

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



Đặng Trần Lê Anh

NGHIÊN CỨU CÁC KỸ THUẬT XỬ LÝ VÀ PHÂN TÍCH LOG

LUẬN VĂN THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

HÀ NỘI - 2019

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



Đặng Trần Lê Anh

NGHIÊN CỨU CÁC KỸ THUẬT XỬ LÝ VÀ PHÂN TÍCH LOG

Chuyên ngành: Hệ thống thông tin

Mã số: 8.48.01.04

LUẬN VĂN THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

NGƯỜI HƯỚNG DẪN KHOA HỌC: TS. HOÀNG XUÂN DẬU

HÀ NỘI - 2019

LỜI CAM ĐOAN

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi, kết quả đạt được trong luận văn là sản phẩm của riêng cá nhân, không sao chép lại của người khác. Trong toàn bộ nội dung của luận văn, những điều được trình bày hoặc là của cá nhân hoặc là được tổng hợp từ nhiều nguồn tài liệu. Tất cả các tài liệu tham khảo đều có xuất xứ rõ ràng và được trích dẫn hợp pháp. Các số liệu, kết quả nêu trong luận văn là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác.

Tác giả luận văn

Đặng Trần Lê Anh

MỤC LỤC

LỜI CAM ĐOAN	i
MỤC LỤC.....	ii
DANH MỤC CÁC KÝ HIỆU, CHỮ VIẾT TẮT	iv
DANH MỤC CÁC BẢNG.....	v
DANH MỤC CÁC HÌNH.....	vi
MỞ ĐẦU.....	1
CHƯƠNG 1 - TỔNG QUAN VỀ LOG TRUY NHẬP VÀ PHÂN TÍCH LOG.....	3
1.1. Tổng quan log truy nhập.....	3
1.1.1. Khái niệm log truy nhập	3
1.1.2. Các dạng log truy nhập.....	5
1.1.3. Thu thập, xử lý và phân tích log truy nhập	13
1.2. Ứng dụng của phân tích log truy nhập	14
1.3. Một số nền tảng và công cụ phân tích log	15
1.3.1. Graylog	15
1.3.2. Logstash.....	17
1.3.3. OSSEC.....	18
1.4. Kết luận chương.....	20
CHƯƠNG 2 - CÁC KỸ THUẬT PHÂN TÍCH LOG TRUY NHẬP	21
2.1. Mô hình xử lý log	21
2.2. Thu thập và tiền xử lý	22
2.2.1. Thu thập log.....	22
2.2.2. Tiền xử lý và chuẩn hóa	23
2.3. Các kỹ thuật phân tích log	30

2.3.1. Các kỹ thuật nhận dạng mẫu	30
2.3.2. Phân tích mẫu	33
2.4. Kết luận chương.....	33
CHƯƠNG 3 - CÀI ĐẶT VÀ THỬ NGHIỆM.....	34
3.1. Giới thiệu nền tảng và công cụ thử nghiệm.....	34
3.1.1. Kiến trúc Graylog	34
3.1.2. Các thành phần của Graylog	36
3.1.3. Các tính năng của Graylog	38
3.2. Cài đặt	41
3.2.1. Các mô đun thu thập log.....	41
3.2.2. Hệ thống xử lý và phân tích log	44
3.3. Các kịch bản thử nghiệm và kết quả.....	49
3.3.1. Các kịch bản thử nghiệm.....	49
3.3.2. Một số kết quả	51
3.4. Kết luận chương.....	56
KẾT LUẬN VÀ KIẾN NGHỊ.....	57
DANH MỤC TÀI LIỆU THAM KHẢO	58

DANH MỤC CÁC KÝ HIỆU, CHỮ VIẾT TẮT

Viết tắt	Tiếng Anh	Tiếng Việt
API	Application Programming Interface	Giao diện lập trình ứng dụng
ASCII	American Standard Code for Information Interchange	Chuẩn mã trao đổi thông tin Hoa Kỳ
CSS	Cascading Style Sheets	Tập tin định kiểu theo tầng
DNS	Domain Name System	Hệ thống tên miền
GELF	Graylog Extended Log Format	Định dạng nhật ký mở rộng Graylog
HTTP	Hypertext Transfer Protocol	Giao thức truyền tải siêu văn bản
ISP	Internet Service Provider	Nhà cung cấp dịch vụ Internet
JSON	JavaScript Object Notation	Một kiểu dữ liệu mở trong JavaScript
LAN	Local area network	Mạng máy tính cục bộ
LDAP	Lightweight Directory Access Protocol	Một giao thức ứng dụng truy cập các cấu trúc thư mục
PHP	Hypertext Preprocessor	Một ngôn ngữ lập trình kịch bản
SNMP	Simple Network Management Protocol	Giao thức quản lý mạng đơn giản
SQL	Structured Query Language	Ngôn ngữ truy vấn mang tính cấu trúc
TCP	Transmission Control Protocol	Giao thức điều khiển truyền vận
UDP	User Datagram Protocol	Giao thức dữ liệu người dùng
UI	User Interface	Giao diện người dùng
URI	Uniform Resource Identifier	Mã định danh tài nguyên thống nhất
URL	Uniform Resource Locator	Đường dẫn tham chiếu tới tài nguyên mạng trên Internet
W3C	World Wide Web Consortium	Tên tổ chức quốc tế lập ra các chuẩn cho Internet

DANH MỤC CÁC BẢNG

Bảng 1.1: Danh sách các tiền tố	8
Bảng 1.2: Các định danh không yêu cầu có tiền tố.....	9
Bảng 1.3: Các định danh cần phải có tiền tố.....	9
Bảng 1.4: Các định dạng dữ liệu sử dụng trong W3C Extended Format	10
Bảng 1.5: Các trường khả dụng trong W3C Extended Format.....	11
Bảng 2.1: Kết hợp địa chỉ IP và User agent.....	24
Bảng 2.2: Kết quả nhận dạng được người dùng 1.....	25
Bảng 2.3: Kết quả nhận dạng được người dùng 2.....	26
Bảng 2.4: Kết quả nhận dạng được người dùng 3.....	26
Bảng 2.5: Ví dụ trường hợp refferer sai.....	29

DANH MỤC CÁC HÌNH

Hình 1.1: Xem Windows log sử dụng công cụ Event Viewer.....	3
Hình 1.2: Các bản ghi log sinh ra bởi máy chủ web Microsoft IIS	4
Hình 1.3: Các khâu của quá trình thu thập, xử lý và phân tích log.....	13
Hình 1.4: Kiến trúc điển hình của hệ thống thu thập, xử lý và phân tích log	13
Hình 1.5: Màn hình quản lý các nguồn thu thập log của Graylog	16
Hình 1.6: Màn hình báo cáo tổng hợp của Graylog	17
Hình 1.7: Mô hình kết hợp hệ thống Logstash/Elasticsearch/Kibana.....	17
Hình 1.8: Giao diện của Kibana hiển thị kết quả xử lý của Logstash.....	18
Hình 1.9: Giao diện người dùng của OSSEC	19
Hình 2.1: Mô hình xử lý log truy nhập	21
Hình 2.2: Một ví dụ về nhận dạng phiên dựa trên thời gian	27
Hình 2.3: Một ví dụ về nhận dạng phiên dựa trên cấu trúc trang web.....	28
Hình 2.4: Ví dụ về tham chiếu sai do cache.....	30
Hình 2.5: Quá trình sử dụng luật kết hợp.....	31
Hình 2.6: Ví dụ sử dụng trực quan hóa dữ liệu.....	33
Hình 3.1: Kiến trúc Graylog tối giản	34
Hình 3.2: Kiến trúc Multi-Node Graylog.....	35
Hình 3.3: Các thành phần và tính năng của Graylog	36
Hình 3.4: Ví dụ về vòng đời của log trong Graylog	39
Hình 3.5: Cấu hình Rsyslog trên Linux	41
Hình 3.6: Kiểm tra port mà Rsyslog sử dụng.....	42
Hình 3.7: Cài đặt NXLog trên Windows Server	42
Hình 3.8: Cấu hình NXLog trên Windows Server.....	43
Hình 3.9: Tạo repository cho Elasticsearch	45
Hình 3.10: Kiểm tra trạng thái Elasticsearch sau khi cài đặt	45
Hình 3.11: Tạo repository cho MongoDB	46
Hình 3.12: Kiểm tra dịch vụ MongoDB sau khi cài đặt	46
Hình 3.13: Cấu hình cho Graylog Server.....	48

Hình 3.14: Giao diện truy cập Graylog Web Interface	48
Hình 3.15: Thêm Microsoft IIS input trong NXLog.....	49
Hình 3.16: Tạo GELF UDP input	50
Hình 3.17: Tạo Syslog UDP input	50
Hình 3.18: Quản lý các nguồn cung cấp log trên Graylog.....	51
Hình 3.19: Giao diện tìm kiếm log của Graylog.....	51
Hình 3.20: Các địa chỉ được truy cập nhiều nhất.....	52
Hình 3.21: Các page được truy cập nhiều nhất.....	52
Hình 3.22: Các user-agent truy cập vào website.....	53
Hình 3.23: Báo cáo các trạng thái HTTP khi truy cập website.....	53
Hình 3.24: Báo cáo các địa chỉ IP truy cập website.....	54
Hình 3.25: Báo cáo thời gian phản hồi khi truy cập website	54
Hình 3.26: Báo cáo thời gian đăng nhập của người dùng.....	55
Hình 3.27: Báo cáo tình trạng đăng nhập của người dùng.....	55
Hình 3.28: Nhận cảnh báo khi có đăng nhập bất thường.....	56

MỞ ĐẦU

Log (còn gọi là nhật ký, hay vết) là các mục thông tin do hệ điều hành, hoặc các ứng dụng sinh ra trong quá trình hoạt động. Mỗi bản ghi log thường được sinh ra theo 1 hoạt động, hoặc sự kiện, nên còn được gọi là nhật ký sự kiện (event log). Các nguồn sinh log phổ biến bao gồm các thiết bị mạng (như router, firewall,...), hệ điều hành, các máy chủ dịch vụ (máy chủ web, máy chủ cơ sở dữ liệu, máy chủ DNS, email,...) và các chương trình ứng dụng. Mục đích của việc thu thập, xử lý và phân tích log bao gồm:

- Kiểm tra sự tuân thủ các chính sách an ninh;
- Kiểm tra sự tuân thủ vấn đề kiểm toán và luật pháp;
- Phục vụ điều tra số;
- Phục vụ phản ứng các sự cố mất an toàn thông tin ;
- Hiểu các hành vi của người dùng trực tuyến, trên cơ sở đó tối ưu hóa hệ thống cho phục vụ tốt hơn cho người dùng hoặc quảng cáo trực tuyến.

Như vậy, việc xử lý và phân tích log có nhiều ứng dụng, đặc biệt trong đảm bảo an toàn thông tin và cải thiện chất lượng hệ thống và các dịch vụ kèm theo, như quảng cáo trực tuyến. Hiện nay, trên thế giới đã có một số nền tảng và công cụ cho thu thập, xử lý và phân tích các dạng log phiên bản thương mại cũng như mã mở như IBM Qradar SIEM, Splunk, Graylog và Logstash,... Tuy nhiên, việc nghiên cứu sâu các phương pháp xử lý và phân tích log và ứng dụng ở Việt Nam vẫn cần được tiếp tục thực hiện. Đây cũng là mục đích của đề tài luận văn này.

Luận văn bao gồm ba chương chính với nội dung như sau:

- Chương 1: Giới thiệu tổng quan về log truy nhập và phân tích log: khái niệm log truy nhập, các dạng log truy nhập, các phương pháp xử lý và phân tích log, ứng dụng của phân tích log và giới thiệu một số nền tảng, công cụ phân tích log.

- Chương 2: Trình bày các kỹ thuật phân tích log truy nhập: mô hình xử lý log, vấn đề thu thập và tiền xử lý log, các kỹ thuật phân tích log như nhận dạng mẫu và phân tích mẫu.

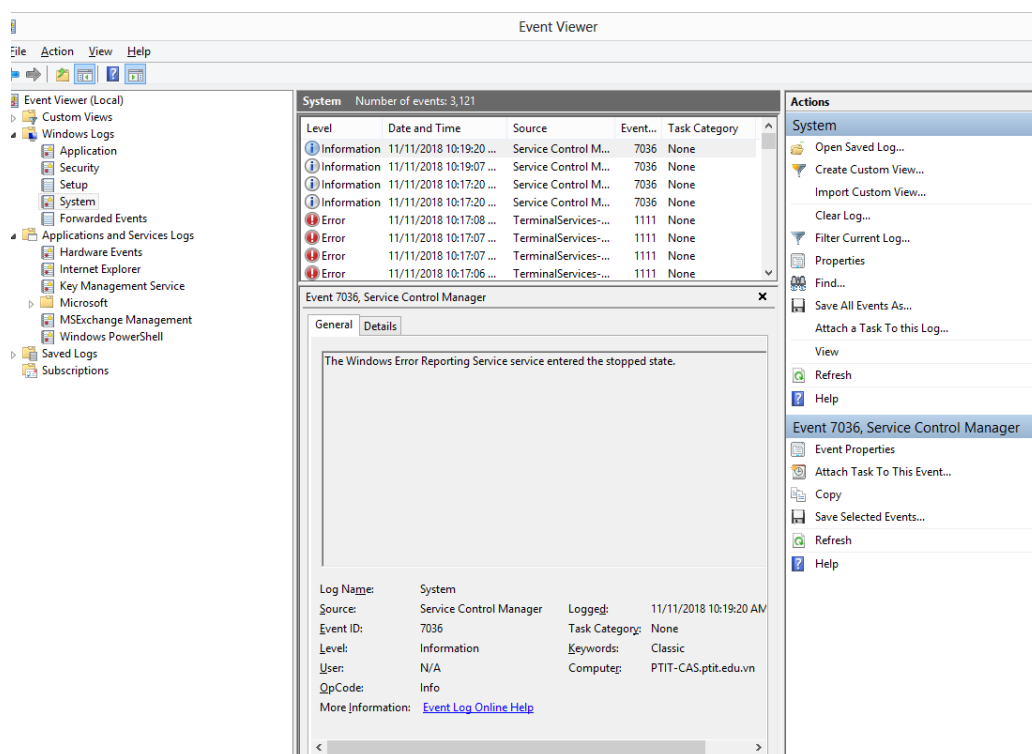
- Chương 3: Trình bày tổng quan về kiến trúc, các thành phần và tính năng của công cụ xử lý, phân tích log là Graylog. Chi tiết quá trình cài đặt các mô-đun thu thập log, hệ thống xử lý, phân tích log của Graylog. Đồng thời, đưa ra một số kịch bản thử nghiệm và kết quả.

CHƯƠNG 1 - TỔNG QUAN VỀ LOG TRUY NHẬP VÀ PHÂN TÍCH LOG

1.1. Tổng quan log truy nhập

1.1.1. Khái niệm log truy nhập

Log truy cập hay nhật ký, hoặc vết truy cập (gọi tắt là log) là một danh sách các bản ghi mà một hệ thống ghi lại khi xuất hiện các yêu cầu truy cập các tài nguyên của hệ thống. Chẳng hạn, log truy cập web (gọi tắt là web log) chứa tất cả các yêu cầu truy nhập các tài nguyên của một website. Các tài nguyên của một website như các file ảnh, các mẫu định dạng và file mã Javascript. Khi một người dùng thăm một trang web để tìm một sản phẩm, máy chủ web sẽ tải xuống thông tin và ảnh của sản phẩm và log truy cập sẽ ghi lại các yêu cầu của người dùng đến các tài nguyên thông tin và ảnh của sản phẩm.

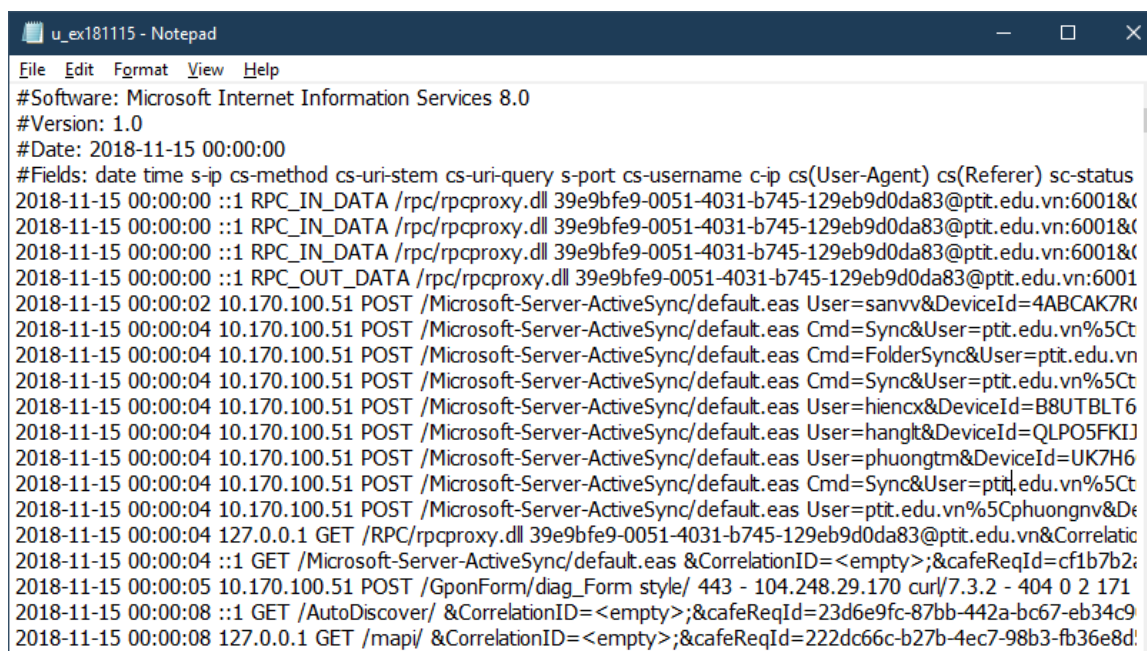


Hình 1.1: Xem Windows log sử dụng công cụ Event Viewer

Có nhiều nguồn sinh log trong hệ thống như log sinh bởi hệ điều hành, log sinh bởi các máy chủ dịch vụ mạng, log sinh bởi các ứng dụng và log sinh bởi các thiết bị mạng và thiết bị đảm bảo an toàn thông tin. Log sinh bởi hệ điều hành thường

bao gồm các bản ghi các sự kiện khởi động hệ thống, sự kiện đăng nhập, đăng xuất của người dùng, yêu cầu truy cập các file, các thư mục, các yêu cầu kích hoạt ứng dụng, các yêu cầu truy cập phần cứng, các yêu cầu truy cập dịch vụ mạng, các lỗi xuất hiện trong quá trình hoạt động... Hệ điều hành Microsoft Windows sử dụng công cụ Event Viewer (hình 1.1), còn các hệ điều hành thuộc họ Unix/Linux sử dụng công cụ Syslog để quản lý và lưu trữ log do bản thân hệ điều hành và các mô-đun phụ trợ sinh ra.

