

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG
-----00-----



BÁO CÁO ĐỒ ÁN

**MÔN HỌC THỐNG TÌM KIẾM, PHÁT HIỆN VÀ
NGĂN NGỪA XÂM NHẬP**

Tên đề tài: ELK Stack

Giảng viên hướng dẫn : ThS. Đỗ Hoàng Hiển

Lớp : NT204.O21.ANTT

Khoá : 16

Sinh viên thực hiện: **MSSV:**

- | | |
|----------------------|----------|
| - Nguyễn Huy Cường | 21520667 |
| - Phan Gia Khánh | 21522213 |
| - Nguyễn Đức Tài | 21521395 |
| - Nguyễn Hoài Phương | 21520408 |
| - Trần Minh Duy | 21522010 |

TP. Hồ Chí Minh, tháng 5 năm 2024

LỜI MỞ ĐẦU

Trong bối cảnh an ninh mạng ngày càng trở nên phức tạp và thách thức, việc phát hiện và ngăn ngừa xâm nhập đã trở thành một nhiệm vụ thiết yếu đối với các tổ chức và doanh nghiệp. Sự phát triển không ngừng của các kỹ thuật tấn công và mối đe dọa mới đòi hỏi các hệ thống giám sát, phát hiện và ngăn ngừa xâm nhập phải luôn sẵn sàng và hiệu quả.

Đồ án được thực hiện trong một học kỳ bởi nhóm sinh viên ngành An toàn thông tin lớp NT204.O21.ANTT, dưới sự hướng dẫn của ThS. Đỗ Hoàng Hiển, nhằm nghiên cứu và triển khai hệ thống ELK Stack - một giải pháp mạnh mẽ và phổ biến trong việc thu thập, phân tích và hiển thị dữ liệu log. ELK Stack bao gồm Elasticsearch, Logstash, Kibana và Beats, sẽ giúp cải thiện hiệu quả giám sát và bảo vệ hệ thống.

Trong đồ án này, nhóm sẽ đi sâu vào tìm hiểu các thành phần của ELK Stack, phân tích và mô tả luồng hoạt động của hệ thống, cũng như triển khai và cấu hình ELK Stack trong môi trường mạng thực tế. Qua đó, đồ án sẽ đánh giá hiệu quả của hệ thống thông qua các tính năng thu thập log, quản lý log, tạo bảng điều khiển tùy chỉnh, và cảnh báo.

Với sự nỗ lực và tâm huyết của toàn bộ nhóm, chúng em hy vọng rằng đồ án này sẽ không chỉ giúp nâng cao hiểu biết và kỹ năng của chúng em trong lĩnh vực an toàn thông tin mà còn đóng góp một phần vào việc bảo vệ hệ thống mạng của các tổ chức trước những mối đe dọa ngày càng gia tăng trong tương lai.

Chúng em xin chân thành cảm ơn giảng viên hướng dẫn ThS. Đỗ Hoàng Hiển đã tận tình hỗ trợ và định hướng trong suốt quá trình thực hiện đồ án. Sự giúp đỡ và chỉ dẫn quý báu của thầy là nguồn động lực to lớn giúp nhóm hoàn thành đồ án này.

MỤC LỤC

CHƯƠNG 1 GIỚI THIỆU	1
1.1 Thực trạng hiện nay	1
1.2 Mục tiêu đồ án	1
1.3 Phương pháp nghiên cứu	2
CHƯƠNG 2 ELK STACK	3
2.1 Giới thiệu chung	3
2.2 Các thành phần trong ELK Stack	5
2.2.1 Elasticsearch	5
2.2.2 Logstash.....	8
2.2.3 Kibana.....	11
2.2.4 Beats	16
2.3 Luồng hoạt động của ELK Stack.....	21
2.3.1 Thu thập dữ liệu.....	22
2.3.2 Xử lý và biến đổi dữ liệu	22
2.3.3 Lưu trữ và tìm kiếm dữ liệu.....	23
2.3.4 Trực quan hóa và phân tích dữ liệu	23
2.4 Khả năng của ELK Stack	23
CHƯƠNG 3 PHÂN TÍCH HỆ THỐNG	25
3.1 Mục tiêu	25
3.2 Mô hình mạng triển khai	25
3.3 Phân tích và mô tả từng thành phần mạng:	28
3.3.1. WAN (Wide Area Network)	28

3.3.2. LAN (Local Area Network).....	28
3.3.3 Manage (Management Network).....	28
3.3.4 DMZ (Demilitarized Zone Network)	29
3.3.5 Kết luận.....	29
CHƯƠNG 4 HIỆN THỰC HỆ THỐNG	30
4.1 Cấu hình mạng trong VMNet	30
4.2 Cài đặt ELK Stack	30
4.2.1 Điều kiện tiên quyết.....	30
4.2.2 Cài đặt và cấu hình Elasticsearch	30
4.2.4 Cài đặt và cấu hình Logstash.....	32
4.3 Cài đặt Web Application Firewall (WAF) và Web Server	32
4.5 Cấu hình Rsyslog.....	35
CHƯƠNG 5 THỰC NGHIỆM VÀ ĐÁNH GIÁ	43
5.1 Kịch bản 1: Thu thập và chuẩn hóa log	43
5.2 Kịch bản 2: Quản lý và filter log	46
5.3 Kịch bản 3: Custom Dashboard Kibana	49
5.4 Kịch bản 4: Machine Learning với ELK Stack	52
5.5 Kịch bản 5. Security Alert	57
5.6 Kịch bản 6. Alert bất thường bằng Kibana.....	60
5.7. Kịch bản 7. Phát hiện outbound attack thông qua phân tích log bằng ELK Stack	63
CHƯƠNG 6 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	71
6.1 Kết luận.....	71
6.2 Hướng phát triển	72
DANH MỤC TÀI LIỆU THAM KHẢO	73

DANH MỤC HÌNH ẢNH

Hình 1 Thành phần của ELK Stack	4
Hình 2 Thành phần của Elasticsearch	5
Hình 3 Kiến trúc và thành phần của Logstash	8
Hình 4 Giao diện xem log của Kibana	11
Hình 5 Kiến trúc Kibana	12
Hình 6 Beats trong ELK Stack data architecture	16
Hình 7 Sơ lược luồng hoạt động của ELK Stack	22
Hình 8 Kiến trúc công nghệ triển khai	25
Hình 9 Kiến trúc mạng triển khai	26
Hình 10 Kiến trúc thu thập log triển khai	26
Hình 11 Kiến trúc bảo mật triển khai	27
Hình 12 Pfsense virtual machine	34
Hình 13 Pfsense GUI	34
Hình 14 Cấu hình syslog để gửi log Firewall và Snort	35
Hình 15 Cấu hình rsylog gửi log của modsecurity	35
Hình 16 Cấu hình rsylog gửi log của apache	36
Hình 17 Cài đặt và cấu hình OSSEC Agent trên Win 7	37
Hình 18 Đoạn lệnh thu thập log	38
Hình 19 Đoạn lệnh gán tag cho log nhận được (P1)	38
Hình 20 Đoạn lệnh gán tag cho log nhận được (P2)	39
Hình 21 Đoạn lệnh chuẩn hóa log pfsense (P1)	39
Hình 22 Đoạn lệnh chuẩn hóa log pfsense (P2)	40
Hình 23 Đoạn lệnh chuẩn hóa log pfsense (P3)	40
Hình 24 Đoạn lệnh chuẩn hóa log modsecurity (P1)	41
Hình 25 Đoạn lệnh chuẩn hóa log modsecurity (P2)	41
Hình 26 Đoạn lệnh chuẩn hóa log OSSEC	41
Hình 27 Đoạn lệnh chuẩn hóa log apache	42
Hình 28 Đoạn lệnh tạo ra các index	42
Hình 29 Kiểm tra log Firewall	44

Hình 30 Kiểm tra log Snort	44
Hình 31 Kiểm tra log modsecurity	45
Hình 32 Kiểm tra log apache	45
Hình 33 Kiểm tra log Ossec server	46
Hình 34 Kiểm tra log ossec agent	46
Hình 35 Quản lý log	47
Hình 36 Quản lý index trên Kibana	47
Hình 37 Filterlog modsecurity	48
Hình 38 Filterlog apache	48
Hình 39 Filter log ossec	49
Hình 40 Filter log pfsense	49
Hình 41 Giao diện tạo Dashboard	51
Hình 42 Dashboard	51
Hình 43 Dashboard kết hợp với filter	52
Hình 44 Phát hiện các bất thường trong số lượng mục nhập	53
Hình 45 Phát hiện các bất thường trên tần suất nhập log	53
Hình 46 Biểu đồ số lượng Log với nhãn là “1”	54
Hình 47 Biểu đồ số lượng Log với nhãn là “2”	54
Hình 48 Biểu đồ số lượng Log với nhãn là “3”	55
Hình 49 Phát hiện bất thường	55
Hình 50 Dự đoán số lượng log của nhãn “1” trong khoảng thời gian 1 ngày	56
Hình 51 Biểu đồ tần suất nhập Log	56
Hình 52 Dự đoán về tần suất nhập log trong vòng 1 ngày	57
Hình 53 Phát hiện bất thường	57
Hình 54 Set rule	58
Hình 55 Set action	59
Hình 56 Kiểm tra security alert	59
Hình 57 Check mail	60
Hình 58 Set rule	61
Hình 59 Set Action	62
Hình 60 Check mail	62

Hình 61 Quét nmap	64
Hình 62 Kết quả quét nmap	65
Hình 63 Kết quả tấn công web	65
Hình 64 Set rule pfsense	66
Hình 65 Set action pfsense	67
Hình 66 Set rule apache	68
Hình 67 Set action apache	69
Hình 68 Check mail alert pfsense	69
Hình 69 Check mail alert modsecurity	70
Hình 70 Check mail alert apache	70

CHƯƠNG 1

GIỚI THIỆU

1.1 Thực trạng hiện nay

Kỷ nguyên số bùng nổ mang đến vô vàng lợi ích, nhưng đồng thời cũng mở ra cánh cửa cho những hiểm họa khôn lường trong thế giới mạng. Các tổ chức, doanh nghiệp ngày càng đối mặt với vô số mối đe dọa ngày càng tinh vi và ẩn晦 hơn. Hàng ngày mỗi hệ thống ghi nhận thu thập vô vàng nhật ký tạo ra khối lượng dữ liệu khổng lồ cần phân tích. Các phương pháp bảo mật truyền thống đang gặp nhiều khó khăn trong việc theo kịp tốc độ phát triển của các mối đe dọa, dẫn đến nhiều thách thức cho doanh nghiệp:

- *Khó khăn trong việc thu thập và quản lý dữ liệu nhật ký hệ thống*: Dữ liệu bảo mật đến từ nhiều nguồn khác nhau, ở nhiều định dạng khác nhau, gây khó khăn cho việc thu thập và tổng hợp.
- *Thiếu khả năng phân tích dữ liệu*: Khối lượng dữ liệu bảo mật khổng lồ cần được phân tích để xác định các mối đe dọa tiềm ẩn, nhưng các công cụ phân tích truyền thống thường không đáp ứng được yêu cầu về tốc độ và hiệu quả.
- *Không thể phản ứng chậm chạp*: Việc phát hiện và phản ứng với các mối đe dọa thường diễn ra thủ công và tốn nhiều thời gian, dẫn đến nguy cơ thiệt hại lớn cho doanh nghiệp.

ELK Stack - giải pháp mã nguồn mở mạnh mẽ - chính là "cứu cánh" cho các tổ chức trong hành trình bảo vệ an ninh mạng. ELK Stack bao gồm 3 "chiến binh" dũng mãnh:

- **Elasticsearch**: Cơ sở dữ liệu lưu trữ và truy vấn dữ liệu bảo mật một cách hiệu quả, hỗ trợ phân tích theo thời gian thực.
- **Logstash**: "Ninja" thu thập và xử lý dữ liệu từ mọi nguồn, mọi định dạng, biến chúng thành thông tin dễ dàng phân tích.
- **Kibana**: Giao diện trực quan, giúp người dùng dễ dàng khám phá, phân tích và hiểu rõ dữ liệu bảo mật.

1.2 Mục tiêu đồ án

Trong phạm vi nghiên cứu của đồ án chúng em tập trung vào các nhiệm vụ hiểu biết, phân tích và trình bày chi tiết cấu trúc ELK Stack, bao gồm từng thành phần (Elasticsearch, Logstash, Kibana) và sự tương tác giữa chúng; hiểu được cách thức ELK Stack thu thập,

lưu trữ, xử lý và phân tích dữ liệu một cách hiệu quả; mô hình hóa và ứng dụng ELK Stack vào một mô hình mạng thu nhỏ, nhằm mô phỏng đa dạng những tác vụ trong doanh nghiệp.

1.3 Phương pháp nghiên cứu

- **Nghiên cứu lý thuyết:** Tài liệu tham khảo: Tìm kiếm và nghiên cứu tài liệu chính thức từ Elastic (nhà phát triển ELK Stack), bao gồm tài liệu hướng dẫn, blog, whitepaper,...
- **Nghiên cứu thực tiễn:** Cài đặt và sử dụng ELK Stack trên môi trường thực tế để trải nghiệm trực tiếp cách thức hoạt động của hệ thống, tham gia các diễn đàn, nhóm thảo luận trực tuyến về ELK Stack để học hỏi kinh nghiệm từ những người dùng khác, trao đổi kiến thức và giải đáp thắc mắc.
- **Phân tích và đánh giá:** Sử dụng ELK Stack để phân tích dữ liệu từ các nguồn thực tế, ví dụ như dữ liệu log hệ thống, log từ người dùng, log từ tường lửa, từ các dịch vụ trong doanh nghiệp,... Đồng thời đánh giá hiệu quả hoạt động của ELK Stack trong các trường hợp thực tế, xác định điểm mạnh, điểm yếu và đề xuất giải pháp cải tiến. So sánh các phương pháp phân tích dữ liệu khác nhau và giải thích lý do lựa chọn ELK Stack cho dự án.
- **Tổng hợp và báo cáo:** Xác định rõ ràng mục tiêu nghiên cứu, phạm vi nghiên cứu, phương pháp nghiên cứu và thời gian hoàn thành. Ghi chép đầy đủ quá trình nghiên cứu, bao gồm các tài liệu tham khảo, kết quả thu được và phân tích. Viết báo cáo trình bày kết quả nghiên cứu một cách rõ ràng, súc tích, logic và khoa học.

CHƯƠNG 2

ELK STACK

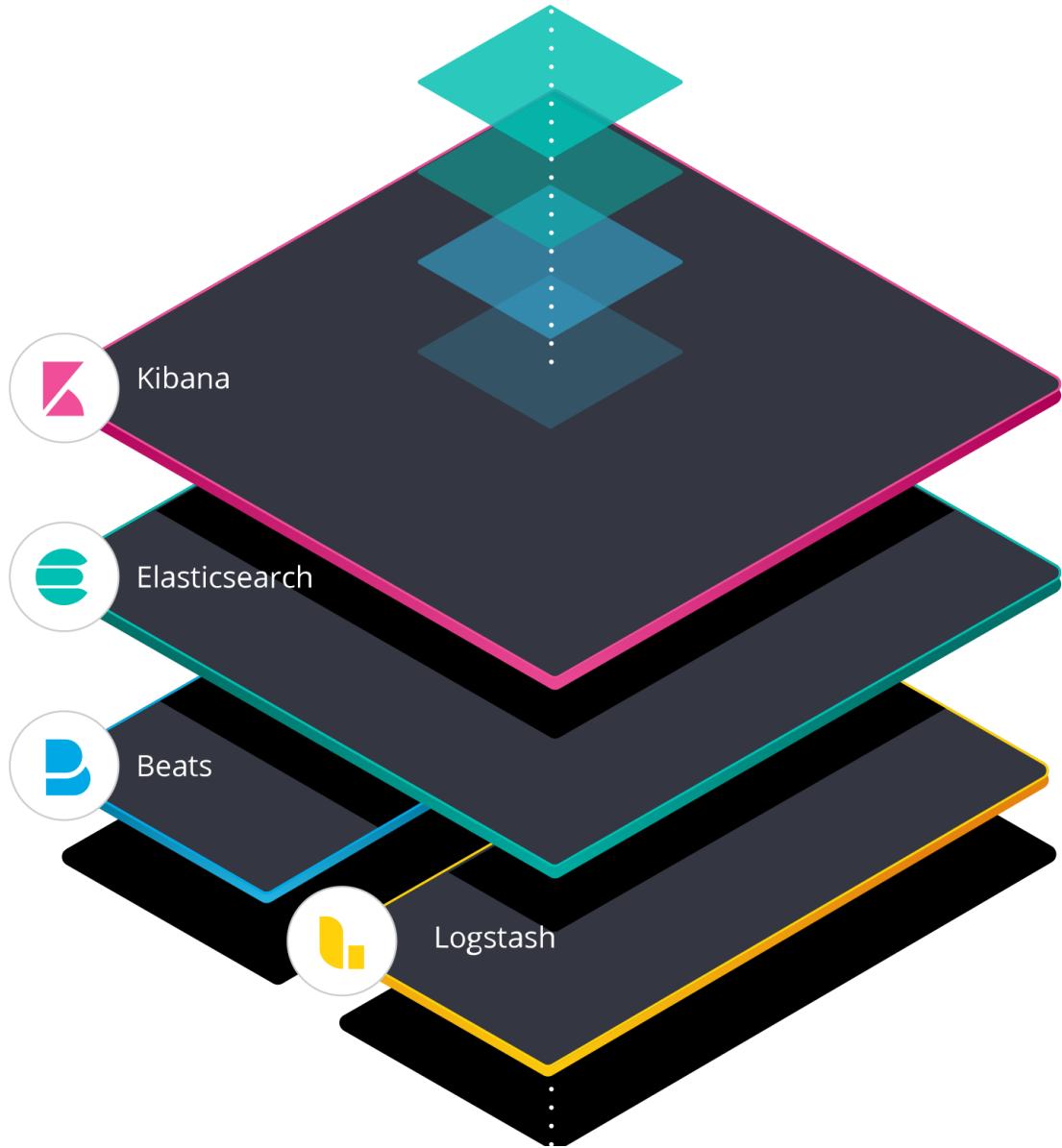
2.1 Giới thiệu chung

ELK Stack là một bộ công cụ mạnh mẽ cho việc thu thập, phân tích và hiển thị dữ liệu nhật ký (log data) trong thời gian thực. ELK là viết tắt của ba công cụ chính:

- + Elasticsearch: Một công cụ tìm kiếm và phân tích mạnh mẽ, có khả năng xử lý và tìm kiếm dữ liệu rất nhanh. Elasticsearch là một hệ thống lưu trữ dữ liệu dạng phi cấu trúc, có khả năng mở rộng và hỗ trợ đầy đủ tính năng của công cụ tìm kiếm.
- + Logstash: Một công cụ thu thập, xử lý và chuyển đổi dữ liệu nhật ký từ nhiều nguồn khác nhau. Logstash cho phép bạn xử lý dữ liệu theo nhiều cách khác nhau trước khi chuyển tiếp chúng đến Elasticsearch.
- + Kibana: Một công cụ hiển thị dữ liệu mạnh mẽ, cho phép người dùng tạo biểu đồ, báo cáo và bảng điều khiển (dashboard) để theo dõi và phân tích dữ liệu nhật ký đã được lưu trữ trong Elasticsearch.

Bên cạnh đó, thường có thêm **Beats** - một bộ công cụ nhẹ để gửi dữ liệu từ các điểm cuối (end points) đến Logstash hoặc Elasticsearch:

- + Beats: Một tập hợp các lightweight data shippers, dùng để thu thập và gửi dữ liệu nhật ký và chỉ số (metrics) đến Elasticsearch hoặc Logstash. Các thành phần phổ biến của Beats bao gồm Filebeat (thu thập file logs), Metricbeat (thu thập hệ thống metrics), Packetbeat (phân tích gói mạng), và nhiều hơn nữa.



Hình 1 Thành phần của ELK Stack

Ứng dụng của ELK Stack:

- + Giám sát hệ thống: Thu thập và phân tích nhật ký hệ thống để phát hiện lỗi và vấn đề.
- + Phân tích bảo mật: Phân tích dữ liệu nhật ký để phát hiện các hoạt động đáng ngờ và sự kiện bảo mật.

+ Theo dõi hiệu suất ứng dụng: Giám sát và tối ưu hóa hiệu suất của các ứng dụng và dịch vụ.

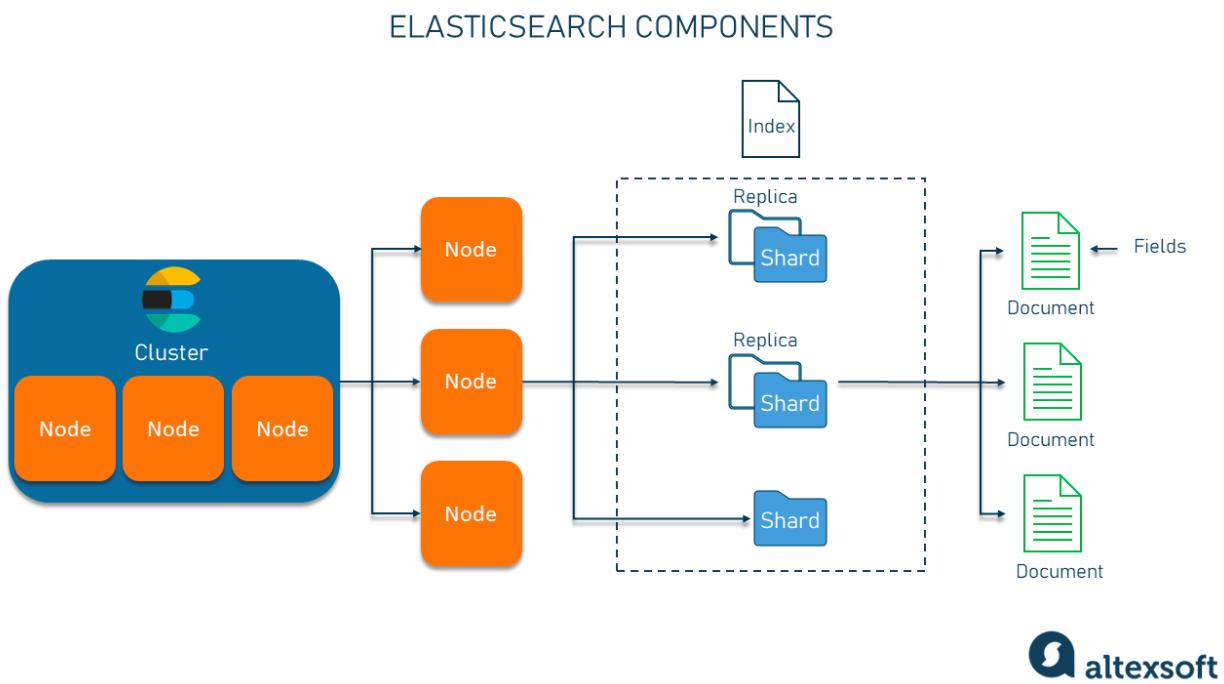
+ Phân tích kinh doanh (Business analysis): Sử dụng dữ liệu nhật ký để phân tích hành vi người dùng và tối ưu hóa chiến lược kinh doanh.

ELK Stack đã trở thành một giải pháp phổ biến cho việc quản lý và phân tích dữ liệu log trong nhiều lĩnh vực và ứng dụng khác nhau nhờ tính linh hoạt, mạnh mẽ và dễ sử dụng của nó.

2.2 Các thành phần trong ELK Stack

2.2.1 Elasticsearch

Elasticsearch là một công cụ tìm kiếm và phân tích mạnh mẽ được xây dựng trên Apache Lucene. Nó được thiết kế để xử lý các yêu cầu tìm kiếm phức tạp với tốc độ nhanh và khả năng mở rộng cao. Dưới đây là các chi tiết cụ thể hơn về Elasticsearch:



Hình 2 Thành phần của Elasticsearch

Kiến trúc và thành phần chính:

1. Cluster: Elasticsearch cluster là một tập hợp các node (máy chủ) làm việc cùng nhau để lưu trữ dữ liệu và cung cấp các chức năng tìm kiếm. Một cluster có thể chứa một hoặc nhiều node, và có một tên duy nhất để nhận diện.

