤNG HỌC MÁY

Ngành: Hệ thống thông tin

Mã số: 8480104

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ HỆ THỐNG THÔNG TIN

NGƯỜI HƯỚNG DẪN:
1. TS. Nguyễn Bá Nghiễn

Hà Nội – 2024

# LỜI CAM ĐOAN

Tôi xin cam đoan đề án tốt nghiệp thạc sĩ ngành Hệ thống thông tin với đề tài "**PHÁT HIỆN TẤN CÔNG WEBSITE SỬ DỤNG HỌC MÁY**" là do tôi nghiên cứu, tổng hợp và thực hiện dưới sự hướng dẫn của TS.Nguyễn Bá Nghiễn.

Toàn bộ nội dung của đề án, những điều được trình bày là của chính cá nhân tôi hoặc là được tham khảo, tổng hợp từ nhiều nguồn tại liệu khác nhau. Tất cả các tài liệu tham khảo, tổng hợp đều được trích xuất nguồn gốc rõ ràng. Các số liệu, kết quả được nêu trong luận văn là trung thực và được nghiên cứu thực nghiệm.

Hà Nội, ngày 10 tháng 03 năm 2025

**Học viên thực hiện**

**Đỗ Minh Hoàn**

# LỜI CẢM ƠN

Trước hết, em xin bày tỏ tình cảm và lòng biết ơn của em tới Thầy TS. Nguyễn Bá Nghiễn. Người đã từng bước hướng dẫn, giúp đỡ em trong quá trình thực hiện đề án tốt nghiệp của mình.

Em xin chân thành cảm ơn Thầy Cô của Khoa Công nghệ thông tin và Khoa đào tạo sau Đại học Trường Đại học Công nghiệp Hà Nội đã dìu dắt, dạy dỗ em cả về kiến thức chuyên môn và tinh thần học tập để em có được những kiến thức thực hiện đồ án tốt nghiệp của mình.

Em xin chân thành cảm ơn Lãnh đạo Nhà trường, các phòng ban và quý Thầy Cô đã giúp đỡ tạo điều kiện tốt nhất cho em trong suốt thời gian học tập tại trường.

Tuy có nhiều cố gắng trong quá trình học tập, cũng như trong quá trình làm đề án tốt nghiệp không thể tránh khỏi những thiếu sót, em rất mong được sự góp ý quý báu của tất cả các thầy cô giáo cũng như tất cả các anh chị để kết quả của em được hoàn thiện hơn.

Một lần nữa em xin chân thành cảm ơn!

# MỤC LỤC

# DANH MỤC CÁC KÝ HIỆU VÀ TỪ VIẾT TẮT

| Viết tắt | Tiếng Anh | Tiếng Việt |
|---|---|---|
| APT | Advanced Persistent Threat | Mối đe dọa dai dẳng nâng cao |
| ASP | Active Server Pages | Trang máy chủ động |
| DDOS | Distributed Denial of Service | Tấn công từ chối dịch vụ phân tán |
| GPT | Generative Pre-trained Transformer | Mô hình sinh học dựa trên học sâu |
| HTTP | Hypertext Transfer Protocol | Giao thức truyền tải siêu văn bản |
| IDS | Intrusion Detection System | Hệ thống phát hiện xâm nhập |
| IP | Internet Protocol | Giao thức Internet |
| IPS | Intrusion Prevention System | Hệ thống ngăn chặn xâm nhập |
| LSTM | Long Short-Term Memory | Bộ nhớ dài-ngắn hạn |
| MITM | Man-In-The-Middle | Tấn công trung gian |
| PHP | PHP: Hypertext Preprocessor | Bộ tiền xử lý siêu văn bản |
| RNN | Recurrent Neural Network | Mạng nơ-ron hồi quy |
| RSA | Rivest-Shamir-Adleman | Thuật toán mã hóa Rivest-Shamir-Adleman |
| SQL | Structured Query Language | Ngôn ngữ truy vấn có cấu trúc |
| SQLi | Structured Query Language Injection | Tấn công chèn mã SQL |
| SSL | Security Sockets Layer | Lớp cổng bảo mật |
| URL | Uniform Resource Locator | Định vị tài nguyên thống nhất |
| XSS | Cross-Site Scripting | Tấn công kịch bản chéo trang |

# DANH MỤC CÁC HÌNH VẼ

# PHẦN MỞ ĐẦU

## Giới thiệu

Trong thời đại số hóa ngày nay, các ứng dụng web đóng vai trò ngày càng quan trọng trong cuộc sống và hoạt động của con người. Tuy nhiên, cùng với sự gia tăng của các ứng dụng web là sự gia tăng các mối đe dọa an ninh mạng, đặc biệt là các cuộc tấn công nhằm vào các lỗ hổng trong các ứng dụng web. Đề tài "Phát hiện tấn công web dựa trên học máy" khám phá cách áp dụng các kỹ thuật học máy để cải thiện khả năng bảo vệ các hệ thống và ứng dụng web trước các cuộc tấn công mạng ngày càng tinh vi và đa dạng. Trong khi các phương pháp truyền thống thường dựa vào quy tắc cứng nhắc hoặc phân tích dấu vết để phát hiện tấn công, học máy mang lại một cách tiếp cận linh hoạt và thông minh hơn. Bằng cách sử dụng các mô hình học máy, chúng ta có thể phân tích dữ liệu từ các log hệ thống, lưu lượng mạng và các đầu vào từ người dùng để nhận diện các mẫu hành vi bất thường và các kiểu tấn công mới mà có thể không được phát hiện bởi các phương pháp truyền thống.

Học máy cho phép xây dựng các mô hình có khả năng học từ dữ liệu lịch sử, nhận diện các dấu hiệu của tấn công, và dự đoán các cuộc tấn công tiềm ẩn với độ chính xác cao. Các kỹ thuật như phân loại, phát hiện bất thường và học máy được áp dụng để cải thiện khả năng phát hiện và phản ứng nhanh chóng với các mối đe dọa. Đặc biệt, phương pháp học máy như mạng nơ-ron và các mạng tích chập có thể giúp nhận diện các tấn công phức tạp hơn mà các phương pháp đơn giản không thể phát hiện được. [7]

Với việc áp dụng học máy, hệ thống có thể tự động điều chỉnh và cập nhật các mô hình phát hiện để đối phó với các kiểu tấn công mới và thay đổi trong hành vi của kẻ tấn công. Điều này giúp tăng cường bảo mật, giảm thiểu rủi ro

và bảo vệ hệ thống web một cách hiệu quả hơn trong môi trường mạng ngày càng phức tạp. Đề tài này không chỉ góp phần vào việc nâng cao khả năng phòng thủ của các hệ thống web mà còn mở ra các cơ hội nghiên cứu và phát triển mới trong lĩnh vực bảo mật mạng.

**Lý do chọn đề tài**

Cùng với sự phát triển nhanh chóng của Internet, số lượng các Website không ngừng gia tăng, trở thành một phần không thể thiếu trong cuộc sống và hoạt động kinh doanh của con người. Tuy nhiên, song song với sự phát triển đó là sự gia tăng đáng kể về số lượng và mức độ tinh vi của các mối đe dọa, đặc biệt là các cuộc tấn công nhằm vào hệ thống Website. Những cuộc tấn công này không chỉ gây tổn hại đến dữ liệu và tài sản số mà còn làm suy giảm niềm tin của người dùng và doanh nghiệp vào môi trường trực tuyến.

Hiện nay, nhiều kỹ thuật bảo mật đã được giới thiệu và triển khai, từ các giải pháp đơn giản như phân quyền người dùng, sử dụng tường lửa (firewall) cho đến các phương pháp nâng cao như hệ thống phát hiện xâm nhập (IDS). Tuy nhiên, những phương pháp truyền thống này vẫn tồn tại những hạn chế nhất định, đặc biệt là khi đối mặt với các kiểu tấn công mới ngày càng phức tạp và khó lường. Các hệ thống phát hiện tấn công phổ biến dựa trên quy tắc (rule-based) hoặc chữ ký (signature-based) thường không đủ linh hoạt để nhận diện và đối phó với các biến thể tấn công mới, bởi chúng phụ thuộc nhiều vào các mẫu tấn công đã biết trước đó.

Trước thực trạng đó, việc nghiên cứu và áp dụng các giải pháp bảo mật dựa trên học máy (machine learning) nổi lên như một hướng đi tiềm năng, mang lại khả năng vượt trội trong việc nhận diện và dự đoán các cuộc tấn công một cách hiệu quả. Học máy không chỉ cho phép phát hiện các hành vi bất thường

trong thời gian thực mà còn có khả năng dự đoán các cuộc tấn công tiềm ẩn với độ chính xác cao, nhờ khả năng học hỏi từ dữ liệu lịch sử và tự động thích nghi với các kiểu tấn công mới.

Luận văn chọn đề tài này không chỉ nhằm mục tiêu nghiên cứu lý thuyết và khám phá tiềm năng của các thuật toán học máy trong lĩnh vực bảo mật, mà còn hướng tới việc phát triển một hệ thống có tính ứng dụng cao trong thực tiễn. Hệ thống này không chỉ giúp tăng cường khả năng bảo vệ Website trước các cuộc tấn công hiện tại mà còn góp phần xây dựng nền tảng vững chắc cho các giải pháp an ninh mạng trong tương lai, đáp ứng tốt hơn nhu cầu của một thế giới số ngày càng phức tạp.

**Tổng quan nghiên cứu**

Đề tài tập trung vào việc ứng dụng các kỹ thuật học máy để nâng cao khả năng bảo vệ các hệ thống và ứng dụng web trước các cuộc tấn công mạng ngày càng tinh vi. Các cuộc tấn công web như SQL Injection, Cross-Site Scripting và Cross-Site Request Forgery đã trở thành mối đe dọa nghiêm trọng đối với an ninh mạng, ảnh hưởng đến sự an toàn và tính toàn vẹn của hệ thống thông tin. Các phương pháp phát hiện tấn công truyền thống thường dựa vào quy tắc cứng nhắc và phân tích dấu vết của tấn công, tuy nhiên, những phương pháp này thường gặp khó khăn trong việc nhận diện các cuộc tấn công mới hoặc tinh vi hơn, do chúng dựa trên các mẫu và quy tắc đã được xác định trước. [2] [4]

Học máy với khả năng học từ dữ liệu và tự động cải thiện hiệu suất mà không cần phải lập trình cụ thể cho từng nhiệm vụ, mang đến một phương pháp tiếp cận mới trong phát hiện tấn công. Bằng cách áp dụng các kỹ thuật học máy như phân loại, phát hiện bất thường và học sâu, chúng ta có thể phát hiện các hành vi bất thường và các kiểu tấn công chưa biết. Các mô hình học máy có

khả năng phân tích dữ liệu từ nhiều nguồn khác nhau, bao gồm các log hệ thống, lưu lượng mạng, và đầu vào từ người dùng, để nhận diện các mẫu hành vi bất thường có thể chỉ ra một cuộc tấn công. Các kỹ thuật như mạng nơ-ron sâu và mạng tích chập cung cấp khả năng nhận diện các mẫu phức tạp và tinh vi hơn mà các phương pháp truyền thống có thể bỏ qua. [8]

Mặc dù việc áp dụng học máy trong phát hiện tấn công web mang lại nhiều lợi ích đáng kể như khả năng phát hiện các tấn công mới và cải thiện độ chính xác của hệ thống phát hiện, nó cũng đối mặt với một số thách thức. Một trong những thách thức chính là yêu cầu về chất lượng và khối lượng dữ liệu, vì các mô hình học máy cần dữ liệu đủ lớn và chính xác để huấn luyện và hoạt động hiệu quả. Thêm vào đó, các mô hình cần được cập nhật thường xuyên để đối phó với các kiểu tấn công mới và thay đổi trong hành vi của kẻ tấn công. Việc duy trì sự cân bằng giữa việc phát hiện tấn công và giảm thiểu số lượng báo cáo giả (false positives) cũng là một yếu tố quan trọng để đảm bảo hiệu quả của hệ thống.

Nghiên cứu trong lĩnh vực này tiếp tục phát triển với các xu hướng như cải thiện khả năng phát hiện trong thời gian thực, tích hợp học máy với các phương pháp bảo mật khác, và áp dụng các kỹ thuật học máy để phát hiện các tấn công ngày càng tinh vi. Những tiến bộ này mở ra cơ hội mới cho việc phát triển các giải pháp phòng chống tấn công web hiệu quả hơn, giúp nâng cao an ninh mạng và bảo vệ hệ thống thông tin khỏi các mối đe dọa ngày càng phức tạp.

**Mục tiêu của đề tài**

Mục tiêu chính của đề tài nghiên cứu là xây dựng được một hệ thống giám sát có đầy đủ chức năng quản trị cấu hình cảnh báo và thiết lập giám sát. Cụ thể, các mục tiêu chi tiết của đề tài bao gồm:

- Thu thập bộ dữ liệu cho việc huấn luyện mô hình học máy đủ lớn.
- Xây dựng hệ thống giám sát.
- Giám sát liên tục các sự kiện trên máy chủ web.
- Xây dựng kịch bản thử nghiệm để kiểm tra và đánh giá tính khả thi và hiệu quả của hệ thống.

**Nội dung nghiên cứu**

Nghiên cứu và phân tích các hình thức tấn công vào website:

- Xác định các loại tấn công phổ biến như SQL injection, cross-site scripting và các kỹ thuật khác mà kẻ tấn công sử dụng để xâm nhập và gây hại cho website. [2]
- Phân tích các dấu hiệu và hành vi bất thường trên các website bị tấn công. [6]

Khắc phục hạn chế của các phương pháp truyền thống:

- Đánh giá các nhược điểm của các phương pháp phát hiện tấn công hiện tại, đặc biệt là những phương pháp dựa trên chữ ký và luật.
- Tìm hiểu nguyên nhân gây ra các cảnh báo sai (false positives) và bỏ sót (false negatives) trong các hệ thống hiện tại.

Ứng dụng học máy để cải thiện khả năng phát hiện tấn công:

- Thiết kế và phát triển các mô hình học máy phù hợp như mạng nơ-ron tích chập ,mạng nơ-ron tái phát và các kiến trúc học máy khác để phát hiện các hành vi bất thường và tấn công vào website.

- Huấn luyện các mô hình học máy với dữ liệu thực tế và dữ liệu mô phỏng các cuộc tấn công vào website.

Xây dựng hệ thống phát hiện tấn công vào website:

- Phát triển một hệ thống hoàn chỉnh có khả năng giám sát các website và phát hiện kịp thời các cuộc tấn công.
- Tích hợp hệ thống phát hiện với các công cụ bảo mật hiện có để cung cấp giải pháp bảo mật toàn diện.

Đánh giá hiệu quả của hệ thống:

- Thực hiện các thử nghiệm trên các bộ dữ liệu kiểm tra để đánh giá độ chính xác, độ nhạy và tính đặc hiệu của hệ thống.
- So sánh hiệu quả của hệ thống học máy với các phương pháp truyền thống và các giải pháp hiện có.

Đề xuất các biện pháp cải thiện và ứng dụng thực tiễn:

- Đề xuất các biện pháp tối ưu hóa mô hình học máy để nâng cao hiệu suất và giảm thiểu tài nguyên tính toán.
- Xác định các lĩnh vực ứng dụng tiềm năng của hệ thống phát hiện tấn công vào website và đề xuất các hướng nghiên cứu tiếp theo.

**Phương pháp nghiên cứu**

Để đạt được mục tiêu nghiên cứu của đề tài, các phương pháp sau có thể được áp dụng, bao gồm:

- Thu thập dữ liệu và phân tích: Thu thập bộ dữ liệu ảnh của các trang web bình thường và bị tấn công thay đổi giao diện, phân tích đặc điểm của các bức ảnh này.

- Nghiên cứu giải pháp đã có trước đây dựa trên các bài báo, blog, xem xét các ưu và nhược điểm, đề xuất phương pháp khắc phục và tiến hành triển khai.
- Nghiên cứu ứng dụng mô hình học máy để huấn luyện mô hình học máy cho hệ thống theo dõi.
- Thực hiện thử nghiệm và đánh giá: Thử nghiệm triển khai vào trang web thực tế và đánh giá khả năng triển khai, tính ổn định và hiệu suất của chúng.

Ngoài ra, trong quá trình thực hiện đề tài, các thư viện mã nguồn sẽ được nghiên cứu và triển khai thực nghiệm. Dưới đây là một số thư viện mã nguồn mở phổ biến có thể được xem xét trong nghiên cứu này:

- PyTorch: PyTorch là thư viện máy học dựa trên thư viện Torch, được sử dụng cho các ứng dụng như thị giác máy tính và xử lý ngôn ngữ tự nhiên.
- Flask: là một nền tảng ứng dụng web lightweight WSGI. Nó được thiết kế để giúp bắt đầu nhanh chóng và dễ dàng phát triển ứng dụng web dựa trên python, với khả năng mở rộng lên các ứng dụng phức tạp. Nó bắt đầu như một trình bao bọc đơn giản xung quanh Werkzeug và Jinja , và đã trở thành một trong những nền tảng ứng dụng web Python phổ biến nhất..
- Scikit-learn : là một thư viện máy học mã nguồn mở cung cấp một tập hợp các thuật toán học máy phổ biến, bao gồm phân loại, hồi quy, phân cụm và giảm chiều, với một giao diện người dùng rất thân thiện và dễ sử dụng. Scikit-learn được xây dựng dựa trên các thư viện mạnh mẽ như NumPy, SciPy và Matplotlib, đảm bảo hiệu suất cao trong các tính toán.

Việc áp dụng các phương pháp và công cụ nghiên cứu một cách khoa học sẽ giúp xây dựng được hiệu quả hệ thống giám sát và phát hiện tấn công theo thời gian thực.

# CHƯƠNG 1 - CƠ SỞ LÝ THUYẾT

## 1.1. TỔNG QUAN VỀ AN NINH MẠNG

### 1.1.1. Giới thiệu về an ninh mạng

An ninh mạng là việc bảo vệ các hệ thống mạng máy tính, thiết bị, và dữ liệu khỏi các cuộc tấn công kỹ thuật số độc hại. Điều này bao gồm việc bảo vệ phần cứng, phần mềm và dữ liệu khỏi các hành vi trộm cắp, phá hoại, hoặc truy cập trái phép.

*Hình 1.1:Hình ảnh về an ninh mạng*

An ninh mạng hoạt động thông qua các biện pháp như kiểm soát truy cập, mã hóa dữ liệu, và sử dụng phần mềm bảo mật để ngăn chặn các mối đe dọa. Các tổ chức và cá nhân đều cần thực hiện các biện pháp an ninh mạng để bảo vệ thông tin và duy trì hoạt động an toàn trên không gian mạng.

### 1.1.2. Lịch sử phát triển của an ninh mạng

Lịch sử phát triển của an ninh mạng là một hành trình dài và phức tạp, bắt đầu từ những năm 1970 và tiếp tục phát triển mạnh mẽ cho đến ngày nay.

- 1971 – Virus máy tính đầu tiên xuất hiện

Mặc dù máy tính cần được phát minh trước khi khái niệm virus máy tính ra đời, nhưng ý tưởng này đã được nhà toán học John von Neumann (1903-1957) khái niệm hóa từ năm 1949. Ông đề xuất nền tảng lý thuyết về thực thể tự nhân bản tự động, hoạt động trong máy tính.

Năm 1971, virus máy tính đầu tiên mang tên Creeper mới xuất hiện trong thực tế. Hoạt động trên hệ điều hành TENEX của các máy tính DEC PDP-10, virus này hiển thị thông báo: "Tôi là Creeper. Hãy bắt tôi nếu bạn có thể!". Dù chỉ là một thử nghiệm vô hại, Creeper đã đặt nền móng cho sự phát triển của các loại virus sau này.

- 1983 – Bằng sáng chế đầu tiên trong lĩnh vực an ninh mạng tại Mỹ

Tháng 9 năm 1983, Viện Công nghệ Massachusetts (MIT) được cấp bằng sáng chế số 4.405.829 cho một "hệ thống và phương thức truyền thông mật mã". Thuật toán RSA, được giới thiệu trong bằng sáng chế này, là một trong những hệ thống mã hóa khóa công khai đầu tiên, đặt nền tảng cho bảo mật hiện đại.

- 1993 – Hội nghị DEF CON đầu tiên

DEF CON, hội nghị an ninh mạng danh tiếng toàn cầu, lần đầu tổ chức vào tháng 6 năm 1993 tại Las Vegas, với khoảng 100 người tham dự. Hiện nay, sự kiện này thu hút hơn 20.000 chuyên gia an ninh mạng, hacker mũ trắng và nhà nghiên cứu công nghệ từ khắp nơi trên thế giới.

- 1995 – Sự ra đời của giao thức SSL 2.0

SSL 2.0, ra mắt tháng 2 năm 1995, đánh dấu bước tiến lớn trong bảo mật web. Giao thức này mã hóa liên lạc giữa máy chủ và trình duyệt, bảo vệ các

giao dịch trực tuyến. Ngày nay, sự hiện diện của "HTTPS" trên địa chỉ website đảm bảo rằng thông tin liên lạc được mã hóa an toàn, ngay cả khi kết nối bị xâm nhập.

- 2003 – Sự xuất hiện của Anonymous

Nhóm hacker nổi tiếng toàn cầu Anonymous đại diện cho phong trào bảo vệ tự do ngôn luận và Internet. Không có lãnh đạo cụ thể, nhóm này thu hút sự chú ý quốc tế qua các cuộc biểu tình và tấn công từ chối dịch vụ (DDoS) nhắm vào các chính quyền, tôn giáo và công ty lớn. Hình ảnh biểu tượng của họ là chiếc mặt nạ Guy Fawkes.

- 2010 – Chiến dịch Ánh ban mai (Operation Aurora)

Cuối năm 2009, Google Trung Quốc và hơn 50 công ty lớn bị tấn công mạng trong chiến dịch Operation Aurora. Các phân tích chỉ ra mục tiêu thực sự là thu thập thông tin tình báo từ các nhà hoạt động nhân quyền và xác định điệp viên tại Mỹ. Chiến dịch này gây thiệt hại ước tính lên đến 100 triệu USD cho mỗi công ty nạn nhân.

- Ngày nay – Vai trò sống còn của an ninh mạng

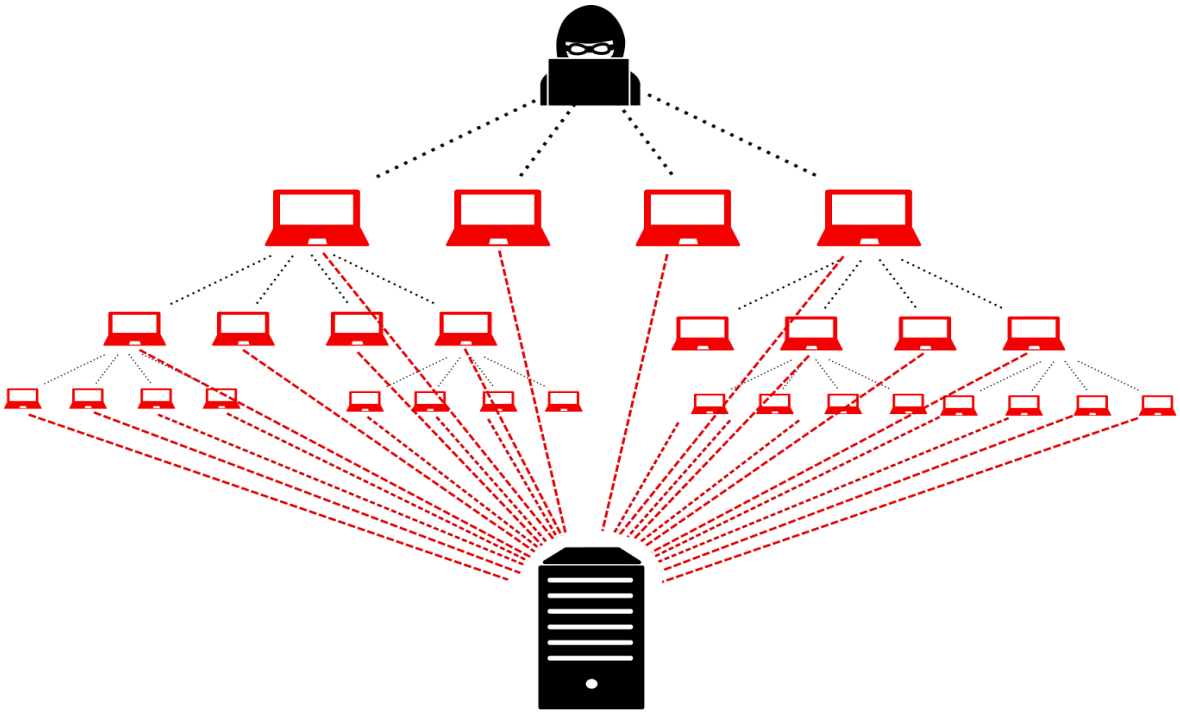
Không gian mạng hiện tại là chiến trường số, nơi các quốc gia và tội phạm mạng đối đầu. Với sự phát triển của trí tuệ nhân tạo (AI) và học máy nâng cao, an ninh mạng đang dần chuyển mình để phân tích hành vi mạng và ngăn chặn các cuộc tấn công. Đối với doanh nghiệp và tổ chức, đầu tư vào bảo mật mạng không chỉ là sự lựa chọn, mà là điều kiện tiên quyết trong kỷ nguyên số.

