



CHƯƠNG 2

THU THẬP DỮ LIỆU

NỘI DUNG

1. Phương pháp thu thập dữ liệu
2. Bộ thu thập dữ liệu
3. Dữ liệu phiên
4. Dữ liệu gói tin đầy đủ
5. Dữ liệu kiểu chuỗi trong gói tin

1. Phương pháp thu thập dữ liệu

- ❑ Kết hợp của cả phần cứng và phần mềm, tạo và thu thập dữ liệu để phát hiện xâm nhập và phân tích dữ liệu NSM
- ❑ Chuyên gia phân tích dữ liệu giỏi cần biết rõ:
 - Các nguồn dữ liệu họ có
 - Nơi lấy được dữ liệu
 - Cách thu thập dữ liệu
 - Lý do thu thập
 - Những gì có thể làm với dữ liệu đó

1.1. Giới thiệu về thu thập dữ liệu

- ❑ Thu thập và phân tích dữ liệu là một công việc vô cùng quan trọng và mất nhiều thời gian
- ❑ Nhiều tổ chức thường không hiểu đầy đủ về dữ liệu của họ
- ❑ Không có cách tiếp cận có cấu trúc để xác định các nguy cơ có thể đến với tổ chức
- ❑ Hậu quả:
 - Nắm bắt lấy bất kỳ dữ liệu tùy biến nào có sẵn để xây dựng chương trình → Lượng dữ liệu quá lớn → Không đủ tài nguyên → Lọc dữ liệu bằng nhân công hoặc các công cụ phân tích không hiệu quả

ACF (applied collection framework)

- ❑ Là khung làm việc được xây dựng để làm giảm sự phức tạp của việc thu thập dữ liệu
- ❑ Giúp tổ chức đánh giá các nguồn dữ liệu cần tập trung trong quá trình thu thập dữ liệu
- ❑ Gồm bốn giai đoạn



ACF - Giai đoạn 1: Xác định nguy cơ

- ❑ Thay vì chỉ xác định các nguy cơ chung, cần xác định các mối nguy cơ cụ thể vào mục tiêu của tổ chức
- ❑ Trả lời câu hỏi: “Tình trạng xấu nhất liên quan đến khả năng sống còn của tổ chức là gì?”
 - Đây là lý do mà chuyên gia an ninh thông tin thường phải cần làm việc với lãnh đạo cấp cao trong giai đoạn đầu của việc xác định yêu cầu thu thập dữ liệu
- ❑ Các nguy cơ thường tác động đến:
 - Tính bí mật
 - Tính toàn vẹn
 - Tính sẵn sàng

ACF - Giai đoạn 1: Xác định nguy cơ

- ❑ Từ nguy cơ đã xác định → thấy được các kỹ thuật và công nghệ cần sử dụng để giải quyết
- ❑ Ví dụ, trong trường hợp nguy cơ lớn nhất với tổ chức là mất tài sản trí tuệ, cần nghiên cứu sâu hơn với câu hỏi:
 - Những thiết bị nào tạo ra dữ liệu nghiên cứu thô, và làm thế nào để dữ liệu đi qua mạng?
 - Nhân viên xử lý dữ liệu nghiên cứu thô bằng những thiết bị nào?
 - Dữ liệu nghiên cứu đã xử lý được lưu trữ trên những thiết bị nào?
 - Ai có quyền truy cập vào dữ liệu nghiên cứu thô và dữ liệu nghiên cứu đã xử lý?
 - Dữ liệu nghiên cứu thô và dữ liệu nghiên cứu đã xử lý có sẵn bên ngoài mạng hay không?
 - Đường dẫn nào bên trong mạng nội bộ có sẵn ở bên ngoài?
 - Mức độ truy cập của làm nhân viên tạm vào dữ liệu nghiên cứu?

ACF - Giai đoạn 1: Xác định nguy cơ

□ Từ đó, có thể xác định được một danh sách các hệ thống có thể bị tấn công, dẫn đến tổn thất về tài sản trí tuệ.

□ Ví dụ như:

- Máy chủ web (web server),
- Máy chủ cơ sở dữ liệu (database server),
- Máy chủ lưu trữ tệp tin (file server),...

ACF - Giai đoạn 2: Định lượng rủi ro

- ❑ Khi xác định được một danh sách các nguy cơ, cần xác định xem nguy cơ nào cần được ưu tiên
- ❑ Thực hiện bằng cách tính toán rủi ro gây ra bởi các nguy cơ tiềm ẩn:

$$\textbf{Ảnh hưởng (I)} \times \textbf{Xác suất (P)} = \textbf{Rủi ro (R)}$$

- ✓ Ảnh hưởng là tác động của nguy cơ đến tổ chức
- ✓ Xác suất là khả năng nguy cơ xuất hiện
- ✓ Mức độ rủi ro mà nguy cơ gây ra đối với sự an toàn của mạng

ACF - Giai đoạn 3:

Xác định nguồn dữ liệu

- ❑ Đi từ nguy cơ có hệ số rủi ro cao nhất, và xem xét bằng chứng thể hiện nguy cơ có thể được nhìn thấy
- ❑ Ví dụ, để kiểm tra nguy cơ tấn công máy chủ lưu trữ tệp tin, cần:
 - Xác định cấu trúc của máy chủ
 - Vị trí trên mạng
 - Người có quyền truy cập
 - Đường dẫn mà dữ liệu đi vào
- ❑ Dựa vào đó để kiểm tra cả hai nguồn dữ liệu dựa trên mạng và dựa trên máy chủ

ACF - Giai đoạn 3: Xác định nguồn dữ liệu

□ Ví dụ về danh sách các loại nguồn dữ liệu

■ Dựa trên mạng:

- Máy chủ lưu trữ tệp tin VLAN – Dữ liệu bắt gói tin đầy đủ
- Máy chủ lưu trữ tệp tin VLAN – Dữ liệu phiên
- Máy chủ lưu trữ tệp tin VLAN – Dữ liệu thống kê thông lượng
- Máy chủ lưu trữ tệp tin VLAN – Dữ liệu cảnh báo NIDS dựa theo chữ ký
- Máy chủ lưu trữ tệp tin VLAN – Dữ liệu cảnh báo IDS dựa theo bất thường
- Upstream Router – Dữ liệu nhật ký tường lửa

■ Dựa trên máy chủ:

- Máy chủ lưu trữ tệp tin – Dữ liệu nhật ký sự kiện OS
- Máy chủ lưu trữ tệp tin – Dữ liệu cảnh báo vi-rút
- Máy chủ lưu trữ tệp tin – Dữ liệu cảnh báo HIDS

ACF - Giai đoạn 4:

Chọn lọc dữ liệu

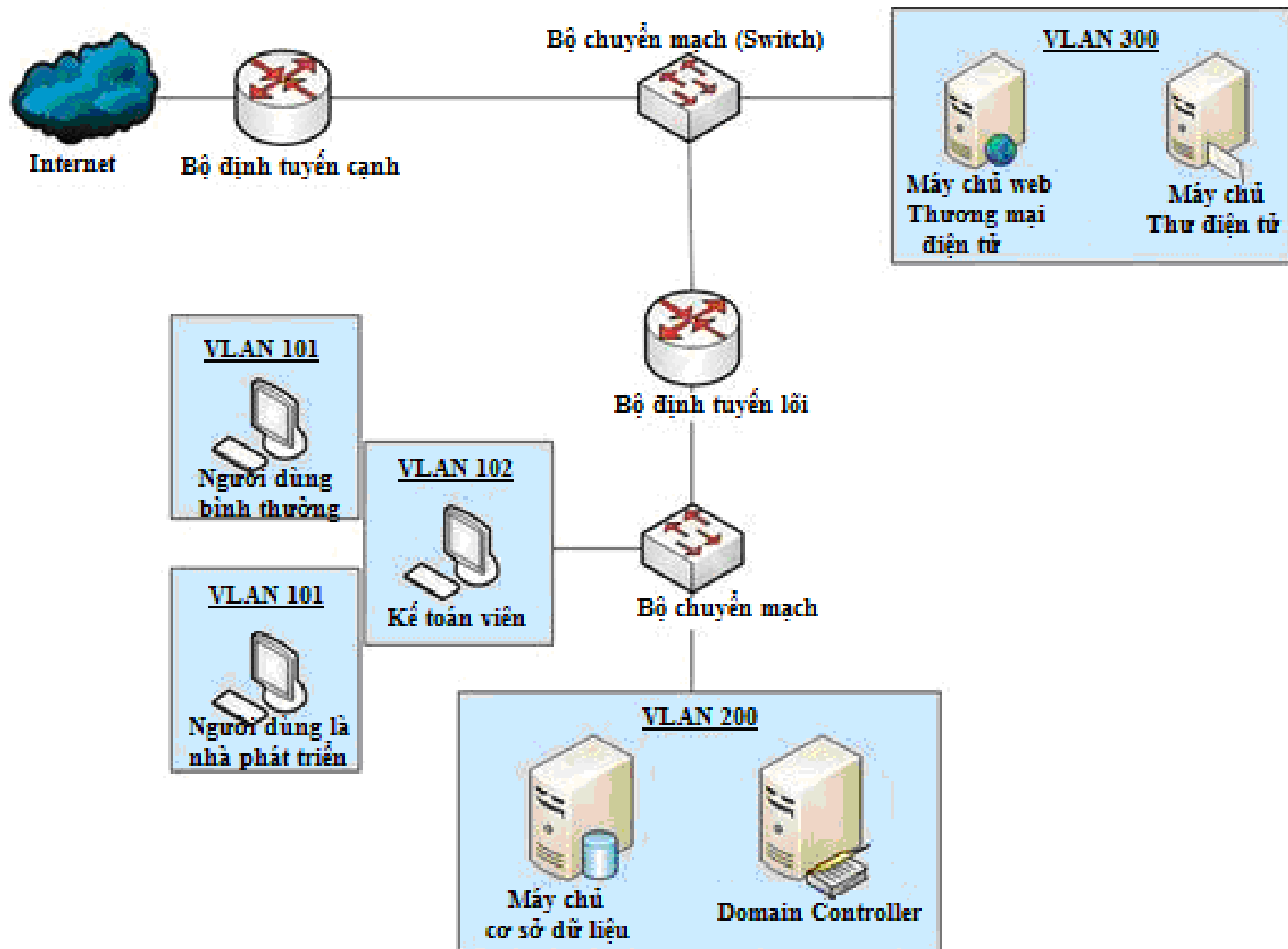
- ❑ Liên quan đến các bước kỹ thuật chiều sâu và cần phải xem xét tất cả các nguồn dữ liệu riêng để xác định giá trị của nó
 - Ví dụ một nguồn dữ liệu rất lớn, việc lưu trữ, xử lý và quản lý có thể lớn hơn nhiều so với giá trị mà nó mang lại, thì sẽ không phải là nguồn dữ liệu tốt
- ❑ Cần phân tích chi phí/lợi ích của các nguồn dữ liệu
 - Tài nguyên phần cứng, phần mềm, nhân công, việc tổ chức và lưu trữ dữ liệu,...
 - Số lượng dữ liệu và thời gian lưu trữ dữ liệu
 - Cần phải giảm tối thiểu chi phí lưu trữ dữ liệu và tăng tối đa độ quan tâm về dữ liệu hữu ích dùng trong việc phân tích

