ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN TỐT NGHIỆP

NGÀNH: CÔNG NGHỆ THÔNG TIN

CHUYÊN NGÀNH: AN TOÀN THÔNG TIN

ĐỀ TÀI:

XÂY DỰNG HỆ THỐNG GIÁM SÁT VÀ PHÁT HIỆN SỰ CỐ HỆ THỐNG TỰ ĐỘNG THÔNG QUA PHÂN TÍCH LOG

Người hướng dẫn: **ThS. NGUYỄN CÔNG DANH**
Sinh viên thực hiện: **VŨ XUÂN HOÀNG**
Số thẻ sinh viên: **102210060**
Lớp: **21TCLC_DT1**

Đà Nẵng, 01/2026

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN TỐT NGHIỆP

NGÀNH: CÔNG NGHỆ THÔNG TIN

CHUYÊN NGÀNH: AN TOÀN THÔNG TIN

ĐỀ TÀI:

XÂY DỰNG HỆ THỐNG GIÁM SÁT VÀ PHÁT HIỆN SỰ CỐ HỆ THỐNG TỰ ĐỘNG THÔNG QUA PHÂN TÍCH LOG

Người hướng dẫn: **ThS. NGUYỄN CÔNG DANH**
Sinh viên thực hiện: **VŨ XUÂN HOÀNG**
Số thẻ sinh viên: **102210060**
Lớp: **21TCLC_DT1**

Đà Nẵng, 01/2026

TÓM TẮT

Tên đề tài: Xây dựng hệ thống giám sát và phát hiện sự cố hệ thống tự động thông qua phân tích log

Sinh viên thực hiện: Vũ Xuân Hoàng

Số thẻ SV: 102210060 Lớp : 21TCLC_DT1

Trong bối cảnh xu thế chuyển đổi số và sự phát triển mạnh mẽ của công nghệ thông tin, các hệ thống công nghệ ngày càng trở nên phức tạp và đóng vai trò quan trọng trong hoạt động của doanh nghiệp và tổ chức. Việc giám sát hệ thống và phát hiện kịp thời các sự cố phát sinh là yêu cầu cấp thiết nhằm đảm bảo tính ổn định, an toàn và liên tục của hệ thống. Tuy nhiên, trên thực tế, quá trình giám sát log hệ thống vẫn còn mang tính thủ công, phân tán và phụ thuộc nhiều vào kinh nghiệm của người quản trị, gây khó khăn trong việc phát hiện sớm các dấu hiệu bất thường và xử lý sự cố kịp thời.

Xuất phát từ thực tế đó, đề tài **"Xây dựng hệ thống giám sát và phát hiện sự cố hệ thống tự động thông qua phân tích log"** được triển khai với mục tiêu phát triển một hệ thống giám sát tập trung, trực quan và dễ sử dụng, tập trung vào các chức năng cốt lõi sau:

- **Thu thập và lưu trữ log tập trung** từ các máy chủ và ứng dụng, hỗ trợ truy vấn và theo dõi theo thời gian thực.
- **Phân tích và phát hiện bất thường tự động** dựa trên mô hình trí tuệ nhân tạo, nhằm nhận diện sớm các dấu hiệu sự cố và hành vi bất thường.
- **Cảnh báo kịp thời** thông qua hệ thống thông báo, hỗ trợ người quản trị nhanh chóng phản ứng và xử lý sự cố.

Mục tiêu cuối cùng của đề tài là cung cấp cho người quản trị hệ thống một giải pháp giám sát hiệu quả, giúp giảm đáng kể thời gian phát hiện và xử lý sự cố so với phương pháp truyền thống, đồng thời nâng cao độ tin cậy và tính an toàn trong vận hành hệ thống công nghệ thông tin.

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên: Vũ Xuân Hoàng Số thẻ sinh viên: 102210060

Lớp: 21TCLC_DT1 Khoa: Công nghệ thông tin Ngành: An toàn thông tin

1. *Tên đề tài đồ án:*

   Xây dựng hệ thống giám sát và phát hiện sự cố hệ thống tự động thông qua phân tích log

2. *Đề tài thuộc diện: ☐ Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện*
3. *Các số liệu và dữ liệu ban đầu:*

   (Không có)
4. *Nội dung các phần thuyết minh và tính toán:*

   Nội dung các phần thuyết minh bao gồm:

   - **Mở đầu:** Phần mở đầu, giới thiệu về như cầu thực tế và lý do thực hiện đề tài, đồng thời giới thiệu tổng quan về đề tài và mục tiêu đạt được, các tính năng và đối tượng
   - **Chương 1 – Cơ sở lý thuyết:** Trình bày các kiến thức, lý thuyết được áp dụng vào trong đề tài
   - **Chương 2 – Ứng dụng AI vào bài toán phân tích log hệ thống:** Giới thiệu trí tuệ nhân tạo, học máy và các kỹ thuật phát hiện bất thường áp dụng trong đề tài.
   - **Chương 3 – Triển khai hệ thống và đánh giá kết quả:** Mô tả quá trình triển khai hệ thống và các kết quả đạt được.
   - **Chương 4 – Kết luận và hướng phát triển:** Tổng kết kết quả, nêu hạn chế và đề xuất hướng phát triển trong tương lai.

5. *Các bản vẽ, đồ thị (ghi rõ các loại và kích thước bản vẽ):*

   (Không có)

6. *Họ tên người hướng dẫn:* ThS. Nguyễn Công Danh

7. *Ngày giao nhiệm vụ đồ án: 17 / 11 / 2025*

8. *Ngày hoàn thành đồ án: 21 / 01 / 2026*

Đà Nẵng, ngày tháng năm 2026

Trưởng Bộ môn …….. **Người hướng dẫn**

LỜI NÓI ĐẦU

Trong bối cảnh xu thế chuyển đổi số và sự phát triển mạnh mẽ của công nghệ thông tin, các hệ thống công nghệ ngày càng trở nên phức tạp và đóng vai trò quan trọng trong hoạt động của doanh nghiệp và tổ chức. Việc đảm bảo hệ thống vận hành ổn định, an toàn và phát hiện kịp thời các sự cố phát sinh là yêu cầu cấp thiết trong công tác quản trị và vận hành hệ thống. Tuy nhiên, trên thực tế, việc giám sát và phân tích log hệ thống vẫn còn mang tính thủ công, phân tán, phụ thuộc nhiều vào kinh nghiệm của người quản trị, gây khó khăn trong việc phát hiện sớm các dấu hiệu bất thường và xử lý sự cố kịp thời.

Đề tài **"Xây dựng hệ thống giám sát và phát hiện sự cố hệ thống tự động thông qua phân tích log"** được thực hiện với mục tiêu xây dựng một hệ thống giám sát tập trung, trực quan và có khả năng tự động phát hiện các hành vi bất thường dựa trên dữ liệu log hệ thống. Hệ thống cho phép thu thập, lưu trữ và trực quan hóa log theo thời gian thực, đồng thời ứng dụng các kỹ thuật học máy để hỗ trợ phát hiện sớm sự cố, góp phần nâng cao hiệu quả vận hành và giảm thiểu rủi ro cho hệ thống.

Thông qua đề tài này, vận dụng các công nghệ giám sát hiện đại kết hợp với trí tuệ nhân tạo vào bài toán thực tế, từ đó rèn luyện kỹ năng nghiên cứu, triển khai hệ thống và giải quyết vấn đề trong lĩnh vực công nghệ thông tin.

Em xin chân thành cảm ơn **Ban Giám hiệu, Khoa Công nghệ Thông tin – Trường Đại học Bách khoa, Đại học Đà Nẵng** đã tạo điều kiện về cơ sở vật chất, tài liệu tham khảo và môi trường học tập thuận lợi để em có thể hoàn thành đồ án tốt nghiệp.

Đặc biệt, em biết ơn **Thạc sĩ Nguyễn Công Danh –** người hướng chính, đã tận tình định hướng đề tài và góp ý chi tiết.

Xin tri ân gia đình và bạn bè – những người luôn là điểm tựa tinh thần, khích lệ em trong suốt quá trình học tập và thực hiện đề tài.

Trong quá trình hoàn báo cáo, chắc chắn còn nhiều thiếu sót; em rất mong nhận được những góp ý quý báu từ Thầy/Cô, anh chị và các bạn để đề tài được hoàn thiện hơn, trở thành hành trang hữu ích cho con đường sự nghiệp sau này.

Một lần nữa, em xin chân thành cảm ơn!

CAM ĐOAN

Em xin cam đoan:

1. Đồ án tốt nghiệp tên đề tài: "Xây dựng hệ thống giám sát và phát hiện sự cố hệ thống tự động thông qua phân tích log" là công trình nghiên cứu của chính cá nhân em dưới sự hướng dẫn trực tiếp của giảng viên **Ths. Nguyễn Công Danh.**
2. Tất cả tài liệu tham khảo đều được trích dẫn đầy đủ và cụ thể.
3. Nếu có những sao chép không hợp lệ, vi phạm, em xin chịu hoàn toàn trách nhiệm.

Sinh viên thực hiện

Vũ Xuân Hoàng

MỤC LỤC

DANH SÁCH CÁC HÌNH VẼ

DANH SÁCH CÁC BẢNG

DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT

Từ viết tắt	Diễn giải
CNTT	Công Nghệ Thông Tin
AI	Artificial Intelligence - Trí tuệ nhân tạo
LLM	Large Language Model - Mô hình ngôn ngữ lớn
AWS	Amazon Web Services
API	Application Programming Interface - Giao diện lập trình ứng dụng
VM	Virtual Machine - Máy ảo
CPU	Central Processing Unit - Bộ xử lý trung tâm
RAM	Random Access Memory - Bộ nhớ truy cập ngẫu nhiên
HTTP	HyperText Transfer Protocol - Giao thức truyền siêu văn bản
IP	Internet Protocol - Giao thức Internet
LSTM	Long Short-Term Memory - Bộ nhớ dài-ngắn hạn
URI	Uniform Resource Identifier - Định danh tài nguyên thống nhất
URL	Uniform Resource Locator - Bộ định vị tài nguyên thống nhất
SSH	Secure Shell - Giao thức mạng

MỞ ĐẦU

1. Giới thiệu

Trong quá trình phát triển và vận hành các hệ thống công nghệ thông tin hiện nay, đặc biệt là các hệ thống máy chủ, ứng dụng web và dịch vụ triển khai trên nền tảng điện toán đám mây, dữ liệu log được sinh ra liên tục nhằm ghi nhận trạng thái hoạt động, thông tin lỗi và các sự kiện của hệ thống. Dữ liệu log là nguồn thông tin quan trọng giúp người quản trị theo dõi, đánh giá hiệu năng và phát hiện các sự cố trong quá trình vận hành.

Tuy nhiên, khi quy mô hệ thống ngày càng mở rộng, số lượng và mức độ phức tạp của log cũng tăng lên đáng kể. Việc giám sát và phân tích log theo phương pháp thủ công hoặc dựa trên các quy tắc cố định trở nên kém hiệu quả, tốn nhiều thời gian và dễ bỏ sót các bất thường. Trong nhiều trường hợp, sự cố chỉ được phát hiện khi hệ thống đã xảy ra lỗi nghiêm trọng, gây gián đoạn dịch vụ và ảnh hưởng đến người dùng.

Trước thực tế đó, nhu cầu xây dựng một hệ thống giám sát log tự động, có khả năng phát hiện sớm các bất thường trong hệ thống là rất cần thiết. Sự phát triển của trí tuệ nhân tạo (AI) và học máy đã mở ra khả năng áp dụng các mô hình phân tích dữ liệu để nhận diện các hành vi bất thường trong log mà không cần can thiệp thủ công. Bên cạnh đó, việc tích hợp các công cụ cảnh báo giúp người vận hành có thể nhanh chóng nắm bắt và xử lý sự cố hiệu quả hơn.

Xuất phát từ những vấn đề nêu trên, đề tài **"Xây dựng hệ thống giám sát và phát hiện sự cố hệ thống tự động thông qua phân tích log"** được thực hiện với mục tiêu xây dựng một hệ thống giám sát log tập trung cho môi trường hệ thống CNTT. Hệ thống cho phép thu thập log tự động, lưu trữ và hiển thị log theo thời gian thực, đồng thời ứng dụng mô hình AI để phát hiện bất thường và gửi cảnh báo khi có sự cố xảy ra.

Thông qua việc thực hiện đề tài, đề tài góp phần vận dụng các kiến thức về hệ thống, mạng máy tính, cơ sở dữ liệu, lập trình và trí tuệ nhân tạo để xây dựng một ứng dụng thực tế. Kết quả của đề tài hướng đến việc nâng cao hiệu quả giám sát và vận hành hệ thống CNTT, đồng thời tạo nền tảng cho việc nghiên cứu và phát triển các giải pháp giám sát hệ thống thông minh trong tương lai.

2. Mục đích và ý nghĩa của đề tài

Mục đích

Đề tài "Xây dựng hệ thống giám sát và phát hiện sự cố hệ thống tự động thông qua phân tích log" được triển khai với các mục tiêu chính sau:

- Xây dựng một nền tảng thống nhất để thu thập, lưu trữ và quản lý toàn bộ dữ liệu log phát sinh từ máy chủ, ứng dụng và các dịch vụ trong hệ thống CNTT, thay thế việc theo dõi log rời rạc trên từng máy hoặc công cụ riêng lẻ.
- Cho phép hệ thống thu thập và phân tích log theo thời gian thực, giảm sự phụ thuộc vào việc kiểm tra thủ công và nâng cao hiệu quả giám sát hệ thống.
- Ứng dụng mô hình AI để nhận diện các hành vi bất thường và dấu hiệu sự cố tiềm ẩn trong dữ liệu log, giúp phát hiện sớm các vấn đề mà phương pháp giám sát truyền thống khó nhận biết.
- Tích hợp cơ chế gửi cảnh báo tự động khi phát hiện bất thường hoặc lỗi hệ thống, hỗ trợ người vận hành nhanh chóng nắm bắt tình trạng và có biện pháp xử lý phù hợp.