Nguồn log sinh bởi các máy chủ dịch vụ mạng, như máy chủ web, máy chủ DNS, máy chủ email và máy chủ cơ sở dữ liệu là một trong các nguồn log phổ biến nhất. Máy chủ web có thể ghi log truy cập các trang web cho từng website dưới dạng các file văn bản thuần với mỗi dòng là một bản ghi log. Các thông tin trong mỗi bản ghi web log có thể khác nhau phụ thuộc vào phiên bản máy chủ web sử dụng. Hình 1.2 minh họa các bản ghi log tạo bởi máy chủ web Microsoft IIS. Các máy chủ tên miền DNS cũng sinh một lượng lớn log trong quá trình xử lý các yêu cầu phân giải tên miền sang địa chỉ IP và ngược lại từ người dùng. Tương tự, các máy chủ email và cơ sở dữ liệu cũng sinh rất nhiều bản ghi log trong quá trình xử lý các yêu cầu từ người dùng cũng như từ các ứng dụng.



```

u_ex181115 - Notepad
File Edit Format View Help
#Software: Microsoft Internet Information Services 8.0
#Version: 1.0
#Date: 2018-11-15 00:00:00
#Fields: date time s-ip cs-method cs-uri-stem cs-uri-query s-port cs-username c-ip cs(User-Agent) cs(Referer) sc-status
2018-11-15 00:00:00 ::1 RPC_IN_DATA /rpc/rpcproxy.dll 39e9bfe9-0051-4031-b745-129eb9d0da83@ptit.edu.vn:6001&
2018-11-15 00:00:00 ::1 RPC_IN_DATA /rpc/rpcproxy.dll 39e9bfe9-0051-4031-b745-129eb9d0da83@ptit.edu.vn:6001&
2018-11-15 00:00:00 ::1 RPC_IN_DATA /rpc/rpcproxy.dll 39e9bfe9-0051-4031-b745-129eb9d0da83@ptit.edu.vn:6001&
2018-11-15 00:00:00 ::1 RPC_OUT_DATA /rpc/rpcproxy.dll 39e9bfe9-0051-4031-b745-129eb9d0da83@ptit.edu.vn:6001
2018-11-15 00:00:02 10.170.100.51 POST /Microsoft-Server-ActiveSync/default.eas User=sanvv&DeviceId=4ABCAK7R
2018-11-15 00:00:04 10.170.100.51 POST /Microsoft-Server-ActiveSync/default.eas Cmd=Sync&User=ptit.edu.vn%5Ct
2018-11-15 00:00:04 10.170.100.51 POST /Microsoft-Server-ActiveSync/default.eas Cmd=FolderSync&User=ptit.edu.vn
2018-11-15 00:00:04 10.170.100.51 POST /Microsoft-Server-ActiveSync/default.eas Cmd=Sync&User=ptit.edu.vn%5Ct
2018-11-15 00:00:04 10.170.100.51 POST /Microsoft-Server-ActiveSync/default.eas User=hiencx&DeviceId=B8UTBLT6
2018-11-15 00:00:04 10.170.100.51 POST /Microsoft-Server-ActiveSync/default.eas User=hanglt&DeviceId=QLPO5FKI
2018-11-15 00:00:04 10.170.100.51 POST /Microsoft-Server-ActiveSync/default.eas User=phuongtm&DeviceId=UK7H6
2018-11-15 00:00:04 10.170.100.51 POST /Microsoft-Server-ActiveSync/default.eas Cmd=Sync&User=ptit.edu.vn%5Ct
2018-11-15 00:00:04 10.170.100.51 POST /Microsoft-Server-ActiveSync/default.eas User=ptit.edu.vn%5Cphuongnv&De
2018-11-15 00:00:04 127.0.0.1 GET /RPC/rpcproxy.dll 39e9bfe9-0051-4031-b745-129eb9d0da83@ptit.edu.vn&Correlatic
2018-11-15 00:00:04 ::1 GET /Microsoft-Server-ActiveSync/default.eas &CorrelationID=<empty>;&cafeReqId=cf1b7b2;
2018-11-15 00:00:05 10.170.100.51 POST /GponForm/diag_Form style/ 443 - 104.248.29.170 curl/7.3.2 - 404 0 2 171
2018-11-15 00:00:08 ::1 GET /AutoDiscover/ &CorrelationID=<empty>;&cafeReqId=23d6e9fc-87bb-442a-bc67-eb34c9
2018-11-15 00:00:08 127.0.0.1 GET /map/ &CorrelationID=<empty>;&cafeReqId=222dc66c-b27b-4ec7-98b3-fb36e8d

```

Hình 1.2: Các bản ghi log sinh ra bởi máy chủ web Microsoft IIS

Các thiết bị mạng và các hệ thống đảm bảo an toàn thông tin cũng là một trong các nguồn sinh nhiều log. Các thiết bị mạng phổ biến như các bộ định tuyến (router), các bộ chuyển mạch (switch) và các hệ thống đảm bảo an toàn thông tin như tường lửa, các hệ thống điều khiển truy cập, các hệ thống phát hiện và ngăn chặn tấn công, xâm nhập cũng sinh nhiều bản ghi log trong quá trình xử lý các yêu cầu truy cập mạng. Log sinh từ các hệ thống này có thể được lưu tại chỗ hoặc xuất ra các hệ thống lưu trữ bên ngoài.

Như vậy, có thể thấy có nhiều nguồn sinh dữ liệu log truy cập với nhiều dạng khác nhau. Tùy vào mục đích sử dụng, người quản trị có thể cấu hình hệ thống để lựa chọn thu thập, quản lý và lưu trữ các thông tin cần thiết cho mỗi dạng log.

1.1.2. Các dạng log truy nhập

Log truy nhập sinh bởi hệ điều hành và các ứng dụng thường có định dạng riêng. Do phần thử nghiệm trong luận văn này được thực hiện trên web log nên mục này giới thiệu các định dạng web log được sử dụng phổ biến hiện nay bao gồm định dạng web log chuẩn của NCSA (NCSA Common Log Format), định dạng web log kết hợp (NCSA Combined Log Format), định dạng web log mở rộng của W3C (W3C Extended Log Format) và định dạng web log của máy chủ web Microsoft IIS (Microsoft IIS Log Format). Trên thực tế hiện nay, mỗi máy chủ web đều hỗ trợ một số định dạng web log trong số các định dạng kể trên. Chẳng hạn, máy chủ web Microsoft IIS hỗ trợ 3 định dạng, bao gồm: NCSA Common Log Format, W3C Extended Log Format và Microsoft IIS Log Format. Ngược lại, máy chủ web Apache hay Apache HTTP Server sử dụng *các chuỗi định dạng* để hỗ trợ 2 định dạng log bao gồm: NCSA Common Log Format và NCSA Combined Log Format. Người quản trị có thể lựa chọn định dạng web log sử dụng để máy chủ sinh các file web log.

1.1.2.1. NCSA Common Log Format

NCSA Common Log Format, hay thường được gọi tắt Common Log Format, là định dạng web log với trường cố định mà không thể tùy chỉnh. Dạng web log này ghi lại các thông tin cơ bản về yêu cầu người dùng, tên của máy khách, tên người dùng, ngày, giờ, loại yêu cầu, mã trạng thái HTTP trả về, số lượng byte gửi về server.

Các trường phân trong mỗi bản ghi log được phân cách bởi dấu trắng. Những trường không chứa dữ liệu sẽ được biểu diễn bằng dấu (-), các ký tự không in được sẽ biểu diễn bởi dấu (+).

Với máy chủ Apache HTTP Server, định dạng Common Log Format có thể được cấu hình nhờ chuỗi định dạng như sau:

LogFormat "%h %l %u %t \"%r\" %>s %b" common

Ví dụ, với Common Log Format thì một đầu mục (entry) sẽ có dạng như sau:

127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 200 2326

Trong đó, các trường thông tin của đầu mục này gồm:

- 127.0.0.1 (tương ứng kí hiệu %h): Địa chỉ IP của máy khách gửi yêu cầu đến máy chủ.
- Trống (-) (tương ứng kí hiệu %l): Định danh của máy khách.
- frank (tương ứng kí hiệu %u): Định danh/tên của người dùng gửi yêu cầu được xác định nhờ thủ tục xác thực HTTP.
- [10/Oct/2000:13:55:36 -0700] (tương ứng kí hiệu %t): Thời gian máy chủ kết thúc xử lý yêu cầu, theo định dạng sau: [day/month/year:hour:minute:second zone], hay ngày/tháng/năm:giờ:phút:giây và múi giờ. Trong đó, day = 2*digit, month = 3*letter; year = 4*digit; hour = 2*digit; minute = 2*digit; second = 2*digit và zone = ('+' | '-') 4*digit.
- "Get /apache_pb.gif HTTP/1.0" (tương ứng kí hiệu \"%r\"): Yêu cầu của máy khách gửi lên máy chủ.
- 200 (tương ứng kí hiệu %>s): Mã trạng thái mà máy chủ gửi trả về cho máy khách.
- 2326 (tương ứng kí hiệu %b): Kích thước của gói tin trả về cho máy khách, không bao gồm header.

1.1.2.2. NCSA Combined Log Format

NCSA Combined Log Format gọi tắt là Combined Log Format về cơ bản tương tự Common Log Format, ngoại trừ việc nó bổ sung thêm hai trường thông tin ở cuối là Referrer (Liên kết tham chiếu) và User agent (Máy khách người dùng). Với Apache HTTP Server, định dạng này có thể được cấu hình nhờ chuỗi định dạng như sau:

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-agent}i\" combined
```

Ví dụ, với Combined Log Format, một đầu mục sẽ như sau:

```
127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif
HTTP/1.0" 200 2326 "http://www.example.com/start.html" "Mozilla/4.08 [en]
(Win98;I;Nav)"
```

Các trường được bổ sung bao gồm:

- `http://www.example.com/start.html` (tương ứng kí hiệu `\"%{Referer}i\"`): Cho biết trang web người dùng đã thăm trước khi đến trang hiện tại.
- `Mozilla/4.08 [en] (Win98; I ;Nav)"` (tương ứng kí hiệu `\"%{User-agent}i\"`): Cho biết thông tin về trình duyệt và hệ điều hành máy khách đang sử dụng.

1.1.2.3. W3C Extended Log Format

Hiện nay, W3C Extended Log Format [6] đề xuất bởi The World Wide Web Consortium (W3C) là định dạng được sử dụng rộng rãi nhất và được hầu hết các máy chủ web hỗ trợ. Định dạng web log này có các khả năng:

- Hỗ trợ kiểm soát những thông tin sẽ được ghi trong web log.
- Hỗ trợ một định dạng web log chung cho cả proxy, máy khách và máy chủ web.
- Cung cấp một cơ chế mạnh mẽ xử lý các vấn đề và các ký tự thoát (character escaping).
- Cho phép trao đổi dữ liệu nhân khẩu học (demographic).
- Hỗ trợ tổng hợp dữ liệu.

Một file log theo định dạng W3C Extended Log chứa một tập hợp các dòng văn bản thuần gồm các ký tự theo chuẩn ASCII (hoặc Unicode) được phân tách bởi ký tự xuống dòng (LF hoặc CRLF). Các file log khác nhau sẽ có ký tự kết thúc dòng khác nhau tùy thuộc vào quy ước kết thúc dòng của nền tảng hoạt động. Trên mỗi dòng thường có một *chỉ thị* (directive) hoặc một *đầu mục* (entry). Phần tiếp theo mô tả chi tiết về 2 thành phần này.

- Các chỉ thị:

Các dòng bắt đầu bằng ký tự “#” thì sẽ chứa các chỉ thị. Chúng chứa các thông tin mô tả về file log. Các chỉ thị với định dạng W3C Extended Log bao gồm:

- Version: *<integer>.<integer>*: Chỉ ra phiên bản của định dạng log được sử dụng.
- Fields: [*<specifier>...*]: Liệt kê danh sách các trường được ghi lại trong tệp log.
- Software: *string*: Chỉ ra phần mềm tạo ra log:
- Start-Date: *<date> <time>*: Ngày và giờ bắt đầu ghi log.
- End-Date: *<date> <time>*: Ngày và giờ kết thúc ghi log.
- Date: *<date> <time>*: Ngày và giờ thêm vào các đầu mục trong log.
- Remark: *<text>*: Các thông tin chú thích. Thông thường, các công cụ phân tích log sẽ bỏ qua dữ liệu trong trường này.

Các chỉ thị *Version* và *Fields* là bắt buộc và đứng trước tất cả các trường khác trong file log. Chỉ thị *Fields* liệt kê một danh sách định danh của trường, xác định thông tin được ghi trong mỗi đầu mục. Các định danh trường có thể là một trong số các kiểu sau: *Identifier* (tên nhận dạng), *Prefix-identifier* (tiền tố tên nhận dạng) và *Prefix* (header) (tiền tố (đề mục)).

Bảng 1.1: Danh sách các tiền tố

Tiền tố	Ý nghĩa
c	Client
s	Server
r	Remote
s	Client đến Server

sc	Server đến Client
sr	Server đến Remote Server (được dùng bởi proxy)
rs	Remote Server đến Server (được dùng bởi proxy)
x	Định danh riêng của ứng dụng

Bảng 1.2: Các định danh không yêu cầu có tiền tố

Định danh	Ý nghĩa
date	Ngày giao dịch hoàn thành, kiểu <date>
time	Thời gian (giờ) giao dịch hoàn thành, kiểu <time>
time-taken	Thời gian để giao dịch được hoàn thành tính bằng giây, kiểu <fixed>
bytes	Số byte đã truyền, kiểu <integer>
cached	Ghi lại số lần cache hit, nếu bằng 0 thì tức là cache miss, kiểu <integer>

Bảng 1.1 liệt kê danh sách các tiền tố (Prefix) cho các định danh, bảng 1.2 cung cấp danh sách các định danh không yêu cầu có tiền tố và bảng 1.3 liệt kê danh sách các định danh phải có tiền tố. Ví dụ, định danh *cs-method* cho biết *method* (phương thức) của gói tin gửi đi bởi client đến server, *sc(Referer)* tương ứng với trường *referer* trong gói tin trả lời, định danh *c-ip* xác định địa chỉ IP của client.

Bảng 1.3: Các định danh cần phải có tiền tố

Định danh	Ý nghĩa
ip	Địa chỉ IP và cổng, kiểu <address>
dns	Tên DNS, kiểu <name>
status	Mã trạng thái, kiểu <integer>
comment	Mô tả trạng thái trả về của mã trạng thái, kiểu <text>
method	Method, kiểu <name>
uri	URL, kiểu <uri>
uri-stem	Phần thân của URL (bỏ qua phần truy vấn), kiểu <uri>
uri-query	Phần truy vấn của URI, kiểu <uri>
host	DNS hostname được sử dụng, kiểu <name>

- Các đầu mục:

Một đầu mục (*entry*) là một dãy các trường liên quan đến một giao dịch HTTP, gồm một dãy các trường được phân cách bởi khoảng trắng hoặc các ký tự tab, không chứa các ký tự ASCII điều khiển và kết thúc bằng ký tự *CR* hoặc *CRLF*. Ý nghĩa của các trường được định nghĩa bởi chỉ thị *#Fields* và nếu một trường không có thông tin trong mục thì nó sẽ được hiển thị một ký tự “-“. Bảng 1.4 mô tả các định dạng dữ liệu sử dụng trong W3C Extended Format và bảng 1.5 liệt kê danh sách các trường khả dụng trong định dạng web log này.

Bảng 1.4: Các định dạng dữ liệu sử dụng trong W3C Extended Format

Định dạng dữ liệu	Mô tả
Integer	Định dạng: <code><integer> = 1*<digit></code> Trong đó, một số integer được biểu diễn như là một dãy các chữ số
Fixed Format Float	Định dạng: <code><fixed> = 1*<digit> [.*<digit>]</code>
URI	Theo chuẩn RCF 1738 và không được phép chứa khoảng trắng hay ký tự điều khiển ASCII.
Date	Định dạng: <code><date> = 4<digit> “-“ 2<digit> “-“ 2<digit></code> Ngày, tháng, năm được ghi với định dạng YYYY-MM-DD. Với YYYY, MM, DD tương ứng là năm, tháng và ngày. Lựa chọn định dạng này giúp sắp xếp dễ dàng hơn.
Time	Định dạng: <code><time> = 2<digit> “:” 2<digit> [“:” 2<digit> [“.” *<digit>]</code> Thời gian được ghi với định dạng HH:MM, HH:MM:SS hoặc HH:MM:SS.S với HH là giờ từ 00-23, MM là phút, SS là giây.
String	Định dạng: <code><string> = “” *<schar> “”</code> . Với <code><schar> = xchar “” “”</code> Các string được đặt trong dấu ngoặc kép, nếu một string chứa dấu ngoặc kép cũng không gây khó hiểu bởi vì các trường được phân tách bởi khoảng trắng.
Text	Định dạng: <code><text> = <char>*</code> Trường text chỉ được sử dụng bởi các chỉ thị.
Address	Định dạng: <code><name> = <integer> [“.” *<integer>] [“.” <integer>]</code> Địa chỉ IP và port (trường port là tùy chọn).

Bảng 1.5: Các trường khả dụng trong W3C Extended Format

Trường	Tên trong file log	Mô tả
Date	date	Ngày giao dịch xảy ra
Time	time	Thời gian giao dịch xảy ra (UTC)
Service Name and Instance Number	s-sitename	Tên dịch vụ và số tiến trình chạy
Server Name	s-computername	Tên của server được tạo trong tệp tin log
Server IP Address	s-ip	Địa chỉ của server được tạo trong tệp tin log
Method	cs-method	Là phương thức yêu cầu, ví dụ như phương thức GET
URI Stem	cs-uri-stem	Đối tượng mục tiêu của phương thức, ví dụ như Default.html
URI Query	cs-uri-query	Universal Resource Identifier, được dùng trong các trang động
Server Port	s-port	Cổng trên server mà đã được cấu hình cho dịch vụ
User Name	cs-name	Tên của người dùng hợp lệ đã truy cập vào server. Người dùng ẩn danh thì được biểu diễn bởi dấu “-”.
Client IP Address	c-ip	Địa chỉ IP của máy khách đã gửi yêu cầu
Protocol Version	cs-version	Phiên bản giao thức HTTP được máy khách sử dụng
User Agent	cs(User-Agent)	Loại trình duyệt mà máy khách đã sử dụng
Cookie	cs(Cookie)	Nội dung của cookie được gửi hoặc nhận, nếu có.
Referrer	cs(Referrer)	Trang web mà người dùng truy cập lần cuối, trang này cung cấp một đường link đến trang web hiện tại.
Host	cs-host	Host header name, nếu có
HTTP Status	sc-status	Mã trạng thái HTTP
Protocol Substatus	sc-substatus	Mã trạng thái phụ giao thức
Win32 Status	sc-win32-status	Mã trạng thái Windows
Bytes Sent	sc-bytes	Số lượng byte được gửi bởi server
Bytes Received	cs-bytes	Số lượng byte nhận và xử lý bởi server
Time Taken	time-taken	Độ dài khoảng thời gian diễn ra hành động (mili giây)

1.1.2.4. Microsoft IIS Log Format

Microsoft IIS là máy chủ web chạy trên hệ điều hành Microsoft Windows Server. Như đã trình bày, IIS hỗ trợ nhiều định dạng web log khác nhau như: NCSA Common Log Format, W3C Extended Log Format và Microsoft IIS Log Format. Các định dạng NCSA Common Log Format và W3C Extended Log Format đã được trình bày ở trên. Mục này tập trung mô tả định dạng Microsoft IIS Log Format.

Microsoft IIS Log Format [7] chứa các thông tin cơ bản như: Địa chỉ IP của máy khách, tên người dùng, ngày, giờ thực hiện yêu cầu, mã trạng thái dịch vụ, số lượng byte đã nhận. Ngoài ra, nó còn chứa các thông tin chi tiết như hành động thực hiện, file đích, thời gian thực hiện. Các trường trong mỗi bản ghi log được phân cách bởi dấu phẩy, những trường không chứa thông tin thay bằng dấu '-', các ký tự không in được thay bằng dấu '+'. Ví dụ, với Microsoft IIS Log Format thì một đầu mục của web log sẽ như sau:

192.168.114.201, -, 03/20/01, 7:55:20, W3SVC2, SALE1, 172.21.13.45, 4502, 163, 3223, 200, 0, GET, /DeptLogo.gif, -,

Trong đó:

- *192.168.114.201* là địa chỉ IP máy khách
- *03/20/01, 7:55:20* là ngày và giờ thực hiện yêu cầu
- *W3SVC2* chỉ tiến trình chạy dịch vụ web
- *SALES1* là tên máy chủ web
- *172.21.13.45* là địa chỉ IP máy chủ web
- *4502* là thời gian xử lý tính bằng mili giây
- *163* là số byte của yêu cầu
- *3223* là số byte của phản hồi (kết quả) máy chủ gửi máy khách
- *200* là mã trạng thái thực hiện yêu cầu (thành công)
- *GET* là phương thức yêu cầu
- */DeptLogo.gif* là file được yêu cầu.

1.1.3. Thu thập, xử lý và phân tích log truy nhập

Thu thập, xử lý và phân tích log là các khâu cơ bản của một hệ thống phân tích log. Hình 1.3 biểu diễn các khâu cụ thể của quá trình thu thập, xử lý và phân tích log thường được áp dụng trên thực tế. Theo đó, các khâu xử lý cụ thể gồm:

- *Thu thập dữ liệu log* là khâu trong đó các bản ghi log thô từ các nguồn sinh log được thu thập và chuyển về trung tâm xử lý.
- *Làm sạch dữ liệu* là khâu trong đó các bản ghi log thô được làm sạch để giảm bớt dữ liệu nhiễu.

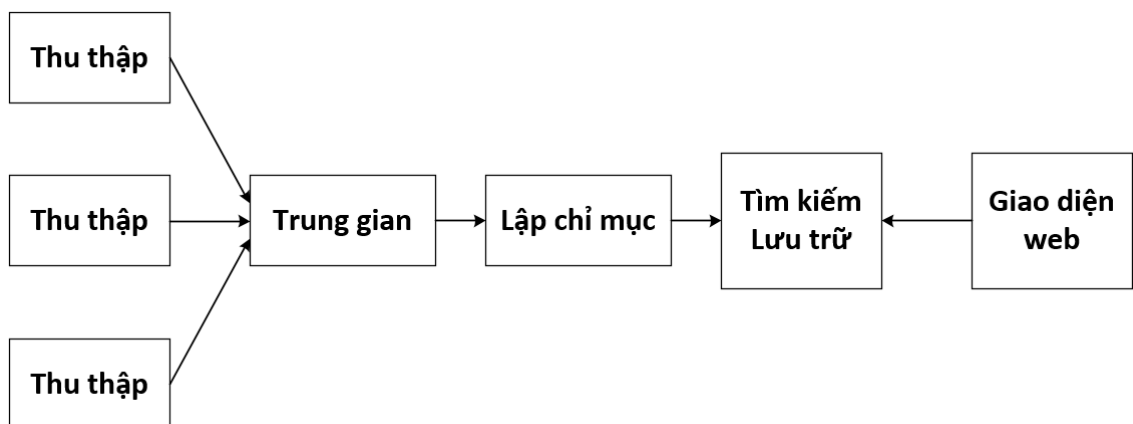


Hình 1.3: Các khâu của quá trình thu thập, xử lý và phân tích log

- *Chuẩn hóa dữ liệu* là khâu chuẩn hóa dữ liệu log. Do log có thể được thu thập từ nhiều nguồn với nhiều định dạng khác nhau nên cần thiết phải được chuẩn hóa và đưa về dạng có cấu trúc, làm đầu vào cho khâu phân tích log.

- *Phân tích dữ liệu* là khâu quan trọng nhất trong quá trình phân tích log. Đây là khâu được áp dụng để trích xuất ra các thông tin quan trọng ứng dụng cho đảm bảo an toàn thông tin và các ứng dụng khác.

- *Kết quả thu được* là khâu kết xuất kết quả ra giao diện người dùng.



Hình 1.4: Kiến trúc điển hình của hệ thống thu thập, xử lý và phân tích log

Hình 1.4 biểu diễn kiến trúc điển hình của hệ thống thu thập, xử lý và phân tích log. Theo đó, các thành phần chính của hệ thống gồm:

- *Thu thập* là mô-đun giám sát, thu thập log từ các nguồn sinh log khác nhau. Các bộ thu thập thường được cài đặt trên các hệ thống được giám sát. Bộ thu thập có thể chỉ đơn giản thu thập các bản ghi log thô và gửi về trung gian, hoặc nó cũng có thể thực hiện các nhiệm vụ làm sạch và chuẩn hóa dữ liệu log.

- *Trung gian* là mô-đun tiếp nhận dữ liệu log từ nhiều nguồn gửi đến. Sau khi tiếp nhận, dữ liệu log được làm sạch, chuẩn hóa và chuyển tiếp cho khâu tiếp theo.

- *Lập chỉ mục* là mô-đun lập chỉ số cho dữ liệu log. Lập chỉ số là một khâu quan trọng phục vụ tìm kiếm, trích chọn dữ liệu log trong khâu tiếp theo.

- *Tìm kiếm & Lưu trữ* là khâu cung cấp các tính năng tìm kiếm, trích chọn các dữ liệu log quan trọng và quản lý, lưu trữ log.

- *Giao diện web* là giao diện người dùng trên nền web cho hệ thống quản lý và phân tích log.

1.2. Ứng dụng của phân tích log truy nhập

Việc phân tích log truy cập thường được thực hiện cho các mục đích: (1) đảm bảo an toàn thông tin cho hệ thống, (2) hỗ trợ khắc phục sự cố hệ thống, (3) hỗ trợ điều tra số và (4) hỗ trợ hiểu được hành vi người dùng trực tuyến.

Có thể thấy, phân tích log truy cập phục vụ đảm bảo an toàn thông tin cho hệ thống là một trong các mục đích chính. Cụ thể, phân tích log truy cập có thể hỗ trợ việc giám sát, kiểm tra việc tuân thủ các chính sách bảo mật, chính sách kiểm toán của cơ quan, tổ chức. Hơn nữa phân tích log truy cập có thể hỗ trợ phản ứng lại các sự cố an toàn thông tin thông qua việc hỗ trợ xác định nguyên nhân và yếu tố gây mất an toàn. Nhiều công cụ đảm bảo an toàn thông tin dựa trên việc giám sát, thu thập, xử lý và phân tích log đã được nghiên cứu, phát triển và triển khai trên thực tế, như IBM QRadar SIEM [5], VNCS Web Monitoring [8] và hệ thống phát hiện xâm nhập OSSEC [10]. Các công cụ này giám sát, thu thập các dạng log sinh bởi hệ điều hành, các dịch vụ, các ứng dụng trong hệ thống cần giám sát nhằm phát hiện các hành vi bất thường và các dạng tấn công, xâm nhập.

Hỗ trợ khắc sự cố hệ thống cũng là một trong các ứng dụng quan trọng của phân tích log truy cập. Phân tích log truy cập giúp loại bỏ bớt các dữ liệu nhiễu, tổng hợp các thông báo lỗi riêng lẻ, giúp xác định nguyên nhân của sự cố hệ thống rõ ràng và chính xác hơn và trên cơ sở đó người quản trị có thể đưa ra biện pháp khắc phục sự cố phù hợp.

Phân tích log truy cập cũng có thể hỗ trợ điều tra số thông qua việc lần vết, xâu chuỗi các sự kiện log riêng lẻ sử dụng các kỹ thuật khai phá dữ liệu và phân tích tương quan. Từ đó, kết quả phân tích log có thể được sử dụng để tạo dựng các bằng chứng số cho các sự cố mất an toàn thông tin.

Hỗ trợ hiểu được hành vi người dùng trực tuyến là một trong các mục đích chính trong phân tích log truy cập, nhất là phân tích log truy cập các website hay web log. Phân tích web log có thể tạo ra các báo cáo sử dụng các trang web của người dùng, bao gồm lưu lượng truy nhập, các trang tham chiếu, phân bố người dùng theo vị trí địa lý và lượng dữ liệu tải xuống. Đồng thời, phân tích log truy cập cũng giúp trích xuất nhiều thông tin quan trọng về hành vi người dùng trực tuyến và trên cơ sở đó có thể hỗ trợ việc tối ưu hóa website, nhằm nâng cao chất lượng dịch vụ cung cấp và trải nghiệm người dùng. Các công cụ phân tích log được phát triển và triển khai trên thực tế cho mục đích này có thể liệt kê bao gồm: Sumo Logic, Logstash, Graylog và Webalizer.

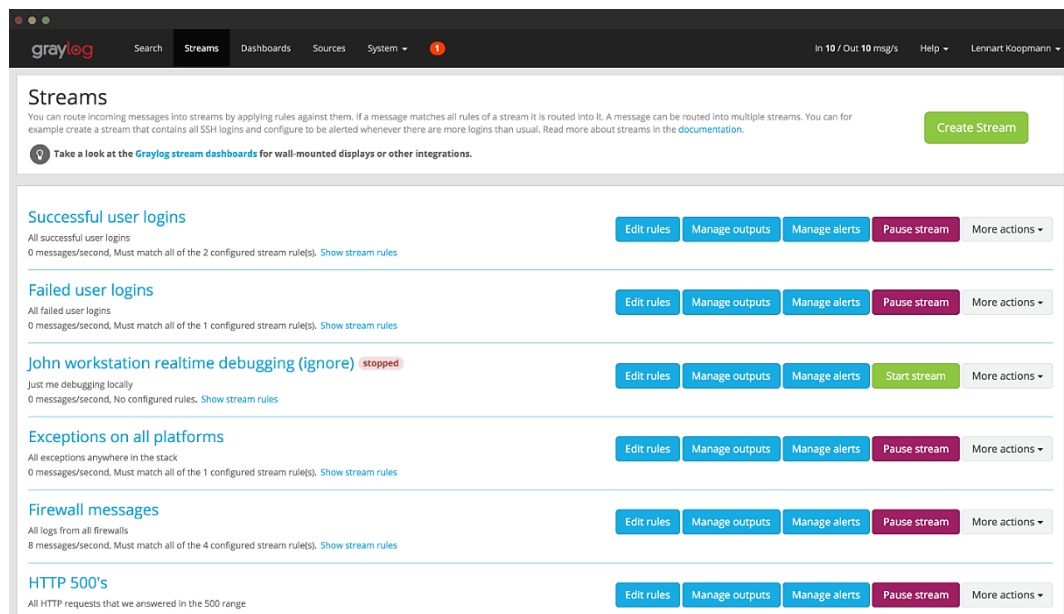
1.3. Một số nền tảng và công cụ phân tích log

Có nhiều nền tảng và công cụ xử lý, phân tích log truy cập thương mại cũng như mã nguồn mở được cung cấp hiện nay như Splunk, Sumo Logic, Monitoring, Logstash, Graylog, LOGalyze, Webalizer và OSSEC... Mục này giới thiệu khái quát về tính năng và các ưu nhược điểm của một số nền tảng và công cụ phân tích log điển hình, bao gồm Graylog, Logstash và OSSEC.

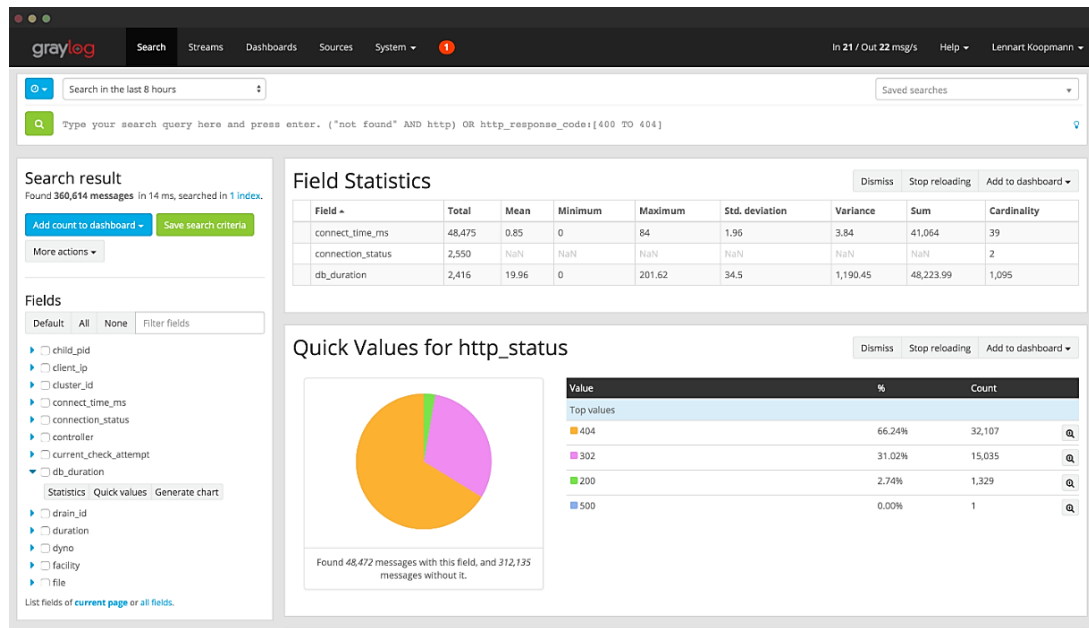
1.3.1. Graylog

Graylog là một nền tảng mã nguồn mở cho phép xử lý, phân tích log truy cập từ nhiều nguồn theo thời gian thực. Việc thu thập dữ liệu log được thực hiện rất mềm dẻo nhờ khả năng hỗ trợ các công cụ thu thập log của các bên thứ ba như beats, fluentd

và nxlog. Hình 1.5 minh họa màn hình quản lý các nguồn thu thập log của Graylog [11]. Graylog có khả năng phân tích hành vi người dùng, ứng dụng cho phép phát hiện và cảnh báo các truy cập bất thường cũng như trích xuất các mẫu hành vi truy cập phục vụ cho tối ưu hóa các trang web. Graylog cũng cho phép ánh xạ từ ID sang tên truy nhập của người dùng và ánh xạ từ địa chỉ IP sang vị trí địa lý. Hình 1.6 biểu diễn màn hình báo cáo tổng hợp của Graylog [11]. Mặc dù Graylog có khả năng nhận dạng các hành vi truy cập bất thường, nhưng nó không cho phép phân tích chuyên sâu các nguy cơ mất an toàn thông tin, như các dấu hiệu xuất hiện các dạng mã độc và các dạng tấn công lên các dịch vụ và tài nguyên mạng.

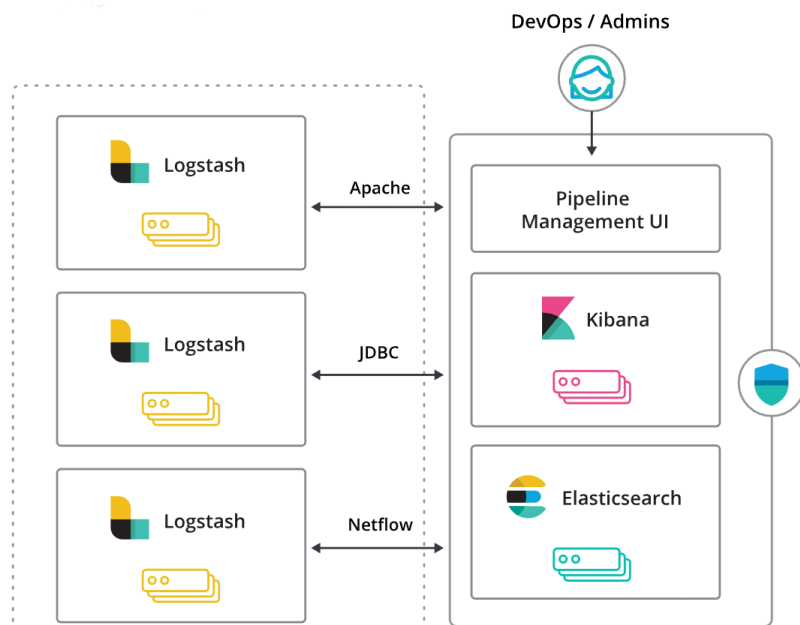


Hình 1.5: Màn hình quản lý các nguồn thu thập log của Graylog



Hình 1.6: Màn hình báo cáo tổng hợp của Graylog

1.3.2. Logstash



Hình 1.7: Mô hình kết hợp hệ thống Logstash/Elasticsearch/Kibana

Logstash là một công cụ mã nguồn mở cho phép thu thập, xử lý và quản lý các file log. Logstash không hoạt động độc lập mà nó được kết hợp sử dụng với công cụ Elasticsearch (tham khảo <https://www.elastic.co>) để lập chỉ số và tìm kiếm dữ liệu, và công cụ Kibana (tham khảo <https://www.elastic.co/products/kibana>) để biểu diễn dữ liệu dưới dạng biểu đồ.



Hình 1.8: Giao diện của Kibana hiển thị kết quả xử lý của Logstash

Hình 1.7 biểu diễn mô hình kết hợp hệ thống xử lý log, gồm Logstash, Elasticsearch và Kibana [12]. Hình 1.8 mô tả giao diện của Kibana hiển thị kết quả xử lý của Logstash [12].

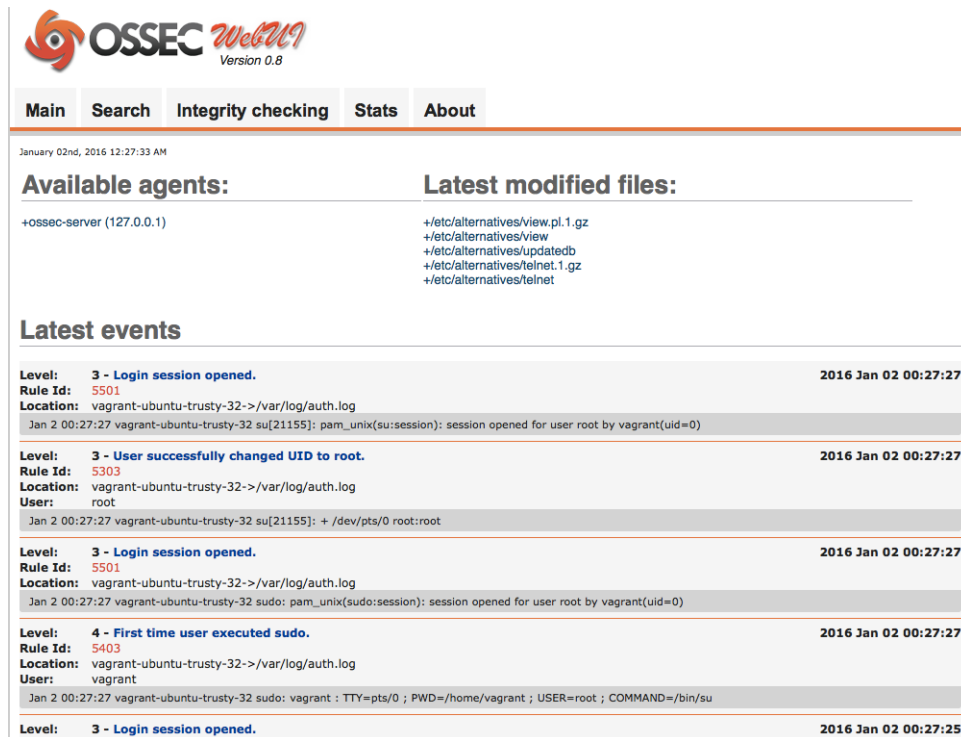
Ưu điểm nổi bật của Logstash là mã mở và do vậy chi phí cài đặt và vận hành tương đối thấp. Tuy nhiên, nhược điểm của Logstash là phụ thuộc vào các công cụ khác được triển khai bằng nhiều ngôn ngữ khác nhau dẫn đến khó khăn trong triển khai và bảo trì hệ thống.

1.3.3. OSSEC

OSSEC là một hệ thống phát hiện xâm nhập cho máy mã nguồn mở, thực hiện phân tích log, kiểm tra tính toàn vẹn, phát hiện rootkit và cảnh báo thời gian thực. OSSEC cung cấp kiến trúc đa nền tảng tập trung, cho phép quản lý bảo mật máy tính từ một vị trí trung tâm.