2. Node: Mỗi instance của Elasticsearch chạy trên một máy chủ vật lý hoặc ảo được gọi là một node. Mỗi node là một phần của cluster và chịu trách nhiệm cho một phần dữ liệu và các hoạt động tìm kiếm.

3. Index: Index là nơi lưu trữ dữ liệu trong Elasticsearch. Một index có thể chứa nhiều loại tài liệu và mỗi tài liệu có nhiều trường (fields). Mỗi index có một tên duy nhất trong cluster.

4. Document: Document là đơn vị lưu trữ cơ bản của dữ liệu trong Elasticsearch. Mỗi document được lưu trữ trong một index và có một ID duy nhất.

5. Shards và Replicas:

+ Shards: Một index có thể được chia thành nhiều shard để tăng cường khả năng xử lý và lưu trữ dữ liệu. Mỗi shard là một instance của Lucene.

+ Replicas: Để đảm bảo tính sẵn sàng và khả năng chịu lỗi, mỗi shard có thể có một hoặc nhiều bản sao (replica). Nếu một node bị hỏng, các replica đảm bảo dữ liệu vẫn có sẵn.

Tính năng chính của Elasticsearch:

1. Tìm kiếm toàn văn bản (Full-text search): Elasticsearch hỗ trợ tìm kiếm toàn văn bản với các tính năng như đánh chỉ mục (indexing), phân tích cú pháp (analyzing), và tìm kiếm (searching) dữ liệu văn bản hiệu quả.

2. Tìm kiếm thời gian thực: Dữ liệu mới được lập chỉ mục gần như ngay lập tức, cho phép tìm kiếm và phân tích trong thời gian thực.

3. Phân tích và tổng hợp dữ liệu (Aggregations): Elasticsearch cung cấp khả năng phân tích và tổng hợp dữ liệu mạnh mẽ, cho phép người dùng thực hiện các phép toán như sum, avg, max, min, stats, histogram, v.v.

4. Khả năng mở rộng: Elasticsearch có khả năng mở rộng ngang tốt, cho phép thêm node mới vào cluster một cách dễ dàng để tăng khả năng xử lý và lưu trữ dữ liệu.

5. APIs RESTful: Elasticsearch cung cấp giao diện API RESTful, cho phép tương tác với cluster thông qua các phương thức HTTP như GET, POST, PUT, DELETE.

6. Hỗ trợ ngôn ngữ lập trình: Elasticsearch có thư viện và client cho nhiều ngôn ngữ lập trình phổ biến như Java, Python, Ruby, PHP, JavaScript, và Go.

7. Giao diện quản lý: Kibana, một công cụ giao diện người dùng, tích hợp với Elasticsearch để cung cấp các bảng điều khiển (dashboard) và báo cáo dữ liệu.

Cách Elasticsearch hoạt động:

1. Indexing: Dữ liệu được gửi đến Elasticsearch qua API RESTful và được lập chỉ mục. Elasticsearch phân tích và lưu trữ dữ liệu trong các shard của index.

2. Searching: Khi một truy vấn tìm kiếm được gửi đến, Elasticsearch sẽ tìm kiếm trong các shard của index và trả về kết quả phù hợp. Quá trình tìm kiếm bao gồm phân tích cú pháp truy vấn, tìm kiếm dữ liệu phù hợp và tổng hợp kết quả.

3. Replication and Fault Tolerance: Elasticsearch tự động sao lưu các shard để đảm bảo tính sẵn sàng của dữ liệu. Nếu một node bị hỏng, các shard replica trên các node khác sẽ đảm nhận công việc.

Ứng dụng của Elasticsearch:

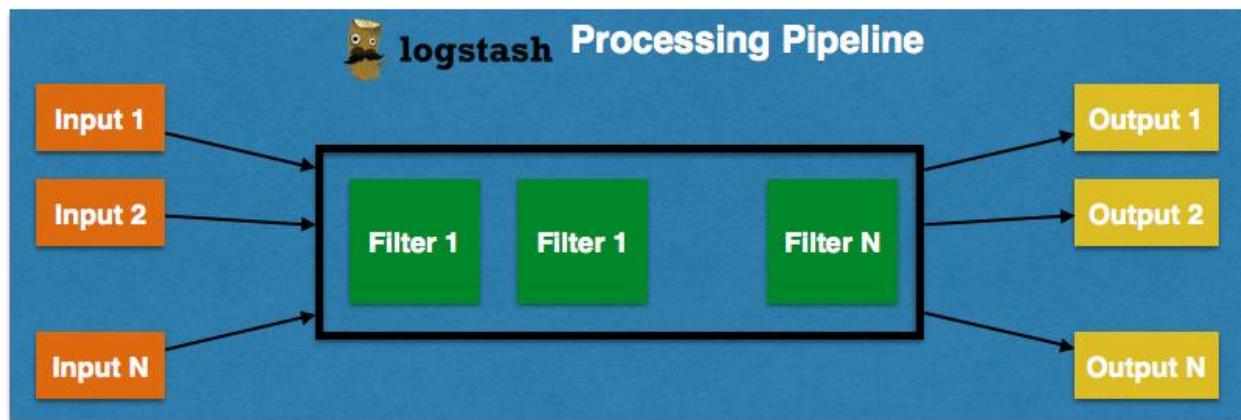
- + Giám sát hệ thống: Theo dõi và phân tích log hệ thống để phát hiện và khắc phục sự cố.
- + Tìm kiếm website: Cung cấp chức năng tìm kiếm toàn văn bản trên các website hoặc ứng dụng.
- + Phân tích dữ liệu: Sử dụng khả năng tổng hợp của Elasticsearch để phân tích dữ liệu lớn.
- + Quản lý thông tin doanh nghiệp: Tìm kiếm và phân tích dữ liệu khách hàng, giao dịch và các thông tin kinh doanh khác.

- + An ninh và bảo mật: Phân tích nhật ký bảo mật để phát hiện các hành vi đáng ngờ và ngăn chặn các mối đe dọa.

Elasticsearch là một công cụ tìm kiếm và phân tích mạnh mẽ, phù hợp với nhiều ứng dụng khác nhau từ giám sát hệ thống đến phân tích dữ liệu kinh doanh. Với khả năng mở rộng tốt và các tính năng phong phú, Elasticsearch là một lựa chọn lý tưởng cho các nhu cầu tìm kiếm và phân tích dữ liệu phức tạp.

2.2.2 Logstash

Logstash là một phần quan trọng trong ELK Stack, đóng vai trò là một công cụ thu thập, xử lý và chuyển đổi dữ liệu từ nhiều nguồn khác nhau trước khi chuyển tiếp chúng đến Elasticsearch hoặc các điểm đến khác. Dưới đây là những chi tiết về Logstash:



Hình 3 Kiến trúc và thành phần của Logstash

Kiến trúc và thành phần Chính:

1. Pipeline: Logstash sử dụng khái niệm pipeline để xử lý dữ liệu. Một pipeline bao gồm ba giai đoạn chính:

- + Inputs: Các đầu vào từ nhiều nguồn khác nhau.
- + Filters: Các bộ lọc để xử lý và biến đổi dữ liệu.
- + Outputs: Các điểm đến cho dữ liệu đã xử lý.

2. Plugin: Logstash sử dụng các plugin để mở rộng và tùy chỉnh chức năng của mình. Có ba loại plugin chính:

+ Input Plugins: Để thu thập dữ liệu từ các nguồn khác nhau (như file, syslog, kafka, beats, http, v.v.).

+ Filter Plugins: Để xử lý và biến đổi dữ liệu (như mutate, grok, date, geoip, json, csv, v.v.).

+ Output Plugins: Để gửi dữ liệu đến các điểm đến khác nhau (như Elasticsearch, file, stdout, http, kafka, v.v.).

3. Configuration Files: Cấu hình của Logstash được quản lý thông qua các file cấu hình, sử dụng ngôn ngữ DSL (Domain-Specific Language). Một file cấu hình Logstash điển hình bao gồm các phần input, filter, và output.

Tính năng chính của Logstash:

1. Thu thập dữ liệu đa nguồn: Logstash có khả năng thu thập dữ liệu từ nhiều nguồn khác nhau, bao gồm file logs, cơ sở dữ liệu, dịch vụ mạng, hệ thống hàng đợi tin nhắn, và nhiều nguồn khác.

2. Xử lý và biến đổi dữ liệu: Sử dụng các bộ lọc mạnh mẽ, Logstash có thể xử lý và biến đổi dữ liệu để phù hợp với yêu cầu của người dùng. Ví dụ, có thể sử dụng bộ lọc grok để phân tích cú pháp và trích xuất dữ liệu từ các chuỗi văn bản phức tạp.

3. Khả năng mở rộng: Logstash có một hệ thống plugin phong phú và có thể dễ dàng mở rộng thông qua các plugin tùy chỉnh.

4. Khả năng mở rộng ngang: Logstash có thể mở rộng ngang, cho phép triển khai trên nhiều node để xử lý khối lượng dữ liệu lớn.

Cách hoạt động của Logstash:

1. Input: Logstash bắt đầu bằng việc thu thập dữ liệu từ các nguồn đầu vào được cấu hình.

Dưới đây là một số ví dụ về input plugin:

```
file { path => "/var/log/syslog" }

http { port => 5044 }

kafka { topics => ["log-topic"] }
```

2. Filter: Sau khi dữ liệu được thu thập, Logstash sẽ áp dụng các bộ lọc để xử lý và biến đổi dữ liệu. Ví dụ, bạn có thể sử dụng filter `grok` để phân tích cú pháp và trích xuất dữ liệu từ các chuỗi log phức tạp:

```
filter {
  grok {
    match => { "message" => "%{COMBINEDAPACHELOG}" }
  }
  date {
    match => [ "timestamp" , "dd/MMM/yyyy:HH:mm:ss Z" ]
  }
  geoip {
    source => "clientip"
  }
}
```

3. Output: Sau khi dữ liệu được xử lý, Logstash sẽ gửi dữ liệu đến các điểm đến được cấu hình. Ví dụ:

```
elasticsearch { hosts => ["localhost:9200"] }

stdout { codec => rubydebug }

file { path => "/path/to/output.log" }
```

Ứng dụng của Logstash:

+ **Thu thập và xử lý log:** Logstash thường được sử dụng để thu thập và xử lý log từ các hệ thống khác nhau trước khi gửi đến Elasticsearch để lưu trữ và phân tích.

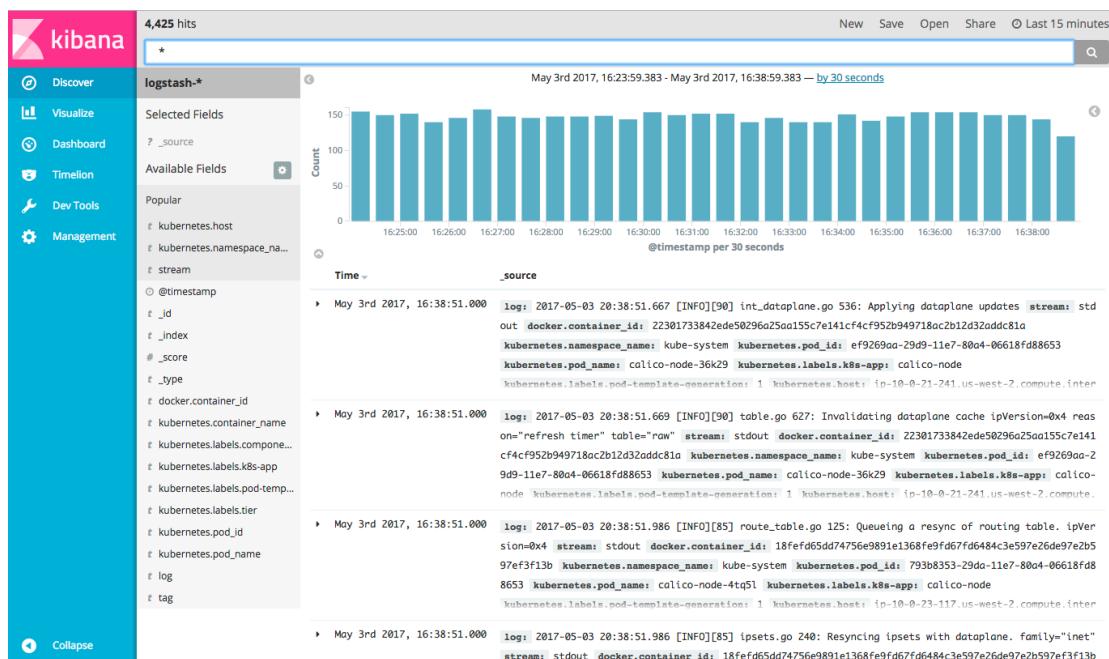
+ **ETL (Extract, Transform, Load):** Logstash có thể thực hiện các tác vụ ETL, trích xuất dữ liệu từ nhiều nguồn, biến đổi dữ liệu để phù hợp với yêu cầu phân tích và sau đó tải dữ liệu vào các hệ thống đích.

+ **Giám sát và bảo mật:** Logstash có thể thu thập dữ liệu giám sát và bảo mật từ nhiều nguồn, xử lý và gửi đến các hệ thống phân tích để phát hiện và phản ứng với các mối đe dọa bảo mật.

Logstash là một công cụ mạnh mẽ và linh hoạt cho việc thu thập, xử lý và chuyển đổi dữ liệu từ nhiều nguồn khác nhau. Với khả năng mở rộng tốt và tích hợp chặt chẽ với ELK Stack, Logstash là một phần không thể thiếu trong việc xây dựng các giải pháp quản lý và phân tích dữ liệu log hiệu quả.

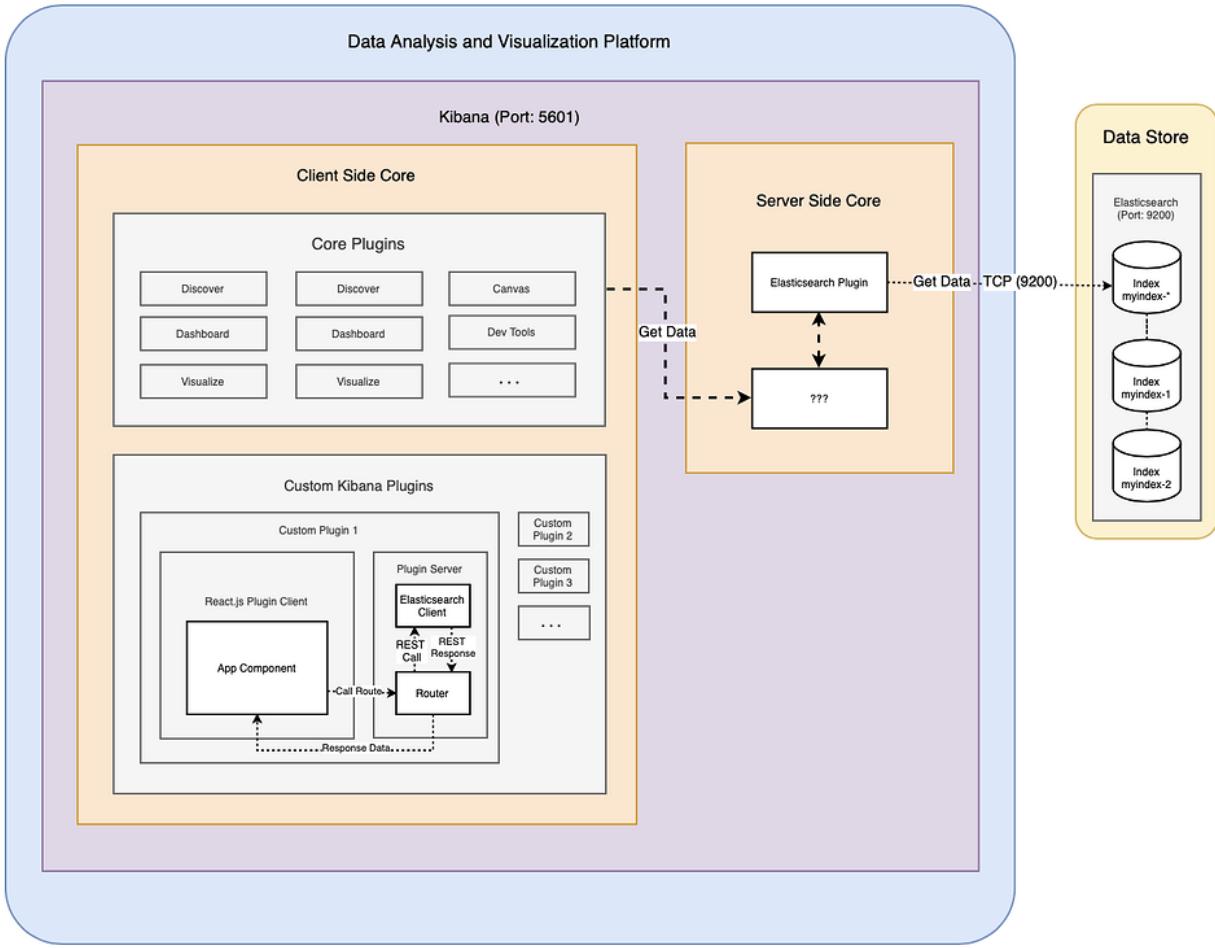
2.2.3 Kibana

Kibana là một công cụ giao diện người dùng mạnh mẽ và linh hoạt cho việc trực quan hóa và phân tích dữ liệu được lưu trữ trong Elasticsearch. Nó là một phần quan trọng của ELK Stack, cung cấp các khả năng hiển thị dữ liệu và tạo báo cáo trực quan. Dưới đây là các chi tiết cụ thể về Kibana:



Hình 4 Giao diện xem log của Kibana

Kiến trúc:



Hình 5 Kiến trúc Kibana

Client Side Core: Đây là phía người dùng của nền tảng, bao gồm các thành phần như Dashboard, Visualize, Discover và Dev Tools. Nó cũng cho phép các plugin tùy chỉnh để mở rộng chức năng.

Server Side Core: Đây là phía máy chủ của nền tảng, bao gồm một client Elasticsearch kết nối với Data Store. Data Store chứa các chỉ mục dữ liệu như Index Pattern 1 và Index Pattern 2.

Get Data: Đây là quá trình trung tâm kết nối Client Side Core và Server Side Core. Nó cho phép dữ liệu được truyền từ máy chủ đến người dùng.

Core Plugins: Đây là các plugin cốt lõi được cung cấp bởi nền tảng. Chúng bao gồm Dashboard, Visualize, Discover và Dev Tools.

Custom Kibana Plugins: Đây là các plugin tùy chỉnh được thêm vào bởi người dùng. Chúng cho phép người dùng mở rộng chức năng của nền tảng.

Data Store: Đây là nơi lưu trữ dữ liệu. Nó chứa các chỉ mục dữ liệu như Index Pattern 1 và Index Pattern 2.

HTTP API: Đây là giao diện lập trình ứng dụng cho phép người dùng tương tác với nền tảng qua HTTP.

Thành phần Chính:

1. Dashboard: Dashboard trong Kibana là nơi bạn có thể tạo và sắp xếp các visualizations để hiển thị dữ liệu một cách trực quan. Dashboard cho phép bạn kéo và thả các biểu đồ, đồ thị và bảng để tạo ra các bộ cục tùy chỉnh.

2. Visualizations: Visualizations là các thành phần cơ bản của Dashboard. Kibana hỗ trợ nhiều loại hình visualizations khác nhau, bao gồm:

+ Bar Charts: Biểu đồ cột.

+ Line Charts: Biểu đồ đường.

+ Pie Charts: Biểu đồ tròn.

+ Data Tables: Bảng dữ liệu.

+ Maps: Bản đồ địa lý.

+ Metric Visualizations: Hiển thị các số liệu cụ thể.

+ Timelion: Công cụ trực quan hóa dữ liệu chuỗi thời gian.

3. Discover: Chức năng Discover cho phép người dùng khám phá và tìm kiếm dữ liệu trong Elasticsearch. Bạn có thể lọc, sắp xếp và xem dữ liệu chi tiết từ các chỉ mục.

4. Canvas: Canvas là một không gian làm việc tùy chỉnh cho phép tạo ra các visualizations tương tác và báo cáo trực quan với khả năng thiết kế cao.

5. Machine Learning: Kibana tích hợp với các tính năng machine learning của Elasticsearch để phát hiện các mô hình và bất thường trong dữ liệu.

6. Dev Tools: Công cụ dành cho nhà phát triển cung cấp Console để chạy các truy vấn Elasticsearch và kiểm tra các chỉ số (indices).

7. Management: Phần quản lý cho phép cấu hình và quản lý các chỉ mục, không gian làm việc, người dùng và các thiết lập khác của Kibana và Elasticsearch.

Tính năng chính của Kibana:

1. Trực quan hóa dữ liệu: Kibana cung cấp nhiều loại biểu đồ và đồ thị để trực quan hóa dữ liệu từ Elasticsearch. Người dùng có thể dễ dàng tạo các visualizations từ dữ liệu bằng cách sử dụng các công cụ kéo thả.

2. Tạo báo cáo và Dashboard: Với Kibana, bạn có thể tạo và chia sẻ các dashboard tương tác để theo dõi và phân tích dữ liệu theo thời gian thực.

3. Phân tích dữ liệu chuỗi thời gian: Kibana có khả năng phân tích dữ liệu chuỗi thời gian, rất hữu ích cho việc giám sát và phân tích logs, metrics và các loại dữ liệu thời gian khác.

4. Khả năng tìm kiếm và lọc dữ liệu: Chức năng Discover cho phép người dùng tìm kiếm và lọc dữ liệu từ các chỉ mục Elasticsearch một cách nhanh chóng và dễ dàng.

5. Canvas: Tạo ra các báo cáo tùy chỉnh và trực quan với các biểu đồ tương tác và khả năng thiết kế cao.

6. Tích hợp Machine Learning: Tích hợp với các tính năng machine learning của Elasticsearch để tự động phát hiện bất thường và các xu hướng trong dữ liệu.

7. Bảo mật và Quản lý người dùng: Kibana hỗ trợ các tính năng bảo mật và quản lý người dùng, cho phép kiểm soát truy cập và phân quyền.

Cách Kibana hoạt động:

1. Kết nối với Elasticsearch: Kibana kết nối với các chỉ mục trong Elasticsearch và truy vấn dữ liệu từ các chỉ mục này để hiển thị.

2. Tạo Visualizations: Người dùng có thể tạo visualizations bằng cách chọn loại biểu đồ, chỉ định dữ liệu nguồn, và cấu hình các thuộc tính hiển thị.

3. Xây dựng Dashboard: Các visualizations có thể được sắp xếp vào dashboard để tạo ra các giao diện tùy chỉnh cho việc giám sát và phân tích.

4. Phân tích Dữ liệu: Kibana cung cấp các công cụ phân tích mạnh mẽ, bao gồm chức năng tìm kiếm, lọc và truy vấn dữ liệu.

5. Chia sẻ và Xuất báo cáo: Các dashboard và visualizations có thể được chia sẻ với các thành viên khác hoặc xuất ra để tạo báo cáo.

Ứng dụng của Kibana:

+ Giám sát hệ thống: Sử dụng để giám sát hiệu suất hệ thống và phát hiện lỗi thông qua các visualizations và dashboard.

+ Phân tích log: Phân tích và trực quan hóa dữ liệu log từ các ứng dụng và hệ thống.

+ Bảo mật: Theo dõi và phân tích dữ liệu bảo mật để phát hiện các hoạt động đáng ngờ.

+ Phân tích kinh doanh: Sử dụng dữ liệu kinh doanh để tạo ra các báo cáo và phân tích hành vi người dùng, doanh số bán hàng và các số liệu kinh doanh khác.