Những mục tiêu trên hướng đến việc xây dựng một hệ thống giám sát log tập trung, tự động và thông minh, góp phần nâng cao độ ổn định và khả năng vận hành của hệ thống CNTT, đồng thời tạo nền tảng để mở rộng và phát triển các chức năng giám sát nâng cao trong tương lai.

Ý nghĩa

- **Ý nghĩa thực tiễn:** Đề tài góp phần đề xuất một giải pháp giám sát hệ thống CNTT tự động và thông minh, giúp giảm sự phụ thuộc vào việc theo dõi log thủ công. Việc phát hiện sớm các bất thường và sự cố giúp rút ngắn thời gian xử lý, giảm thiểu gián đoạn dịch vụ và nâng cao độ ổn định của hệ thống. Ngoài ra, cơ chế cảnh báo và hỗ trợ phân tích giúp người vận hành tiếp cận thông tin sự cố nhanh chóng và hiệu quả hơn.
- **Ý nghĩa học thuật và đào tạo:** Đề tài tạo điều kiện vận dụng tổng hợp các kiến thức chuyên ngành CNTT như hệ thống máy tính, mạng, lập trình và trí tuệ nhân tạo vào việc xây dựng một ứng dụng thực tế. Quá trình thực hiện đề tài góp phần rèn luyện kỹ năng phân tích yêu cầu, thiết kế kiến trúc hệ thống, triển khai, kiểm thử và đánh giá kết quả. Kết quả của đề tài là nền tảng cho việc nghiên cứu và phát triển các hệ thống giám sát thông minh trong môi trường CNTT.

3. Đối tượng và phạm vi của đề tài

Đối tượng nghiên cứu của đề tài là dữ liệu log hệ thống được sinh ra trong quá trình vận hành các hệ thống công nghệ thông tin, bao gồm máy chủ, ứng dụng web và các dịch vụ mạng. Các dữ liệu log này phản ánh trạng thái hoạt động của hệ thống, các sự kiện truy cập, lỗi phát sinh cũng như những hành vi bất thường có thể dẫn đến sự cố hoặc gián đoạn dịch vụ. Bên cạnh đó, đề tài còn tập trung nghiên cứu các phương pháp phân tích log, kỹ thuật giám sát hệ thống và mô hình trí tuệ nhân tạo có khả năng phát hiện bất thường dựa trên dữ liệu log.

Ngoài dữ liệu log, đối tượng nghiên cứu của đề tài còn bao gồm các công cụ và nền tảng hỗ trợ giám sát hệ thống như hệ thống thu thập log, lưu trữ log tập trung, công cụ trực quan hóa dữ liệu và cơ chế cảnh báo sự cố. Mô hình học máy được áp dụng trong đề tài đóng vai trò quan trọng trong việc tự động phân tích log, nhận diện các mẫu hành vi bất thường và hỗ trợ người quản trị hệ thống trong công tác giám sát và vận hành.

Phạm vi nghiên cứu của đề tài được giới hạn trong việc xây dựng và triển khai một hệ thống giám sát log tập trung cho môi trường hệ thống CNTT ở quy mô thử nghiệm. Hệ thống tập trung vào việc thu thập log từ các máy chủ và ứng dụng web, thực hiện lưu trữ, truy vấn và hiển thị log theo thời gian thực. Trên cơ sở dữ liệu log thu thập được, đề tài áp dụng các mô hình trí tuệ nhân tạo nhằm phát hiện bất thường và cảnh báo sớm các sự cố hệ thống.

Đề tài không đi sâu vào việc xử lý toàn bộ các loại tấn công phức tạp hay triển khai hệ thống ở quy mô doanh nghiệp lớn, mà chủ yếu tập trung vào việc nghiên cứu, thiết kế và đánh giá hiệu quả của giải pháp giám sát và phát hiện sự cố dựa trên phân tích log. Kết quả đạt được của đề tài đóng vai trò là nền tảng ban đầu, có thể mở rộng và phát triển trong tương lai để áp dụng cho các hệ thống CNTT có quy mô lớn hơn và yêu cầu cao hơn về độ tin cậy và an toàn.

4. Những kết quả dự kiến đạt được

Thông qua việc nghiên cứu và triển khai đề tài "Xây dựng hệ thống giám sát và phát hiện sự cố hệ thống tự động thông qua phân tích log", đồ án hướng tới đạt được các kết quả chính sau:

- **Xây dựng được một hệ thống giám sát log tập trung** cho môi trường hệ thống CNTT, cho phép thu thập, lưu trữ và truy vấn log từ máy chủ và ứng dụng một cách thống nhất, thay thế phương pháp theo dõi log thủ công và phân tán.
- **Triển khai thành công cơ chế giám sát và trực quan hóa log theo thời gian thực**, giúp người quản trị dễ dàng theo dõi tình trạng hoạt động của hệ thống, phát hiện sớm các dấu hiệu bất thường thông qua dashboard trực quan.
- **Ứng dụng mô hình học máy không giám sát Isolation Forest** để phân tích dữ liệu log và tự động phát hiện các hành vi bất thường, phục vụ cho bài toán phát hiện sự cố hệ thống và các hành vi tấn công tiềm ẩn.
- **Xây dựng và đánh giá mô hình phát hiện bất thường** thông qua các chỉ số định lượng như Precision, Recall và F1-score, đồng thời phân tích sự đánh đổi giữa các chỉ số này nhằm lựa chọn ngưỡng phát hiện phù hợp với yêu cầu vận hành thực tế.
- **Tích hợp cơ chế cảnh báo tự động**, cho phép hệ thống gửi thông báo kịp thời đến người quản trị khi phát hiện sự cố hoặc hành vi bất thường, góp phần rút ngắn thời gian phản ứng và xử lý sự cố.
- **Đề xuất một mô hình giám sát hệ thống thông minh có tính khả thi cao**, có thể mở rộng và ứng dụng trong thực tế, đồng thời làm nền tảng cho các nghiên

cứu và phát triển tiếp theo trong lĩnh vực giám sát hệ thống và an toàn thông tin.

Những kết quả dự kiến đạt được không chỉ đáp ứng mục tiêu nghiên cứu của đồ án mà còn góp phần nâng cao hiệu quả giám sát, vận hành và đảm bảo an toàn cho các hệ thống công nghệ thông tin hiện đại.

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

1.1. Tổng quan về công nghệ sử dụng

1.1.1. Dữ liệu log và ứng dụng kỹ thuật phân tích log trong giám sát hệ thống

Log hệ thống là các bản ghi phản ánh trạng thái hoạt động, sự kiện và lỗi phát sinh trong quá trình vận hành của hệ điều hành, ứng dụng và dịch vụ mạng. Dữ liệu log là nguồn thông tin quan trọng phục vụ giám sát, phân tích sự cố và đảm bảo hệ thống hoạt động ổn định, an toàn.

Việc giám sát hệ thống dựa trên dữ liệu log có thể được triển khai theo hai hướng tiếp cận chính. Hướng thứ nhất là sử dụng các phần mềm và nền tảng giám sát sẵn có, đặc biệt là các giải pháp mã nguồn mở, nhằm thu thập, lưu trữ và trực quan hóa log một cách tập trung. Hướng thứ hai là xây dựng các mô hình phân tích log, trong đó dữ liệu log được xử lý và phân tích bằng các phương pháp thống kê hoặc học máy để tự động phát hiện các hành vi bất thường và dấu hiệu sự cố.

Trên cơ sở các hướng tiếp cận này, nhiều công cụ và công nghệ giám sát log đã được phát triển và ứng dụng rộng rãi. Các phần tiếp theo sẽ trình bày tổng quan về những công nghệ được sử dụng trong hệ thống giám sát log.

1.1.2. Grafana

Grafana là một nền tảng mã nguồn mở mạnh mẽ, được sử dụng để phân tích và trực quan hóa dữ liệu từ nhiều nguồn khác nhau. Công cụ này giúp người dùng theo dõi và phân tích hiệu suất của hệ thống cũng như ứng dụng theo thời gian thực [1].

Thông qua các bảng điều khiển (dashboard) tương tác, Grafana hiển thị dữ liệu dưới dạng biểu đồ và đồ thị trực quan, có tính thẩm mỹ cao và dễ tùy chỉnh. Nhờ khả năng giám sát linh hoạt và thiết lập cảnh báo kịp thời, Grafana trở thành công cụ rất phổ biến trong lĩnh vực DevOps và giám sát hệ thống.

Tính năng:

- **Kết nối đa dạng nguồn dữ liệu:** Hỗ trợ nhiều loại nguồn như Prometheus, InfluxDB, MySQL, Elasticsearch, Loki, AWS CloudWatch, v.v…
- **Bảng điều khiển tùy chỉnh:** Cho phép tạo dashboard với nhiều loại biểu đồ (đường, cột, gauge,...) để hiển thị dữ liệu một cách trực quan, dễ hiểu.
- **Giám sát theo thời gian thực:** Hiển thị dữ liệu liên tục, giúp nhận diện vấn đề ngay lập tức.
- **Cảnh báo (Alerting):** Thiết lập cảnh báo khi các chỉ số vượt ngưỡng, gửi thông báo qua Email, Slack,....
- **Mã nguồn mở và linh hoạt:** Cộng đồng phát triển mạnh mẽ, có thể mở rộng tính năng bằng plugin, phù hợp với nhiều môi trường từ on-premise đến cloud.

Vai trò trong đề tài:

Grafana được sử dụng để hiển thị log, hỗ trợ theo dõi và đánh giá trạng thái hệ thống trước và sau khi áp dụng các mô hình AI.

1.1.3. Loki

Loki là một hệ thống logging và lưu trữ log cho các hệ thống phân tán và điều khiển vận hành. Nó được tạo ra để hoạt động cùng với Grafana, một công cụ giám sát mã nguồn mở phổ biến, như một giải pháp toàn diện cho việc giám sát hệ thống [2].

Đặc điểm nổi bật của Loki:

- Lập chỉ mục theo nhãn (labels).
- Tối ưu cho môi trường cloud-native.
- Sử dụng ngôn ngữ truy vấn LogQL.
- Tích hợp chặt chẽ với Grafana.

Vai trò trong đề tài:

Loki đóng vai trò là kho lưu trữ log tập trung, cung cấp dữ liệu đầu vào cho quá trình trực quan hóa và phân tích log bằng trí tuệ nhân tạo.

1.1.4. Promtail

Promtail là một bộ thu log được tạo ra đặc biệt cho Loki. Nó sử dụng cùng cơ chế khám phá dịch vụ của Prometheus và có các tính năng tương tự để gắn thẻ, chuyển đổi và lọc logs trước khi đưa vào Loki [3].

Đặc điểm nổi bật của Promtail:

- Hoạt động theo mô hình agent, chạy trực tiếp trên node cần giám sát.
- Hỗ trợ cấu hình linh hoạt bằng YAML (YAML Ain’t Markup Language).
- Gắn nhãn (labels) cho log để phục vụ truy vấn và phân loại.
- Hỗ trợ tiền xử lý log như lọc, parse và transform dữ liệu.

Vai trò trong đề tài:

Promtail được sử dụng để thu thập log từ các máy chủ và ứng dụng, đảm bảo dữ liệu log được gửi liên tục và đồng bộ về hệ thống phân tích trung tâm.

1.1.5. AWS

AWS là nền tảng điện toán đám mây cung cấp hạ tầng linh hoạt cho việc triển khai hệ thống CNTT [4].

Đặc điểm nổi bật:

- Hạ tầng linh hoạt, dễ mở rộng.
- Độ sẵn sàng cao.
- Hỗ trợ nhiều dịch vụ CNTT.

Vai trò trong đề tài:

AWS được sử dụng để triển khai thử nghiệm hệ thống, đánh giá khả năng vận hành trong môi trường thực tế.

1.1.6. Nginx

Nginx là web server mã nguồn mở hiệu năng cao, thường được sử dụng làm web server và reverse proxy trong các hệ thống hiện đại [5].

Đặc điểm nổi bật:

- Hoạt động theo mô hình bất đồng bộ.
- Xử lý số lượng lớn request cùng lúc.
- Hỗ trợ reverse proxy cho các dịch vụ phía sau.

Vai trò trong đề tài:

Nginx được sử dụng làm web server và reverse proxy, đồng thời tạo ra các log truy cập và log lỗi để phục vụ cho quá trình giám sát và phát hiện sự cố hệ thống.

1.1.7. Telegram

Telegram là một nền tảng nhắn tin đa nền tảng, cung cấp khả năng trao đổi thông tin theo thời gian thực thông qua Internet. Telegram hỗ trợ nhiều hình thức giao tiếp như tin nhắn văn bản, hình ảnh, tệp tin và đặc biệt là Telegram Bot API, cho phép các hệ thống bên ngoài tự động gửi và nhận thông báo thông qua bot.

Nhờ đặc tính đơn giản, tốc độ nhanh và khả năng tích hợp linh hoạt, Telegram được sử dụng rộng rãi trong các hệ thống giám sát và cảnh báo nhằm gửi thông tin sự cố kịp thời đến người quản trị.