### 1.1.3. Các phương thức tấn công mạng phổ biến

#### *1.1.3.1. Tấn công từ chối dịch vụ (DoS/DDoS)*

Một cuộc tấn công DDoS nhằm chiếm dụng tài nguyên của hệ thống, khiến hệ thống không thể phản hồi các yêu cầu dịch vụ bình thường. Khác với DoS,

DDoS được thực hiện từ một mạng lưới lớn các máy tính bị lây nhiễm phần mềm độc hại và được kiểm soát bởi hacker.



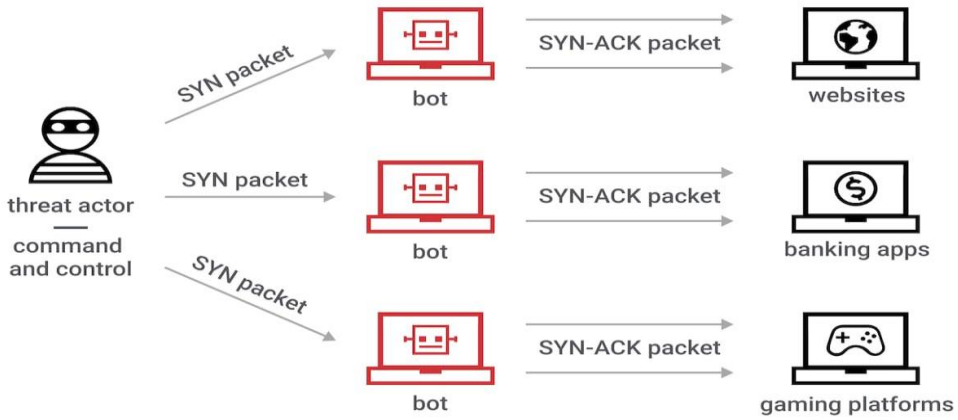
*Hình 1.2 : Tấn công distributed denial-of-service (DDoS)*

Không giống các cuộc tấn công khác nhằm chiếm quyền truy cập vào hệ thống, DDoS không mang lại lợi ích trực tiếp ngay lập tức cho kẻ tấn công. Tuy nhiên, với một số hacker, việc khiến hệ thống không thể hoạt động đã là một mục tiêu đủ thỏa mãn. Trong một số trường hợp, nếu hệ thống bị tấn công là của đối thủ cạnh tranh trong kinh doanh, lợi ích tiềm tàng cho kẻ tấn công sẽ rất đáng kể.

Một mục đích khác của DDoS có thể là đưa hệ thống offline để tạo điều kiện triển khai các loại tấn công khác, chẳng hạn như chiếm quyền điều khiển. Có nhiều dạng tấn công DoS và DDoS khác nhau, trong đó phổ biến nhất bao gồm TCP SYN flood, Teardrop, Smurf, ping-of-death và botnet.

### *1.1.3.2. Tấn công TCP SYN flood*

Trong kiểu tấn công TCP SYN flood, kẻ tấn công lợi dụng quá trình sử dụng bộ nhớ đệm trong giai đoạn handshake khởi tạo kết nối TCP. Kẻ tấn công liên tục gửi một lượng lớn yêu cầu kết nối tới hệ thống mục tiêu, nhưng không phản hồi lại khi hệ thống trả lời các yêu cầu đó.

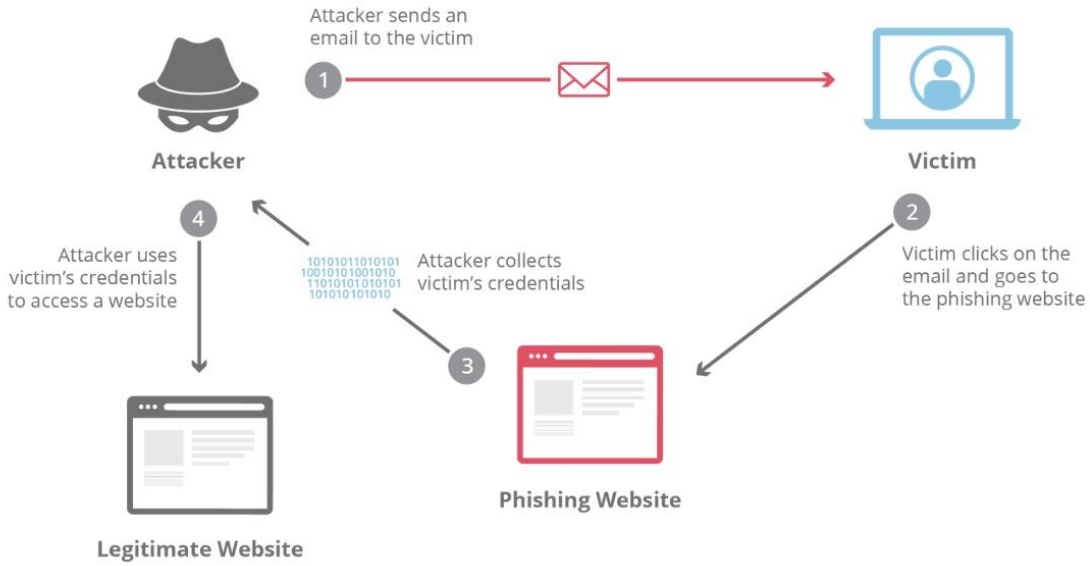



*Hình 1.3: Tấn công TCP SYN flood*

Hành vi này khiến hệ thống mục tiêu phải chờ đợi phản hồi từ phía thiết bị của kẻ tấn công, dẫn đến việc hàng đợi xử lý kết nối bị quá tải. Khi hàng đợi đầy, hệ thống có thể gặp sự cố hoặc trở nên không khả dụng do không thể xử lý thêm các yêu cầu kết nối hợp lệ.

### *1.1.3.3. Tấn công giả mạo (Phishing)*

Phishing (tấn công giả mạo) là hình thức tấn công mạng bằng cách giả mạo thành một đơn vị uy tín để chiếm lòng tin, thường được sử dụng để đánh cắp thông tin cá nhân, dữ liệu người dùng (thông tin đăng nhập; số thẻ ngân hàng;…).




*Hình 1.4: Tấn công giả mạo*

Thông thường, Hacker sẽ giả mạo là ngân hàng, ví điện tử, trang giao dịch trực tuyến hoặc các cơ quan, tổ chức doanh nghiệp để lừa người dùng chia sẻ các thông tin cá nhân như: tài khoản và mật khẩu đăng nhập, mật khẩu giao dịch, thẻ tín dụng và các thông tin quan trọng khác. Phương thức tấn công này thường được thực hiện thông qua việc gửi Email, tin nhắn, cuộc gọi. Người dùng khi mở Email và Click vào đường Link giả mạo sẽ được yêu cầu đăng nhập.

- Phương thức giả mạo Email:

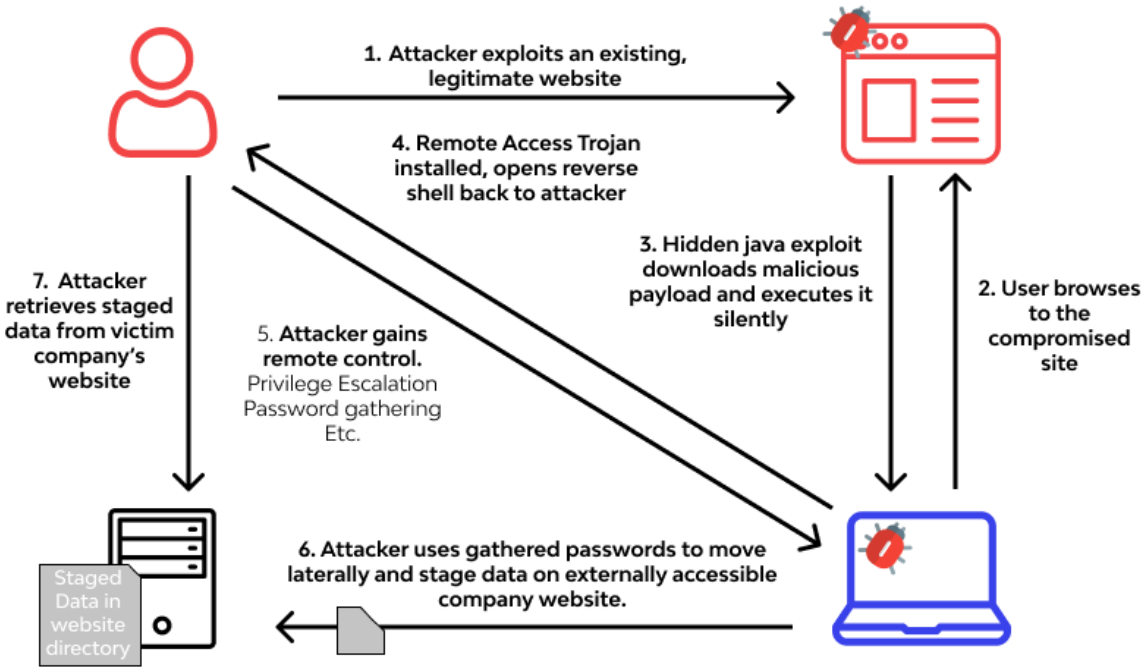
Đây là hình thức Phishing thường được sử dụng phổ biến. Tin tặc sẽ gửi Email đến người dùng dưới danh nghĩa của một đơn vị/tổ chức uy tín nhằm dẫn dụ người dùng truy cập đến Website giả mạo. Những Email giả mạo thường rất tinh vi và rất giống với Email chính chủ, khiến người dùng nhầm lẫn và trở thành nạn nhân của cuộc tấn công.

- Phương thức giả mạo Website:

Giả mạo Website trong tấn công Phishing là làm giả một trang chứ không phải toàn bộ Website. Trang được làm giả thường là trang đăng nhập để đánh cắp thông tin của người dùng.

### *1.1.3.4. Tấn công Drive-by download*

Drive-by download là một phương pháp phổ biến để phát tán malware. Hacker tìm kiếm các trang web không được bảo vệ và đặt một tập lệnh độc hại vào code HTTP hoặc PHP trên một trong các trang đó. Tập lệnh này có thể cài đặt malware trực tiếp vào máy tính của người nào mà truy cập trang web đó hoặc nó có thể hướng người dùng đến trang web do hacker kiểm soát.



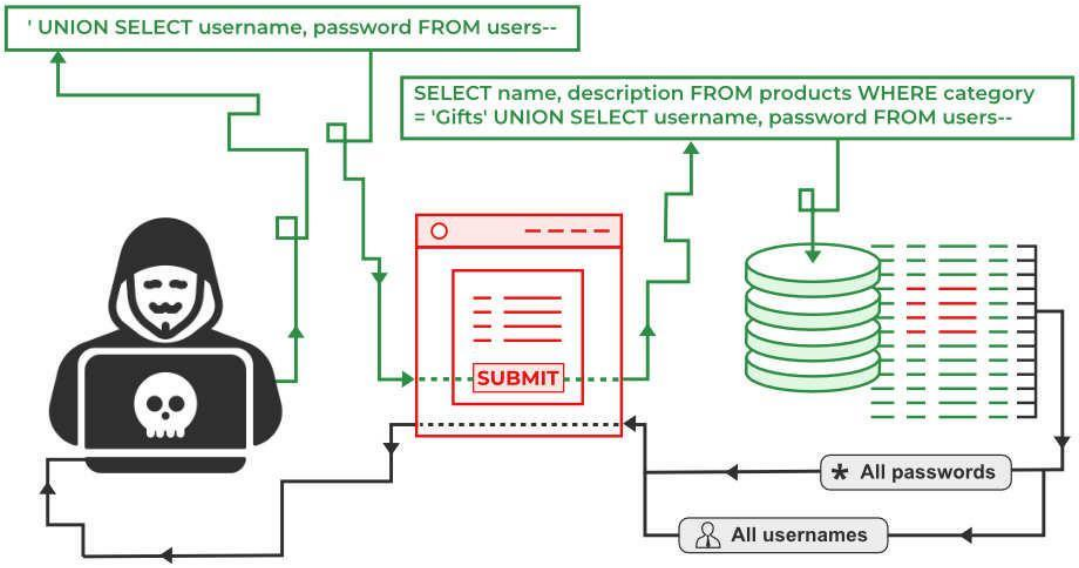

*Hình 1.5: Tấn công Drive-by download*

Những Drive-by download có thể xảy ra khi truy cập trang web hoặc xem email hoặc cửa sổ pop-up. Không giống như nhiều loại tấn công an ninh mạng khác, drive-by không dựa vào người để làm bất kỳ điều gì chủ động kích hoạt cuộc tấn công, bạn không phải nhấp vào nút tải xuống hoặc mở file đính kèm email độc hại để bị nhiễm. Một Drive-by download có thể tận dụng lợi thế của

ứng dụng, hệ điều hành hoặc trình duyệt web browser có lỗi bảo mật do cập nhật không thành công hoặc thiếu các bản cập nhật.

### *1.1.3.5. Tấn công SQL injection*

SQL injection đã trở thành một vấn đề phổ biến với các trang web database-driven. Nó xảy ra khi một người dùng thực thi một truy vấn query SQL đến database thông qua dữ liệu input từ máy client đến server. Các lệnh SQL được chèn vào input data-plane (ví dụ: thay vì đăng nhập hoặc mật khẩu) để chạy các lệnh SQL được xác định trước.




*Hình 1.6: Tấn công SQL Injection*

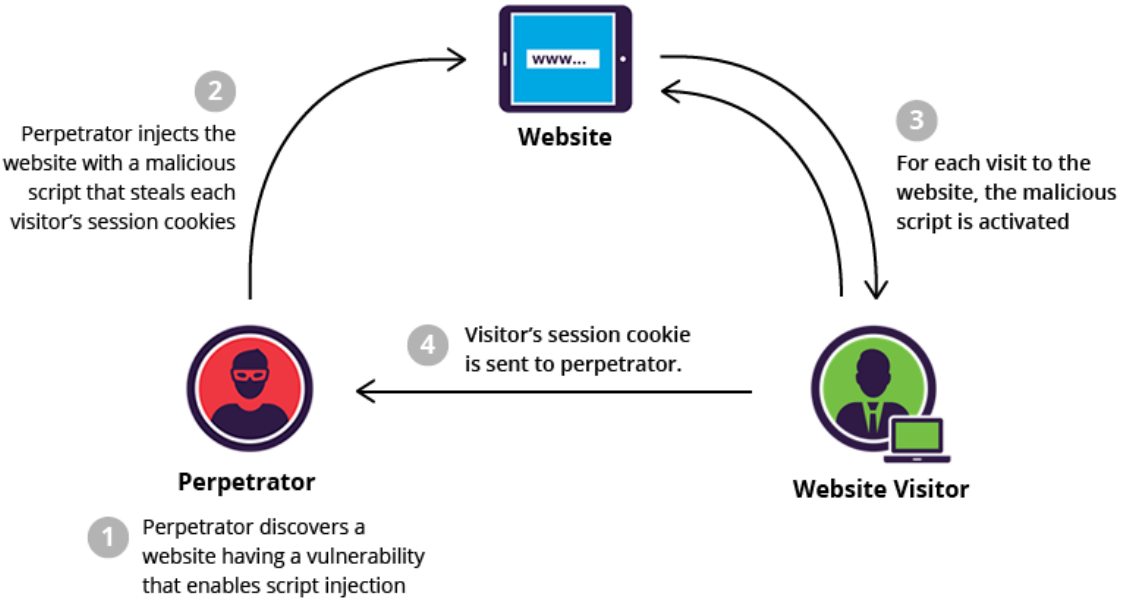
Khai thác SQL Injection thành công có thể đọc dữ liệu nhạy cảm từ database, sửa đổi (chèn, cập nhật hoặc xóa) database, thực thi các hoạt động quản trị (chẳng hạn như tắt máy) trên database, khôi phục nội dung của một file nhất định và trong một số trường hợp, nó còn ra lệnh cho hệ điều hành.

Lỗ hổng đối với kiểu tấn công an ninh mạng này phụ thuộc vào thực tế là SQL không phân biệt thực sự giữa các data plane và dữ liệu. Do đó, SQL injection chủ yếu hoạt động nếu một trang web sử dụng SQL động. Ngoài ra, SQL

injection rất phổ biến với các ứng dụng PHP và ASP vì sự thịnh hành của các interface có chức năng cũ hơn.

### *1.1.3.6. Tấn công Cross-site scripting (XSS)*

Các cuộc tấn công XSS sử dụng tài nguyên web của bên thứ ba để chạy các tập lệnh trong trình duyệt browser web hoặc ứng dụng liên quan tập lệnh. Cụ thể, kẻ tấn công tiêm nhiễm một payload chứa JavaScript độc hại vào database của trang web. Khi người dùng yêu cầu một trang từ trang web, trang web sẽ truyền trang, với payload của hacker là một phần của nội dung HTML, đến trình duyệt browser của người dùng, nơi thực thi tập lệnh độc hại.

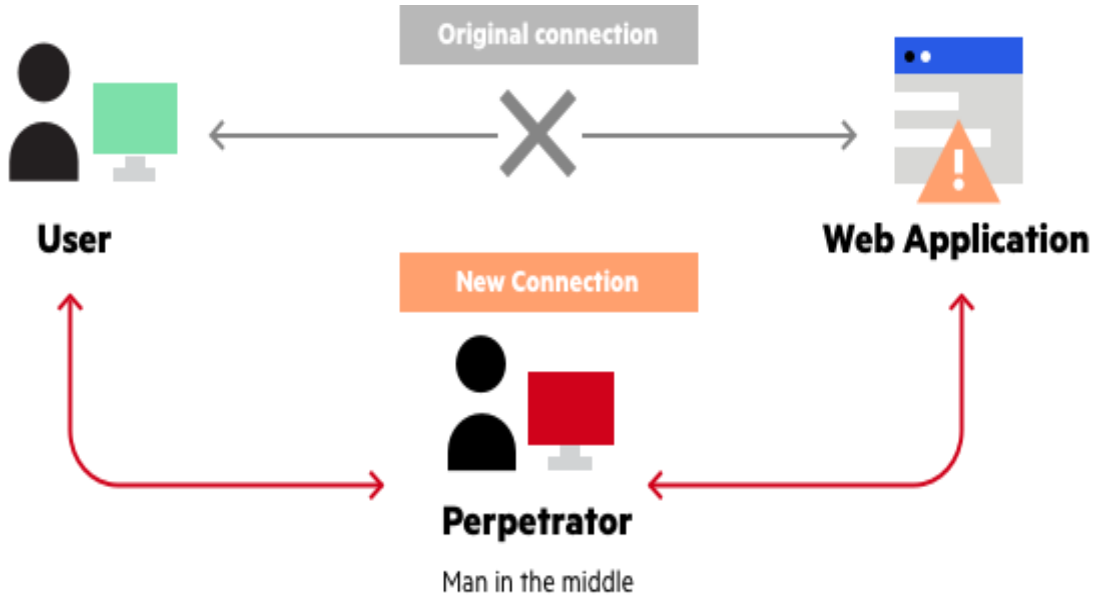



*Hình 1.7: Tấn công Cross-site scripting*

Tấn công XSS sẽ đánh cắp cookie của người dùng đến server của hacker và hacker có thể trích xuất nó và sử dụng nó để chiếm quyền điều khiển phiên bản. Hậu quả nguy hiểm nhất xảy ra khi XSS được sử dụng để khai thác các lỗ hổng. Những lỗ hổng này có thể cho phép hacker không chỉ ăn cắp cookie mà còn ghi lại các lần gõ phím, chụp ảnh màn hình, khám phá và thu thập thông tin mạng cũng như truy cập và điều khiển từ xa máy của người dùng.

### *1.1.3.7. Tấn công Man-in-the-Middle*

Man-in-the-Middle là một loại tấn công mạng trong đó kẻ tấn công xâm nhập vào thông tin liên lạc giữa hai bên mà họ không hay biết. Kẻ tấn công có thể nghe lén cuộc trao đổi, đánh cắp dữ liệu nhạy cảm hoặc thậm chí chỉnh sửa dữ liệu được truyền đi. MITM thường xảy ra trên các mạng Wi-Fi không an toàn, router bị xâm nhập, hoặc khi thiết bị bị nhiễm phần mềm độc hại.



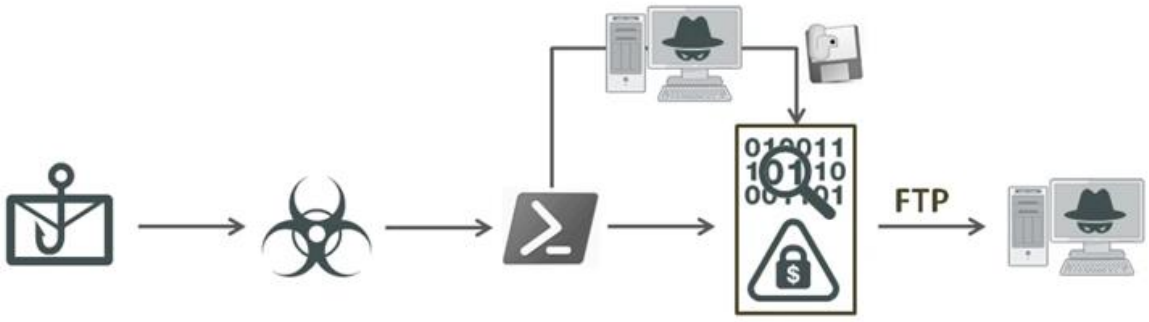

*Hình 1.8: Tấn công Man-in-the-Middle*

Các hình thức phổ biến của MITM bao gồm nghe lén, thay đổi dữ liệu, chiếm quyền điều khiển phiên bằng cách đánh cắp cookie, hoặc hạ cấp kết nối mã hóa từ HTTPS xuống HTTP để thu thập thông tin dưới dạng văn bản thường.

### *1.1.3.8. Tấn công bằng phần mềm độc hại (Malware Attack)*

Tấn công bằng phần mềm độc hại (Malware attack) là một loại tấn công mạng trong đó kẻ tấn công sử dụng phần mềm độc hại để xâm nhập, phá hoại hoặc kiểm soát hệ thống máy tính của nạn nhân mà không được phép. Phần mềm độc hại có thể tồn tại dưới nhiều dạng khác nhau, bao gồm virus, trojan, worm (sâu máy tính), ransomware, spyware (phần mềm gián điệp) và adware

(phần mềm quảng cáo). Mỗi loại phần mềm độc hại đều có mục đích khác nhau, từ việc đánh cắp thông tin cá nhân, theo dõi hoạt động của nạn nhân, cho đến mã hóa dữ liệu để đòi tiền chuộc.

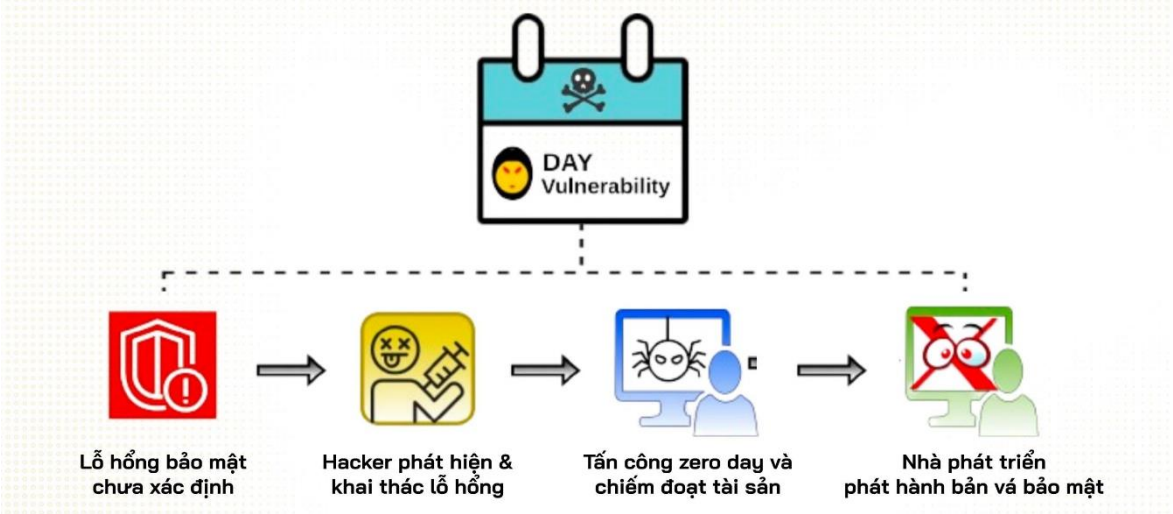



*Hình 1.9: Tấn công bằng phần mềm độc hại*

Phần mềm độc hại thường lây lan qua các phương tiện như email chứa tệp đính kèm độc hại, trang web độc hại, tải xuống không an toàn, hoặc thậm chí là qua các thiết bị lưu trữ như USB. Khi phần mềm độc hại xâm nhập thành công vào hệ thống, nó có thể gây ra những hậu quả nghiêm trọng như làm hỏng dữ liệu, làm chậm hệ thống, hoặc chiếm quyền điều khiển hoàn toàn hệ thống của nạn nhân.

#### *1.1.3.9. Tấn công khai thác lỗ hổng Zero-day (Zero-day Attack)*

Khai thác lỗ hổng Zero-day (Zero day attack) là một hình thức tấn công mạng lợi dụng các lỗ hổng bảo mật chưa được phát hiện hoặc chưa được nhà phát triển khắc phục. Đây là những lỗ hổng chưa được công khai, vì vậy không có biện pháp bảo vệ hoặc bản vá (patch) nào sẵn có để ngăn chặn. Khi kẻ tấn công phát hiện lỗ hổng này, chúng có thể nhanh chóng khai thác trước khi nhà phát triển hoặc người dùng có thể ứng phó, gây ra những rủi ro nghiêm trọng cho hệ thống hoặc dữ liệu.