ACF - Giai đoạn 4: Chọn lọc dữ liệu

- ❑ Trên cơ sở đó, xây dựng cơ sở hạ tầng thích hợp cho việc thu thập dữ liệu
- ❑ Dữ liệu liên tục được thu thập, được sử dụng cho phát hiện xâm nhập và phân tích theo sự phát triển hệ thống mạng của tổ chức, và sẽ luôn cần phải xem xét lại chiến lược thu thập dữ liệu

1.2. Ví dụ tình huống: Cửa hàng bán lẻ

- ❑ Thiết lập một hệ thống NSM cho cửa hàng bán lẻ trực tuyến, sử dụng trang web. Toàn bộ doanh thu là từ việc bán hàng qua trang web
- ❑ Sơ đồ mạng gồm:
 - Máy chủ truy nhập công khai trong một DMZ, nằm phía trong bộ định tuyến
 - Người dùng và máy chủ mạng nội bộ ở các VLAN khác nhau bên trong bộ định tuyến lõi
 - Chưa có bất kỳ cảm biến nào do chưa xác định được nhu cầu thu thập dữ liệu



Sơ đồ mạng của cửa hàng bán lẻ

Bước 1: Xác định nguy cơ

- ❑ Tính bảo mật: trang web thu thập và lưu trữ các thông tin của khách hàng trong CSDL.
 - Có thể bị tấn công vào CSDL qua trang web
- ❑ Tính sẵn sàng: Kẻ tấn công có thể thực hiện một cuộc tấn công làm cho trang web thương mại điện tử không tiếp cận được với khách hàng
 - Tấn công từ chối dịch vụ
- ❑ Tính toàn vẹn: Kẻ tấn công có thể thực hiện một cuộc tấn công trong đó cho phép họ dùng ứng dụng web một cách không có chủ ý
 - Ví dụ: mua sản phẩm mà không có giao dịch về tiền, tấn công người dùng để truy cập vào phần back-end

Bước 2: Định lượng rủi ro

Nguy cơ	Ảnh hưởng	Xác suất	Rủi ro
Đánh cắp thông tin thẻ tín dụng của khách hàng – tấn công ứng dụng web	4	4	16
Đánh cắp thông tin thẻ tín dụng của khách hàng – tấn công người dùng nội mạng	4	2	8
Làm gián đoạn các dịch vụ thương mại điện tử – DoS	4	2	8
Làm gián đoạn các dịch vụ thương mại điện tử – tấn công tài sản bên ngoài	5	3	15
Làm gián đoạn các dịch vụ thương mại điện tử – tấn công tài sản nội mạng	5	2	10
Sử dụng dịch vụ thương mại điện tử không chủ ý – tấn công ứng dụng web	2	4	8
Sử dụng dịch vụ thương mại điện tử không chủ ý – tấn công tài sản nội mạng	2	1	2

Ưu tiên những nguy cơ có rủi ro cao

Nguy cơ	Ảnh hưởng	Xác suất	Rủi ro
Đánh cắp thông tin thẻ tín dụng của khách hàng – tấn công ứng dụng web	4	4	16
Làm gián đoạn các dịch vụ thương mại điện tử – tấn công tài sản bên ngoài	5	3	15
Làm gián đoạn các dịch vụ thương mại điện tử – tấn công tài sản nội mạng	5	2	10
Sử dụng dịch vụ thương mại điện tử không chủ ý – tấn công ứng dụng web	2	4	8
Làm gián đoạn các dịch vụ thương mại điện tử – DoS	4	2	8
Đánh cắp thông tin thẻ tín dụng của khách hàng – tấn công từ bên trong nội bộ	4	2	8
Sử dụng dịch vụ thương mại điện tử không chủ ý – tấn công tài sản nội mạng	2	1	2

Bước 3: Xác định nguồn dữ liệu

❑ **Với nguy cơ: Đánh cắp thông tin thẻ tín dụng của khách hàng – tấn công ứng dụng web. Ta có thể:**

- Thu thập và kiểm tra các giao dịch máy chủ web với người dùng bên ngoài để phát hiện ra những hành vi bất thường
 - có thể đặt một bộ cảm biến ở cạnh mạng
- Thu thập dữ liệu nhật ký ứng dụng cụ thể của các máy chủ web
- Kiểm tra các giao dịch đến máy chủ cơ sở dữ liệu
 - cần đặt một cảm biến thứ hai có khả năng hiển thị trong mạng nội bộ
- Thu thập dữ liệu về các bản ghi ứng dụng cụ thể của các máy chủ cơ sở dữ liệu để xem xét các hoạt động của nó

Bước 3: Xác định nguồn dữ liệu

□ Kế hoạch này tạo ra danh sách các nguồn dữ liệu như sau:

- ❖ Dữ liệu bắt gói tin đầy đủ, dữ liệu phiên, dữ liệu kiểu chuỗi trong gói tin, sử dụng NIDS dựa trên chữ ký và NIDS dựa trên bất thường, được thu thập qua cảm biến DMZ.
- ❖ Dữ liệu bắt gói tin đầy đủ, dữ liệu phiên, dữ liệu kiểu chuỗi trong gói tin, sử dụng NIDS dựa trên chữ ký và NIDS dựa trên bất thường, được thu thập qua cảm biến nội mạng.
- ❖ Dữ liệu nhật ký ứng dụng máy chủ web
- ❖ Dữ liệu nhật ký ứng dụng máy chủ cơ sở dữ liệu

Bước 3: Xác định nguồn dữ liệu

❑ **Với nguy cơ: Làm gián đoạn các dịch vụ thương mại điện tử – tấn công tài sản bên ngoài.**

- Có thể bao gồm cả tấn công ứng dụng web.
- Có hai tài sản bên ngoài cần bảo vệ là máy chủ web, và máy chủ thư điện tử
- Dữ liệu nhật ký tường lửa là nguồn dữ liệu điều tra rất hữu ích.
- cần có một cảm biến để thu thập dữ liệu qua giao diện mạng.
- Cần thu thập nhật ký cụ thể của ứng dụng, bao gồm nhật ký máy chủ web, cơ sở dữ liệu và thư điện tử.
- Cần thu thập thêm nhật ký bảo mật và hệ điều hành, cùng với dữ liệu nhật ký chống vi-rút và dữ liệu cảnh báo IDS dựa trên máy chủ.

Bước 3: Xác định nguồn dữ liệu

□ Kế hoạch này tạo ra danh sách các nguồn dữ liệu như sau:

- ❖ Dữ liệu nhật ký tường lửa cạnh mạng
- ❖ Dữ liệu bắt gói tin đầy đủ, dữ liệu phiên, dữ liệu kiểu chuỗi trong gói tin, sử dụng NIDS dựa trên chữ ký và NIDS dựa trên bất thường, được thu thập qua cảm biến DMZ
- ❖ Dữ liệu nhật ký ứng dụng máy chủ cơ sở dữ liệu
- ❖ Dữ liệu nhật ký ứng dụng máy chủ thư điện tử
- ❖ Dữ liệu nhật ký bảo mật và hệ điều hành của máy chủ thư điện tử và máy chủ web
- ❖ Dữ liệu cảnh báo chống vi-rút của máy chủ thư điện tử và máy chủ web
- ❖ Dữ liệu cảnh báo HIDS của máy chủ thư điện tử và máy chủ web

Bước 3: Xác định nguồn dữ liệu

❑ Với nguy cơ: Làm gián đoạn các dịch vụ thương mại điện tử – tấn công tài sản nội mạng.

- Chỉ có các máy chủ trong VLAN 200 và những người dùng là nhà phát triển trong VLAN 103 là có quyền truy nhập vào DMZ từ bên trong mạng
 - cần triển khai một cảm biến ở bên trong mạng để thu thập các dữ liệu từ các thiết bị này
- Nếu kẻ tấn công chiếm được quyền sử dụng máy của người dùng là nhà phát triển trong nội mạng, hắn sẽ có quyền truy nhập đến DMZ, tác động đến DNS
 - cần thu thập dữ liệu của các hệ thống có liên quan và các nhật ký bảo mật, dữ liệu cảnh báo HIDS và chống vi-rút, thu thập nhật ký tường lửa từ các bộ định tuyến nội mạng, từ DNS

Bước 3: Xác định nguồn dữ liệu

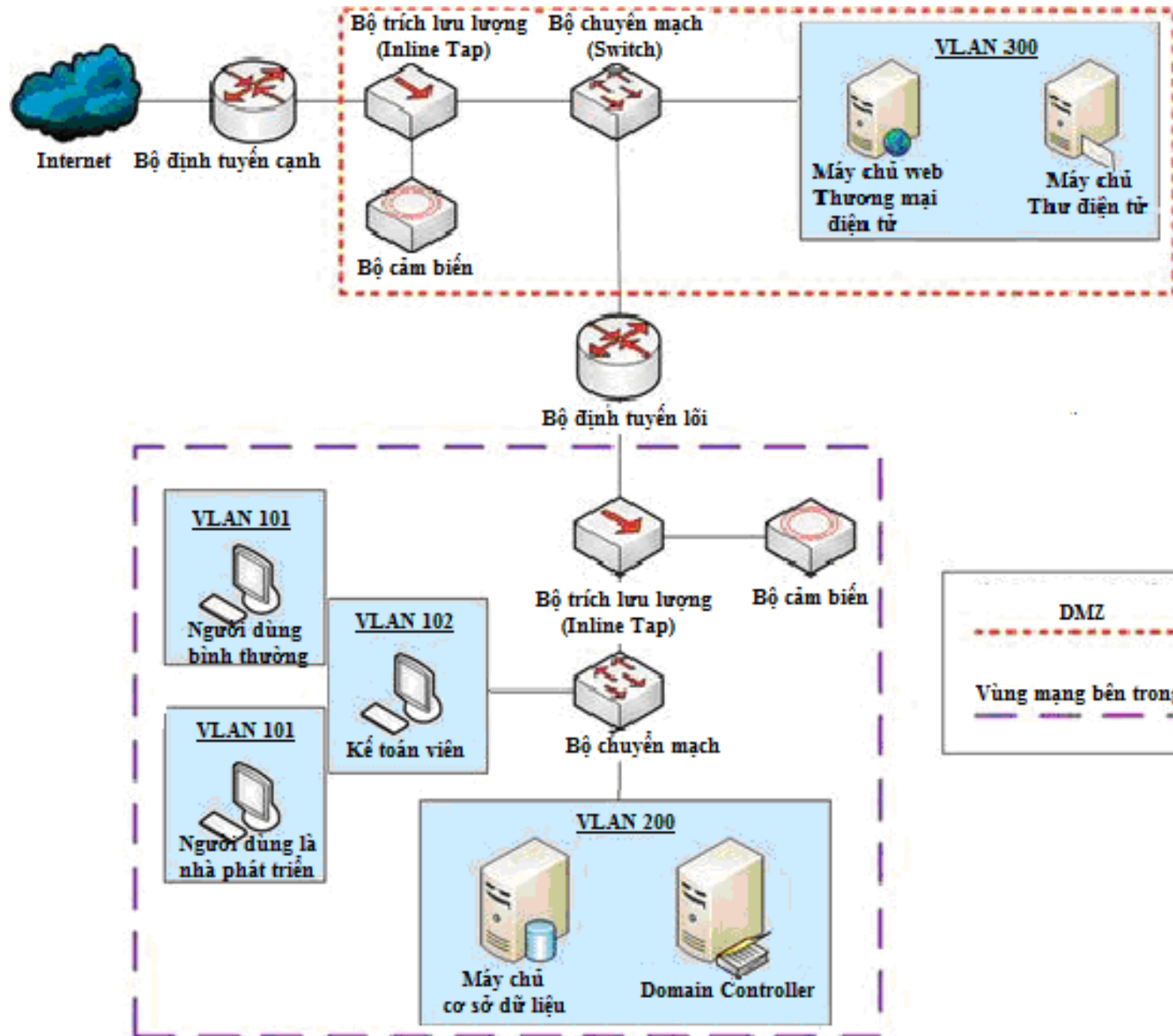
□ Kế hoạch này tạo ra danh sách các nguồn dữ liệu như sau:

❖ Dựa trên mạng:

- Dữ liệu nhật ký tường lửa bên cạnh mạng, bên trong mạng
- Dữ liệu bắt gói tin đầy đủ, dữ liệu phiên, sử dụng NIDS dựa trên chữ ký và NIDS dựa trên bất thường, được thu thập qua cảm biến DMZ
- Dữ liệu bắt gói tin đầy đủ, dữ liệu phiên, dữ liệu kiểu chuỗi trong gói tin, sử dụng NIDS dựa trên chữ ký và NIDS dựa trên bất thường, được thu thập qua cảm biến nội mạng

❖ Dựa trên máy chủ:

- Nhật ký dữ liệu máy chủ web, cơ sở dữ liệu, và ứng dụng điều khiển miền.
- Dữ liệu nhật ký bảo mật và hệ điều hành máy chủ web, VLAN 200 và VLAN 103
- Dữ liệu cảnh báo chống vi-rút máy chủ web, VLAN 200 và VLAN 103
- Dữ liệu cảnh báo HIDS máy chủ web, VLAN 200 và VLAN 103



Sơ đồ mạng mới với các cảm biến

Bước 4: Chọn lọc dữ liệu

□ Dựa trên mạng:

- Dữ liệu nhật ký tường lửa bên cạnh mạng
 - Bên trong → Từ chối bên ngoài
- Dữ liệu nhật ký tường lửa bên trong (lỗi mạng)
 - Bên ngoài → Cho phép/Từ chối bên trong
 - Bên trong → Từ chối bên ngoài
- Cảm biến DMZ – Dữ liệu bắt gói tin đầy đủ
 - Bên ngoài → Các cổng web bên trong
 - Bên ngoài → Các cổng thư điện tử bên trong
 - Bên trong → Các cổng thư điện tử bên ngoài
- Cảm biến DMZ – Dữ liệu phiên
 - Tất cả các bản ghi

Bước 4: Chọn lọc dữ liệu

□ Dựa trên mạng:

- Cảm biến DMZ – NIDS dựa trên chữ ký
 - Các luật tập trung vào tấn công ứng dụng web: SQL injection, XSS,...
 - Các luật tập trung vào tấn công máy chủ web
 - Các luật tập trung vào tấn công máy chủ thư điện tử
- Cảm biến DMZ –NIDS dựa trên bất thường
 - Các luật tập trung vào những bất thường trong nội dung thư và web
- Cảm biến nội mạng – Dữ liệu bắt gói tin đầy đủ
 - Bên trong → Các IP máy chủ web
 - Bên trong → Nhà phát triển VLAN 103
 - Bên ngoài → Máy chủ VLAN 200

Bước 4: Chọn lọc dữ liệu

❑ Dựa trên mạng:

- Cảm biến nội mạng – Dữ liệu phiên
 - Tất cả các bản ghi
- Cảm biến nội mạng – Dữ liệu kiểu chuỗi trong gói tin
 - Nhà phát triển VLAN 103 → Bên ngoài
- Cảm biến nội mạng – NIDS dựa trên chữ ký
 - Các luật tập trung vào tấn công cơ sở dữ liệu
 - Các luật tập trung vào tấn công và các hoạt động quản trị bộ điều khiển miền
 - Các luật phần mềm độc hại chung
- Cảm biến nội mạng – NIDS dựa trên bất thường
 - Các luật tập trung vào tương tác cơ sở dữ liệu bất thường

Bước 4: Chọn lọc dữ liệu

□ Dựa trên máy chủ:

- Dữ liệu nhật ký máy chủ thư điện tử, máy chủ web, máy chủ cơ sở dữ liệu và ứng dụng điều khiển miền
 - Máy chủ thư điện tử – Tạo và sửa đổi tài khoản
 - Máy chủ web – Các giao dịch từ miền con xử lý thanh toán
 - Máy chủ web – Các giao dịch từ miền con quản trị
 - Máy chủ cơ sở dữ liệu – Tạo và sửa đổi tài khoản
 - Máy chủ cơ sở dữ liệu – Các giao dịch thanh toán
 - Máy chủ cơ sở dữ liệu – Các giao dịch quản trị
 - Bộ điều khiển miền– Tạo và sửa đổi tài khoản
 - Bộ điều khiển miền– Tạo và sửa đổi máy tính

Bước 4: Chọn lọc dữ liệu

□ Dựa trên máy chủ:

- Dữ liệu nhật ký bảo mật và hệ điều hành máy chủ thư điện tử, máy chủ web, VLAN 200 và VLAN 103
 - Tạo và sửa đổi tài khoản
 - Các thông báo phần mềm được cài đặt
 - Các thông báo cập nhật hệ thống
 - Thông báo khởi động lại hệ thống
- Dữ liệu cảnh báo chống vi-rút máy chủ thư điện tử, máy chủ web, VLAN 200 và VLAN 103
 - Tất cả dữ liệu cảnh báo
- Dữ liệu cảnh báo HIDS máy chủ thư điện tử, máy chủ web và VLAN 103 Alert Data
 - Cảnh báo liên quan đến những thay đổi tệp tin hệ thống chính
 - Thay đổi liên quan đến tạo/sửa đổi tài khoản.

2. Bộ thu thập dữ liệu

- ❑ Ngoài con người, cảm biến là thành phần quan trọng nhất trong các hệ thống NSM
- ❑ Mỗi cảm biến là một thiết bị phát hiện hoặc đo lường tính chất vật lý hoặc các bản ghi, chỉ báo hoặc đáp ứng với nó
- ❑ Trong NSM, cảm biến là một sự kết hợp của phần cứng và phần mềm được sử dụng để thực hiện một hoặc một số chức năng trong chu trình NSM là **thu thập dữ liệu, phát hiện xâm nhập và phân tích dữ liệu**

2.1. Các loại dữ liệu NSM

□ Dữ liệu

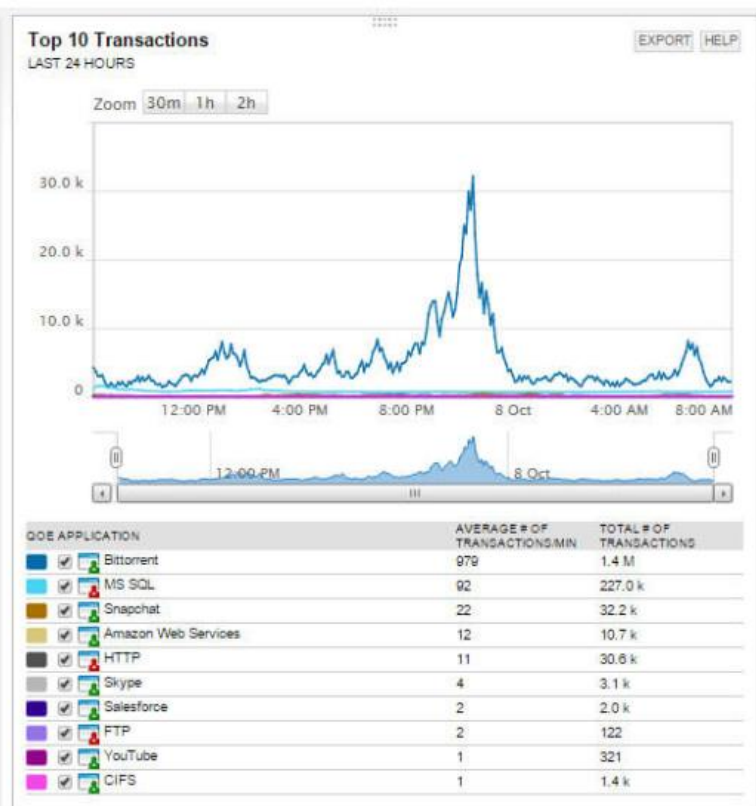
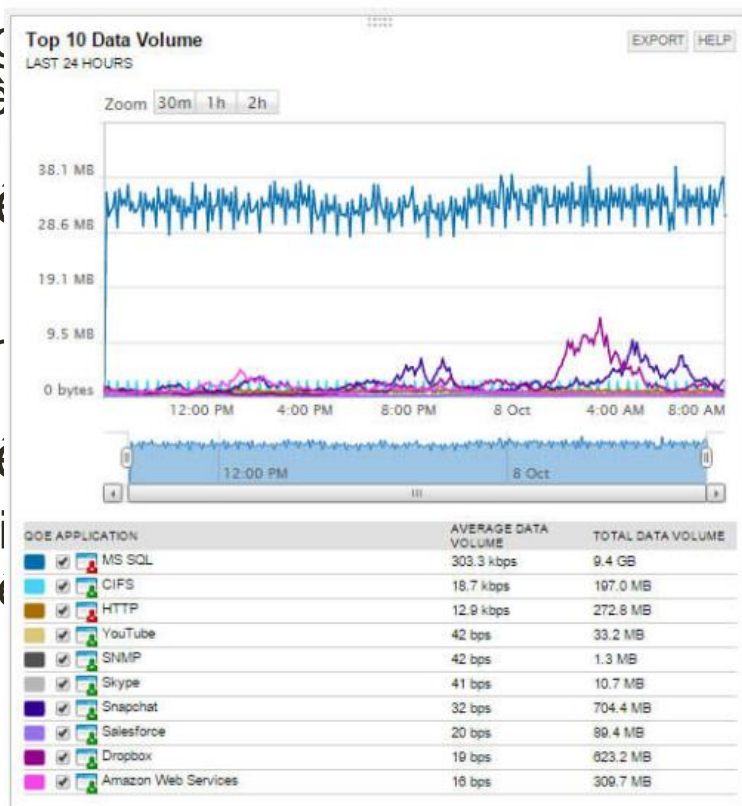
- Cung cấp thông tin về lưu lượng truyền