OSSEC có thể kiểm tra tính toàn vẹn của các file hệ thống, phát hiện rootkit và có một công cụ phân tích log mạnh mẽ có khả năng phân tích gần như mọi loại log được tạo trên một hệ thống. Việc phân tích log có thể được thực hiện đối với một số dịch vụ như Apache, Bind, LDAP và bản ghi log bên thứ ba từ các thiết bị như

Cisco. Ngoài ra, OSSEC còn chứa mô-đun hành động phản hồi có thể phản ứng lại các cuộc tấn công hoặc mối đe dọa được phát hiện.



Hình 1.9: Giao diện người dùng của OSSEC

Hình 1.9 thể hiện giao diện người dùng của OSSEC [10]. OSSEC cung cấp các tính năng chính sau đây:

Giám sát toàn vẹn tập tin: Còn được gọi là syscheck, là một xác nhận hợp lệ định kỳ về tính toàn vẹn của hệ điều hành hoặc các ứng dụng file bằng cách so sánh trạng thái hiện tại và giá trị được lưu trữ đã biết. Nó là một phần rất quan trọng trong việc phát hiện xâm nhập, và nó thường sử dụng các hàm băm để kiểm tra, phát hiện các thay đổi. OSSEC sử dụng mã MD5/SHA1 để giám sát các file cấu hình quan trọng trong một hệ thống.

Phân tích log thời gian thực: OSSEC hỗ trợ phân tích log thời gian thực, có nghĩa là sự kiện được kiểm tra ngay sau khi được tạo ra. Trong OSSEC, có hai quá trình khác nhau chịu trách nhiệm cho việc theo dõi log là *logcollector* và *analysisd*. Logcollector chạy trên máy khách và chịu trách nhiệm giám sát các sự kiện hệ thống được tạo và thu thập chúng. Analysisd chạy trên giao diện chính, chịu trách nhiệm giải mã, lọc và phân loại sự kiện.

Phát hiện Rootkit: Rootkit là một ứng dụng độc hại ẩn trong hệ thống được thiết kế để cấp quyền truy cập vào hệ thống và chiếm quyền quản trị. OSSEC sử dụng công cụ phát hiện rootkit, còn được gọi là rootcheck để phát hiện và xác định các loại rootkit.

Cảnh báo và hành động phản hồi: OSSEC đi kèm với một tập hợp các công cụ hành động phản hồi được xác nhận trước có thể kích hoạt bởi máy khách hoặc máy chủ như một phản ứng khi một điều kiện cho hành động phản hồi được đáp ứng.

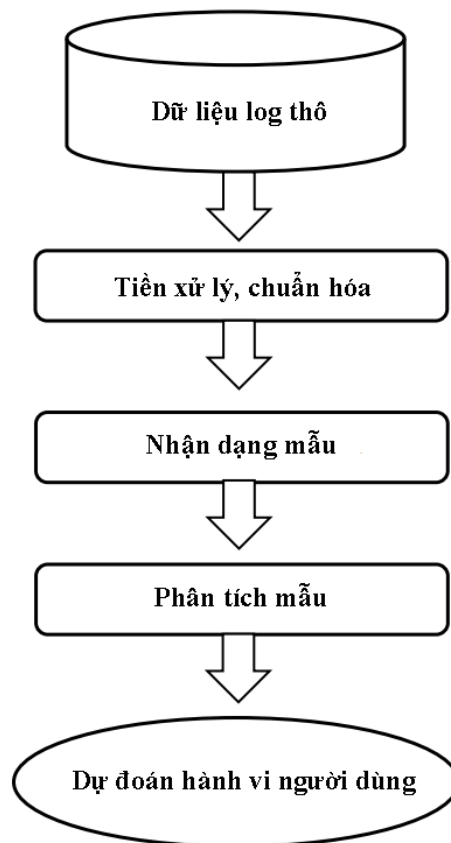
1.4. Kết luận chương

Chương này đã trình bày khái quát về log truy nhập, các nguồn sinh log, tổng quan về thu thập, xử lý và phân tích log. Chương cũng giới thiệu chi tiết các định dạng log truy nhập phổ biến, các khâu xử lý, phân tích log cũng như ứng dụng của phân tích log. Đồng thời, chương cũng khảo sát một số nền tảng và công cụ phân tích log phổ biến hiện nay.

CHƯƠNG 2 - CÁC KỸ THUẬT PHÂN TÍCH LOG TRUY NHẬP

2.1. Mô hình xử lý log

Hình 2.1 mô tả mô hình xử lý log truy nhập, mô hình gồm các pha chính: Pha tiền xử lý và chuẩn hóa; Pha nhận dạng mẫu; Pha phân tích mẫu; Pha dự đoán hành vi người dùng. Do luận văn này chủ yếu thực hiện các thực nghiệm trên web log, nên các mục của chương này tập trung trình bày về các kỹ thuật xử lý và phân tích web log.



Hình 2.1: Mô hình xử lý log truy nhập

Tiền xử lý và chuẩn hóa:

Trong pha này, hệ thống nhận dữ liệu log từ các nguồn khác nhau, trích xuất các thông tin cần thiết và đưa về một định dạng thống nhất. Ngoài ra, pha này còn chịu trách nhiệm tiền xử lý một số thông tin như: người dùng, phiên làm việc... Pha này gồm các bước sau: làm sạch và hợp nhất dữ liệu, nhận dạng người dùng, nhận

dạng phiên làm việc. Trong xử lý web log, còn bổ sung thêm bước nhận dạng pageview, hoàn thiện đường dẫn.

Nhận dạng mẫu:

Pha này sử dụng các phương pháp và thuật toán như: thống kê, học máy, khai phá dữ liệu, nhận dạng mẫu để xác định các mẫu của người dùng. Trong phân tích web log, các mẫu cơ bản cần xác định bao gồm: các trang web ưa thích, thời gian xem trung bình mỗi trang web, các lĩnh vực quan tâm... Pha này có thể sử dụng các kỹ thuật phân tích dữ liệu như: phân tích thống kê, phân cụm, phân lớp, luật kết hợp, các mẫu tuần tự, hay mô hình hóa phụ thuộc.

Phân tích mẫu:

Pha này có nhiệm vụ phân tích các mẫu đã tìm được ở pha trước, chỉ ra các mẫu không có nhiều giá trị và loại bỏ chúng khỏi quá trình phân tích log. Pha này được thực hiện nhờ các câu truy vấn SQL, hoặc sử dụng phân tích xử lý trực tuyến hay cũng có thể nhờ các kỹ thuật hiển thị hóa dữ liệu để lọc và phân tích mẫu.

Dự đoán hành vi người dùng:

Sau khi đã phân tích và lọc các mẫu, những mẫu còn lại sẽ được dùng để đưa ra các kết luận về hành vi người dùng. Với phân tích web log, các hành vi người dùng điển hình gồm: các trang web thường xuyên truy cập, các lĩnh vực quan tâm, thời gian trung bình xem mỗi trang web...

2.2. Thu thập và tiền xử lý

2.2.1. Thu thập log

Log truy nhập có thể được sinh ra ở nhiều vị trí khác nhau trong mạng, do đó có nhiều cách để thu thập log. Log có thể được nhận từ nhiều nguồn khác nhau như: từ file, từ mạng internet, hay từ đầu ra của các ứng dụng khác... Một số nguồn thu thập log cụ thể có thể kể ra như:

- Nhận các sự kiện từ framework Elastic Beats.
- Đọc các kết quả truy vấn từ một cụm Elasticsearch.
- Lấy các sự kiện từ file log.
- Nhận đầu ra của các công cụ dòng lệnh như là một sự kiện.

- Tạo các sự kiện dựa trên các bản tin SNMP.
- Đọc các bản tin syslog.
- Đọc sự kiện từ một TCP socket.
- Đọc sự kiện thông qua UDP.
- Đọc sự kiện thông qua một UNIX socket.

Trong phạm vi luận văn này, ta sử dụng phương pháp lấy các sự kiện để xử lý từ file log, đọc các bản tin syslog và đọc sự kiện thông qua UDP.

2.2.2. *Tiền xử lý và chuẩn hóa*

Quá trình tiền xử lý và chuẩn hóa thực hiện việc làm sạch và hợp nhất dữ liệu từ nhiều nguồn khác nhau, nhận dạng người dùng, nhận dạng phiên làm việc, nhận dạng các pageview... kết hợp dữ liệu clickstream với nội dung trang web hay dữ liệu cá nhân người dùng. Quá trình này cung cấp các dữ liệu tối ưu và thống nhất cho quá trình phân tích web log.

2.2.2.1. Làm sạch và hợp nhất dữ liệu

Ở những trang web lớn, các nội dung log được lưu ở nhiều nguồn khác nhau. Hợp nhất dữ liệu cho phép tổng hợp dữ liệu từ các file log có dạng khác nhau. Trong trường hợp các nguồn dữ liệu này không có cơ chế dùng chung định danh phiên để hợp nhất dữ liệu thì có thể dùng các phương pháp dựa trên kinh nghiệm như dựa trên trường “referrer” trong server log, kết hợp với các phương pháp nhận dạng người dùng và nhận dạng phiên làm việc để có thể thực hiện hợp nhất dữ liệu. Làm sạch dữ liệu nhằm xóa bỏ các tham chiếu không liên quan hoặc không quan trọng cho mục đích phân tích log như: các file CSS của trang web, các file icon, âm thanh của trang web. Quá trình này còn xóa bỏ các trường dữ liệu của file log không cung cấp nhiều thông tin quan trọng cho quá trình phân tích log như phiên bản giao thức HTTP. Ngoài ra, việc làm sạch dữ liệu còn xóa bỏ các tham chiếu là kết quả do các crawler hoặc các công cụ tìm kiếm thực hiện. Có thể duy trì một danh sách các crawler của các công cụ tìm kiếm phổ biến để có thể phát hiện và xóa bỏ kết quả log của chúng. Một phương pháp khác để phát hiện các crawler là dựa vào giao thức hoạt động của chúng, đó là bắt đầu phiên làm việc trên một website, nó đầu tiên sẽ truy cập vào file

“robot.txt” của trang web. Dựa vào đặc điểm này, ta có thể xóa bỏ các phiên làm việc của crawler trên website.

2.2.2.2. Nhận dạng người dùng

Trong trường hợp website truy cập không có các cơ chế xác thực thì phương pháp dùng để phân biệt các người dùng truy cập là dựa vào cookie. Phương pháp này cho kết quả với độ chính xác cao, tuy nhiên do các lo ngại tính riêng tư nên không phải tất cả các người dùng đều cho phép trình duyệt lưu cookie.

Nếu chỉ dùng địa chỉ IP thì không đủ để nhận dạng người dùng riêng biệt. Nguyên nhân chủ yếu do các ISP proxy server sẽ gán lại địa chỉ IP cho người dùng sau một khoảng thời gian nhất định. Ngoài ra, có thể có nhiều người dùng trong một mạng LAN sẽ sử dụng chung một địa chỉ public IP. Vì vậy, trường hợp hai lần truy cập khác nhau tuy có cùng địa chỉ IP nhưng lại từ hai người dùng khác nhau là hoàn toàn có thể xảy ra.

Để tăng tính chính xác của việc nhận dạng người dùng dựa trên địa chỉ IP, ta có thể kết hợp thêm các thông tin khác nhau như user agent hay refferer.

Bảng 2.1 mô tả một ví dụ về nhận dạng người dùng sử dụng kết hợp địa chỉ IP và user agent. Bảng 2.2, 2.3, 2.4 cho kết quả sau khi nhận dạng được người dùng riêng biệt.

Bảng 2.1: Kết hợp địa chỉ IP và User agent

STT	Địa chỉ IP	URL	Ref	Agent
1	1.2.3.4	A	-	Mozilla; Windows NT
2	1.2.3.4	B	A	Mozilla; Windows NT
3	2.3.4.5	C	-	Mozilla; Linux
4	2.3.4.5	B	C	Mozilla; Linux
5	2.3.4.5	E	C	Mozilla; Linux
6	1.2.3.4	C	A	Mozilla; Windows NT
7	2.3.4.5	D	B	Mozilla; Linux

8	1.2.3.4	A	-	Mozilla; Linux
9	1.2.3.4	E	C	Mozilla; Windows NT
10	1.2.3.4	C	A	Mozilla; Linux
11	1.2.3.4	B	C	Mozilla; Linux
12	1.2.3.4	D	B	Mozilla; Linux
13	1.2.3.4	E	D	Mozilla; Linux
14	1.2.3.4	A	-	Mozilla; Windows NT
15	1.2.3.4	C	A	Mozilla; Windows NT
16	1.2.3.4	F	C	Mozilla; Linux
17	1.2.3.4	F	C	Mozilla; Windows NT
18	1.2.3.4	B	A	Mozilla; Windows NT
19	1.2.3.4	D	B	Mozilla; Windows NT
20	1.2.3.4	B	A	Mozilla; Windows NT

Bảng 2.2: Kết quả nhận dạng được người dùng 1

STT theo thời gian	Địa chỉ IP	URL	Ref
1	1.2.3.4	A	-
2	1.2.3.4	B	A
6	1.2.3.4	C	A
9	1.2.3.4	E	C
14	1.2.3.4	A	-
17	1.2.3.4	F	C
18	1.2.3.4	B	A
19	1.2.3.4	D	B
20	1.2.3.4	B	A

Bảng 2.3: Kết quả nhận dạng được người dùng 2

STT theo thời gian	Địa chỉ IP	URL	Ref
3	2.3.4.5	C	-
4	2.3.4.5	B	C
5	2.3.4.5	E	C
7	2.3.4.5	D	B

Bảng 2.4: Kết quả nhận dạng được người dùng 3

STT theo thời gian	Địa chỉ IP	URL	Ref
8	1.2.3.4	A	-
10	1.2.3.4	C	A
11	1.2.3.4	B	C
12	1.2.3.4	D	B
13	1.2.3.4	E	D
16	1.2.3.4	F	C

2.2.2.3. Nhận dạng phiên làm việc

Quá trình nhận dạng phiên làm việc là phân các bản ghi hoạt động của người dùng thành các phiên, mỗi phiên biểu diễn một lần truy cập website của người dùng đó. Với những website không có cơ chế để xác thực người dùng cũng như là các cơ chế bổ sung khác như nhúng thêm định danh phiên (session id) thì phải dùng các phương pháp dựa trên kinh nghiệm - heuristics methods để nhận dạng phiên làm việc. Ta xem tập các phiên thực tế của người dùng trên website là R . Một phân loại phiên dựa trên kinh nghiệm - sessionization heuristic h được thử để ánh xạ R thành tập hợp các phiên C_h . Thông thường, các phân loại phiên dựa trên kinh nghiệm gồm hai loại chính: dựa vào thời gian hoặc dựa vào cấu trúc website.

Phân loại dựa vào thời gian dựa vào việc ước lượng khoảng thời gian giữa các yêu cầu để phân biệt các phiên liên tiếp. Trong khi phân loại dựa trên cấu trúc của website dựa trên cấu trúc của trang web và trường referrer trong web log để phân biệt các phiên.

Với hai loại trên thì một log của máy chủ web có thể được chia thành các phiên dựa trên các phương pháp phân loại cụ thể như sau:

- h1: Tổng thời gian của một phiên thường không vượt quá một ngưỡng θ nhất định. Cho t_0 là thời gian của yêu cầu đầu tiên trong phiên S , yêu cầu với thời gian là t sẽ được gán vào phiên S nếu nó thỏa mãn: $t - t_0 \leq \theta$.

- h2: Khoảng thời gian mà người dùng xem một trang web thường không quá một giới hạn δ . Với t_1 là thời gian của yêu cầu đã được gán cho phiên S , yêu cầu tiếp theo với thời gian t_2 sẽ được gán cho phiên S nếu như nó thỏa mãn $t_2 - t_1 \leq \delta$.

- h-ref: Một yêu cầu q sẽ được gán cho phiên S nếu trường referrer của q có liên quan đến S . Nếu không thì q được xem như là yêu cầu đầu tiên của một phiên mới. Chú ý rằng, với phương pháp này có thể dẫn tới trường hợp là một yêu cầu q có thể thuộc nhiều phiên khác nhau bởi vì nó có thể cùng lúc liên quan tới nhiều phiên trước đó. Trong trường hợp này, các thông tin khác sẽ được bổ sung để tránh việc mơ hồ khi nhận dạng các phiên. Ví dụ, q có thể được gán cho phiên thỏa mãn điều kiện ở trên và được cập nhật mới gần đây nhất.

Người dùng 1	Time	IP	URL	Ref	Phiên 1	0:01	1.2.3.4	A	-
	0:01	1.2.3.4	A	-		0:09	1.2.3.4	B	A
	0:09	1.2.3.4	B	A		0:19	1.2.3.4	C	A
	0:19	1.2.3.4	C	A		0:25	1.2.3.4	E	C
	0:25	1.2.3.4	E	C	Phiên 2	1:15	1.2.3.4	A	-
	1:15	1.2.3.4	A	-		1:26	1.2.3.4	F	C
	1:26	1.2.3.4	F	C		1:30	1.2.3.4	B	A
	1:30	1.2.3.4	B	A		1:36	1.2.3.4	D	B
	1:36	1.2.3.4	D	B					

Hình 2.2: Một ví dụ về nhận dạng phiên dựa trên thời gian

Hình 2.2 mô tả một ví dụ về nhận dạng phiên dựa trên kinh nghiệm theo phương pháp h1, với $\theta = 30$ phút. Nếu ta áp dụng phương pháp h2 với $\theta = 10$ phút thì kết quả có thể chia thành 3 phiên như sau: A - B - C - E; A và F - B - D.

Người dùng 1	Time	IP	URL	Ref	Phiên 1	0:01	1.2.3.4	A	-
	0:01	1.2.3.4	A	-		0:09	1.2.3.4	B	A
	0:09	1.2.3.4	B	A		0:19	1.2.3.4	C	A
	0:19	1.2.3.4	C	A		0:25	1.2.3.4	E	C
	0:25	1.2.3.4	E	C		1:26	1.2.3.4	F	C
	1:15	1.2.3.4	A	-	Phiên 2				
	1:26	1.2.3.4	F	C		1:15	1.2.3.4	A	-
	1:30	1.2.3.4	B	A		1:30	1.2.3.4	B	A
	1:36	1.2.3.4	D	B		1:36	1.2.3.4	D	B

Hình 2.3: Một ví dụ về nhận dạng phiên dựa trên cấu trúc trang web

Hình 2.3 mô tả một ví dụ về nhận dạng phiên sử dụng phương pháp h-ref có cùng tập dữ liệu đầu vào với ví dụ ở hình 2.2. Trong ví dụ này, với yêu cầu F có thời gian là 1:26 thì sẽ phân làm hai phiên là A-B-C-E và A. Yêu cầu F được thêm vào phiên đầu bởi vì trường Ref của nó là C có liên quan đến phiên 1. Yêu cầu B với thời gian là 1:30 có thể thuộc cả hai phiên, bởi vì trường Ref của nó là A đều liên quan đến cả hai phiên. Trường hợp này, B được thêm vào phiên 2 bởi vì đó là phiên mới nhất được cập nhật.

2.2.2.4. Nhận dạng pageview

Việc nhận dạng các trang người dùng xem - pageview phụ thuộc nhiều vào cấu trúc cũng như là nội dung của trang web. Mỗi pageview có thể được xem là một tập hợp các đối tượng web hay các sự kiện phát sinh. Ví dụ như click vào một đường dẫn, xem một trang sản phẩm, thêm sản phẩm vào giỏ hàng. Với các trang web động, thì một pageview có thể kết hợp các nội dung tĩnh và động được tạo bởi server dựa trên tập các tham số đầu vào.

Ngoài ra, ta có thể xem các pageview như tập hợp các trang, các đối tượng liên quan đến cùng một lĩnh vực nào đó. Ví dụ, với các trang web thương mại điện tử, các pageview có thể tương ứng với các sự kiện phát sinh khác nhau như: xem sản phẩm, đăng ký tài khoản, thay đổi giỏ hàng, thanh toán...

Các thuộc tính cơ bản cần phải có của một pageview bao gồm: pageview id (thường là một URL), loại pageview (ví dụ như: trang chủ, trang sản phẩm, trang thanh toán...) và các metadata khác (ví dụ như các từ khóa hay các thuộc tính của sản phẩm).

2.2.2.5. Hoàn thiện đường dẫn

Một phần quan trọng khác trong quá trình tiền xử lý và chuẩn hóa, thường được thực hiện sau khi nhận dạng phiên làm việc đó là hoàn thiện đường dẫn. Phía máy khách hoặc proxy server có thể lưu lại cache của các trang web mà người dùng truy cập, dẫn đến việc thiếu hoặc sai sót tham chiếu của những trang web này trên server log. Ví dụ, trong cùng một phiên làm việc, người dùng truy cập một trang web A 2 lần thì sau lần đầu truy cập, trang web A được proxy server lưu lại trong cache của nó. Đến lần truy cập thứ hai, yêu cầu gửi đi thì proxy server sẽ trả về cho máy khách trang web A nó đã lưu lại từ trước mà không gửi yêu cầu truy cập lên máy chủ web, điều này dẫn đến yêu cầu truy cập trang web A lần thứ hai không được lưu lại trên server log.

Hình 2.4 mô tả một ví dụ về việc tham chiếu - referrer bị thiếu.

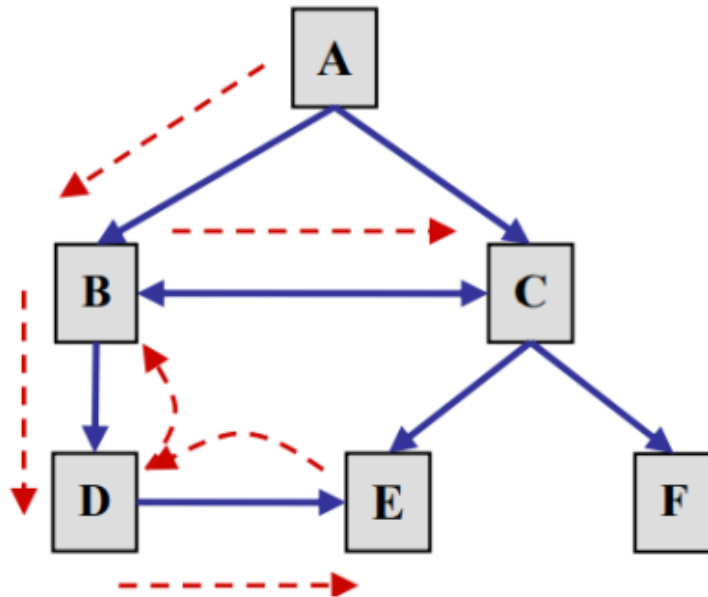
Quá trình truy cập các trang web của người dùng: $A \rightarrow B \rightarrow D \rightarrow E \rightarrow D \rightarrow B \rightarrow C$.

Sau khi truy cập trang web E, người dùng quay trở về trang D rồi trở về trang B, sau đó chuyển sang trang C. Bảng 2.5 mô tả trường URL và Referrer trong file log của server trong trường hợp này:

Bảng 2.5: Ví dụ trường hợp referrer sai

URL	Referrer
A	-
B	A
D	B
E	D
C	B

Việc quay trở về trang D và trang B sẽ không có trong server log do chúng đã được lưu lại trong cache của proxy server giữa máy khách và máy chủ web. Trong file log, ta sẽ thấy sau khi truy cập trang E, người dùng sẽ truy cập vào trang C với tham chiếu là trang B.



Hình 2.4: Ví dụ về tham chiếu sai do cache.

Với vấn đề này, ta có thể sử dụng các phương pháp dựa trên kinh nghiệm kết hợp với cấu trúc của website để phát hiện tham chiếu bị thiếu hoặc sai để đưa ra phương án giải quyết phù hợp.

2.3. Các kỹ thuật phân tích log

2.3.1. Các kỹ thuật nhận dạng mẫu

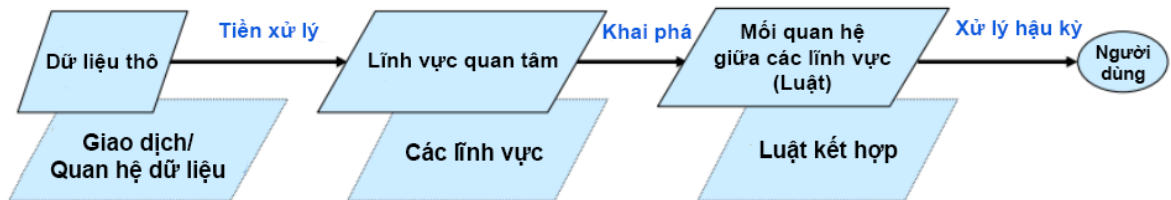
2.3.1.1. Phân tích thống kê

Thống kê là một kỹ thuật phổ biến nhất trong phân tích log. Bằng cách phân tích các file phiên làm việc của người dùng, ta có thể thực hiện các phương pháp thống kê khác nhau như: lấy trung bình, tần suất... với các biến khác nhau như: các trang đã xem, số lượt xem, thời gian xem mỗi trang web. Nhiều công cụ phân tích hiện nay cho kết quả là các báo cáo định kỳ về các thống kê của trang web như: các trang web được truy cập nhiều nhất, thời gian trung bình xem một trang web, số lượt truy cập trung bình một trang web...