*Hình 1.10: Tấn công khai thác lỗ hổng Zero-day*

Trong các cuộc tấn công Zero-day, kẻ tấn công thường sử dụng phần mềm độc hại hoặc khai thác mã độc nhằm xâm nhập hệ thống, đánh cắp thông tin, phá hủy dữ liệu, hoặc chiếm quyền điều khiển thiết bị. Những lỗ hổng này có thể tồn tại trong phần mềm, hệ điều hành, ứng dụng, hoặc các phần cứng như router và thiết bị IoT. Do tính chất bất ngờ của Zero-day, các hệ thống mục tiêu rất dễ bị tổn thương trước khi phát hiện ra mối đe dọa và phát hành bản vá bảo mật.

### 1.1.4. Ảnh hưởng của các cuộc tấn công đến bảo mật thông tin

Các phương thức tấn công mạng ngày càng trở nên tinh vi, đa dạng và phức tạp, khiến cho an ninh mạng trở thành một trong những thách thức hàng đầu đối với các tổ chức và cá nhân. Các cuộc tấn công bằng phần mềm độc hại (Malware attack), trong đó kẻ tấn công sử dụng virus, trojan, hoặc ransomware để xâm nhập và phá hoại hệ thống, là một trong những phương thức phổ biến nhất. Những phần mềm độc hại này có thể được cài đặt qua email lừa đảo, tải xuống tệp tin độc hại, hoặc qua các liên kết không an toàn, dẫn đến việc chiếm quyền điều khiển hệ thống hoặc đánh cắp thông tin cá nhân. Tấn công Man-in-the-Middle (MITM), khi kẻ tấn công chen ngang vào giao tiếp giữa hai bên mà

không bị phát hiện, cũng là một nguy cơ lớn đối với các giao dịch trực tuyến, dẫn đến việc đánh cắp thông tin đăng nhập, dữ liệu tài chính hoặc thông tin nhạy cảm khác. Tương tự, khai thác lỗ hổng Zero-day (Zero-day attack) là khi các lỗ hổng chưa được công khai hoặc khắc phục bị lợi dụng, tạo điều kiện cho kẻ xấu tấn công hệ thống mà chưa có bất kỳ biện pháp phòng ngừa nào. Các cuộc tấn công từ chối dịch vụ (DDoS) cũng đặc biệt nguy hiểm, khi các hệ thống hoặc trang web bị quá tải bởi lượng truy cập ảo khổng lồ, làm gián đoạn dịch vụ và gây tổn thất kinh tế cho các tổ chức.

Những phương thức tấn công mạng này không chỉ thể hiện sự đa dạng về kỹ thuật mà còn khai thác những điểm yếu về cả công nghệ lẫn con người, làm cho việc bảo vệ hệ thống ngày càng khó khăn hơn. Khi một cuộc tấn công thành công, ảnh hưởng đối với bảo mật thông tin là rất nghiêm trọng. Trước hết, các cuộc tấn công mạng có thể dẫn đến rò rỉ dữ liệu nhạy cảm, bao gồm thông tin cá nhân, tài chính, và thông tin nội bộ của doanh nghiệp. Những thông tin này, nếu rơi vào tay kẻ xấu, có thể bị sử dụng để thực hiện các hành vi lừa đảo, đánh cắp danh tính hoặc bán cho các bên thứ ba trên thị trường chợ đen, gây thiệt hại nghiêm trọng cho cá nhân và tổ chức.

Không chỉ dừng lại ở việc rò rỉ thông tin, các cuộc tấn công mạng còn gây ra gián đoạn hoạt động của hệ thống thông tin. Ví dụ, các cuộc tấn công từ chối dịch vụ (DDoS) có thể làm tê liệt toàn bộ hệ thống mạng, khiến cho các dịch vụ trực tuyến không thể hoạt động, ảnh hưởng đến hoạt động kinh doanh và gây mất mát về doanh thu. Đối với các tổ chức lớn như ngân hàng, bệnh viện, hay các cơ quan chính phủ, sự gián đoạn này có thể gây ra những hệ lụy nghiêm trọng, bao gồm cả việc gián đoạn cung cấp dịch vụ thiết yếu cho khách hàng.

Một hậu quả khác từ các cuộc tấn công mạng là không đảm bảo được tính toàn vẹn của dữ liệu. Kẻ tấn công có thể không chỉ đánh cắp mà còn chỉnh sửa hoặc phá hủy dữ liệu quan trọng, gây khó khăn trong việc khôi phục thông tin

và quản lý dữ liệu. Điều này ảnh hưởng đến tính chính xác và độ tin cậy của dữ liệu, dẫn đến những quyết định sai lầm hoặc mất mát dữ liệu vĩnh viễn.

Bên cạnh đó, một trong những ảnh hưởng lớn nhất của các cuộc tấn công mạng là mất niềm tin của người dùng. Khi thông tin cá nhân hoặc dữ liệu của khách hàng bị đánh cắp, người dùng có thể mất lòng tin vào hệ thống bảo mật của doanh nghiệp hoặc dịch vụ mà họ sử dụng. Điều này không chỉ ảnh hưởng đến danh tiếng của tổ chức, mà còn dẫn đến thiệt hại về lâu dài khi người dùng quyết định từ bỏ dịch vụ, làm giảm doanh thu và uy tín của doanh nghiệp. Đối với các công ty lớn, sự sụt giảm niềm tin từ phía khách hàng có thể mất rất nhiều thời gian để khắc phục.

### 1.1.5. Phương pháp phát hiện tấn công

Từ những hậu quả nghiêm trọng của các cuộc tấn công mạng như rò rỉ dữ liệu nhạy cảm, gián đoạn hoạt động, mất tính toàn vẹn dữ liệu, và làm suy giảm niềm tin của người dùng, có thể thấy rằng việc đảm bảo an toàn cho các hệ thống thông tin là một vấn đề phức tạp trong thời đại hiện nay. Các cuộc tấn công ngày càng tinh vi, có khả năng vượt qua những lớp bảo mật truyền thống và nhắm vào những điểm yếu mới phát sinh. Trong khi các biện pháp khắc phục sau khi bị tấn công giúp giảm thiểu thiệt hại, thì những tổn thất về dữ liệu, uy tín, và tài chính có thể rất khó khôi phục hoàn toàn. Chính vì vậy, việc phát hiện tấn công mạng sớm và hiệu quả trở thành yếu tố vô cùng quan trọng.

Việc phát hiện tấn công mạng không chỉ đơn thuần là phát hiện khi cuộc tấn công đã xảy ra mà còn phải nhận diện được các dấu hiệu tấn công từ sớm, từ đó ngăn chặn kịp thời trước khi kẻ xấu có thể gây ra thiệt hại. Điều này đòi hỏi các hệ thống bảo mật phải có khả năng giám sát liên tục các hoạt động trên mạng, phát hiện các hành vi bất thường và tự động cảnh báo khi phát hiện các dấu hiệu tiềm ẩn. Các cuộc tấn công thường để lại những dấu vết nhỏ ban đầu, từ những hành vi truy cập đáng ngờ, tăng đột biến lưu lượng mạng, hoặc việc

gửi các gói dữ liệu không bình thường. Khi phát hiện kịp thời các dấu hiệu này, tổ chức có thể ngăn chặn cuộc tấn công ngay từ giai đoạn đầu, giảm thiểu rủi ro mất mát và gián đoạn.

Để thực hiện việc này, các công cụ và hệ thống phát hiện tấn công mạng hiện đại được sử dụng rộng rãi, bao gồm hệ thống phát hiện xâm nhập (IDS), hệ thống ngăn chặn xâm nhập (IPS), cũng như các nền tảng phân tích dữ liệu lớn (Big Data) và trí tuệ nhân tạo (AI). IDS và IPS giám sát lưu lượng mạng và cảnh báo khi phát hiện những hành vi không bình thường hoặc đáng ngờ, giúp quản trị viên có thể phản ứng ngay lập tức. AI và học máy (machine learning) hiện đang được ứng dụng rộng rãi để phân tích hành vi, nhận diện các mẫu tấn công phức tạp mà các phương pháp truyền thống khó phát hiện, chẳng hạn như các cuộc tấn công Zero-day hay các mối đe dọa chưa từng biết trước đó.

Ngoài ra, giám sát hành vi người dùng là một phương pháp phát hiện tấn công hiệu quả. Công nghệ này tập trung vào việc phát hiện các hoạt động bất thường từ phía người dùng, như việc truy cập không hợp lý, cố gắng truy cập vào dữ liệu nhạy cảm hoặc thực hiện các thao tác không thường thấy. Bằng cách so sánh hành vi hiện tại với các mô hình hành vi bình thường giúp phát hiện các cuộc tấn công nội gián hoặc tấn công mạng từ các tài khoản người dùng bị chiếm đoạt.

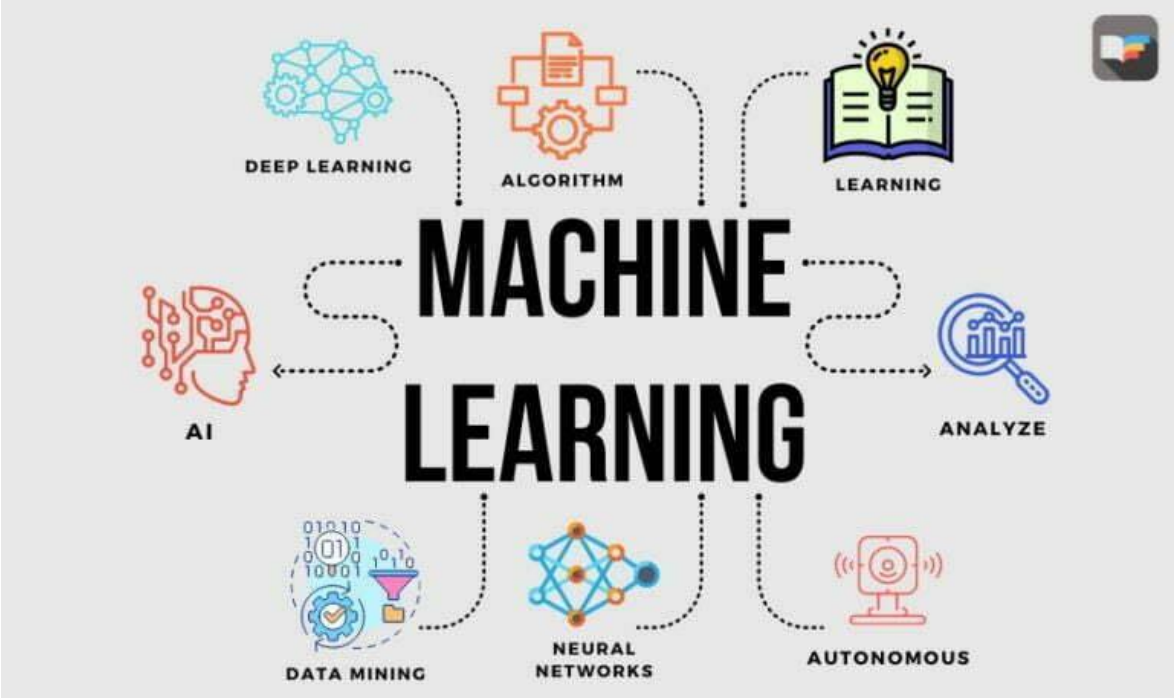
Trong nhiều trường hợp, việc phát hiện tấn công không chỉ dừng lại ở việc nhận diện các cuộc tấn công mạng thông thường mà còn mở rộng đến việc phát hiện các cuộc tấn công mục tiêu, như tấn công có chủ đích APT. Đây là các cuộc tấn công được lên kế hoạch kỹ lưỡng, thực hiện qua nhiều giai đoạn nhằm xâm nhập và duy trì sự hiện diện bên trong hệ thống mà không bị phát hiện. APT đòi hỏi phải có các biện pháp phát hiện tấn công phức tạp hơn, kết hợp nhiều phương pháp phân tích như giám sát hành vi, phân tích dấu vết tấn công,

và thậm chí cả thông tin tình báo mạng để hiểu rõ bối cảnh và mục tiêu của kẻ tấn công.

## 1.2. Học máy

### 1.2.1. Giới thiệu về học máy

Học máy là một lĩnh vực của trí tuệ nhân tạo liên quan đến việc nghiên cứu và xây dựng các kĩ thuật cho phép các hệ thống "học" tự động từ dữ liệu để giải quyết những vấn đề cụ thể. Các thuật toán học máy xây dựng một mô hình dựa trên dữ liệu mẫu, được gọi là dữ liệu huấn luyện, để đưa ra dự đoán hoặc quyết định mà không cần được lập trình chi tiết về việc đưa ra dự đoán hoặc quyết định này. Ví dụ như các máy có thể "học" cách phân loại thư điện tử xem có phải thư rác hay không và tự động xếp thư vào thư mục tương ứng. Học máy rất gần với suy diễn thống kê tuy có khác nhau về thuật ngữ. Một nhánh của học máy là học sâu phát triển rất mạnh mẽ gần đây và có những kết quả vượt trội so với các phương pháp học máy khác Học máy có liên quan lớn đến thống kê, vì cả hai lĩnh vực đều nghiên cứu việc phân tích dữ liệu, nhưng khác với thống kê, học máy tập trung vào sự phức tạp của các giải thuật trong việc thực thi tính toán.




*Hình 1.11: Machine Learning*

Hiện nay học máy được áp dụng rộng rãi bao gồm máy truy tìm dữ liệu, chẩn đoán y khoa, phát hiện thẻ tín dụng giả, phân tích thị trường chứng khoán, phân loại các chuỗi DNA, nhận dạng tiếng nói và chữ viết, dịch tự động, chơi trò chơi và cử động robot.

### 1.2.2. Lịch sử phát triển của học máy

Lịch sử phát triển của học máy có thể được chia thành các giai đoạn quan trọng từ khởi đầu đến nay, với những bước tiến lớn trong việc phát triển lý thuyết, thuật toán và ứng dụng thực tiễn.

**Giai đoạn khởi đầu (1940 - 1950)**

Học máy bắt nguồn từ khái niệm về trí tuệ nhân tạo (AI) và toán học. Các nhà khoa học bắt đầu nghiên cứu về cách máy móc có thể học từ dữ liệu.

- 1943: Warren McCulloch và Walter Pitts đưa ra mô hình toán học đầu tiên cho một neuron, nền tảng của mạng nơ-ron nhân tạo.

- 1950: Alan Turing đề xuất "Turing Test", đánh dấu mối quan tâm của ông về việc máy móc có thể mô phỏng quá trình tư duy của con người.
- 1957: Perceptron, một thuật toán học nơ-ron đơn giản, được phát triển bởi Frank Rosenblatt, đặt nền tảng cho các mô hình mạng nơ-ron hiện đại.

**Giai đoạn khám phá và nghiên cứu (1960 - 1980)**

- 1960: Học máy bắt đầu được áp dụng trong một số lĩnh vực như trò chơi và xử lý ngôn ngữ tự nhiên. Tuy nhiên, thuật toán perceptron bị giới hạn khi chỉ giải quyết được các bài toán tuyến tính, dẫn đến thời kỳ tạm lắng trong nghiên cứu mạng nơ-ron.
- 1970 - 1980: Lý thuyết học và thuật toán về tối ưu hóa bắt đầu được phát triển. Các thuật toán học không giám sát như K-means và phương pháp học có giám sát như cây quyết định, hồi quy logistic, và hồi quy tuyến tính được mở rộng.

**Giai đoạn tăng tốc (1990 - 2000)**

- 1990: Sự phát triển của các thuật toán như Support Vector Machines (SVM) và học máy dựa trên xác suất (như mạng Bayes và Hidden Markov Models) đã làm thay đổi cục diện của ngành.
- 1997: IBM Deep Blue đánh bại nhà vô địch cờ vua Garry Kasparov, minh chứng cho sức mạnh của AI và học máy.
- 2000: Các kỹ thuật học máy được áp dụng rộng rãi hơn nhờ vào sự phát triển của máy tính mạnh mẽ hơn, dữ liệu phong phú hơn và các thuật toán mới.
- Sự trỗi dậy của học sâu và AI hiện đại (2010 - nay)
- 2010: Sự ra đời và phổ biến của học sâu (deep learning) với các mô hình mạng nơ-ron phức tạp, đặc biệt là mạng nơ-ron tích chập (CNN) và mạng nơ-ron tuần tự (RNN). Các mô hình này đạt được kết quả ấn tượng trong

các lĩnh vực như thị giác máy tính, nhận dạng giọng nói, và xử lý ngôn ngữ tự nhiên.

- 2012: Mạng nơ-ron sâu của Geoffrey Hinton và nhóm nghiên cứu chiến thắng trong cuộc thi ImageNet, đánh dấu một cột mốc quan trọng.
- 2016: AlphaGo của Google DeepMind đánh bại Lee Sedol trong môn cờ vây, sử dụng các phương pháp học máy tiên tiến.
- 2017-nay: Các mô hình học máy như GPT (Generative Pretrained Transformer) và BERT đã tạo ra cuộc cách mạng trong việc xử lý ngôn ngữ tự nhiên và trí tuệ nhân tạo nói chung.

**Tương lai của học máy**

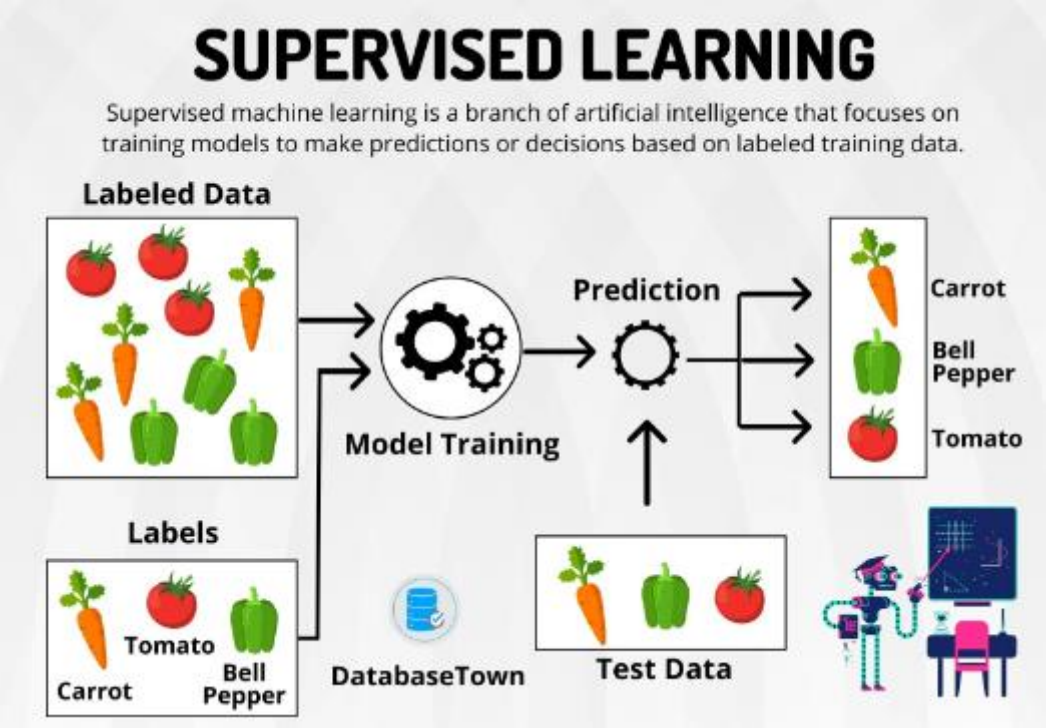
Các thuật toán học máy đang ngày càng trở nên mạnh mẽ hơn với khả năng xử lý khối lượng lớn dữ liệu, học từ các tình huống phức tạp, và tự động hóa quy trình ra quyết định. Học máy hiện tại cũng đang được kết hợp với các công nghệ như trí tuệ nhân tạo tăng cường (reinforcement learning) và lượng tử (quantum computing) để tạo ra những đột phá mới.

### 1.2.3. Các thuật toán học máy phổ biến

Học máy bao gồm nhiều thuật toán khác nhau, mỗi loại có những đặc điểm riêng và phù hợp với các bài toán cụ thể. Dưới đây là các thuật toán học máy phổ biến, được chia thành ba nhóm chính: học có giám sát, học không giám sát, và học tăng cường.

#### *1.2.3.1. Học máy có giám sát (Supervised Learning)*

Học máy có giám sát là phương pháp học máy dựa trên các cặp dữ liệu đầu vào và đầu ra có nhãn. Thuật toán được cung cấp một tập dữ liệu bao gồm các đặc trưng và các nhãn tương ứng, và nhiệm vụ của nó là học từ dữ liệu này để đưa ra dự đoán cho các dữ liệu mới chưa biết nhãn.




*Hình 1.12: Học máy có giám sát*

Học máy có giám sát được xây dựng để khi dữ liệu được đưa vào, chúng sẽ điều chỉnh trọng lượng của tệp dữ liệu sao cho phù hợp nhất, nhằm chắc chắn rằng mô hình sẽ chỉ nhận thông tin ở mức vừa đủ, tránh quá tải hoặc thiếu thông tin.

Học máy có giám sát được ứng dụng một số phương pháp như: Naive Bayes, Hồi quy logistic, thuật toán SVM, mạng nơ-ron,…Trong đó, một ứng dụng thực tiễn có thể kể đến như học máy giám sát giúp phân loại thư rác từ hộp thư đến vào một thư mục riêng biệt.

**Hồi quy tuyến tính (Linear Regression)**

- Cách hoạt động: Hồi quy tuyến tính là thuật toán dự đoán các giá trị liên tục. Nó cố gắng mô hình hóa mối quan hệ giữa các biến đầu vào (biến độc lập) và biến đầu ra (biến phụ thuộc) bằng cách khớp một đường thẳng vào dữ liệu.

- Ứng dụng: Dự đoán giá trị bất động sản, dự báo nhu cầu thị trường, dự đoán doanh thu kinh doanh.

**Hồi quy logistic (Logistic Regression)**

- Cách hoạt động: Thuật toán này chủ yếu được sử dụng cho các bài toán phân loại nhị phân (binary classification). Nó không đưa ra giá trị số liên tục như hồi quy tuyến tính, mà là xác suất thuộc về một trong hai lớp.
- Ứng dụng: Phân loại thư điện tử thành spam hoặc không spam, dự đoán bệnh có hay không dựa trên các thông số sức khỏe, phân loại khách hàng tiềm năng.

**Cây quyết định (Decision Tree)**

- Cách hoạt động: Cây quyết định là một mô hình phân loại dựa trên một chuỗi các câu hỏi có/không (yes/no) để đi đến một kết quả cuối cùng. Cấu trúc cây này giúp đơn giản hóa quá trình ra quyết định.
- Ứng dụng: Đánh giá rủi ro tín dụng, chẩn đoán y khoa, chiến lược tiếp thị khách hàng.

**Rừng ngẫu nhiên (Random Forest)**

- Cách hoạt động: Đây là một thuật toán học theo tổ hợp (ensemble learning) kết hợp nhiều cây quyết định để tạo thành một mô hình mạnh mẽ hơn. Nó cải thiện độ chính xác và giúp tránh overfitting, đặc biệt khi dữ liệu có tính đa dạng.
- Ứng dụng: Phân loại hình ảnh, phân tích dữ liệu tài chính, dự đoán gian lận.

**K-Nearest Neighbors (K-NN)**

- Cách hoạt động: K-NN là một thuật toán dựa trên khoảng cách. Để dự đoán lớp của một điểm dữ liệu mới, nó kiểm tra lớp của các điểm gần nhất trong dữ liệu huấn luyện. Điểm dữ liệu mới được phân vào lớp mà hầu hết các điểm gần nhất thuộc về.

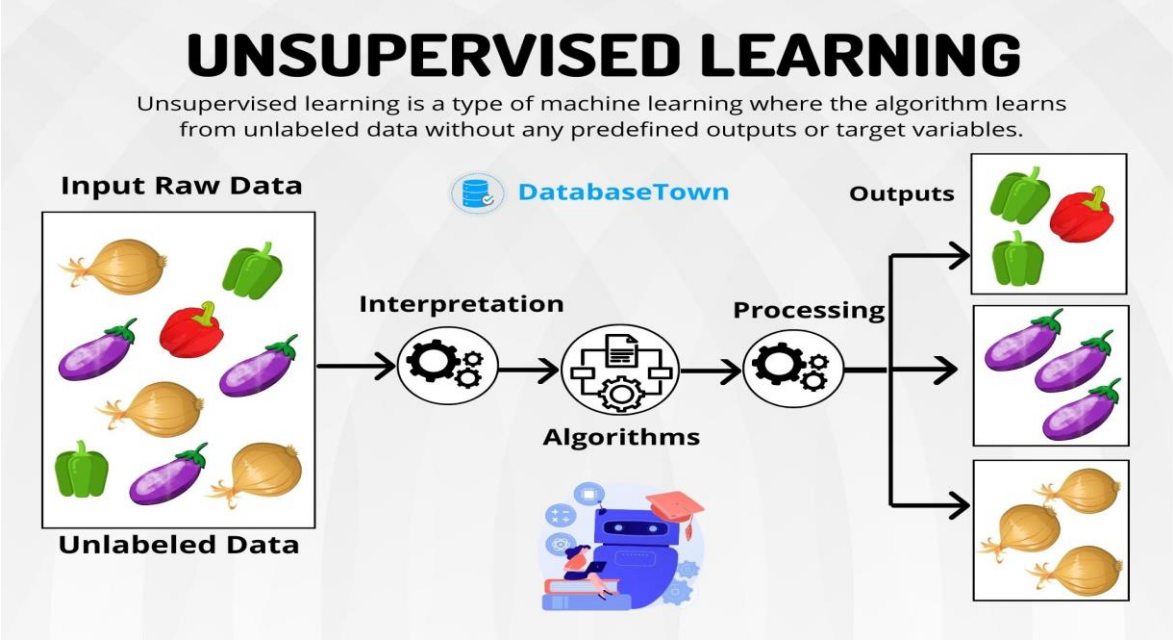
- Ứng dụng: Phân loại hình ảnh, nhận diện ký tự quang học (OCR), phân loại khách hàng.