Đặc điểm nổi bật:

- Hỗ trợ Telegram Bot API, cho phép tích hợp dễ dàng với các hệ thống giám sát và ứng dụng bên ngoài.
- Hoạt động theo thời gian thực, độ trễ thấp.
- Không yêu cầu hạ tầng phức tạp phía người dùng, chỉ cần ứng dụng Telegram trên thiết bị di động hoặc máy tính.
- Hỗ trợ gửi tin nhắn, cảnh báo và dữ liệu dạng text một cách tự động.

Vai trò trong đề tài:

Telegram được sử dụng như kênh cảnh báo tự động cho hệ thống giám sát và phát hiện sự cố. Khi mô hình AI phát hiện các hành vi bất thường hoặc sự cố hệ thống từ dữ liệu log, thông tin cảnh báo sẽ được gửi ngay lập tức đến người quản trị thông qua Telegram Bot. Cơ chế này giúp rút ngắn thời gian phát hiện và phản ứng sự cố, nâng cao hiệu quả vận hành và đảm bảo an toàn cho hệ thống CNTT.

1.2. Kết chương

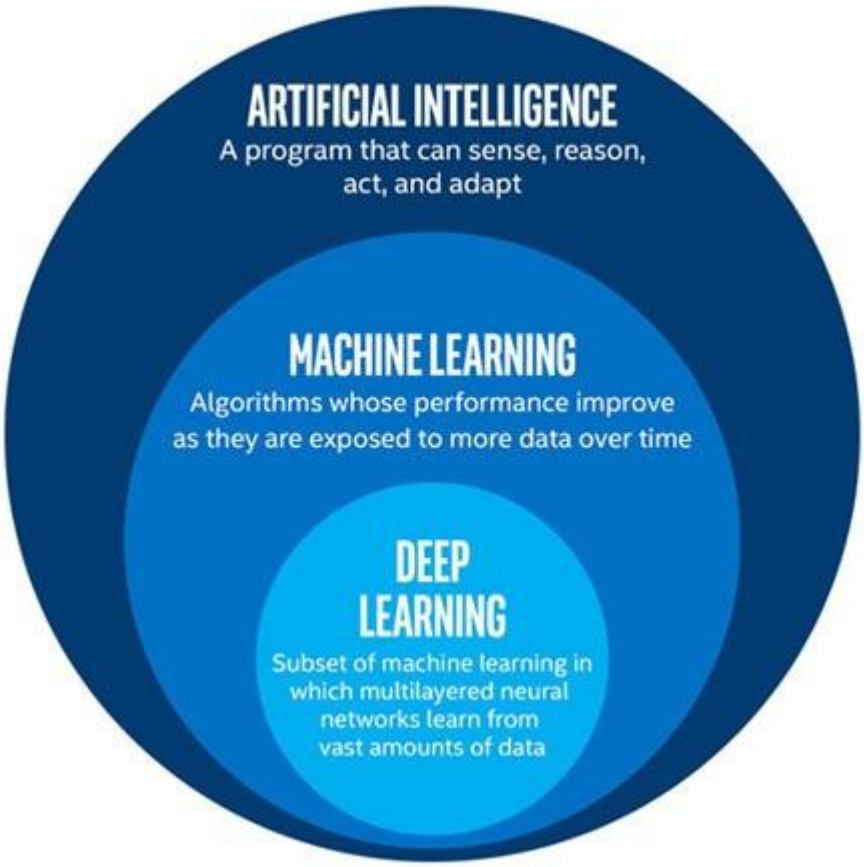
Chương 1 đã trình bày các cơ sở lý thuyết và tổng quan về những công nghệ được sử dụng trong đề tài "Xây dựng ứng dụng phân tích log tích hợp AI nhằm phát hiện và cảnh báo sự cố hệ thống". Nội dung chương tập trung giới thiệu các công nghệ thu thập, lưu trữ và trực quan hóa log, các kỹ thuật trí tuệ nhân tạo phục vụ phát hiện bất thường, cũng như các nền tảng hỗ trợ triển khai hệ thống.

Những kiến thức được trình bày trong chương này là nền tảng cho việc phân tích yêu cầu và thiết kế kiến trúc hệ thống ở các chương tiếp theo, đặc biệt là Chương 2 – Ứng dụng AI vào bài toán phân tích log hệ thống.

CHƯƠNG 2: ỨNG DỤNG AI VÀO BÀI TOÁN PHÂN TÍCH LOG HỆ THỐNG

2.1. Khái niệm về trí tuệ nhân tạo

Trí tuệ nhân tạo (AI - Artificial Intelligence) là một công nghệ cho phép máy móc, đặc biệt là máy tính, thực hiện các nhiệm vụ đòi hỏi trí tuệ như con người. Không giống như lập trình truyền thống chỉ dựa trên logic, trí tuệ nhân tạo sử dụng các hệ thống học máy (machine learning) để học hỏi và mô phỏng các hoạt động như suy nghĩ, lập luận và tự thích nghi [6].



Hình 2.1 - Trí tuệ nhân tạo

Ứng dụng của trí tuệ nhân tạo rất đa dạng, từ việc nhận dạng hình ảnh, xử lý ngôn ngữ tự nhiên, đến hỗ trợ khách hàng thông qua chatbot. AI cũng giúp phân tích khối lượng dữ liệu lớn để đưa ra các dự đoán thông minh, cải thiện hiệu quả kinh doanh và giải quyết các vấn đề phức tạp. Mục tiêu cuối cùng của trí tuệ nhân tạo AI là tạo ra các hệ thống thông minh có khả năng tương tác và đáp ứng linh hoạt như con người, từ đó mang lại giá trị to lớn trong nhiều lĩnh vực.

Đối với các hệ thống công nghệ thông tin hiện đại, AI đóng vai trò quan trọng trong việc tự động phân tích dữ liệu log, nhận diện các hành vi bất thường và hỗ trợ phát hiện sớm các sự cố tiềm ẩn mà các phương pháp giám sát truyền thống khó đáp ứng hiệu quả.

2.2. Học máy trong Trí tuệ nhân tạo

Machine Learning (Máy học) là một lĩnh vực của Trí tuệ nhân tạo (AI), cho phép hệ thống máy tính có khả năng tự "học" từ dữ liệu để thực hiện các tác vụ cụ thể mà không cần được lập trình một cách tường minh. Machine Learning dựa trên các thuật toán có khả năng nhận dạng mẫu và tối ưu hóa theo dữ liệu, giúp máy tính nâng cao hiệu quả theo thời gian.Thay vì phải tuân theo một bộ quy tắc được mã hóa cứng, các thuật toán Machine Learning sẽ phân tích một lượng lớn dữ liệu đầu vào, nhận diện các quy luật, mẫu (patterns) tiềm ẩn và từ đó xây dựng nên một mô hình (model) để đưa ra dự đoán hoặc quyết định, phản ánh sự phát triển của trí tuệ nhân tạo. Mục tiêu chính là tự động hóa và thông minh hóa các quy trình, giúp máy tính có thể tự cải thiện hiệu suất dựa trên "kinh nghiệm" thu thập được từ dữ liệu [7].

Trong bối cảnh giám sát hệ thống CNTT, học máy được ứng dụng để phân tích dữ liệu log, dữ liệu hiệu năng và sự kiện vận hành nhằm phát hiện các dấu hiệu bất thường, từ đó hỗ trợ cảnh báo và xử lý sự cố kịp thời.

2.3. Phân loại các phương pháp học máy

2.3.1. Học có giám sát

Đây là phương pháp phổ biến nhất. Với Học có giám sát, mô hình được huấn luyện bằng một bộ "dữ liệu đã được gán nhãn" (labeled data). Điều này có nghĩa là mỗi điểm dữ liệu đầu vào (input) đều đi kèm với một kết quả đầu ra (output) chính xác. Mục tiêu của mô hình là học ra một quy luật tổng quát để ánh xạ từ input đến output. Sau khi được huấn luyện, mô hình có thể dự đoán kết quả cho những dữ liệu mới chưa từng thấy trước đây. Các ví dụ điển hình bao gồm: phân loại email là spam hay không spam, dự đoán giá nhà dựa trên các đặc điểm như diện tích, vị trí, số phòng ngủ, hoặc nhận dạng chữ viết tay [7].

2.3.2. Học không giám sát

Trái ngược với học có giám sát, Học không giám sát làm việc với "dữ liệu không được gán nhãn" (unlabeled data). Vì không có kết quả đầu ra đúng để đối chiếu, mục tiêu của mô hình là tự mình khám phá ra các cấu trúc, các mẫu hoặc các mối quan hệ ẩn giấu bên trong dữ liệu. Thay vì dự đoán, phương pháp này tập trung vào việc khám phá. Các ứng dụng phổ biến của học không giám sát bao gồm: phân cụm khách hàng (customer segmentation) dựa trên hành vi mua sắm để xây dựng chiến lược marketing hiệu quả, giảm chiều dữ liệu để trực quan hóa, hoặc phát hiện các giao dịch tài chính bất thường có khả năng gian lận [7].

2.3.3. Học tăng cường

Học tăng cường (Reinforcement Learning) là một phương pháp học độc đáo, trong đó một mô hình (được gọi là "tác nhân" - agent) học cách hành động trong một môi trường cụ thể để tối đa hóa phần thưởng tích lũy. Tác nhân sẽ thực hiện các hành động, quan sát trạng thái mới của môi trường và nhận về một tín hiệu "phần thưởng" (reward) nếu hành động đó tốt hoặc "hình phạt" (penalty) nếu hành động đó xấu. Thông qua quá trình thử và sai liên tục, tác nhân sẽ dần học được một "chính sách" (policy) - tức là chiến lược hành động tối ưu trong mọi tình huống. Các ví dụ kinh điển của học tăng cường bao gồm xe tự lái, robot tự động hóa trong nhà máy, bot chơi game, và tối ưu hóa hệ thống chuỗi cung ứng [7].

2.3.4. Học bán giám sát

Học bán giám sát là phương pháp lai giữa học có giám sát và không giám sát. Trong thực tế, việc gán nhãn cho dữ liệu thường rất tốn kém và mất thời gian, trong khi dữ liệu chưa gán nhãn lại rất dồi dào. Học bán giám sát ra đời để giải quyết vấn đề này. Mô hình sẽ được huấn luyện trên một tập dữ liệu nhỏ đã được gán nhãn và một tập dữ liệu lớn hơn nhiều chưa được gán nhãn. Mục tiêu là tận dụng cấu trúc của dữ liệu chưa được gán nhãn để cải thiện độ chính xác và hiệu suất của mô hình so với việc chỉ học trên dữ liệu có nhãn. Các ứng dụng thực tế bao gồm nhận dạng khuôn mặt trong một kho ảnh khổng lồ hoặc phân loại nội dung website [7].

2.4. Học máy không giám sát và bài toán phát hiện bất thường

Học máy không giám sát được ứng dụng rộng rãi trong nhiều bài toán thực tiễn, trong đó nổi bật là phân cụm dữ liệu, giảm chiều dữ liệu và phát hiện bất thường.

Phát hiện bất thường (Anomaly Detection) là bài toán xác định các mẫu dữ liệu có hành vi khác biệt đáng kể so với phần lớn dữ liệu còn lại. Trong bối cảnh hệ thống CNTT, các bất thường thường tương ứng với lỗi vận hành, sự cố hệ thống hoặc hành vi tấn công.

Do dữ liệu sự cố thường hiếm và khó thu thập nhãn, các phương pháp học máy không giám sát được xem là lựa chọn phù hợp cho bài toán phát hiện và cảnh báo sự cố hệ thống.

2.5. Bài toán phát hiện sự cố hệ thống và yêu cầu đặt ra

Trong các hệ thống giám sát, dữ liệu đầu vào thường bao gồm log hệ thống, chỉ số hiệu năng (CPU, bộ nhớ, độ trễ, lưu lượng mạng, v.v.) được thu thập liên tục theo thời gian. Các sự cố hệ thống thường biểu hiện dưới dạng các mẫu dữ liệu bất thường so với trạng thái vận hành bình thường.

Do đó, bài toán phát hiện sự cố hệ thống đặt ra các yêu cầu chính sau:

- Phát hiện sớm các dấu hiệu bất thường.
- Hoạt động hiệu quả với dữ liệu lớn và đa chiều.
- Không phụ thuộc vào dữ liệu huấn luyện có nhãn.
- Đáp ứng yêu cầu xử lý gần thời gian thực để phục vụ cảnh báo.

