Kỷ yếu Hội nghị Quốc gia lần thứ X về Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR), Đà Nẵng, ngày 17-18/08/2017
DOI: 10.15625/vap.2017.00028

GIẢI PHÁP LỌC DỮ LIỆU NHẬT KÝ PROXY DỰA TRÊN CÔNG NGHỆ DỮ LIỆU LỚN

Châu Lê Sa Lin[1], Ngô Bá Hùng[2], Đinh Thế An Huy[3]

[1,3] Khoa Công nghệ thông tin & Truyền thông, Trường Cao đẳng Kinh tế - Kỹ thuật Cần Thơ, Cần Thơ, Việt Nam

[2] Khoa Công nghệ thông tin & Truyền thông, Trường Đại học Cần Thơ, Cần Thơ, Việt Nam

clsalin@ctec.edu.vn, nbhung@cit.ctu.edu.vn, dtahuy@ctec.edu.vn

TÓM TẮT: *Dữ liệu nhật ký của proxy có ý nghĩa quan trọng trong vấn đề an ninh mạng và giám định mạng. Bài báo này sẽ đề xuất một giải pháp cho phép lọc bỏ bớt các dữ liệu dư thừa từ dữ liệu nhật ký sinh ra của các proxy mà không làm mất đi các thông tin phục vụ cho những yêu cầu an ninh mạng và giám định mạng về sau. Giải pháp được xây dựng dựa trên các công nghệ xử lý luồng dữ liệu của lĩnh vực dữ liệu lớn. Kết quả cho thấy, giải pháp lọc đề xuất cho phép giảm dung lượng cần phải lưu trữ cho dữ liệu nhật ký đạt hơn 40%.*

Từ khóa: *Nhật ký proxy; Lọc dữ liệu; An ninh mạng; Giám định mạng; Dữ liệu lớn.*

I. GIỚI THIỆU

Proxy là một dịch vụ ủy nhiệm làm nhiệm vụ chuyển tiếp thông tin và kiểm soát truy cập tạo sự an toàn cho việc truy cập Internet của các client. Proxy đóng vai trò quan trọng trong việc hỗ trợ lưu thông trên mạng như: kiểm soát truy cập, giám sát lưu thông, lưu trữ tạm (*caching)* nhằm tiết kiệm băng thông. Proxy cho phép client truy cập mạng thông qua một máy tính khác với một tài khoản duy nhất, máy tính này gọi là *Proxy server*. Ngoài việc người quản trị có thể quản lý, theo dõi hoạt động người dùng trực tiếp trên hệ thống *Proxy server*, người quản trị còn có thể theo dõi người dùng dựa trên các dữ liệu nhật ký sinh ra proxy. *Dữ liệu nhật ký proxy* (proxy log) là tất cả những dữ liệu được proxy server ghi lại một cách chi tiết như: những hoạt động của người dùng, tình trạng hoạt động của hệ thống, của các ứng dụng, và của các thiết bị đã và đang hoạt động trong hệ thống. Việc ghi dữ liệu nhật ký proxy giúp cho người quản trị theo dõi tốt các hoạt động của người dùng, nhằm cải thiện khả năng quản lý hệ thống, quản lý người dùng, quản lý các vấn đề về cân bằng tải cũng như để phát hiện ra các cuộc tấn công bên ngoài vào hệ thống như tấn công DDoS...

Dữ liệu nhật ký proxy đóng vai trò hết sức quan trọng trong quá trình kiểm soát *an ninh mạng* (network security) và *điều tra mạng* (network forensics). Kiểm soát *an ninh mạng* (network security) nhằm phát hiện, ứng phó ngăn chặn các cuộc tấn công mạng từ bên trong cũng như bên ngoài hệ thống [3]. *Điều tra mạng* là một nhánh của khoa học điều tra số liên quan đến việc giám sát và phân tích lưu lượng mạng máy tính nhằm phục vụ cho việc thu thập thông tin, chứng cứ pháp lý hay phát hiện các xâm nhập [3]. Việc ghi lại nhật ký proxy được thực hiện một cách liên tục. Nếu lưu trữ trong thời gian dài sẽ làm cho dữ liệu nhật ký proxy phình to vượt quá khả năng lưu trữ của các tổ chức. Các nhà quản trị mạng có xu hướng chỉ lưu giữ dữ liệu nhật ký proxy trong một khoảng thời gian gần nhất, ví dụ như một tháng. Điều này gây ảnh hưởng lớn đến công tác thu thập, phân tích sự cố an ninh mạng, điều tra mạng nếu tác nhân phá hoại đã thực hiện hoạt động của mình trong khoảng thời gian dài trước đó. Phân tích ban đầu cho thấy rằng, ngoài các dữ liệu hữu ích phục vụ cho công tác an ninh mạng và điều tra mạng, dữ liệu nhật ký proxy còn chứa cả một lượng lớn những dữ liệu không có ý nghĩa cho mục tiêu an ninh mạng và điều tra mạng. Chính vì thế, bài báo này đề xuất một giải pháp cho phép lọc bỏ bớt các dữ liệu dư thừa từ dữ liệu nhật ký sinh ra của các proxy để giảm dung lượng cần thiết để lưu trữ dữ liệu nhật ký proxy mà vẫn đảm bảo không làm mất đi các thông tin hữu ích cho những yêu cầu an ninh mạng và giám định mạng về sau. Giải pháp được xây dựng dựa trên các công nghệ xử lý luồng dữ liệu của lĩnh vực dữ liệu lớn. Kết quả thử nghiệm cho thấy, giải pháp lọc đề xuất có thể giảm dung lượng cần phải lưu trữ cho dữ liệu nhật ký proxy lên đến hơn 40%. Phần kế tiếp sẽ trình bày các nghiên cứu có liên quan đến vấn đề bài báo đặt ra. Mô hình lọc dữ liệu nhật ký proxy sẽ được đề xuất trong phần thứ III; Chi tiết cài đặt mô hình thử nghiệm dựa trên các công nghệ của lĩnh vực tính toán dữ liệu lớn sẽ được trình bày trong phần thứ IV. Phần thứ V là kết quả thực nghiệm và đánh giá hiệu suất lọc của mô hình đề xuất. Cuối cùng là phần kết luận và các công việc cho tương lai để cải tiến và phát triển tiếp mô hình đề xuất.