□ Dữ liệu

- Tóm tắt lưu lượng
- Không phân tích chi tiết

□ Dữ liệu

- Dữ liệu chi tiết về lưu lượng



Các loại dữ liệu NSM

- ❑ Dữ liệu kiểu chuỗi trong gói tin (PSTR)
 - Lấy từ dữ liệu FPC, và tồn tại như một dạng dữ liệu trung gian giữa dữ liệu FPC và dữ liệu phiên.
 - Ví dụ: chuỗi văn bản rõ từ tiêu đề (header) của các giao thức (dữ liệu trong phần tiêu đề của HTTP)
- ❑ Dữ liệu nhật ký
 - Tác tệp tin nhật ký thô được tạo ra từ thiết bị, hệ thống hoặc ứng dụng.
 - Ví dụ: nhật ký web-proxy, nhật ký tường lửa, dữ liệu SYSLOG ...
- ❑ Dữ liệu cảnh báo
 - Mô tả của các cảnh báo, và con trỏ chỉ đến dữ liệu bất thường
 - Kích thước nhỏ.

2.2. Phân loại

- ❑ **Cảm biến chỉ thu thập dữ liệu (collection-only sensor)**
 - Ghi nhật ký những dữ liệu đã thu thập như FPC và dữ liệu phiên vào đĩa, và đôi khi tạo ra dữ liệu khác
 - Thường được dùng trong các tổ chức lớn, các công cụ phát hiện xâm nhập cần truy nhập dữ liệu thu thập từ xa để thực hiện xử lý
- ❑ **Cảm biến nửa chu trình (half-cycle sensor)**
 - Thực hiện tất cả các chức năng của một bộ cảm biến chỉ thu thập dữ liệu, với việc bổ sung thực hiện nhiệm vụ phát hiện xâm nhập.
 - Ví dụ: ghi dữ liệu PCAP vào ổ đĩa, nhưng cũng sẽ chạy một NIDS
 - Khi thực hiện phân tích, dữ liệu sẽ được đưa trở lại thiết bị khác thay vì được phân tích trên chính cảm biến
 - Loại cảm biến này được triển khai phổ biến nhất

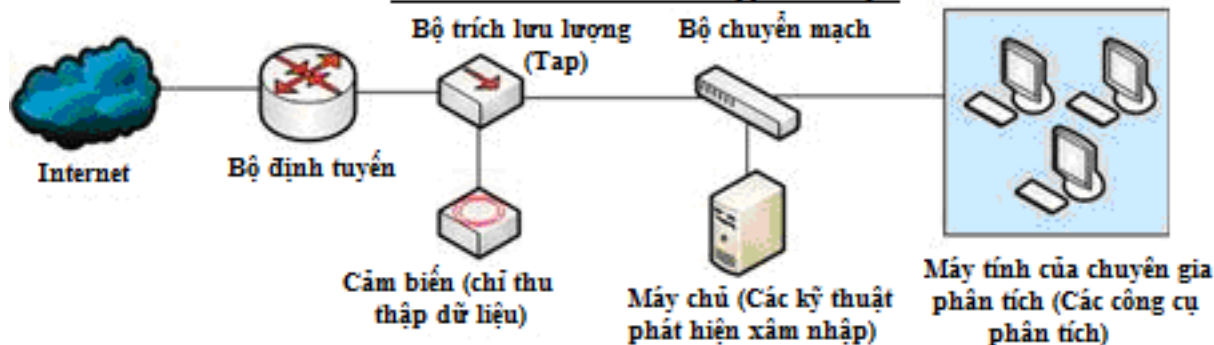
Phân loại

❑ Cảm biến phát hiện chu trình đầy đủ (full cycle detection sensor)

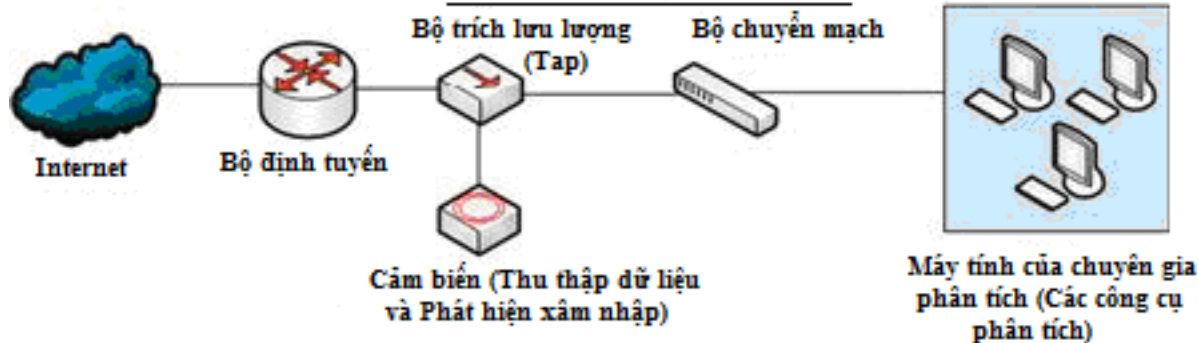
- Thực hiện đầy đủ các chức năng của chu trình NSM, bao gồm thu thập dữ liệu, phát hiện xâm nhập và phân tích dữ liệu
- Hầu hết các nhiệm vụ của NSM đều được thực hiện trên chính cảm biến
- Thường được dùng trong các tổ chức rất nhỏ
- Trong 3 loại cảm biến, sử dụng cảm biến nửa chu trình nhiều nhất, do:
 - ❖ Dễ dàng cài đặt các công cụ phát hiện trên cùng hệ thống mà dữ liệu được thu thập
 - ❖ An toàn hơn do không tương tác trực tiếp với dữ liệu thô

Phân loại

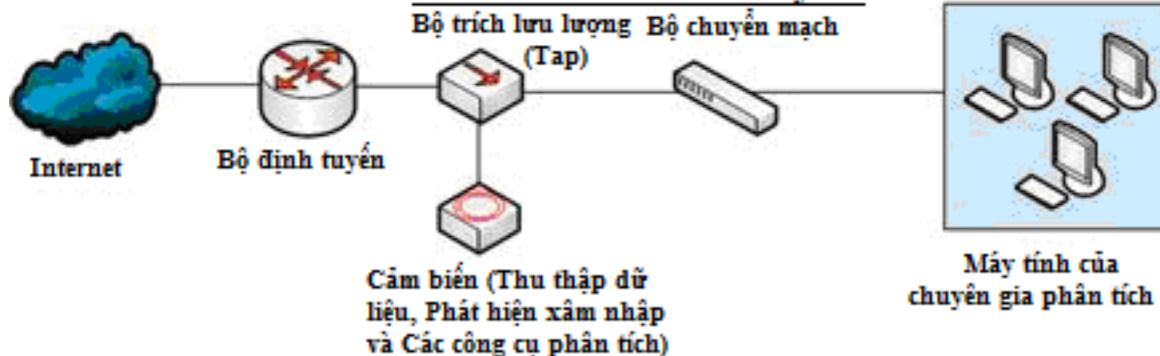
Cảm biến chỉ thu thập dữ liệu



Cảm biến nửa chu trình



Cảm biến chu trình đầy đủ



2.3. Phần cứng

- ❑ Phần cứng tin cậy, nên thuộc cấp độ của máy chủ
- ❑ Cần xác định số lượng tài nguyên phần cứng cần thiết bao gồm:
 - Các loại cảm biến được triển khai
 - Số lượng dữ liệu được thu thập bởi các cảm biến
 - Số lượng dữ liệu cần được lưu giữ

Phần cứng

- ❑ Cách thường dùng là thiết lập và cấu hình một cảm biến tạm thời
 - Xác định vị trí cần cài đặt trên mạng
 - Sử dụng một cổng SPAN (SPAN port) hoặc một bộ trích dữ liệu mạng (network tap) để dẫn lưu lượng dữ liệu vào thiết bị
 - Cài đặt các công cụ thu thập dữ liệu, phát hiện xâm nhập và phân tích dữ liệu vào các cảm biến để xác định các yêu cầu về hiệu suất của các công cụ riêng lẻ

Phần cứng

☐ Chú ý:

- CPU: phụ thuộc loại cảm biến triển khai. Cảm biến phát hiện xâm nhập cần nhiều CPU
- Bộ nhớ: cũng phụ thuộc vào loại cảm biến. Nên để khe cắm trống để nâng cấp sau này
- Ổ cứng lưu trữ: tùy thuộc loại cảm biến, cần đánh giá lại thường xuyên

☐ Các bước cần cho đánh giá lưu trữ:

- Tính toán lưu lượng thu thập
- Xác định thời gian lưu trữ khả thi cho mỗi loại dữ liệu
- Bổ sung nhu cầu lưu trữ cho các loại cảm biến

Phần cứng

❑ Giao diện mạng:

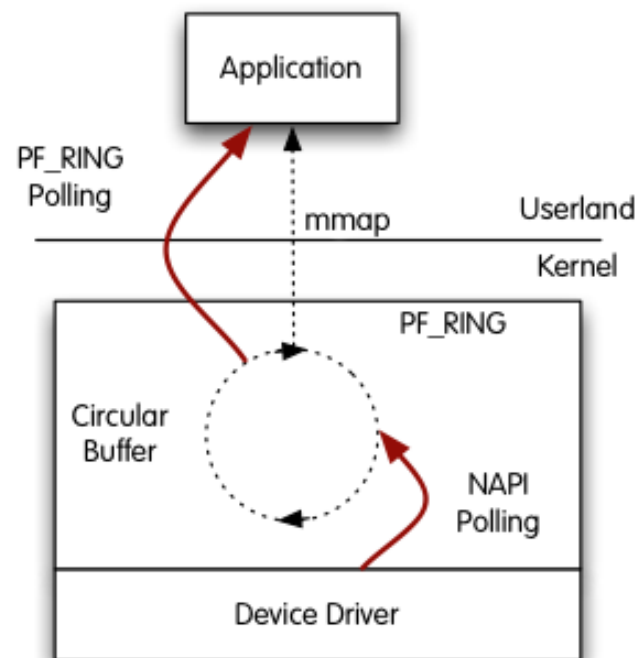
- Là thành phần phần cứng quan trọng nhất trong các cảm biến.
- Mỗi cảm biến nên luôn có tối thiểu hai NIC, một để truy cập vào máy chủ, hoặc quản trị hoặc phân tích dữ liệu, cái còn lại để thu thập dữ liệu
- Số lượng NIC được sử dụng sẽ phụ thuộc vào lượng băng thông gửi qua liên kết và các bộ trích dữ liệu mạng
- Cần đánh giá về lưu lượng mạng sẽ thu thập để xác định nhu cầu về NIC
 - Ví dụ: đánh giá lượng truy cập vào một liên kết thông qua việc giám sát trên một bộ định tuyến hoặc một chuyển mạch, dựa trên: (1) đỉnh điểm của lưu lượng (đo bằng Mbps), và (2) băng thông trung bình (thông lượng) mỗi ngày (đo bằng Mbps)