2.6. Isolation Forest

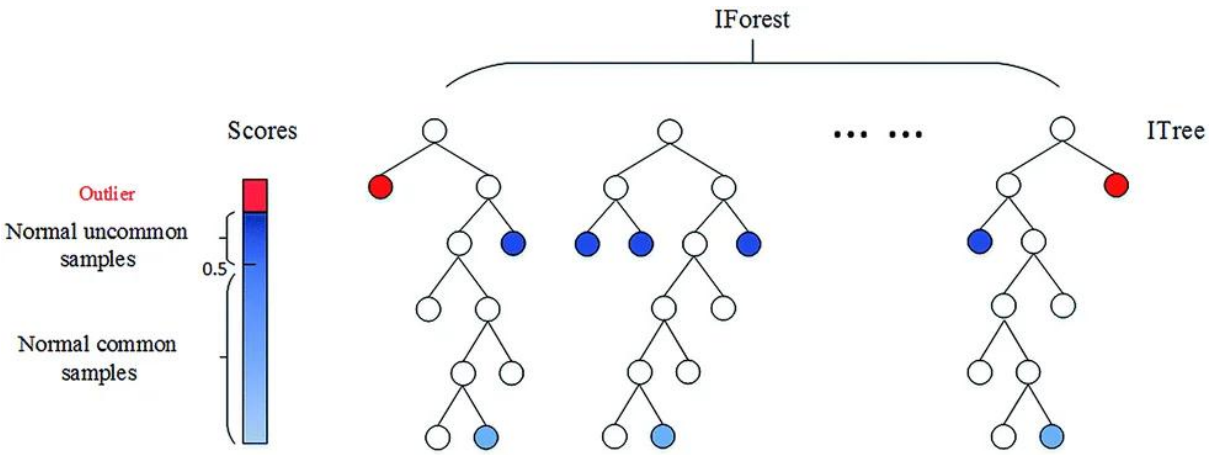
2.6.1. Khái niệm về Isolation Forest

Isolation Forest (iForest) là một phương pháp phát hiện bất thường (anomaly detection) không giám sát, dựa trên ý tưởng cô lập (isolation) các điểm dữ liệu bất thường thay vì mô hình hóa dữ liệu bình thường. Khác với các phương pháp truyền thống thường xây dựng hồ sơ của dữ liệu bình thường rồi tìm các điểm lệch chuẩn, Isolation Forest trực tiếp tìm cách tách riêng các điểm bất thường ra khỏi tập dữ liệu [8].

Ý tưởng cốt lõi của Isolation Forest dựa trên hai đặc tính quan trọng của dữ liệu bất thường:

- Thường chiếm số lượng rất ít.
- Có giá trị thuộc tính khác biệt rõ rệt so với phần lớn dữ liệu còn lại.

Nhờ đó, các điểm bất thường có xu hướng bị cô lập chỉ sau một số ít phép phân chia ngẫu nhiên, trong khi các điểm bình thường cần nhiều bước phân chia hơn. Isolation Forest tận dụng đặc điểm này để đánh giá mức độ bất thường của từng điểm dữ liệu thông qua độ dài đường đi trong cây.



Hình 2.2 - Thuật toán Isolation Forest

2.6.2. Nguyên lý hoạt động

Isolation Forest hoạt động dựa trên một tập hợp các cây cô lập (Isolation Tree – iTree), trong đó mỗi cây được xây dựng bằng cách phân chia dữ liệu một cách ngẫu nhiên.

Quá trình xây dựng một Isolation Tree diễn ra như sau:

- Tại mỗi nút, thuật toán chọn ngẫu nhiên một thuộc tính.
- Sau đó chọn ngẫu nhiên một giá trị phân chia nằm giữa giá trị nhỏ nhất và lớn nhất của thuộc tính đó.
- Dữ liệu được chia thành hai nhánh dựa trên điều kiện phân chia.

- Quá trình lặp lại cho đến khi mỗi điểm dữ liệu bị cô lập riêng lẻ hoặc cây đạt đến chiều cao giới hạn.

Trong cấu trúc này, độ dài đường đi (path length) của một điểm dữ liệu được xác định là số cạnh mà điểm đó đi qua từ nút gốc đến nút lá. Các điểm bất thường thường có độ dài đường đi ngắn, do chúng dễ bị tách ra khỏi phần còn lại của dữ liệu chỉ sau vài lần phân chia ngẫu nhiên. Ngược lại, các điểm bình thường cần nhiều lần phân chia hơn và do đó có đường đi dài hơn.

Isolation Forest xây dựng nhiều Isolation Tree (tạo thành một "rừng") bằng cách lấy các mẫu con ngẫu nhiên (sub-sampling) từ tập dữ liệu ban đầu. Độ dài đường đi trung bình của một điểm qua tất cả các cây trong rừng được sử dụng để đánh giá mức độ bất thường của điểm đó. Cách tiếp cận này giúp Isolation Forest đạt được độ phức tạp tuyến tính, sử dụng ít bộ nhớ và đặc biệt hiệu quả với tập dữ liệu lớn hoặc dữ liệu có số chiều cao [8].

2.6.3. Các chỉ số đánh giá trong Isolation Forest

Độ dài đường đi : Độ dài đường đi $h(x)$ của một điểm dữ liệu $x$là số cạnh mà điểm đó đi qua trong một Isolation Tree từ nút gốc đến khi bị cô lập tại nút lá. Đây là chỉ số nền tảng phản ánh mức độ "dễ bị cô lập" của điểm dữ liệu. Điểm bất thường có giá trị $h(x)$ nhỏ hơn đáng kể so với các điểm bình thường.

- $h(x)$ ngắn → dễ bị cô lập → khả năng cao là bất thường.
- $h(x)$ dài → khó bị cô lập → khả năng cao là bình thường.

Độ dài đường đi kỳ vọng : Vì mỗi cây được xây dựng ngẫu nhiên, độ dài đường đi của một điểm có thể khác nhau giữa các cây. Do đó, Isolation Forest sử dụng giá trị kỳ vọng của độ dài đường đi $E(h(x))$, được tính bằng trung bình độ dài đường đi của điểm đó trên toàn bộ các cây trong rừng.

Điểm bất thường (Anomaly Score) :

Từ giá trị $E(h(x))$, Isolation Forest định nghĩa điểm bất thường $s(x)$ như sau:

$$s(x,n) = 2^{-\frac{E(h(x))}{c(n)}}$$

trong đó $c(n)$là độ dài đường đi trung bình của một cây khi dữ liệu có n điểm, được ước lượng dựa trên lý thuyết cây nhị phân tìm kiếm.

Trong đó:

- $E(h(x))$: Là độ dài đường dẫn trung bình của điểm x trên tập hợp các cây $iTrees$.
- $c(n)$: Là yếu tố chuẩn hóa. Đây là độ dài đường dẫn trung bình của một tìm kiếm không thành công trong Cây Nhị Phân Tìm Kiếm (BST) với n nút. Nó

giúp chuẩn hóa điểm số để so sánh được giữa các tập dữ liệu có kích thước khác nhau.

Công thức tính $c(n)$(dựa trên lý thuyết BST):

$$c(n) = 2H(n-1) - \frac{2(n-1)}{n}$$

Trong đó $H(i)$là số điều hòa, có thể ước lượng bằng:

$$H(i) \approx \ln(i) + 0.5772156649$$

(với 0.5772156649 là hằng số Euler).

Ý nghĩa của điểm bất thường:

- Nếu $E(h(x)) \to 0$(đường dẫn rất ngắn), thì $s \to 1$.
  Kết luận**:** Điểm x có khả năng rất cao là bất thường.
- Nếu $E(h(x)) \to n-1$(đường dẫn rất dài), thì $s \to 0$.
  Kết luận**:** Điểm x là điểm bình thường.
- Nếu $E(h(x)) \approx c(n)$(đường dẫn trung bình), thì $s \approx 0.5$.
  Kết luận**:** Không rõ ràng, điểm x không thực sự khác biệt.

Chỉ số này cho phép **xếp hạng dữ liệu theo mức độ bất thường**, từ đó lựa chọn các điểm có điểm số cao nhất làm các bất thường tiềm năng.

2.7. Lý do lựa chọn Isolation Forest

Isolation Forest đặc biệt phù hợp cho bài toán phát hiện và cảnh báo sự cố hệ thống do các đặc điểm sau:

- Không yêu cầu dữ liệu huấn luyện có nhãn
- Hoạt động hiệu quả với dữ liệu đa chiều
- Chi phí tính toán thấp, phù hợp với giám sát liên tục
- Có thể triển khai trong các hệ thống cảnh báo thời gian thực

Trong đề tài này, Isolation Forest được sử dụng để phát hiện các hành vi bất thường trong dữ liệu vận hành hệ thống, từ đó đưa ra cảnh báo sớm nhằm hỗ trợ công tác giám sát và xử lý sự cố.

2.8. Kết chương

Trong chương 2, đồ án đã trình bày tổng quan về trí tuệ nhân tạo và các phương pháp học máy liên quan, đặc biệt tập trung vào học máy không giám sát – hướng tiếp cận phù hợp cho bài toán phát hiện bất thường trong dữ liệu log hệ thống. Chương này đã làm rõ bản chất của bài toán phát hiện sự cố hệ thống, các yêu cầu đặt ra trong môi trường vận hành thực tế, cũng như những thách thức khi dữ liệu sự cố thường khan hiếm và khó gán nhãn.

Bên cạnh đó, thuật toán Isolation Forest đã được phân tích chi tiết về nguyên lý hoạt động, ưu điểm và khả năng ứng dụng trong giám sát hệ thống CNTT. Việc lựa chọn Isolation Forest được chứng minh là phù hợp nhờ khả năng phát hiện bất thường hiệu quả, chi phí tính toán thấp và không phụ thuộc vào dữ liệu huấn luyện có nhãn.

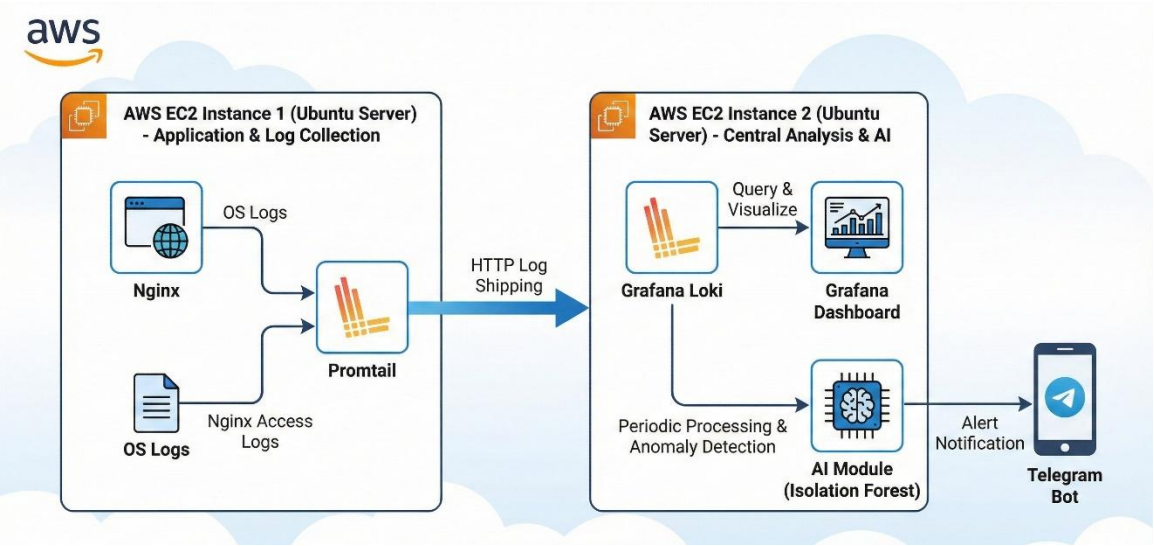
Những nội dung trình bày trong chương 2 đóng vai trò là cơ sở lý thuyết quan trọng, làm nền tảng cho việc triển khai mô hình phát hiện bất thường trên dữ liệu log trong các chương tiếp theo. Trên cơ sở đó, chương 3 sẽ tập trung vào việc hiện thực hóa hệ thống, triển khai thực nghiệm và đánh giá hiệu quả của mô hình trong môi trường giám sát thực tế.

CHƯƠNG 3: TRIỂN KHAI HỆ THỐNG VÀ ĐÁNH GIÁ KẾT QUẢ

3.1 Kiến trúc tổng thể hệ thống

Hệ thống giám sát và phát hiện sự cố hệ thống được thiết kế theo kiến trúc phân lớp nhằm tự động hóa quá trình thu thập log, phân tích và cảnh báo sự cố. Log từ các máy chủ và ứng dụng được thu thập bằng Promtail và gửi về hệ thống lưu trữ tập trung Grafana Loki. Dữ liệu log được trực quan hóa trên Grafana để phục vụ giám sát theo thời gian thực.

Song song với quá trình giám sát, log được tiền xử lý và đưa vào mô hình AI Isolation Forest để phát hiện các hành vi bất thường. Khi phát hiện sự cố, hệ thống tự động gửi cảnh báo đến người quản trị thông qua Telegram Bot, giúp kịp thời xử lý và giảm thiểu ảnh hưởng đến hệ thống.



Hình 3.1 - Sơ đồ kiến trúc tổng thể hệ thống

3.2 Thiết lập môi trường triển khai

Hệ thống được triển khai trên môi trường máy chủ ảo sử dụng nền tảng AWS EC2, nhằm đảm bảo tính linh hoạt, khả năng mở rộng và thuận tiện cho quá trình thử nghiệm. Các máy chủ sử dụng hệ điều hành Ubuntu Server, phù hợp cho việc cài đặt và vận hành các dịch vụ giám sát log và mô hình AI.

Môi trường triển khai được chia thành các máy chủ đảm nhiệm các chức năng khác nhau. Một máy chủ được sử dụng để chạy các dịch vụ ứng dụng và thu thập log, bao gồm Nginx và Promtail. Máy chủ còn lại triển khai các dịch vụ trung tâm gồm Grafana Loki, Grafana Dashboard và module phân tích log bằng AI.

Promtail được cấu hình để thu thập log từ hệ điều hành và log truy cập của Nginx, sau đó gửi log về Loki thông qua giao thức HTTP. Grafana được cấu hình kết nối với Loki làm nguồn dữ liệu (data source) để truy vấn và hiển thị log theo thời gian thực.