**Mạng nơ-ron nhân tạo (Artificial Neural Networks - ANN)**

- Cách hoạt động: Mạng nơ-ron nhân tạo (ANN) là mô hình lấy cảm hứng từ cấu trúc của não người, bao gồm các nơ-ron kết nối với nhau. ANN có khả năng học các mô hình phức tạp thông qua các lớp nơ-ron ẩn, đặc biệt hữu ích trong việc xử lý dữ liệu lớn và phức tạp.
- Ứng dụng: Nhận diện giọng nói, xử lý ngôn ngữ tự nhiên, xe tự hành, phân tích hình ảnh y tế.

### *1.2.3.2. Học máy không giám sát (Unsupervised Learning)*

Nếu học máy có giám sát thực hiện phân tích và dự đoán kết quả dựa trên các dữ liệu đã gắn nhãn, thì học máy không giám sát sẽ thực hiện công việc đó dựa trên các dữ liệu không được gắn nhãn. Các thuật toán phải tự học cấu trúc dữ liệu, tìm kiếm các mẫu hoặc nhóm trong dữ liệu. Các thuật toán này thường được sử dụng để khám phá cấu trúc tiềm ẩn hoặc giảm kích thước dữ liệu.




*Hình 1.13: Học máy không giám sát*

Học máy không giám sát được sử dụng để giảm số lượng tính năng trong một mô hình thông qua quá trình giảm kích thước. Phân tích thành phần chính (PCA) và phân tích giá trị đơn lẻ (SVD) là hai cách tiếp cận phổ biến cho việc này. Các thuật toán khác được sử dụng trong machine learning không giám sát bao gồm neural network, phân cụm K-mean, phương pháp phân nhóm xác suất.

**Phân cụm K-Means (K-Means Clustering)**

- **Cách hoạt động:** K-Means là một trong những thuật toán phân cụm phổ biến nhất. Nó chia dữ liệu thành K nhóm dựa trên tính tương đồng giữa các điểm. Mỗi cụm được đại diện bởi trung tâm (centroid) của nó, và các điểm dữ liệu được gán vào cụm có centroid gần nhất.
- **Ứng dụng:** Phân khúc khách hàng, nhận diện hình mẫu trong dữ liệu, phát hiện bất thường.

**Phân tích thành phần chính (Principal Component Analysis - PCA)**

- **Cách hoạt động:** PCA là một kỹ thuật giảm chiều dữ liệu, giúp giảm số lượng biến đầu vào mà vẫn giữ được phần lớn thông tin trong dữ liệu. PCA tạo ra các thành phần chính (principal components) - là các biến mới được xây dựng từ các biến gốc.
- **Ứng dụng:** Giảm chiều dữ liệu trong xử lý ảnh, xử lý tín hiệu, nhận diện khuôn mặt.

**Phân cụm phân cấp (Hierarchical Clustering)**

- **Cách hoạt động:** Phân cụm phân cấp xây dựng một cây phân cấp (dendrogram) để mô hình hóa quan hệ giữa các điểm dữ liệu. Các cụm con được tạo dần dần từ dưới lên, hoặc từ trên xuống, cho đến khi đạt được số cụm mong muốn.
- **Ứng dụng:** Phân loại tài liệu, phân nhóm gen, phân tích thị trường.

**Thuật toán Apriori**

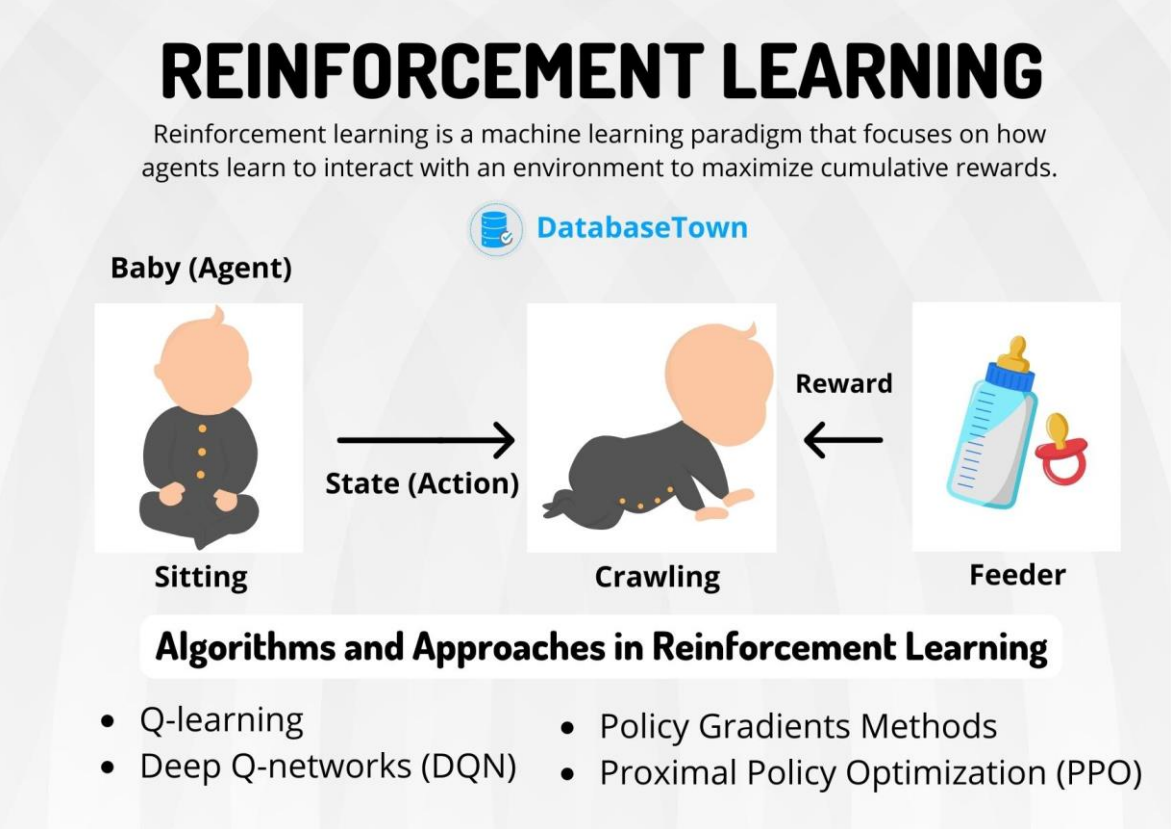
- **Cách hoạt động:** Thuật toán Apriori chủ yếu được sử dụng để khai thác các mẫu phổ biến trong các tập dữ liệu giao dịch lớn. Nó tìm kiếm các kết hợp thường xuất hiện cùng nhau trong dữ liệu, từ đó đưa ra các luật liên kết (association rules).
- **Ứng dụng:** Phân tích giỏ hàng (market basket analysis), hệ thống đề xuất sản phẩm, khai thác dữ liệu khách hàng.

**Mô hình hỗn hợp Gaussian (Gaussian Mixture Model - GMM)**

- **Cách hoạt động:** GMM là một mô hình xác suất giả định rằng dữ liệu được sinh ra từ sự kết hợp của nhiều phân phối Gaussian khác nhau. Nó cho phép phân cụm mềm (soft clustering), nơi một điểm dữ liệu có thể thuộc về nhiều cụm với các xác suất khác nhau.
- **Ứng dụng:** Phân tích âm thanh, phân loại tài liệu, xử lý ảnh y khoa.

#### *1.2.3.3. Học tăng cường (Reinforcement Learning)*

Học tăng cường là một nhánh của học máy tập trung vào việc đưa ra quyết định để tối đa hóa phần thưởng tích lũy trong một tình huống nhất định. Không giống như học máy có giám sát, dựa trên tập dữ liệu đào tạo với các câu trả lời được xác định trước, học máy tang cường liên quan đến việc học thông qua kinh nghiệm. Trong học máy tăng cường, nếu một tác nhân học cách đạt được mục tiêu trong một môi trường không chắc chắn, có khả năng phức tạp bằng cách thực hiện các hành động và nhận phản hồi thông qua phần thưởng hoặc hình phạt.




*Hình 1.14: Học máy tăng cường*

**Q-learning**

- **Cách hoạt động:** Q-learning là một thuật toán không cần mô hình (model-free), trong đó agent học cách hành động bằng cách tối ưu hàm giá trị Q, mô tả giá trị kỳ vọng của một hành động cụ thể trong một trạng thái nhất định.
- **Ứng dụng:** Robot học tự hành động, học chơi game, quản lý tài nguyên trong hệ thống.

**Deep Q-Network - (DQN)**

- **Cách hoạt động:** DQN là sự kết hợp của Q-learning và mạng nơ-ron sâu. Thay vì sử dụng bảng Q đơn giản, DQN sử dụng mạng nơ-ron để xấp xỉ hàm giá trị Q, giúp xử lý các môi trường phức tạp với không gian trạng thái lớn.

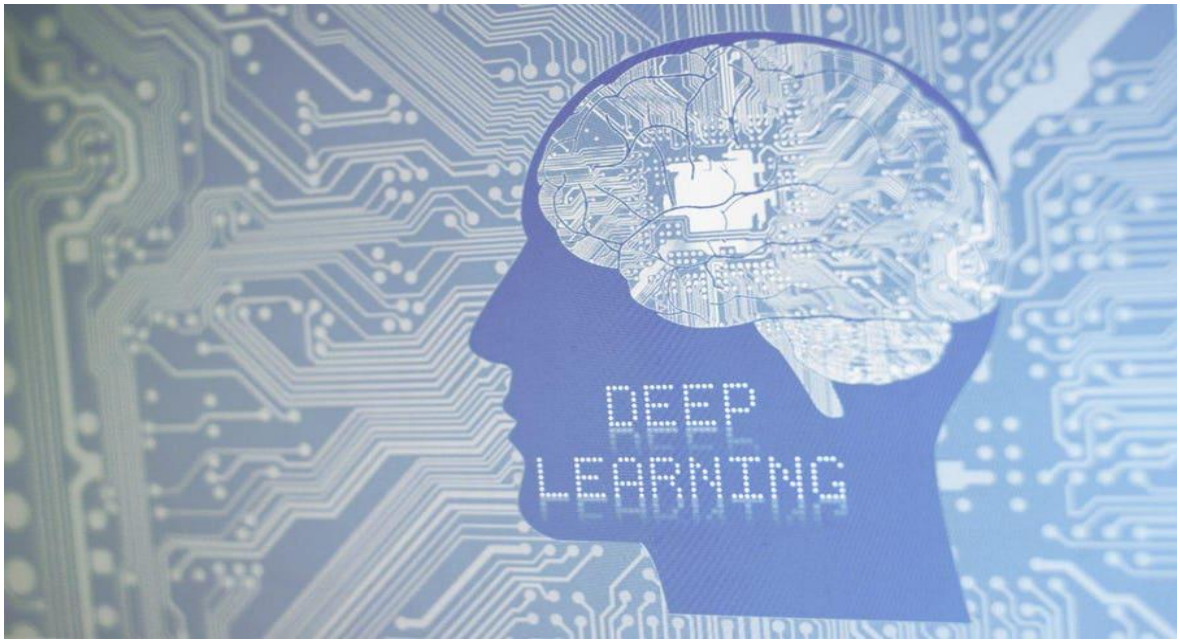
- **Ứng dụng:** Game AI (như AlphaGo), điều khiển robot, quản lý năng lượng.

**State Action Reward State Action – (SARSA)**

- **Cách hoạt động:** Giống như Q-learning, SARSA cũng là một thuật toán học tăng cường. Điểm khác biệt là SARSA cập nhật giá trị Q dựa trên hành động mà agent thực sự chọn thay vì hành động tối ưu nhất có thể như Q-learning.
- **Ứng dụng:** Điều khiển hệ thống động, học chiến lược trong các bài toán phức tạp.

### *1.2.3.4. Học sâu (Deep Learning)*

Học sâu là một tập hợp con của học máy sử dụng mạng nơ-ron nhiều lớp, được gọi là mạng nơ-ron sâu, để mô phỏng khả năng ra quyết định phức tạp của não người. Một số dạng học sâu hỗ trợ hầu hết các ứng dụng trí tuệ nhân tạo trong cuộc sống của chúng ta ngày nay.




*Hình 1.15: Hình ảnh về học sâu*

Sự khác biệt chính giữa học sâu và học máy là cấu trúc của kiến trúc mạng nơ-ron cơ bản. Các mô hình học máy truyền thống "Nondeep" sử dụng mạng nơ-ron đơn giản với một hoặc hai lớp tính toán. Các mô hình học sâu sử dụng ba lớp trở lên nhưng thường là hàng trăm hoặc hàng nghìn lớp để đào tạo các mô hình.

Trong khi các mô hình học có giám sát yêu cầu dữ liệu đầu vào có cấu trúc, được gắn nhãn để tạo ra đầu ra chính xác, các mô hình học sâu có thể sử dụng học không giám sát. Với học không giám sát, các mô hình học sâu có thể trích xuất các đặc điểm, tính năng và mối quan hệ mà chúng cần để tạo ra đầu ra chính xác từ dữ liệu thô, không có cấu trúc. Ngoài ra, các mô hình này thậm chí có thể đánh giá và tinh chỉnh đầu ra của chúng để tăng độ chính xác.

Học sâu là một khía cạnh của khoa học dữ liệu thúc đẩy nhiều ứng dụng và dịch vụ cải thiện tự động hóa, thực hiện các nhiệm vụ phân tích và vật lý mà không cần sự can thiệp của con người. Điều này cho phép nhiều sản phẩm và dịch vụ hàng ngày chẳng hạn như trợ lý kỹ thuật số, điều khiển từ xa TV hỗ trợ giọng nói, phát hiện gian lận thẻ tín dụng, xe tự lái và AI tạo ra.

**Feedforward Neural Network – (FNN)**

- **Cách hoạt động**: FNN là loại mạng đơn giản nhất. Thông tin được truyền từ đầu vào, qua một hoặc nhiều lớp ẩn, đến đầu ra mà không có vòng lặp. Mỗi nơ-ron trong một lớp kết nối đầy đủ với các nơ-ron của lớp tiếp theo. Các trọng số của các kết nối này sẽ được điều chỉnh thông qua quá trình huấn luyện để giảm sai số giữa dự đoán và kết quả thực tế.
- **Ứng dụng**: Phân loại hình ảnh, nhận dạng chữ viết tay, nhận dạng giọng nói cơ bản, dự báo tài chính và phân tích dữ liệu đơn giản.

**Convolutional Neural Network – (CNN)**

- **Cách hoạt động**: CNN sử dụng các lớp tích chập để tự động học các đặc trưng từ dữ liệu. Các lớp này giúp phát hiện các mẫu như cạnh, góc, hay chi tiết cụ thể trong hình ảnh. Thay vì kết nối đầy đủ như FNN, CNN chỉ kết nối một phần nhỏ, giúp giảm số lượng tham số và xử lý dữ liệu có cấu trúc như hình ảnh hiệu quả hơn. Lớp pooling giúp giảm kích thước không gian của các đặc trưng, trong khi lớp fully connected ở cuối được dùng để dự đoán.
- **Ứng dụng**: Nhận dạng khuôn mặt, phân loại hình ảnh, phát hiện vật thể, phân đoạn hình ảnh, thị giác máy tính, chẩn đoán y khoa từ hình ảnh (X-ray, MRI).

**Recurrent Neural Network – (RNN)**

- **Cách hoạt động**: RNN có cấu trúc vòng lặp giúp lưu giữ thông tin từ bước trước để sử dụng trong bước sau, phù hợp với dữ liệu chuỗi. Một nơ-ron trong RNN không chỉ nhận thông tin từ nơ-ron lớp trước mà còn từ chính nó. Điều này cho phép RNN xử lý các dữ liệu có sự phụ thuộc tuần tự như văn bản, âm thanh hay chuỗi thời gian.
- **Ứng dụng**: Dự báo chuỗi thời gian, nhận diện giọng nói, dịch ngôn ngữ tự động, phân tích cảm xúc trong văn bản, nhận diện hoạt động người dùng từ video.

**Long Short-Term Memory – (LSTM)**

- **Cách hoạt động**: LSTM là biến thể của RNN, giải quyết vấn đề của RNN khi không thể nhớ các thông tin dài hạn. LSTM sử dụng ba cửa (gates): cửa vào (input gate), cửa quên (forget gate), và cửa ra (output gate) để kiểm soát luồng thông tin qua các tế bào bộ nhớ (cell states).

Điều này giúp LSTM lưu giữ hoặc loại bỏ thông tin theo yêu cầu, cho phép học từ các chuỗi dữ liệu dài.

- **Ứng dụng**: Dự báo chuỗi thời gian dài hạn, dịch ngôn ngữ tự động, nhận dạng giọng nói liên tục, phân tích dữ liệu tài chính, xử lý âm nhạc và văn bản.

**Transformers**

- **Cách hoạt động**: Transformer dựa trên cơ chế attention, cho phép mỗi từ trong chuỗi có thể "chú ý" đến bất kỳ từ nào khác trong chuỗi, thay vì chỉ dựa vào thông tin từ từ liền kề như RNN. Mô hình này giúp xử lý chuỗi dữ liệu song song nhanh hơn và hiệu quả hơn so với các mô hình tuần tự. Nó có hai thành phần chính: Encoder để xử lý chuỗi đầu vào và Decoder để dự đoán chuỗi đầu ra.
- **Ứng dụng**: Mô hình ngôn ngữ lớn (GPT, BERT), dịch máy, trả lời câu hỏi, tóm tắt văn bản, phát hiện ngôn ngữ, xử lý ngôn ngữ tự nhiên (NLP) hiện đại.

**Mạng nơ-ron đối kháng (Generative Adversarial Network - GAN)**

- **Cách hoạt động**: GAN bao gồm hai mạng: Generator và Discriminator. Generator cố gắng tạo ra dữ liệu giả sao cho giống dữ liệu thật nhất có thể, trong khi Discriminator cố gắng phân biệt dữ liệu thật và giả. Hai mạng này "cạnh tranh" với nhau trong quá trình huấn luyện, và qua đó, Generator dần dần tạo ra dữ liệu giả có chất lượng cao hơn.
- **Ứng dụng**: Tạo hình ảnh, video, âm thanh giả (deepfake), tăng cường dữ liệu, tạo hình ảnh từ văn bản, mô phỏng dữ liệu cho các mô hình máy học, cải tiến ảnh chụp (ảnh cũ, ảnh mờ).

**Mạng nơ-ron tích lũy (Residual Neural Network - ResNet)**

- **Cách hoạt động**: ResNet giải quyết vấn đề về gradient bị suy giảm trong các mạng sâu. Thay vì chỉ truyền trực tiếp qua từng lớp, nó sử dụng các kết nối tắt (skip connections), cho phép thông tin có thể "nhảy qua" nhiều lớp mà không bị biến đổi quá nhiều. Điều này giúp ResNet có thể xây dựng các mạng rất sâu mà không mất đi thông tin.
- **Ứng dụng**: Nhận dạng và phân loại hình ảnh, phát hiện đối tượng trong hình ảnh, phân đoạn hình ảnh, các bài toán trong thị giác máy tính cần mạng học sâu hơn.

### 1.2.4. Ứng dụng của học máy trong lĩnh vực an ninh mạng

Học máy đang ngày càng trở thành một công cụ mạnh mẽ trong việc bảo mật mạng và phát hiện tấn công mạng nhờ khả năng tự động học từ dữ liệu và phát hiện các mẫu bất thường hoặc các mối đe dọa.

#### *1.2.4.1.Phát hiện tấn công mạng*

Học máy giúp nhận diện các mẫu hành vi bất thường trong dữ liệu mạng, từ đó phát hiện các cuộc tấn công như SQL Injection (SQLi), Cross-Site Scripting (XSS), và tấn công từ chối dịch vụ (DDoS). Phân tích log truy cập và lưu lượng mạng cho phép xác định các dấu hiệu của các mối đe dọa này sớm hơn, giúp ngăn chặn kịp thời.

#### *1.2.4.2. Phân tích hành vi người dùng và hệ thống*

Bằng cách phân tích sâu hành vi của người dùng và hệ thống, học máy giúp phát hiện các hành vi bất thường, giảm thiểu nguy cơ tấn công từ nội bộ, chẳng hạn như việc lạm dụng quyền truy cập hoặc các thiết bị bị xâm nhập.

#### *1.2.4.3. Phát hiện mã độc và phần mềm độc hại*

Các mô hình học máy có khả năng phân tích đặc trưng hành vi của mã độc và phần mềm độc hại, giúp phát hiện và ngăn chặn phần mềm độc hại trước khi chúng gây ra tổn hại.

#### *1.2.4.4. Phòng chống xâm nhập*

Các hệ thống phát hiện và phòng chống xâm nhập dựa trên học máy có thể phát hiện các cuộc tấn công đang diễn ra hoặc sắp diễn ra, từ đó tự động ngăn chặn hoặc đưa ra cảnh báo kịp thời.

#### *1.2.4.5. Dự đoán và ngăn ngừa mối đe dọa tương lai*

Dựa vào phân tích dữ liệu lịch sử, học máy có khả năng dự đoán các mối đe dọa tiềm ẩn trong tương lai, giúp các chuyên gia an ninh mạng xây dựng chiến lược phòng ngừa hiệu quả và giảm thiểu rủi ro.

#### *1.2.4.6. Phát hiện gian lận và giả mạo*

Trong các lĩnh vực như giao dịch trực tuyến và xác thực, học máy được sử dụng để phát hiện các hoạt động lừa đảo, nhận diện các giao dịch bất thường hoặc hành vi mạo danh, đảm bảo tính an toàn cho người dùng và hệ thống.

#### *1.2.4.7. Tự động hóa phản ứng với sự cố*

Học máy có thể tự động hóa quá trình phản ứng và xử lý sự cố an ninh mạng, giúp tăng tốc độ phản ứng và giảm tải cho các chuyên gia, từ đó nâng cao hiệu quả bảo mật cho hệ thống.

Nhờ khả năng phát hiện mẫu phức tạp và phân tích lượng dữ liệu lớn, học máy đã và đang trở thành một công cụ không thể thiếu trong việc bảo vệ hệ thống và dữ liệu khỏi các mối đe dọa an ninh mạng ngày càng tinh vi.

## 1.3. Kết luận chương

Chương 1 đã trình bày khái quát về an ninh mạng, lịch sử phát triển của an ninh mạng cũng như các phương thức tấn công phổ biến trong an ninh mạng từ đó sẽ cho thấy rõ được ảnh hưởng của các cuộc tấn công mạng đối với hệ thống. Bên cạnh đó giúp chúng ta hiểu được các phương pháp phát hiện tấn công mạng. Đồng thời trình bày tổng quát về học máy, các thuật toán học máy phổ biến và ứng dụng của học máy đối với lĩnh vực anh ninh mạng

# CHƯƠNG 2 - NGHIÊN CỨU PHƯƠNG PHÁP PHÁT HIỆN TẤN CÔNG WEB DỰA VÀO CÔNG NGHỆ HỌC MÁY

## 2.1. Phương pháp tiếp cận

Đối với phương pháp phát hiện tấn công web dựa trên phân tích log và học máy phương pháp tiếp cận chủ yếu tập trung vào việc thu thập và phân tích dữ liệu từ các file log của máy chủ web. Các file log này ghi lại chi tiết các yêu cầu truy cập như địa chỉ IP của người dùng, thời gian yêu cầu, phương thức HTTP, URL được truy cập, mã trạng thái HTTP, và kích thước phản hồi.

Những thông tin này có thể cung cấp nhiều dấu hiệu về hoạt động bất thường và nguy cơ tấn công. Để phân tích và nhận diện các mẫu hành vi bất thường trong log, nghiên cứu sẽ áp dụng các kỹ thuật học máy tiên tiến. Các mô hình học máy được lựa chọn sẽ tự động phân tích dữ liệu log và nhận diện các mẫu bất thường, giúp cảnh báo kịp thời về các cuộc tấn công tiềm ẩn.

Bằng cách sử dụng học máy, chúng ta có thể tự động hóa quá trình phát hiện tấn công, nâng cao hiệu quả và độ chính xác của hệ thống bảo mật, từ đó bảo vệ tốt hơn cho các ứng dụng web khỏi các mối đe dọa an ninh.