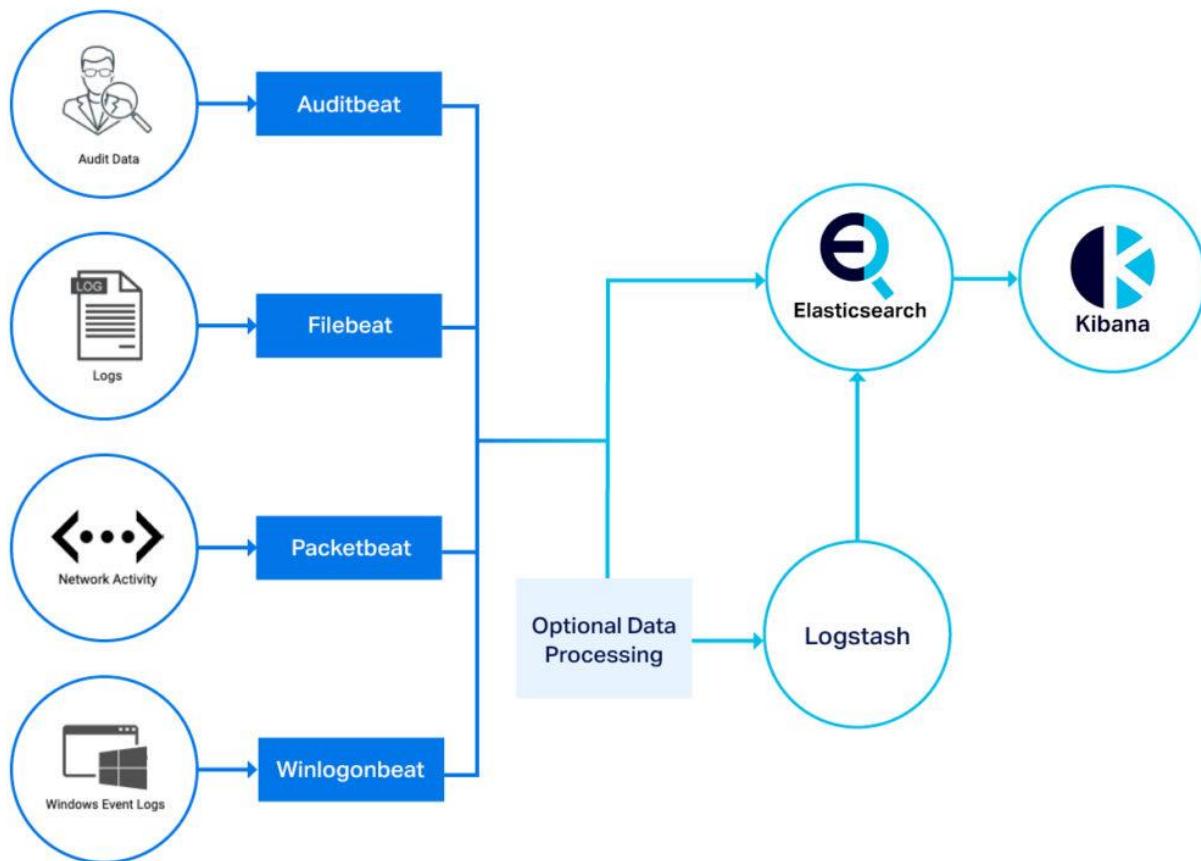
+ Giám sát hiệu suất ứng dụng: Theo dõi hiệu suất và tình trạng của các ứng dụng, phát hiện các vấn đề và tối ưu hóa hiệu suất.

Kibana là một công cụ không thể thiếu trong việc trực quan hóa và phân tích dữ liệu trong ELK Stack. Với các tính năng phong phú và khả năng mở rộng, Kibana cho phép người

dùng dễ dàng tạo các dashboard tùy chỉnh, phân tích dữ liệu và tạo báo cáo. Đây là một công cụ mạnh mẽ giúp nâng cao hiệu quả quản lý và phân tích dữ liệu trong nhiều lĩnh vực khác nhau.

2.2.4 Beats

Beats là một bộ công cụ nhẹ (lightweight shippers) được thiết kế để thu thập và gửi dữ liệu từ các điểm cuối (endpoints) đến Logstash hoặc Elasticsearch. Beats là thành phần bổ sung của ELK Stack, giúp thu thập và chuyển tiếp dữ liệu từ nhiều nguồn khác nhau một cách hiệu quả và dễ dàng. Dưới đây là các chi tiết cụ thể về Beats:



Hình 6 Beats trong ELK Stack data architecture

Tổng quan về Beats:

1. Kiến trúc: Beats hoạt động như các agent cài đặt trên máy chủ, thu thập dữ liệu và gửi nó tới Logstash hoặc Elasticsearch để xử lý và lưu trữ.

2. Cấu hình: Beats được cấu hình thông qua các tệp cấu hình YAML, cho phép người dùng dễ dàng định nghĩa các nguồn dữ liệu và các điểm đến.

Các thành phần chính của Beats

1. Filebeat

- Mục đích: Thu thập và chuyển tiếp dữ liệu log từ các file log.
- Cách hoạt động: Theo dõi và đọc nội dung mới từ các file log, sau đó gửi dữ liệu tới Logstash hoặc Elasticsearch.
- Ứng dụng: Thu thập log từ ứng dụng, hệ thống, và các dịch vụ khác.

Ví dụ cấu hình Filebeat:

```
filebeat.inputs:  
- type: log  
  paths:  
    - /var/log/*.log  
  
output.elasticsearch:  
  hosts: ["localhost:9200"]
```

2. Metricbeat

- Mục đích: Thu thập các chỉ số hệ thống và dịch vụ.
- Cách hoạt động: Định kỳ thu thập các chỉ số và gửi chúng đến Logstash hoặc Elasticsearch.
- Ứng dụng: Giám sát hiệu suất hệ thống, máy chủ, container, và các dịch vụ.

Ví dụ cấu hình Metricbeat:

```
metricbeat.modules:  
- module: system  
  metricsets:  
    - cpu
```

```
- memory  
- network  
period: 10s  
hosts: ["localhost"]  
  
output.elasticsearch:  
  hosts: ["localhost:9200"]
```

3. Packetbeat

- Mục đích: Thu thập và phân tích lưu lượng mạng.
- Cách hoạt động: Bắt gói dữ liệu mạng và phân tích các giao thức mạng, sau đó gửi dữ liệu tới Logstash hoặc Elasticsearch.
- Ứng dụng: Giám sát hiệu suất mạng và phân tích các giao thức.

Ví dụ cấu hình Packetbeat:

```
packetbeat.interfaces.device: any  
packetbeat.protocols:  
  - type: http  
    ports: [80, 8080, 9200]  
  
output.elasticsearch:  
  hosts: ["localhost:9200"]
```

4. Winlogbeat

- Mục đích: Thu thập và chuyển tiếp các sự kiện nhật ký Windows.
- Cách hoạt động: Đọc các sự kiện từ Windows Event Log và gửi chúng đến Logstash hoặc Elasticsearch.
- Ứng dụng: Giám sát và phân tích sự kiện bảo mật và hệ thống trên Windows.

Ví dụ cấu hình Winlogbeat:

```
winlogbeat.event_logs:  
  - name: Application  
  - name: Security  
  - name: System
```

```
output.elasticsearch:  
  hosts: ["localhost:9200"]
```

5. Auditbeat

- Mục đích: Thu thập và chuyển tiếp dữ liệu kiểm toán.
- Cách hoạt động: Theo dõi và ghi lại các hoạt động kiểm toán trên hệ thống, sau đó gửi dữ liệu đến Logstash hoặc Elasticsearch.
- Ứng dụng: Giám sát an ninh và kiểm toán hệ thống.

Ví dụ cấu hình Auditbeat:

```
auditbeat.modules:  
- module: audit  
  audit_rule_files: [ '${path.config}/audit.rules.d/*.conf' ]  
  audit_rules: |  
    -a always,exit -F arch=b64 -S execve -k exec  
  
output.elasticsearch:  
  hosts: ["localhost:9200"]
```

6. Heartbeat

- Mục đích: Kiểm tra tính khả dụng và hiệu suất của các dịch vụ và hệ thống.
- Cách hoạt động: Định kỳ gửi các yêu cầu tới các dịch vụ để kiểm tra tính khả dụng và đo lường độ trễ, sau đó gửi kết quả đến Logstash hoặc Elasticsearch.
- Ứng dụng: Giám sát tính khả dụng và độ trễ của các dịch vụ web, ứng dụng, và cơ sở dữ liệu.

Ví dụ cấu hình Heartbeat:

```
heartbeat.monitors:  
- type: http  
  urls: ["http://localhost:9200"]  
  schedule: '@every 10s'
```

```
output.elasticsearch:  
  hosts: ["localhost:9200"]
```

7. Functionbeat

- Mục đích: Thu thập và chuyển tiếp dữ liệu từ các nền tảng serverless như AWS Lambda.
- Cách hoạt động: Cấu hình để các hàm serverless gửi dữ liệu đến Logstash hoặc Elasticsearch.
- Ứng dụng: Thu thập log và chỉ số từ các dịch vụ serverless.

Ví dụ cấu hình Functionbeat:

```
functionbeat.provider.aws.functions:  
  - name: cloudwatch  
    enabled: true  
    type: cloudwatch_logs  
    triggers:  
      - log_group_name: /aws/lambda/my-function  
  
output.elasticsearch:  
  hosts: ["localhost:9200"]
```

Lợi ích của Beats:

- 1. Nhẹ và hiệu quả:** Beats là các công cụ nhẹ, dễ cài đặt và cấu hình, không tiêu tốn nhiều tài nguyên hệ thống.
- 2. Dễ dàng tích hợp:** Beats dễ dàng tích hợp với Logstash và Elasticsearch, giúp xây dựng một pipeline thu thập và xử lý dữ liệu mạnh mẽ.
- 3. Đa dạng:** Có nhiều loại Beats phù hợp với nhiều loại dữ liệu và yêu cầu giám sát khác nhau.
- 4. Khả năng mở rộng:** Beats có thể dễ dàng mở rộng để thu thập dữ liệu từ nhiều nguồn và gửi dữ liệu đến nhiều điểm đến khác nhau.

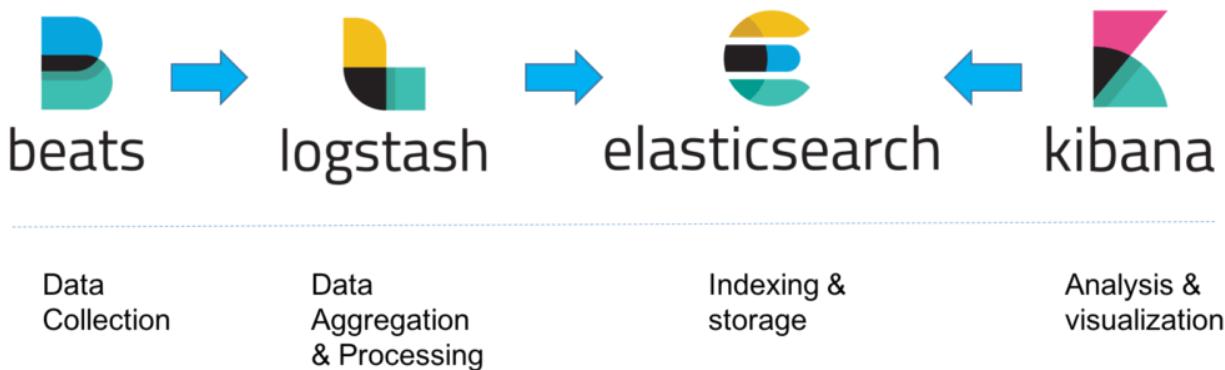
Ứng dụng của Beats:

- + Giám sát hệ thống: Thu thập và gửi dữ liệu log và chỉ số hệ thống để phân tích và giám sát.
- + An ninh và kiểm toán: Thu thập dữ liệu sự kiện bảo mật và kiểm toán để phát hiện các hoạt động đáng ngờ.
- + Giám sát mạng: Phân tích lưu lượng mạng để giám sát hiệu suất và phát hiện các vấn đề mạng.
- + Kiểm tra tính khả dụng: Kiểm tra và theo dõi tính khả dụng và hiệu suất của các dịch vụ web và ứng dụng.
- + Thu thập dữ liệu Serverless: Thu thập log và chỉ số từ các nền tảng serverless như AWS Lambda.

Beats là một thành phần quan trọng của ELK Stack, cung cấp các công cụ nhẹ và hiệu quả để thu thập và gửi dữ liệu từ các điểm cuối đến Logstash và Elasticsearch. Với nhiều loại Beats phù hợp với nhiều loại dữ liệu khác nhau, Beats giúp đơn giản hóa quá trình thu thập và phân tích dữ liệu, giúp các tổ chức giám sát hệ thống, bảo mật và hiệu suất một cách hiệu quả.

2.3 Luồng hoạt động của ELK Stack

ELK Stack (Elasticsearch, Logstash, và Kibana) hoạt động cùng nhau để thu thập, xử lý, lưu trữ, và trực quan hóa dữ liệu từ nhiều nguồn khác nhau. Dưới đây là luồng hoạt động của ELK Stack:



Hình 7 Sơ lược luồng hoạt động của ELK Stack

2.3.1 Thu thập dữ liệu

Beats là các agent nhẹ được cài đặt trên các máy chủ hoặc điểm cuối để thu thập dữ liệu. Sử dụng beats để thu thập dữ liệu.

Luồng hoạt động:

1. Các agent của Beats được cài đặt và cấu hình trên các máy chủ.
2. Beats thu thập dữ liệu từ các nguồn cụ thể (log files, hệ thống, mạng, v.v.).
3. Beats gửi dữ liệu tới Logstash hoặc trực tiếp tới Elasticsearch.

2.3.2 Xử lý và biến đổi dữ liệu

Logstash là một pipeline xử lý dữ liệu mạnh mẽ, có khả năng thu thập, biến đổi và gửi dữ liệu đến các điểm đích khác nhau. Sử dụng Logstash để xử lý và biến đổi dữ liệu.

Luồng hoạt động:

1. Logstash nhận dữ liệu từ các nguồn đầu vào.
2. Dữ liệu được xử lý và biến đổi thông qua các bộ lọc.
3. Dữ liệu đã xử lý được gửi tới Elasticsearch để lưu trữ.

2.3.3 Lưu trữ và tìm kiếm dữ liệu

Elasticsearch là một công cụ tìm kiếm và phân tích phân tán, được thiết kế để lưu trữ, tìm kiếm và phân tích khối lượng lớn dữ liệu thời gian thực. Sử dụng Elasticsearch để lưu trữ và tìm kiếm dữ liệu

Luồng hoạt động:

1. Elasticsearch nhận dữ liệu từ Logstash hoặc trực tiếp từ Beats.
2. Dữ liệu được lưu trữ trong các indices.
3. Elasticsearch cung cấp các khả năng tìm kiếm và phân tích dữ liệu thời gian thực.

2.3.4 Trực quan hóa và phân tích dữ liệu

Kibana là một công cụ trực quan hóa và phân tích dữ liệu, cung cấp giao diện người dùng để tương tác với dữ liệu trong Elasticsearch. Sử dụng Kibana để trực quan hóa và phân tích dữ liệu.

Luồng hoạt động:

1. Kibana kết nối với Elasticsearch để truy xuất dữ liệu.
2. Người dùng tạo và quản lý các visualizations, dashboards, và báo cáo.
3. Dữ liệu được hiển thị dưới dạng trực quan hóa để dễ dàng phân tích và giám sát.

2.4 Khả năng của ELK Stack

Đọc log từ nhiều nguồn: Logstash có thể đọc được log từ rất nhiều nguồn đến từ nhiều hệ thống như IDPS, Firewall, WAF, cho đến log file của server, endpoint, log database, UDP hay REST request.

Dễ tích hợp: Logstash được thiết kế để dễ dàng tích hợp với nhiều công nghệ và hệ thống khác nhau. Nó tương thích với Nginx hay Apache, MSSQL, MongoDB hay Redis. Logstash có khả năng đọc hiểu và xử lý log từ các công nghệ trên. Điều này giúp việc tích hợp ELK Stack vào môi trường hiện có trở nên đơn giản và thuận tiện..

Dễ dàng triển khai và hoàn toàn miễn phí: ELK Stack là một open source vì vậy nó hoàn toàn miễn phí đồng thời nó được sử dụng khá phổ biến và có nhiều tài liệu hỗ trợ cách sử dụng vì vậy nó dễ dàng cài đặt và triển khai.

Khả năng scale tốt: ELK Stack được thiết kế để có khả năng mở rộng dễ dàng. Cả Logstash và Elasticsearch (thành phần lưu trữ) có thể chạy trên nhiều node, cho phép mở rộng hệ thống khi cần. Khi muốn xử lý log nhiều hơn, thêm node cho Logstash và Elasticsearch sẽ giúp tăng khả năng xử lý log và tải.

Search và filter mạnh mẽ: Elasticsearch, thành phần lưu trữ trong ELK Stack, là một công cụ tìm kiếm phân tán và mạnh mẽ. Nó cho phép lưu trữ thông tin kiểu NoSQL và hỗ trợ tìm kiếm toàn văn bản (Full-Text Search). Điều này giúp việc thực hiện các truy vấn phức tạp trên log và lọc dữ liệu dễ dàng và hiệu quả.

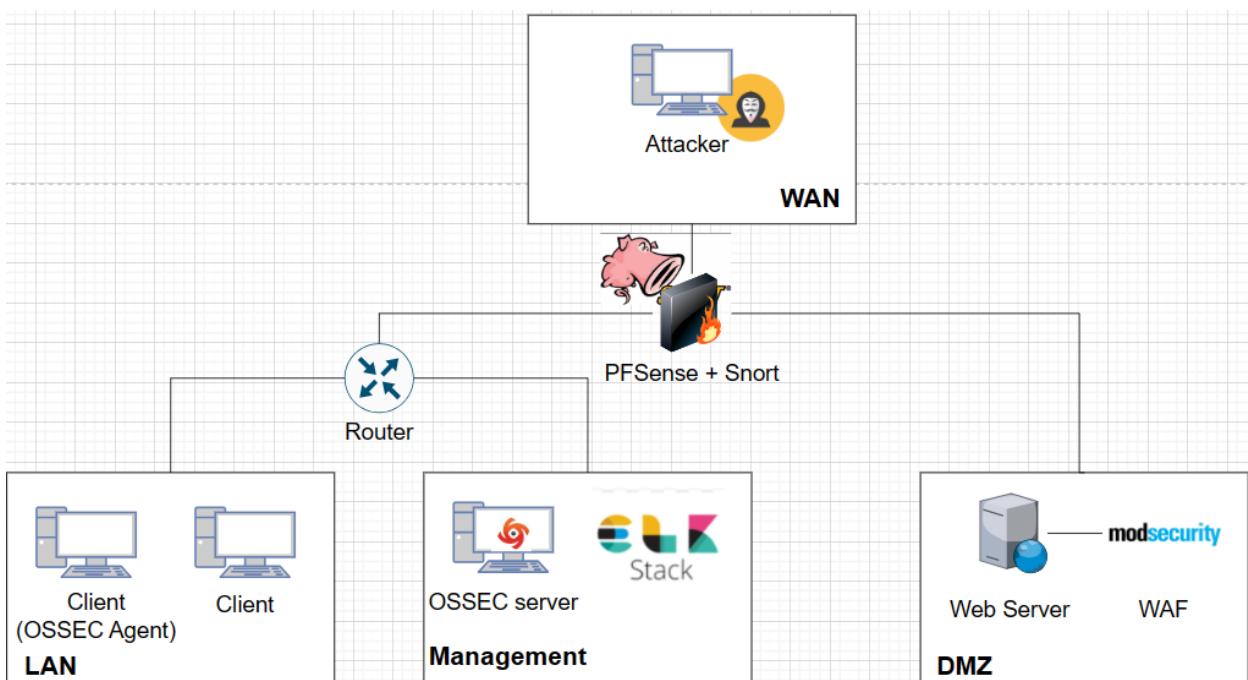
Cộng đồng mạnh, tutorial nhiều: ELK Stack có một cộng đồng người dùng rộng lớn và năng động. Vì nó được sử dụng rộng rãi trong các công ty và tổ chức, có rất nhiều tài liệu, tutorial, và hướng dẫn trực tuyến để học và sử dụng ELK Stack.

CHƯƠNG 3 PHÂN TÍCH HỆ THỐNG

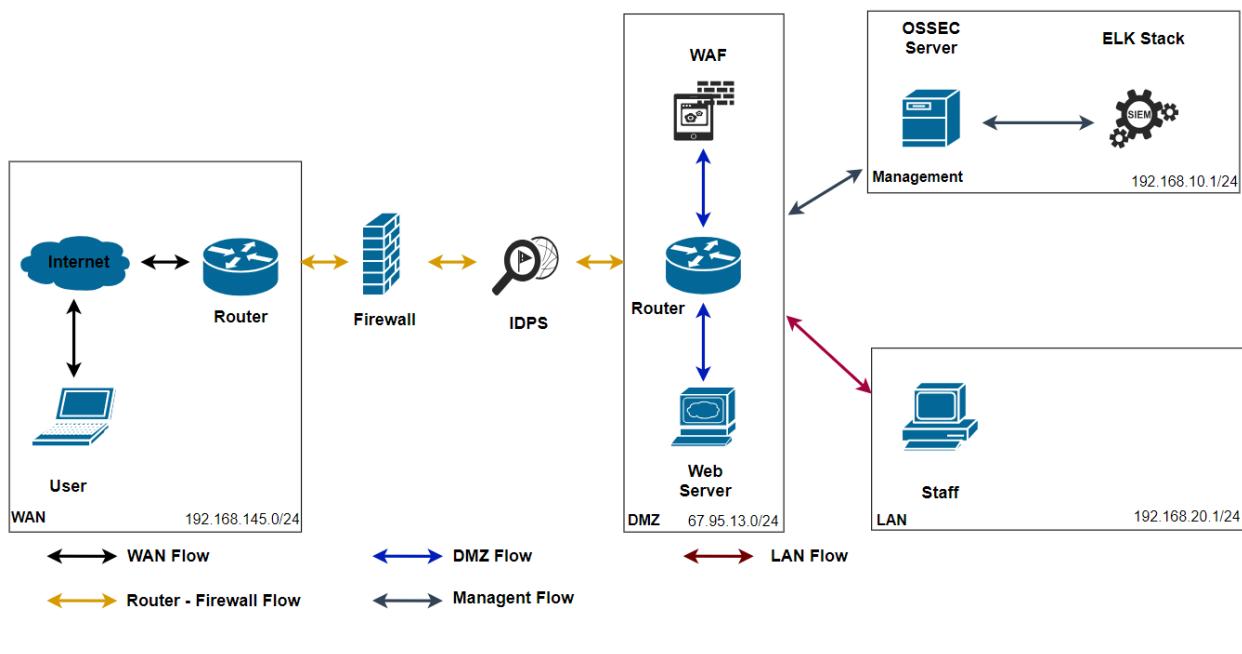
3.1 Mục tiêu

Thu thập và Phân tích Log: Do nhu cầu ngày càng tăng về việc thu thập, lưu trữ và phân tích log từ các hệ thống và ứng dụng khác nhau, việc triển khai một hệ thống hiệu quả để xử lý lượng log lớn trở nên quan trọng. Trong phạm vi đồ án này nhóm sẽ tích hợp ELK stack với topology mạng như hình dưới để lấy log