Mô hình AI Isolation Forest được huấn luyện và triển khai trực tiếp trên máy chủ phân tích. Hệ thống định kỳ xử lý log mới, phát hiện các hành vi bất thường và kích hoạt cơ chế cảnh báo. Khi phát hiện sự cố, thông tin cảnh báo được gửi tự động đến người quản trị thông qua Telegram Bot.

Việc tách biệt các thành phần triển khai giúp hệ thống hoạt động ổn định, dễ quản lý và thuận tiện cho việc mở rộng hoặc nâng cấp trong tương lai.

3.3 Triển khai hệ thống giám sát

Hệ thống giám sát được triển khai dựa trên các thành phần chính gồm Nginx, Promtail, Grafana Loki và Grafana. Cấu hình chi tiết của từng thành phần được trình bày như sau:

No	VM	Hardware/OS	IP Address		Application	Open Port
			Private	Public		
1	HoangWeb	vCPU-2, Memory-4GiB, SSD-50GiB/ Ubuntu 24.04.3 LTS	172.31.2.178	100.52.175.158	Nginx, Promtail	22, 80, 443, ICMP
2	HoangMon	vCPU-2, Memory-4GiB, SSD-100GiB/ Ubuntu 24.04.3 LTS	172.31.12.134	100.31.144.126	Loki, Grafana	22, 3000,3100, ICMP

Bảng 1 - Thông tin các VM trong mô hình triển khai

3.3.1. Nginx

- Phiên bản: nginx/1.24.0 (Ubuntu)
- Chức năng: Web server, sinh log truy cập HTTP
- Địa chỉ truy cập: http://100.52.175.158/
- Thư mục log: /var/log/nginx

Nginx được sử dụng làm dịch vụ web chính của hệ thống, đồng thời là nguồn sinh log phục vụ cho quá trình giám sát và phân tích hành vi truy cập.

3.3.2. Promtail

- Phiên bản: 3.6.3 (branch: release-3.6.x, revision: 9385bc63)
- Chức năng: Thu thập và phân tích log từ Nginx, gửi log về Loki
- File cấu hình: /etc/promtail/promtail.yaml

```yaml
server:
  http_listen_port: 9080
  grpc_listen_port: 0

positions:
  filename: /var/lib/promtail/positions.yaml

clients:
- url: http://172.31.12.134:3100/loki/api/v1/push

scrape_configs:
- job_name: nginx
  static_configs:
  - targets:
    - localhost
    labels:
      job: nginx
      host: ubuntu-nginx
      __path__: /var/log/nginx/access.log

- job_name: auth
    static_configs:
      - targets:
          - localhost
        labels:
          job: auth
          __path__: /var/log/auth.log
```

Hình 3.2 – Cấu hình Promtail

Promtail được cấu hình để đọc log từ thư mục /var/log/nginx, thực hiện phân tích và gắn nhãn cho các bản ghi log trước khi gửi về hệ thống lưu trữ trung tâm.

3.3.3. Loki

- Phiên bản: 2.9.8 (branch: release-2.9.x, revision: 94e0029)
- Chức năng: Tiếp nhận, lưu trữ và truy vấn log từ Promtail

Grafana Loki đóng vai trò là hệ thống lưu trữ log tập trung, hỗ trợ truy vấn log theo thời gian và theo nhãn, phục vụ cho quá trình giám sát và phân tích.

3.3.4. Grafana

- Phiên bản: 12.3.1
- Địa chỉ truy cập: http://100.31.144.126:3000/
- Nguồn dữ liệu: Grafana Loki
- Dashboard:
    - Panel hiển thị log
    - Biểu đồ phân bố IP truy cập
    - Timeline hoạt động hệ thống

Grafana được sử dụng để trực quan hóa dữ liệu log theo thời gian thực, hỗ trợ người quản trị theo dõi và phân tích tình trạng hoạt động của hệ thống.

3.3.5. Mô tả chi tiết

Trên máy chủ HoangWeb, dịch vụ Nginx được cài đặt để cung cấp ứng dụng web và sinh log truy cập HTTP. Các file log truy cập của Nginx được lưu trữ tại thư mục /var/log/nginx và đóng vai trò là nguồn dữ liệu chính cho hệ thống giám sát.

Promtail được cài đặt cùng máy chủ HoangWeb và cấu hình để thu thập log từ Nginx. Promtail đọc các file log, thực hiện phân tích và gắn nhãn cho từng bản ghi log như địa chỉ IP nguồn, URI truy cập, mã trạng thái HTTP và thời gian xảy ra sự kiện. Sau đó, log được gửi về máy chủ giám sát thông qua giao thức HTTP.

Máy chủ HoangMon triển khai Grafana Loki để tiếp nhận và lưu trữ log tập trung. Loki lưu trữ log theo mô hình nhãn, cho phép truy vấn nhanh các bản ghi log theo thời gian và theo từng thuộc tính cụ thể. Cách tiếp cận này giúp tối ưu tài nguyên lưu trữ và phù hợp với bài toán giám sát log quy mô lớn.

Grafana được cài đặt trên cùng máy chủ HoangMon và cấu hình kết nối với Loki làm nguồn dữ liệu. Các dashboard giám sát được xây dựng nhằm hiển thị tổng số request theo thời gian, phân bố IP truy cập, các mã trạng thái HTTP và các URI bị truy cập nhiều nhất. Thông qua các dashboard này, người quản trị có thể theo dõi tình trạng hoạt động của hệ thống và phát hiện sớm các dấu hiệu bất thường ở tầng HTTP.

Kết quả triển khai cho thấy hệ thống giám sát hoạt động ổn định, thu thập và hiển thị log liên tục theo thời gian thực, tạo nền tảng dữ liệu đầu vào cho việc phân tích và phát hiện bất thường bằng mô hình AI ở các bước tiếp theo.

3.4 Kịch bản test DDoS

Nhằm kiểm chứng khả năng giám sát và phát hiện các hành vi bất thường ở tầng HTTP, đề tài xây dựng các kịch bản test mô phỏng tấn công DDoS (Layer 7) dưới dạng HTTP flood/scan/burst traffic. Các kịch bản được thực hiện bằng PowerShell từ máy client, gửi request đến web server Nginx tại http://100.52.175.158/. Trong quá trình test, hệ thống giám sát (Promtail → Loki → Grafana) ghi nhận log theo thời gian thực và hỗ trợ truy vấn/quan sát các chỉ số như tổng số request, top IP truy cập, mã trạng thái (đặc biệt 404) và URI bị truy cập nhiều nhất.

Test case 1 – HTTP Flood vào URI không tồn tại (scan/flood)

Mục tiêu của kịch bản này là tạo lưu lượng lớn đến một endpoint không tồn tại nhằm mô phỏng hành vi vừa dò quét vừa gây tải. Hệ thống kỳ vọng ghi nhận số lượng request tăng đột biến và tỷ lệ phản hồi 404 tăng rõ rệt.

Hình 3.3 - Dashboard giám sát lưu lượng trong kịch bản HTTP Flood vào URI không tồn tại

Kết quả từ Hình 3.4 cho thấy tổng số HTTP request tăng đột biến trong khoảng thời gian ngắn khi kịch bản flood vào URI không tồn tại được kích hoạt. Đồng thời, số lượng phản hồi mã lỗi 404 chiếm tỷ lệ lớn trong tổng số request. Điều này phản ánh đúng đặc điểm của hành vi vừa dò quét tài nguyên vừa tạo tải lên hệ thống. Dashboard giám sát thể hiện rõ IP nguồn phát sinh lưu lượng bất thường và các URI không hợp lệ bị truy cập liên tục, cho thấy hệ thống có khả năng phát hiện và trực quan hóa các dấu hiệu tấn công dạng scan/flood.

Test case 2 – Flood liên tục vào trang chính (flood kéo dài)

Kịch bản mô phỏng tấn công flood đơn giản bằng cách gửi request liên tục đến URI hợp lệ trong thời gian dài (dừng bằng Ctrl+C). Mục tiêu là tạo lưu lượng bền vững để quan sát khả năng giám sát theo thời gian thực và xu hướng tăng liên tục của tổng request.

Hình 3.4 - Dashboard giám sát lưu lượng trong kịch bản flood liên tục vào trang chính

Trong kịch bản flood kéo dài vào trang chính, biểu đồ tổng số HTTP request theo thời gian cho thấy lưu lượng tăng đều và duy trì ở mức cao trong suốt quá trình thử nghiệm. Không xuất hiện các đỉnh lưu lượng đột ngột, phù hợp với đặc trưng của tấn công flood bền vững. Kết quả này chứng minh hệ thống giám sát có khả năng theo dõi tải liên tục theo thời gian thực và phản ánh chính xác xu hướng gia tăng của lưu lượng truy cập hợp lệ vào một endpoint cụ thể.

Test case 3 – URL Scanning / Reconnaissance (dò quét nhiều đường dẫn)

Kịch bản mô phỏng hành vi trinh sát (recon) bằng cách truy cập nhiều URL khác nhau (đa phần không tồn tại). Kỳ vọng hệ thống thể hiện rõ nhóm IP có nhiều request 404 và các URI bị dò quét.

Hình 3.5 - Dashboard thể hiện hành vi dò quét URL (URL Scanning/Reconnaissance)

Hình 3.6 thể hiện rõ sự gia tăng số lượng request dẫn đến mã lỗi 404, đồng thời nhiều URI khác nhau bị truy cập trong cùng một khoảng thời gian. Các IP nguồn có tần suất request 404 cao được làm nổi bật trên dashboard, phản ánh hành vi trinh sát và dò quét tài nguyên của kẻ tấn công. Kết quả cho thấy hệ thống có khả năng phân biệt các hành vi truy cập bất thường thông qua việc phân tích mã phản hồi HTTP và danh sách các URI bị truy cập.

Test case 4 – Burst traffic (gửi song song, mô phỏng botnet nhỏ)

Kịch bản tạo các request song song trong thời gian rất ngắn để mô phỏng burst traffic dạng bot. Mục tiêu là tạo các "đỉnh" lưu lượng ngắn, từ đó quan sát mức độ đột biến trên biểu đồ theo thời gian và top IP/URI tương ứng.

Hình 3.6 - Dashboard giám sát burst traffic mô phỏng tấn công gửi song song

Trong kịch bản burst traffic, biểu đồ tổng số request xuất hiện các đỉnh lưu lượng cao trong thời gian rất ngắn, sau đó nhanh chóng giảm xuống. Đây là đặc trưng của các đợt tấn công gửi song song nhằm tạo tải đột biến trong thời gian ngắn. Dashboard giám sát phản ánh rõ các đỉnh lưu lượng này, đồng thời xác định được các IP và URI liên quan tại từng thời điểm. Kết quả cho thấy hệ thống có độ nhạy đủ cao để phát hiện các đợt tấn công ngắn nhưng có cường độ lớn.

Test case 5 – Targeted URI Flood (flood tập trung vào một endpoint)

Kịch bản mô phỏng tấn công tập trung vào một URI cụ thể (ví dụ /login), thường gặp trong tấn công vào các chức năng nhạy cảm. Kỳ vọng dashboard thể hiện URI bị tập trung truy cập và tổng số request tăng mạnh trong khoảng thời gian ngắn.

Hình 3.7 - Dashboard giám sát tấn công flood tập trung vào một URI cụ thể

Kết quả từ Hình 3.8 cho thấy một URI cụ thể bị tập trung truy cập với số lượng request vượt trội so với các URI khác. Tổng lưu lượng tăng mạnh trong khoảng thời gian ngắn và dashboard thể hiện rõ endpoint bị nhắm mục tiêu. Điều này phù hợp với kịch bản tấn công vào các chức năng nhạy cảm của hệ thống. Qua đó có thể nhận thấy hệ thống giám sát hỗ trợ hiệu quả trong việc xác định tài nguyên đang bị tấn công.

Dashboard như trong hình dùng để giám sát và phát hiện các hành vi bất thường ở tầng HTTP.

- Panel đầu tiên cho thấy tổng số request theo thời gian. Khi có DDoS hoặc flood, panel này sẽ tăng đột biến.
- Panel thứ hai cho biết IP nào đang gửi nhiều request nhất, giúp xác định nguồn tấn công.
- Panel thứ ba tập trung vào các request 404, thường xuất hiện khi attacker dò quét URL hoặc endpoint.
- Panel cuối cùng cho biết endpoint nào đang bị truy cập nhiều nhất, giúp xác định mục tiêu bị tấn công.

Qua các kịch bản test, có thể thấy mỗi loại hành vi sẽ kích hoạt các panel khác nhau. Việc kết hợp cả 4 panel giúp phân biệt rõ giữa truy cập bình thường, scan và DDoS.

Kết luận :

Hệ thống giám sát được triển khai dựa trên nginx – promtail – Loki – Grafana đã đáp ứng tốt mục tiêu thu thập, phân tích và trực quan hóa log truy cập HTTP trong môi trường lab.

Thông qua bốn panel chính (P1–P4), hệ thống có khả năng:

- Giám sát lưu lượng truy cập theo thời gian thực.

- Phát hiện các hành vi bất thường như HTTP flood, burst traffic và URL scanning.
- Xác định nguồn truy cập đáng nghi (theo IP) và mục tiêu bị tập trung tấn công (theo URI).
- Phân biệt giữa các kiểu hành vi khác nhau ngay cả khi các request đều trả về trạng thái 200 OK.
- Các kịch bản test cho thấy hệ thống phát hiện dựa trên hành vi, không phụ thuộc vào lỗi ứng dụng, phù hợp với các tình huống DDoS ở tầng ứng dụng (Layer 7).
- Dashboard được xây dựng đơn giản, dễ quan sát.