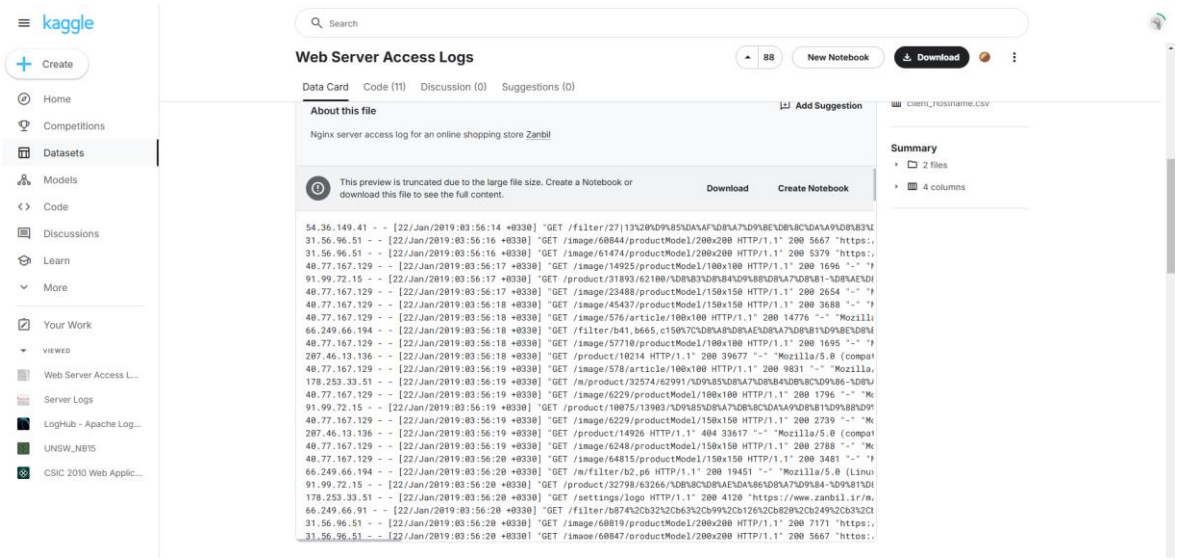
## 2.2. Thu thập dữ liệu đầu vào

Việc thu thập được một số lượng lớn về log của các trang web bình thường và bị tấn công là hết sức khó khăn, vì hiện chưa có kho lưu trữ dataset mẫu nào trên thế giới có sẵn về dữ liệu này.

### 2.2.1. Dữ liệu về các trang web bị tấn công

Dữ liệu từ các trang web bị tấn công là thành phần quan trọng để mô hình phát hiện tấn công website nhận diện được các mẫu hành vi bất thường trong log. Việc thu thập loại dữ liệu này không dễ dàng, vì đa số tổ chức không công khai thông tin liên quan đến các sự cố an ninh mạng của họ. Do đó, để xây dựng một tập dữ liệu đủ lớn và đa dạng, có thể áp dụng các phương pháp như:

- Sử dụng các bộ dữ liệu công khai

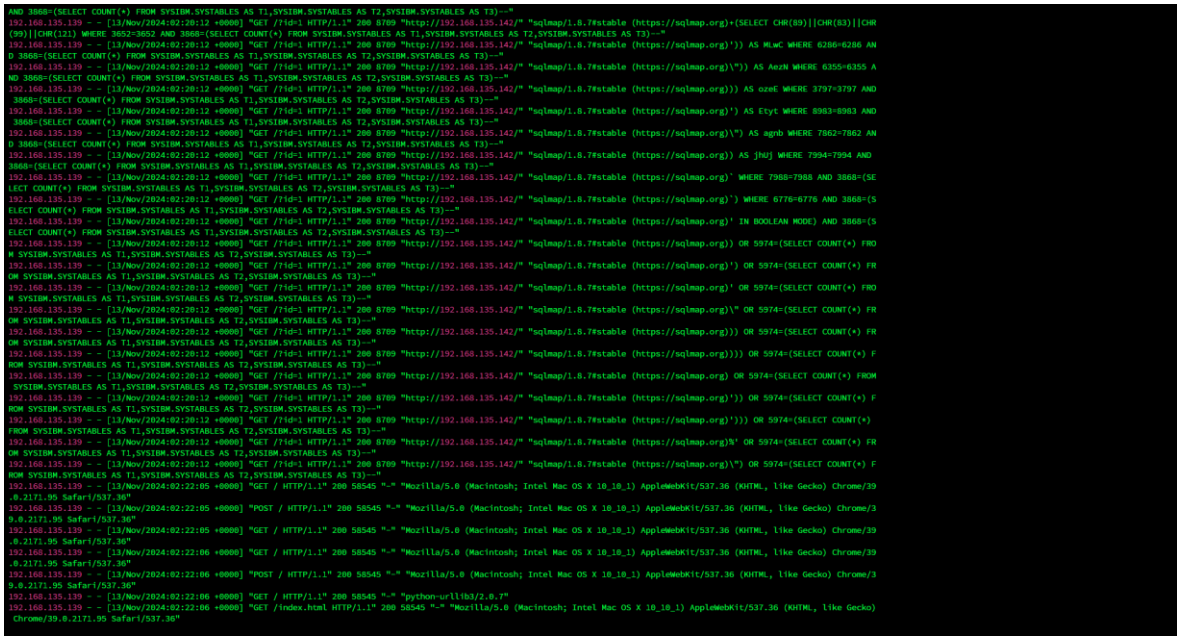



*Hình 2.1: Sử dụng bộ dữ liệu công khai*

Một số tổ chức hoặc nền tảng nghiên cứu an ninh mạng cung cấp các tập dữ liệu liên quan đến các kiểu tấn công phổ biến. Chẳng hạn, CICIDS (Canadian Institute for Cybersecurity) cung cấp nhiều bộ dữ liệu về các cuộc tấn công mạng, bao gồm các log mô phỏng các kiểu tấn công như SQL Injection, DDoS, và Brute Force. Kaggle cũng là một nguồn tham khảo hữu ích với các tập dữ liệu được cộng đồng chia sẻ. Tuy nhiên, các tập dữ liệu công khai này thường không đầy đủ và có thể không phù hợp hoàn toàn với log Apache.

- Mô phỏng tấn công trong môi trường thử nghiệm

Đây là một phương pháp hiệu quả và chủ động để thu thập log từ các trang web bị tấn công. Đầu tiên, cần triển khai một máy chủ Apache với cấu hình cơ bản hoặc có các lỗ hổng bảo mật có kiểm soát. Sau đó, sử dụng các công cụ như Metasploit, OWASP ZAP, hoặc SQLMap để thực hiện các cuộc tấn công phổ biến như SQL Injection, Cross-Site Scripting (XSS), hoặc Distributed Denial-of-Service (DDoS).



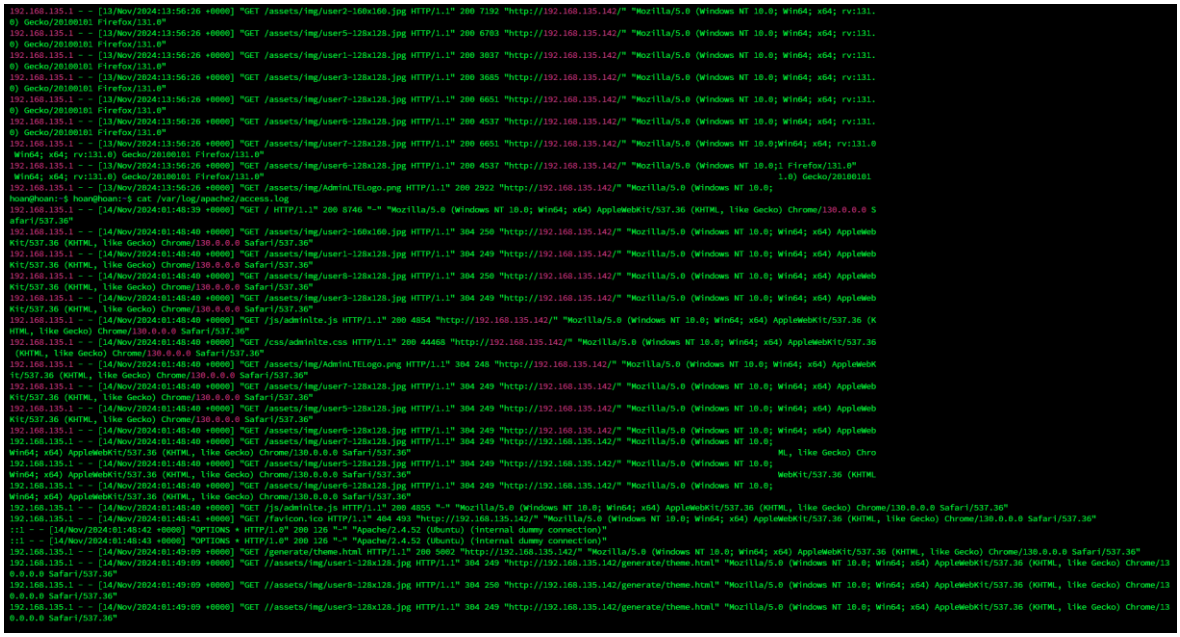

*Hình 2.2: Mô phỏng tấn công SQLi trên server*

Quá trình này giúp tạo ra một tập log đa dạng, phản ánh các kiểu tấn công khác nhau trong môi trường thực tế.

### 2.2.2. Dữ liệu về các trang web không bị tấn công

Log từ các trang web hoạt động bình thường là cơ sở để mô hình học máy hiểu được hành vi hợp lệ và phân biệt nó với các hành vi bất thường. Việc thu thập loại dữ liệu này dễ dàng hơn so với dữ liệu từ các trang web bị tấn công, nhưng vẫn cần đảm bảo chất lượng và tính đại diện của dữ liệu. Một số cách tiếp cận bao gồm:

- Thu thập log từ các website tự triển khai:




*Hình 2.3: Thu thập log từ server tự triển khai*

Có thể xây dựng một trang web mẫu, đảm bảo cấu hình bảo mật đầy đủ và không chứa các lỗ hổng. Sau đó, mời người dùng thực hiện các thao tác hợp lệ như duyệt trang, đăng nhập, tìm kiếm thông tin, và tải nội dung. Dữ liệu log Apache từ các hoạt động này sẽ cung cấp một cái nhìn rõ ràng về hành vi bình thường trên website.

- Sử dụng các log mẫu từ dịch vụ hosting:

Nhiều nhà cung cấp dịch vụ hosting như AWS, Google Cloud, hoặc DigitalOcean có thể cung cấp log từ các trang web không bị tấn công trong một thời gian dài. Các log này thường được ghi nhận tự động và có độ chính xác cao, nhưng vẫn cần kiểm tra kỹ để đảm bảo rằng không chứa các hành vi bất thường hoặc tiềm ẩn nguy cơ.

## 2.3. Lựa chọn và xây dựng mô hình học máy

Việc lựa chọn và xây dựng mô hình học máy là một trong những bước quan trọng trong việc phát triển hệ thống vì mô hình sẽ quyết định độ chính xác và hiệu quả trong việc nhận diện các cuộc tấn công.

### 2.3.1. Lựa chọn thuật toán học máy

Việc lựa chọn thuật toán học máy phù hợp phụ thuộc vào đặc điểm dữ liệu log của Apache và mục tiêu cụ thể của bài toán. Dữ liệu log ghi lại thông tin chi tiết về các yêu cầu truy cập như địa chỉ IP, thời gian, phương thức HTTP (GET, POST, DELETE), URL, mã trạng thái HTTP (200, 404, 500,...), và kích thước phản hồi. Các đặc trưng này cung cấp cơ sở để phân tích hành vi người dùng, phát hiện truy cập bất thường, dự đoán tải hệ thống hoặc phát hiện tấn công mạng.

Đối với mục tiêu phát hiện các yêu cầu bất thường hoặc hành vi tiềm ẩn nguy cơ, các thuật toán học không giám sát thường được ưu tiên do dữ liệu log hiếm khi có nhãn cụ thể. Các thuật toán phổ biến bao gồm K-means, DBSCAN, và Isolation Forest, mỗi thuật toán đều có ưu điểm trong việc nhận diện các hành vi khác biệt hoặc bất thường trong dữ liệu.

Khi bài toán yêu cầu phân loại cụ thể (có nhãn), các thuật toán học có giám sát như Logistic Regression, Random Forest, và Support Vector Machine trở thành lựa chọn hiệu quả. Các mô hình này đặc biệt hữu ích khi cần xác định hoặc dự đoán loại hành vi cụ thể dựa trên các đặc trưng đã được xử lý.

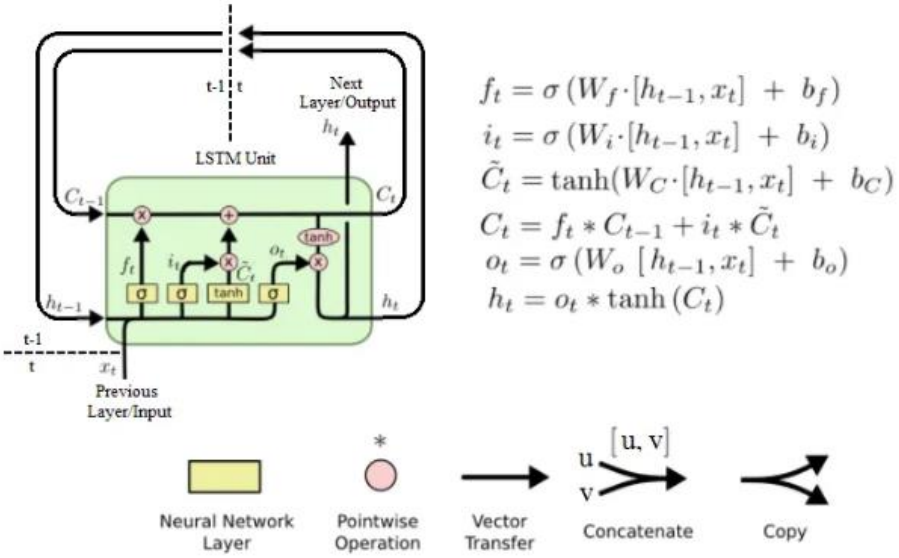
Tuy nhiên, với bài toán yêu cầu phân tích chuỗi thời gian hoặc dự báo, như dự đoán tải hệ thống hoặc xu hướng truy cập, mô hình LSTM được lựa chọn do khả năng vượt trội trong xử lý dữ liệu chuỗi thời gian phức tạp. LSTM, một biến thể của mạng nơ-ron hồi tiếp có thể lưu giữ thông tin dài hạn trong chuỗi và phát hiện các mẫu trong dữ liệu log. Điều này đặc biệt hữu ích khi phân tích các tương quan hoặc xu hướng trong dãy truy cập log theo thời gian.

Mặc dù các mô hình đơn giản như Linear Regression cũng có thể áp dụng để dự báo ngắn hạn nhưng mà LSTM vượt trội hơn trong việc xử lý các bài toán dự báo dài hạn với dữ liệu phức tạp. Sự lựa chọn này không chỉ giúp nắm

bắt các mối quan hệ phức tạp mà còn tăng độ chính xác khi phân tích dữ liệu log theo thời gian.

### 2.3.2. Kiến trúc mô hình LSTM

LSTM là một mạng hồi quy đặc biệt (RNN) được thiết kế cho các bài toán xử lý thông tin dạng chuỗi . LSTM được cho là có kết quả cao hơn rất nhiều so với RNN. Khác với RNN, LSTM chứa một số đơn vị đặc biệt gọi là các khối nhớ trong các lớp ẩn hồi quy. Các khối nhớ này sẽ chứa các ô nhớ được kết nối lẫn nhau để lưu thông tin trạng thái. Bên cạnh đó, một số đơn vị đặc biệt khác gọi là các cổng sẽ có nhiệm vụ kiểm soát các dòng thông tin. Mỗi khối nhớ bao gồm ba loại cổng: cổng đầu vào, cổng đầu ra và cổng quên.




*Hình 2.4: Kiến trúc mô hình LSTM*

#### *2.3.2.1. Cổng quên*

- **Chức năng**: Xác định thông tin nào trong trạng thái cũ $(C_{t-1})$ cần được giữ lại hoặc loại bỏ.
- **Công thức**: : $f_t = \sigma(\mathrm{W}_f \cdot [h_{t-1}, x_t] + b_f)$ (1)

Trong đó:

- $x_t$: đầu vào tại thời điểm t.
- $h_{t-1}$: trạng thái ẩn từ thời điểm trước đó.
- $W_f, b_f$: trọng số và bias của cổng quên.
- σ: hàm sigmoid.

#### *2.3.2.2. Cổng đầu vào*

- **Chức năng**: Quyết định thông tin mới nào từ đầu vào sẽ được thêm vào trạng thái nhớ hiện tại.
- **Công thức**:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \qquad (2)$$

và trạng thái mới:

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \qquad (3)$$

Trong đó:

- $i_t$: Đầu ra của cổng đầu vào (giá trị nằm trong khoảng 0-1).
- $\tilde{C}_t$: Giá trị thông tin mới được tính toán.

#### *2.3.2.3. Cổng đầu ra*

- **Chức năng**: Xác định trạng thái ẩn ($h_t$) được xuất ra từ LSTM.
- **Công thức**:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \qquad (4)$$

và trạng thái ẩn:

$$h_t = o_t \cdot \tanh(C_t)$$

Trong đó:

- $o_t$: Đầu ra của cổng đầu ra.
- $h_t$: Trạng thái ẩn tại thời điểm t.

#### *2.3.2.4. Trạng thái bộ nhớ*

- Trạng thái nhớ $C_t$ được cập nhật bởi công thức:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \qquad (5)$$

### 2.3.3. Cơ chế hoạt động của mô hình LSTM

Cơ chế hoạt động của mô hình LSTM được tổ chức theo từng bước thời gian ($t$) với sự truyền thông tin qua các trạng thái.

#### *2.3.3.1. Tiếp nhận đầu vào tại mỗi bước thời gian*

- Đầu vào $x_t$: Là dữ liệu tại thời điểm t. Ví dụ: một từ trong câu hoặc một giá trị trong chuỗi thời gian.
- Trạng thái ẩn trước đó $h_{t-1}$**:** Mang thông tin từ các bước thời gian trước.
- Trạng thái ô nhớ trước đó $C_{t-1}$ : Lưu trữ thông tin dài hạn.

#### *2.3.3.2. Các phép tính bên trong một LSTM cell*

**(a) Tính toán cổng quên ($f_t$)**

- Mục tiêu là quyết định giữ lại hay loại bỏ thông tin từ $C_{t-1}$
- $f_t$*:* Là một vectơ giá trị nằm trong khoảng (0, 1), đại diện cho mức độ "giữ lại" từng phần của $C_{t-1}$.

**(b) Tính toán cổng đầu vào ($i_t$) và thông tin mới ($\widetilde{C}_t$)**

Mục tiêu là xác định thông tin mới nào sẽ được thêm vào ô nhớ.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (6)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (7)$$

- $i_t$: Xác định phần thông tin nào từ $\widetilde{C}_t$ sẽ được lưu.
- $\tilde{C}_t$: Là thông tin mới được tính toán từ $x_t$ và $h_{t-1}$.

**(c) Cập nhật trạng thái ô nhớ ($C_{t-1}$)**

Kết hợp thông tin mới và thông tin được giữ lại từ $C_{t-1}$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \quad (8)$$

- $C_t$: Trạng thái ô nhớ cập nhật, mang thông tin dài hạn.

**(d) Tính toán cổng đầu ra ($o_t$) và trạng thái ẩn ($h_t$)**

Mục tiêu là xác định phần nào của trạng thái $C_t$ sẽ được "lấy ra" làm đầu ra hiện tại.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (9)$$

$$h_t = o_t \cdot \tanh(C_t) \quad (10)$$

- $h_t$: Trạng thái ẩn, dùng làm thông tin đầu ra và truyền sang bước thời gian tiếp theo.

***2.3.3.3. Truyền thông tin qua các bước thời gian***

- Tại mỗi bước t, $h_t$ (trạng thái ẩn) được dùng để:

- Truyền thông tin sang bước t + 1.
- Thực hiện dự đoán hoặc xử lý theo yêu cầu bài toán.

- $C_t$ đảm bảo rằng các thông tin quan trọng được duy trì qua nhiều bước thời gian.

#### *2.3.3.4. Cơ chế tóm gọn*

- Nhận thông tin từ bước trước ($x_t$ , $h_{t-1}$ , $C_{t-1}$)
- Tính toán các cổng ($f_t$, $i_t$, $o_t$) và trạng thái ô nhớ ($C_t$).
- Tạo đầu ra ($h_t$) và truyền thông tin sang bước kế tiếp.

## 2.4. Xây dựng mô hình học máy

Quá trình xây dựng mô hình học máy được chia thành các bước chính bao gồm chuẩn bị dữ liệu, lựa chọn mô hình, huấn luyện, và đánh giá hiệu suất. Đây là các bước cụ thể để xây dựng mô hình phát hiện tấn công mạng dựa trên dữ liệu log của Apache.

**Chuẩn bị dữ liệu**

- Thu thập dữ liệu: Tập hợp dữ liệu log của Apache từ hệ thống thực tế, hoặc từ các bộ dữ liệu có sẵn bao gồm các thông tin như địa chỉ IP, thời gian, phương thức HTTP, URL được truy cập, mã trạng thái HTTP, và kích thước phản hồi.

**Tiền xử lý dữ liệu**

- Loại bỏ dữ liệu nhiễu: Xóa bỏ các bản ghi không cần thiết hoặc không hợp lệ (như bản ghi bị thiếu thông tin).
- Chuẩn hóa dữ liệu: Mã hóa địa chỉ IP, chuyển đổi thời gian sang định dạng số, và chuẩn hóa các giá trị như mã trạng thái hoặc kích thước phản hồi về cùng một thang đo.

- Tạo đặc trưng: Trích xuất các đặc trưng quan trọng từ dữ liệu log, như số lượng yêu cầu từ một địa chỉ IP trong một khoảng thời gian nhất định hoặc tỷ lệ mã trạng thái lỗi.

**Lựa chọn mô hình**

Dựa trên phân tích yêu cầu bài toán, lựa chọn mô hình LSTM được đánh giá là phù hợp để xử lý dữ liệu log theo chuỗi thời gian. Mô hình này có khả năng:

- Ghi nhớ các trạng thái trước đó trong chuỗi dữ liệu.
- Nắm bắt các mối quan hệ dài hạn giữa các sự kiện, từ đó phát hiện các mẫu bất thường hoặc hành vi tiềm ẩn nguy cơ.

**Xây dựng và huấn luyện mô hình**

- Chia dữ liệu: Tách dữ liệu thành tập huấn luyện (training), tập kiểm tra (testing), và tập đánh giá (validation) theo tỷ lệ thích hợp, chẳng hạn 70%-20%-10%.
- Xây dựng mô hình LSTM:
- Thiết kế kiến trúc: Bao gồm các lớp LSTM kết nối với các lớp Dense (fully connected) để tạo đầu ra dự đoán.
- Cấu hình tham số: Đặt số lượng tế bào LSTM, kích thước batch, số epoch, và sử dụng các kỹ thuật như dropout để giảm thiểu hiện tượng overfitting.

**Huấn luyện mô hình**

- Tối ưu hóa bằng các thuật toán như Adam để tăng tốc độ hội tụ.
- Theo dõi quá trình huấn luyện qua biểu đồ hàm mất mát và độ chính xác trên tập kiểm tra.

**Đánh giá mô hình**

Mô hình được đánh giá dựa trên các chỉ số

- Precision, Recall, F1-score: Sử dụng cho các bài toán phát hiện bất thường hoặc phân loại.

**Kiểm tra khả năng phát hiện bất thường**

- Dùng dữ liệu giả lập tấn công mạng để đánh giá khả năng phát hiện các yêu cầu bất thường.
- Đo tỷ lệ phát hiện (True Positive Rate) và tỷ lệ báo sai (False Positive Rate).
- Tinh chỉnh và triển khai

**Tinh chỉnh mô hình**

- Điều chỉnh siêu tham số như LSTM, learning rate, hoặc kích thước batch để cải thiện hiệu suất.
- Áp dụng các phương pháp tăng cường dữ liệu nếu dữ liệu log chưa đủ phong phú.

**Triển khai mô hình**

- Tích hợp mô hình vào hệ thống giám sát thời gian thực.
- Kết hợp với các công cụ cảnh báo để tự động hóa việc phát hiện và phản ứng với các tấn công mạng.

Việc xây dựng mô hình LSTM không chỉ đòi hỏi kỹ thuật học máy mà còn cần hiểu biết sâu về đặc trưng dữ liệu log và yêu cầu của bài toán. Điều này đảm bảo mô hình hoạt động hiệu quả và đáp ứng các mục tiêu bảo mật hệ thống.

## 2.5. Huấn luyện mô hình

### 2.5.1. Chuẩn bị tập dữ liệu

Để huấn luyện mô hình học máy, bước chuẩn bị tập dữ liệu là vô cùng quan trọng. Tập dữ liệu phải được chuẩn bị kỹ lưỡng để đảm bảo chất lượng và độ tin cậy cao nhất.

#### *2.5.1.1. Thu thập dữ liệu*

Dữ liệu được lấy từ các tệp log chứa thông tin về các yêu cầu HTTP. Tệp data_attack.txt chứa các log có các yêu cầu tấn công trong khi data_clean.log chứa các log bình thường. Các log này được gắn nhãn tương ứng với 1 (tấn công) và 0 (bình thường).

#### *2.5.1.2. Xử lý dữ liệu*

Dữ liệu log chứa nhiều thông tin không cần thiết. Chúng ta tập trung vào các cột quan trọng như địa chỉ IP, thời gian, phương thức HTTP, và URL yêu cầu.

Sau khi tải dữ liệu từ hai file sẽ sử dụng một hàm chuyển đổi thời gian từ log thành định dạng datetime của Python, xử lý cả trường hợp có hoặc không có múi giờ.