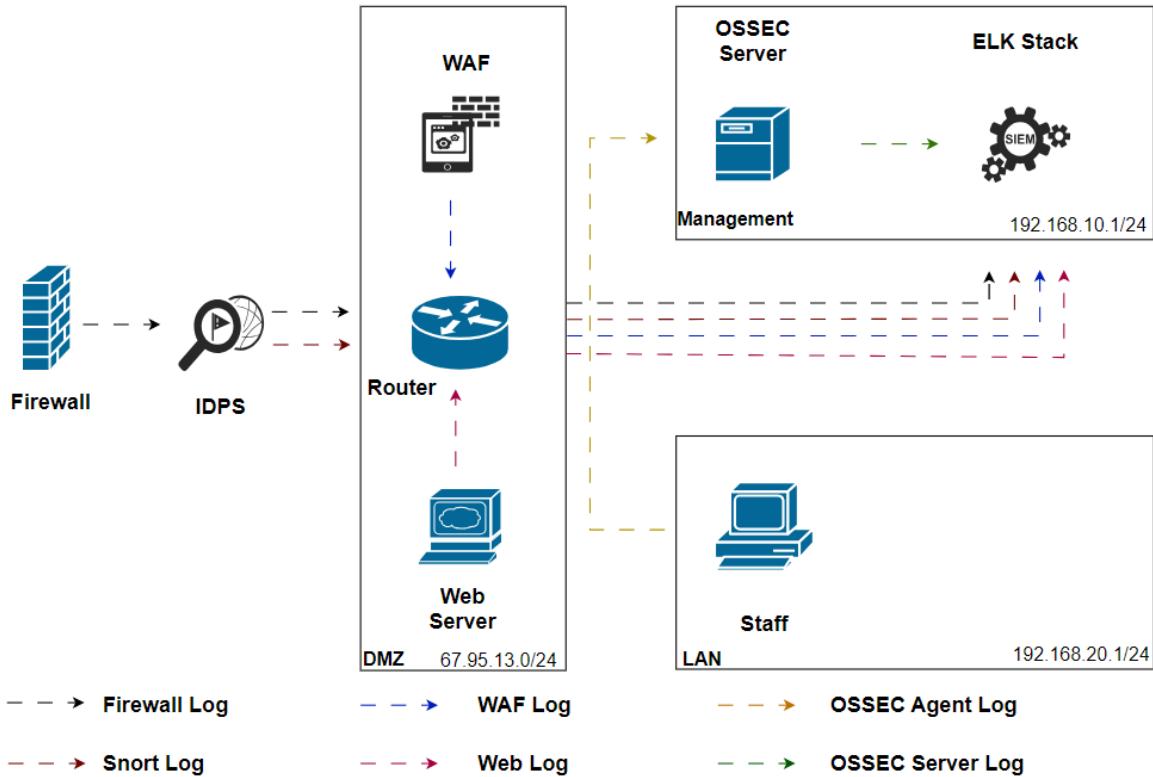
3.2 Mô hình mạng triển khai



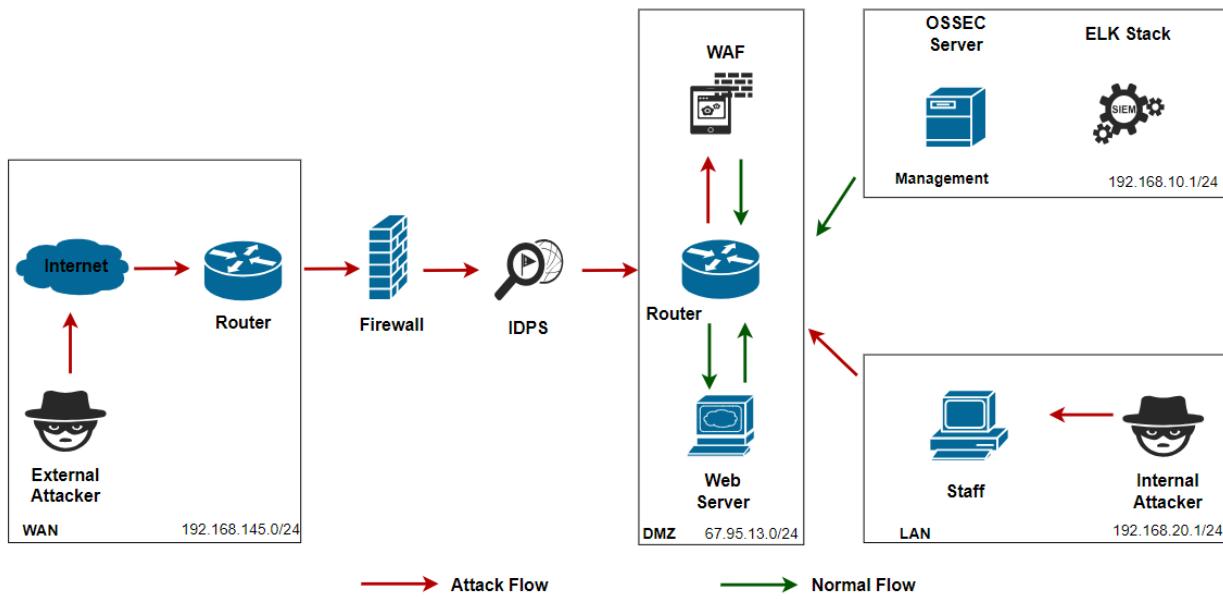
Hình 8 Kiến trúc công nghệ triển khai



Hình 9 Kiến trúc mạng triển khai



Hình 10 Kiến trúc thu thập log triển khai



Hình 11 Kiến trúc bảo mật triển khai

Bảng chi tiết địa chỉ IP của các thành phần trong mạng

Thiết bị	Interface	Ipv4 Address	Default Gateway
PFSense + Snort	NAT	192.168.145.145/24	192.168.145.1
	VMnet7	67.95.13.1/24	
	VMnet9	192.168.5.1/24	
Router	VMnet9	192.168.5.2/24	192.168.5.1
	VMnet2	192.168.10.1/24	
	VMnet3	192.168.20.1/24	
WebServer	VMnet7	67.95.13.100/24	67.95.13.1
OSSEC Server	VMnet2	192.168.10.150/24	192.168.10.1
ELK Stack	VMnet2	192.168.10.100/24	192.168.10.1
Client	VMnet3	192.168.20.100/24	192.168.20.1
Attacker	NAT	192.168.145.136/24	192.168.145.1

3.3 Phân tích và mô tả từng thành phần mạng:

Trong mô hình triển khai này, chúng ta sử dụng VMware phiên bản 17.5 để tạo ra một hệ thống bao gồm bốn vùng mạng: WAN, LAN, Manage và DMZ. Mỗi vùng mạng sẽ có một chức năng và nhiệm vụ cụ thể để đảm bảo sự an toàn và hiệu quả của hệ thống.

3.3.1. WAN (Wide Area Network)

Chức năng: Kết nối hệ thống mạng nội bộ của tổ chức với Internet bên ngoài.

Nhiệm vụ:

- Tiếp nhận và xử lý lưu lượng mạng từ Internet.
- Đảm bảo an ninh thông qua việc sử dụng các thiết bị và phương pháp bảo mật như tường lửa (PFSense) và hệ thống phát hiện xâm nhập (IDS Snort).
- Kiểm soát truy cập từ bên ngoài vào hệ thống nội bộ.

3.3.2. LAN (Local Area Network)

Chức năng: Mạng nội bộ phục vụ cho việc kết nối và trao đổi dữ liệu giữa các máy tính và thiết bị trong tổ chức.

Nhiệm vụ:

- Cung cấp môi trường mạng ổn định và an toàn cho các user và thiết bị nội bộ.
- Hỗ trợ chia sẻ tài nguyên như web server,....
- Đảm bảo tốc độ và hiệu suất cao cho các ứng dụng nội bộ.

3.3.3 Manage (Management Network)

Chức năng: Khu vực mạng dành riêng cho việc quản lý và giám sát các thiết bị và dịch vụ mạng.

Nhiệm vụ:

- Cung cấp quyền truy cập quản trị an toàn cho các quản trị viên hệ thống.
- Hỗ trợ giám sát và quản lý các thiết bị mạng như switch, router và máy chủ.
- Đảm bảo rằng các công cụ quản lý và giám sát không bị can thiệp từ các khu vực mạng khác.

3.3.4 DMZ (Demilitarized Zone Network)

Chức năng: Khu vực mạng trung gian giữa mạng nội bộ (LAN) và mạng bên ngoài (WAN), được thiết kế để lưu trữ các dịch vụ công cộng như web server, mail server.

Nhiệm vụ:

- Cung cấp một lớp bảo mật bổ sung, giúp bảo vệ mạng nội bộ khỏi các cuộc tấn công từ Internet.
- Lưu trữ các dịch vụ mà người dùng bên ngoài có thể truy cập, giảm thiểu rủi ro cho mạng nội bộ.
- Kiểm soát và giám sát lưu lượng mạng đến và đi từ các dịch vụ công cộng.

3.3.5 Kết luận

Mô hình triển khai này trên VMware phiên bản 17.5 với bốn vùng mạng WAN, LAN, Manage và DMZ giúp tạo ra một hệ thống mạng linh hoạt và bảo mật. Mỗi vùng mạng được cấu hình và quản lý chặt chẽ để đáp ứng các yêu cầu cụ thể về chức năng và an ninh, đảm bảo sự hoạt động liên tục và hiệu quả của toàn hệ thống.

Trong đó các loại log sẽ lấy bao gồm:

1. OS: auth log, message log
2. Service: Apache, ModSecurity
3. Security: FireWall (PFsense), Snort, OSSEC agent log, OSSEC server log

Việc lấy các loại log này sẽ được nhóm thực hiện ở những kịch bản khác nhau, chi tiết ở chương 5

CHƯƠNG 4

HIỆN THỰC HỆ THỐNG

4.1 Cấu hình mạng trong VMNet

4.2 Cài đặt ELK Stack

Để cài đặt ELK Stack, chúng ta sẽ thực hiện cài đặt lần lượt các thành phần của nó gồm Elasticsearch, Kibana, Logstash

4.2.1 Điều kiện tiên quyết

Trong chương này, chúng ta sẽ thực hiện cài đặt trên Ubuntu 22.04 với cấu hình tối thiểu là 4GB RAM và 2 CPUs.

Cài đặt OpenJDK 11 cho Ubuntu 22.04. Hướng dẫn đầy đủ [tại đây](#).

Cài đặt và cấu hình nginx proxy cho Kibana. Hướng dẫn đầy đủ [tại đây](#).

4.2.2 Cài đặt và cấu hình Elasticsearch

Các thành phần Elasticsearch không có sẵn trong kho gói mặc định của Ubuntu. Tuy nhiên, chúng có thể được cài đặt bằng APT sau khi thêm danh sách nguồn gói của Elastic.

Bước 1: Thêm kho lưu trữ Elasticsearch vào hệ thống của mình:

```
curl -fsSL https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo gpg --dearmor -o /usr/share/keyrings/elastic.gpg
```

Bước 2: Thêm danh sách Elastic source vào thư mục sources.list.d

```
echo "deb [signed-by=/usr/share/keyrings/elastic.gpg] https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-7.x.list
```

Bước 3: Cập nhật danh sách gói của bạn để APT sẽ đọc Elastic source mới và cài đặt Elasticsearch

```
sudo apt update && sudo apt install elasticsearch
```

Bước 4: Sử dụng công cụ nano để chỉnh sửa cấu hình của Elasticsearch trong file elasticsearch.yml

```
sudo nano /etc/elasticsearch/elasticsearch.yml
```

Bước 5: Khởi động và cho phép elasticsearch hoạt động:

```
sudo systemctl start elasticsearch && sudo systemctl enable elasticsearch
```

4.2.3 Cài đặt và cấu hình Kibana

Bước 1: Sử dụng apt để cài đặt Kibana:

```
sudo apt install kibana
```

Bước 2: Khởi động và cho phép Kibana hoạt động:

```
sudo systemctl start kibana && sudo systemctl enable kibana
```

Bước 3: Tạo người dùng và mật khẩu Kibana quản trị và lưu trữ chúng trong tệp htpasswd.users

```
echo "<ten_dang_nhap>:`openssl passwd -apr1`" | sudo tee -a /etc/nginx/htpasswd.users
```

Bước 4: Tạo một nginx server block file và cấu hình

```
sudo nano /etc/nginx/sites-available/your_domain
```

Nội dung file / etc/nginx/sites-available/your_domain:

```
server {  
    listen 80;  
  
    server_name your_domain;  
  
    auth_basic "Restricted Access";  
    auth_basic_user_file /etc/nginx/htpasswd.users;  
  
    location / {  
        proxy_pass http://localhost:<PORT>;  
        proxy_http_version 1.1;  
        proxy_set_header Upgrade $http_upgrade;
```

```
proxy_set_header Connection 'upgrade';
proxy_set_header Host $host;
proxy_cache_bypass $http_upgrade;
}
}
```

Bước 5: Kích hoạt cấu hình mới bằng cách tạo một symbolic link đến thư mục sites-enabled

```
sudo ln -s /etc/nginx/sites-available/your_domain /etc/nginx/sites-enabled/your_domain
```

Bước 6: Reload nginx

```
sudo systemctl reload nginx
```

4.2.4 Cài đặt và cấu hình Logstash

Bước 1: Cài đặt bằng gói apt:

```
sudo apt install logstash
```

Bước 2: Di chuyển đến thư mục /etc/logstash/conf.d để thực hiện các cấu hình cần thiết

- + Ví dụ, cấu hình nơi thiết lập Filebeat tại /etc/logstash/conf.d/02-beats-input.conf
- + Cấu hình output tại /etc/logstash/conf.d/30-elasticsearch-output.conf

Bước 3: Kiểm tra cấu hình Logstash

```
sudo -u logstash /usr/share/logstash/bin/logstash --path.settings /etc/logstash -t
```

Bước 4: Khi cấu hình hoàn tất, khởi động và cho phép logstash hoạt động

```
sudo systemctl start logstash && sudo systemctl enable logstash
```

4.3 Cài đặt Web Application Firewall (WAF) và Web Server

4.3.1. Cài đặt Web Server

- Trong đồ án này, nhóm sử dụng DVWA có nhiều lỗ hổng web trên đó để xây dựng web server. Từ đó attacker có thể khai thác lỗ hổng trên DVWA để nhóm có thể thực hiện thực nghiệm của mình.

- Damn Vulnerable Web Application (DVWA) là một ứng dụng mã nguồn PHP/MySQL tập hợp sẵn các lỗi logic về bảo mật ứng dụng web trong mã nguồn PHP. Lỗi logic khi lập trình có thể áp dụng đối với các loại ngôn ngữ lập trình nhằm giảm thiểu khả năng tạo ra lỗ hổng bảo mật từ tự duy lập trình chưa cẩn thận.
- Hệ điều hành sử dụng để cài đặt: Ubuntu 22.04
- Các bước thực hiện: [Tai đây](#)

4.3.2. Cài đặt Web Application Firewall (WAF)

- Trong đồ án này, nhóm sử dụng ModSecurity để bảo vệ web server và đưa log về cho ELK Stack.
- Mod Security là một module tường lửa có thể tích hợp với các Web Application Server (máy chủ ứng dụng web) như Apache, IIS, Nginx cho phép phân tích và ghi nhật ký các luồng dữ liệu HTTP/S. Mod Security đứng trước Web Server, làm nhiệm vụ như một firewall để kiểm soát truy cập vào Web Server. Các thông tin đi từ bên ngoài vào và bên trong ra sẽ được kiểm soát chặt chẽ để tránh những thông tin có thể gây hại cho Web Server hay là việc rò rỉ các thông tin đặc biệt từ Web Server đến Client.

- Các bước thực hiện: [Tai đây](#)

4.4. Cài đặt và cấu hình Firewall, Snort

- Trong đồ án này, nhóm sử dụng pfsense để thực hiện chức năng firewall và snort trong hệ thống.
- Pfsense là một ứng dụng có chức năng định tuyến vào tường lửa mạng và miễn phí dựa trên nền tảng FreeBSD có chức năng định tuyến và tường lửa rất mạnh. Pfsense được cấu hình qua giao diện GUI trên nền web nên có thể quản lý một cách dễ dàng. Nó hỗ trợ lọc theo địa chỉ nguồn, đích, cũng như port nguồn hay port đích đồng thời hỗ trợ định tuyến và có thể hoạt động trong chế độ bridge hay transparent. Nếu sử dụng pfsense là gateway,

ta cũng có thể thấy rõ việc hỗ trợ NAT và port forward trên pfSense cũng như thực hiện cân bằng tải hay failover trên các đường mạng.

```
*** Welcome to pfSense 2.7.2-RELEASE (amd64) on pfSense ***

WAN (wan)      -> em0      -> v4/DHCP4: 192.168.145.145/24
LAN (lan)      -> em1      -> v4: 192.168.5.1/24
OPT1 (opt1)    -> em2      -> v4: 67.95.13.1/24

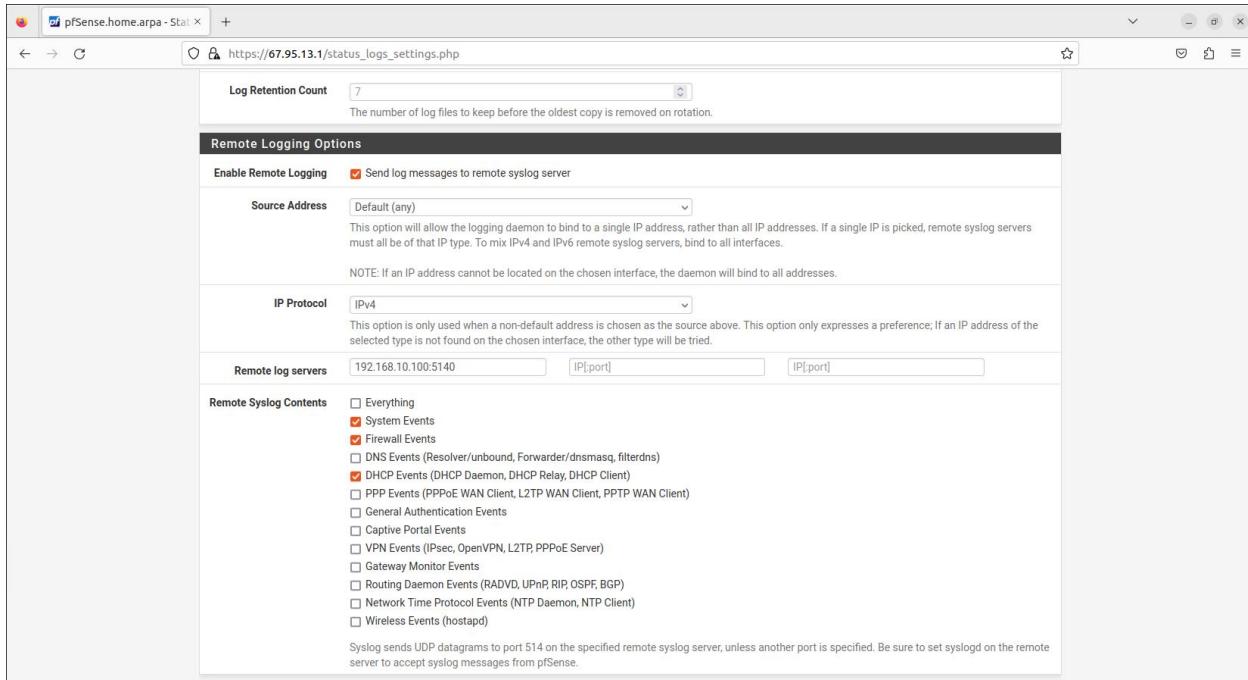
0) Logout (SSH only)          9) pfTop
1) Assign Interfaces          10) Filter Logs
2) Set interface(s) IP address 11) Restart webConfigurator
3) Reset webConfigurator password 12) PHP shell + pfSense tools
4) Reset to factory defaults   13) Update from console
5) Reboot system               14) Enable Secure Shell (sshd)
6) Halt system                 15) Restore recent configuration
7) Ping host                   16) Restart PHP-FPM
8) Shell
```

Hình 12 PfSense virtual machine

The screenshot shows the pfSense 2.7.2-RELEASE (amd64) dashboard. At the top, there's a warning message: "WARNING: The 'admin' account password is set to the default value. Change the password in the User Manager." Below this, the main navigation bar includes links for System, Interfaces, Firewall, Services, VPN, Status, Diagnostics, and Help. The central area has two main sections: "System Information" on the left and "Netgate Services And Support" on the right. The "System Information" section contains detailed hardware and software specifications. The "Netgate Services And Support" section provides information about community support and upgrade options, including a link to Netgate.com.

Hình 13 PfSense GUI

- Cấu hình syslog để gửi log Firewall và Snort:



Hình 14 Cấu hình syslog để gửi log Firewall và Snort

4.5 Cấu hình Rsyslog

- Trong đồ án này nhóm sử dụng Rsyslog thay vì filebeat để gửi log với chức năng tương tự với filebeat
- Rsyslog là một phần mềm mã nguồn mở sử dụng trên Linux dùng để chuyển tiếp các log message đến một địa chỉ trên mạng (log receiver, log server) Nó thực hiện giao thức syslog cơ bản, đặc biệt là sử dụng TCP cho việc truyền tải log từ client tới server. Hiện nay rsyslog là phần mềm được cài đặt sẵn trên hầu hết hệ thống Unix và các bản phân phối của Linux như : Fedora, openSUSE, Debian, Ubuntu, Red Hat Enterprise Linux, FreeBSD...

```
web@web-virtual-machine: /etc/rsyslog.d
GNU nano 6.2                                     modsecurity.conf
$template modsecurity,"<%PRI%>%timegenerated% %HOSTNAME% %syslogtag% %msg%"
# file access
$InputFileName /var/log/apache2/error.log
$InputFileTag web.apache.mod-security.pro.myapp.www1:
$InputFileStateFile stat-file1-ModSecurityAudit
$InputFileSeverity info
$InputFileFacility local7
$InputFilePollInterval 1
$InputFilePersistStateInterval 1
$InputRunFileMonitor
if $syslogtag contains 'web.apache.mod-security' and $syslogfacility-text == 'local7' then @@192.168.10.100:5140;modsecurity :syslogtag, contains, "web.apache.mod-security"
```

Hình 15 Cấu hình rsyslog gửi log của modsecurity

```

GNU nano 6.2
$ModLoad lmodule
$InputFilePollInterval 10
$PrivDropToGroup adm
$WorkDirectory /var/spool/rsyslog

# Apache Access Log
$InputFileName /var/log/apache2/access.log
$InputFileTag apache: #must change to 'apache' for AlienVault plugin
$InputFileStateFile stat-apache-access
$InputFileSeverity info
$InputFileFacility local2
$InputFilePersistStateInterval 20000
$InputRunFileMonitor

```

Hình 16 Cấu hình rsyslog gửi log của apache

4.6. Cài đặt và cấu hình Host-based IDPS:

- Trong đồ án này nhóm sử dụng OSSEC để thực hiện chức năng này.
- IDS là hệ thống phát hiện các dấu hiệu của tấn công xâm nhập, đồng thời có thể khởi tạo các hành động trên thiết bị khác để ngăn chặn tấn công. Khác với tường lửa, IDS không thực hiện các thao tác ngăn chặn truy nhập mà chỉ theo dõi các hoạt động trên mạng để tìm ra các dấu hiệu của tấn công và cảnh báo cho người quản trị mạng.

4.6.1. Cài đặt và cấu hình OSSEC Server

- Đây là phần trung tâm và quan trọng nhất của OSSEC. Server là nơi lưu trữ dữ liệu. Tất cả các luật, bộ giải mã (decoder) cũng được lưu trữ trên server.
- Server còn đảm nhận nhiệm vụ quản lý các agent. Các agent kết nối với máy chủ trên cổng 1514 hoặc 514, giao thức UDP. Kết nối với cổng này phải được cho phép để các agent kết nối với manager.