Nhìn chung, hệ thống đã rõ cách sinh log, cách phân tích log và cách phát hiện hành vi bất thường ở tầng HTTP bằng công cụ mã nguồn mở.

3.5 Chuẩn bị dữ liệu

3.5.1. Nguồn dữ liệu

Nguồn dữ liệu sử dụng trong đề tài bao gồm log truy cập web server (Nginx và Apache) và log hệ điều hành Linux. Log Nginx được thu thập trực tiếp từ hệ thống triển khai thực nghiệm, trong khi log Apache và log Linux được lấy từ bộ dữ liệu log công khai phục vụ nghiên cứu.

Các file log web server chứa thông tin về địa chỉ IP, thời gian truy cập, URI và mã trạng thái HTTP, phục vụ phân tích hành vi truy cập và phát hiện bất thường ở tầng ứng dụng. Log hệ điều hành Linux (như syslog, auth.log) phản ánh các sự kiện ở mức hệ thống, giúp mở rộng phạm vi giám sát và hỗ trợ phát hiện sự cố tổng thể.

```
2026-01-16T10:15:30 ip-1 sshd[12345]: Accepted publickey for ubuntu from 10.0.0.5 port 55231 ssh2
2026-01-16T10:20:01 ip-1 CRON[12346]: pam_unix(cron:session): session opened for user root(uid=0)
204.76.203.219 - - [16/Jan/2026:10:30:23 +0700] "GET / HTTP/1.1" 200 409 "-" "Mozilla/5.0"
2026-01-16T11:01:12 ip-1 sshd[23456]: Invalid user admin from 185.23.45.67 port 58921
2026-01-16T11:01:15 ip-1 sshd[23456]: Failed password for root from 185.23.45.67 port 58921 ssh2
2026-01-16T11:01:18 ip-1 sshd[23456]: Failed password for root from 185.23.45.67 port 58921 ssh2
2026-01-16T11:01:21 ip-1 sshd[23456]: Failed password for root from 185.23.45.67 port 58921 ssh2
213.209.159.181 - - [16/Jan/2026:11:05:12 +0700] "GET /.git/config HTTP/1.1" 404 134 "-" "Mozilla/5.0"
91.232.238.112 - - [16/Jan/2026:11:07:12 +0700] "GET /admin/config.php HTTP/1.0" 404 162 "-" "xfa1"
2026-01-16T11:10:22 ip-1 sudo: ubuntu : TTY=pts/0 ; PWD=/home/ubuntu ; USER=root ; COMMAND=/bin/bash
```

Hình 3.8 - Dữ liệu file log hệ thống

Do dữ liệu thực tế không có nhãn sẵn, nghiên cứu áp dụng phương pháp **weak supervision**, trong đó các dòng log được gán nhãn bất thường dựa trên các luật heuristics, ví dụ:

- Dòng log chứa “Failed password” hoặc “Invalid user” được coi là bất thường.
- Truy cập các đường dẫn như /admin, /.git, /config.php được coi là hành vi đáng nghi.
- Sử dụng quyền root hoặc sudo được xem là hành vi có rủi ro cao.

3.5.2. Tiền xử lý và trích xuất đặc trưng

Mỗi dòng log được chuyển đổi thành một vector đặc trưng nhị phân, bao gồm các thuộc tính:

- is_failed_login: đăng nhập thất bại.
- is_success_login: đăng nhập thành công.
- is_root_access: truy cập quyền root hoặc sử dụng sudo.
- is_admin_path: truy cập đường dẫn quản trị.
- is_web: truy cập web.
- is_cron: thực thi tác vụ cron.

Các đặc trưng này phản ánh các hành vi bảo mật và vận hành phổ biến trong hệ thống, đồng thời giúp mô hình phân biệt giữa hoạt động bình thường và bất thường.

3.6 Huấn luyện mô hình

3.6.1. Mô hình sử dụng

Nghiên cứu sử dụng thuật toán Isolation Forest, một phương pháp học máy không giám sát chuyên dùng cho phát hiện bất thường. Mô hình được huấn luyện với các tham số chính:

- Số cây (n_estimators): 200,
- Tỷ lệ dữ liệu bất thường giả định (contamination): 5%,
- Trạng thái ngẫu nhiên (random_state): 42.

Dữ liệu được chia thành tập huấn luyện (80%) và tập kiểm tra (20%) để phục vụ đánh giá mô hình.

3.6.2. Quy trình huấn luyện

Quy trình huấn luyện gồm các bước:

- Đọc và tiền xử lý log hệ thống,
- Trích xuất vector đặc trưng cho từng dòng log,
- Huấn luyện mô hình Isolation Forest trên tập huấn luyện,
- Lưu mô hình và cấu trúc đặc trưng để sử dụng trong giai đoạn phát hiện.

3.6.3. Phương pháp đánh giá

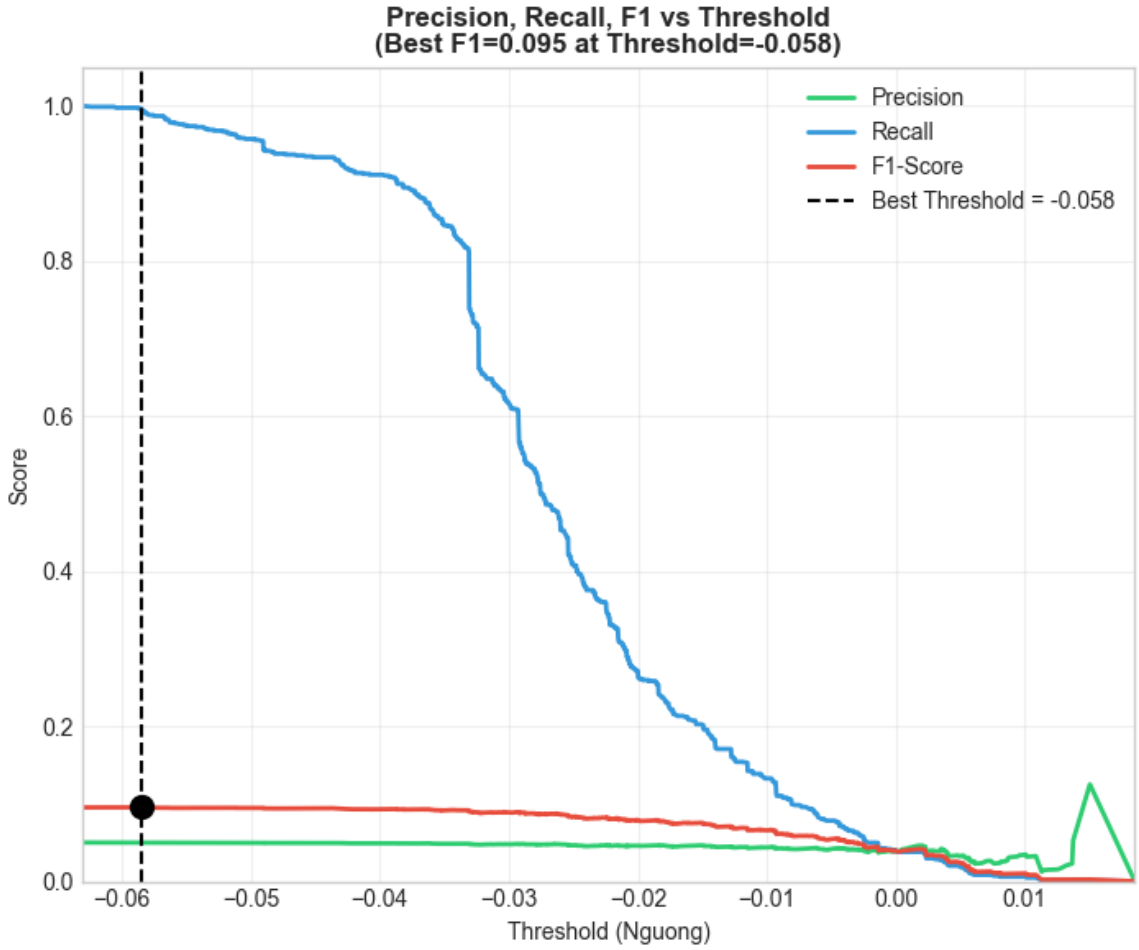
3.6.3.1 Đánh giá định lượng

Do dữ liệu không có nhãn chuẩn, nghiên cứu sử dụng nhãn giả lập (weak labels) dựa trên các luật bảo mật để đánh giá mô hình. Các chỉ số đánh giá bao gồm:

- Precision: tỷ lệ cảnh báo đúng trên tổng số cảnh báo,
- Recall: tỷ lệ phát hiện đúng bất thường trên tổng số bất thường thực tế,
- F1-score: trung bình điều hòa giữa Precision và Recall.

3.6.3.2 Đường cong Precision-Recall

Để trực quan hóa mối quan hệ đánh đổi giữa Precision và Recall khi thay đổi ngưỡng phát hiện, đề tài xây dựng đường cong Precision – Recall theo threshold, như minh họa trong Hình 3.10.



Hình 3.9 – Biểu đồ đường cong Precision-Recall

Biểu đồ cho phép quan sát:

- Sự thay đổi của Precision, Recall và F1-score khi điều chỉnh ngưỡng.
- Điểm ngưỡng tối ưu tương ứng với giá trị F1-score lớn nhất, được sử dụng làm cơ sở lựa chọn threshold cho hệ thống phát hiện sự cố trong thực tế..

Việc phân tích đường cong cho phép lựa chọn ngưỡng phù hợp với yêu cầu vận hành cụ thể, ví dụ:

- Ưu tiên Recall cao trong các hệ thống yêu cầu không bỏ sót sự cố nghiêm trọng,
- Ưu tiên Precision cao trong các hệ thống yêu cầu cảnh báo chính xác nhằm tránh quá tải cảnh báo.

3.6.4. Kết quả thực nghiệm

3.6.4.1 Kết quả đánh giá

Kết quả thực nghiệm cho thấy mô hình Isolation Forest đạt giá trị F1-score cao nhất tại ngưỡng threshold ≈ –0.058, với F1-score xấp xỉ 0.095. Tại ngưỡng này, mô hình đạt được sự cân bằng tương đối giữa khả năng phát hiện bất thường (Recall) và độ chính xác của các cảnh báo (Precision).

Ở các giá trị threshold thấp hơn, Recall đạt mức rất cao do mô hình gán nhiều điểm dữ liệu là bất thường, tuy nhiên Precision giảm mạnh do xuất hiện nhiều cảnh báo giả. Ngược lại, khi threshold tăng dần, Precision có xu hướng cải thiện nhưng Recall suy giảm đáng kể, dẫn đến việc bỏ sót nhiều sự cố thực tế.

3.6.4.2 Phân tích đánh đổi Precision-Recall

Phân tích đường cong Precision–Recall cho thấy:

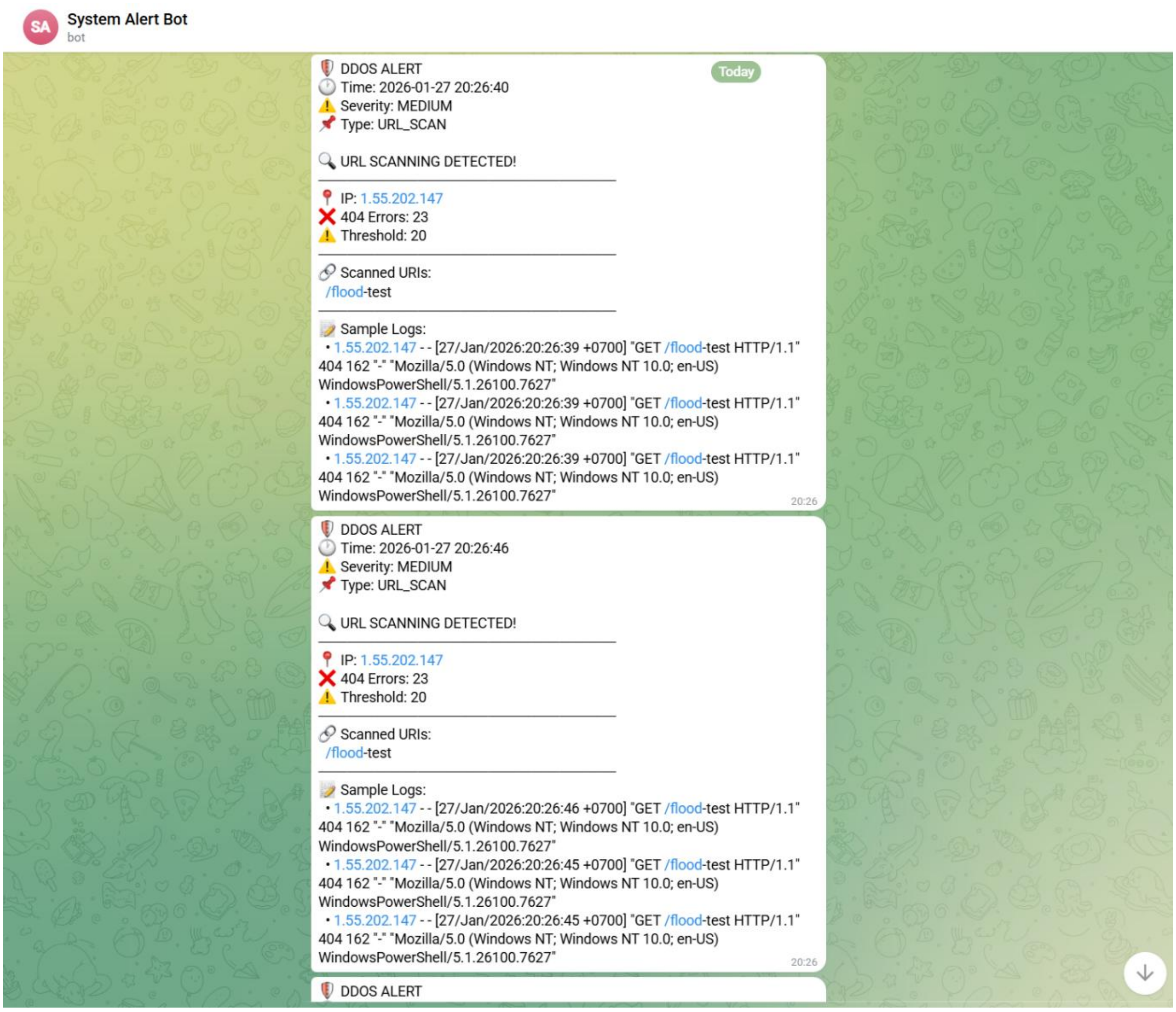
- Khi giảm ngưỡng, Recall tăng lên đáng kể, giúp phát hiện nhiều bất thường hơn nhưng làm giảm Precision do tăng số cảnh báo sai.
- Khi tăng ngưỡng, Precision được cải thiện, phù hợp với các hệ thống yêu cầu cảnh báo chính xác, tuy nhiên một số bất thường có thể bị bỏ sót.