Và ngay sau đó sẽ sử dụng một biểu thức chính quy (regex) để trích xuất thông tin của từng dòng log. Các thông tin này bao gồm địa chỉ IP, thời gian, phương thức HTTP, URL được yêu cầu, phiên bản HTTP, mã trạng thái, kích thước phản hồi, và thông tin về trình duyệt người dùng (user-agent) để có thể thuận tiện cho việc xử lý.

```
def parse_log_time(timestamp):
    try:
        # Cố gắng khớp với định dạng có múi giờ
        return datetime.strptime(timestamp, "%d/%b/%Y:%H:%M:%S %z")
    except ValueError:
        # Nếu thất bại, khớp với định dạng không có múi giờ
        return datetime.strptime(timestamp, "%d/%b/%Y:%H:%M:%S")


def load_data(file_path, label):
    columns = ['ip', 'datetime', 'method', 'url', 'http_version', 'status', 'size', 'user_agent', 'sql_injection', 'xss_attack', 'label']
    data = pd.DataFrame(columns=columns)

    with open(file_path, 'r') as file:
        lines = file.readlines()

    for line in lines:
        try:
            pattern = r'(?P<ip>\d{1,3}(?:\.\d{1,3}){3}) - - \[(?P<datetime>\d{2}/[A-Za-z]{3}/\d{4}:\d{2}:\d{2}:\d{2}(?::\d{2})?(?:\s[+-]\d{4})?)\] "(?P<method>GET|POST|PUT|DELETE|HEAD|OPTIONS) (?P<url>[^\"]+) HTTP/(?P<http_version>\
            match = re.match(pattern, line)

            if match:
                # Lấy thông tin từ nhóm đã đặt tên
                ip_address = match.group("ip")
                timestamp = match.group("datetime")
                request_method = match.group("method")
                request_url = match.group("url")
                http_version = match.group("http_version")
                status_code = int(match.group("status_code"))

                # Kiểm tra nếu kích thước là '-' thì đặt là None
                response_size = match.group("size")
                response_size = int(response_size) if response_size != '-' else None

                user_agent = match.group("user_agent")

                # Chuyển đổi datetime
                date_time = parse_log_time(timestamp)

                new_data = {
                    'ip': ip_address,
                    'datetime': date_time,
                    'method': request_method,
                    'url': request_url,
                    'http_version': http_version,
                    'status': status_code
```

*Hình 2.5: Hàm xử lý dữ liệu đầu vào*

Hàm parse_log_time dùng để chuyển đổi thời gian (timestamp) trong log từ chuỗi ký tự thành đối tượng datetime. Đầu tiên, hàm cố gắng khớp chuỗi thời gian theo định dạng có chứa múi giờ (%d/%b/%Y:%H:%M:%S %z). Nếu không thành công, nó sẽ thử chuyển đổi theo định dạng không có múi giờ (%d/%b/%Y:%H:%M:%S). Điều này giúp đảm bảo tính linh hoạt khi xử lý các log có sự khác biệt về định dạng thời gian.

Hàm load_data thực hiện nhiệm vụ chính là đọc file log, phân tích từng dòng, và lưu trữ dữ liệu đã xử lý vào một DataFrame của thư viện pandas. Hàm bắt đầu bằng cách tạo một DataFrame rỗng với các cột ip, datetime, method, url, http_version, status, size, user_agent, sql_injection, xss_attack, và label. Điều này giúp chuẩn bị cấu trúc dữ liệu để lưu trữ các thông tin cần thiết từ log.

```python
#SQL Injection
def detect_sql_injection(url):
    sql_patterns = [
        r"(\%27)|(\')|(\-\-)|(\%23)|(#)",
        r"((\%3D)|(=))[^\n]*((\%27)|(\')|(\-\-)|(\%3B)|(;))",
        r"\b(OR|AND)\b.+\b(=|LIKE|IN)\b",
        r"(UNION(\%20| )SELECT(\%20| ))",
        r"(SELECT.+FROM.+WHERE)|(INSERT INTO.+VALUES)",
        r"(EXEC(\s|\+)+(xp_cmdshell|sp_executesql))"
    ]
    for pattern in sql_patterns:
        if re.search(pattern, url, re.IGNORECASE):
            return 1
    return 0

#XSS
def detect_xss(url):
    xss_patterns = [
        r"<.*?>",
        r"(document\.cookie|document\.write|window\.location|<script>)",
        r"(<.*?javascript:.*?>)|((\%3C)|<).*?script.*?((\%3E)|>)",
        r"(onerror=|onload=|alert\()"
    ]
    for pattern in xss_patterns:
        if re.search(pattern, url, re.IGNORECASE):
            return 1
    return 0

#DDoS
def detect_ddos(df, request_threshold=100, time_window=timedelta(minutes=1)):
    df['ddos_attack'] = 0
    df = df.sort_values(by=['ip', 'datetime'])
    for ip in df['ip'].unique():
        ip_data = df[df['ip'] == ip]
        for i in range(len(ip_data) - request_threshold):
            if ip_data.iloc[i + request_threshold]['datetime'] - ip_data.iloc[i]['datetime'] <= time_window:
                df.loc[ip_data.index[i:i + request_threshold], 'ddos_attack'] = 1
                break
    return df
```

*Hình 2.6: Các hàm xử lý dữ liệu*

Các hàm trên được thiết kế để phát hiện ba loại tấn công mạng phổ biến: SQL Injection, XSS (Cross-Site Scripting), và DDoS (Distributed Denial of Service).

Hàm detect_sql_injection được sử dụng để kiểm tra xem một URL có chứa dấu hiệu của tấn công SQL Injection hay không. SQL Injection là một loại tấn công nhằm chèn các câu lệnh SQL độc hại vào các truy vấn cơ sở dữ liệu. Hàm này sử dụng một danh sách các biểu thức chính quy để xác định các mẫu dữ liệu nghi ngờ, chẳng hạn như UNION SELECT, ' OR 1=1, hoặc DROP TABLE. Nếu URL khớp với bất kỳ mẫu nào trong danh sách, hàm sẽ trả về giá trị 1 (phát hiện tấn công), ngược lại trả về 0.

Tương tự, hàm detect_xss sẽ kiểm tra xem URL có chứa các mẫu dữ liệu liên quan đến tấn công XSS hay không. XSS là tấn công tiêm mã độc JavaScript vào các trang web, thường nhắm vào việc đánh cắp thông tin hoặc làm hỏng giao diện web. Danh sách các biểu thức regex trong hàm này được sử dụng để tìm kiếm các dấu hiệu như thẻ <script>, các sự kiện onerror hoặc alert, cũng như truy cập các thuộc tính trình duyệt như document.cookie. Nếu phát hiện mẫu khớp, hàm trả về 1, biểu thị URL có khả năng bị tấn công XSS.

Cuối cùng, hàm detect_ddos phát hiện các cuộc tấn công DDoS dựa trên tần suất yêu cầu từ mỗi địa chỉ IP trong một khoảng thời gian cụ thể. Dữ liệu log được tổ chức dưới dạng một DataFrame với các cột ip và datetime. Hàm này kiểm tra xem số lượng yêu cầu từ một IP có vượt quá ngưỡng request_threshold trong khoảng thời gian time_window hay không (mặc định là 100 yêu cầu trong 1 phút). Nếu vượt ngưỡng, IP đó được đánh dấu là một nguồn tấn công DDoS với ddos_attack = 1.

### 2.5.2. Huấn luyện mô hình

Quá trình huấn luyện mô hình là một bước quan trọng trong việc xây dựng hệ thống phát hiện tấn công. Dữ liệu đầu vào cần được chuẩn bị kỹ lưỡng để đảm bảo mô hình học tập hiệu quả. Tiếp theo sẽ được xử lý mất cân bằng lớp bằng kỹ thuật oversampling, dữ liệu sẽ được chia thành các tập huấn luyện và kiểm tra. Dữ liệu bao gồm các thông tin như phương thức HTTP, URL và thông tin người dùng được kết hợp thành chuỗi văn bản bằng cách sử dụng bộ mã hóa từ vựng (Tokenizer). Để đảm bảo tính thống nhất và giảm thiểu lỗi.

```
        coefficients = np.asarray(values[1:], dtype='float32')
        embedding_index[word] = coefficients

# Create embedding matrix
embedding_matrix = np.zeros((5000, 100))
for word, i in tokenizer.word_index.items():
    if i < 5000:
        embedding_vector = embedding_index.get(word)
        if embedding_vector is not None:
            embedding_matrix[i] = embedding_vector

# Build LSTM model
model = Sequential([
    Embedding(input_dim=5000, output_dim=100, weights=[embedding_matrix], input_length=100, trainable=False),
    SpatialDropout1D(0.3),
    LSTM(100, return_sequences=True, dropout=0.3, recurrent_dropout=0.3),
    BatchNormalization(),
    LSTM(50, dropout=0.3, recurrent_dropout=0.3),
    Dense(32, activation='relu'),
    Dropout(0.3),
    Dense(1, activation='sigmoid')
])

# Compile model
model.compile(
    optimizer=tf.keras.optimizers.Adam(learning_rate=0.001),
    loss='binary_crossentropy',
    metrics=['accuracy', tf.keras.metrics.Precision(), tf.keras.metrics.Recall()]
)

# Add callbacks
early_stopping = EarlyStopping(monitor='val_loss', patience=10, restore_best_weights=True)
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=3, min_lr=1e-6)

# Train model
history = model.fit(
    X_train_padded, y_train,
    epochs=20,
    validation_data=(X_test_padded, y_test),
    batch_size=32,
    callbacks=[early_stopping, reduce_lr]
)
```

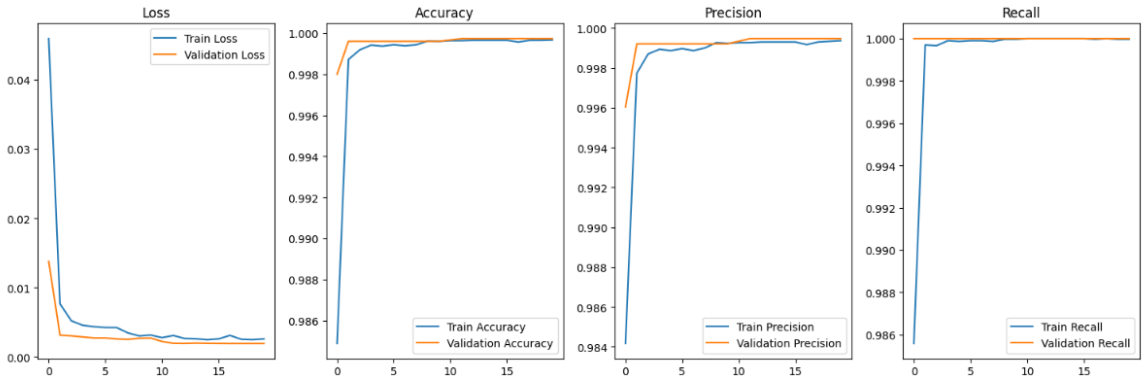
*Hình 2.7: Các hàm huấn luyện mô hình*

Mô hình sử dụng các lớp Dropout và BatchNormalization, để tối ưu hóa và giảm nguy cơ overfitting. Quá trình huấn luyện được giám sát bởi các hàm callback như EarlyStopping và ReduceLROnPlateau nhằm cải thiện hiệu suất và tránh việc mô hình huấn luyện quá lâu mà không tăng độ chính xác.

## 2.6. Đánh giá mô hình

| epoch | accuracy | loss | precision | recall | val_accuracy | val_loss | val_precision | val_recall | learning_rate |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.9849 | 0.0459 | 0.9842 | 0.9856 | 0.9980 | 0.0138 | 0.9960 | 1.0 | 0.001 |
| 2 | 0.9987 | 0.0077 | 0.9977 | 0.9997 | 0.9996 | 0.0031 | 0.9992 | 1.0 | 0.001 |
| 3 | 0.9992 | 0.0052 | 0.9987 | 0.9997 | 0.9996 | 0.0030 | 0.9992 | 1.0 | 0.001 |
| 4 | 0.9994 | 0.0046 | 0.9989 | 0.9999 | 0.9996 | 0.0029 | 0.9992 | 1.0 | 0.001 |
| 5 | 0.9994 | 0.0044 | 0.9989 | 0.9999 | 0.9996 | 0.0027 | 0.9992 | 1.0 | 0.001 |
| 6 | 0.9994 | 0.0042 | 0.9990 | 0.9999 | 0.9996 | 0.0027 | 0.9992 | 1.0 | 0.001 |
| 7 | 0.9994 | 0.0042 | 0.9989 | 0.9999 | 0.9996 | 0.0026 | 0.9992 | 1.0 | 0.001 |
| 8 | 0.9994 | 0.0035 | 0.9990 | 0.9999 | 0.9996 | 0.0025 | 0.9992 | 1.0 | 0.001 |
| 9 | 0.9996 | 0.0030 | 0.9993 | 1.0 | 0.9996 | 0.0027 | 0.9992 | 1.0 | 0.001 |
| 10 | 0.9996 | 0.0032 | 0.9992 | 1.0 | 0.9996 | 0.0027 | 0.9992 | 1.0 | 0.001 |
| 11 | 0.9996 | 0.0028 | 0.9993 | 1.0 | 0.9997 | 0.0022 | 0.9993 | 1.0 | 0.0002 |
| 12 | 0.9996 | 0.0031 | 0.9993 | 1.0 | 0.9997 | 0.0020 | 0.9995 | 1.0 | 0.0002 |
| 13 | 0.9996 | 0.0027 | 0.9993 | 1.0 | 0.9997 | 0.0020 | 0.9995 | 1.0 | 0.0002 |
| 14 | 0.9996 | 0.0026 | 0.9993 | 1.0 | 0.9997 | 0.0020 | 0.9995 | 1.0 | 0.0002 |
| 15 | 0.9996 | 0.0025 | 0.9993 | 1.0 | 0.9997 | 0.0020 | 0.9995 | 1.0 | 0.0002 |
| 16 | 0.9996 | 0.0026 | 0.9993 | 1.0 | 0.9997 | 0.0020 | 0.9995 | 1.0 | 4.00E-05 |
| 17 | 0.9996 | 0.0031 | 0.9992 | 1.0 | 0.9997 | 0.0019 | 0.9995 | 1.0 | 4.00E-05 |
| 18 | 0.9996 | 0.0025 | 0.9993 | 1.0 | 0.9997 | 0.0019 | 0.9995 | 1.0 | 4.00E-05 |
| 19 | 0.9996 | 0.0025 | 0.9993 | 1.0 | 0.9997 | 0.0019 | 0.9995 | 1.0 | 8.00E-06 |
| 20 | 0.9997 | 0.0026 | 0.9994 | 1.0 | 0.9997 | 0.0019 | 0.9995 | 1.0 | 8.00E-06 |

*Hình 2.8: Chi tiết kết quả quá trình huấn luyện*




*Hình 2.9: Biểu đồ kết quả quá trình huấn luyện*

Trong quá trình huấn luyện, accuracy trên tập huấn luyện tăng đều qua các epoch, đạt mức 0.9997 ở epoch cuối cùng. Điều này cho thấy mô hình đã học

tốt các mẫu trong tập huấn luyện. Đồng thời, validation accuracy cũng tăng dần và đạt 0.9997 ở epoch cuối, rất gần với accuracy của tập huấn luyện. Đây là dấu hiệu cho thấy mô hình tổng quát hóa tốt, không bị overfitting.

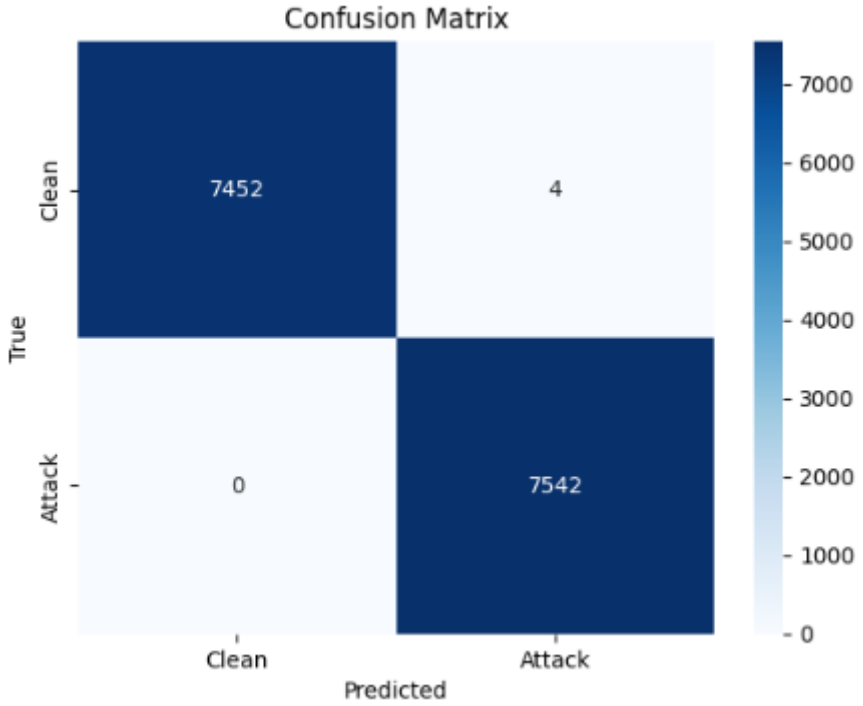
Đối với loss, giá trị này giảm nhanh chóng qua các epoch đầu tiên và ổn định ở mức rất thấp từ epoch 12 trở đi. Training loss giảm từ 0.0459 ở epoch 1 xuống còn 0.0026 ở epoch 20, trong khi validation loss giảm từ 0.0138 xuống 0.0019. Điều này chứng tỏ mô hình dần tiếp cận trạng thái hội tụ, và việc giảm loss ở tập kiểm tra là một tín hiệu tích cực về khả năng học sâu.

Hai chỉ số quan trọng để đánh giá độ chính xác của mô hình là precision và recall đều đạt giá trị rất cao gần bằng 1.0 trên cả tập huấn luyện và tập kiểm tra. Precision cao trong khoảng 0.9994-0.9999 cho thấy mô hình dự đoán rất chính xác các mẫu dương tính, với rất ít trường hợp bị dương tính giả. Trong khi đó, recall đạt giá trị tuyệt đối 1.0 từ epoch 10 trở đi trên tập kiểm tra, chứng minh mô hình không bỏ sót bất kỳ mẫu dương tính nào.

Learning rate ban đầu được đặt ở mức 0.001 và giảm dần theo từng giai đoạn của quá trình huấn luyện, với giá trị nhỏ nhất là 8e-06. Kết hợp cơ chế giảm tốc độ học đã giúp mô hình hội tụ một cách hiệu quả, đặc biệt trong các epoch cuối, khi mô hình tiến gần đến trạng thái tối ưu. Nhờ vậy, mô hình có thể tránh được hiện tượng dao động hoặc không hội tụ ở các điểm gần cực trị.

#### *2.6.1.1.Ma trận nhầm lẫn*

Dựa vào ma trận nhầm lẫn, có thể thấy rằng mô hình hoạt động tốt với tỷ lệ dự đoán đúng cao. Số lượng các trường hợp True Positive và True Negative rất lớn, trong khi số lượng False Positive và False Negative nhỏ, cho thấy mô hình có độ chính xác cao và ít gặp sai sót trong phân loại.



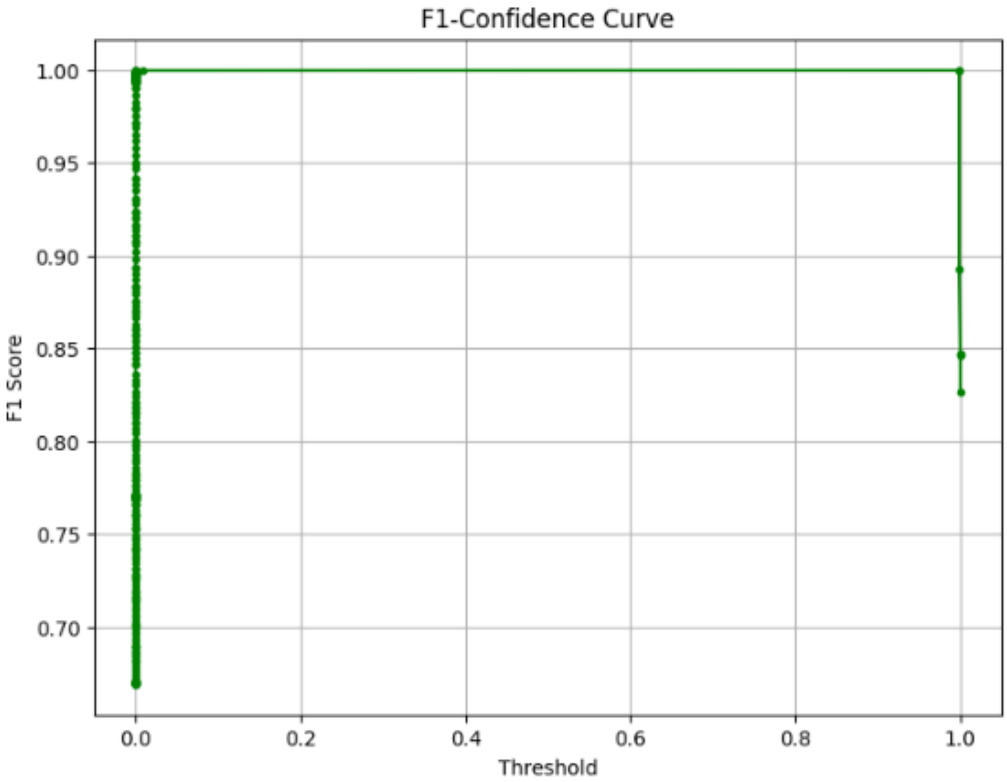

*Hình 2.10: Ma trận nhầm lẫn*

- True Positive (TP): Có 7,542 trường hợp "Attack" được mô hình dự đoán chính xác là "Attack". Điều này khẳng định rằng mô hình có khả năng nhận diện rất tốt các trường hợp tấn công.
- True Negative (TN): Có 7,452 trường hợp "Clean" được mô hình dự đoán chính xác là "Clean". Điều này chứng tỏ mô hình cũng có độ chính xác cao trong việc phân loại các truy vấn bình thường, đảm bảo rằng phần lớn các hoạt động hợp lệ không bị nhận diện sai.
- False Positive (FP): Có 4 trường hợp "Clean" bị nhầm lẫn là "Attack". Đây là số lượng báo động giả rất thấp, cho thấy mô hình đã hạn chế hiệu quả hiện tượng nhận diện nhầm các truy vấn sạch thành tấn công.
- False Negative (FN): Có 0 trường hợp "Attack" bị nhầm lẫn là "Clean". Kết quả này đã cho thấy không có trường hợp tấn công nào bị bỏ sót, đảm bảo rằng hệ thống có thể phản ứng nhanh chóng và hiệu quả với mọi mối đe dọa.

### *2.6.1.2. Đường cong điểm F1*

Đường cong F1-Confidence mô tả mối quan hệ giữa điểm số F1 và ngưỡng dự đoán.

- Trục ngang (Threshold): Biểu thị ngưỡng dự đoán, với các giá trị từ 0 đến 1. Ngưỡng dự đoán xác định mức độ tự tin của mô hình khi phân loại một trường hợp là "Attack" hay "Clean". Ví dụ, nếu ngưỡng là 0.5, các trường hợp có xác suất lớn hơn 0.5 sẽ được phân loại là "Attack", ngược lại sẽ được phân loại là "Clean".
- Trục dọc (F1 Score): Biểu thị điểm số F1, là một chỉ số đánh giá hiệu suất của mô hình, kết hợp giữa độ chính xác dự đoán và độ nhạy. Điểm số F1 càng cao thì mô hình càng có hiệu quả tốt trong việc cân bằng giữa việc phát hiện đúng các trường hợp và hạn chế báo động giả.



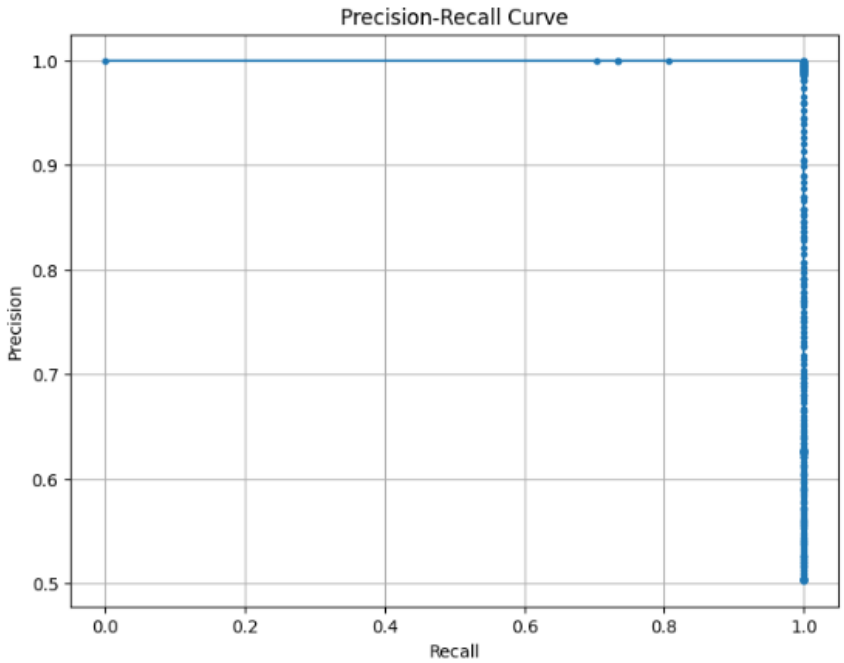

*Hình 2.11: Đường cong điểm F1*

Từ đồ thị có thể thấy mô hình duy trì F1 score gần như tối đa (xấp xỉ 1.0) trên phần lớn các ngưỡng dự đoán, đặc biệt ở khoảng giữa biểu đồ. Biểu đồ phản ánh rằng mô hình có hiệu suất cao và ổn định ở một khoảng rộng các ngưỡng dự đoán. Sự ổn định này cho thấy mô hình có khả năng cân bằng tốt giữa Precision và Recall, giúp đảm bảo tính chính xác và bao phủ trong dự đoán. F1 score đạt giá trị tối đa ở vùng trung tâm của biểu đồ trong khoảng ngưỡng từ 0.4 đến 0.6.