II. CÁC NGHIÊN CỨU LIÊN QUAN

Có nhiều phần mềm có thể làm proxy server như ISA trên Windows, Squid [6] trên Unix/Linux. Trong nghiên cứu này, bài báo tập trung vào nguồn proxy log do Squid sinh ra nên. Đối với Squid, dữ liệu nhật ký được lưu trong các tập tin nhật ký (log file), với mỗi dòng văn bản (text) là một mẫu tin ghi nhận về một sự kiện phát sinh có liên quan đến Squid. Mẫu tin nhật ký của Squid gồm nhiều trường ngăn cách với nhau bởi một khoảng trắng, ví dụ "*1438112845.178 1052 192.168.9.131 TCP_MISS/200 11141 GET http://filehippo.com/ - HIER_DIRECT/108.168.208.206 text/html*". Trường đầu tiên là nhãn thời gian ghi nhận thời điểm của sự kiện phát sinh, các trường kế tiếp là thông tin về sự kiện phát sinh. Số lượng mẫu tin sinh ra bởi trong dữ liệu nhật ký proxy thường rất lớn, ví dụ ở Đại học Cần Thơ, Theo khảo sát thực tế tìm hiểu được thì hàng ngày dung lượng proxy log được hệ thống lưu trữ lại trên 4GB. Còn vào những ngày cao điểm như các đợt thi online, đăng ký môn học, công bố điểm thi,… thì dung lượng log này lớn hơn nhiều lần.

Trong số đó, có mẫu tin hoặc có trường có ý nghĩa trong việc giám sát an ninh mạng, nhưng cũng có những mẫu tin và trường không liên quan gì. Nghiên cứu [5] đã trình bày một số cách làm sạch dữ liệu web ở các máy trạm (client) bằng cách loại bỏ những website hoặc các khung (frame) được gọi thực thi tự động khi client truy cập vào một website nào đó, chẳng hạn các trang quảng cáo hoặc các đoạn mã được viết bằng ngôn ngữ javascript sẽ tự động được kích hoạt. Nghiên cứu [19] trình bày cách làm giảm dung lượng các tập tin lưu trữ dữ liệu nhật ký proxy ở giai đoạn tiền xử lý dữ liệu bằng cách làm sạch dữ liệu không cần thiết trong các log file và tìm các IP đích trong quá trình các Client truy cập, từ đó loại bỏ những dữ liệu được sinh ra giống nhau từ cùng một IP. Nghiên cứu [20] trình bày tổng quát cấu trúc của các tập tin dữ liệu nhật ký proxy cũng như các dữ liệu không cần thiết trong các tập tin này, chẳng hạn như các định dạng tập tin *.jpeg, *.gif, *.dll,.... từ đó định hướng để làm sạch dữ liệu nhật ký proxy để giảm dung lượng lưu trữ. Ngoài ra một số công cụ quản lý dữ liệu lớn có thể áp dụng để xử lý dữ liệu nhật ký proxy. Sarg [6] cho phép thu thập và xử lý dữ liệu nhật ký proxy. Với Sarg, người quản trị có thể dễ dàng nắm bắt được các thông tin đến Squid server. Những báo cáo được Sarg ghi nhận lại thông qua các thông tin log Squid ghi nhận tại file access.log. Proxy log Explorer [30] là phần mềm phân tích và xử lý proxy log dữ liệu nhật ký proxy nhanh và mạnh nhất để theo dõi giám sát hệ thống server proxy server cho công ty, trog các trung tâm dữ liệu. Phần mềm này có thể tự động nhận dạng các file log tập tin dữ liệu nhật ký proxy. Nghiên cứu [9] phân tích các ảnh hưởng của log file, phân tích log file. Tuy nhiên tác giả [9] chưa đề xuất cách tích hợp các thành một hệ thống hoàn chỉnh, chỉ sử dụng các công cụ rời rạc. Trong nghiên cứu [19] tác giả tập trung vào việc làm sạch dữ liệu nhật ký và giảm dung lượng các tập tin dữ liệu nhật ký proxy ở phía client. Các vấn đề hạn chế của nghiên cứu trên là nếu làm sạch dữ liệu từ phía client sẽ khó khăn trong khâu cài đặt vì phải cài trực tiếp trên client, tốn thời gian đôi khi phần mềm làm ra không tương thích với các máy client. Trong nghiên cứu [20] tác giả tập trung vào việc làm sạch dữ liệu trong giai đoạn tiền xử lý tại server. Và việc thực hiện chức năng này hoàn toàn offline, làm sạch từ file log kết hợp với việc giảm dung lượng cho file log và tập trung chủ yếu vào việc loại bỏ những log file được sinh ra giống nhau từ IP trùng lặp. Ngoài ra, những phần mềm ứng dụng liên quan việc xử lý dữ liệu nhật ký *proxy* thì tốn chi phí để cài đặt và nó không thực hiện được chức năng lọc bỏ dữ liệu tiền xử lý mà các phần mềm ứng dụng đó thiên về dạng thống kê và báo cáo hơn là xử lý dữ liệu log.

III. MÔ HÌNH LỌC DỮ LIỆU NHẬT KÝ PROXY

Mục tiêu của nghiên cứu này là đưa ra một giải pháp cho phép lọc bớt các dữ liệu nhật ký sinh ra từ proxy server để giảm dung lượng cần phải cung cấp cho việc lưu dữ liệu nhật ký proxy trong một khoảng thời gian dài phục vụ công tác giám an ninh mạng và giám định mạng. Giải pháp lọc phải đảm bảo không làm mất đi các thông tin cần thiết để phục vụ cho mục tiêu an ninh mạng về sau. Gọi L là dữ liệu nhật ký ban đầu sinh ra bởi một proxy server (Squid). L bao gồm nhiều dòng log mỗi dòng là một *mẫu tin nhật ký* đại diện cho một sự kiện cụ thể, hay một hoạt động mà proxy server ghi lại dựa vào sự tác động của người dùng. Trong một sự kiện của mẫu tin nhật ký thì chứa 11 thuộc tính bao gồm: Timestamp, Elapsed, IP_client, Code/Status, Size, Method, URL, rfc931, Peerstatus, Destination IP và Type. Mỗi thuộc tính thể hiện một giá trị nội dung tương ứng cho sự kiện đó. [6] Gọi C là tập hợp các mối quan tâm của nhà quản trị mạng về vấn đề an ninh mà họ có thể tìm ra từ việc phân tích dữ liệu proxy L. Mỗi C_i tương ứng với một quan tâm của người quản trị: i=1→N.