Phần cứng

❏ Cân bằng tải: Yêu cầu vùng đệm socket:

- Khi lưu lượng mạng đã được đưa đến card mạng, cần xem xét vấn đề cân bằng tải trong cảm biến qua các luồng ứng dụng hoặc luồng xử lý khác nhau
- Ví dụ: vùng đệm socket mạng Linux truyền thống không phù hợp với phân tích lưu lượng hiệu năng cao. PF_Ring (thư viện xử lý gói tin) của Luca Deri thì lại phù hợp, hỗ trợ cả Bro, Snort, hoặc Suricata

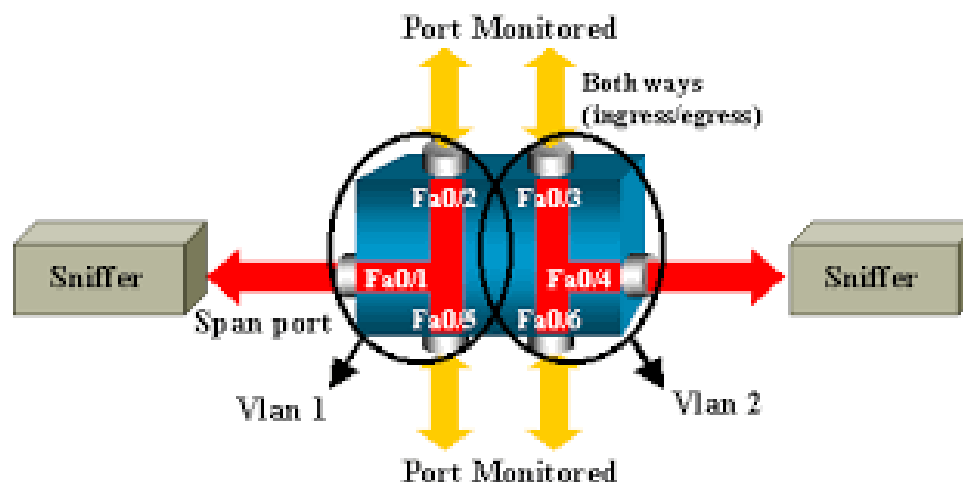
- (1) từng gói tin luân chuyển theo vòng,
- (2) đảm bảo toàn bộ dòng lưu chuyển gói tin được chuyển giao cho một quá trình duy nhất hoặc đi đến cảm biến



Vanilla PF_RING

Phần cứng

- ❑ Các cổng SPAN và bộ trích dữ liệu mạng (network tap):
 - Là thiết bị thu các gói tin đến các bộ cảm biến
- ❑ Cổng SPAN là cách đơn giản nhất để thu được các gói tin đến cảm biến do là chức năng của switch



2.4. Hệ điều hành

- ❑ Phổ biến nhất là Linux hoặc BSD
- ❑ Nền tảng hệ điều hành được chọn là không quá quan trọng
- ❑ Thường dựa trên *nix do hầu hết các công cụ được thiết kế để thu thập dữ liệu, phát hiện xâm nhập và phân tích dữ liệu được xây dựng để làm việc trên các nền tảng này

2.5. Vị trí đặt

- ❑ Quyết định quan trọng nhất phải được thực hiện khi lập kế hoạch thu thập dữ liệu NSM là vị trí vật lý đặt các cảm biến trên mạng
- ❑ Vị trí này quyết định:
 - Có thể bắt được dữ liệu gì
 - Phát hiện nào có thể có được liên quan đến dữ liệu đó
 - Mức độ mở rộng cho việc phân tích được đến đâu

Cách xác định vị trí đặt

❑ Sử dụng các tài nguyên thích hợp

- Nên tích cực tham gia vào quá trình sắp đặt mạng ngay trong giai đoạn đầu, nhằm hiểu rõ nhất về cấu trúc và thiết kế sơ đồ mạng của tổ chức

❑ Các điểm đi vào/đi ra mạng

- Lý tưởng là nên đặt một bộ cảm biến ngay tại điểm đi vào/đi ra mạng
 - như cổng gateway của Internet, các mạng VPN truyền thống, và các liên kết đối tác
- Trong các mạng nhỏ hơn, có thể triển khai cảm biến tại đường biên trên cạnh của mạng

Cách xác định vị trí đặt

- ❑ Tầm nhìn của địa chỉ Internet cục bộ
 - Quan trọng là khả năng xác định thiết bị nội bộ nào là đối tượng chính của một cảnh báo
- ❑ Đánh giá tài sản quan trọng
 - Cần phải có quy định tài sản nào là quan trọng nhất cần bảo vệ
 - Từ đó có thể đặt các cảm biến một cách hợp lý, gần nhất với những tài sản quan trọng
- ❑ Tạo các sơ đồ hiển thị cảm biến
 - Quan trọng khi được dùng để tham khảo cho quá trình điều tra của các chuyên gia phân tích
 - Mục tiêu của sơ đồ mạng là cho các chuyên gia phân tích nhanh chóng biết được những tài sản nào mà một cảm biến bảo vệ và những tài sản nào đã ra ngoài vùng bảo vệ đó

Cách xác định vị trí đặt

❑ Các thành phần cần thiết nhất của một sơ đồ mạng bao gồm:

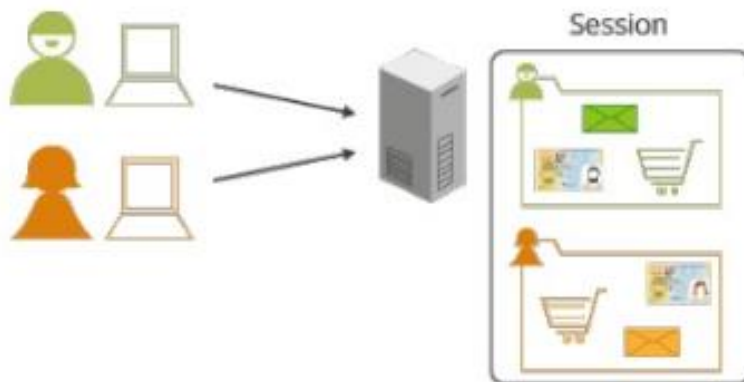
- Khái quát logic mức cao của mạng
- Tất cả các thiết bị định tuyến, proxy, hoặc gateway có ảnh hưởng đến lưu lượng mạng
- Địa chỉ IP trong/ngoài của thiết bị định tuyến, proxy, và các gateway
- Máy trạm, máy chủ hoặc các thiết bị khác - nên được hiển thị theo nhóm trừ khi đó là các thiết bị đặc biệt quan trọng
- Dải địa chỉ IP cho các nhóm máy trạm, máy chủ, và các thiết bị
- Tất cả các cảm biến NSM, và các vùng/khu vực phù hợp mà cảm biến có trách nhiệm bảo vệ.

2.6. Bảo mật

- ❑ Sự an toàn của các cảm biến nên được coi là tối quan trọng do chứa các thông tin mạng vô cùng nhạy cảm
- ❑ Một số bước có thể được thực hiện để đảm bảo sự an toàn cho các cảm biến:
 - Cập nhật hệ điều hành và phần mềm
 - Bảo mật hệ điều hành
 - Hạn chế truy cập Internet
 - Tối thiểu hóa cài đặt phần mềm
 - Phân đoạn VLAN
 - IDS dựa trên máy chủ
 - Hai yếu tố xác thực
 - IDS dựa trên mạng

3. Dữ liệu phiên

- ❑ Là bản tóm tắt các thông tin liên lạc giữa hai thiết bị mạng
- ❑ Như là một cuộc hội thoại hoặc một luồng lưu lượng
- ❑ Là một trong những hình thức linh hoạt và hữu ích nhất của dữ liệu NSM
- ❑ Có một số điểm mạnh duy nhất có thể cung cấp giá trị đáng kể cho các chuyên gia phân tích NSM

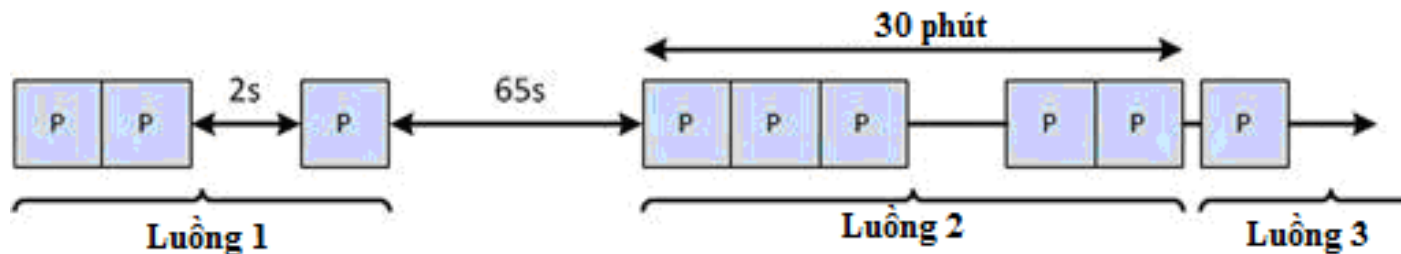


3.1. Luồng dữ liệu

- ❑ Là một bản ghi tổng hợp của các gói tin
- ❑ Ở đây tập trung chủ yếu vào công cụ SiLK
 - SiLK(System for Internet Level Knowledge):-SiLK là phần mềm mã nguồn mở của nhóm Network Situational Awareness(NetSA) tại CERT để truy vấn và phân tích dữ liệu NetFlow.
 - Bộ SiLK cho phép truy vấn nhanh chóng và hiệu quả lưu lượng mạng rất lớn theo thứ tự để xác định các hiện tượng tổng hợp phức tạp hoặc trích xuất các sự kiện riêng lẻ
 - SiLK thực sự là một cơ sở dữ liệu tại dòng lệnh. Mỗi công cụ thực hiện một truy vấn cụ thể, thao tác hoặc tổng hợp dữ liệu và các lệnh được kết nối với nhau để tạo ra các kết quả.