- Cài đặt và cấu hình OSSEC Server: [Tai đây](#)

- Cấu hình gửi syslog sang cho ELK Stack:

```

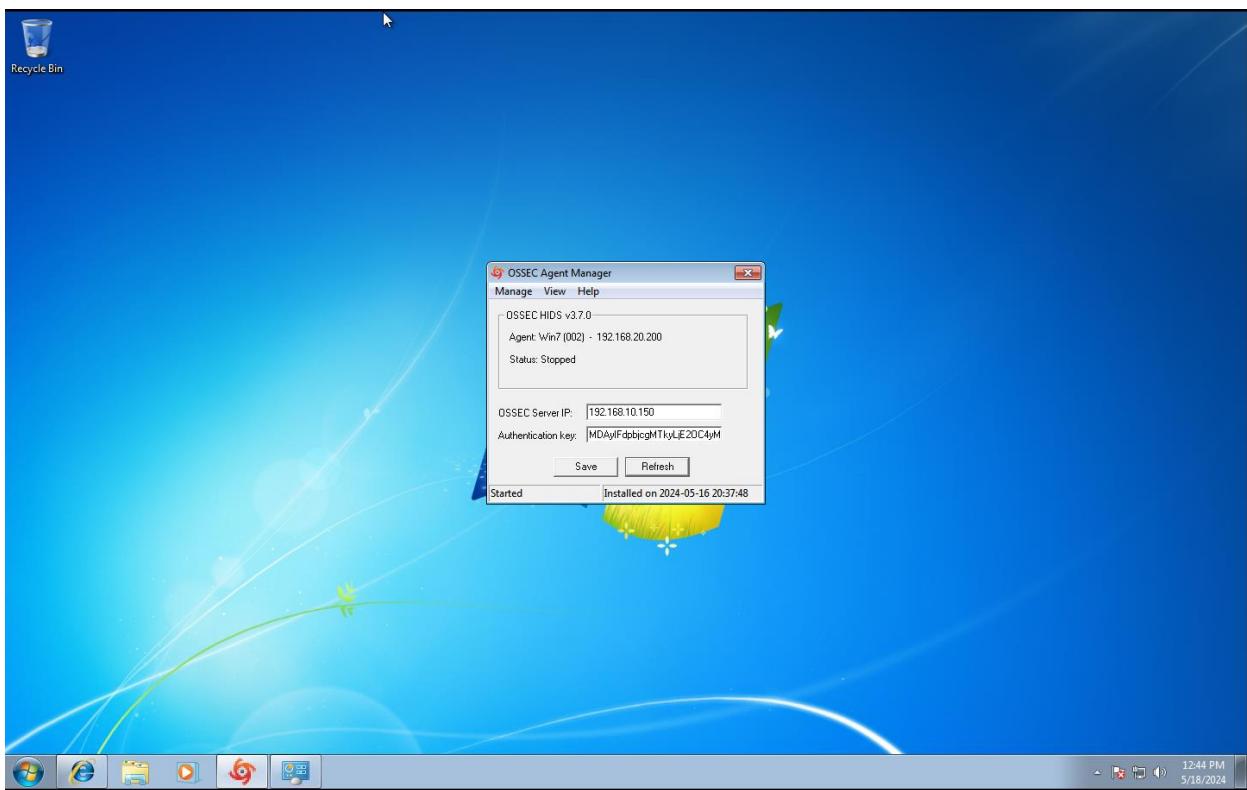
<syslog_output>
<server>192.168.10.100</server>
<port>5140</port>

```

```
<format>default</format>  
</syslog_output>
```

4.6.2. Cài đặt và cấu hình OSSEC Agent

- Là một chương trình nhỏ, hoặc tập hợp các chương trình, được cài đặt trên các hệ thống được giám sát.
- Agent sẽ thu thập thông tin và gửi cho manager để phân tích và so sánh. Một số thông tin được thu thập trong thời gian thực, những thông tin khác theo định kỳ.



Hình 17 Cài đặt và cấu hình OSSEC Agent trên Win 7

4.7. Thu thập, chuẩn hóa, lưu trữ log trên ELK Stack

- Đoạn lệnh thu thập log:

```

GNU nano 6.2                                         01-inputs.conf
input {
  tcp {
    type => "syslog"
    port => 5140
  }
}

input {
  udp {
    type => "syslog"
    port => 5140
  }
}

input {
  udp {
    type => "apachesyslog"
    port => 5141
  }
}

```

Hình 18 Đoạn lệnh thu thập log

- Đoạn lệnh gán tag cho các log nhận được để phân loại:

```

filter {
  if [type] == "syslog" {
    if [host] =~ /192\.168\.\d{1,3}\.1/ {
      mutate {
        add_tag => ["pfSense", "Ready"]
      }
    }

    if [host] =~ /67\.95\.13\.100/ {
      mutate {
        add_tag => ["ModSecurity", "Ready"]
      }
    }

    if [host] =~ /192\.168\.\d{1,3}\.\d{2}/ {
      mutate {
        add_tag => ["ossec", "Ready"]
      }
    }

    if "Ready" not in [tags] {
      mutate {
        add_tag => [ "syslog" ]
      }
    }
  }

  if [type] == "apachesyslog" {
    if [host] =~ /67\.95\.13\.100/ {
      mutate {
        add_tag => ["apache", "Ready"]
      }
    }
  }
}

filter {
  if [type] == "syslog" {
    mutate {
      remove_tag => "Ready"
    }
  }
}

if [type] == "apachesyslog" {
  mutate {

```

Hình 19 Đoạn lệnh gán tag cho log nhận được (P1)

```
if [type] == "apachesyslog" {
    mutate {
        remove_tag => "Ready"
    }
}
filter {
    if "syslog" in [tags] {
        grok {
            match => [ "message" => "%{SYSLOGTIMESTAMP:syslog_timestamp} %{SYSLOGHOST:syslog_hostname} %{DATA:syslog_program}(?:\[ %{POSINT:syslog_pid}\])?: %{GREEDYDATA:syslog_message}" ]
            add_field => [ "received_at", "%{@timestamp}" ]
            add_field => [ "received_from", "%{host}" ]
        }
        syslog_pri { }
        date {
            match => [ "syslog_timestamp", "MMM d HH:mm:ss", "MMM dd HH:mm:ss" ]
            locale => "en"
        }
        if !("_grokparsefailure" in [tags]) {
            mutate {
                replace => [ "@source_host", "%{syslog_hostname}" ]
                replace => [ "@message", "%{syslog_message}" ]
            }
        }
        mutate {
            remove_field => [ "syslog_hostname", "systog_message", "syslog_timestamp" ]
        }
    }
}
```

Hình 20 Đoạn lệnh gán tag cho log nhận được (P2)

- Đoạn lệnh chuẩn hóa log pfSense

```
GNU nano 6.2                                     11-pfsense.conf

filter {
  if "pfsense" in [tags] {
    grok {
      add_tag => [ "firewall" ]
      match => [ "message", "<?<evtid.*>(<?<datetime>(?>Jan(?>uary)?|Feb(?>ruary)?|Mar(?>ch)?|Apr(?>il)?|May|Jun(?>e)?|Jul(?>y)?|Aug(?>ust)?|Sep(?>tember)?|Oct(?>ober)?|Nov(?>emb?)"
    }
    mutate {
      gsub => [ "datetime", " ", " " ]
    }
    date {
      match => [ "datetime", "MMM dd HH:mm:ss" ]
      timezone => "Asia/Ho_Chi_Minh"
    }
    mutate {
      replace => [ "message", "%{msg}" ]
    }
    mutate {
      remove_field => [ "msg", "datetime" ]
    }
    if [prog] =~ /^dhcpd$/ {
      mutate {
        add_tag => [ "dhcpd" ]
      }
    }
    grok {
      patterns_dir => ["/etc/logstash/conf.d/patterns"]
      match => [ "message", "%{DHCPD}" ]
    }
  }
  if [prog] =~ /suricata/ {
    mutate {
      add_tag => [ "SuricataIDPS" ]
    }
    grok {
      patterns_dir => ["/etc/logstash/conf.d/patterns"]
      match => [ "message", "%{PFSENSE_SURICATA}" ]
    }
  }
  if ![_geoip] and [_ds_src_ip] !~ /(10.|192.|168\.)/ {
    geoip {
      add_tag => [ "GeoIP" ]
      source => "_ds_src_ip"
      database => "/etc/logstash/GeoLite2-City.mmdb"
    }
  }
}
```

Hình 21 Đoạn lệnh chuẩn hóa log pfsense (P1)

```

if [prog] =~ /^suricata/ {
    mutate [
        add_tag => [ "ET-Sig" ]
        add_field => [ "Signature_Info", "http://doc.emergingthreats.net/bin/view/Main/%{[ids_sig_id]}" ]
    ]
}
if [prog] =~ /charonS/ {
    mutate [
        add_tag => [ "ipsec" ]
    ]
}
if [prog] =~ /barnyard2/ {
    mutate [
        add_tag => [ "barnyard2" ]
    ]
}
if [prog] =~ /openvpn/ {
    mutate [
        add_tag => [ "openvpn" ]
    ]
}
if [prog] =~ /ntpd/ {
    mutate [
        add_tag => [ "ntpd" ]
    ]
}
if [prog] =~ /php-fpm/ {
    mutate [
        add_tag => [ "web_portal" ]
    ]
    grok {
        patterns_dir => ["/etc/logstash/conf.d/patterns"]
        match => [ "message", "%{PFSENSE_APP}%{PFSENSE_APP_DATA}" ]
    }
    mutate [
        lowercase => [ 'pfSense_ACTION' ]
    ]
}
if [prog] =~ /apinger/ {

```

Hình 22 Đoạn lệnh chuẩn hóa log pfSense (P2)

```

if [prog] =~ /apinger/ {
    mutate [
        add_tag => [ "apinger" ]
    ]
}
if [prog] =~ /^filterlog$/ {
    mutate [
        remove_field => [ "msg", "datetime" ]
    ]
    grok {
        add_tag => [ "firewall" ]
        patterns_dir => ["/etc/logstash/conf.d/patterns"]
        match => [ "message", "%{PFSENSE_LOG_DATA}%{PFSENSE_IP_SPECIFIC_DATA}%{PFSENSE_IP_DATA}%{PFSENSE_PROTOCOL_DATA}",
                    "message", "%{PFSENSE_IPv4_SPECIFIC_DATA}%{PFSENSE_IP_DATA}%{PFSENSE_PROTOCOL_DATA}",
                    "message", "%{PFSENSE_IPv6_SPECIFIC_DATA}%{PFSENSE_IP_DATA}%{PFSENSE_PROTOCOL_DATA}" ]
    }
    mutate [
        lowercase => [ 'proto' ]
    ]
    if ![geolp] and [src_ip] !~ /^(10\.|192\.|168\.)/ {
        geolp [
            add_tag => [ "GeoIP" ]
            source => "src_ip"
            database => "/etc/logstash/GeoLite2-City.mmdb"
        ]
    }
}

```

Hình 23 Đoạn lệnh chuẩn hóa log pfSense (P3)

- Đoạn lệnh chuẩn hóa log modsecurity:

```

filter {
  if "ModSecurity" in [tags] {
    # Extract event time, log severity level, source of attack (client), and the alert message.
    grok {
      match => { "message" => "(?<event_time>%{MONTH}\s%{MONTHDAY}\s%{TIME}\s%{YEAR})\s[\:\%{LOGLEVEL:log_level}.*client\s%{IPORHOST:src_ip}:\d+]\s(?<alert_message>.*)" }
    }
    # Extract Rules File from Alert Message
    grok {
      match => { "alert_message" => "(?<rulesfile>[file \"/(.+).conf\"])" }
    }
    grok {
      match => { "rulesfile" => "(?<rules_file>/.+.conf)" }
    }
    # Extract Attack Type from Rules File
    grok {
      match => { "rulesfile" => "(?<attack_type>[A-Z]+-[A-Z][^.]+)" }
    }
    # Extract Rule ID from Alert Message
    grok {
      match => { "alert_message" => "(?<ruleid>[id \"(\d+)\"])" }
    }
    grok {
      match => { "ruleid" => "(?<rule_id>\d+)" }
    }
    # Extract Attack Message (msg) from Alert Message
    grok {
      match => { "alert_message" => "(?<msg>[msg \s(.*)\"])" }
    }
    grok {
      match => { "msg" => "(?<alert_msg>\"(.*)\")" }
    }
    # Extract the User/Scanner Agent from Alert Message
    grok {
      match => { "alert_message" => "(?<scanner>User-Agent' \sValue: '(.*)')" }
    }
    grok {
      match => { "scanner" => "(?<user_agent>:(.*))" }
    }
    grok {
      match => { "alert_message" => "(?<matched_data>(Matched Data:+.+)\"\])\s\[severity" }
    }
    grok {
      match => { "alert_message" => "(?<agent>User-Agent: (.*)')'" }
    }
  }
}

```

Hình 24 Đoạn lệnh chuẩn hóa log modsecurity (P1)

```

    # Extract the Request URI
    grok {
      match => { "alert_message" => "(uri \"%{URIPATH:request_uri}\")" }
    }
    grok {
      match => { "alert_message" => "(?<ref>referer: (.*))" }
    }
    grok {
      match => { "ref" => "(?<referer> (.*))" }
    }
    mutate {
      # Remove unnecessary characters from the fields.
      gsub => [
        "alert_msg", "[\"]", "",
        "user_agent", "[;:]'", "", "",
        "user_agent", "\\\s*", "",
        "referer", "\\\\s*", ""
      ]
      # Remove the Unnecessary fields so we can only remain with
      # General message, rules_file, attack_type, rule_id, alert_msg, user_agent, hostname (being attacked), Request URI and Referer.
      remove_field => [ "alert_message", "rulesfile", "ruleid", "msg", "scanner", "agent", "ref" ]
    }
  }
}

```

Hình 25 Đoạn lệnh chuẩn hóa log modsecurity (P2)

- Đoạn lệnh chuẩn hóa log OSSEC:

```

GNU nano 6.2
filter {
  if "ossec" in [tags] {
    grok {
      match => { "message" => "%{SYSLOGTIMESTAMP:syslog_timestamp} %{SYSLOGHOST:syslog_host} %{DATA:syslog_program}: Alert Level: %{NONNEGINT:Alert_Level}; Rule: %{NONNEGINT:Rule} %"
      }
      add_field => [ "ossec_server", "%{host}" ]
    }
    mutate {
      remove_field => [ "message", "syslog_timestamp", "syslog_program", "syslog_host", "syslog_message", "syslog_pid", "@version", "type", "host" ]
    }
  }
}

```

Hình 26 Đoạn lệnh chuẩn hóa log OSSEC

- Đoạn lệnh chuẩn hóa log apache:

```

GNU nano 6.2                                         14-apache.conf *
filter {
  if "apache" in [tags] {
    grok {
      match => { "message" => ["%{IPORHOST:clientip} - %{DATA:username} \[%{HTTPDATE:http_date}\] \"%{WORD:method} %{DATA:path} HTTP/%{NUMBER:apache_http_version}\" %{NUMBER:code}" }
      "%{IPORHOST:clientip} - %{DATA:username} \[%{HTTPDATE:http_date}\]\\" -\"%{NUMBER:code} -" ]
    }
    mutate {
      rename => {
        "clientip" => "apache_remote_ip"
        "username" => "apache_user"
        "http_date" => "apache_access_time"
        "method" => "apache_method"
        "path" => "apache_path"
        "code" => "apache_code"
        "apache_http_version" => "apache_http_version"
        "sent_bytes" => "apache_sent_bytes"
        "referrer" => "apache_referrer"
        "agent" => "apache_agent"
      }
    }
  }
}

```

Hình 27 Đoạn lệnh chuẩn hóa log apache

- Đoạn lệnh tạo ra các index để hiển thị log:

```

GNU nano 6.2                                         40-outputs.conf
output {
  if "pfSense" in [tags] {
    elasticsearch {
      hosts => ["localhost"]
      index => "pfSense-%{+YYYY.MM.dd}"
      user => "elastic"
      password => "elastic"
    }
  }
  else if "ModSecurity" in [tags] {
    elasticsearch {
      hosts => ["localhost"]
      index => "modsecurity-%{+YYYY.MM.dd}"
      user => "elastic"
      password => "elastic"
    }
  }
  else if "ossec" in [tags] {
    elasticsearch {
      hosts => ["localhost"]
      index => "ossec-%{+YYYY.MM.dd}"
      user => "elastic"
      password => "elastic"
    }
  }
  else if "apache" in [tags] {
    elasticsearch {
      hosts => ["localhost"]
      index => "apache-%{+YYYY.MM.dd}"
      user => "elastic"
      password => "elastic"
    }
  }
}

```

Hình 28 Đoạn lệnh tạo ra các index

CHƯƠNG 5

THỰC NGHIỆM VÀ ĐÁNH GIÁ

Để thể hiện rõ hiệu quả của ELK stack, trong phạm vi đồ án lần này nhóm thực hiện triển khai 7 kịch bản dựa trên mô hình ở chương 3 bao gồm:

5.1 Kịch bản 1: Thu thập và chuẩn hóa log

Mô tả cơ bản: Tính năng này tập trung vào việc thu thập dữ liệu từ nhiều nguồn khác nhau và chuẩn hóa các loại log khác nhau để có thể xử lý chúng một cách hiệu quả trên ELK Stack. Các nguồn log có thể bao gồm hệ điều hành (OS), dịch vụ (như Apache), và các công cụ bảo mật (ví dụ như ModSecurity, PF, OSSEC). Việc chuẩn hóa log là quan trọng để đảm bảo rằng dữ liệu thu thập được có cấu trúc đồng nhất và dễ dàng xử lý.

Mô tả chi tiết:

1. Thu thập log từ nhiều nguồn

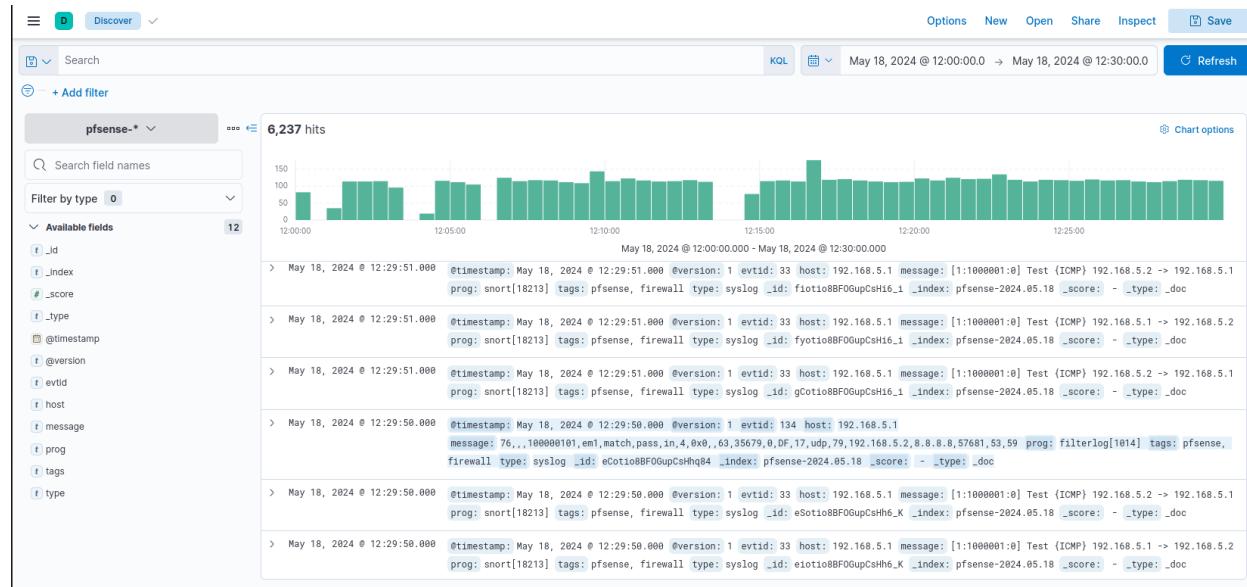
- **Hệ điều hành (OS):** Đối với các hệ điều hành như Linux, Windows, hoặc macOS, ELK Stack có thể sử dụng các agent như Beats hoặc Logstash để thu thập dữ liệu nhật ký từ các tập tin log hệ thống.
- **Dịch vụ (Service):** Các dịch vụ như Apache web server cũng tạo ra các file log để ghi lại các hoạt động của người dùng và các sự kiện của hệ thống. Logstash có thể được cấu hình để thu thập dữ liệu từ các tệp log này.
- **Công cụ bảo mật (Security):** Các công cụ bảo mật như ModSecurity, PF (Packet Filter), hoặc OSSEC (Open Source Host-based Intrusion Detection System) cung cấp các log về các sự kiện bảo mật và tấn công. ELK Stack có thể tích hợp với các công cụ này để thu thập thông tin về các trường hợp đáng ngờ hoặc các hành vi không bình thường.

2. Chuẩn hóa log

- Dữ liệu log từ các nguồn khác nhau thường có định dạng và cấu trúc khác nhau. Logstash có thể được sử dụng để chuẩn hóa dữ liệu log bằng cách ánh xạ các trường dữ liệu đến một cấu trúc chung.
- Sử dụng các bộ lọc và quy tắc để tách các trường dữ liệu cụ thể từ log và chuyển đổi chúng thành định dạng chuẩn mà Elasticsearch có thể hiểu.