Gọi T là một hàm phân tích dữ liệu L để tìm ra C. Ta có T(L) → C. Khi đó ta cần tìm một hàm lọc dữ liệu F nhận đầu vào là dữ liệu nhật ký L để sinh ra một tập dữ liệu nhật ký mới L' sao cho kích thước của L' thì nhỏ hơn kích thước của L nhưng vẫn đảm bảo tồn tại một hàm T' có thể phân tích tập dữ liệu L' để tìm ra được lại C, tức ta có: T'(L') →

C. Như vậy lúc này hệ thống sẽ lưu trữ dữ liệu nhất ký proxy L' có kích thước nhỏ hơn so với dữ liệu nhật ký proxy ban đầu L, sao cho dựa vào L' các nhà quản trị mạng vẫn có thể tìm ra được các mối quan tâm của họ liên quan đến vấn đề an ninh mạng. Qua khảo sát các nghiên cứu có liên quan [5][19][20] và các công cụ lọc dữ liệu[6][30] kết hợp với việc phỏng vấn một số nhà quản trị mạng của Trường Đại học Cần Thơ, mối quan tâm C của các nhà quản trị mạng trên dữ liệu nhật ký của proxy có thể được chia thành 3 nhóm chính đó là Thống kê, tìm kiếm và theo dõi giám sát.

- ***Các mối quan tâm C phổ biến nhất về mặt thống kê:***
  - C_1: Thống kê số lượng địa chỉ IP_Client truy cập vào hệ thống nhiều nhất trong một giờ, một ngày/ một tháng/một năm.
  - C_2: Thống kê số lượng địa chỉ IP_Client truy cập vào địa chỉ URL nhiều nhất trong 1 ngày.
  - C_3: Thống kê số lượng địa chỉ IP_ Destination được gửi yêu cầu truy cập nhiều nhất.
  - C_4: Thống kê số lượng URL nào được chấp nhận truy cập nhiều nhất.
  - C_5: Thống kê số lượng URL nào bị từ chối truy cập.
  - C_6: Thống kê danh sách URL được Proxy server trả yêu cầu cho người dùng được lấy từ cache server.
  - C_7: Tổng số yêu cầu của IP_Client đến server với dung lượng trong ngày là bao nhiêu.
  - C_8: Thống kê danh sách địa chỉ URL được truy cập nhiều nhất.
  - C_9: Thống kê top 10 IP_client đã truy cập vào hệ thống nhiều nhất.

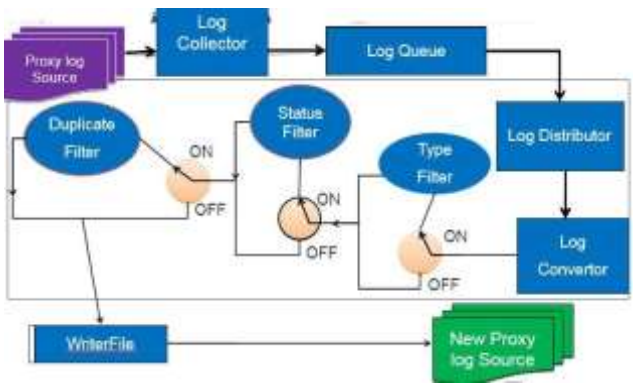
- C_{10}: Tìm top 10 IP_ Destination được người dùng gửi yêu cầu đến nhiều nhất.
- C_{11}: Tìm top 10 URL được người dùng truy cập nhiều nhất, tổng dung lượng yêu cầu.
- C_{12}: Thống kê các kiểu file được người dùng yêu cầu truy cập trong một ngày/ một tháng/ một năm.

- ***Các quan tâm C phổ biến nhất về vấn đề tìm kiếm:***
  - C_{13}: Tìm thông tin chi tiết về địa chỉ IP_Client truy cập vào địa chỉ URL nhiều nhất theo ngày/ tháng/năm.
  - C_{14}: Tìm thông tin chi tiết về các kiểu file mà người dùng gửi yêu cầu truy cập nhiều nhất.
  - C_{15}: Tìm chi tiết các địa chỉ IP_Client bị từ chối phục vụ khi truy cập vào URL cấm theo giờ/ ngày/ tháng /năm.?
  - C_{16}: Tìm danh sách địa chỉ URL được lưu trữ sẵn trên Cache của server proxy.
  - C_{17}: Tìm danh sách các URL được yêu cầu và tổng dung lượng yêu cầu của từng URL theo ngày.
  - C_{18}: Tìm dung lượng yêu cầu tải về từ Ip_client nhiều nhất trong một ngày là bao nhiêu.

- ***Mối quan tâm về vấn đề theo dõi giám sát hệ thống:*** Nhà quản trị có thể theo dõi hệ thống, quản lý các vấn đề về cân bằng tải cũng như để phát hiện ra các cuộc tấn công DDoS.

Từ các mối quan tâm phổ biến của tập C trên, tiến thành phân tích cấu trúc và ngữ nghĩa của dữ liệu L sẽ cho biết được cần loại bỏ những dữ liệu nào để có được L' mà vẫn đảm bảo tìm ra được C từ L', tức là xác định được hàm lọc $F(L) \rightarrow L'$.