3.1. Luồng dữ liệu

- ❑ Là một bản ghi tổng hợp của các gói tin
- ❑ Ở đây tập trung chủ yếu vào công cụ SiLK
 - Một luồng được xác định dựa trên 5 thuộc tính, tạo thành bộ 5 chuẩn, gồm: *địa chỉ IP nguồn, cổng nguồn, địa chỉ IP đích, cổng đích và giao thức vận chuyển*
 - Có ba điều kiện mà luồng dữ liệu có thể được kết thúc:
 - Tự hết thời gian
 - Hết thời gian chờ
 - Hết thời gian hoạt động



Kết thúc luồng chờ và luồng hoạt động

NetFlow

- ❑ Phát triển bởi Cisco vào năm 1990 và đã trải qua 9 phiên bản của NetFlow trong hơn 20 năm, NetFlow v5 và v9 là hai chuẩn NetFlow thông dụng nhất
- ❑ NetFlow v5 là giải pháp truy cập NetFlow tốt nhất vì hầu hết các thiết bị định tuyến hiện đại hỗ trợ NetFlow v5
 - NetFlow v5 cung cấp thông tin theo chuẩn bộ-5 cũng như tất cả các số liệu thống kê cần thiết để phân tích các gói tin
 - Không hỗ trợ giao thức IPv6
- ❑ NetFlow v9 có tất cả các tính năng của v5
 - Người quản trị có thể sử dụng NetFlow v9 để tạo ra luồng tương tự như luồng v5
 - NetFlow V9 hỗ trợ IPv6

IPFIX

- ❑ Nhiều điểm chung với NetFlow v9 vì nó được xây dựng dựa trên định dạng tương tự
- ❑ IPFIX là định dạng dựa trên mẫu, hướng bản ghi, và xuất dạng nhị phân
- ❑ Đơn vị cơ bản để truyền dữ liệu là thông điệp
- ❑ Sự khác biệt giữa NetFlow v9 và IPFIX là ở chức năng
- ❑ IPFIX được coi là khá linh hoạt

Các loại lưu lượng khác

- ❑ Một lựa chọn khác có thể thay thế cho NetFlow và IPFIX là sFlow
 - Lấy mẫu luồng để làm giảm tải cho CPU bằng cách chỉ dùng mẫu đại diện của dữ liệu trên liên kết
 - sFlow cũng được tích hợp vào các thiết bị và các giải pháp phần cứng
- ❑ Ngoài ra có Jflow được cung cấp bởi thiết bị Juniper; AppFlow được cung cấp bởi Citrix,...

3.2. Thu thập dữ liệu phiên

- ❑ Cần 2 thành phần là một bộ sinh luồng và một bộ thu thập dữ liệu
- ❑ Bộ sinh luồng là thành phần phần cứng hoặc phần mềm, có trách nhiệm tạo ra các luồng dữ liệu
 - Phân tích các dữ liệu khác, hoặc là thu thập dữ liệu mạng trực tiếp từ giao diện mạng
- ❑ Bộ thu thập luồng là phần mềm có nhiệm vụ nhận luồng dữ liệu từ bộ sinh luồng và lưu chúng lại theo định dạng có thể phục hồi lại được

Thu thập dữ liệu phiên

❑ Sinh luồng dữ liệu từ dữ liệu FPC trong khi đang thu thập FPC

- FPC hay bị lọc, hoặc có thể mất gói tin
- Mất dữ liệu luồng → Phương pháp này không được khuyến khích

❑ Thường bắt trực tiếp dữ liệu trên liên kết theo cùng cách mà dữ liệu FPC hoặc dữ liệu cảnh báo NIDS được tạo ra

- Thực hiện bằng phần mềm trên máy tính, hoặc thông qua một thiết bị mạng như bộ định tuyến
- 2 dạng: (1) theo thiết bị thì gọi là "sinh theo phần cứng", và (2) theo phần mềm thì gọi là "sinh theo phần mềm".

Thu thập dữ liệu phiên

❑ Sinh luồng dữ liệu theo phần cứng:

- Có thể tạo ra một số phiên bản của dữ liệu luồng bằng cách tận dụng phần cứng hiện có
- Bộ định tuyến có khả năng thu nhận luồng sẽ được cấu hình với địa chỉ mạng của bộ thu thập dữ liệu đích và luồng dữ liệu từ giao diện của bộ định tuyến sẽ được gửi tới đích đó.
- Hầu hết các thiết bị Cisco có khả năng tạo dữ liệu NetFlow

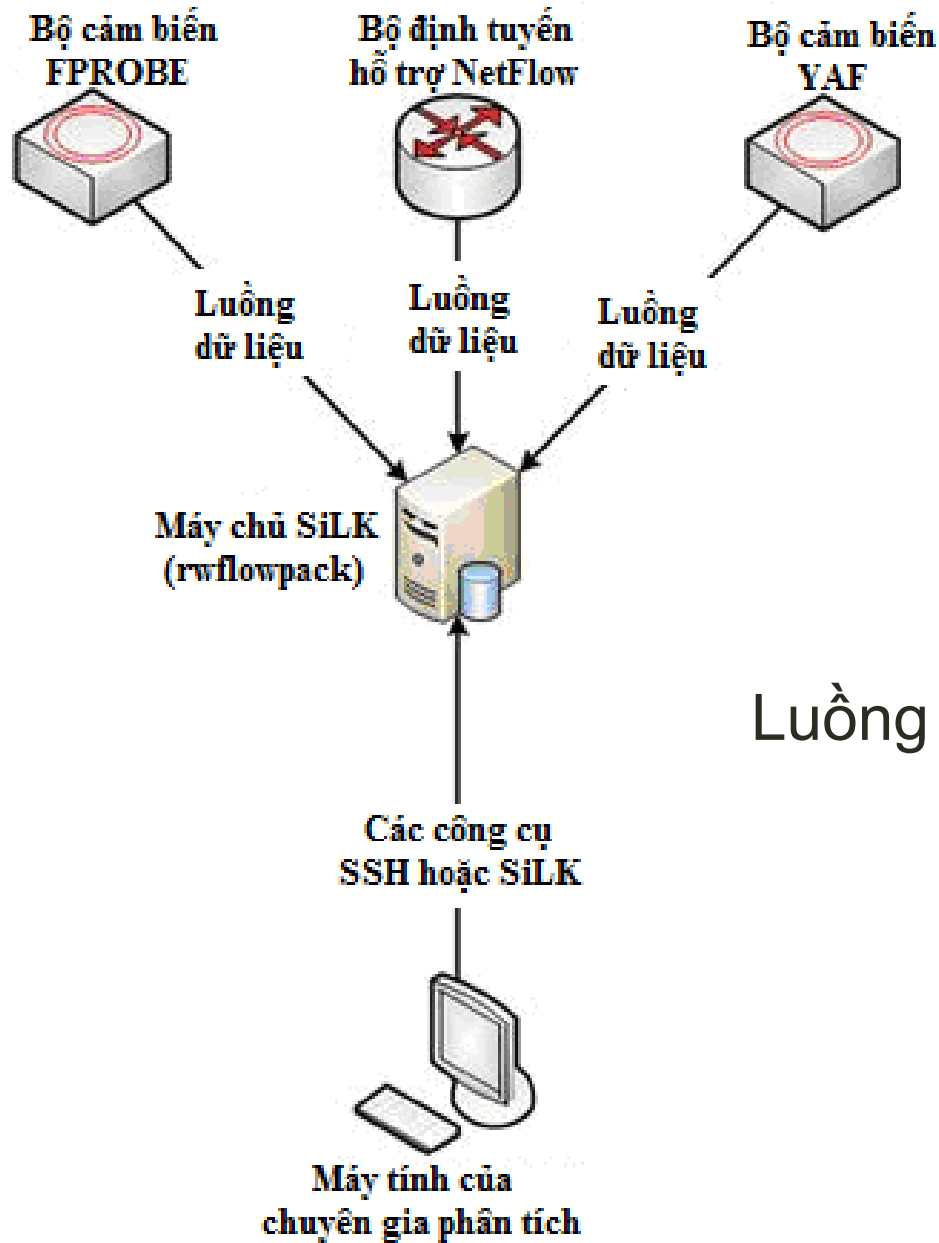
Thu thập dữ liệu phiên

□ Sinh luồng dữ liệu theo phần mềm:

- Đa số các cài đặt NSM đều dựa trên sinh theo phần mềm
- Có nhiều ưu điểm vượt trội, trong đó ưu điểm lớn nhất là sự linh hoạt khi triển khai phần mềm
- Sinh luồng bằng phần mềm liên quan đến:
 - Thực hiện một daemon trên cảm biến để thu thập và chuyển tiếp luồng dữ liệu dựa trên một cấu hình cụ thể
 - Luồng dữ liệu này được tạo ra từ dữ liệu đi qua các giao diện thu thập dữ liệu
- Ví dụ giải pháp phần mềm cho sinh luồng là Fprobe và YAF:
 - Fprobe là giải pháp sinh luồng NetFlow tối giản, có sẵn trong hầu hết các bản phân phối Linux hiện đại và có thể được cài đặt trên một cảm biến dễ dàng
 - YAF là một công cụ tạo luồng IPFIX để tạo ra các bản ghi IPFIX dùng cho SiLK, tạo ra bởi nhóm CERT NetSA

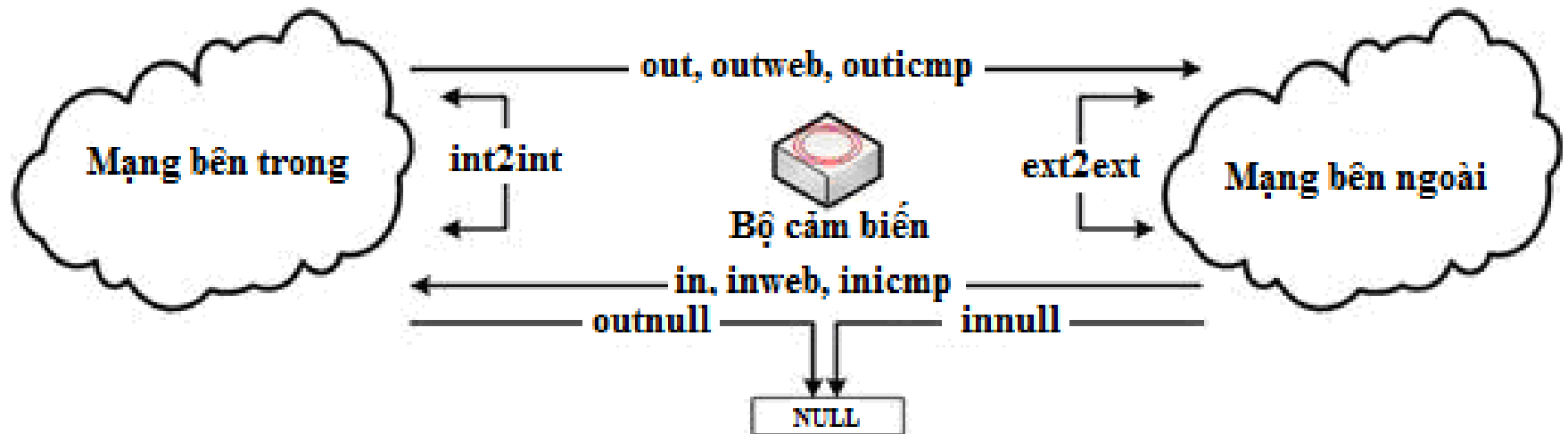
3.3. Thu thập và phân tích luồng dữ liệu với SiLK