Điều này chứng minh khả năng linh hoạt của mô hình trong việc thích ứng với các mục tiêu vận hành khác nhau.

3.6.5. Triển khai hệ thống phát hiện và cảnh báo thời gian thực

Mô hình sau khi huấn luyện được triển khai trong hệ thống giám sát thời gian thực, với cơ chế:

- Quét file log định kỳ mỗi 10 giây,
- Phát hiện các dòng log có điểm bất thường vượt ngưỡng,
- Ghi nhận cảnh báo vào file log nội bộ,
- Gửi thông báo tức thời qua nền tảng Telegram cho quản trị viên hệ thống.



Hình 3.10 - Cảnh báo tự động qua Telegram

Việc tích hợp cơ chế cảnh báo tự động giúp nâng cao khả năng phản ứng sự cố và giảm thời gian phát hiện các hành vi nguy hiểm trong hệ thống.

3.6.6. Sinh dữ liệu kiểm thử và đánh giá thực nghiệm

Để kiểm chứng hệ thống, nghiên cứu xây dựng tập log kiểm thử mô phỏng các kịch bản tấn công phổ biến như:

- Tấn công brute-force SSH,
- Truy cập tài nguyên quản trị trái phép,
- Thực thi lệnh với quyền root.

Kết quả cho thấy hệ thống có khả năng phát hiện hiệu quả các hành vi bất thường này và phát sinh cảnh báo kịp thời, chứng minh tính khả thi của mô hình trong môi trường thực tế.

3.7 Thu thập và xử lý log

3.7.1. Tổng quan

Hệ thống được xây dựng theo mô hình log pipeline tập trung, trong đó log phát sinh từ web server được thu thập, xử lý, lưu trữ và phân tích theo thời gian thực nhằm phục vụ giám sát và phát hiện tấn công.

Luồng xử lý log như sau: Nginx → Promtail → Loki → Grafana

3.7.2. Thu thập log tại Web Server (Nginx)

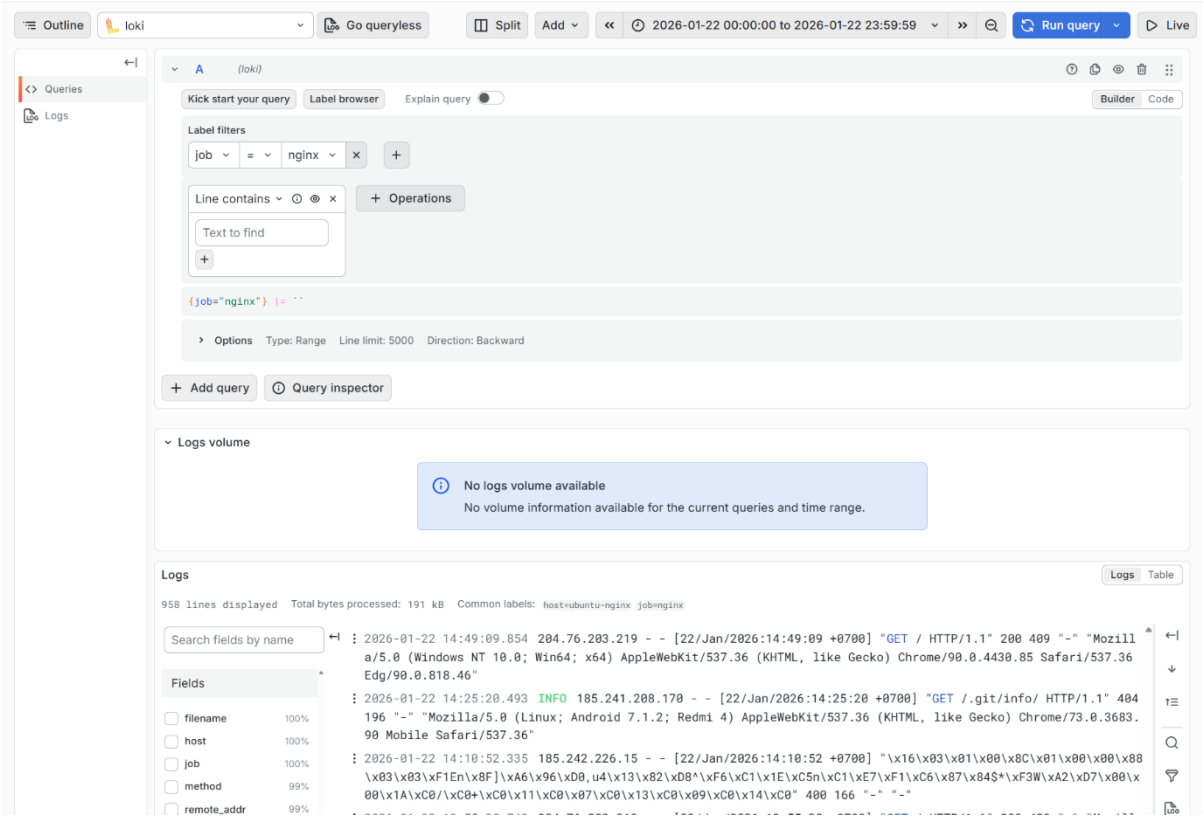
Trong hệ thống được triển khai, Web Server Nginx đóng vai trò là thành phần cung cấp dịch vụ web và đồng thời là nguồn sinh log chính phục vụ cho quá trình giám sát và phát hiện sự cố. Nginx tự động ghi lại toàn bộ các yêu cầu HTTP của client thông qua các file log truy cập (access log) và log lỗi (error log), bao gồm các thông tin quan trọng như địa chỉ IP nguồn, thời gian truy cập, phương thức HTTP, URI được yêu cầu và mã trạng thái phản hồi. Các file log này được lưu trữ mặc định tại thư mục /var/log/nginx trên máy chủ Web Server. Việc thu thập log tại tầng web cho phép hệ thống ghi nhận đầy đủ hành vi truy cập của người dùng, từ đó tạo cơ sở dữ liệu đầu vào quan trọng cho việc phân tích lưu lượng, phát hiện bất thường và nhận diện các hành vi tấn công ở tầng ứng dụng HTTP.

3.7.3. Thu thập và xử lý log bằng Promtail

Promtail được triển khai trên cùng máy chủ với Nginx và hoạt động như một agent thu thập log. Promtail được cấu hình để theo dõi các file log của Nginx trong thư mục /var/log/nginx, đọc các bản ghi mới phát sinh theo thời gian thực và thực hiện tiền xử lý trước khi gửi về hệ thống lưu trữ tập trung. Trong quá trình xử lý, Promtail thực hiện phân tích cấu trúc log, trích xuất các trường thông tin quan trọng và gắn nhãn (labels) cho từng bản ghi, bao gồm địa chỉ IP nguồn, URI truy cập, mã trạng thái HTTP và loại log. Việc gắn nhãn giúp tối ưu khả năng truy vấn, phân loại và phân tích log ở các bước tiếp theo. Sau khi xử lý, log được Promtail gửi về Grafana Loki thông qua giao thức HTTP, đảm bảo dữ liệu log được truyền liên tục và đồng bộ về hệ thống giám sát trung tâm.

3.7.4. Lưu trữ và truy vấn log tại Loki

Grafana Loki được sử dụng làm hệ thống lưu trữ log tập trung cho toàn bộ dữ liệu thu thập từ các máy chủ trong hệ thống. Loki tiếp nhận log từ Promtail và lưu trữ theo mô hình dựa trên nhãn, thay vì lập chỉ mục toàn bộ nội dung log như các hệ thống logging truyền thống. Cách tiếp cận này giúp giảm đáng kể chi phí lưu trữ và tài nguyên xử lý, đồng thời vẫn đảm bảo khả năng truy vấn hiệu quả theo thời gian và theo các thuộc tính quan trọng. Thông qua ngôn ngữ truy vấn LogQL, Loki cho phép truy vấn linh hoạt các bản ghi log phục vụ cho việc giám sát, phân tích hành vi truy cập và làm dữ liệu đầu vào cho mô hình phát hiện bất thường. Việc sử dụng Loki giúp hệ thống đảm bảo khả năng mở rộng, đáp ứng yêu cầu lưu trữ log lớn và hỗ trợ tốt cho quá trình giám sát và phân tích log theo thời gian thực.



Hình 3.11 - Minh họa truy vấn và hiển thị log bằng Loki

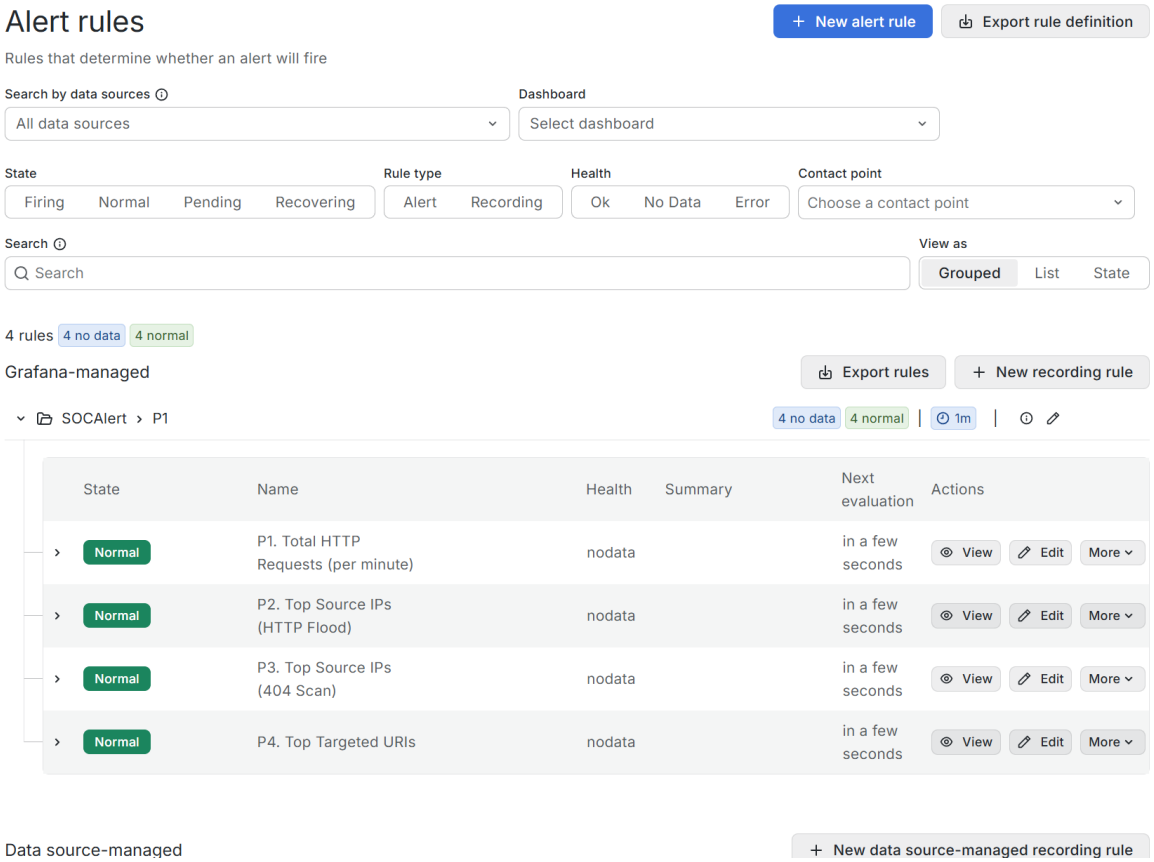
3.8 Phân tích, hiển thị, và cảnh báo bằng Grafana

Grafana được sử dụng làm nền tảng trung tâm để phân tích, trực quan hóa và cảnh báo dữ liệu log trong hệ thống giám sát. Thông qua việc kết nối Grafana với Grafana Loki làm nguồn dữ liệu, hệ thống cho phép truy vấn và hiển thị log theo thời gian thực dựa trên các tiêu chí như khoảng thời gian, địa chỉ IP nguồn, URI truy cập và mã trạng thái HTTP. Các dashboard được xây dựng nhằm cung cấp cái nhìn tổng quan về trạng thái hoạt động của hệ thống, bao gồm tổng số request theo thời gian, phân bố IP truy cập, các URI bị truy cập nhiều nhất và tỷ lệ phản hồi theo từng mã trạng thái.