Ý tưởng về hàm lọc dữ liệu F được mô hình hóa như hình 1. Trong hình 1, dữ liệu log ban đầu L được đại diện bằng thành phần Proxy log source, đó là các tập dữ liệu nhật ký được thu thập trực tuyến về nơi đặt hàm F nhờ *Bộ phận thu nhận proxy log* (Log Collector). Để tránh tình trạng dữ liệu đỗ về ào ạt và xảy ra hiện tượng nghẹt cổ chai tại F, mô hình đề xuất xây dựng thêm một thành phần *Hàng đợi* (Log Queue) để tách dữ liệu nhật ký thành nhiều luồng, mỗi luồng theo một chủ đề khác nhau. Việc chia theo chủ đề cho dữ liệu nhật ký do người dùng định nghĩa để dễ quản lý log vì có nhiêu loại log không riêng gì proxy log. Dữ liệu nhật ký sẽ được chuyển đổi kiểu phù hợp trước khi đưa vào bộ lọc.



Hình 1. Mô hình tổng quát giải pháp lọc dữ liệu nhật ký prxoy

Mô hình đề xuất 3 bộ lọc chính, mỗi bộ lọc có 2 trạng thái ON/OFF. Tùy theo người quản trị muốn sử dụng chức năng bộ lọc nào thì đặt trạng thái bộ lọc đó lên ON. Các bộ lọc chính có ý nghĩa như sau:

- ✓ ***Bộ nhận dữ liệu và phân phối*** (Log Distributor): Nhận dữ liệu từ hàng đợi và phân phát dữ liệu nhận được
- ✓ ***Bộ phận thực hiện phân tích và chuyển đổi log*** (Log Convertor): nhận trực tiếp dữ liệu từ bộ phận nhận dữ liệu đầu vào dưới dạng các json, thực hiện chuyển các json về dạng LogItem.
- ✓ ***Bộ lọc dựa trên kiểu*** (Type Filter): là bộ lọc loại bỏ các mẫu tin nhật ký dựa vào kiểu dữ liệu của tài nguyên (do người dùng mô tả).
- ✓ ***Bộ lọc dựa trên kết quả truy cập*** (Status Filter): Loại bỏ các mẫu tin nhật ký dựa vào thuộc tính là kết quả trả của một yêu cầu truy cập.
- ✓ ***Bộ lọc trùng*** (Duplicate Filter): là bộ lọc xử lý loại bỏ các mẫu tin nhật ký trùng nhau. Đây là bộ lọc trực tuyến, dòng các mẫu tin nhật ký của proxy đổ vào bộ lọc được chia thành những khoảng/đoạn liên tiếp nhau trong một khoảng thời gian ví dụ 30 giây và việc tính toán dữ liệu trùng sẽ được thực hiện trong phạm vi của từng khoảng dữ liệu.

Sau khi đi ra khỏi các bộ lọc thành phần File Writer sẽ xuất lại dữ liệu nhật ký proxy vào một kho lưu trữ New proxy log Source. Đây chính là dữ liệu nhật ký proxy mới L' có được sau khi đã lọc bỏ các dữ liệu thừa từ L. Kích thước của L' được trông đợi sẽ nhỏ hơn so với kích thước của L ban đầu.

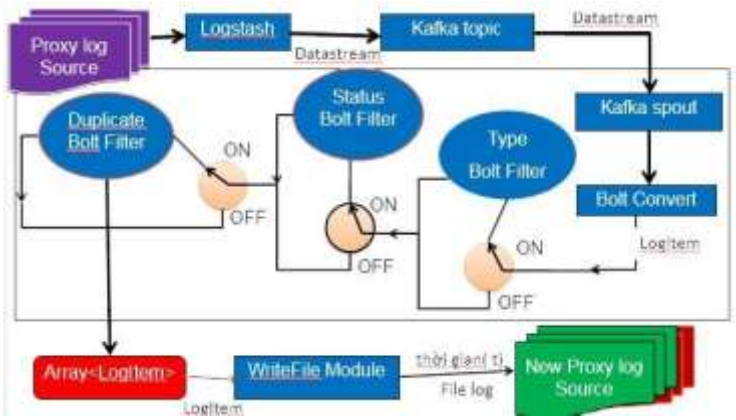
IV. CÀI ĐẶT MÔ HÌNH LỌC DỮ LIỆU NHẬT KÝ PROXY

Để có thể lọc được một khối lượng lớn dữ liệu nhật ký proxy sinh ra trong một đơn vị thời gian, bài báo đề nghị cài đặt mô hình lọc dữ liệu nhật ký proxy đã được đề xuất ở phần trước dựa trên công nghệ xử lý dữ liệu lớn (big data). Apache Storm[14] đã được chọn lựa cho việc cài đặt mô hình lọc dữ liệu nhật ký proxy. Apache Storm là một dự án mã nguồn mở được thiết kế theo kiến trúc phân tán được áp dụng trong việc xử lý dữ liệu luồng. Một hệ thống Storm bao gồm hai thành phần chính là: *Nút chủ* (master node) còn gọi là Nimbus và các *Máy thợ* (worker node) còn gọi là Supervisor. Tập hợp các dữ liệu trong storm gọi là *Luồng* (Stream). Ngoài ra, thành phần đảm nhiệm để xử lý dữ liệu liên tục gọi là Topology. Trong Topology có 2 thành phần chính là *Đầu vào* (Spout) và *Nút xử lý* (Bolt). Spout có nhiệm vụ nhận dữ liệu đầu vào. Bolt có nhiệm vụ xử lý dữ liệu đầu vào. Bolt làm việc hoàn toàn dựa trên kiến trúc đã quy hoạch ban đầu. Dữ liệu đầu ra của một Bolt có thể đưa vào một Bolt khác để tiếp tục xử lý. Bolt có thể thực hiện nhiều chức năng như: gọi các hàm xử lý, thu thập các trường dữ liệu trong một tập hợp, phân tích các luồng dữ liệu, thực hiện xử lý giao tác luồng (streaming-join), tương tác với cơ sở dữ liệu và nhiều tính năng khác...