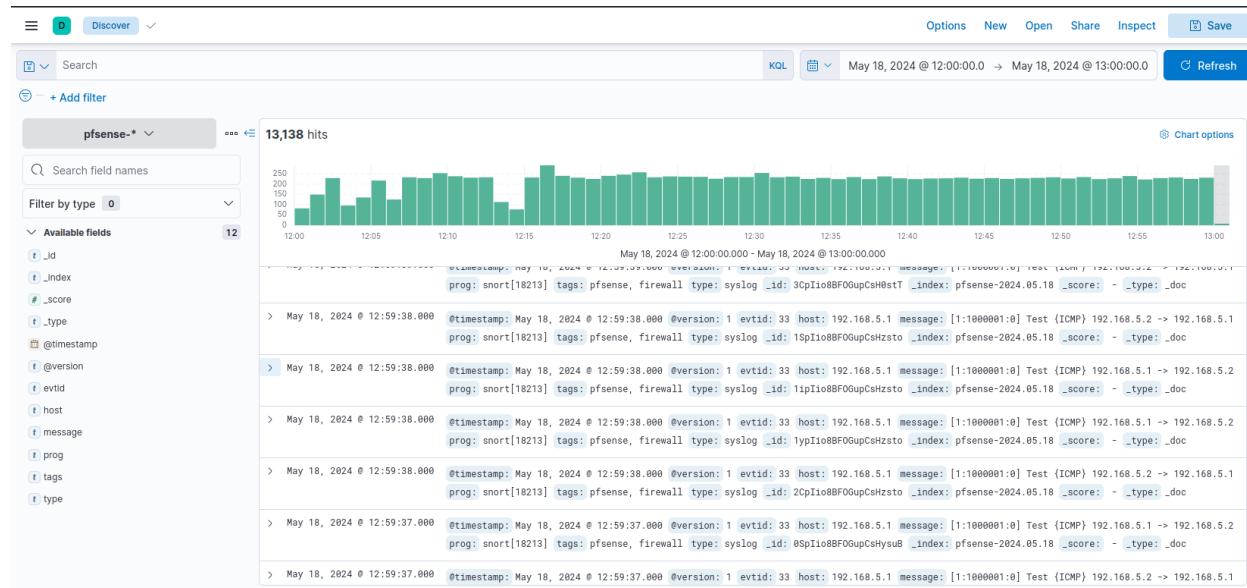
Kết quả Demo:

Vào index pfsense để kiểm tra Firewall

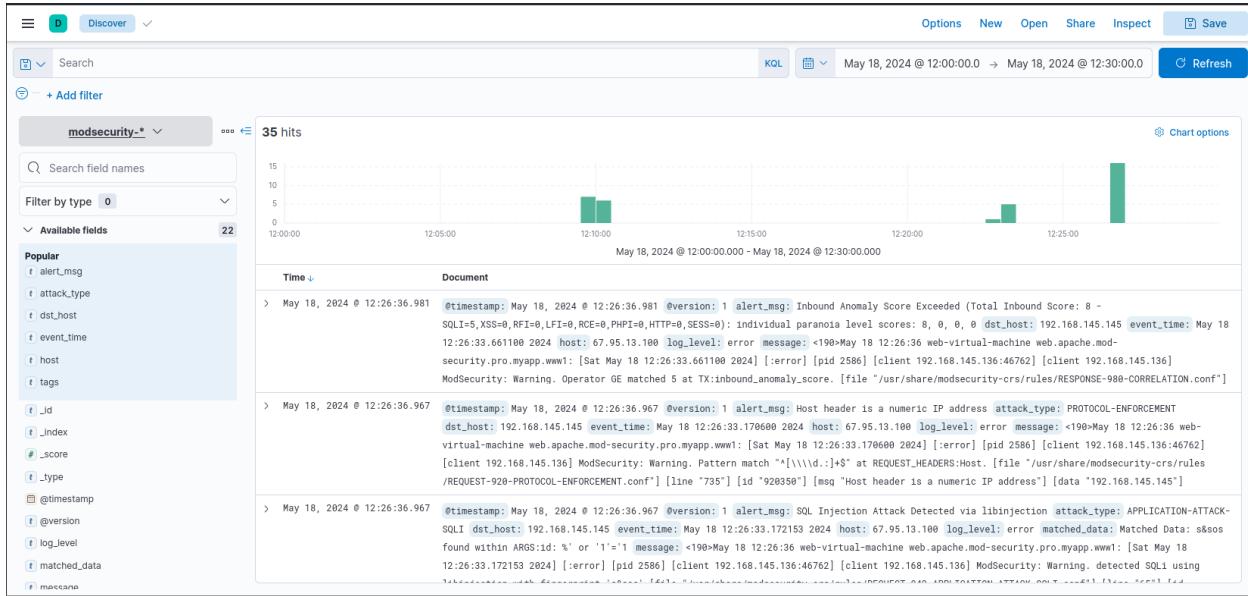


Hình 29 Kiểm tra log Firewall

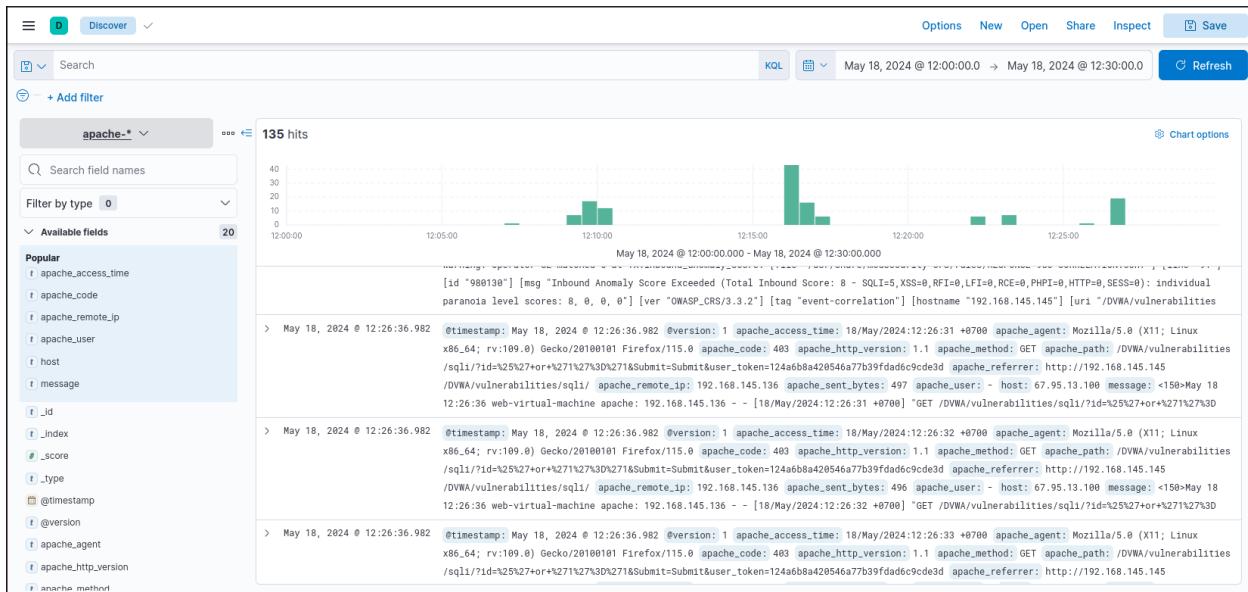
Vào index pfsense để kiểm tra Snort



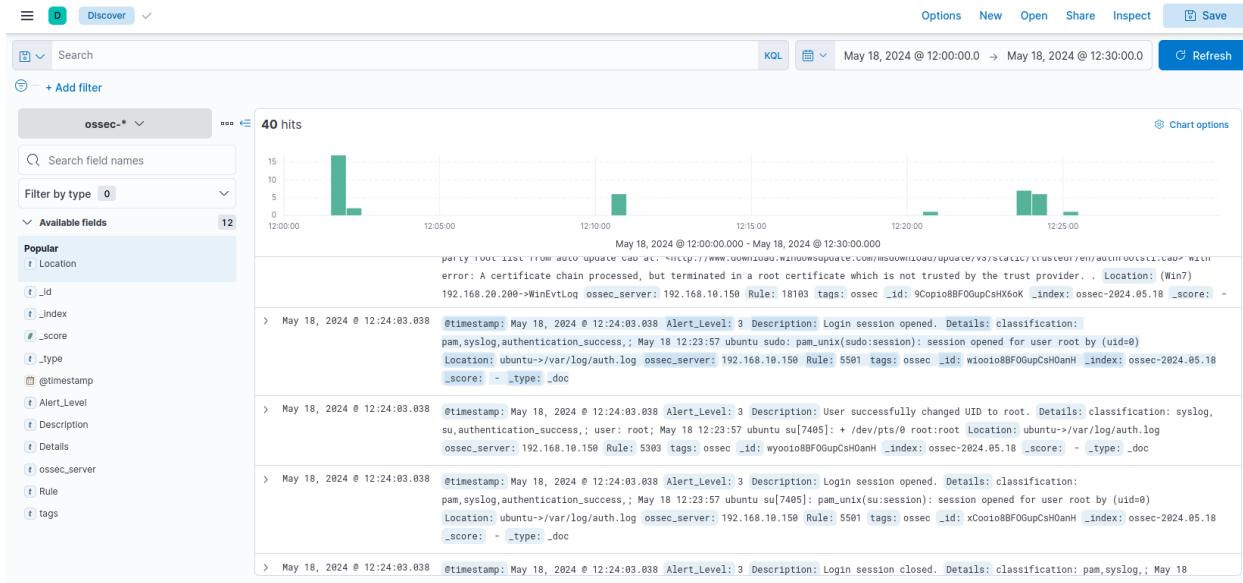
Hình 30 Kiểm tra log Snort



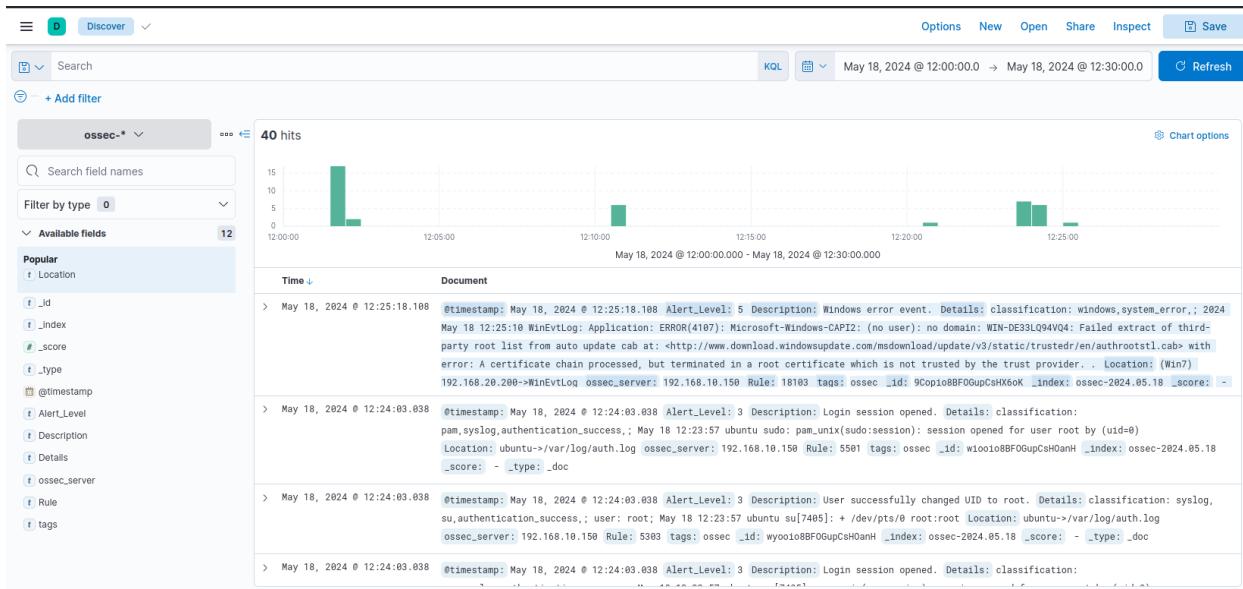
Hình 31 Kiểm tra log modsecurity



Hình 32 Kiểm tra log apache



Hình 33 Kiểm tra log Ossec server



Hình 34 Kiểm tra log ossec agent

5.2 Kịch bản 2: Quản lý và filter log

Mô tả cơ bản: Tính năng này tập trung vào việc sử dụng Logstash để lọc và xử lý dữ liệu log thu thập từ các nguồn khác nhau trước khi đưa vào Elasticsearch để lưu trữ và phân tích. Qua việc áp dụng các bộ lọc và quy tắc, Logstash giúp làm sạch và chuẩn hóa dữ

liệu log, loại bỏ thông tin không cần thiết, trích xuất thông tin quan trọng, và chuyển đổi định dạng dữ liệu.

Mô tả chi tiết:

Sau khi lọc, Logstash cung cấp các plugin để chuẩn hóa dữ liệu log, đảm bảo rằng chúng đều tuân thủ một định dạng chuẩn.

Các plugin có thể được sử dụng để chuyển đổi định dạng timestamp, địa chỉ IP, hoặc các trường dữ liệu khác để phù hợp với yêu cầu cụ thể của hệ thống hoặc ứng dụng.

Kết quả demo:

Name	Health	Status	Primaries	Replicas	Docs count	Storage size	Data stream
pfsense-2024.05.14	● yellow	open	1	1	6171	496.8kb	
pfsense-2024.05.18	● yellow	open	1	1	67163	5.6mb	
pfsense-2024.05.17	● yellow	open	1	1	3026	713.6kb	
pfsense-2024.05.16	● yellow	open	1	1	3488	674kb	
pfsense-2024.05.15	● yellow	open	1	1	12694	826.8kb	
ossec-2024.05.16	● yellow	open	1	1	2729	655.6kb	
ossec-2024.05.18	● yellow	open	1	1	71	100.9kb	
apache-2024.05.16	● yellow	open	1	1	51	88.3kb	
modsecurity-2024.05.17	● yellow	open	1	1	355	322kb	
modsecurity-2024.05.18	● yellow	open	1	1	44786	17.7mb	

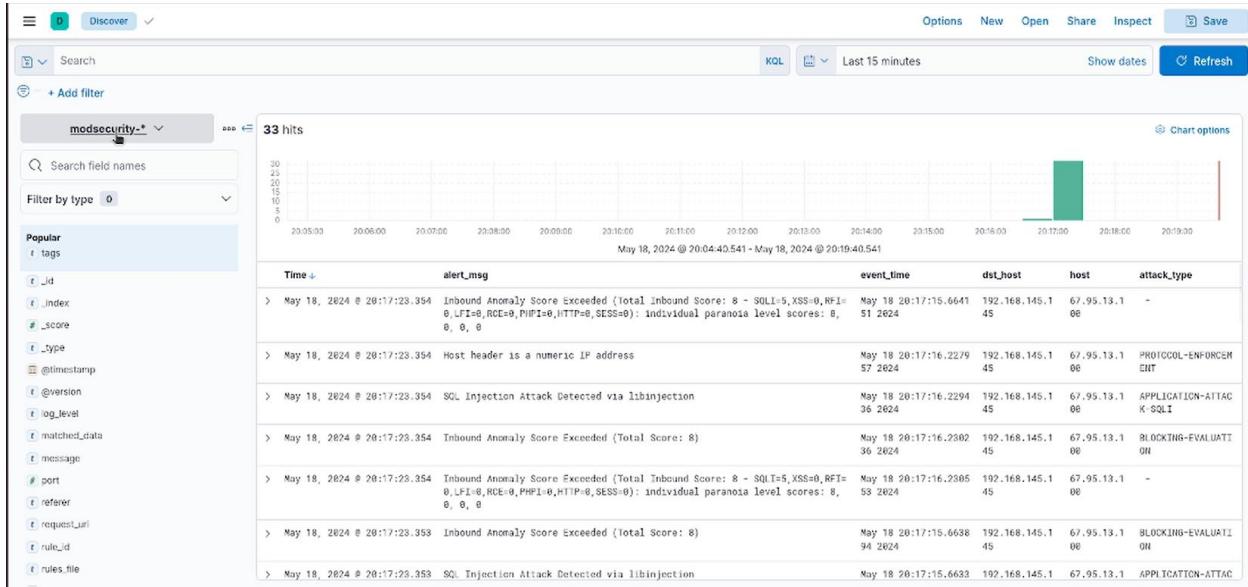
Hình 35 Quản lý log

The screenshot shows the Elasticsearch Settings page with the following index patterns listed:

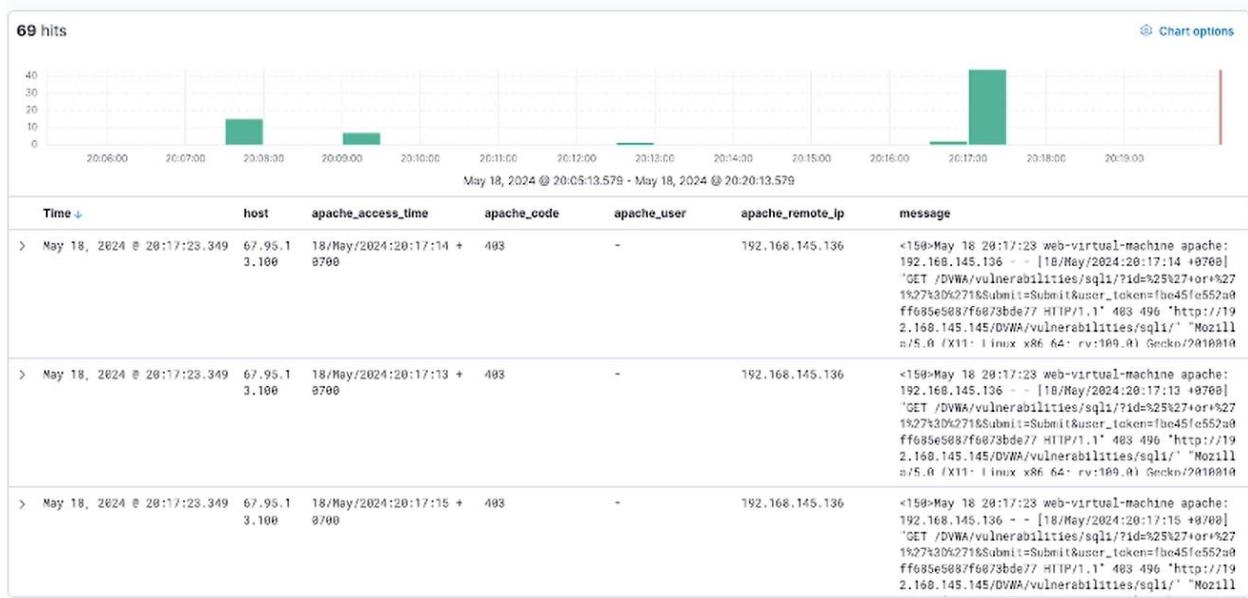
- pfsense-* (selected)
- apache-*
- kibana_sample_data_logs
- logs-*
- metrics-*
- modsecurity-*
- ossec-*
- traces-apm*,apm-* ,logs-apm*,apm-* ,metrics-apm*,apm-*

At the bottom, there are pagination controls: "Rows per page: 10" and page number "1".

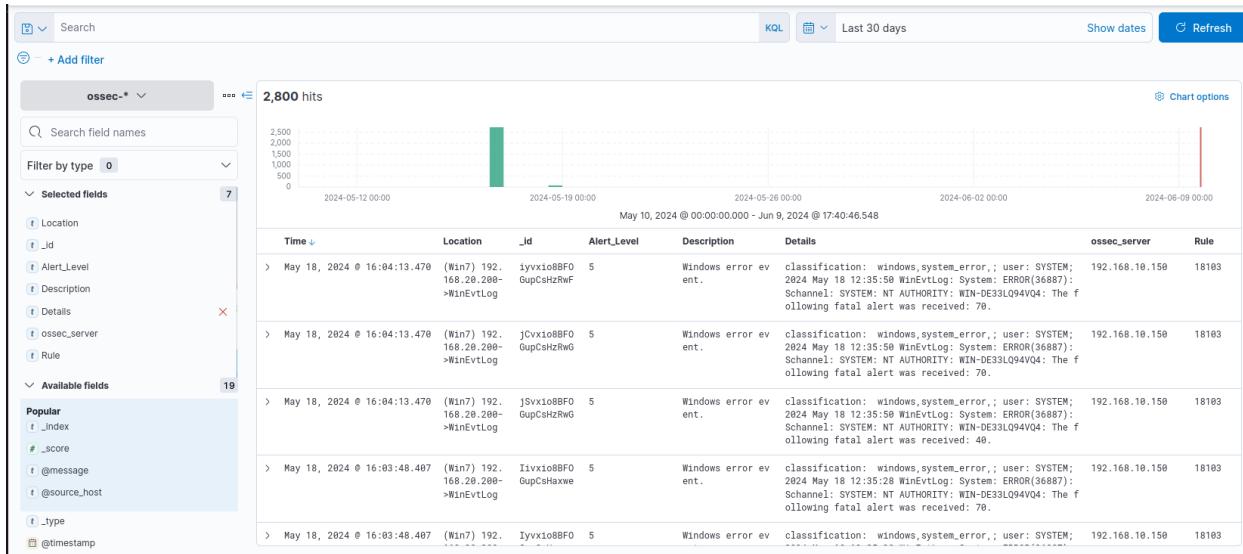
Hình 36 Quản lý index trên Kibana



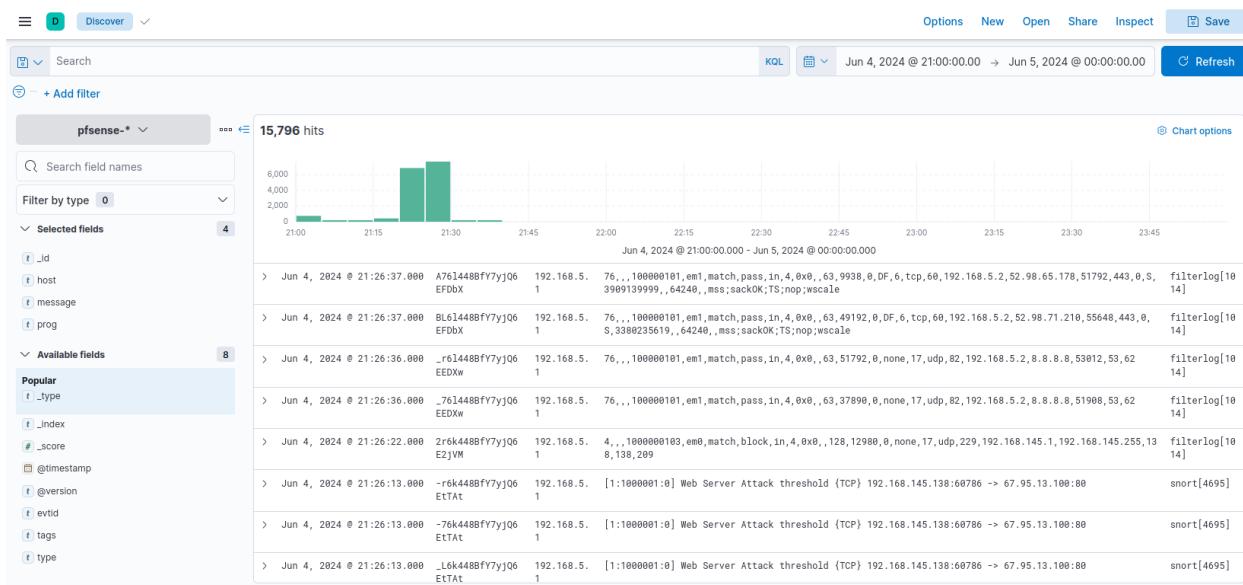
Hình 37 Filterlog modsecurity



Hình 38 Filterlog apache



Hình 39 Filter log ossec



Hình 40 Filter log pfSense

5.3 Kịch bản 3: Custom Dashboard Kibana

Mô tả cơ bản: Tính năng này tập trung vào việc tạo ra các bảng điều khiển (dashboard) tùy chỉnh và trực quan hóa dữ liệu log bằng cách sử dụng Kibana, một thành phần quan trọng của ELK Stack. Người dùng có thể tạo ra các bảng điều khiển theo nhu cầu cụ thể

của họ, bao gồm các biểu đồ, biểu đồ đường, biểu đồ cột, và các thành phần trực quan khác để hiển thị thông tin một cách rõ ràng và dễ hiểu.

Mô tả chi tiết:

1. Tạo Dashboard Tùy Chỉnh:

- Kibana cho phép người dùng tạo ra các bảng điều khiển tùy chỉnh dựa trên nhu cầu cụ thể của họ.
- Người dùng có thể chọn và sắp xếp các biểu đồ, đồ thị, và các thành phần trực quan khác trên dashboard để hiển thị thông tin một cách logic và hợp lý.

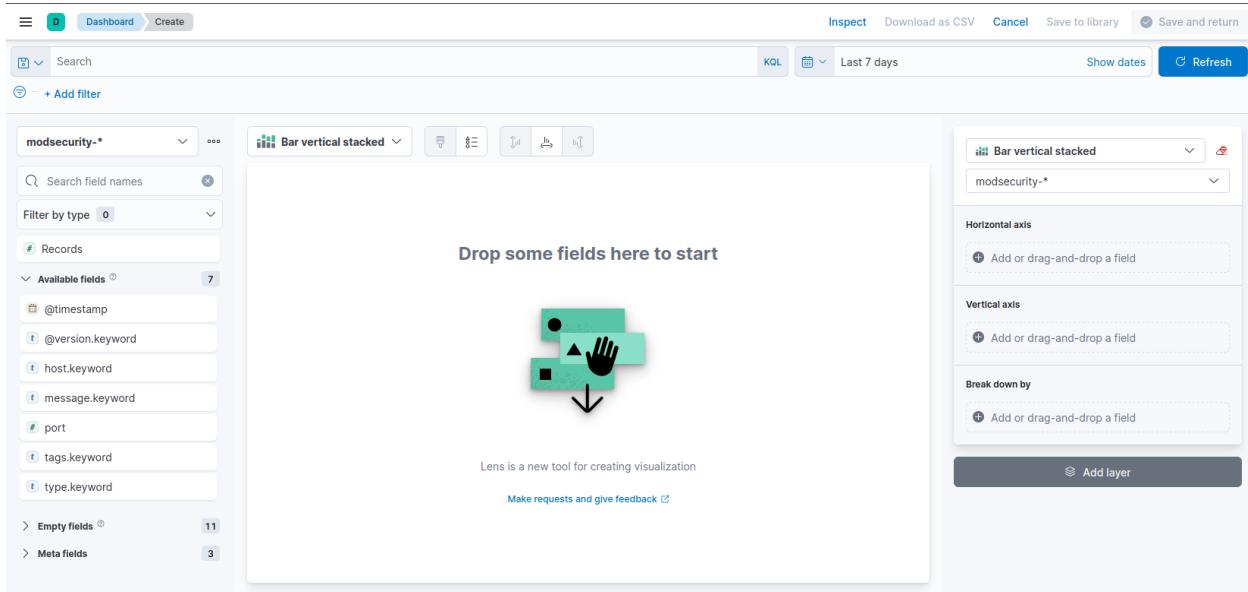
2. Trực Quan Hóa Dữ Liệu:

- Kibana cung cấp một loạt các loại biểu đồ và đồ thị để trực quan hóa dữ liệu log, bao gồm biểu đồ đường, biểu đồ cột, biểu đồ vùng, bản đồ, và nhiều loại biểu đồ khác.
- Người dùng có thể chọn loại biểu đồ phù hợp với dữ liệu của họ để hiển thị thông tin một cách hiệu quả nhất.