#### *2.6.1.3. Đường cong thu hồi chính xác*

Đường cong Precision-Recall (Precision-Recall Curve), dùng để đánh giá hiệu suất của mô hình phân loại.

- Precision (Độ chính xác dự đoán): Tỷ lệ các dự đoán "Attack" đúng trên tổng số các dự đoán là "Attack".
- Recall (Độ nhạy): Tỷ lệ các trường hợp "Attack" thực sự được phát hiện (dự đoán đúng) so với tổng số trường hợp "Attack".




*Hình 2.12: Đường cong thu hồi chính xác*

Đặc điểm của đường cong:

- Trục hoành (Recall): đại diện cho tỷ lệ các trường hợp dương tính thực sự được mô hình phát hiện. Recall cao thể hiện khả năng mô hình phát hiện hầu hết các trường hợp đúng.
- Trục tung (Precision): phản ánh tỷ lệ các dự đoán dương tính đúng so với tổng số dự đoán dương tính. Precision cao nghĩa là số lượng dự đoán nhầm giảm và mô hình tập trung vào các dự đoán chính xác.

Trong đồ thị, có thể thấy rằng ở phần lớn giá trị của Recall, Precision duy trì ở mức rất cao (gần bằng 1). Điều này cho thấy mô hình có khả năng hoạt động hiệu quả và duy trì độ chính xác cao trong việc dự đoán các trường hợp dương tính ngay cả khi tăng khả năng bao phủ .Tuy nhiên, khi Recall gần đạt 1, có một sự sụt giảm nhanh chóng về Precision. Điều này xảy ra khi mô hình cố gắng bao phủ tất cả các trường hợp dương tính, dẫn đến một số lượng nhỏ các dự đoán sai, làm giảm độ chính xác.

## 2.7. Kết luận chương 2

Chương 2 đã tập trung nghiên cứu và triển khai các phương pháp phát hiện tấn công web dựa trên công nghệ học máy. Đầu tiên, phương pháp tiếp cận được xác định nhằm đảm bảo quá trình phát hiện tấn công diễn ra một cách hiệu quả và toàn diện. Tiếp theo, việc thu thập và phân loại dữ liệu về các trang web bị tấn công và không bị tấn công được tiến hành một cách kỹ lưỡng, tạo nền tảng dữ liệu phong phú và đáng tin cậy. Việc thực hiện lựa chọn thuật toán học máy phù hợp, xây dựng và huấn luyện mô hình dựa trên tập dữ liệu được chuẩn bị một cách cẩn thận. Các giai đoạn huấn luyện và đánh giá được thực hiện nghiêm ngặt để đảm bảo mô hình đạt được độ chính xác cao trong việc phát hiện các loại tấn công web như SQLi, XSS, và DDoS.

# CHƯƠNG 3 - XÂY DỰNG HỆ THỐNG PHÁT HIỆN VÀ CẢNH BÁO TẤN CÔNG WEB

## 3.1. Đặt vấn đề

Trong thời đại ngày nay vấn đề an toàn thông tin trở nên vô cùng quan trọng. Mặc dù nhiều công nghệ an toàn thông tin hiện đại đã được phát triển với độ chính xác ngày càng cao, cuộc chiến giữa các biện pháp bảo mật và các kỹ thuật tấn công của tin tặc vẫn không ngừng diễn ra. Các công nghệ hiện tại vẫn chưa thể ngăn chặn hoàn toàn mọi kiểu tấn công tinh vi. Do đó, trong những năm gần đây, các nhà nghiên cứu đã chuyển hướng tập trung vào việc phát triển các công cụ phát hiện tấn công website để có thể phát hiện và ngăn chặn các cuộc tấn công một cách sớm nhất tránh gây ảnh hưởng tới hệ thống tổ chức.

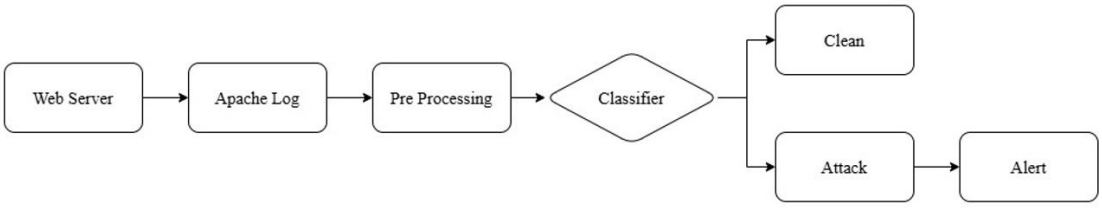
Các mối nguy hại có thể đến từ cả bên trong và bên ngoài tổ chức, và việc hệ thống bị xâm nhập là điều không thể tránh khỏi. Thách thức đặt ra là làm thế nào để phát hiện và ứng phó với các mối nguy hại này một cách nhanh chóng và chính xác. Mặc dù các máy chủ web ngày nay được bảo vệ bởi các lớp an ninh chắc chắn, không phải tổ chức nào cũng đủ kinh phí để triển khai đầy đủ các giải pháp bảo mật. Các giải pháp hiện tại cũng chưa hoàn toàn toàn diện, vẫn có những kỹ thuật tinh vi có thể vượt qua. Việc các website của các tổ chức bị tấn công đã cho thấy sự thiếu sót trong các lớp bảo mật. Quản trị viên cần nắm bắt thông tin nhanh chóng để đưa ra các biện pháp điều tra và ứng phó, giảm thiểu thiệt hại cho tổ chức.

Hệ thống cảnh báo phát hiện tấn công website sẽ hỗ trợ quản trị viên trong tình huống này. Bằng cách theo dõi trang web từ góc nhìn của người dùng thông thường, hệ thống có thể phát hiện nhanh chóng các thay đổi ngay trong log của hệ thống do kẻ tấn công gây ra và lập tức gửi cảnh báo cho quản trị viên. Đặc biệt, hệ thống có thể tích hợp với các hoạt động của hệ thống SOC để nâng cao hiệu quả phản ứng và bảo vệ hệ thống.

## 3.2. Xây dựng hệ thống cảnh báo

### 3.2.1. Các thành phần của hệ thống cảnh báo

Hệ thống cảnh báo của nghiên cứu có những thành phần như sau:




*Hình 3.1: Các thành phần của hệ thống*

Đầu tiên, máy chủ web tiếp nhận các yêu cầu HTTP từ người dùng hoặc các ứng dụng khác và ghi lại chi tiết của mỗi yêu cầu vào tệp Apache log, bao gồm thông tin như địa chỉ IP, URL truy cập, và phương thức HTTP. Tiếp theo, dữ liệu nhật ký được đưa vào giai đoạn tiền xử lý, nơi nó được làm sạch, chuẩn hóa, và chuyển đổi sang định dạng phù hợp để phân tích. Sau khi tiền xử lý, dữ liệu được chuyển đến bộ phân loại, nơi các yêu cầu được phân thành hai loại: "Clean" (hợp lệ) hoặc "Attack" (có dấu hiệu tấn công). Nếu yêu cầu là hợp lệ, nó sẽ được xử lý tiếp hoặc bỏ qua. Ngược lại, nếu phát hiện dấu hiệu tấn công, hệ thống sẽ kích hoạt cơ chế cảnh báo, gửi thông báo đến đội ngũ bảo mật để có biện pháp ứng phó kịp thời. Quy trình này giúp tự động hóa việc phát hiện các cuộc tấn công để đảm bảo an toàn cho hệ thống.

### 3.2.2. Triển khai hệ thống

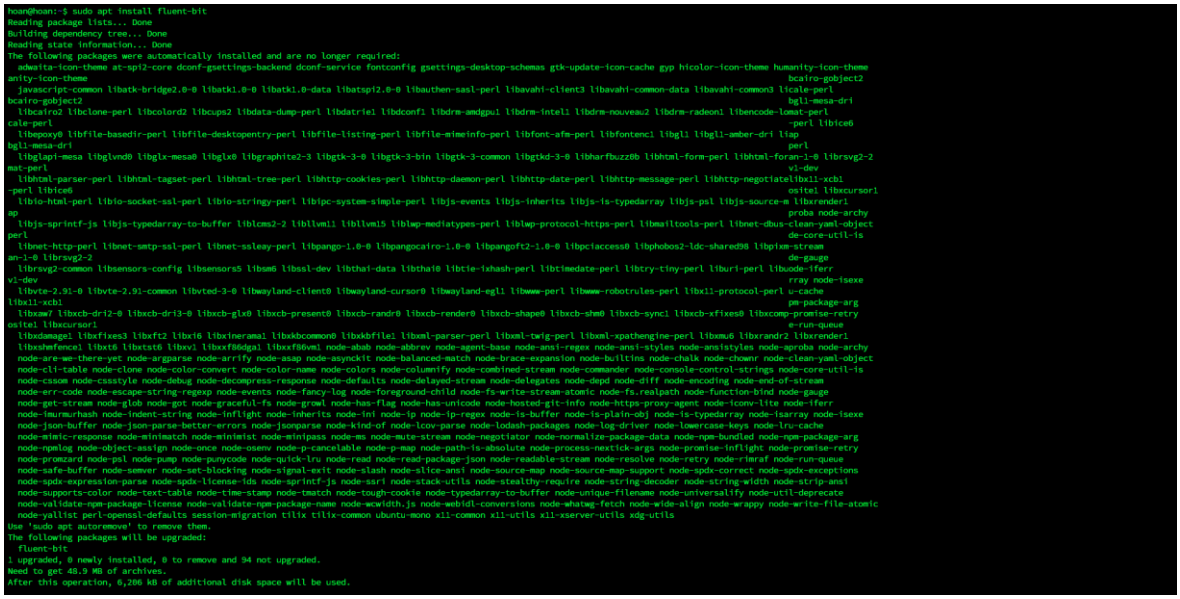
Sử dụng fluent-bit để gửi log apache từ server web sang server xử lý.

Fluent-bit là một phần mềm mã nguồn mở, nhẹ và có khả năng mở rộng cao, được thiết kế để thu thập và xử lý dữ liệu telemetry từ nhiều nguồn khác nhau sau đó phân tích, lọc và phân loại log trước khi gửi đến các đích đến đã

xác định như cơ sở dữ liệu, hệ thống lưu trữ đám mây hoặc các nền tảng phân tích dữ liệu.

Fluent-bit nổi bật nhờ khả năng hỗ trợ mạnh mẽ cho các môi trường đám mây và container, cho phép tích hợp dễ dàng với các công cụ như Kubernetes và Docker. Ngoài ra, nó còn cung cấp nhiều plugin cho các input, filter và output, giúp người dùng dễ dàng mở rộng và tùy chỉnh chức năng theo nhu cầu cụ thể. Một trong những ưu điểm lớn của Fluent Bit là khả năng hoạt động hiệu quả ngay cả trong các môi trường hạn chế tài nguyên, với hiệu suất cao và mức tiêu thụ tài nguyên thấp, giúp đảm bảo hoạt động liên tục và đáng tin cậy của hệ thống.

### *3.2.2.1. Thu thập log từ hệ thống*




*Hình 3.2: Cài đặt fluent-bit*

Sau khi đã cài đặt thành công fluent-bit trên hệ thống web server cần cấu hình fluent-bit để gửi log sang server xử lý và gửi lên giao diện của hệ thống ElasticSearch

```
# Đọc log từ /var/log/apache2/access.log
[INPUT]
    Name             tail
    Path             /var/log/apache2/access.log*
    Tag              access
    Refresh_Interval 5
    Rotate_Wait      5
    Mem_Buf_Limit    5MB
    Skip_Long_Lines  On
    Parser           apache2

[INPUT]
    Name             tail
    Path             /var/log/apache2/access.log
    Tag              access.log
    Refresh_Interval 5
    Rotate_Wait      5
    Mem_Buf_Limit    5MB
    Skip_Long_Lines  On

# Output 1: Gửi log đến Server B
[OUTPUT]
    Name        http
    Match       access.log
    Host        192.168.135.144
    Port        5000
    URI         /webhook
    tls         Off
    Format      json
    Json_date_key    timestamp
    Json_date_format iso8601

# Output 2: Gửi log đến ElasticSearch
[OUTPUT]
    Name             es
    Match            access
    Host             192.168.135.144
    Port             9200
    HTTP_User        elastic
    HTTP_Passwd      changeme
    Suppress_Type_Name On
    Logstash_Format    On
    Retry_Limit        False
    Replace_Dots       On
```

*Hình 3.3: Cấu hình trong fluent-bit*

- INPUT

Name: tail

- Đây là tên của plugin đầu vào (tail), nghĩa là hệ thống sẽ theo dõi các dòng mới được thêm vào file log.

Path: /var/log/apache2/access.log

- Đường dẫn đến file log của Apache mà hệ thống sẽ theo dõi.

Tag: access.log

- Nhãn để phân biệt log này với các log khác.

Refresh_Interval: 5

- Thời gian (giây) giữa các lần kiểm tra file log để xem có dòng mới hay không.

Rotate_Wait: 5

- Thời gian chờ (giây) để xử lý các log sau khi file log được xoay vòng (log rotation).

Mem_Buf_Limit: 5MB

- Giới hạn bộ nhớ sử dụng để lưu trữ các log tạm thời trong bộ nhớ trước khi gửi đi.

Skip_Long_Lines: On

- Nếu các dòng log quá dài, hệ thống sẽ bỏ qua chúng thay vì cố gắng xử lý.

- OUTPUT

Name: http

- Tên của plugin đầu ra (http), nghĩa là hệ thống sẽ gửi log qua giao thức HTTP.

Match: access.log

- Nhãn của các log mà plugin đầu ra này sẽ xử lý (phải khớp với nhãn trong phần INPUT).

Host: 192.168.135.143

- Địa chỉ IP của server B nơi các log sẽ được gửi đến.

Port: 8080

- Cổng của server xử lý để nhận các log.

URI: /api/logs

- Đường dẫn API của server B nơi các log sẽ được gửi đến.

tls: Off

- Kết nối không sử dụng TLS (kết nối không được mã hóa).

Format: json

- Định dạng của các log được gửi đi (ở đây là định dạng JSON).

Json_date_key: timestamp

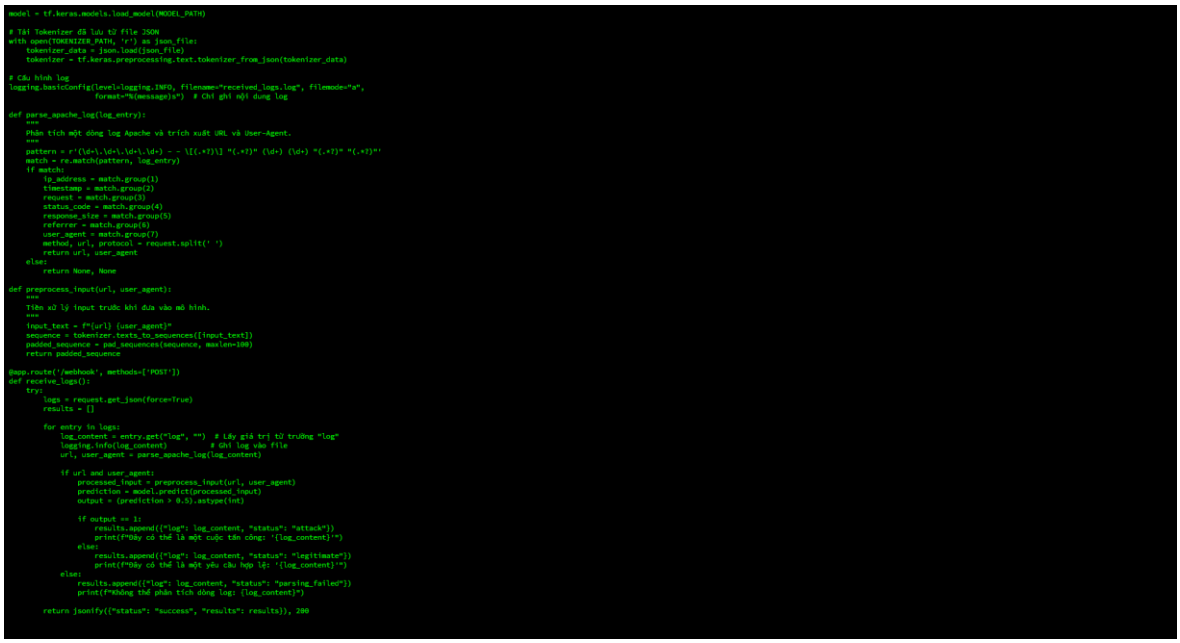
- Khóa JSON được sử dụng để lưu trữ thời gian của log.

Json_date_format: iso8601

- Định dạng của thời gian (ISO 8601).

### *3.2.2.2. Phân tích và xử lý log*

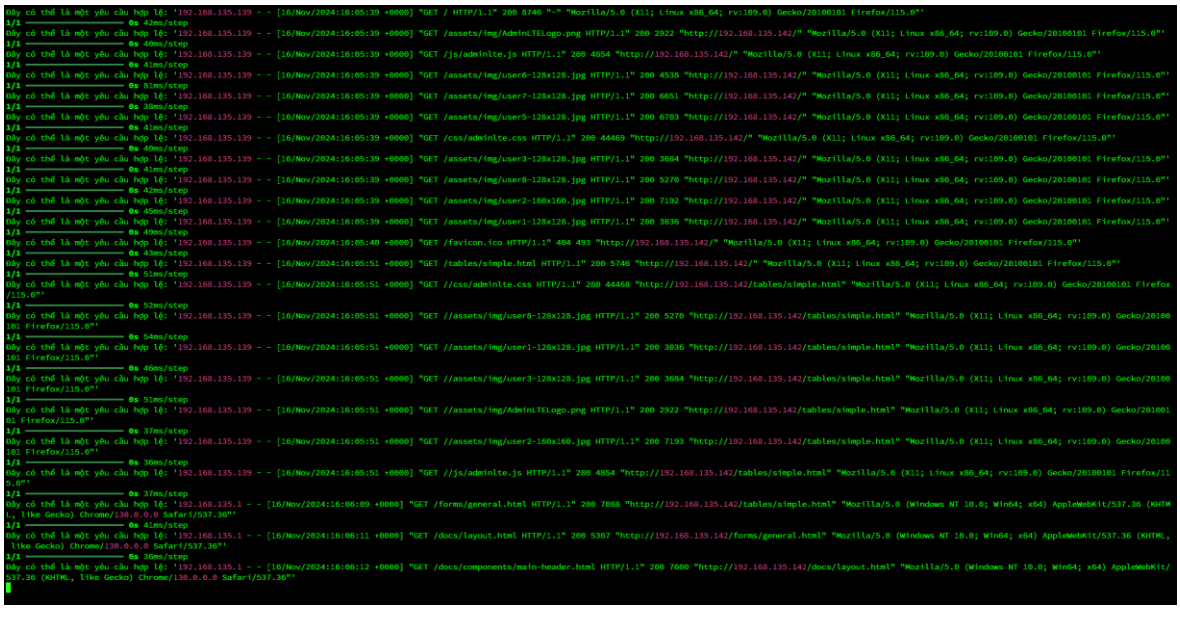
Bên server xử lý cần tạo một đoạn code python để có thể nhận được dữ liệu từ bên web server gửi sang



```
model = tf.keras.models.load_model(MODEL_PATH)

# Tải Tokenizer đã lưu từ file JSON
with open(TOKENIZER_PATH, 'r') as json_file:
    tokenizer_data = json.load(json_file)
    tokenizer = tf.keras.preprocessing.text.tokenizer_from_json(tokenizer_data)

# Cấu hình log
logging.basicConfig(level=logging.INFO, filename="received_logs.log", filemode="a",
                    format="%(message)s")  # Chỉ ghi nội dung log

def parse_apache_log(log_entry):
    """
    Phân tích một dòng log Apache và trích xuất URL và User-Agent.
    """
    pattern = r'(\d+\.\d+\.\d+\.\d+) - - \[(.*?)\] "(.*?)" (\d+) (\d+) "(.*?)" "(.*?)"'
    match = re.match(pattern, log_entry)
    if match:
        ip_address = match.group(1)
        timestamp = match.group(2)
        request = match.group(3)
        status_code = match.group(4)
        response_size = match.group(5)
        referrer = match.group(6)
        user_agent = match.group(7)
        method, url, protocol = request.split(' ')
        return url, user_agent
    else:
        return None, None

def preprocess_input(url, user_agent):
    """
    Tiền xử lý input trước khi đưa vào mô hình.
    """
    input_text = f"{url} {user_agent}"
    sequence = tokenizer.texts_to_sequences([input_text])
    padded_sequence = pad_sequences(sequence, maxlen=100)
    return padded_sequence

@app.route('/webhook', methods=['POST'])
def receive_logs():
    try:
        logs = request.get_json(force=True)
        results = []

        for entry in logs:
            log_content = entry.get("log", "")  # Lấy giá trị từ trường "log"
            logging.info(log_content)           # Ghi log vào file
            url, user_agent = parse_apache_log(log_content)

            if url and user_agent:
                processed_input = preprocess_input(url, user_agent)
                prediction = model.predict(processed_input)
                output = (prediction > 0.5).astype(int)

                if output == 1:
                    results.append({"log": log_content, "status": "attack"})
                    print(f"Đây có thể là một cuộc tấn công: {log_content}")
                else:
                    results.append({"log": log_content, "status": "legitimate"})
                    print(f"Đây có thể là một yêu cầu hợp lệ: {log_content}")
            else:
                results.append({"log": log_content, "status": "parsing_failed"})
                print(f"Không thể phân tích dòng log: {log_content}")

        return jsonify({"status": "success", "results": results}), 200
```

*Hình 3.4: Code Python nhận và xử lý dữ liệu từ Webserver*

Dữ liệu sau khi nhận được thì sẽ được tự động phân tích xem có phải một trường hợp tấn công hay không