Loại phân tích thống kê này có nhiều thông tin hữu ích cho cải thiện hiệu năng của hệ thống hay cho việc marketing.

2.3.1.2. Luật kết hợp

Phương pháp này nhằm phát hiện ra các luật kết hợp giữa các thành phần dữ liệu trong cơ sở dữ liệu. Mẫu đầu ra của giải thuật khai phá dữ liệu là tập luật kết hợp tìm được.



Hình 2.5: Quá trình sử dụng luật kết hợp

Hình 2.5 mô tả cách ta có thể sử dụng luật kết hợp. Có thể lấy một ví dụ đơn giản về luật kết hợp như sau: Phân tích cơ sở dữ liệu bán hàng nhận được thông tin về những khách hàng mua card màn hình cũng có khuynh hướng mua quạt tản nhiệt trong cùng lần mua được miêu tả trong luật kết hợp sau:

“Mua card màn hình → Mua quạt tản nhiệt”

[Độ hỗ trợ: 4%, độ tin cậy: 70%]

Độ hỗ trợ và độ tin cậy là hai độ đo của sự đáng quan tâm của luật. Chúng tương ứng phản ánh sự hữu ích và sự chắc chắn của luật đã khám phá.

Độ hỗ trợ 4% có nghĩa là 4% của tất cả các tác vụ đã phân tích chỉ ra rằng card màn hình và quạt tản nhiệt là đã được mua cùng nhau. Còn độ tin cậy 70% có nghĩa là 70% các khách hàng mua card màn hình cũng mua quạt tản nhiệt.

2.3.1.3. Phân lớp

Bài toán phân lớp là quá trình phân lớp một đối tượng dữ liệu vào một hay nhiều lớp đã cho trước nhờ một mô hình phân lớp (model). Mô hình này được xây dựng dựa trên một tập dữ liệu được xây dựng trước đó có gán nhãn (hay còn gọi là tập huấn luyện). Quá trình phân lớp là *quá trình gán nhãn* cho đối tượng dữ liệu.

Nhiệm vụ của bài toán phân lớp là cần tìm một mô hình phân lớp để khi có dữ liệu mới thì có thể xác định được dữ liệu đó thuộc vào phân lớp nào. Có nhiều bài toán phân lớp dữ liệu như phân lớp nhị phân (binary), phân lớp đa lớp (multiclass), phân lớp đa trị.

Trong phân tích log truy nhập, phân lớp thường dùng để ánh xạ một người dùng vào một lớp hay một loại cụ thể. Việc phân lớp trong phân tích web log có thể được thực hiện nhờ các thuật toán học máy có giám sát như: cây quyết định, thuật toán Naive Bayes, thuật toán K láng giềng gần nhất... Ví dụ, việc phân lớp log máy chủ có thể giúp phân loại được 46% người dùng đặt hàng các sản phẩm ở trang ‘laptop dell’ có độ tuổi từ 18-23 và sống ở miền Bắc là chủ yếu.

2.3.1.4. Phân cụm

Phân cụm là kỹ thuật rất quan trọng trong khai phá dữ liệu, nó thuộc lớp các phương pháp học không giám sát trong học máy. Có rất nhiều định nghĩa khác nhau về kỹ thuật này, nhưng về bản chất ta có thể hiểu phân cụm là các quy trình tìm các nhóm đối tượng đã cho vào các cụm (clusters) sao cho các đối tượng trong cùng một cụm tương tự nhau và các đối tượng khác cụm thì không tương tự nhau.

Mục đích của phân cụm là tìm ra bản chất bên trong các nhóm của dữ liệu. Các thuật toán phân cụm đều sinh ra các cụm. Tuy nhiên, không có tiêu chí nào được xem là tốt nhất để đánh giá hiệu quả của phân tích phân cụm, điều này phụ thuộc vào mục đích của phân cụm như: data reduction, “natural clusters”, “useful” clusters, outlier detection.

Trong phân tích log, có hai kiểu phân cụm có thể được thực hiện: usage cluster và page cluster.

Việc phân cụm những người dùng có mẫu giống nhau có nhiều thông tin giá trị cho marketing và thương mại điện tử. Ví dụ, với những nhóm người nhất định thì có thể đưa ra những gợi ý mua hàng phù hợp với sở thích của nhóm người dùng đó mà thôi.

Mặt khác, phân cụm các trang web giúp nhận biết được các nhóm trang web có nội dung liên quan đến nhau. Thông tin này đặc biệt hữu ích cho các công cụ tìm kiếm, nhờ những thông tin này chúng có thể đưa ra các trang gợi ý phù hợp với truy vấn của người dùng.

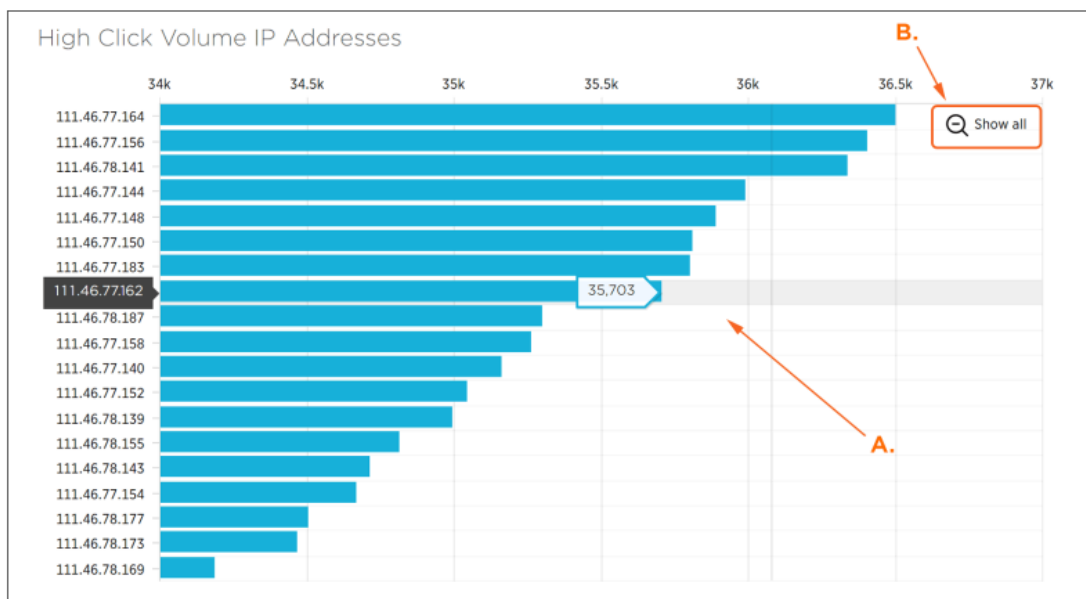
2.3.2. Phân tích mẫu

Đây là bước cuối cùng của quá trình phân tích log truy nhập. Quá trình này nhằm lọc ra những luật hay những mẫu không có nhiều giá trị đã được tạo ra ở bước nhận dạng mẫu.

Có nhiều phương pháp để thực hiện việc này, một trong các phương pháp phổ biến và được sử dụng nhiều nhất là nhờ các câu truy vấn SQL hoặc cũng có thể sử dụng phân tích xử lý trực tuyến - OLAP.

Ngoài ra, ở bước này ta cũng áp dụng các kỹ thuật trực quan hóa dữ liệu như các sơ đồ, biểu đồ thống kê để phục vụ phân tích các mẫu.

Hình 2.6 mô tả một ví dụ sử dụng trực quan hóa dữ liệu. Ta thấy rằng biểu diễn dữ liệu bằng biểu đồ, đồ thị thống kê giúp dễ dàng nhận ra được sự tương quan dữ liệu cũng như nhận ra xu hướng phát triển của dữ liệu.



Hình 2.6: Ví dụ sử dụng trực quan hóa dữ liệu

2.4. Kết luận chương

Chương 2 đã giới thiệu về mô hình xử lý log truy nhập và nêu được các phương pháp thu thập log hiện nay. Chương cũng trình bày các bước tiền xử lý và chuẩn hóa log như: làm sạch và hợp nhất dữ liệu, nhận dạng người dùng, nhận dạng phiên làm việc, nhận dạng pageview và hoàn thiện đường dẫn. Ngoài ra, chương này cũng đã nêu chi tiết các bước để phân tích log bao gồm nhận dạng mẫu và phân tích mẫu.

CHƯƠNG 3 - CÀI ĐẶT VÀ THỬ NGHIỆM

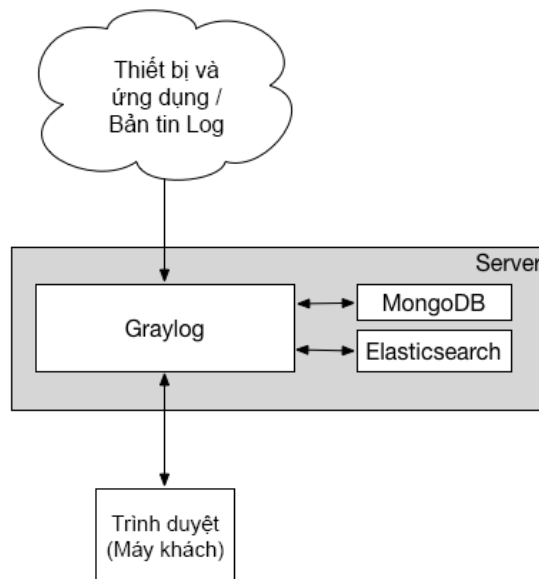
3.1. Giới thiệu nền tảng và công cụ thử nghiệm

3.1.1. Kiến trúc Graylog

Graylog là một trong những công cụ quản lý log mã nguồn mở, phân tích các bản ghi đến, trích xuất dữ liệu quan trọng từ chúng, cung cấp tính năng tìm kiếm và trực quan hóa nhật ký trên giao diện web. Graylog được viết bằng Java và sử dụng một vài công cụ mã nguồn mở như Elasticsearch, MongoDB. Hai công cụ này kết hợp với Graylog và Graylog UI tạo thành một giải pháp quản lý log mạnh mẽ.

Mỗi hệ thống Graylog tối thiểu bao gồm Graylog Server, MongoDB và Elasticsearch. Mỗi thành phần này đều yêu cầu bắt buộc và không thể thay thế bằng bất kỳ công cụ nào khác.

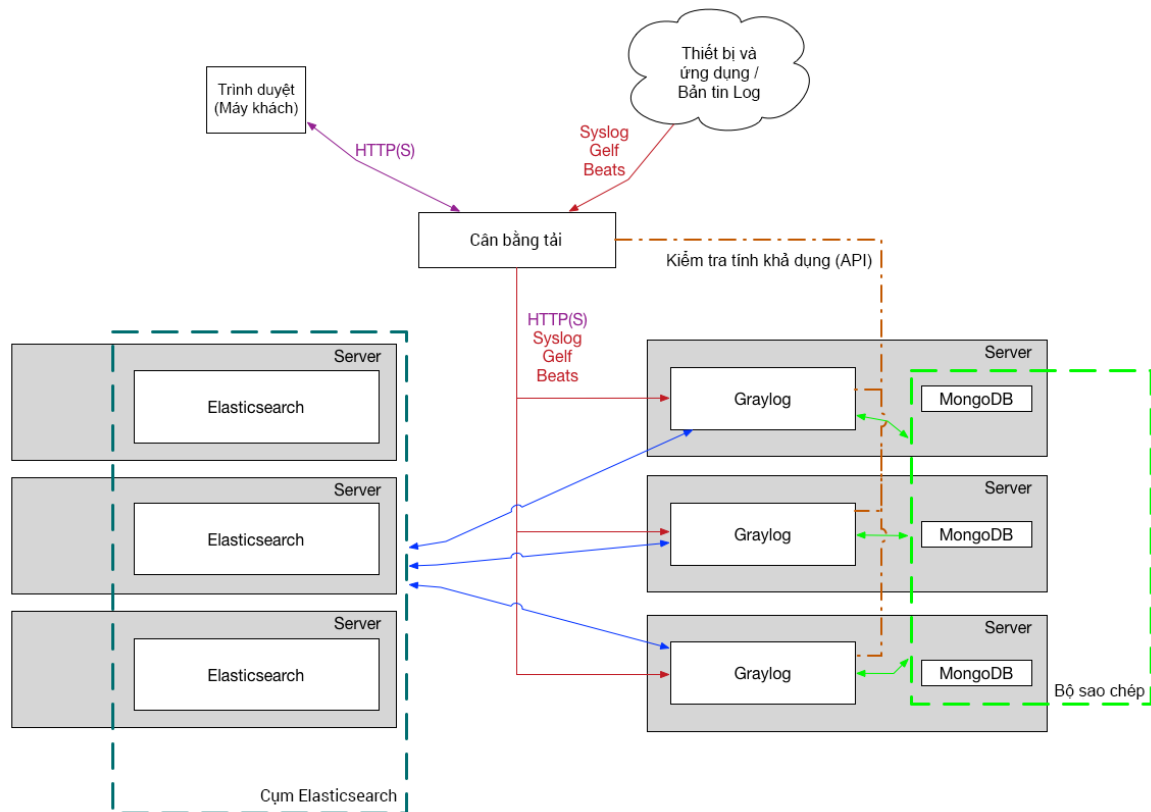
Trong một mô hình triển khai Graylog tối giản, tất cả ba thành phần được cài đặt trên một máy chủ duy nhất. Một thiết lập Graylog tối giản có thể được sử dụng cho các hệ thống nhỏ, ít quan trọng hoặc để thử nghiệm. Hình 3.1 thể hiện kiến trúc Graylog tối giản [11], không có thành phần nào thừa và có thể thiết lập một cách dễ dàng, nhanh chóng.



Hình 3.1: Kiến trúc Graylog tối giản

Trong một hệ thống multi-node đơn giản, các thành phần Graylog và Elasticsearch đều nằm trên các máy chủ riêng của chúng. Hầu hết, MongoDB đều được cài đặt trên cùng một máy chủ với Graylog vì nó được sử dụng chủ yếu cho thông tin cấu hình ứng dụng. Tải trên MongoDB thấp nên nó thường không cần đến máy chủ riêng.

Đối với môi trường lớn hơn, hoặc khi yêu cầu tính khả dụng cao, Graylog có thể được triển khai với cấu hình multi-node phức tạp. Cả Graylog và Elasticsearch đều có thể được nhóm lại để cung cấp khả năng phục hồi trong trường hợp lỗi nút. Hệ thống multi-node thường được triển khai để xử lý một khối lượng lớn các bản ghi log. Hình 3.2 thể hiện mô hình triển khai hệ thống multi-node của Graylog [11].



Hình 3.2: Kiến trúc Multi-Node Graylog

Thiết kế multi-node phức tạp sẽ được triển khai cho các môi trường hoạt động lớn hơn. Nó bao gồm hai hoặc nhiều nút Graylog phía sau bộ cân bằng tải có nhiệm vụ phân phối tải xử lý. Bộ cân bằng tải có thể ping các nút Graylog qua HTTP trên

Graylog REST API để kiểm tra xem chúng còn sống hay không và loại các nút chết ra khỏi cụm.

Có một vài quy tắc chung khi mở rộng tài nguyên cho Graylog như sau:

- Các nút Graylog nên tập trung vào CPU. Chúng cũng phục vụ giao diện web cho người dùng.
- Các nút Elasticsearch nên có càng nhiều RAM càng tốt và các ổ đĩa cứng nhanh nhất có thể. Tất cả mọi thứ phụ thuộc vào tốc độ I/O ở đây.
- MongoDB lưu trữ metadata và thông tin về cấu hình nên không cần nhiều tài nguyên.

Các bản tin được nhập vào chỉ được lưu trữ trong Elasticsearch. Nếu bị mất dữ liệu trong cụm Elasticsearch thì các bản tin sẽ bị mất, trừ khi đã tạo bản sao lưu trước đó.

3.1.2. Các thành phần của Graylog

Graylog bao gồm bốn thành phần chính, đó là Graylog UI, Graylog Server, MongoDB và ElasticSearch.

GRAYLOG	GRAYLOG UI	<ul style="list-style-type: none"> - cung cấp giao diện web cho người dùng - cung cấp khả năng tìm kiếm và phân tích - cung cấp giao diện cho môi trường cấu hình Graylog
	GRAYLOG SERVER	<ul style="list-style-type: none"> - chứa công cụ xử lý log - tích hợp tất cả các thành phần của Graylog
	MongoDB	<ul style="list-style-type: none"> - được sử dụng để lưu trữ dữ liệu cấu hình - chứa metadata, chẳng hạn như thông tin người dùng hoặc cấu hình luồng
	ElasticSearch	<ul style="list-style-type: none"> - dùng để lưu trữ bản tin - cung cấp công cụ tìm kiếm mạnh mẽ và nhanh chóng

Hình 3.3: Các thành phần và tính năng của Graylog

Elasticsearch là một công cụ tìm kiếm mã nguồn mở rất mạnh và có khả năng mở rộng cao. Có thể tìm kiếm, phân tích và lưu trữ một lượng lớn dữ liệu và nó

hoạt động như một công cụ phân tích gần như thời gian thực. Có nghĩa là có một độ trễ nhỏ giữa thời gian khi dữ liệu được lập chỉ mục và khi chúng có sẵn để tìm kiếm. Elasticsearch lưu trữ các chỉ mục theo định dạng tinh vi được tối ưu hóa cho tìm kiếm toàn văn bản. Chỉ mục là tập hợp dữ liệu, trong Elasticsearch được gọi là tài liệu, với các đặc điểm tương tự. Graylog sử dụng một cụm Elasticsearch chuyên dụng có thể bao gồm nhiều nút. Tất cả các nút Elasticsearch được định nghĩa trong file cấu hình chính của Graylog: `/etc/graylog/server/server.conf`. Graylog cũng hỗ trợ phát hiện nút tự động để có danh sách các nút Elasticsearch có sẵn. Cụm Elasticsearch được Graylog sử dụng có thể bao gồm nhiều nút trong đó một nút là một thể hiện của Elasticsearch. Một nút có thể lưu trữ dữ liệu hoặc bản sao dữ liệu. Mục đích của việc lưu trữ các bản sao dữ liệu trong chuyển đổi dự phòng là trong trường hợp nút chính bị hỏng, nút lưu trữ bản sao được đẩy lên vai trò của nút chính và không có dữ liệu nào bị mất. Các nút mới được thêm vào cụm Elasticsearch để tăng hiệu suất. Điều đó có nghĩa rằng hiệu suất của máy chủ Graylog là được đánh giá cao bởi hiệu quả của cụm Elasticsearch.

MongoDB là một cơ sở dữ liệu NoSQL lưu trữ dữ liệu trong theo cấu trúc có định dạng JSON. Graylog sử dụng MongoDB để lưu trữ các thông tin cấu hình, metadata và web UI, chẳng hạn như người dùng, quyền, luồng, chỉ mục, thông tin cấu hình, v.v. MongoDB không lưu trữ dữ liệu log, cũng không phải chạy trên một máy chủ chuyên dụng bởi vậy nó không có tác động lớn đến máy chủ Graylog.

Graylog User Interface cho phép truy cập vào giao diện web trực quan, cung cấp khả năng tìm kiếm, phân tích và làm việc với dữ liệu tổng hợp. Graylog UI tìm nạp tất cả dữ liệu thông qua HTTP từ Graylog REST API. API được sử dụng làm kênh giao tiếp chính giữa máy chủ UI và máy chủ Graylog. Ưu điểm là với dữ liệu từ REST API, có thể xây dựng lối vào riêng theo nhu cầu.

Graylog Server là một thành phần chịu trách nhiệm nhận dữ liệu từ máy khách và mục đích chính của nó là tích hợp và giao tiếp với các thành phần khác.

3.1.3. Các tính năng của Graylog

Một nhật ký được nhận bởi máy chủ Graylog, sau đó được xử lý bởi Bộ lọc bản tin, là bộ xử lý bản tin chịu trách nhiệm phân tích cú pháp, thay đổi và thiết lập các trường tĩnh cho một nhật ký hợp lệ. Log được thay đổi theo các quy tắc được xác định trước và được định tuyến thành các danh mục được gọi là Luồng. Đối với các luồng khác nhau, chúng ta có thể xác định các quy tắc dựa trên các quy tắc cụ thể. Trên mỗi luồng, có một bộ chỉ mục khác được áp dụng. Bộ chỉ mục kiểm soát các bản tin được lưu trữ trong Elasticsearch như thế nào, ví dụ, số lượng các phân đoạn Elasticsearch hoặc các chính sách xoay vòng và lưu trữ. Từ luồng, nhật ký được chuyển tiếp đến một hệ thống khác hoặc lưu trữ cục bộ trên máy chủ Graylog.

3.1.3.1. Thu thập log

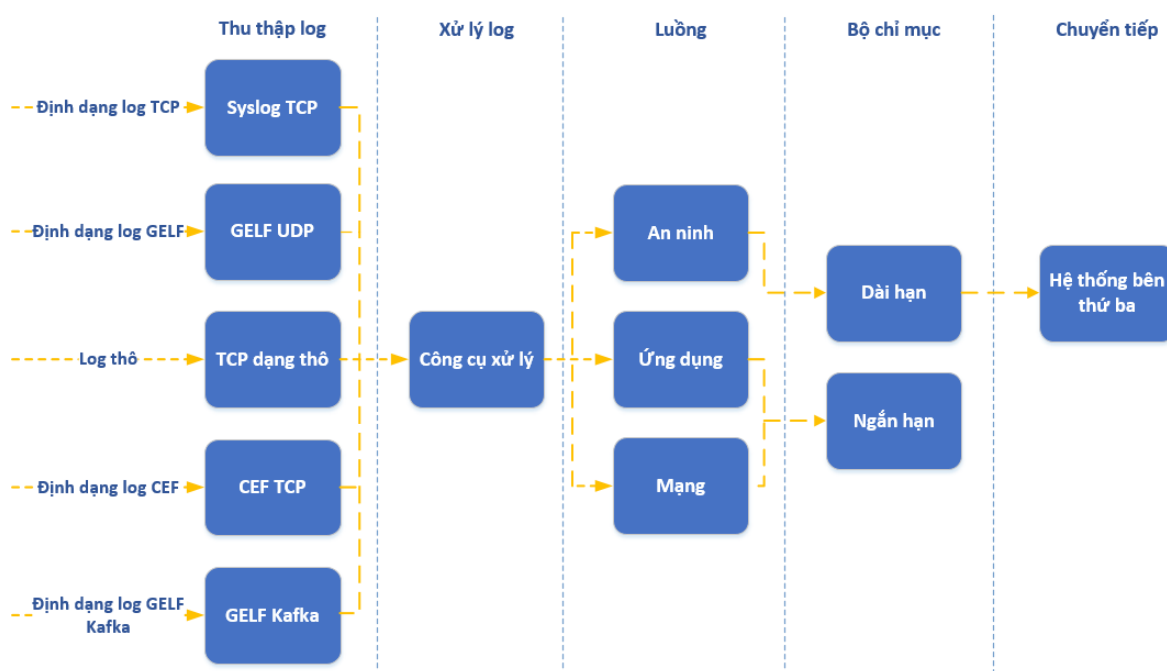
Graylog hỗ trợ ba loại nguồn dữ liệu đầu vào khác nhau:

- *Các giao thức và định dạng chuẩn:* Syslog là giao thức được sử dụng phổ biến nhất để gửi bản tin sự kiện. Giao thức Syslog có thể được sử dụng để ghi lại các loại sự kiện khác nhau và nó được hỗ trợ bởi một loạt các thiết bị. Bản tin sự kiện có thể được tạo bằng định dạng này bằng cách sử dụng rsyslog, đây là trình quản lý nhật ký mặc định trong các bản phân phối Linux phổ biến hoặc công cụ syslog-ng. Graylog cũng có thể nhận log thô, ở dạng thô hoặc ở định dạng JSON. Graylog có hỗ trợ sẵn cho các giao thức truyền tải TCP, UDP và hàng đợi vận chuyển Apache Kafka và RabbitMQ (AMQP2). Việc thu thập thông điệp nhật ký nội bộ của Graylog không được hỗ trợ theo mặc định nhưng có thể sử dụng plugin của bên thứ ba cho phép thu thập log nội bộ.

- *Công cụ thu thập bên thứ ba:* Graylog hỗ trợ một hệ thống gọi là *Bộ thu thập Graylog*, một dịch vụ cho các hệ thống Windows và Linux được sử dụng như một bộ thu thập log. Bộ thu thập log được cài đặt trên máy chủ chuyển tiếp các bản ghi nhật ký hoặc Eventlog tới máy chủ Graylog. Mỗi bộ thu thập chứa thông tin cấu hình như địa chỉ máy chủ Graylog, định dạng log cho phép. Bộ thu thập sử dụng NXLog, Filebeat hoặc Winlogbeat để thu thập nhật ký máy chủ. Bộ thu thập chuyển tiếp các

bản tin đã thu thập đến một địa chỉ IP và cổng máy chủ Graylog được cho phép, trên đó máy chủ Graylog đang hoạt động.

- *GELF*: Graylog có định dạng log riêng được gọi là Graylog Extended Log Format (GELF), là một chuỗi JSON nên được sử dụng đặc biệt để chuyển tiếp và xử lý các log ứng dụng. GELF hỗ trợ nhiều ngôn ngữ lập trình và có khả năng ghi lại mọi sự khác biệt được tạo ra bởi một ứng dụng cụ thể. GELF cung cấp cấu trúc nén và tối ưu hóa cho các mục đích của Graylog.



Hình 3.4: Ví dụ về vòng đời của log trong Graylog

3.1.3.2. Xử lý

Xử lý log đã nhận được thực hiện trong *Luồng Graylog*. Luồng là các nhóm ảo của log cho phép phân loại log theo các quy tắc được chỉ định. Nghĩa là có thể nhóm các bản ghi theo các trường khác nhau, chẳng hạn như mức độ nghiêm trọng của log hoặc địa chỉ IP nguồn. Các Luồng hỗ trợ hai loại quy tắc khác nhau. Đầu tiên là khi một bản tin phải khớp với tất cả các quy tắc được đưa ra (logic AND) hoặc khi một bản tin phải khớp với ít nhất một trong các quy tắc được đưa ra (logic OR). Log đến được xử lý một cách trực tiếp trong *Bộ lọc bản tin*. Bộ lọc bản tin là chuỗi chịu trách nhiệm phân tích log, thiết lập các trường tĩnh và gán log cho các luồng thích

hợp. Hệ thống này phân tích các bản tin bởi một thành phần được gọi là *Bộ trích xuất*, trích xuất các trường tĩnh từ một bản tin log. Cấu trúc của mỗi định dạng log là khác nhau, đó là lý do tại sao các trình giải nén di động có thể được sử dụng cho các định dạng khác nhau. *Bộ chỉ mục* là một sự tổng hợp kiểm soát cách mà log được lưu trữ trên máy chủ Graylog. Nó định nghĩa các chính sách luân phiên, lưu trữ và cấu hình bộ nhớ Elasticsearch. Có thể thiết lập các chính sách luân chuyển khác nhau dựa trên kích thước luồng, thời gian hoặc số lượng bản tin và các chính sách lưu trữ được sử dụng để xóa log cũ nhằm ngăn không cho sử dụng quá nhiều dung lượng ổ đĩa. Mỗi luồng có một tập chỉ mục riêng của nó, tức là đối với các nhóm log khác nhau, có thể sử dụng các chính sách khác nhau.

3.1.3.3. Chuyển tiếp và lưu trữ

Graylog có thể chuyển tiếp log tới các hệ thống khác hoặc lưu chúng cục bộ trên máy chủ. Graylog hỗ trợ chuyển tiếp log tới các hệ thống khác như SIEM hoặc một máy chủ Linux khác và định dạng được hỗ trợ duy nhất là GELF. Lưu trữ các bản tin là điều cần thiết cho các mục đích phân tích. Nó rất quan trọng nếu chúng ta muốn phân tích log trong các khoảng thời gian khác nhau và so sánh kết quả từ chúng. Hay nếu chúng ta muốn tìm kiếm, đồng thời hiển thị và theo dõi các thay đổi theo thời gian. Đối với những trường hợp như vậy, log phải có sẵn. Log cũ hơn trong một thời gian nhất định không bắt buộc phải có sẵn ở bất cứ lúc nào nên được lưu trữ. Chính sách lưu trữ được hiểu cho mỗi bộ chỉ mục. Graylog chỉ có thể lưu trữ log cục bộ và không lưu trữ log trên các hệ thống bên ngoài khác như cơ sở dữ liệu hoặc NAS.

Hình 3.4 cho thấy các quá trình khác nhau, bao gồm thu thập và xử lý log, luồng, chỉ mục và chuyển tiếp mà hệ thống Graylog cung cấp. Trên hình, chúng ta có thể thấy 5 đầu vào log là Syslog TCP, GELF UDP, Log thô, CEF TCP, GELF Kafka được xác nhận để nhận log. Ta cũng thấy các ví dụ về 3 luồng là an ninh, ứng dụng, mạng và 2 bộ chỉ mục được dùng để ghi lại, lưu trữ log dài hạn và ngắn hạn. Ta thấy có thể chuyển tiếp log cho hệ thống của bên thứ ba. Đây là một ví dụ về cách hệ thống Graylog có thể được kết hợp.

3.2. Cài đặt

3.2.1. Các mô đun thu thập log

3.2.1.1. Cài đặt Rsyslog trên máy chủ Linux

Trên máy các máy chủ chạy Linux ta cài đặt ứng dụng Rsyslog để thu thập log từ máy chủ. Trong đề tài này, ta tiến hành cài đặt Rsyslog trên máy chủ web chạy CentOS 7.

- Tiến hành cài đặt Rsyslog:

```
yum install rsyslog
```

- Kiểm tra trạng thái:

```
systemctl status rsyslog.service
```

- Cấu hình file “rsyslog.conf”

```
vi /etc/rsyslog.conf
```

```
$ModLoad imudp
```

```
$UDPServerRun 514
```

```
$ModLoad imtcp
```

```
$InputTCPServerRun 514
```

```
*.*@10.99.3.47:514
```

Trong đó 10.99.3.47 là địa chỉ IP của máy chủ sẽ cài đặt Graylog Server

```
# For more information see /usr/share/doc/rsyslog-*/rsyslog_conf.html
# If you experience problems, see http://www.rsyslog.com/doc/troubleshooting.html

#### MODULES ####

# The imjournal module below is now used as a message source instead of
$ModLoad imuxsock # provides support for local system logging (e.g. via journal)
$ModLoad imjournal # provides access to the systemd journal
$ModLoad imklog # reads kernel messages (the same are read from journal)
$ModLoad immark # provides --MARK-- message capability

# Provides UDP syslog reception
$ModLoad imudp
$UDPServerRun 514
*.*@10.99.3.47:514_

# Provides TCP syslog reception
$ModLoad imtcp
$InputTCPServerRun 514
```

Hình 3.5: Cấu hình Rsyslog trên Linux

- Khởi động lại Rsyslog

```
systemctl restart rsyslog.service
```

```
systemctl status rsyslog.service
```

- Kiểm tra port 514 có hoạt động:

```
yum provides */netstat
```

```
yum install net-tools
```

```
netstat -antup | grep 514
```

```

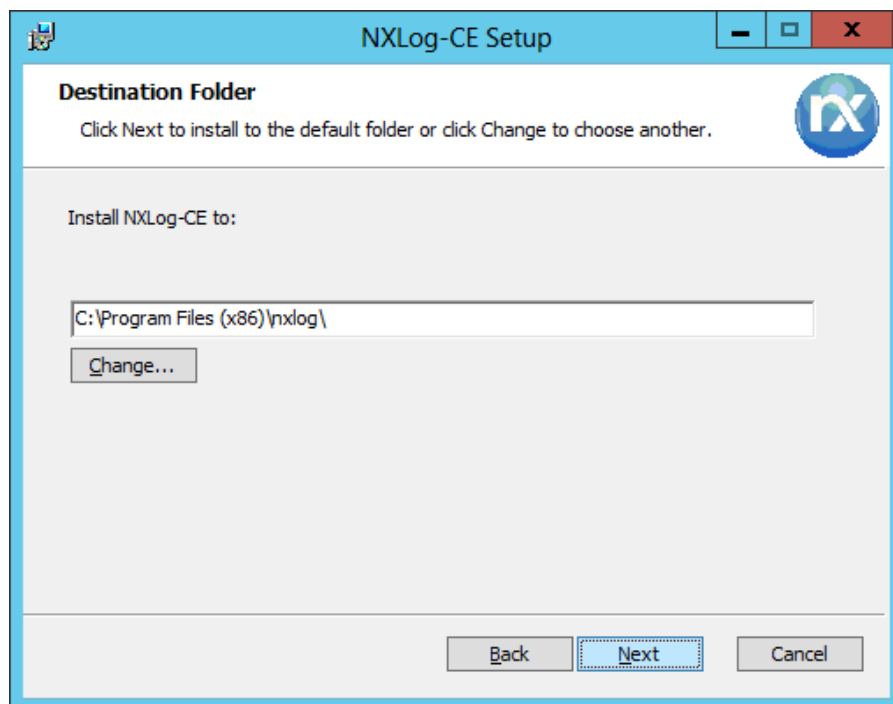
[root@localhost ~]# netstat -antup | grep 514
tcp        0      0 0.0.0.0:514          0.0.0.0:*            LISTEN      892/rsyslogd
tcp6       0      0 :::514              :::*                  LISTEN      892/rsyslogd
tcp6       0      0 127.0.0.1:37514     127.0.0.1:9200       TIME_WAIT   -
udp        0      0 0.0.0.0:514          0.0.0.0:*            892/rsyslogd
udp6       0      0 :::514              :::*                  892/rsyslogd
[root@localhost ~]#

```

Hình 3.6: Kiểm tra port mà Rsyslog sử dụng

3.2.1.2. Cài đặt NXLog trên máy chủ Windows

Trên các máy chủ chạy hệ điều hành Windows, ta cài đặt phần mềm NXLog để thu thập log từ máy chủ.



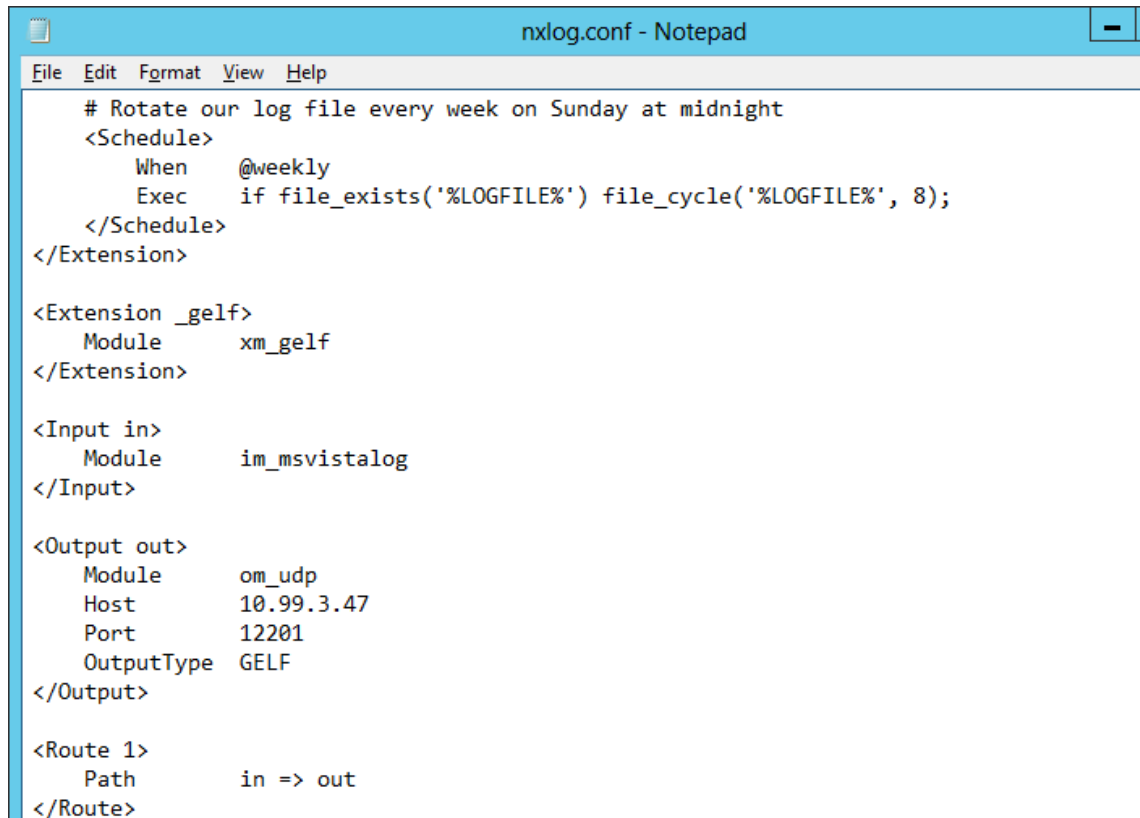
Hình 3.7: Cài đặt NXLog trên Windows Server

- Tải NXLog từ địa chỉ:

<https://nxlog.co/products/nxlog-community-edition/download>

- Cài đặt NXLog

- Cấu hình NXLog để gửi bản tin tới máy chủ Graylog:



```

# Rotate our log file every week on Sunday at midnight
<Schedule>
    When    @weekly
    Exec    if file_exists('%LOGFILE%') file_cycle('%LOGFILE%', 8);
</Schedule>
</Extension>

<Extension _gelf>
    Module    xm_gelf
</Extension>

<Input in>
    Module    im_msvistalog
</Input>

<Output out>
    Module    om_udp
    Host      10.99.3.47
    Port      12201
    OutputType GELF
</Output>

<Route 1>
    Path      in => out
</Route>

```

Hình 3.8: Cấu hình NXLog trên Windows Server

Thêm cấu hình vào file “C:\Program Files (x86)\nxlog\conf\nxlog.conf”

```

<Extension _gelf>
    Module    xm_gelf
</Extension>

<Input in>
    Module    im_msvistalog
</Input>

<Output out>
    Module    om_udp
    Host      10.99.3.47

```

```

    Port      12201
    OutputType GELF
</Output>
<Route 1>
    Path      in => out
</Route>

```

- Run services.msc, Start the nxlog service

3.2.2. Hệ thống xử lý và phân tích log

Đối với hệ thống xử lý và phân tích log, ta cài đặt các thành phần Elasticsearch, MongoDB, Graylog Server trên cùng một máy chủ với cấu hình như sau:

- CentOS 7 (64bit)
- 4 GB RAM
- 40 GB HDD

3.2.2.1. Cài đặt Elasticsearch

- Cài đặt Java trước khi tiến hành cài đặt Elasticsearch

```
yum install java
```

- Thêm GPG signing key cho Elasticsearch:

```
rpm --import https://artifacts.elastic.co/GPG-KEY-elasticsearch
```

- Thêm Elasticsearch repository bằng câu lệnh:

```
vi /etc/yum.repos.d/elasticsearch.repo
```

```
[elasticsearch-5.x]
```

```
name=Elasticsearch repository for 5.x packages
```

```
baseurl=https://artifacts.elastic.co/packages/5.x/yum
```

```
gpgcheck=1
```

```
gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch
```

```
enabled=1
```

```
autorefresh=1
```

```
type=rpm-md
```

```
[elasticsearch-5.x]
name=Elasticsearch repository for 5.x packages
baseurl=https://artifacts.elastic.co/packages/5.x/yum
gpgcheck=1
gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch
enabled=1
autorefresh=1
type=rpm-md
```

Hình 3.9: Tạo repository cho Elasticsearch

- Cài đặt Elasticsearch:

```
yum install elasticsearch
```

- Khởi động lại dịch vụ Elasticsearch:

```
chkconfig --add elasticsearch
```

```
systemctl daemon-reload
```

```
systemctl start elasticsearch.service
```

```
systemctl enable elasticsearch.service
```

- Kiểm tra healthy của Elasticsearch:

```
curl -XGET 'http://localhost:9200/_cluster/health?pretty=true'
```

```
systemctl status elasticsearch.service
```

```
[root@localhost ~]# systemctl status elasticsearch
■ elasticsearch.service - Elasticsearch
   Loaded: loaded (/usr/lib/systemd/system/elasticsearch.service; enabled;
   Active: active (running) since Mon 2018-11-12 18:50:55 +07; 43s ago
     Docs: http://www.elastic.co
   Process: 890 ExecStartPre=/usr/share/elasticsearch/bin/elasticsearch-sy
, status=0/SUCCESS)
  Main PID: 901 (java)
    CGroup: /system.slice/elasticsearch.service
            └─901 /bin/java -Xms2g -Xmx2g -XX:+UseConcMarkSweepGC -XX:CMSI

Nov 12 18:50:55 localhost.localdomain systemd[1]: Starting Elasticsearch.
Nov 12 18:50:55 localhost.localdomain systemd[1]: Started Elasticsearch.
[root@localhost ~]# _
```

Hình 3.10: Kiểm tra trạng thái Elasticsearch sau khi cài đặt

3.2.2.2. Cài đặt MongoDB

- Thêm repository cho MongoDB:

```
vi /etc/yum.repos.d/mongodb-org-3.2.repo
```

```
[mongodb-org-3.2]
```

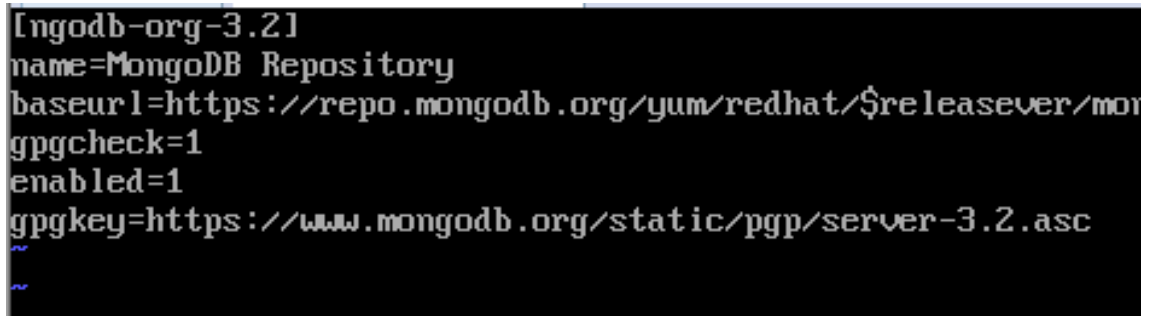
```
name=MongoDB Repository
```

```
baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mongodb-  
org/3.2/x86_64/
```

```
gpgcheck=1
```

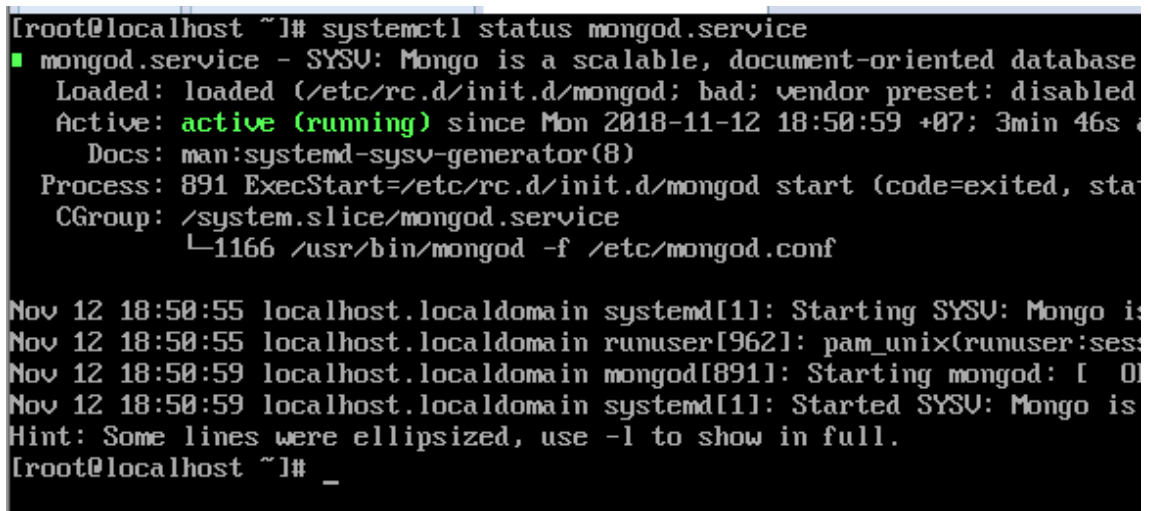
```
enabled=1
```

```
gpgkey=https://www.mongodb.org/static/pgp/server-3.2.asc
```



```
[mongodb-org-3.2]
name=MongoDB Repository
baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mo
gpgcheck=1
enabled=1
gpgkey=https://www.mongodb.org/static/pgp/server-3.2.asc
~
~
```

Hình 3.11: Tạo repository cho MongoDB