Hình 2 là sự cài đặt của mô hình lọc dữ liệu nhật ký proxy đã trình bày ở hình 1. Trong đó:

- Kafka [31] spout: là thành phần nhận dữ liệu đầu vào từ hệ thống.
- Bolt convert: là thành phần chuyển kiểu dữ liệu proxy log thành kiểu dữ liệu đối tượng để phục vụ cho việc lọc dữ liệu.
- Type Bolt filter: Chức năng lọc mẫu tin nhật ký dựa vào kiểu tài nguyên.
- Status Botl filter: Chức năng lọc dữ liệu dựa vào thuộc tính kết quả trả về.
- Duplicate Bolt Filter: Chức năng lọc mẫu tin nhật ký dựa vào các thuộc tính có nội dung trùng nhau.

Ngoài ra, các thành phần còn lại đã được giới thiệu trên mô hình tổng quát.



Hình 2. Mô hình chi tiết giải pháp lọc dữ liệu proxy log dựa trên công nghệ Storm

Ý tưởng của giải pháp này là dữ liệu nhật ký proxy được hệ thống Logstash [8] thu thập sẽ đưa vào từng topic do người dùng định nghĩa, dữ liệu truyền đi theo dạng luồng (stream). Kafka [31] spout có nhiệm vụ nhận các luồng này và đưa vào Bolt Convert để chuyển kiểu dữ liệu dạng json về dạng đối tượng (LogItem). Sau khi dữ liệu được chuyển kiểu tại đây, hệ thống có 3 bộ lọc với 2 trạng thái ON hoặc OFF. Nếu được thiết đặt ở trạng thái ON, dữ liệu đi qua bộ lọc sẽ được xử lý theo chức năng của bộ lọc. Người quản trị có thể sử dụng bất kỳ bộ lọc nào, có thể sử dụng cùng lúc tất cả bộ lọc, hoặc có thể kết hợp bộ lọc lại hoặc dùng mỗi lần 1 bộ lọc. Người quản trị chỉ cần OFF bộ lọc nào đó thì dữ liệu sẽ không đi qua bộ lọc đó nữa.

Bolt convert: Với số thuộc tính ban đầu trong từng mẫu tin nhật ký của dữ liệu nhật lý proxy squid là 11, sau khi qua bộ chuẩn hóa sẽ thêm 1 cột thuộc tính thể hiện số lần xuất hiện trong file log mới.

Giải thuật: Input: Kafka spout nhận các dòng sự kiện log từ kafka topic

Output: LogItemitem, sau khi qua hệ thống Bolt convert thì proxy log được chuyển về dạng LogItem

1. JSon currentLog = KafkaSpout.output();
2. LogItem item = GetLogItem(currentLog);
3. KafkaBolt.setOutput(item) Ví dụ: Một dòng log có định dạng Json như sau:

1438113066.093412 192.168.9.131 TCP_MISS/200 1960 GET

http://cache.filehippo.com/img/ex/3351 youcam6_icon.png - HIER_DIRECT/108.161.189.5 image/png.

Sau khi chuyển định dạng về Logitem có nội dung như sau: *m_dTime = 1438113066.093 ; m_iElapsed = 412;*

m_stClientIP = 192.168.9.131; m_stCode = TCP_MISS; m_stStatus = 200; m_lSize = 1960; m_stMethod = GET; m_stURL = http://cache.filehippo.com/img/ex/3351 youcam6_icon.png- m_stRFC931 = -; m_stPeerStatus = HIER_DIRECT; m_stDesIP = 108.161.189.5; m_stType = image/png; m_iNumExist = 1

Bộ lọc dựa trên kiểu (TypeFilter): Chức năng chính của thành phần này là loại bỏ những mẫu tin trong dữ liệu nhật ký proxy ghi nhận lại việc truy cập đến một tài nguyên thuộc kiểu dữ liệu không thuộc mối quan tâm về an ninh mạng, ví dụ các truy cập vào các tập tin có kiểu CSS. Ta thấy rằng, trong khi người dùng truy cập vào một tài nguyên bất kỳ thì hệ thống luôn ghi nhật ký lại địa chỉ của tài nguyên vào trong dữ liệu nhật ký proxy. Trong trang web mà người dùng truy cập gọi bao nhiêu tài nguyên thì hệ thống điều ghi lại địa chỉ các tài nguyên này. Các tài nguyên là một tập tin CSS không liên quan gì đến hệ thống cơ sở dữ liệu cũng như bảo mật của website.

Giải thuật: Input: LogItem item

Output: LogItemitem, sự kiện log theo đã bỏ thuộc tính kiểu dữ liệu css

Processing step:

1. If(item. m_stType =="css") then exit

   Else

2. KafkaBolt.setOutput(item)

Bộ lọc dựa trên kết quả truy cập (Status Filter): Chức năng chính của bộ lọc này là loại bỏ dòng log dựa trên kết quả trả về của một yêu cầu thực thi. Có rất nhiều kết quả như: 400,404, 500,502, 100…, ở đây bài báo ví dụ các sự kiện log có chứa kết quả thực thi là 400 hay 404. Mã code trả về 400 tương ứng với những trường hợp client gửi yêu cầu đến server cú pháp không hợp lệ, cookie của client bị lỗi, trình duyệt web bị lỗi, sử dụng url không chính xác hoặc đưa ra các yêu cầu không đúng cú phát của giao thức HTTP. Mã code 404 có ý nghĩa là không tìm thấy, lỗi, có nghĩa là người dùng có thể giao tiếp với máy chủ nhưng nó không thể xác định vị trí các tập tin hoặc yêu cầu tài nguyên. Có thể là tài nguyên mà người dùng gửi yêu cầu đã bị xóa nên kết quả code/status báo lỗi không tìm thấy yêu cầu đó.

Giải thuật : Input: LogItemitem

Output: LogItemitem, sự kiện log đã loại bỏ kết quả code/status 400 và code/status 404

Processing steps:

1. If(item. m_stStatus =="400" || item. m_stStatus =="404") then exit

else

2. KafkaBolt.setOutput(item)

Bộ lọc trùng (DuplicateFilter): Bộ lọc này thực hiện chức năng lọc dữ liệu dựa vào các thuộc tính có nội dung trùng nhau trong một khoảng thời gian quy định (ví dụ: 10s, 20s, 30s…). Nếu thời gian càng cao thì khả năng trùng càng nhiều và dung lượng sẽ giảm nhiều hơn. Sau khi kiểm tra các thuộc tính giống nhau bộ lọc sẽ giữ lại một dòng sự kiện đại diện cho các dòng sự kiện bị trùng và thêm cột đếm số lượng dòng sự kiện bị trùng lặp vào phía sau dòng sự kiện đại diện.