- ❑ SiLK (System for Internet-Level Knowledge) - là một bộ thu thập luồng, có thể dễ dàng, nhanh chóng lưu trữ, truy cập, phân tích, và hiển thị dữ liệu luồng
- ❑ Có khả năng phân tích luồng nhanh chóng và hiệu quả, mà không lập kịch bản phức tạp, tiêu tốn quá nhiều CPU
- ❑ SiLK là một tập hợp bao gồm các ngôn ngữ C, Python, và Perl, hoạt động trong hầu hết các môi trường UNIX
- ❑ Hai thành phần: hệ thống đóng gói và bộ phân tích
 - Hệ thống đóng gói là phương pháp mà SiLK thu thập và lưu trữ dữ liệu luồng theo một định dạng gốc phù hợp
 - Bộ phân tích là một bộ công cụ thu thập dữ liệu dùng để lọc, hiển thị, sắp xếp, đếm,... dữ liệu; kết hợp theo dạng chuỗi liên tiếp với nhau giữa các công cụ



Luồng công việc của SiLK

Các loại luồng của SiLK



Các công cụ phân tích trong SiLK

- ❑ Hơn 55 công cụ phân tích trong cài đặt của SiLK
- ❑ Các công cụ phân tích làm việc như là một đơn vị liên kết chặt chẽ, với khả năng đưa dữ liệu từ một công cụ sang công cụ khác một cách liền mạch
- ❑ Công cụ được sử dụng nhiều nhất trong bộ công cụ phân tích là Rwfiler
 - Đưa các tệp dữ liệu nhị phân SiLK và các bộ lọc qua chúng để cung cấp những dữ liệu cụ thể mà chuyên gia phân tích yêu cầu
 - Xem thêm trên: <http://www.appliednsm.com/silk-on-security-onion/>

Các công cụ phân tích trong SiLK

- ❑ Rwstats tạo ra các dữ liệu thống kê dựa trên các trường giao thức chỉ định.
- ❑ Rwcount đếm gói tin và byte dữ liệu.
- ❑ Rwcut chọn lựa các trường dữ liệu còn rwuniq có thể giúp phân loại.
- ❑ Rwidquery có thể nhận đầu vào là file luật của Snort hay file cảnh báo, và giúp chỉ ra luồng nào từ dữ liệu đầu vào tương ứng với luật hoặc cảnh báo, từ đó tạo ra lời gọi rwfilter để tạo ra luồng phù hợp.
- ❑ Thư viện PySiLK cho phép gọi các lời gọi hàm API từ Python.

Lọc luồng dữ liệu với Rwfilter

□ Ví dụ là cần kiểm tra mức độ quá rối gây ra bởi một máy chủ vi phạm với một địa chỉ IP duy nhất

- sử dụng các lệnh `rwfilter` cùng với ít nhất một đầu vào, một đầu ra và một tùy chọn phân vùng
- tùy chọn địa chỉ IP nào đó (any-address option)
- tùy chọn ngày bắt đầu (start-date) và ngày kết thúc (end-date)
- type = tùy chọn all (muốn cả luồng đi vào (inbound) và luồng đi ra (outbound))
- pass = tùy chọn stdout (cho phép vượt qua đầu ra `rwcut` (thông qua biểu tượng sổ thẳng (|)) để có thể được hiển thị trong cửa sổ của thiết bị đầu cuối)

Lệnh `rwfilter` như sau:

```
rwfilter --any-address=1.2.3.4 --start-date=2013/06/22:11 --end-date=2013/06/22:16  
--type=all --pass=stdout | rwcut
```


Thu thập và phân tích luồng dữ liệu với Argus

- ❑ Công cụ giúp thực hiện thu thập và phân tích luồng dữ liệu trong các hệ thống NSM, nó là sản phẩm của CERT-CC
- ❑ Năm 1991, Argus chính thức được hỗ trợ bởi CERT
- ❑ Cung cấp một cái nhìn có hệ thống toàn diện về tất cả lưu lượng mạng trong thời gian thực
- ❑ Argus là một bộ phân tích luồng hai chiều, có nghĩa là sẽ theo dõi cả hai bên của cuộc hội thoại trên mạng và báo cáo số liệu cho cùng luồng dữ liệu
- ❑ Có công cụ phân tích thống kê và kỹ thuật phát hiện/cảnh báo riêng

Kiến trúc của Argus

- ❑ Gồm hai phần chính nằm trong hai gói
- ❑ Thành phần "Argus" chung:
 - Ghi lại lưu lượng dữ liệu thu được vào ổ đĩa qua một giao diện mạng của một thiết bị nào đó
 - Ghi dữ liệu vào ổ đĩa để truyền đi liên tục hoặc duy trì một kết nối đến máy chủ an toàn trung tâm để truyền dữ liệu đi liên tục
 - Nằm trên cảm biến và truyền dữ liệu về máy chủ log trung tâm
- ❑ Argus Client:
 - Thu thập dữ liệu từ các bộ sinh bên ngoài: đọc từ các tệp tin nhật ký, thư mục, hoặc một kết nối socket liên tục để phân tích thời gian thực
 - Là công cụ phân tích chính trong suốt thời gian sử dụng Argus

Thu thập dữ liệu cơ bản với Argus

- ❑ Công cụ **ra** cung cấp các phương tiện ban đầu cho việc lọc và duyệt dữ liệu thô được thu thập bởi Argus
- ❑ **ra** phải có khả năng truy cập vào một tập dữ liệu để hoạt động

Ví dụ lệnh dùng **ra** để xử lý chuẩn đầu vào và xuất chuẩn đầu ra vào một tệp tin :

```
cat /nsm/sensor_data/<interface>/argus/<file> | ra -w
```

```
--ip and host 67.205.2.30 | racluster -M rmon -m proto -s proto pkts  
bytes
```

3.5. Lưu trữ dữ liệu phiên

- ❑ Dữ liệu phiên là khá nhỏ, nhưng nếu không kiểm soát sẽ tăng nhiều lên
- ❑ Lượng dữ liệu lưu trữ phụ thuộc vào tầm quan trọng của dữ liệu đối với tổ chức và bảng thông mà tổ chức có
- ❑ Tuy nhiên, nên giữ các luồng dữ liệu về các phiên liên lạc
- ❑ Đối với SiLK, có một bảng tính dự phòng: <http://tools.netsa.cert.org/releases/SiLK-Provisioning-v3.3.xlsx>
- ❑ Quản lý các bản ghi nhật ký mạng, bằng cách thực hiện kiểm tra định kỳ tất cả dữ liệu và thực hiện xóa bỏ dữ liệu cũ (rollover) khi cần thiết hoặc theo một chu kỳ thời gian

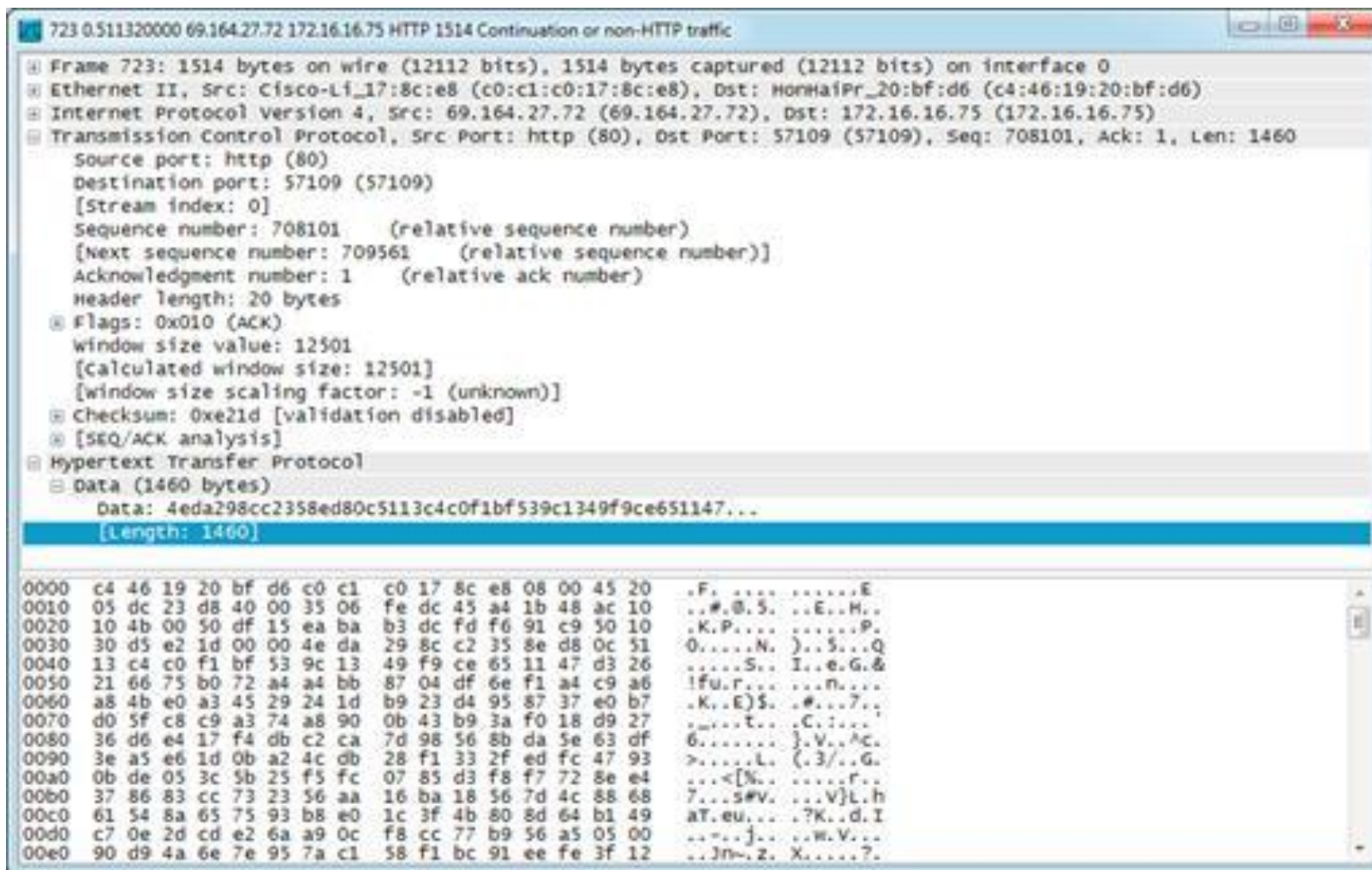
4. Dữ liệu bắt gói tin đầy đủ

- ❑ FPC cung cấp thông tin đầy đủ về tất cả các gói dữ liệu được truyền giữa hai điểm đầu cuối
- ❑ Xem xét một số công cụ bắt gói tin đầy đủ của dữ liệu PCAP như Netsniff-NG, Daemonlogger, và Dumpcap
- ❑ Lập kế hoạch lưu trữ và duy trì dữ liệu FPC, bao gồm cả vấn đề "cắt tỉa" bớt số lượng dữ liệu FPC được lưu trữ