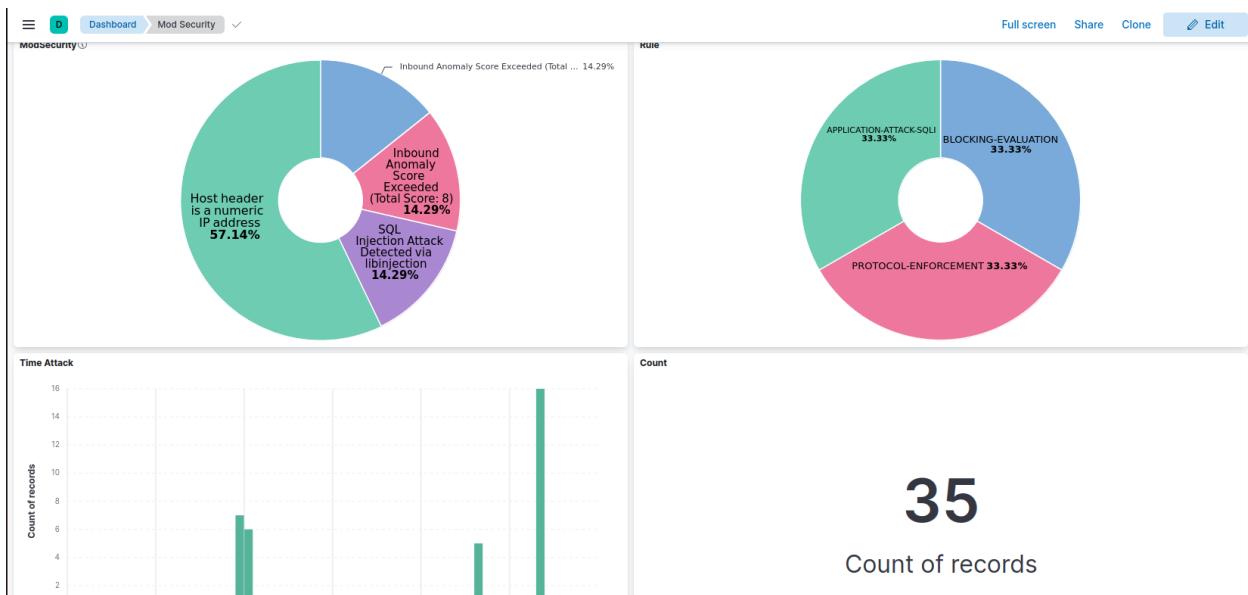
3. Tương Tác và Lọc Dữ Liệu:

- Bằng cách sử dụng các tính năng tương tác của Kibana, người dùng có thể tương tác với các biểu đồ và đồ thị trên dashboard để lọc dữ liệu hoặc chọn các phần tử cụ thể để hiển thị thông tin chi tiết.
- Các bộ lọc có thể được áp dụng trực tiếp trên các bảng điều khiển để hiển thị dữ liệu theo các tiêu chí nhất định.

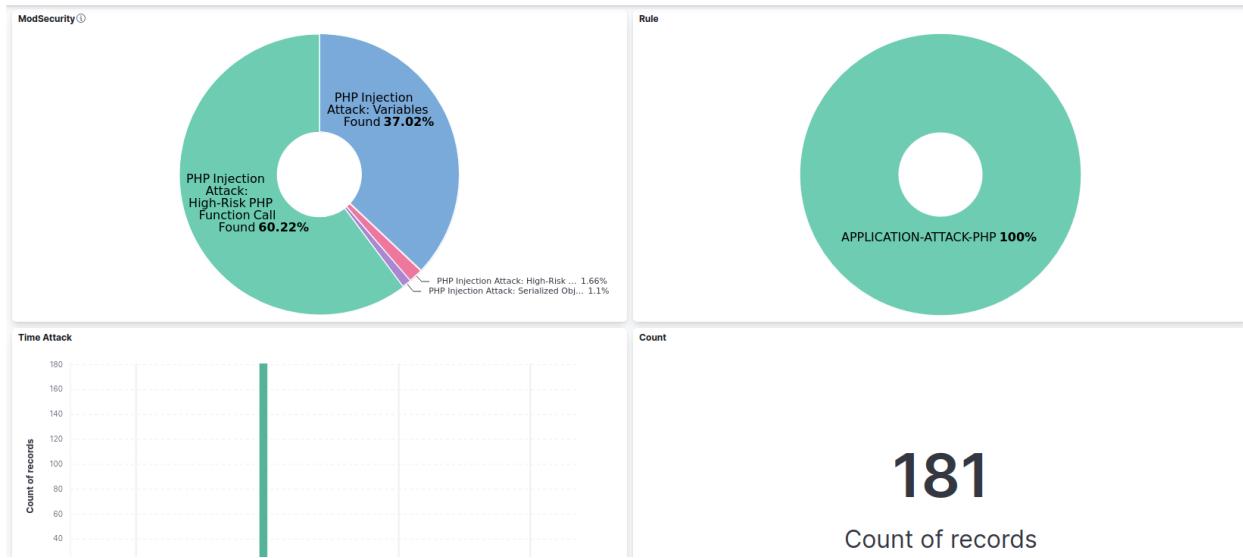
Demo:



Hình 41 Giao diện tạo Dashboard



Hình 42 Dashboard



Hình 43 Dashboard kết hợp với filter

5.4 Kịch bản 4: Machine Learning với ELK Stack

Mô tả cơ bản: Tính năng này tập trung vào việc tích hợp công nghệ Machine Learning vào ELK Stack để tự động phát hiện các xu hướng, biểu hiện bất thường và dự đoán các sự kiện trong dữ liệu log. Bằng cách sử dụng các thuật toán máy học, ELK Stack có thể tự động học và điều chỉnh mô hình để cải thiện khả năng phát hiện sự cố và cảnh báo trước khi chúng xảy ra.

Mô tả chi tiết:

1. Phát hiện bất thường:

- Công nghệ Machine Learning được sử dụng để xây dựng mô hình dự đoán hành vi bình thường của hệ thống dựa trên dữ liệu log lịch sử.
- Khi có dữ liệu mới được nhập vào, các mô hình này sẽ tự động phân tích và so sánh với hành vi bình thường để phát hiện ra các biểu hiện bất thường và cảnh báo người quản trị.

2. Dự đoán xu hướng và sự kiện:

- Các thuật toán Machine Learning có thể được sử dụng để dự đoán các xu hướng tương lai dựa trên dữ liệu lịch sử.

- ELK Stack có thể sử dụng các mô hình dự đoán để cảnh báo về các sự kiện tiềm ẩn hoặc xu hướng đột phá trong hệ thống.

Kết quả demo:

Job settings Job config Datafeed Counts JSON Job messages Datafeed preview Forecasts Annotations Model snapshots

General

job_id	kibana-logs-ui-default-default-log-entry-categories-count
job_type	anomaly_detector
job_version	7.17.21
create_time	2024-05-17 23:12:19
model_snapshot_id	1717943129
groups	logs-ui
description	Logs UI: Detects anomalies in count of log entries by category
model_snapshot_retention_days	10
daily_model_snapshot_retention_after_days	1
results_index_name	custom-kibana-logs-ui-default-default-log-entry-categories-count
allow_lazy_open	
state	opened
assignment_explanation	
open_time	518088s

Custom settings

created_by	ml-module-logs-ui-categories
job_revision	2
logs_source_config	{"indexPattern":"kibana_sample_data_logs**","timestampField":"@timestamp","bucketSpan":900000,"datasetFilter":[{"type":"includeAll"}]}

Node

name	node-1
------	--------

Hình 44 Phát hiện các bất thường trong số lượng mục nhập

Job settings Job config Datafeed Counts JSON Job messages Datafeed preview Forecasts Annotations Model snapshots

General

job_id	kibana-logs-ui-default-default-log-entry-rate
job_type	anomaly_detector
job_version	7.17.21
create_time	2024-05-17 23:11:53
model_snapshot_id	1717943146
groups	logs-ui
description	Logs UI: Detects anomalies in the log entry ingestion rate
model_snapshot_retention_days	10
daily_model_snapshot_retention_after_days	1
results_index_name	custom-kibana-logs-ui-default-default-log-entry-rate
allow_lazy_open	
state	opened
assignment_explanation	
open_time	518116s

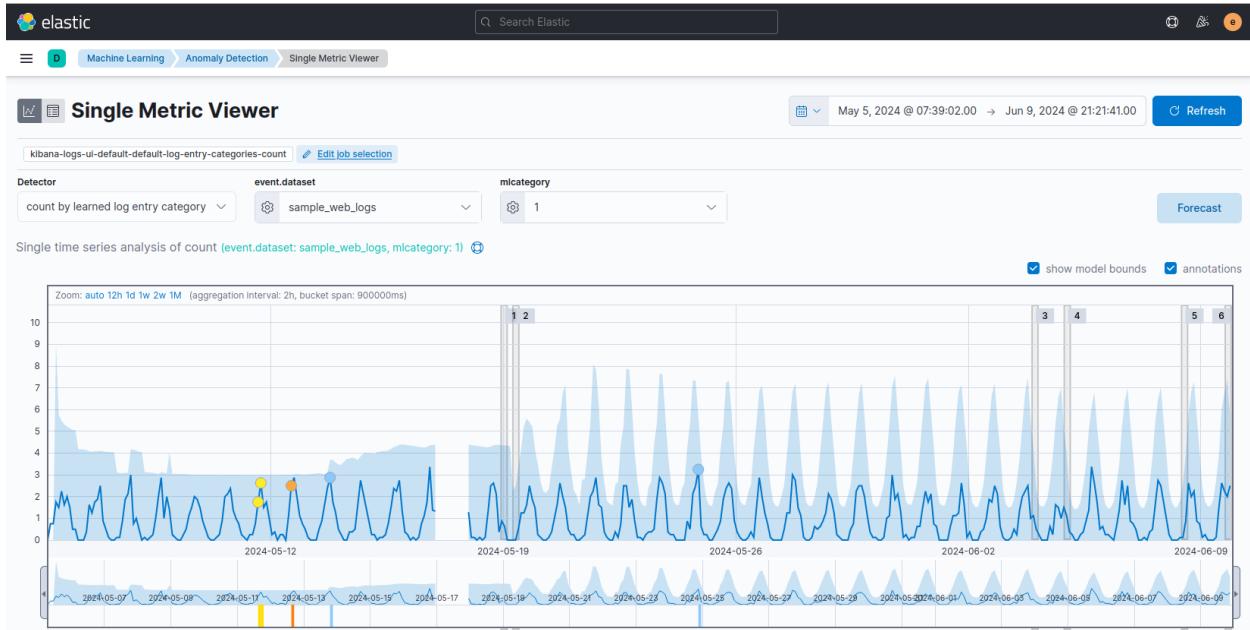
Custom settings

created_by	ml-module-logs-ui-analysis
job_revision	2
logs_source_config	{"indexPattern":"kibana_sample_data_logs**","timestampField":"@timestamp","bucketSpan":900000}

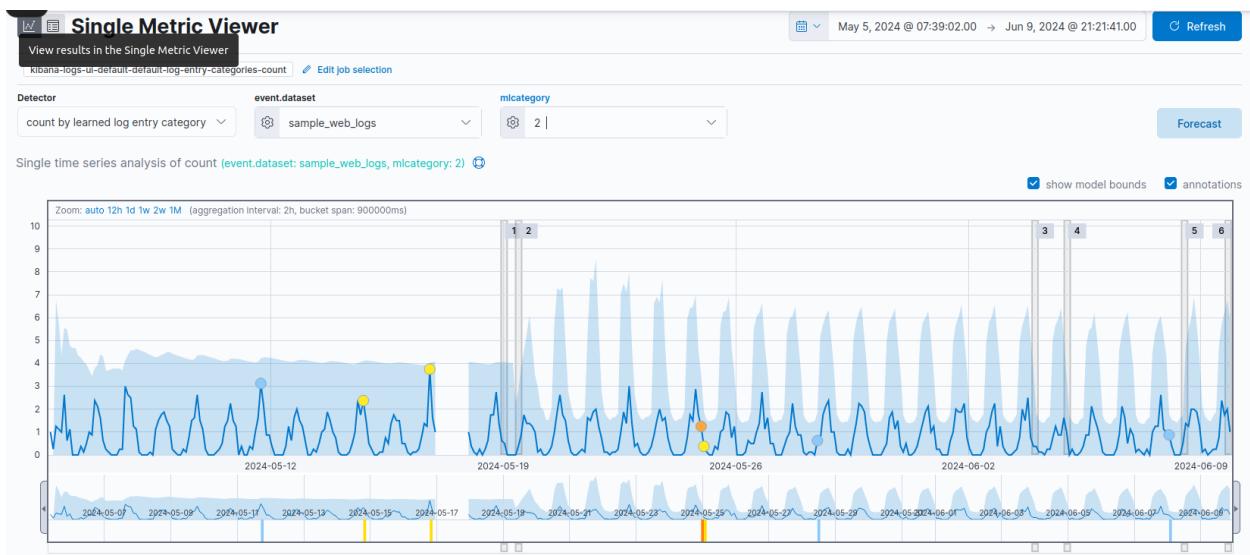
Node

name	node-1
------	--------

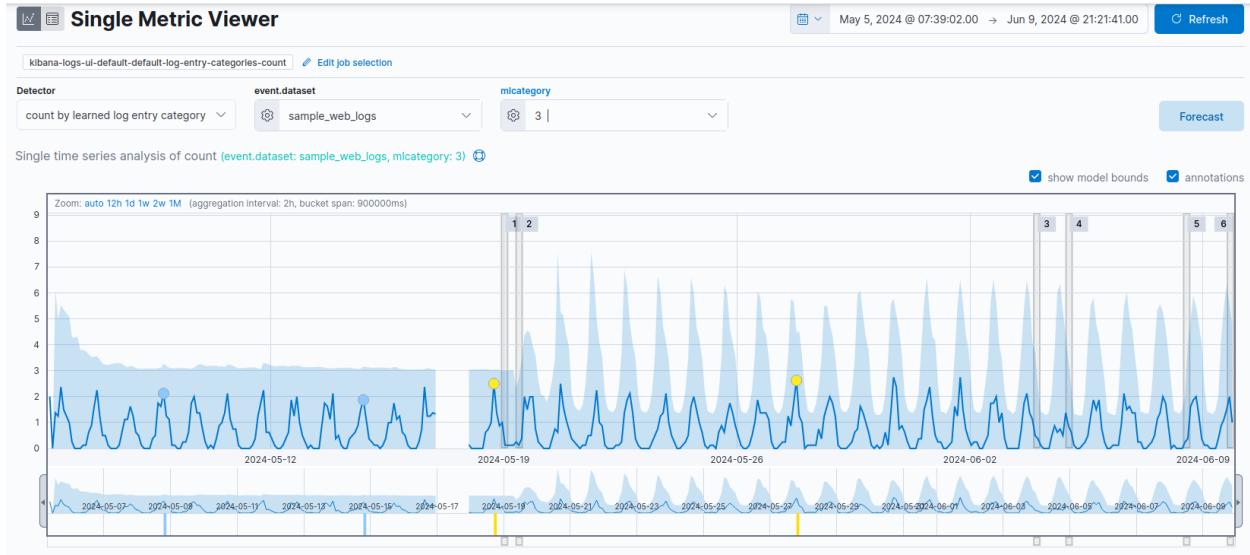
Hình 45 Phát hiện các bất thường trên tần suất nhập log



Hình 46 Biểu đồ số lượng Log với nhãn là “1”



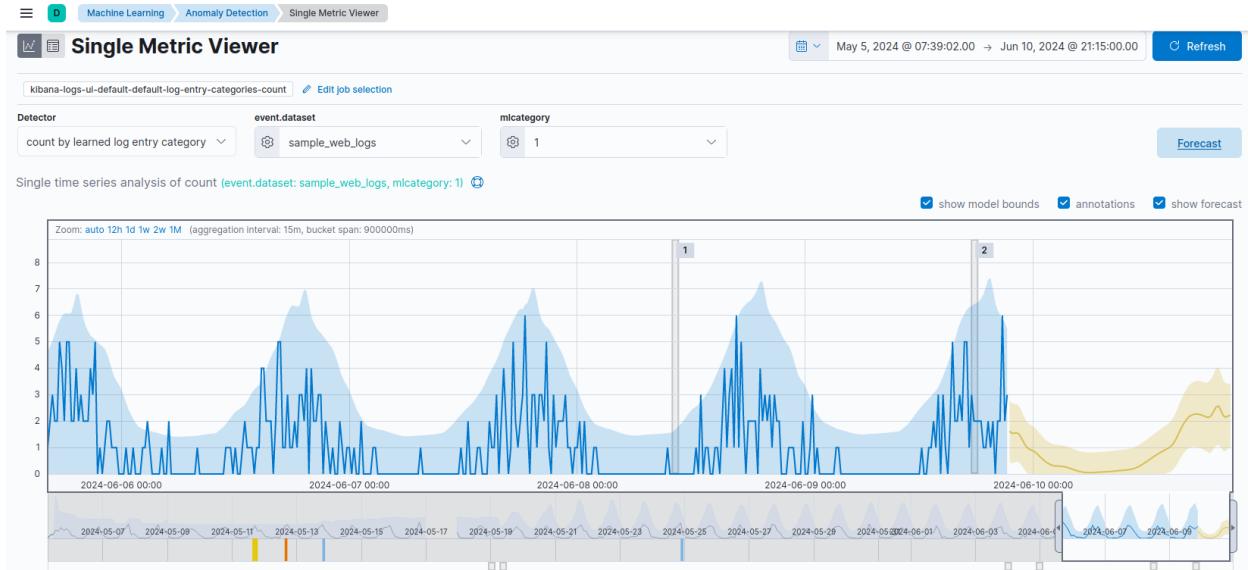
Hình 47 Biểu đồ số lượng Log với nhãn là “2”



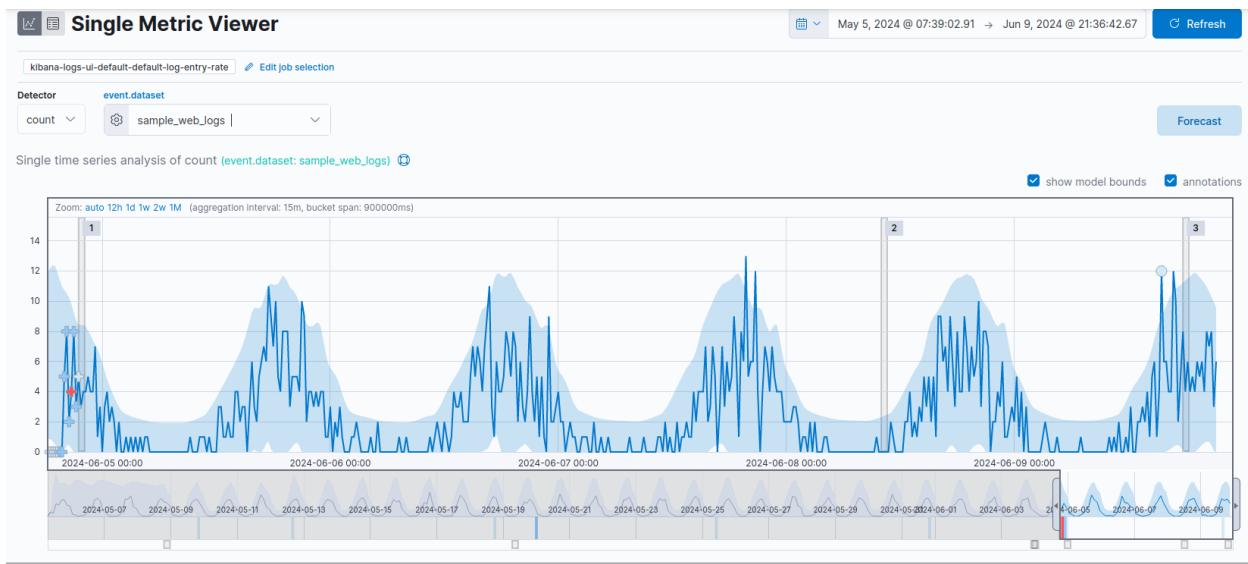
Hình 48 Biểu đồ số lượng Log với nhãn là “3”

> Annotations Total: 6										
Anomalies										
Severity	warning	Interval	Auto	①	Time	Severity ② ↓	Detector	Found for	Influenced by	Actual ③
May 12th 2024	69	count by learned log entry category	mcategory 1 ④ ⑤	event.dataset: sample_web_logs mcategory: 1	9	0.998	↑ 9x higher	223.876.0.27 - - [2018-07-22T00... ⑥		
May 11th 2024	39	count by learned log entry category	mcategory 1 ④ ⑤	event.dataset: sample_web_logs mcategory: 1	7	1	↑ 7x higher	223.876.0.27 - - [2018-07-22T00... ⑥		
May 24th 2024	9	count by learned log entry category	mcategory 1 ④ ⑤	event.dataset: sample_web_logs mcategory: 1	6	1.76	↑ 3x higher	223.876.0.27 - - [2018-07-22T00... ⑥		
May 13th 2024	4	count by learned log entry category	mcategory 1 ④ ⑤	event.dataset: sample_web_logs mcategory: 1	8	0.822	↑ 10x higher	223.876.0.27 - - [2018-07-22T00... ⑥		

Hình 49 Phát hiện bất thường



Hình 50 Dự đoán số lượng log của nhãn “1” trong khoảng thời gian 1 ngày



Hình 51 Biểu đồ tần suất nhập Log



Hình 52 Dự đoán về tần suất nhập log trong vòng 1 ngày

Anomalies									
Severity	warning	Interval	Auto	Detected	Influenced by	Actual	Typical	Description	Actions
>	June 4th 2024	97	count	sample_web_logs	event.dataset: sample_web_logs ⊕ ⊖	4	5.01	↓ 1.3x lower	⚙️
>	May 19th 2024	5	count	sample_web_logs	event.dataset: sample_web_logs ⊕ ⊖	5	1.82	↑ 3x higher	⚙️
>	May 12th 2024	1	count	sample_web_logs	event.dataset: sample_web_logs ⊕ ⊖	12	3.52	↑ 3x higher	⚙️
>	May 18th 2024	< 1	count	sample_web_logs	event.dataset: sample_web_logs ⊕ ⊖	17	7.01	↑ 2x higher	⚙️
>	May 31st 2024	< 1	count	sample_web_logs	event.dataset: sample_web_logs ⊕ ⊖	16	6.4	↑ 3x higher	⚙️
>	May 25th 2024	< 1	count	sample_web_logs	event.dataset: sample_web_logs ⊕ ⊖	4	0.451	↑ 9x higher	⚙️
>	May 9th 2024	< 1	count	sample_web_logs	event.dataset: sample_web_logs ⊕ ⊖	13	4.57	↑ 3x higher	⚙️
>	June 9th 2024	< 1	count	sample_web_logs	event.dataset: sample_web_logs ⊕ ⊖	12	4.36	↑ 3x higher	⚙️

Hình 53 Phát hiện bất thường

5.5 Kịch bản 5. Security Alert

Mô tả cơ bản: Tính năng này tập trung vào việc cài đặt các cảnh báo để phát hiện các sự kiện bất thường trong dữ liệu log và tự động gửi thông báo qua email cho người quản trị hoặc nhóm phụ trách cách nhau 1 giờ.