Giải thuật: Input: LogItem item, tất cả sự kiện proxy log nhận từ kafka

Output: LogItemitem, sự kiện proxy log đã được xử lý trùng Processing steps:

1. int index = KiemTraTrung(item, A);
2. if (index >0)
   {

A.update(item, index)

}

else

{

A.insert(item)

}

3. Output: ArrayList< LogItem> A

V. THỰC NGHỆM VÀ ĐÁNH GIÁ KẾT QUẢ

a. Dữ liệu dùng để đánh giá

Bài báo sử dụng 2 tập dữ liệu để đánh giá hiệu quả của bộ lọc. Tập dữ liệu thứ 1 (Dataset 1) sử dụng nguồn log được giả định từ một máy ảo và cấu trúc hoàn toàn giống với nguồn log thực tế và dung lượng là 8,040 MB với số dòng nhật ký được sinh ra là 51,344. Tập dữ liệu 2 (Dataset 2) sử dụng nguồn log được cung cấp tại địa chỉ [31] và dung lượng là 12,196 MB với số dòng nhật ký được sinh ra là 95,141 dòng.

Bảng 4.1. Bảng dữ liệu sử dụng trong thực nghiệm

STT	Dữ liệu	Dung lượng (MB)	Số mẫu tin	Ghi chú
1	DataSet1	8,040	51,344	*Dữ liệu giả định từ máy ảo*
2	DataSet2	12,196	95,141	*Dữ liệu tham khảo[31]*

b. Phương pháp thực nghiệm

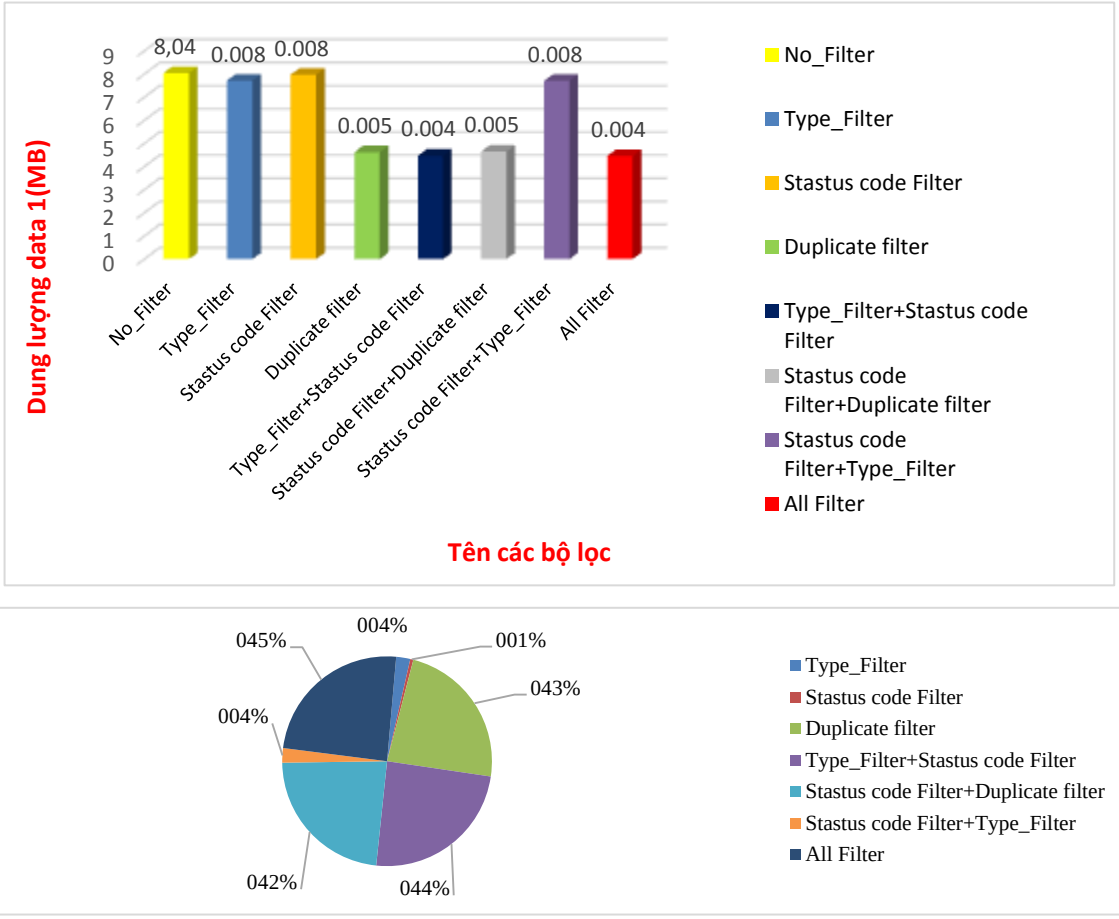
Bài báo tiến hành thực nghiệm dựa trên 3 trường hợp tổng quát sau: sử dụng 1 bộ lọc, sử dụng kết hợp 2 bộ lọc, và sự dụng cả 3 bộ lọc. Mỗi một trường hợp thực nghiệm sẽ cho kết quả khác nhau và trường hợp sử dụng 3 bộ lọc cùng lúc sẽ cho kết quả cao nhất với dung lượng giảm hơn 40%.