```
[root@localhost ~]# systemctl status mongod.service
■ mongod.service - SYSV: Mongo is a scalable, document-oriented database
   Loaded: loaded (/etc/rc.d/init.d/mongod; bad; vendor preset: disabled)
   Active: active (running) since Mon 2018-11-12 18:50:59 +07; 3min 46s
     Docs: man:systemd-sysv-generator(8)
   Process: 891 ExecStart=/etc/rc.d/init.d/mongod start (code=exited, sta
   CGroup: /system.slice/mongod.service
           └─1166 /usr/bin/mongod -f /etc/mongod.conf

Nov 12 18:50:55 localhost.localdomain systemd[1]: Starting SYSV: Mongo is
Nov 12 18:50:55 localhost.localdomain runuser[962]: pam_unix(runuser:ses
Nov 12 18:50:59 localhost.localdomain mongod[891]: Starting mongod: [ 0
Nov 12 18:50:59 localhost.localdomain systemd[1]: Started SYSV: Mongo is
Hint: Some lines were ellipsized, use -l to show in full.
[root@localhost ~]# _
```

Hình 3.12: Kiểm tra dịch vụ MongoDB sau khi cài đặt

- Cài đặt MongoDB:

```
yum install mongodb-org
```

- Khởi động dịch vụ MongoDB:

```
chkconfig --add mongod
systemctl daemon-reload
systemctl enable mongod.service
systemctl start mongod.service
```

- Kiểm tra dịch vụ MongoDB:

```
systemctl status mongod.service
```

3.2.2.3. Cài đặt Graylog

- Thêm Graylog repository:

```
rpm -Uvh https://packages.graylog2.org/repo/packages/graylog-2.4-
repository_latest.rpm
```

- Cài đặt Graylog Server

```
yum install graylog-server
```

- Cài đặt EPEL repo, cài đặt “pwgen” để tạo secret key cho Graylog:

```
yum install epel-release
```

```
yum install pwgen
```

- Tạo secret key cho Graylog:

```
pwgen -N 1 -s 96 (lưu lại kết quả)
```

- Thiết lập hash password cho root user sử dụng Graylog web server:

```
echo -n password | sha256sum (lưu lại kết quả)
```

- Chỉnh sửa file “server.conf”

```
vi /etc/graylog/server/server.conf
```

```
password_secret =
```

```
root_password_sha2 =
```

```
root_timezone = Asia/Ho_Chi_Minh
```

```
elasticsearch_shards = 1
```

```
elasticsearch_replicas = 0
```

```
rest_listen_uri = http://10.99.3.47:9000/api/
```

```
web_listen_uri = http://10.99.3.47:9000/
```

```
rest_transport_uri = http://10.99.3.47:9000/api/
```



```

root_timezone = Asia/Ho_Chi_Minh

# Set plugin directory here (relative or absolute)
plugin_dir = /usr/share/graylog-server/plugin

# REST API listen URI. Must be reachable by other Graylog server nodes if you use
# When using Graylog Collectors, this URI will be used to receive heartbeat messages
# accessible for all collectors.
rest_listen_uri = http://10.99.3.47:9000/api/

# REST API transport address. Defaults to the value of rest_listen_uri. Except
# i
# is set to a wildcard IP address (0.0.0.0) the first non-loopback IPv4 system
# If set, this will be promoted in the cluster discovery APIs, so other nodes n
# this address and it is used to generate URLs addressing entities in the REST
# _uri)
# You will need to define this, if your Graylog server is running behind a HTTP
# ing
# the scheme, host name or URI.
# This must not contain a wildcard address (0.0.0.0).
rest_transport_uri = http://10.99.3.47:9000/api/

```

Hình 3.13: Cấu hình cho Graylog Server

- Khởi động Graylog Server

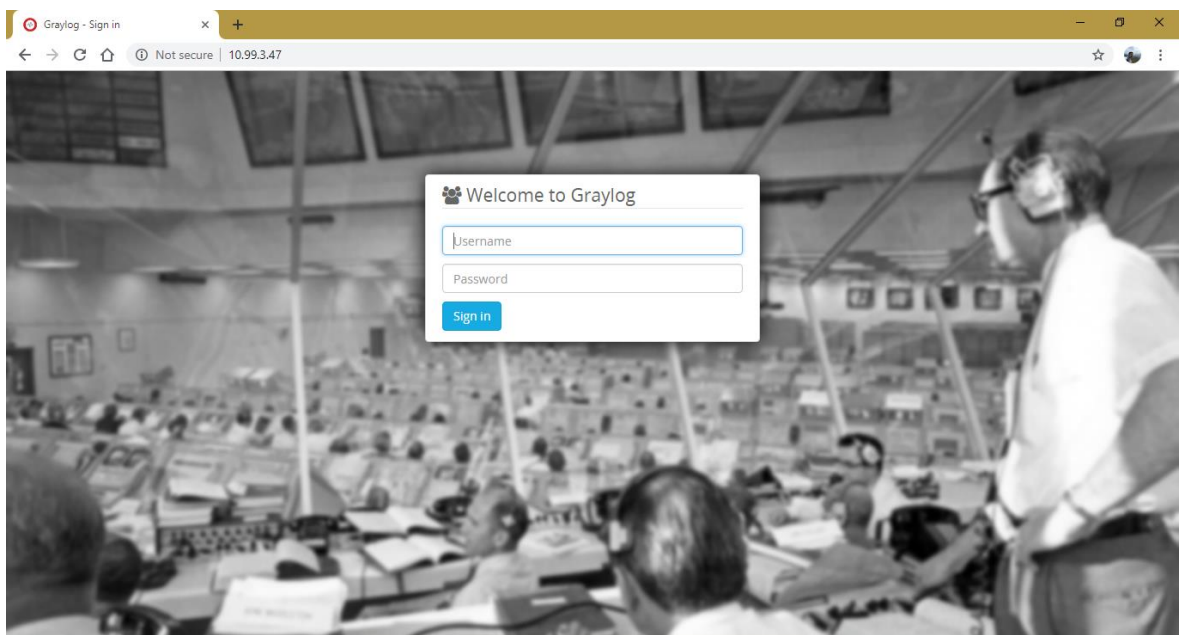
```
chkconfig --add graylog-server
```

```
systemctl daemon-reload
```

```
systemctl start graylog-server.service
```

```
systemctl enable graylog-server.service
```

Sau khi cài đặt thành công, ta có thể truy cập Graylog Web Interface.



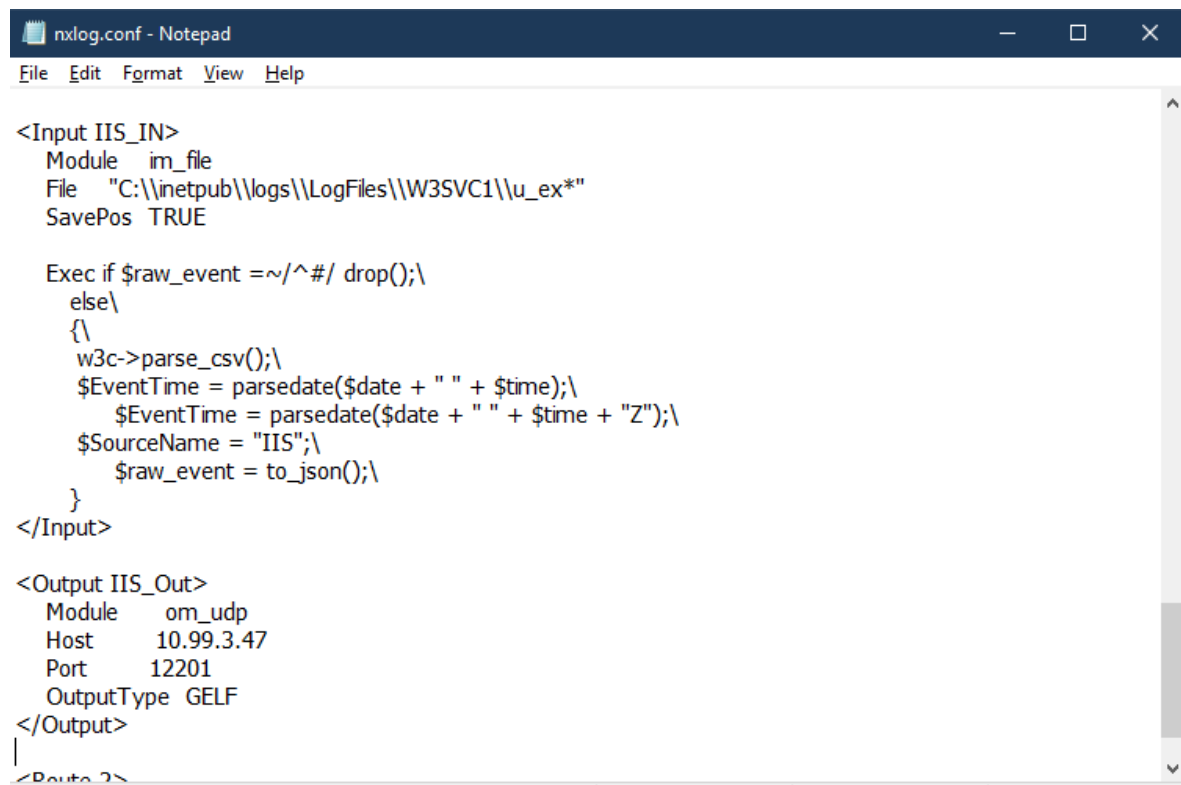
Hình 3.14: Giao diện truy cập Graylog Web Interface

3.3. Các kịch bản thử nghiệm và kết quả

3.3.1. Các kịch bản thử nghiệm

Cài đặt mô đun thu thập log lên một số máy chủ như web server, mail server... để chuyển log về Graylog Server, sau đó quản lý tập trung các nguồn log thông qua Graylog Web Interface:

- Thêm input trong NXLog để thu thập log của máy chủ web Microsoft IIS:



```

nxlog.conf - Notepad
File Edit Format View Help

<Input IIS_IN>
  Module im_file
  File "C:\\inetpub\\logs\\LogFiles\\W3SVC1\\u_ex*"
  SavePos TRUE

  Exec if $raw_event =~ /^#/ drop();\
    else\
    {\
      w3c->parse_csv();\
      $EventTime = parsedate($date + " " + $time);\
      $EventTime = parsedate($date + " " + $time + "Z");\
      $SourceName = "IIS";\
      $raw_event = to_json();\
    }
</Input>

<Output IIS_Out>
  Module om_udp
  Host 10.99.3.47
  Port 12201
  OutputType GELF
</Output>