Mô tả chi tiết:

1. Cài đặt Cảnh báo Bất Thường:

- Sử dụng Logstash hoặc Kibana để cài đặt các cảnh báo để phát hiện các biểu hiện bất thường trong dữ liệu log.
- Các cảnh báo có thể được xác định dựa trên các điều kiện nhất định, như số lượng lớn hành vi đăng nhập không hợp lệ, lưu lượng mạng không thường xuyên, hoặc các sự kiện bảo mật quan trọng.

2. Gửi Thông Báo Email:

- Khi một cảnh báo được kích hoạt, Logstash hoặc Kibana có thể được cấu hình để tự động gửi thông báo qua Microsoft Teams cho người quản trị hoặc nhóm phụ trách.
- Thông báo email có thể chứa số lượng alert được tạo ra trong vòng 1 giờ

Kết quả demo:

The screenshot shows the 'Rule type' configuration page for a 'Threshold' rule. The 'Selected' option is highlighted under 'Rule type'. The 'Index patterns' section contains the pattern 'modsecurity-*'. The 'Custom query' section includes a dropdown set to '@timestamp : *' and a 'KQL' button. The 'Group by' section has dropdowns for 'All results' and 'Count'. The 'Threshold' section shows a value of '1' with an operator ' \geq '. The 'Unique values' section has a dropdown. The 'Timeline template' section is set to 'None'.

Hình 54 Set rule

Actions

Actions frequency

Hourly

Select when automated actions should be performed if a rule evaluates as true.

Actions

Send Alert via Teasm

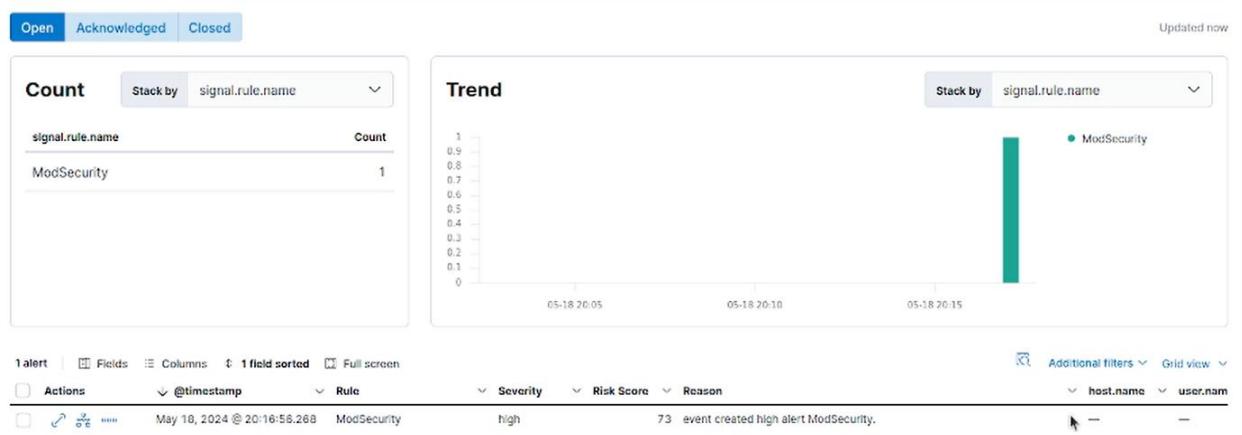
Microsoft Teams connector Add connector

Send Alert via Teasm

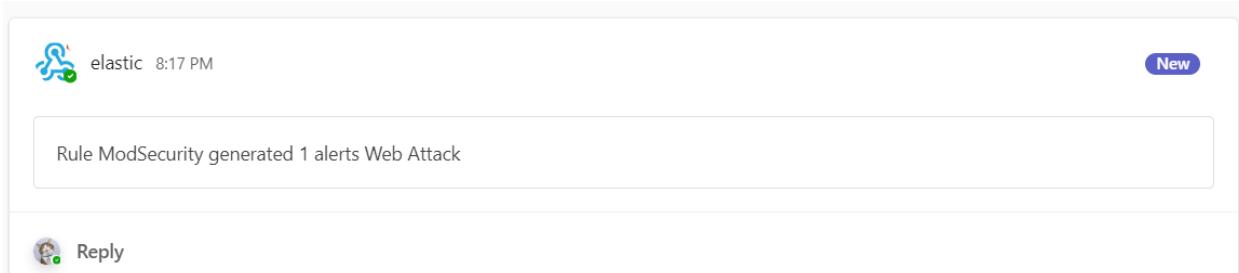
Message

Rule {{context.rule.name}} generated {{state.signals_count}} alerts
Web Attack

Hình 55 Set action



Hình 56 Kiểm tra security alert



Hình 57 Check mail

5.6 Kích bản 6. Alert bất thường bằng Kibana

Mô tả cơ bản: Tính năng này tập trung vào việc cài đặt các cảnh báo để phát hiện các sự kiện bất thường trong dữ liệu log và tự động gửi thông báo qua Microsoft Teams cho người quản trị hoặc nhóm phụ trách. Cụ thể, nhóm triển khai Alert khi tấn công dịch vụ Web.

Mô tả chi tiết:

1. Cài đặt Cảnh báo Bất Thường:

- Sử dụng Logstash hoặc Kibana để cài đặt các cảnh báo để phát hiện các biểu hiện bất thường trong dữ liệu log.
- Các cảnh báo có thể được xác định dựa trên các điều kiện nhất định, như số lượng lớn hành vi đăng nhập không hợp lệ, lưu lượng mạng không thường xuyên, hoặc các sự kiện bảo mật quan trọng.

2. Gửi Thông Báo Email:

- Khi một cảnh báo được kích hoạt, Logstash hoặc Kibana có thể được cấu hình để tự động gửi thông báo qua MicrosoftTeams cho người quản trị hoặc nhóm phụ trách.
- Thông báo email có thể chứa thông tin chi tiết về sự kiện bất thường, bao gồm các tên alert, giá trị nhận được, mô tả chi tiết và thời gian timestamp

Kết quả demo:

X

Edit rule

Check every [?](#)

1



minute [▼](#)

Notify [?](#)

Only on status change



Index threshold

Alert when an aggregated query meets the threshold. [Documentation](#) ↗

Select an index

INDEX modsecurity-*

WHEN count()

OVER all documents

Define the condition

IS ABOVE 1

FOR THE LAST 1 minute

Hình 58 Set rule

Actions

The screenshot shows the configuration of an action step. At the top, there is a dropdown menu with the option "Send Alert via Teasm". Below it, a "Run when" dropdown is set to "Threshold met". Under the "Microsoft Teams connector" section, the action name "Send Alert via Teasm" is selected. In the "Message" section, the message template is defined as:

```
alert '{{alertName}}' is active for group '{{context.group}}':  
- Value: {{context.value}}  
- Conditions Met: {{context.conditions}} over {{params.timeWindowSize}}  
{{params.timeWindowUnit}}  
- Timestamp: {{context.date}}
```

Select a connector type

Hình 59 Set Action

The screenshot shows a Microsoft Teams message from a user named "elastic" at 8:17 PM. The message displays an alert summary:

```
alert 'Test' is active for group 'all documents':  
• Value: 9  
• Conditions Met: count is greater than 1 over 1m  
• Timestamp: 2024-05-18T13:17:21.490Z
```

Below the message, there is a "Reply" button.

Hình 60 Check mail

5.7. Kịch bản 7. Phát hiện outbound attack thông qua phân tích log bằng ELK Stack

Mô tả cơ bản: Sử dụng ELK Stack để thu thập Log từ nhiều nguồn như Firewall, NIDPS (Snort), Apache, ModSecurity. Từ đó sẽ phân tích và xuất hiện cảnh báo Alert qua Microsoft Teams khi số lượng Log tăng lên quá mức threshold. Attacker sử dụng tools nmap để tiến hành quét mạng và nikto để truyền các payload tấn công vào web application.

Mô tả chi tiết:

1. Cài đặt Cảnh báo Bất Thường:

- Sử dụng Logstash hoặc Kibana để cài đặt các cảnh báo để phát hiện các biểu hiện bất thường trong dữ liệu log.
- Các cảnh báo có thể được xác định dựa trên các điều kiện nhất định, như số lượng lớn hành vi đăng nhập không hợp lệ, lưu lượng mạng không thường xuyên, hoặc các sự kiện bảo mật quan trọng.

2. Gửi Thông Báo Email:

- Khi một cảnh báo được kích hoạt, Logstash hoặc Kibana có thể được cấu hình để tự động gửi thông báo qua MicrosoftTeams cho người quản trị hoặc nhóm phụ trách.
- Thông báo email có thể chứa thông tin chi tiết về sự kiện bất thường, bao gồm các tên alert, giá trị nhận được, mô tả chi tiết và thời gian timestamp

3. Tấn công mạng:

- Attacker sử dụng công cụ nmap để quét toàn vùng mạng để kiểm tra có port web nào đang mở hay không
- Sau khi xác định được mục tiêu đang mở port web HTTP 80 thì Attacker tiến hành sử dụng công cụ nikto để đưa các payload độc hại vào web để có thể gây hại cho trang web của hệ thống. Từ đó phục vụ mục đích của Attacker

Kết quả demo:

- Rule của modsecurity tương tự như kịch bản trên.

```
(kali㉿kali)-[~]
$ nmap -v -A 192.168.145.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-18 09:56 EDT
NSE: Loaded 156 scripts for scanning. Kali Forums Kali NetHunter Exploit-DB Goo
NSE: Script Pre-scanning.
Initiating NSE at 09:56
Completed NSE at 09:56, 0.00s elapsed
Initiating NSE at 09:56
Completed NSE at 09:56, 0.00s elapsed
Initiating NSE at 09:56 [connection to access this resource.
Completed NSE at 09:56, 0.00s elapsed
Initiating Ping Scan at 09:56
Scanning 256 hosts [2 ports/host]
Completed Ping Scan at 09:56, 2.41s elapsed (256 total hosts)
Initiating Parallel DNS resolution of 3 hosts. at 09:56
Completed Parallel DNS resolution of 3 hosts. at 09:56, 0.07s elapsed
Nmap scan report for 192.168.145.0 [host down]
Nmap scan report for 192.168.145.1 [host down]
Nmap scan report for 192.168.145.3 [host down]
Nmap scan report for 192.168.145.4 [host down]
Nmap scan report for 192.168.145.5 [host down]
Nmap scan report for 192.168.145.6 [host down]
Nmap scan report for 192.168.145.7 [host down]
Nmap scan report for 192.168.145.8 [host down]
Nmap scan report for 192.168.145.9 [host down]
```

Hình 61 Quét nmap

```

Nmap scan report for 192.168.145.136
Host is up (0.0014s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 9.6p1 Debian 4 (protocol 2.0)
| ssh-hostkey:
|   256 97:21:9c:92:a5:11:1f:9b:38:ad:d1:d9:07:df:b0:61 (ECDSA)
|_  256 14:16:17:0b:64:e3:fa:b8:62:75:7c:69:01:3d:44:6d (ED25519)
80/tcp    open  http     Apache httpd 2.4.58 ((Ubuntu))
| http-methods:
|_ Supported Methods: HEAD GET POST OPTIONS
|_http-title: Apache2 Debian Default Page: It works
|_http-server-header: Apache/2.4.58 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 192.168.145.145
Host is up (0.0028s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
80/tcp    open  http     Apache httpd 2.4.52 ((Ubuntu))
| http-server-header: Apache/2.4.52 (Ubuntu)
|_http-title: 403 Forbidden

NSE: Script Post-scanning.
Initiating NSE at 09:59
Completed NSE at 09:59, 0.00s elapsed
Initiating NSE at 09:59
Completed NSE at 09:59, 0.00s elapsed
Initiating NSE at 09:59
Completed NSE at 09:59, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 256 IP addresses (3 hosts up) scanned in 179.77 seconds

```

Hình 62 Kết quả quét nmap

```

[kali㉿kali]:~$ nikto -h 192.168.145.145
- Nikto v2.5.0
+ Target IP:      192.168.145.145
+ Target Hostname: 192.168.145.145
+ Target Port:    80
+ Start Time:    2024-05-18 09:59:45 (GMT-4)
+ Server: Apache/2.4.52 (Ubuntu)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The Content-Security-Policy header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-security-policy/
+ No CGI Directories found (use --C all to force check all possible dirs)
+ Apache/2.4.52 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ /: Server may leak inodes via Etags, header found with file /, inode: 29af, size: 6183a0bd218d5, mtime: gzip. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1418
+ OPTIONS: Allowed HTTP Methods: GET, POST, OPTIONS, HEAD .
+ 8874 requests: 0 errors(s) and 5 item(s) reported by remote host
+ End Time:    2024-05-18 10:00:48 (GMT-4) (55 seconds)

+ 1 host(s) tested

```

Hình 63 Kết quả tấn công web

X

Edit rule

Check every [?](#)

1



minute [▼](#)

Notify [?](#)

Only on status change [▼](#)

Index threshold

Alert when an aggregated query meets the threshold. [Documentation](#)

Select an index

INDEX pfsense-*

WHEN count()

OVER all documents

Define the condition

IS ABOVE 500

FOR THE LAST 1 minute

Hình 64 Set rule pfsense

Actions

The screenshot shows a configuration screen for a Zapier action. At the top, there is a header with a dropdown arrow, the text "Send Alert via Teasm", and a red delete icon. Below this is a "Run when" section with a dropdown menu set to "Threshold met". Underneath is a "Microsoft Teams connector" section with a "Send Alert via Teasm" action selected. A "Message" field contains a template for the alert message, which includes context variables like {{alertName}}, {{context.value}}, {{context.conditions}}, {{params.timeWindowSize}}, {{params.timeWindowUnit}}, and {{context.date}}.

```
alert '{{alertName}}' is active for group '{{context.group}}':  
- Value: {{context.value}}  
- Conditions Met: {{context.conditions}} over {{params.timeWindowSize}}  
{{params.timeWindowUnit}}  
- Timestamp: {{context.date}}
```

Hình 65 Set action pfSense

X

Edit rule

Check every ?

1



minute ▼

Notify ?

Only on status change



Index threshold

Alert when an aggregated query meets the threshold. [Documentation](#) ↗

Select an index

INDEX apache-*

WHEN count()

OVER all documents

Define the condition

IS ABOVE 1000

FOR THE LAST 1 minute

Hình 66 Set rule apache

Actions

The screenshot shows the configuration of an alert action. At the top, there is a dropdown menu with the option "Send Alert via Teasm". Below it, a "Run when" dropdown is set to "Threshold met". Under the "Microsoft Teams connector" section, the action name "Send Alert via Teasm" is selected. In the "Message" box, the following template is defined:

```
alert '{{alertName}}' is active for group '{{context.group}}':  
- Value: {{context.value}}  
- Conditions Met: {{context.conditions}} over {{params.timeWindowSize}}  
{{params.timeWindowUnit}}  
- Timestamp: {{context.date}}
```

Hinh 67 Set action apache

The screenshot shows a Microsoft Teams message from the user "elastic" at 8:57 PM. The message content is:

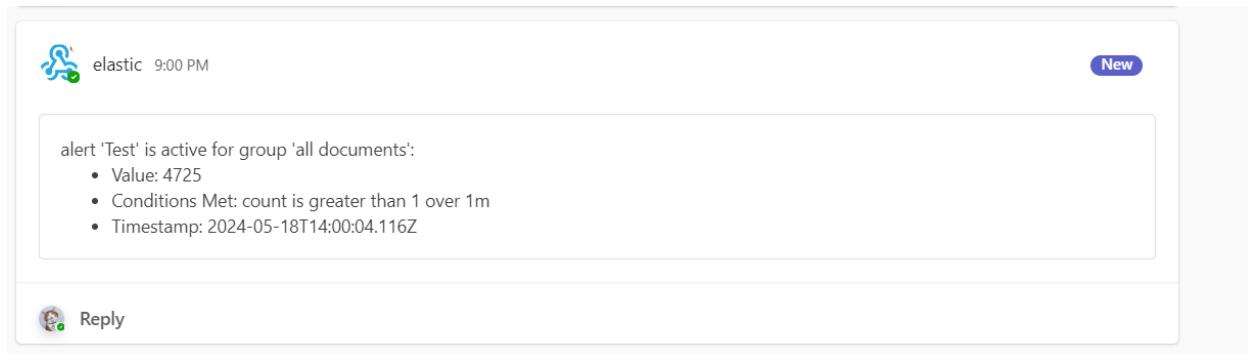
```
alert 'Alertpfsense' is active for group 'all documents':

- Value: 899
- Conditions Met: count is greater than 500 over 1m
- Timestamp: 2024-05-18T13:57:25.032Z

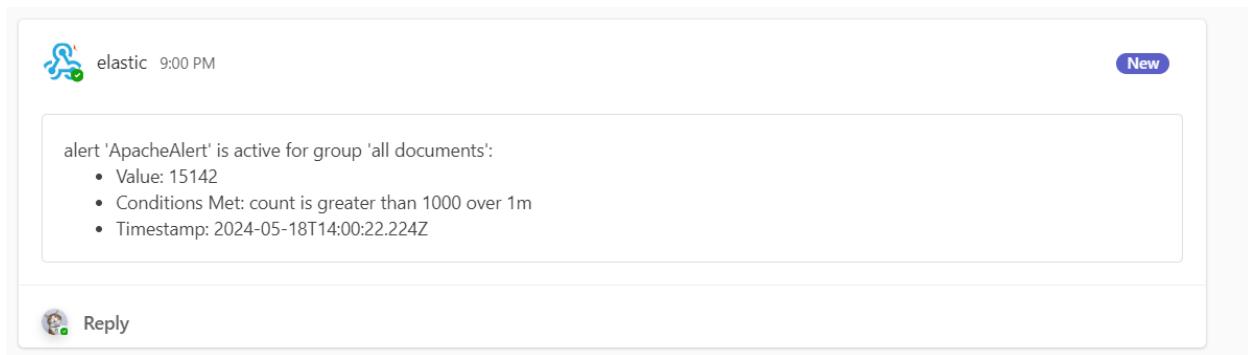
```

Below the message, there is a "Reply" button.

Hinh 68 Check mail alert pfsense



Hình 69 Check mail alert modsecurity



Hình 70 Check mail alert apache

CHƯƠNG 6

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1 Kết luận

ELK Stack (Elasticsearch, Logstash, Kibana) là một bộ công cụ quan trọng trong việc thu thập, phân tích và trực quan hóa dữ liệu nhật ký theo thời gian thực. Với sự phát triển không ngừng của dữ liệu và nhu cầu phân tích dữ liệu lớn, việc nâng cao hiệu suất, bảo mật, và khả năng tích hợp của ELK Stack là điều cần thiết.

Trong phạm vi đồ án nhóm đã thực hiện được các kịch bản như sau:

1. Thu thập từ nhiều nguồn và chuẩn hóa log
2. Quản lý và filter log
3. Custom Dashboard và visualize
4. Machine Learning cho ELK stack
5. Alert bất thường và trigger thông báo tới Email thông qua Security Alert và Kibana

Trong quá trình tìm hiểu đồ án nhò nhóm nhận thấy rằng ELK Stack ngày càng mạnh mẽ và hiệu quả hơn trong việc xử lý và phân tích dữ liệu lớn. Các cải tiến về hiệu suất và mở rộng khả năng tích hợp đã giúp các tổ chức có thể thu thập, phân tích và trực quan hóa dữ liệu một cách nhanh chóng và chính xác hơn. Tích hợp AI và Machine Learning giúp tự động hóa việc phân tích dữ liệu, phát hiện các sự kiện bất thường, và dự đoán xu hướng. Bảo mật và tuân thủ được nâng cao, đảm bảo an toàn dữ liệu và tuân thủ các quy định pháp luật. Việc dễ dàng triển khai và quản lý giúp tiết kiệm thời gian và nguồn lực, trong khi các công cụ trực quan hóa và báo cáo tiên tiến cung cấp cái nhìn sâu sắc hơn về dữ liệu. Cuối cùng, sự hỗ trợ từ cộng đồng và phát triển mở rộng ELK Stack tiếp tục cải tiến và đáp ứng nhu cầu ngày càng cao của người dùng.

Tóm lại, ELK Stack đang tiến hóa để trở thành một giải pháp toàn diện hơn cho việc quản lý và phân tích dữ liệu nhật ký, giúp các tổ chức tối ưu hóa quy trình làm việc và đưa ra quyết định dựa trên dữ liệu một cách hiệu quả.

6.2 Hướng phát triển

Trong phạm vi đồ án môn học, nhóm chỉ triển khai mô phỏng lại các vùng mạng cơ bản nhằm đáp ứng về tài nguyên máy và thời gian để hoàn thành kịp tiến độ. Việc này giúp nhóm tập trung vào việc hiểu và áp dụng các khái niệm cơ bản của ELK Stack mà không gặp phải các thách thức về triển khai trên quy mô lớn. Tuy nhiên, thực tế trong các doanh nghiệp vừa và lớn, mô hình mạng phức tạp hơn nhiều. Điều này đòi hỏi phải có nhiều server thu thập log để tăng khả năng chịu lỗi, tăng tính linh hoạt, cũng như thuận tiện trong quá trình nâng cấp hoặc sửa chữa một log server bất kỳ. Việc triển khai nhiều server giúp đảm bảo rằng hệ thống luôn hoạt động ổn định và dữ liệu được thu thập liên tục ngay cả khi một hoặc một vài server gặp sự cố. Ngoài ra, một hệ thống phân tán còn cho phép mở rộng dễ dàng khi nhu cầu về lưu trữ và xử lý dữ liệu tăng lên, mà không gây gián đoạn cho các dịch vụ hiện có.

DANH MỤC TÀI LIỆU THAM KHẢO

1. N. Đ. H. Học, “ELK Stack - 3 anh em siêu nhân trong quản lý logs,” *Viblo*, Jun. 12, 2024. https://viblo.asia/p/elk-stack-3-anh-em-sieu-nhan-trong-quan-ly-logs-oVIYLArZ8W#_gioi-thieu-0
2. “Bài 1: Elasticsearch là gì? Những điều bạn cần biết về Elasticsearch.” <https://stringee.com/vi/blog/post/elasticsearch-la-gi>
3. “Index of /logging/.” <https://blog.cloud365.vn/logging/>
4. S. Ulili, “How to Collect, Process, and Ship Log Data with Rsyslog,” *Better Stack Community*, Nov. 23, 2023. https://betterstack.com/community/guides/logging/rsyslog-explained/?fbclid=IwZXh0bgNhZW0CMTAAAR16Bb0VNBk_a37uzHFbxLYhr5xwzv7Ve4pJ418bb1mDZjq4_vPkux4EYrM_aem_ASdoP3eab888DDq_Bv_WdSjNBr4qxZqQ5Oc07-vAywS8VzQ0BIzhkaJsqU32Veex70yzMZVdHj4HanztVkJW68Y
5. 3ilson, “PfSense (V2.4.0+) + ELK (V5.6.3+),” *YouTube*. Nov. 03, 2017. [Online]. Available: <https://www.youtube.com/watch?v=of2ymhr9G3I>
6. E. Glass and J. Camisso, “How to install ElasticSearch, Logstash, and Kibana (Elastic Stack) on Ubuntu 22.04,” *DigitalOcean*, Apr. 26, 2022. https://www.digitalocean.com/community/tutorials/how-to-install-elasticsearch-logstash-and-kibana-elastic-stack-on-ubuntu-22-04?fbclid=IwZXh0bgNhZW0CMTAAAR3cIXQeSgYUwUbc9VT-ucgYbc9HCz-qZl3gyv31V_CJtIVinAftXTSJfmY_aem_AScoOYUe5dSCs9LIDkkvy7tCIWczODn1NsiM3xYtyH5XhEy6RtwuIEIwjA5OLY13x8-vcz7QmqpnEBppsAVqQZ7
7. “Elastic (ELK) Stack features list | Elastic,” *Elastic*. <https://www.elastic.co/elastic-stack/features>