c. **Pháp** pháp đánh giá thực nghiệm

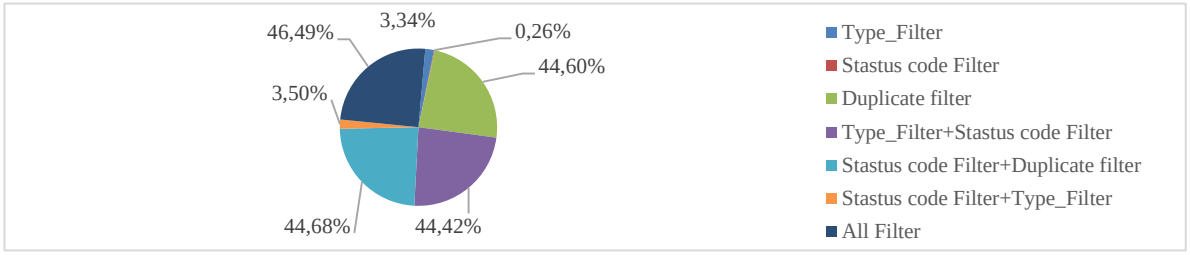
Đánh giá tính hiệu quả của phương pháp lọc dữ liệu nhật ký proxy dựa vào kết quả thu được sau khi dữ liệu đi qua từng bộ lọc có dung lượng giảm hơn dung lượng ban đầu và số dòng sự kiện giảm nhưng vẫn đảm bảo thông tin mà người quản trị quan tâm.

Đánh giá tính đầy đủ thông tin mà người dùng quan tâm, bài báo sẽ thực hiện việc tìm kiếm thông tin mà người quản trị yêu cầu từ nguồn dữ liệu ban đầu (L) và thông tin từ nguồn dữ liệu sau khi sử dụng giải pháp lọc(L'). Cả 2 nguồn dữ liệu đều cho ra kết quả như nhau.

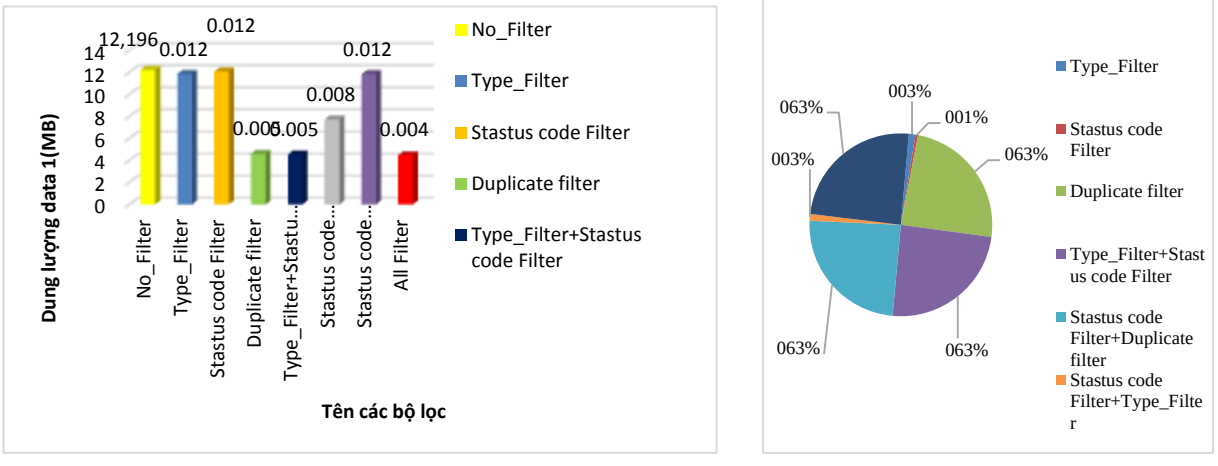
d. Kết quả thực nghiệm



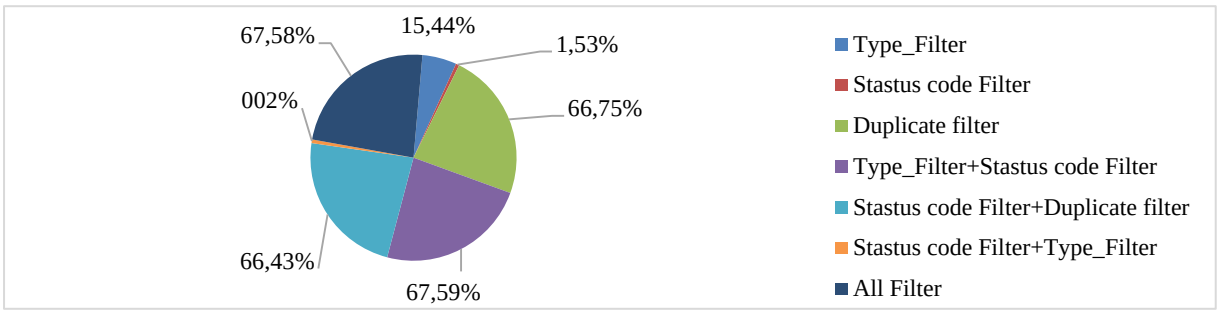
Hình 3. Thể hiện tỷ lệ % dung lượng giảm sau khi qua các trường hợp với tập dữ liệu Dataset 1



Hình 4. Thể hiện tỷ lệ % dòng log đã được loại bỏ của tập Datset 1



Hình 5. Thể hiện tỷ lệ % dung lượng giảm sau khi qua các trường hợp với tập dữ liệu Dataset 2



Hình 6. Thể hiện tỷ lệ % dòng log đã được loại bỏ của tập Datset 2

VI. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Dữ liệu nhật ký sinh ra từ các proxy có ý nghĩa quan trọng trong đảm nảo an ninh mạng và điều tra mạng. Khối lượng dữ liệu nhật ký proxy sinh ra hàng ngày là rất lớn đòi hỏi tốn nhiều dung lượng lưu trữ. Các nhà quản trị thường có xu hướng chỉ lưu lại dữ liệu nhật ký gần đây nhất, ví dụ trong vòng một tháng. Điều này làm mất đi các chứng cứ quan trọng trong việc đảm bảo an ninh mạng và điều tra mạng. Bài báo này đề nghị một giải pháp lọc bỏ bớt các dữ liệu dư thừa không có ý nghĩa đối với vấn đề an ninh mạng và điều tra mạng nhằm giảm bớt dung lượng cần thiết cho việc lưu trữ dữ liệu nhật ký proxy trong một khoảng thời gian dài. Bài báo đã phân tích và tiến hành tìm hiểu các yêu cầu thực tế dựa trên các quan tâm của nhà quản trị mạng, các phần mềm phân tích nhật ký proxy và các bài báo liên quan đến vấn đề xử lý dữ liệu nhật ký proxy. Từ đó đề xuất mô hình lọc dữ liệu dựa trên công nghệ xử lý dữ liệu lớn là Storm áp dụng giải pháp lọc nhật ký proxy theo thời gian thực. Kết quả của việc áp dụng mô hình là giảm dung lượng lưu trữ tại kho nhật ký proxy dựa vào 3 bộ lọc chính: Bộ lọc dựa trên kiểu, Bộ lọc dựa trên kết quả truy cập và Bộ lọc trùng. Kết quả thử nghiệm cho thấy, dung lượng của dữ liệu nhật ký sau khi qua bộ lọc có thể giảm hơn 40% trong khi vẫn di trì được các thông tin liên quan đến vấn đề an ninh mạng mà các nhà quản trị mạng quan tâm.