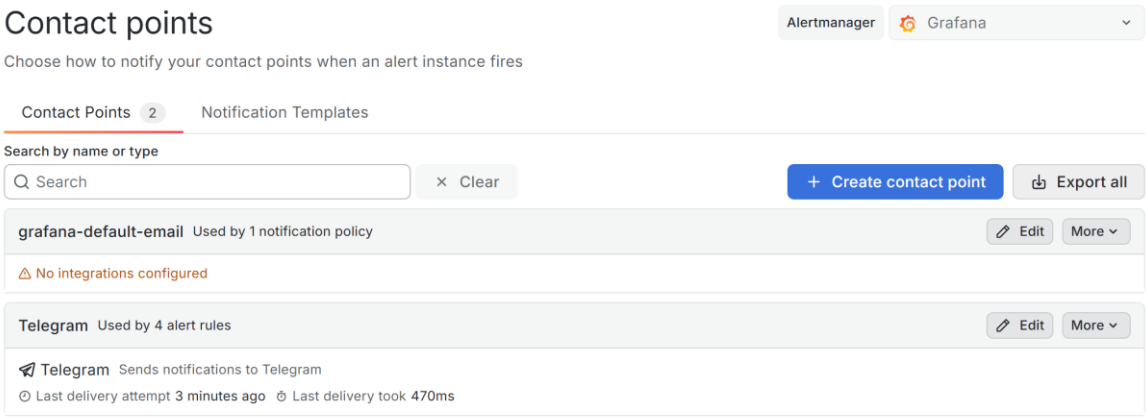
Bên cạnh chức năng hiển thị, Grafana còn được sử dụng để hỗ trợ phân tích hành vi truy cập và phát hiện sớm các dấu hiệu bất thường. Thông qua việc quan sát các biểu đồ và truy vấn LogQL, người quản trị có thể nhận diện các hiện tượng như lưu lượng tăng đột biến, tỷ lệ lỗi HTTP (đặc biệt là mã 404) gia tăng bất thường hoặc một URI bị truy cập tập trung trong thời gian ngắn. Những thông tin này đóng vai trò quan trọng trong việc đánh giá tình trạng vận hành của hệ thống và hỗ trợ quá trình điều tra nguyên nhân sự cố.

Ngoài ra, hệ thống cảnh báo tự động được cấu hình trực tiếp trên Grafana thông qua các Alert Rules. Khi các điều kiện bất thường được xác định, chẳng hạn như số lượng request vượt ngưỡng cho phép hoặc mô hình AI phát hiện hành vi bất thường, Grafana sẽ kích hoạt cảnh báo và gửi thông báo đến người quản trị thông qua các

Contact Points đã cấu hình, cụ thể là Telegram Bot. Cơ chế cảnh báo này giúp rút ngắn thời gian phát hiện và phản ứng sự cố, nâng cao hiệu quả giám sát và đảm bảo hệ thống được vận hành ổn định, an toàn trong môi trường thực tế.



Hình 3.12 - Giao diện cấu hình và quản lý các Alert Rules trên Grafana



Hình 3.13 - Giao diện cấu hình các Contact points trên Grafana

3.9 Kết chương

Chương 3 đã trình bày chi tiết quá trình triển khai hệ thống giám sát và phát hiện sự cố hệ thống tự động thông qua phân tích log, từ thiết kế kiến trúc tổng thể, thiết lập môi trường triển khai đến xây dựng các kịch bản kiểm thử và đánh giá kết quả thực nghiệm. Hệ thống được triển khai dựa trên các công cụ mã nguồn mở gồm Nginx, Promtail, Grafana Loki và Grafana, kết hợp với mô hình học máy Isolation Forest để phát hiện bất thường trong dữ liệu log.

Thông qua các kịch bản mô phỏng tấn công và hành vi bất thường, kết quả cho thấy hệ thống có khả năng thu thập, lưu trữ và trực quan hóa log theo thời gian thực, đồng thời phát hiện hiệu quả các dấu hiệu bất thường trong quá trình vận hành. Việc tích hợp mô hình AI giúp hệ thống vượt qua hạn chế của phương pháp giám sát dựa trên luật cố định, cho phép nhận diện các hành vi bất thường phức tạp và chưa từng xuất hiện trước đó.

Ngoài ra, cơ chế cảnh báo tự động thông qua Telegram Bot giúp người quản trị nhanh chóng nhận được thông tin sự cố và kịp thời đưa ra biện pháp xử lý, góp phần nâng cao khả năng phản ứng và giảm thiểu rủi ro cho hệ thống. Những kết quả đạt được trong chương này đã chứng minh tính khả thi và hiệu quả của giải pháp đề xuất.

Trên cơ sở các kết quả triển khai và đánh giá, chương 4 sẽ tổng kết những đóng góp chính của đồ án, chỉ ra các hạn chế còn tồn tại và đề xuất các hướng phát triển nhằm hoàn thiện và mở rộng hệ thống trong tương lai.

CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1. Kết quả đạt được

Trong phạm vi đồ án, đề tài "Xây dựng hệ thống giám sát và phát hiện sự cố hệ thống tự động thông qua phân tích log" đã hoàn thành các mục tiêu đề ra ban đầu. Hệ thống giám sát được xây dựng theo kiến trúc tập trung, cho phép thu thập, lưu trữ và hiển thị log hệ thống theo thời gian thực thông qua bộ công cụ Grafana – Loki – Promtail. Dữ liệu log từ Web Server được xử lý hiệu quả, hỗ trợ người quản trị theo dõi tình trạng hoạt động của hệ thống và nhanh chóng nhận diện các dấu hiệu bất thường ở tầng ứng dụng HTTP.

Bên cạnh đó, đồ án đã ứng dụng thành công mô hình học máy không giám sát Isolation Forest để phân tích dữ liệu log và phát hiện các hành vi bất thường mà không cần dữ liệu huấn luyện có nhãn. Kết quả thực nghiệm cho thấy mô hình có khả năng nhận diện các kịch bản tấn công và sự cố phổ biến như HTTP flood, dò quét URL và burst traffic, đồng thời cho phép lựa chọn ngưỡng phát hiện phù hợp thông qua phân tích các chỉ số Precision, Recall và F1-score. Việc tích hợp cơ chế cảnh báo tự động qua Telegram giúp rút ngắn đáng kể thời gian phát hiện và phản ứng sự cố so với phương pháp giám sát thủ công.

Tổng thể, hệ thống được triển khai hoạt động ổn định trong môi trường thử nghiệm, đáp ứng yêu cầu giám sát liên tục, phát hiện sớm sự cố và hỗ trợ người quản trị trong quá trình vận hành hệ thống. Kết quả của đồ án không chỉ có giá trị thực tiễn trong giám sát và bảo đảm an toàn hệ thống CNTT, mà còn là nền tảng để tiếp tục nghiên cứu, mở rộng và phát triển các giải pháp giám sát thông minh trong tương lai.

Mặc dù đã đạt được các mục tiêu đề ra, đề tài vẫn còn tồn tại một số hạn chế nhất định. Thứ nhất, hệ thống được triển khai và đánh giá chủ yếu trong môi trường thử nghiệm với số lượng máy chủ và kịch bản sự cố còn hạn chế, do đó chưa phản ánh đầy đủ tính phức tạp và đa dạng của các hệ thống CNTT quy mô lớn trong thực tế. Các kịch bản kiểm thử tập trung chủ yếu vào các hành vi bất thường ở tầng HTTP, trong khi chưa xem xét đầy đủ các dạng sự cố khác như lỗi ứng dụng nội bộ, sự cố tài nguyên hệ thống hoặc các tấn công phức tạp ở nhiều tầng.

Thứ hai, mô hình Isolation Forest được sử dụng trong đề tài là mô hình học máy không giám sát, phụ thuộc nhiều vào chất lượng dữ liệu log đầu vào và các tham số cấu hình. Trong một số trường hợp, mô hình có thể phát sinh cảnh báo giả hoặc bỏ sót các bất thường có đặc điểm gần với hành vi vận hành bình thường. Việc lựa chọn ngưỡng phát hiện hiện tại vẫn mang tính thực nghiệm, chưa có cơ chế tự động điều chỉnh theo sự thay đổi của đặc điểm dữ liệu theo thời gian.

Ngoài ra, hệ thống chưa được tối ưu hoàn toàn về hiệu năng và khả năng mở rộng khi khối lượng log tăng lớn hoặc khi triển khai trên nhiều máy chủ đồng thời. Việc phân

tích log và phát hiện bất thường hiện được thực hiện theo chu kỳ, chưa đáp ứng hoàn toàn yêu cầu xử lý thời gian thực ở quy mô lớn. Bên cạnh đó, cơ chế cảnh báo hiện tại mới dừng ở mức gửi thông báo đơn lẻ, chưa hỗ trợ phân loại mức độ nghiêm trọng của sự cố hoặc tổng hợp cảnh báo để giảm tải cho người quản trị.

Những hạn chế trên là cơ sở quan trọng để đề tài tiếp tục được cải tiến và mở rộng trong các nghiên cứu và triển khai tiếp theo.

4.2. Hướng phát triển

Trong tương lai, hệ thống có thể được mở rộng và hoàn thiện theo các hướng sau:

- **Mở rộng nguồn dữ liệu:** Tích hợp thêm nhiều loại log khác nhau như log mạng, log ứng dụng web, log cơ sở dữ liệu, log bảo mật (IDS/IPS, firewall) nhằm nâng cao khả năng phát hiện sự cố toàn diện.
- **Cải tiến trích xuất đặc trưng:** Áp dụng các phương pháp xử lý ngôn ngữ tự nhiên (NLP), embedding hoặc mô hình ngôn ngữ lớn để biểu diễn log dưới dạng vector giàu ngữ nghĩa hơn.
- **So sánh và kết hợp mô hình:** Thử nghiệm và so sánh Isolation Forest với các mô hình khác như Autoencoder, LSTM, One-Class SVM hoặc kết hợp nhiều mô hình (ensemble) để cải thiện hiệu năng.
- **Học thích nghi và cập nhật mô hình:** Xây dựng cơ chế học trực tuyến hoặc bán giám sát, cho phép mô hình tự cập nhật khi hệ thống thay đổi hành vi theo thời gian.
- **Phân tích nguyên nhân và giải thích mô hình:** Tích hợp các kỹ thuật giải thích (Explainable AI) để làm rõ nguyên nhân dẫn đến cảnh báo, hỗ trợ quản trị viên trong việc xử lý sự cố.
- **Triển khai thực tế và tối ưu hiệu năng:** Đưa hệ thống vào môi trường sản xuất, tối ưu tốc độ xử lý, khả năng mở rộng và tích hợp với các nền tảng giám sát hiện có như SIEM hoặc SOC.
- **Phát triển giao diện người dùng:** Xây dựng dashboard trực quan để hiển thị log, cảnh báo, biểu đồ đánh giá và lịch sử sự cố, giúp người dùng theo dõi hệ thống một cách thuận tiện.

Tài liệu tham khảo

[1] P. F. Team, “What is Grafana?,” [Trực tuyến]. Available: https://pandorafms.com/blog/what-is-grafana-used-for/.

[2] G. Labs, “Grafana Loki Documentation,” [Trực tuyến]. Available: https://grafana.com/docs/loki/latest/.

[3] G. Labs, “Promtail Documentation,” [Trực tuyến]. Available: https://grafana.com/docs/loki/latest/send-data/promtail/.

[4] A. W. Services, “Điện toán đám mây với AWS,” [Trực tuyến]. Available: https://aws.amazon.com/vi/what-is-aws/.

[5] Nginx, “Nginx Documentation,” [Trực tuyến]. Available: https://nginx.org/en/docs/.

[6] N. M. Hải, “Trí tuệ nhân tạo AI là gì? Khám phá lợi ích và thách thức,” [Trực tuyến]. Available: https://vnptai.io/vi/blog/detail/tri-tue-nhan-tao-ai-la-gi.

[7] B. C. Học, “Machine learning (Máy học) là gì? Phân loại và ứng dụng Machine Learning,” [Trực tuyến]. Available: https://cloudgo.vn/machine-learning-la-gi.

[8] F. T. K. M. T. a. Z.-H. Z. Liu, “Isolation Forest,” [Trực tuyến]. Available: https://www.researchgate.net/profile/Fei-Tony-Liu/publication/224384174_Isolation_Forest/links/5bbd5c0ca6fdcc9552dd04f0/Isolation-Forest.pdf.

[9] M. Biradar, “Grafana, Loki, and Promtail for Visualization on AWS EC2 Instance,” [Trực tuyến]. Available: https://medium.com/@maheshbiradar8887/grafana-loki-and-promtail-for-visualization-on-aws-ec2-instance-dcdfd2f0c98d.

[10] Cloudflare, “How to DDoS | DoS and DDoS attack tools,” [Trực tuyến]. Available: https://www.cloudflare.com/learning/ddos/ddos-attack-tools/how-to-ddos/.

[11] C. O'Sullivan, “Isolation Forest Guide: Explanation and Python Implementation,” [Trực tuyến]. Available: https://www.datacamp.com/tutorial/isolation-forest.