<Route 2>

```

Hình 3.15: Thêm Microsoft IIS input trong NXLog

- Tạo Input GELF UDP để nhận log định dạng GELF thông qua cổng 12201

Editing Input appliance-gelf-udp


☐ Global

Should this input start on all nodes

Node

f65901da / graylog-beta

On which node should this input start

Title

appliance-gelf-udp

Bind address

0.0.0.0

Address to listen on. For example 0.0.0.0 or 127.0.0.1.

Port

12201

Port to listen on.

Receive Buffer Size (optional)

1048576

The size in bytes of the recvBufferSize for network connections to this input.

Override source (optional)**Hình 3.16: Tạo GELF UDP input**

- Tạo Input Syslog UDP để nhận log từ Syslog qua cổng 514

Editing Input appliance-syslog-udp


☐ Global

Should this input start on all nodes

Node

f65901da / graylog-beta

On which node should this input start

Title

appliance-syslog-udp

Bind address

0.0.0.0

Address to listen on. For example 0.0.0.0 or 127.0.0.1.

Port

514

Port to listen on.

Receive Buffer Size (optional)

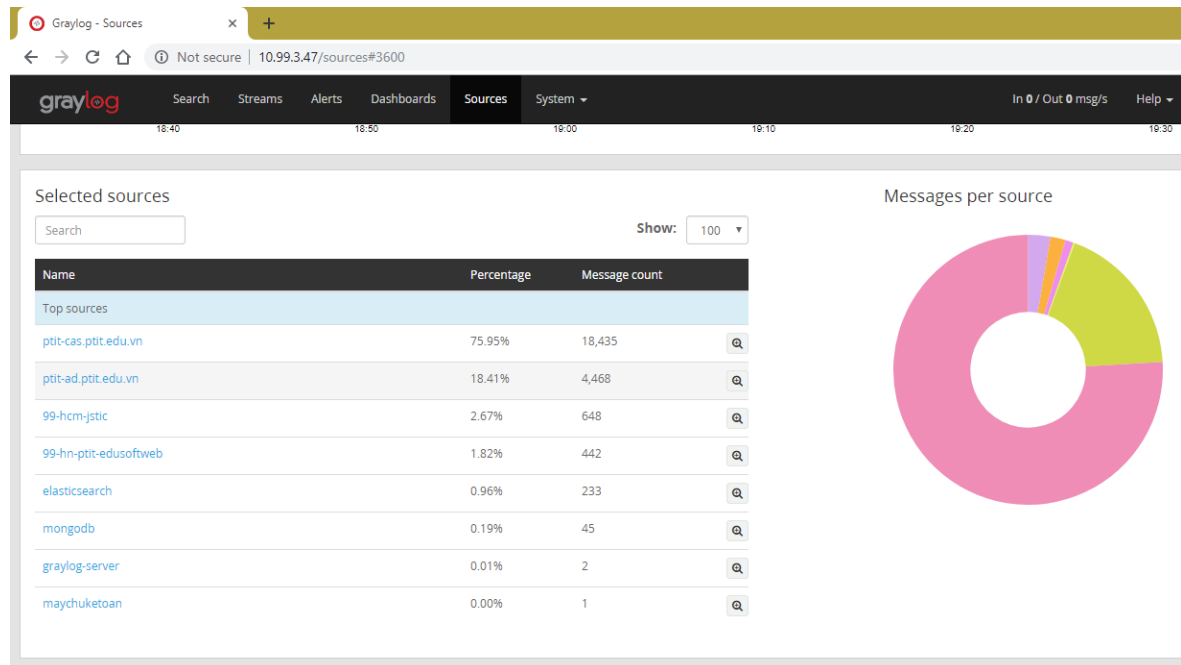
262144

The size in bytes of the recvBufferSize for network connections to this input.

Override source (optional)**Hình 3.17: Tạo Syslog UDP input**

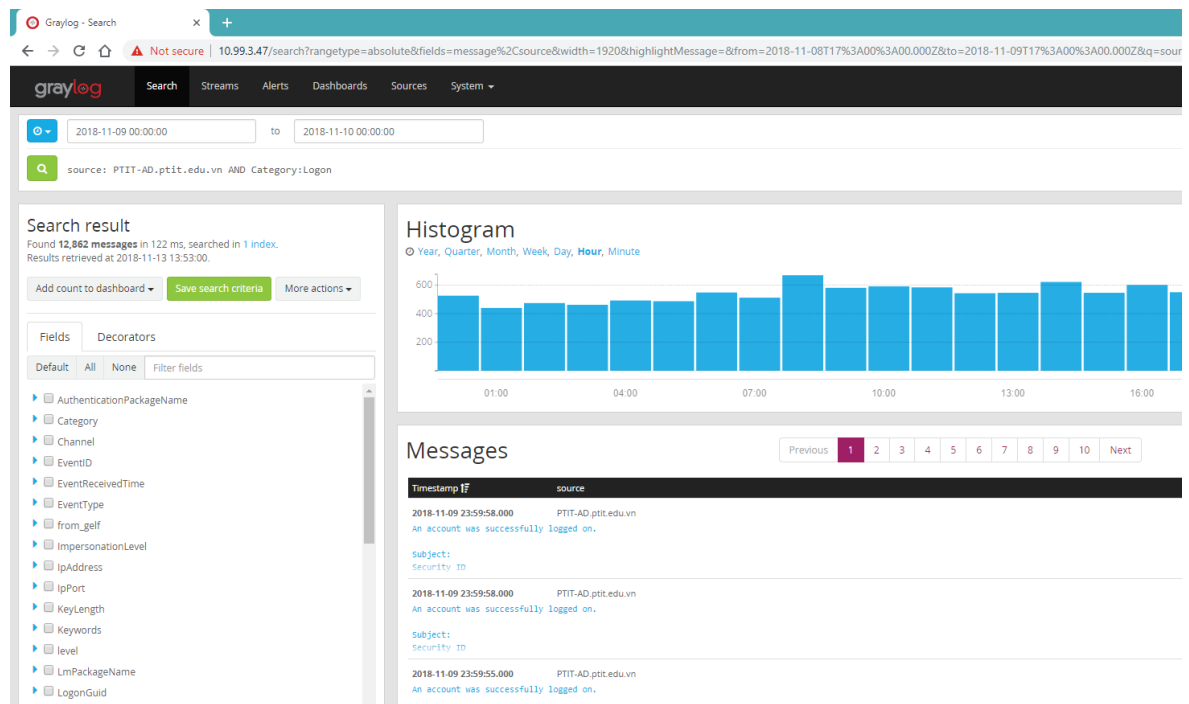
3.3.2. Một số kết quả

- Quản lý tập trung các nguồn máy chủ cung cấp log:



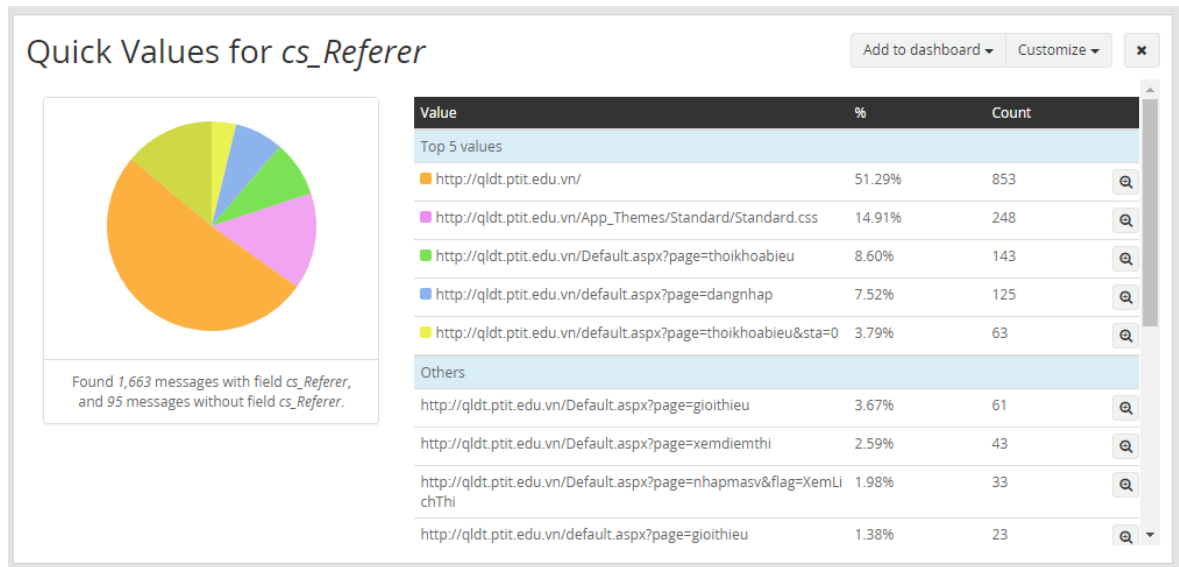
Hình 3.18: Quản lý các nguồn cung cấp log trên Graylog

- Xem log theo thời gian thực, quản lý log với công cụ tìm kiếm mạnh mẽ sử dụng Elasticsearch.

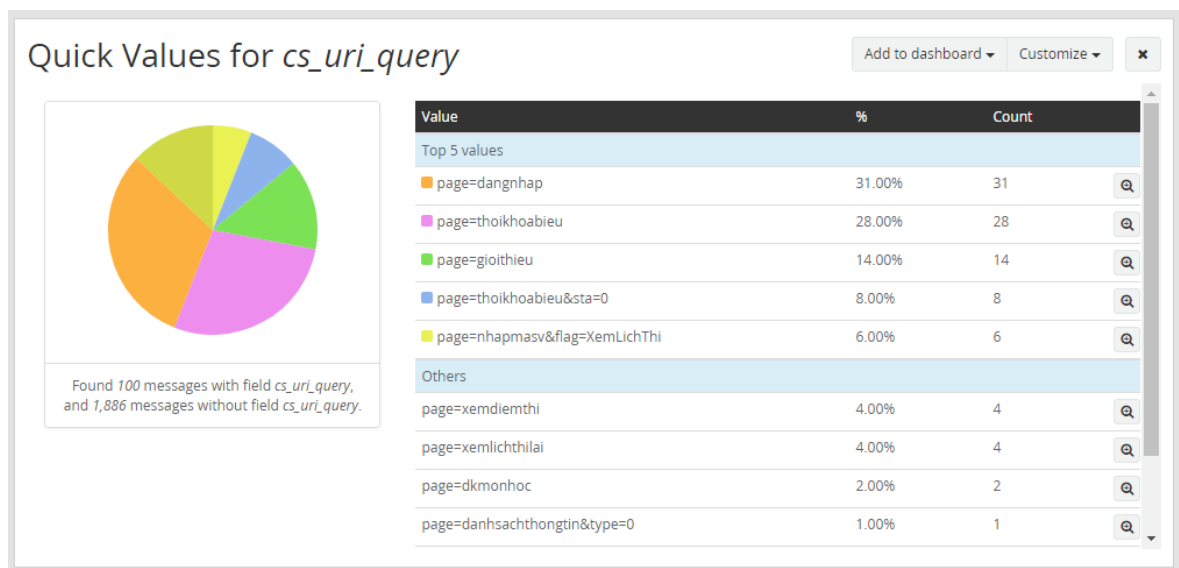


Hình 3.19: Giao diện tìm kiếm log của Graylog

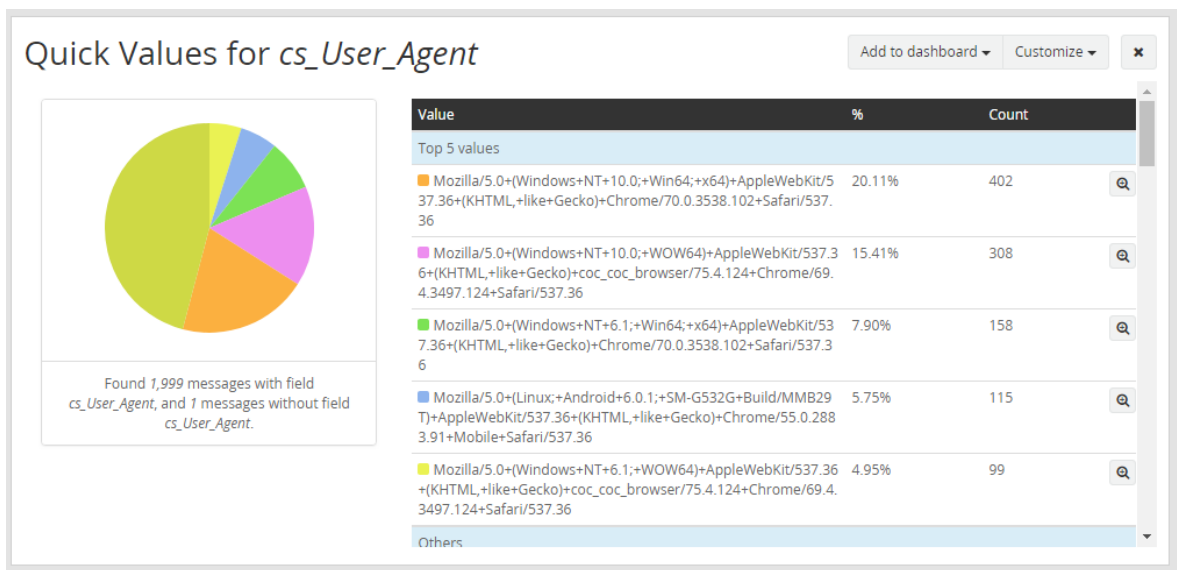
- Xem các báo cáo về thông tin truy cập website:



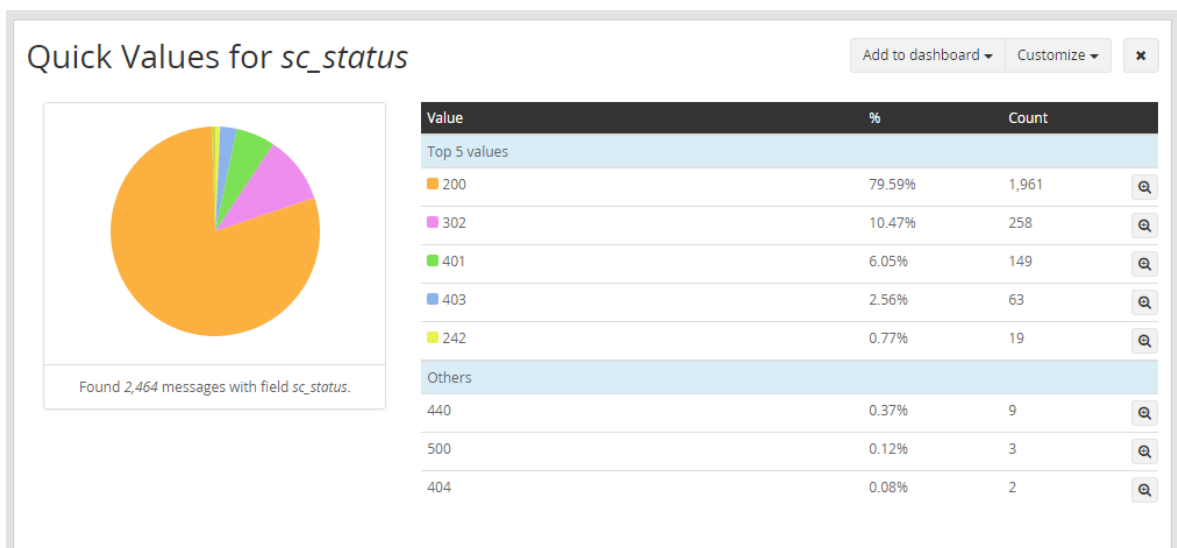
Hình 3.20: Các địa chỉ được truy cập nhiều nhất



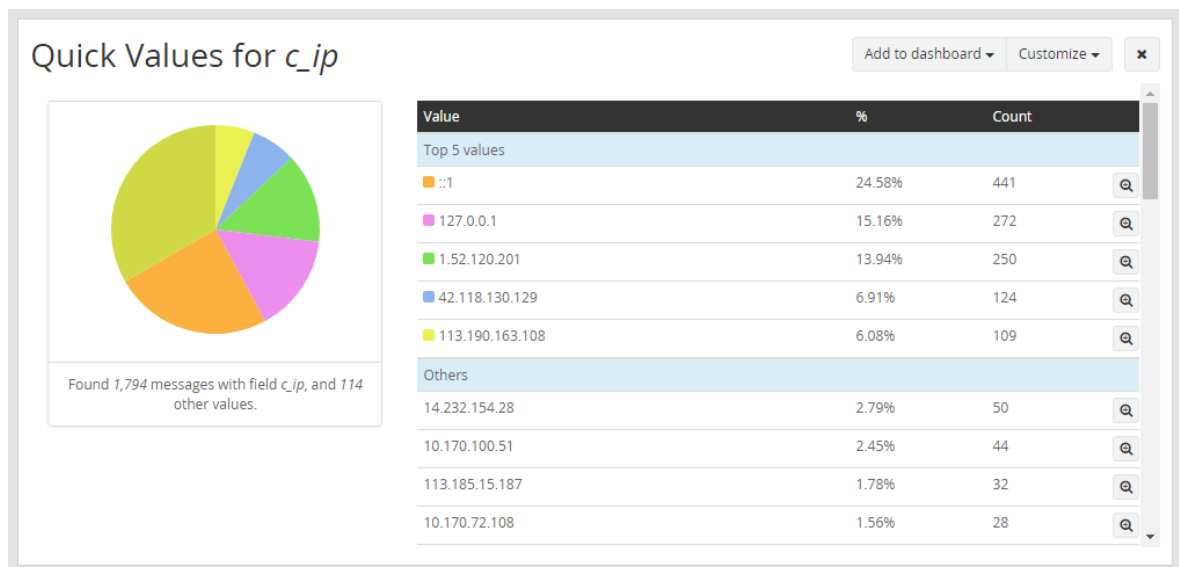
Hình 3.21: Các page được truy cập nhiều nhất



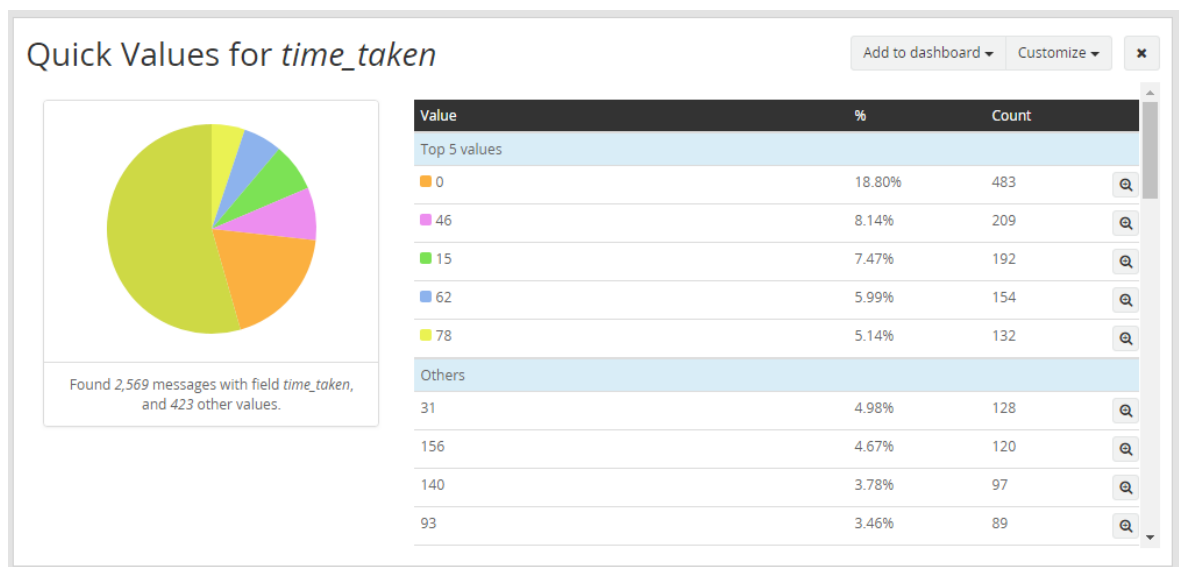
Hình 3.22: Các user-agent truy cập vào website



Hình 3.23: Báo cáo các trạng thái HTTP khi truy cập website

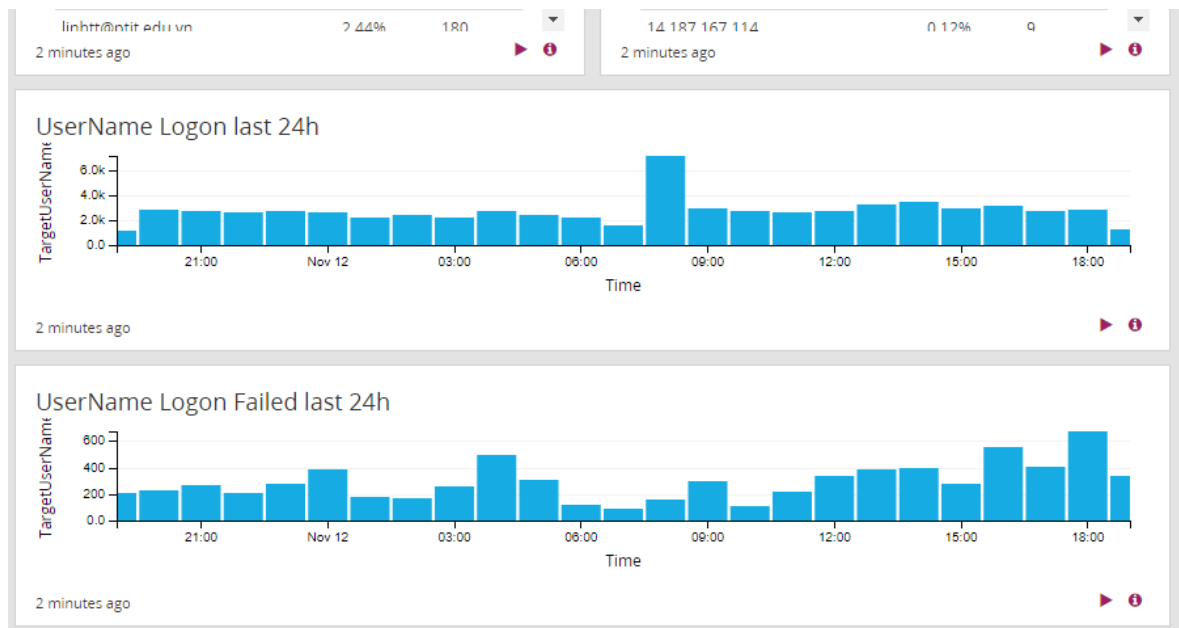


Hình 3.24: Báo cáo các địa chỉ IP truy cập website



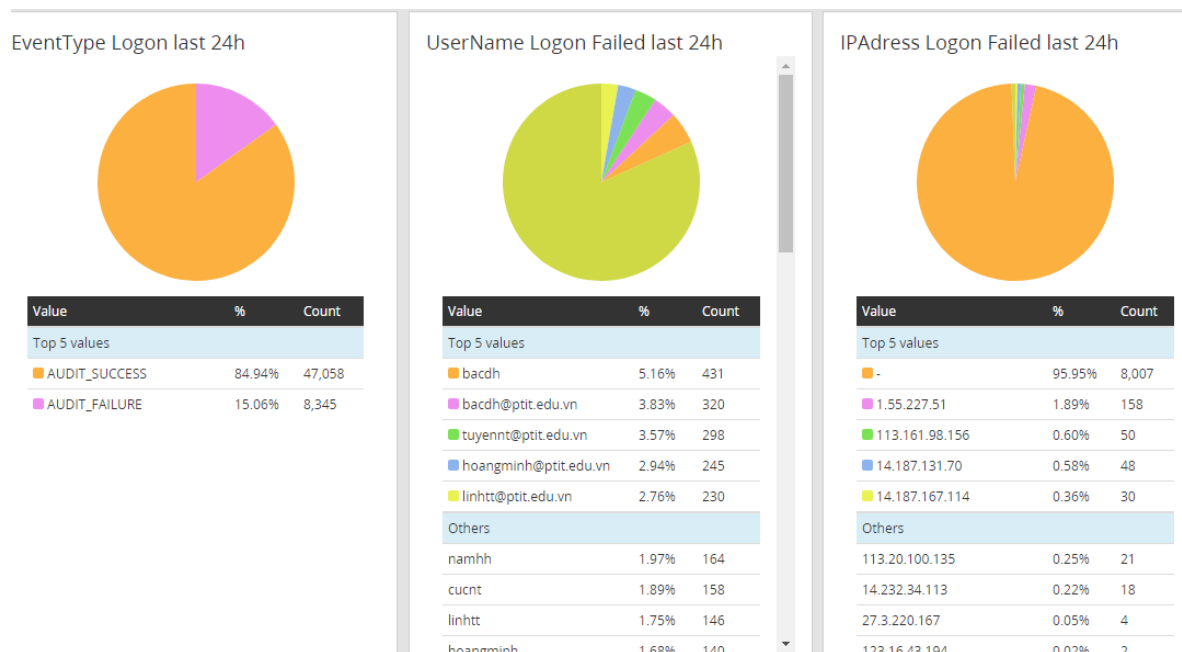
Hình 3.25: Báo cáo thời gian phản hồi khi truy cập website

- Quản lý mail server, xem thời gian hoạt động của người dùng, từ đó có thể biết được thời điểm nào trong ngày có nhiều người truy cập nhất, thời điểm nào bị tấn công nhiều nhất.



Hình 3.26: Báo cáo thời gian đăng nhập của người dùng

- Tạo các báo cáo về tình trạng đăng nhập của người dùng, từ đó biết được user nào đang bị tấn công, xem các địa chỉ IP tấn công, từ đó có thể chặn chúng trên tường lửa.



Hình 3.27: Báo cáo tình trạng đăng nhập của người dùng

- Nhận cảnh báo khi có sự kiện bất thường dựa vào các quy tắc được thiết lập trước, ví dụ như khi có user đăng nhập thất bại quá 5 lần trong vòng 1 phút hoặc khi một service nào đó bị tắt.

Alerts

Check your alerts status from here. Currently displaying all alerts.

Logon Failed on stream Logon Failed Unresolved

Triggered at 2018-11-13 19:41:00, **still ongoing**.

Reason: Stream had 99 messages in the last 1 minutes with trigger condition more than 5 messages. (Current grace time: 1 minutes)
Type: Message Count Alert Condition

Logon Failed on stream Logon Failed Resolved

Triggered at 2018-11-13 19:36:00, resolved at 2018-11-13 19:39:21.

Reason: Stream had 75 messages in the last 1 minutes with trigger condition more than 2 messages. (Current grace time: 1 minutes)
Type: Message Count Alert Condition

Hình 3.28: Nhận cảnh báo khi có đăng nhập bất thường

3.4. Kết luận chương

Chương 3 đã trình bày khái quát về kiến trúc, các thành phần và tính năng của Graylog. Chương cũng mô tả quá trình cài đặt và thử nghiệm thu thập, sau đó xử lý dữ liệu log, từ đó xuất ra các báo cáo về tình trạng truy cập website, các user đang bị tấn công, các địa chỉ IP tấn công, cũng như các cảnh báo khi có bất thường.

KẾT LUẬN VÀ KIẾN NGHỊ

Luận văn này tập trung nghiên cứu về log truy nhập, các dạng log truy nhập, các kỹ thuật xử lý và phân tích log. Cụ thể luận văn đã đạt được các kết quả sau:

- Nghiên cứu các kỹ thuật xử lý và phân tích log để biết được tình trạng hoạt động của các máy chủ dịch vụ, nắm bắt hành vi người dùng, nhận biết khả năng mất an toàn thông tin hệ thống, giúp nâng cao hiệu quả trong công tác vận hành, quản trị hệ thống dịch vụ.
- Giúp hiểu rõ quá trình xử lý log, kỹ thuật phân tích log, các công cụ hỗ trợ xử lý, phân tích log, từ đó có thể lập phương án triển khai các hệ thống xử lý và phân tích log hoạt động hiệu quả.
- Đưa ra mô hình thử nghiệm với đầy đủ các bước thu thập, chuẩn hóa, xử lý và phân tích log, có thể triển khai sử dụng trong thực tế.

Luận văn có thể phát triển tiếp theo hướng như sau:

Tiếp tục thử nghiệm với nhiều loại log khác. Xây dựng hệ thống cảnh báo mất an toàn thông tin với các bước xử lý được thực hiện một cách tự động như: tự động gửi tin nhắn, email cho người quản trị khi có hiện tượng bất thường; tự động chuyển các địa chỉ IP bất thường sang hệ thống tường lửa và chặn nó... Nghiên cứu ứng dụng việc xử lý và phân tích log vào nhiều lĩnh vực khác nhau.

DANH MỤC TÀI LIỆU THAM KHẢO

- [1] Phạm Duy Lộc, Hoàng Xuân Dâu (2018), Khảo sát các nền tảng và kỹ thuật xử lý log truy cập dịch vụ mạng cho phát hiện nguy cơ mất an toàn thông tin, tập 8, Số 2 (2018), Chuyên san Khoa học Tự nhiên và Công nghệ, Tạp chí Khoa học công nghệ Đại học Đà Lạt.
- [2] Roger Meyer (2008), Detecting Attacks on Web Applications from Log Files, SANS Institute.
- [3] Shaimaa Ezzat Salama, Mohamed I. Marie, Laila M. El-Fangary, Yehia K. Helmy (2011), Web Server Logs Preprocessing for Web Intrusion Detection, journal of Computer and Information Science Vol. 4, No. 4, July 2011, Canadian Center of Science and Education.
- [4] Faradzhullaev, R. (2008). Analysis of Web server log files and attack detection. Journal of Automatic Control and Computer Sciences, 42(1), 50-54.
- [5] IBM QRadar SIEM (2017), <https://www.ibm.com/ms-en/marketplace/ibm-qradar-siem>, truy cập tháng 1.2017.
- [6] Extended Log File Format, <https://www.w3.org/TR/WD-logfile.html>, truy cập tháng 11.2018.
- [7] IIS Log File Formats, [https://msdn.microsoft.com/enus/library/ms525807\(v=vs.90\).aspx](https://msdn.microsoft.com/enus/library/ms525807(v=vs.90).aspx), truy cập tháng 11.2018.
- [8] VNCS (2018) - Giải pháp giám sát website tập trung, <http://vncs.vn/portfolio/giai-phap-giam-sat-websites-tap-trung>, truy cập tháng 11.2018.
- [9] Webalizer (2018), <http://www.webalizer.org>, truy cập tháng 11.2018.
- [10] OSSEC (2018), <https://github.com/ossec>, truy cập tháng 11.2018.
- [11] Graylog (2018), <https://www.graylog.org>, truy cập tháng 11.2018.
- [12] Logstash (2018), <http://logstash.net>, truy cập tháng 11.2018.
- [13] Rsyslog (2018), <https://www.rsyslog.com>, truy cập tháng 11.2018.
- [14] NXLog (2018), <https://nxlog.co>, truy cập tháng 11.2018.
- [15] Elasticsearch (2018), <https://www.elastic.co>, truy cập tháng 11.2018.
- [16] MongoDB (2018), <https://www.mongodb.com>, truy cập tháng 11.2018.