Hiện tại mô hình đề xuất chỉ lọc trên dữ liệu nhật ký proxy. Trong tương lai có thể áp dụng mô hình đã đề xuất cho các loại dữ liệu nhật ký khác.

TÀI LIỆU THAM KHẢO

[1] Ankit Toshniwal and et al., (2014), "Storm@twitter", SIGMOD '14 Proceedings of the 2014 ACM SIGMOD international conference on Management of data, pp. 147-156.

[2] Apachar Spark, (2005), Log management for effective incident response, Network Security, vol. 9, pp. 4-7.

[3] Bùi Thị Thơm, (2015), Nghiên cứu xây dựng công cụ hỗ trợ phân tích gói tin trong điều tra mạng, Đại học Thái Nguyên

[4] Dario Forte, (2005), Log management for effective incident response.

[5] Harald Weinreich, Hartmut Obendorf, Eelco Herder, Data Cleaning Methods for Client and Proxy logs.

[6] Logstash (2015). <https://www.elastic.co/products/logstash>, accessed: 05/05/2015.

[7] Lê Nguyễn Hoài Phong, Phạm Văn Toàn, (2012), Triển khai Squid Proxy server và quản lý log bằng Sarg.

[8] Lê Thanh Sang, (2015), Nghiên cứu xây dựng kho dữ liệu nhật ký hệ thống phục vụ giám sát an ninh mạng tại trung tâm dữ liệu Đại học Cần Thơ, Trường Đại học Cần Thơ.

[9] Mathieu Gorge, (2007), Making sense of log management for security purposes - an approach to best practice log collection, analysis and management, Computer Fraud & Security, (5), pp: 5-10.

[10] Nuclear Automation, (December-2014), Data Processing and Monitoring System.

[11] Ms. Shashi Sahu1, Mr. Omprakash Dewanagan (June, 2015), Enhanced Log Cleaner with User and Session based Clustering for Effective Log Analysis, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 4 Issue 6.

[12] Marcus J. Ranum, (2014), System Logging and Log Analysis.

[13] Michael Noll, (Dec-2014), Running a Multi-Broker Apache Kafka 0.8 Cluster on a Single Node.

[14] M. Tim Jones (April 2013), Process real-time big data with Twitter Storm, IBM Developer Works, pp. 1-9.

[15] EMC Proven Professional Knowledge Sharing, (2013), "Analytics on big fast data using real time stream data processing architecture".

[16] Peter Zadrozny and Raghu Kodali, (2013), Big Data Analytics Using Splunk, Apress Media, LLC, New York City.

[17] Patricio Córdova, (2015), Analysis of Real Time Stream Processing Systems Considering Latency, University of Toronto.

[18] P. Taylor Goetz, Brian O'Neill, (March, 2014), Storm Blueprints: Patterns for Distributed Real-time Computation, Published by Packt Publishing Ltd, pp: 95 -124.

[19] Priyanka Verma, Dr.Nishtha Kesswani, Web Usage mining framework for Data Cleaning and IP address Identification.

[20] Priyanka Patil, Ujwala Patil, (2012), "Preprocessing of web server log file for web mining" in proceedings of National Conference on Emerging Trends in Computer Technology (NCETCT 2012),vol. 2, pp. 14-18.

[21] James Turnbull, (2014), The Logstash Book, Version v1.4.3 Publisher by You Lulu Inc.

[22] Jay Kreps, Neha Narkhede, Jun Rao (2015), A Distributed Messaging System for Log Processing. LinkedIn Corp. Kafka.

[23] Jason, M. O'Kane, A Gentle Introduction to ROS, chapter 4 Log messages.Publisher by Create Space Independent Publishing Platform, 2013.

[24] Jay Kreps, I Heart Logs. Publisher by O'Reilly Media, 2014.

[25] Quinton Anderson, (August 2013), Storm Real-time Processing Cookbook, Published by Packt Publishing Ltd, pp.51-89.

[26] SafeSquid SWG. https://www.safesquid.com/content-filtering/what-tools-can-be-used-analyze-generate-reports-safesquid-logs.

[27] Tom Rayder, (2013), Nagios Core Administration Cookbook, Packt Publishing, 2013.

[28] Trần Văn Cường, (2015), Giải pháp nền tảng cho hệ thống tích hợp dữ liệu lớn và không đồng nhất, Trường Đại học Quốc gia Hà Nội.

[29] Valdman J., (2001), Log File Analysis, Technical report, University of West Bohemia in Pilsen.

[30] http://www.exacttrend.com/proxylogexplorer/.

[31] http://kafka.apache.org/

[32] http://log-sharing.dreamhosters.com/

A METHOD FOR FILTERING PROXY LOG BASING ON BIG DATA TECHNOLOGY

Chau Le Sa Lin, Ngo Ba Hung, Dinh The An Huy

***ABSTRACT**: Proxy log has an importance signification in network security and network forensics. This paper proposes a method for filtering proxy log to eliminate the data that are not relevant to network security and network forensics to reduce the storage space for proxy log. The proposed method is built based on big data technologies. As a result, the proposed method can reduce the size of proxy log until 40%.*

***Key words:** Proxy log; Log filter; Network security; Network forensics, Big data..*