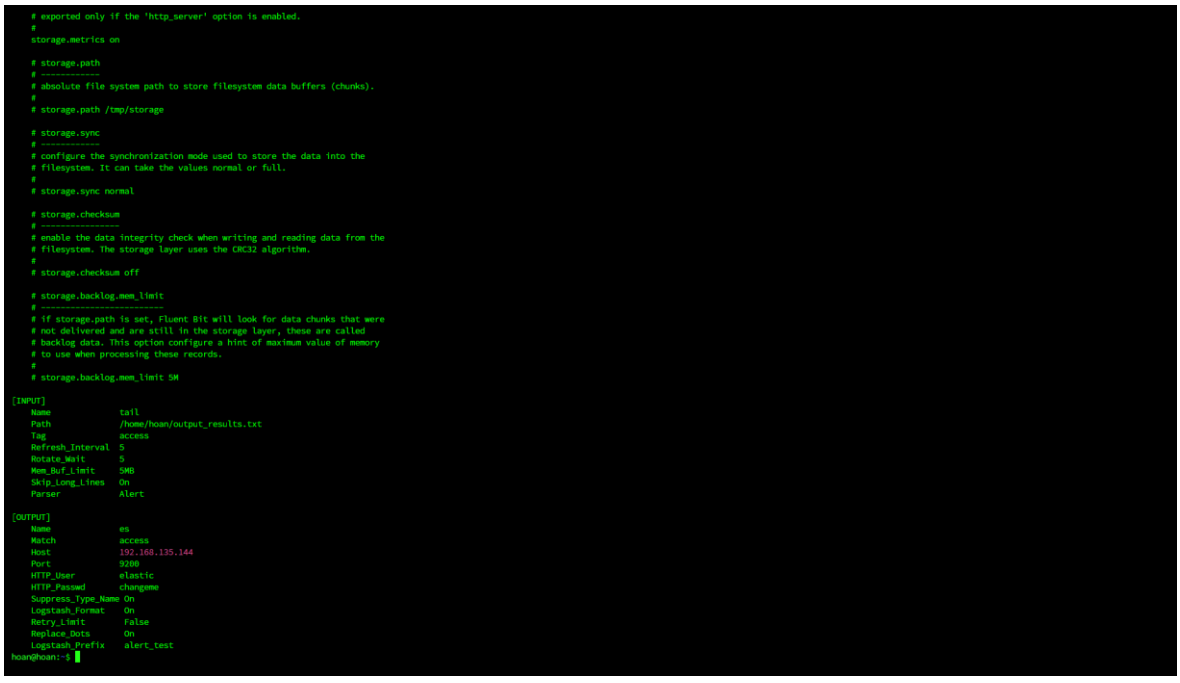


```
Đây có thể là một yêu cầu hợp lệ: '192.168.135.139 - - [16/Nov/2024:16:05:39 +0000] "GET / HTTP/1.1" 200 8746 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"'
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
Đây có thể là một yêu cầu hợp lệ: '192.168.135.139 - - [16/Nov/2024:16:05:39 +0000] "GET /assets/img/AdminLTELogo.png HTTP/1.1" 200 2922 "http://192.168.135.142/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"'
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
Đây có thể là một yêu cầu hợp lệ: '192.168.135.139 - - [16/Nov/2024:16:05:39 +0000] "GET /js/adminlte.js HTTP/1.1" 200 4854 "http://192.168.135.142/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"'
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step
Đây có thể là một yêu cầu hợp lệ: '192.168.135.139 - - [16/Nov/2024:16:05:39 +0000] "GET /assets/img/user6-128x128.jpg HTTP/1.1" 200 4538 "http://192.168.135.142/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"'
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 81ms/step
Đây có thể là một yêu cầu hợp lệ: '192.168.135.139 - - [16/Nov/2024:16:05:39 +0000] "GET /assets/img/user7-128x128.jpg HTTP/1.1" 200 6051 "http://192.168.135.142/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"'
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 38ms/step
Đây có thể là một yêu cầu hợp lệ: '192.168.135.139 - - [16/Nov/2024:16:05:39 +0000] "GET /assets/img/user5-128x128.jpg HTTP/1.1" 200 6703 "http://192.168.135.142/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"'
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step
Đây có thể là một yêu cầu hợp lệ: '192.168.135.139 - - [16/Nov/2024:16:05:39 +0000] "GET /css/adminlte.css HTTP/1.1" 200 44469 "http://192.168.135.142/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"'
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step
Đây có thể là một yêu cầu hợp lệ: '192.168.135.139 - - [16/Nov/2024:16:05:39 +0000] "GET /assets/img/user3-128x128.jpg HTTP/1.1" 200 3684 "http://192.168.135.142/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"'
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step
Đây có thể là một yêu cầu hợp lệ: '192.168.135.139 - - [16/Nov/2024:16:05:39 +0000] "GET /assets/img/user8-128x128.jpg HTTP/1.1" 200 5270 "http://192.168.135.142/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"'
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
Đây có thể là một yêu cầu hợp lệ: '192.168.135.139 - - [16/Nov/2024:16:05:39 +0000] "GET /assets/img/user2-160x160.jpg HTTP/1.1" 200 7192 "http://192.168.135.142/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"'
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
Đây có thể là một yêu cầu hợp lệ: '192.168.135.139 - - [16/Nov/2024:16:05:39 +0000] "GET /assets/img/user1-128x128.jpg HTTP/1.1" 200 3036 "http://192.168.135.142/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"'
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
Đây có thể là một yêu cầu hợp lệ: '192.168.135.139 - - [16/Nov/2024:16:05:40 +0000] "GET /favicon.ico HTTP/1.1" 404 493 "http://192.168.135.142/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"'
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
Đây có thể là một yêu cầu hợp lệ: '192.168.135.139 - - [16/Nov/2024:16:05:51 +0000] "GET /tables/simple.html HTTP/1.1" 200 5746 "http://192.168.135.142/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"'
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step
Đây có thể là một yêu cầu hợp lệ: '192.168.135.139 - - [16/Nov/2024:16:05:51 +0000] "GET //css/adminlte.css HTTP/1.1" 200 44468 "http://192.168.135.142/tables/simple.html" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox
/115.0"'
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step
Đây có thể là một yêu cầu hợp lệ: '192.168.135.139 - - [16/Nov/2024:16:05:51 +0000] "GET //assets/img/user8-128x128.jpg HTTP/1.1" 200 5270 "http://192.168.135.142/tables/simple.html" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100
101 Firefox/115.0"'
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 54ms/step
Đây có thể là một yêu cầu hợp lệ: '192.168.135.139 - - [16/Nov/2024:16:05:51 +0000] "GET //assets/img/user1-128x128.jpg HTTP/1.1" 200 3036 "http://192.168.135.142/tables/simple.html" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100
101 Firefox/115.0"'
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
Đây có thể là một yêu cầu hợp lệ: '192.168.135.139 - - [16/Nov/2024:16:05:51 +0000] "GET //assets/img/user3-128x128.jpg HTTP/1.1" 200 3684 "http://192.168.135.142/tables/simple.html" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100
101 Firefox/115.0"'
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step
Đây có thể là một yêu cầu hợp lệ: '192.168.135.139 - - [16/Nov/2024:16:05:51 +0000] "GET //assets/img/AdminLTELogo.png HTTP/1.1" 200 2922 "http://192.168.135.142/tables/simple.html" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100
101 Firefox/115.0"'
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 37ms/step
Đây có thể là một yêu cầu hợp lệ: '192.168.135.139 - - [16/Nov/2024:16:05:51 +0000] "GET //assets/img/user2-160x160.jpg HTTP/1.1" 200 7193 "http://192.168.135.142/tables/simple.html" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100
101 Firefox/115.0"'
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 36ms/step
Đây có thể là một yêu cầu hợp lệ: '192.168.135.139 - - [16/Nov/2024:16:05:51 +0000] "GET //js/adminlte.js HTTP/1.1" 200 4854 "http://192.168.135.142/tables/simple.html" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/11
5.0"'
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 37ms/step
Đây có thể là một yêu cầu hợp lệ: '192.168.135.1 - - [16/Nov/2024:16:06:09 +0000] "GET /forms/general.html HTTP/1.1" 200 7088 "http://192.168.135.142/tables/simple.html" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTM
L, like Gecko) Chrome/130.0.0.0 Safari/537.36"'
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step
Đây có thể là một yêu cầu hợp lệ: '192.168.135.1 - - [16/Nov/2024:16:06:11 +0000] "GET /docs/layout.html HTTP/1.1" 200 5387 "http://192.168.135.142/forms/general.html" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
 like Gecko) Chrome/130.0.0.0 Safari/537.36"'
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 36ms/step
Đây có thể là một yêu cầu hợp lệ: '192.168.135.1 - - [16/Nov/2024:16:06:12 +0000] "GET /docs/components/main-header.html HTTP/1.1" 200 7608 "http://192.168.135.142/docs/layout.html" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/
537.36 (KHTML, like Gecko) Chrome/130.0.0.0 Safari/537.36"'
```

*Hình 3.5: Các trường hợp được phân tích là hợp lệ*

*Hình 3.6: Các trường hợp được phân tích là tấn công*

Sau khi phân tích được hoàn tất hệ thống sẽ sử dụng fluent-bit để đẩy log lên hệ thống Elasticsearch để trực quan hóa dữ liệu

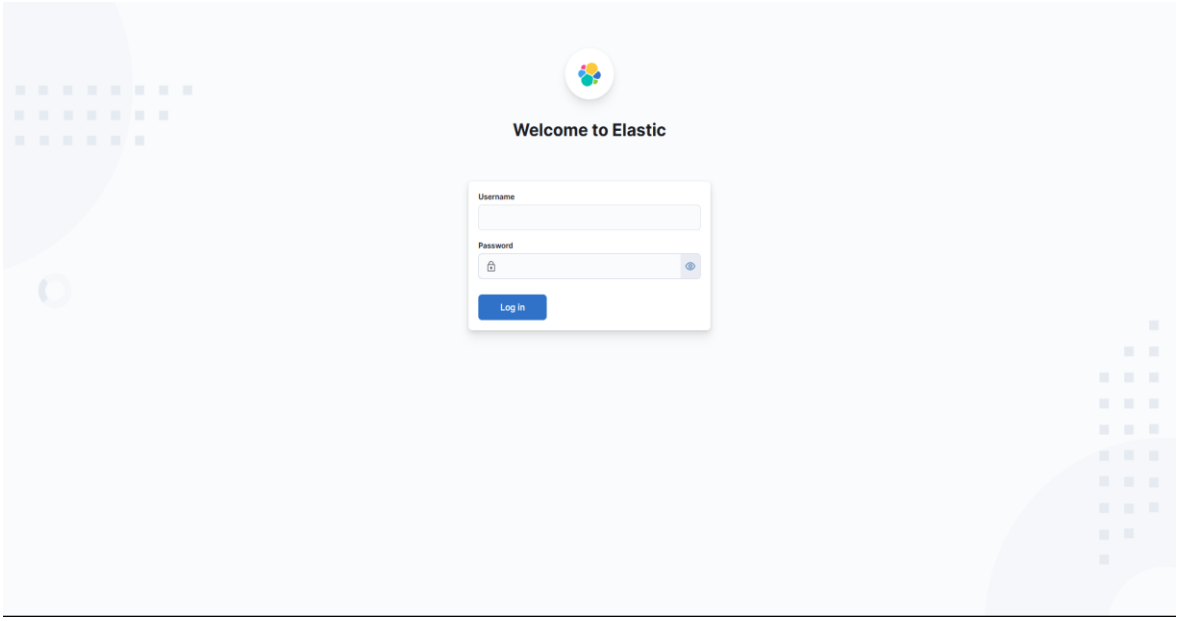


```
    # exported only if the 'http_server' option is enabled.
    #
    storage.metrics on

    # storage.path
    # ------------
    # absolute file system path to store filesystem data buffers (chunks).
    #
    # storage.path /tmp/storage

    # storage.sync
    # ------------
    # configure the synchronization mode used to store the data into the
    # filesystem. It can take the values normal or full.
    #
    # storage.sync normal

    # storage.checksum
    # ----------------
    # enable the data integrity check when writing and reading data from the
    # filesystem. The storage layer uses the CRC32 algorithm.
    #
    # storage.checksum off

    # storage.backlog.mem_limit
    # -------------------------
    # if storage.path is set, Fluent Bit will look for data chunks that were
    # not delivered and are still in the storage layer, these are called
    # backlog data. This option configure a hint of maximum value of memory
    # to use when processing these records.
    #
    # storage.backlog.mem_limit 5M

[INPUT]
    Name              tail
    Path              /home/hoan/output_results.txt
    Tag               access
    Refresh_Interval  5
    Rotate_Wait       5
    Mem_Buf_Limit     5MB
    Skip_Long_Lines   On
    Parser            Alert

[OUTPUT]
    Name              es
    Match             access
    Host              192.168.135.144
    Port              9200
    HTTP_User         elastic
    HTTP_Passwd       changeme
    Suppress_Type_Name On
    Logstash_Format    On
    Retry_Limit        False
    Replace_Dots       On
    Logstash_Prefix    alert_test
hoan@hoan:~$
```

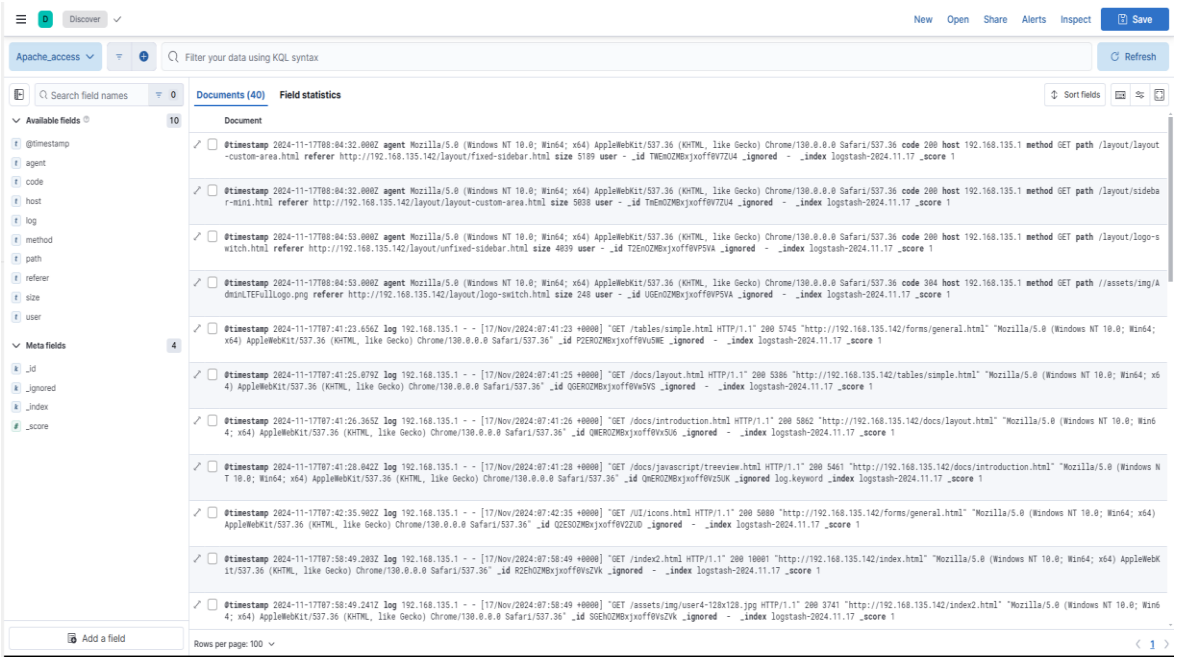
*Hình 3.7: Cấu hình fluent-bit đẩy log đã phân tích lên hệ thống Elasticsearch*

### 3.2.3. Giao diện hệ thống

Elasticsearch là một nền tảng mạnh mẽ được thiết kế để lưu trữ, tìm kiếm và phân tích dữ liệu lớn theo thời gian thực. Với khả năng tìm kiếm và phân tích dữ liệu nhanh chóng, Elasticsearch thường được sử dụng để trực quan hóa dữ liệu sau khi phân tích. Nó cung cấp một giao diện trực quan cho phép người dùng dễ dàng truy cập, tương tác và phân tích dữ liệu qua các biểu đồ, bảng và báo cáo. Thông qua việc tích hợp với các công cụ như Kibana, Elasticsearch giúp doanh nghiệp có thể theo dõi, phân tích và đưa ra quyết định dựa trên dữ liệu một cách hiệu quả và trực quan. Điều này đặc biệt hữu ích trong việc theo dõi hoạt động hệ thống, phân tích log hoặc khai thác dữ liệu phục vụ nhu cầu kinh doanh và an ninh mạng.

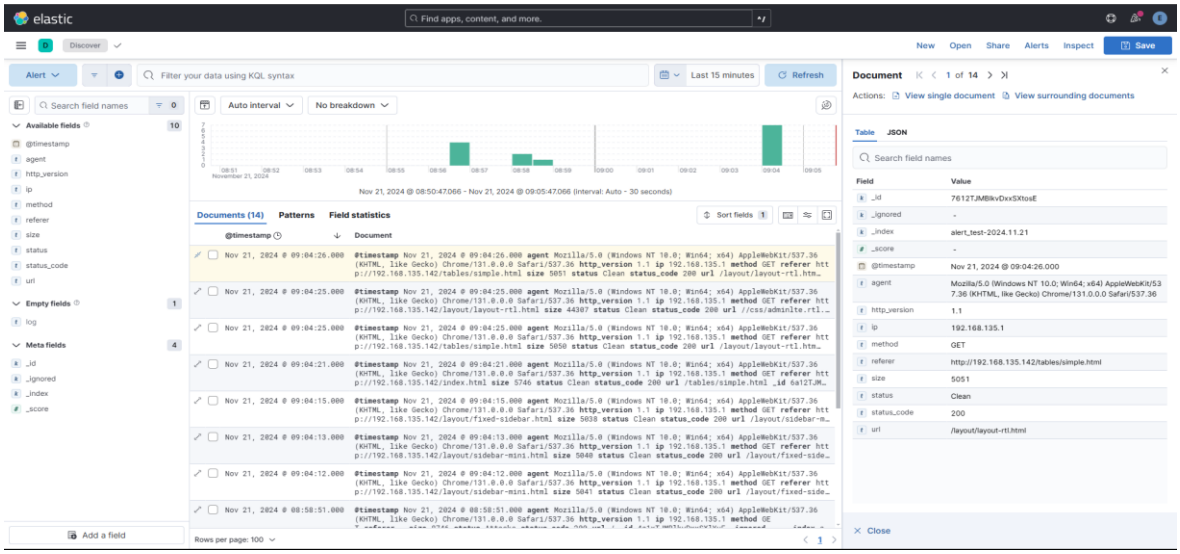



*Hình 3.8: Giao diện đăng nhập của hệ thống*

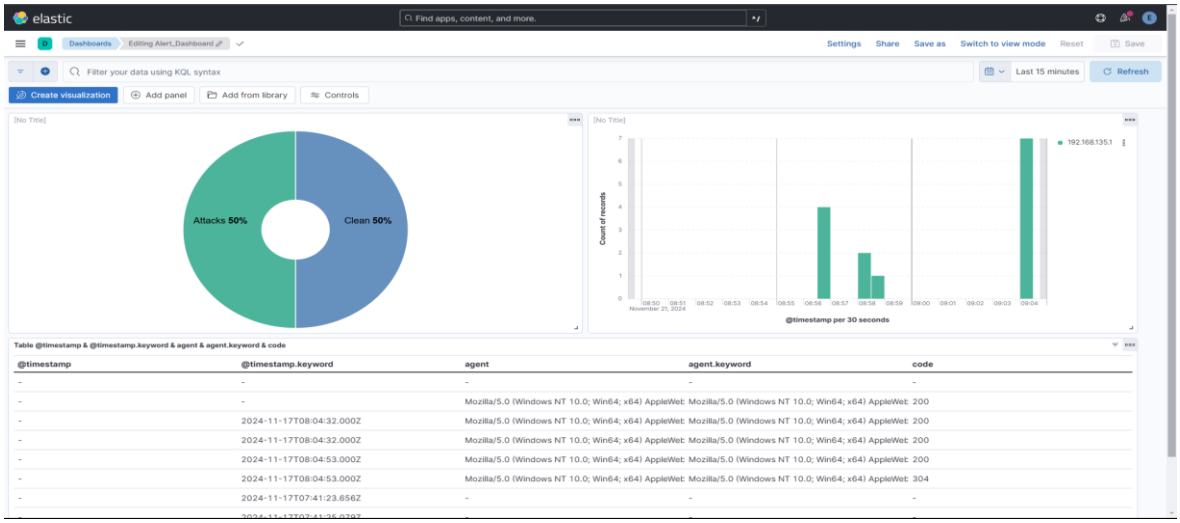



*Hình 3.9: Hình ảnh log được đẩy về hệ thống Elastic*

Hình ảnh cho thấy các bản ghi log của một hệ thống đã được đẩy về và hiển thị trên giao diện của Elastic Stack, cụ thể là Kibana. Các log được phân loại theo trường dữ liệu như @timestamp, agent, host, và method. Giao diện hiển thị từng dòng log chứa thông tin chi tiết về thời gian truy cập, địa chỉ IP của client, phương pháp HTTP (GET/POST), và tài nguyên được yêu cầu trên máy chủ. Những thông tin này giúp người quản trị hệ thống theo dõi các hoạt động trên máy chủ web, phân tích hành vi truy cập, phát hiện các vấn đề bất thường hoặc hành vi tiềm ẩn nguy cơ. Công cụ mạnh mẽ như Elastic Stack giúp tổ chức tối ưu hóa việc giám sát và quản lý log một cách trực quan và hiệu quả.



*Hình 3.10: Hình ảnh log được mô hình đánh giá*




*Hình 3.11: Giao diện thống kê*

## 3.3. Kết luận chương 3

Chương 3 đã trình bày về việc triển khai hệ thống cảnh báo tấn công website. Quá trình triển khai bao gồm các bước từ cài đặt môi trường, xử lý các luồng của dữ liệu, tích hợp mô hình với các chức năng cảnh báo vào một giao diện để dễ dàng theo dõi cho người dùng. Hệ thống bước đầu cho thấy khả năng phát hiện tấn công với độ chính xác cao, đáp ứng yêu cầu thời gian thực và tạo tiền đề cho các cải tiến trong tương lai.

# KẾT LUẬN VÀ KHUYẾN NGHỊ

## Kết quả đạt được

Qua quá trình tìm hiểu, nghiên cứu tham khảo các nguồn tài liệu cùng với sự hướng dẫn tận tình cả thầy giáo. Nghiên cứu đã đặt được một số kết quả như sau:

- Xây dựng được một mô hình phát hiện được các cuộc tấn công Website sử dụng mô hình học máy
- Xây dựng được một hệ thống cảnh báo và có giao diện tương tác với người dùng đơn giản
- Khắc phục được các điểm yếu của các giải pháp phát hiện trước đó và đạt được độ chính xác cao

## Hạn chế

Bên cạnh những kết quả đạt được, nghiên cứu vẫn còn một số hạn chế về tập dữ liệu huấn luyện chưa được đa dạng.

## Hướng phát triển

Để có thể giải quyết được một số mặt hạn chế của đề tài thì cần có những hướng phát triển như sau:

- Cần thu thập thêm nhiều tập dữ liệu huấn luyện.
- Xây dựng, tích hợp thêm hệ thống phát hiện dựa trên chữ kí, tăng khả năng phát hiện các trường hợp khó phát hiện khi sử dụng học máy để giúp giảm cảnh báo sai và tỉ lệ bỏ sót
- Cải tiến thêm để có thể nhận dạng và phân tích trên tất cả các log khác như log Nginx,…
- Cải tiến phát hiện được các payload trong body của POST request

# DANH MỤC TÀI LIỆU THAM KHẢO


[1] Government Portal, "Various emerging cybersecurity issues and forecasts for 2024," Online. Available: https://baochinhphu.vn/nhieu-van-de-noi-com-ve-an-ninh-mang-va-du-bao-nam-2024-102240118155026211.htm.

[2] OWASP, "OWASP ModSecurity Core Rule Set," Online. Available: https://www.owasp.org/index.php/Category:OWASP_ModSecurity_Core_Rule_Set_Project. [Accessed: June 2020].

[3] A. Baranwal, "Approaches to Detect SQL Injection & XSS in Web Applications," Online. Available: https://blogs.ubc.ca/computersecurity/files/2012/04/ABaranwal_ApproachesToDetectSQLinjection_XSSinWebApplication.pdf.

[4] K. Kemalis and T. Tzouramanis, "SQL-IDS: A Specification-based Approach for SQL Injection Detection," in *Proc. ACM Symp. Appl. Comput. (SAC'08)*, Fortaleza, Ceará, Brazil, 2008.

[5] P. Bisht and V. N. Venkatakrishnan, "XSS-GUARD: Precise Dynamic Prevention of Cross-Site Scripting Attacks," in *Proc. 5th Conf. Detect. Intrusions Malware & Vulnerability Assess. (DIMVA)*, LNCS 5137, pp. 23-43, 2008.

[6] J. Liang, W. Zhao, and W. Ye, "Anomaly-Based Web Attack Detection: A Deep Learning Approach," in *Proc. Int. Conf. Netw., Commun. Comput. (ICNCC 2017)*, Kunming, China, 2017.

[7] M. N. Huda, M. R. Hasan, and M. A. Alam, "Web Application Attacks Detection Using Machine Learning Techniques," *Int. J. Online Biomed. Eng. (iJOE)*, vol. 15, no. 12, 2019.

[8] D. Kar, S. Panigrahi, and S. Sundararajan, "SQLIDDS: SQL Injection Detection Using Document Similarity Measure," *J. Comput. Security*, vol. 24, no. 4, pp. 507–539, 2016.

[9] ResearchGate, "Detecting XSS Attacks Using Ensemble Learning," Online. Available: https://www.researchgate.net/publication/372629272_Phat_hien_tan_cong_XSS_su_dung_hoc_may_ket_hop_Detect_XSS_Attack_Using_Ensemble_Learning.

[10] Vietnix, "What is a Cyberattack?," Online. Available: https://vietnix.vn/tan-cong-mang-la-gi/.

[11] S. Rathor, "Simple RNN vs. GRU vs. LSTM: Difference Lies in More Flexible Control," *Medium*, 2021. Online. Available: https://medium.com/@saurabh.rathor092/simple-rnn-vs-gru-vs-lstm-difference-lies-in-more-flexible-control-5f33e07b1e57.

[12] "https://www.researchgate.net/publication/372629272_Phat_hien_tan_cong_XSS_su_dung_hoc_may_ket_hop_Detect_XSS_Attack_Using_Ensemble_Learning," [Online].

[13] NTTuan, "Lesson 14: Long Short-Term Memory (LSTM)," Online. Available: https://nttuan8.com/bai-14-long-short-term-memory-lstm/.

[14] T. T. Nguyen, "Detection of common web attacks using machine learning based on web logs," *Proc. Vietnam Academy of Science and Technology (VAST)*, 2021. Online. Available: https://vap.ac.vn/Portals/0/TuyenTap/2021/6/18/db78b606b7cc49b6873265977492fa99/60_FAIR2020_paper_155.pdf.

[15] H. T. Tran, "Application of representation learning for phishing attack detection," *Master Thesis, Posts and Telecommunications Institute of*

*Technology (PTIT)*, 2020. Online. Available: https://ptithcm.edu.vn/wp-content/uploads/2023/03/2020_HTTT_TranHuynhTien_LV.pdf.

[16] N. H. Vo and T. B. Le, "Study on solutions for detecting web attacks using machine learning," *Doctoral Dissertation, Graduate University of Science and Technology (GUST)*, 2022. Online. Available: https://gust.edu.vn/media/30/uftai-ve-tai-day30597.pdf.

[17] M. A. Rahman, M. A. A. Mamun, and M. A. Hossain, "Web Attack Intrusion Detection System Using Machine Learning Algorithms," *Int. J. of Online and Biomedical Engineering (iJOE)*, vol. 15, no. 12, 2019. Online. Available: https://online-journals.org/index.php/i-joe/article/view/45249.

[18] S. Sharma and R. Kumar, "Deep Learning Techniques for Web-Based Attack Detection in Industry 5.0," *Technologies*, vol. 11, no. 4, p. 107, 2023. Online. Available: https://www.mdpi.com/2227-7080/11/4/107.

[19] Y. Wang, Z. Li, and X. Zhang, "Detection of Web Attacks Using Machine Learning Based URL Analysis," in *Proc. IEEE Int. Conf. Comput. and Commun. Secur. (CCS)*, 2022. Online. Available: https://ieeexplore.ieee.org/document/9847838.

[20] M. K. Khan, A. Alam, and F. Ahmad, "Web Application Attacks Detection Using Machine Learning Techniques," *ResearchGate*, 2021. Online. Available: https://www.researchgate.net/publication/329514915_Web_Application_Attacks_Detection_Using_Machine_Learning_Techniques.

[21] H. Liu, J. Wang, and T. Chen, "Detecting Command Injection Attacks in Web Applications Based on Machine Learning," *Sci. Rep.*, vol. 14,

no. 74350, 2024. Online. Available: https://www.nature.com/articles/s41598-024-74350-3.