4.1. Một số công cụ

- ❑ Định dạng phổ biến nhất của dữ liệu FPC là PCAP
- ❑ Libpcap là thư viện giúp tương tác với PCAP
- ❑ Dumpcap, Tcpdump, Wireshark,... Sử dụng libpcap

PCAP nhìn trong Wireshark



Dumpcap

❑ Là một công cụ đơn giản bắt gói tin từ một giao diện mạng và ghi chúng vào đĩa

❑ <https://www.winpcap.org/ntar/draft/PCAP-Dump文件格式.html>

❑ Khi đã cài đặt Wireshark (cùng với trình điều khiển libpcap đi kèm), có thể bắt các gói tin bằng cách gọi công cụ Dumpcap và chọn một giao diện mạng:

dumpcap -i eth1

❑ Hạn chế:

- Không phù hợp trong tình huống cần hiệu suất cao khi mức thông lượng cao, có thể dẫn đến các gói tin bị mất
- Sự đơn giản của công cụ này làm hạn chế tính linh hoạt của nó



Daemonlogger

- ❑ Là một ứng dụng ghi log gói tin được thiết kế đặc biệt để sử dụng trong môi trường NSM, thuộc chương trình phát triển IDS
- ❑ Sử dụng libpcap để bắt gói tin từ mạng, gồm có hai chế độ hoạt động
 - Chế độ hoạt động chính là để bắt các gói tin từ mạng và ghi chúng trực tiếp vào đĩa.
 - Chế độ còn lại cho phép bắt gói tin từ mạng và ghi vào một giao diện mạng thứ hai

daemonlogger -i eth1

- ❑ Daemonlogger thực hiện tốt hơn so với Dumpcap tại mức thông lượng cao, nó vẫn có thể bị hạn chế trong một số môi trường doanh nghiệp lớn hơn

Netsniff-NG



- ❑ Là một công cụ bắt gói hiệu suất cao được thiết kế bởi Daniel Borkmann
- ❑ Không dựa vào libpcap mà sử dụng cơ chế zero-copy
 - bắt gói tin đầy đủ trên các liên kết thông lượng cao
- ❑ Bắt gói với cơ chế RX_RING zero-copy, truyền gói tin với TX_RING
 - có khả năng đọc các gói tin từ một giao diện và chuyển hướng chúng vào một giao diện khác
 - khả năng lọc các gói tin bị bắt giữa các giao diện

Để bắt gói tin với Netsniff-NG, cần phải xác định một đầu vào và một đầu ra:

```
netsniff-ng -i eth1 -o data.pcap
```

Netsniff-NG



❑ Đầu ra tiến trình Netsniff-NG

```
sanders@kiowa:~$ sudo netsniff-ng -i eth1 -o data.pcap -s
Running! Hang up with ^C!

21376 packets incoming
21376 packets passed filter
0 packets failed filter (out of space)
0.0000% packet droprate
25 sec, 915574 usec in total
```

- ❑ Trong nhiều thử nghiệm, Netsniff-NG là một trong những công cụ FPC tốt nhất trong trường hợp liên kết có thông lượng cao.
- ❑ Netsniff-NG là công cụ FPC chuẩn mực, và được kèm mặc định trong bộ công cụ SO

4.2. Lựa chọn công cụ thu thập

- ❑ Dumpcap và Daemonlogger thường làm việc tốt trong hầu hết các tình huống có ít hoặc không mất gói tin
- ❑ Thông lượng càng lớn → càng dễ mất gói tin
- ❑ Cần công cụ như Netsniff-NG để hoạt động trong môi trường có tỷ lệ lưu lượng rất cao
- ❑ Lịch sử của công cụ thu thập FPC chủ yếu xoay quanh việc tạo ra dữ liệu "tốt nhất":
 - Không phải là các công cụ có thể xử lý được dữ liệu nhanh nhất
 - Công cụ làm mất mát gói tin ít nhất trên cảm biến
 - Có đủ các tính năng để đảm bảo dữ liệu được lưu trữ theo một định dạng chuẩn

4.3. Lập kế hoạch thu thập

- ❑ FPC có độ ưu tiên cao, do có thể tạo ra gần như tất cả các loại dữ liệu chính khác từ dữ liệu mạng
- ❑ Dữ liệu FPC sẽ luôn luôn là lớn nhất so với bất kỳ kiểu dữ liệu nào khác trên mỗi đơn vị thời gian
- ❑ Lưu ý về thông lượng, hoặc tỷ lệ trung bình của lưu lượng mạng qua giao diện đang theo dõi
 - Cần phải có một cổng giám sát đặc biệt trước khi triển khai cảm biến để đảm bảo rằng các cảm biến sẽ có đủ tài nguyên cần thiết hỗ trợ việc thu thập và phát hiện trên quy mô mong muốn

Lập kế hoạch thu thập

❑ Những cân nhắc khi lưu trữ:

- Xác định số lượng dữ liệu FPC cần lưu trữ là rất quan trọng
- Cần lựa chọn chiến lược duy trì theo thời gian hoặc kích thước, xác định mức hoạt động tối thiểu và lý tưởng
- Chiến lược dựa trên thời gian sẽ giữ lại dữ liệu PCAP với một khoảng thời gian ít nhất, ví dụ, 24 giờ
- Chiến lược dựa trên quy mô sẽ giữ lại một số tối thiểu dữ liệu PCAP, thường được phân bổ bởi khối lượng ổ cứng cụ thể, ví dụ, 10 TB dữ liệu PCAP
- Trên lý thuyết, việc đo thông lượng trung bình trên một giao diện có thể cho phép xác định cần bao nhiêu dữ liệu.
- Đồng thời cần cân nhắc các thời điểm peak

Lập kế hoạch thu thập

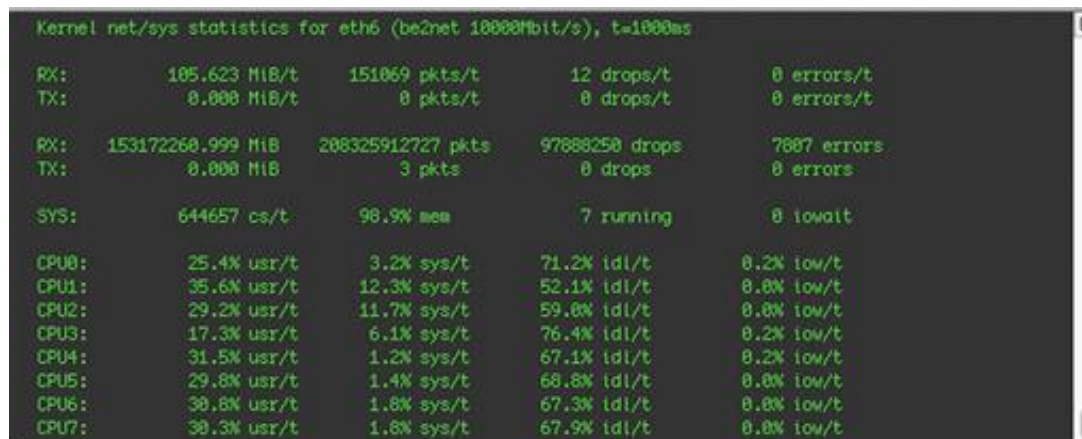
❑ Quản lý dữ liệu FPC dựa trên tổng số lượng dữ liệu được lưu trữ đơn giản hơn một chút và mang lại những tính năng an toàn vốn có.

- Cần xác định lượng tối đa không gian đĩa có thể cấp cho dữ liệu FPC. Một khi dữ liệu được lưu trữ đạt đến giới hạn này, dữ liệu FPC cũ nhất sẽ bị loại bỏ để nhường chỗ cho dữ liệu mới thu thập.
- Như đã thấy trước đây, Daemonlogger là một giải pháp FPC có tính năng này.

Lập kế hoạch thu thập

❑ Tính thông lượng giao diện cảm biến với Netsniff-NG và IFPPS:

- ifpps, là một phần của Netsniff-NG
- ifpps tạo ra số liệu thống kê chi tiết thông lượng hiện tại của giao diện được chọn, các dữ liệu khác liên quan đến CPU, đĩa I/O và thống kê hệ thống khác
- Hạn chế: không cung cấp chức năng để áp dụng một bộ lọc tới giao diện đang bắt gói tin → khó muốn giảm bớt FPC



```
Kernel net/sys statistics for eth6 (be2net 10000Mbit/s), t=1000ms
```

RX:	105.623 MiB/t	151069 pkts/t	12 drops/t	0 errors/t
TX:	0.000 MiB/t	0 pkts/t	0 drops/t	0 errors/t
RX:	153172260.999 MiB	208325912727 pkts	97888250 drops	7807 errors
TX:	0.000 MiB	3 pkts	0 drops	0 errors
SYS:	644657 cs/t	98.9% mem	7 running	0 iowait
CPU0:	25.4% usr/t	3.2% sys/t	71.2% idl/t	0.2% iow/t
CPU1:	35.6% usr/t	12.3% sys/t	52.1% idl/t	0.0% iow/t
CPU2:	29.2% usr/t	11.7% sys/t	59.0% idl/t	0.0% iow/t
CPU3:	17.3% usr/t	6.1% sys/t	76.4% idl/t	0.2% iow/t
CPU4:	31.5% usr/t	1.2% sys/t	67.1% idl/t	0.2% iow/t
CPU5:	29.8% usr/t	1.4% sys/t	68.8% idl/t	0.0% iow/t
CPU6:	30.8% usr/t	1.8% sys/t	67.3% idl/t	0.0% iow/t
CPU7:	30.3% usr/t	1.8% sys/t	67.9% idl/t	0.0% iow/t

Lập kế hoạch thu thập

- ❑ Tính thông lượng giao diện cảm biến với dữ liệu phiên:
 - Là cách linh hoạt nhất để tính toán, thống kê thông lượng
 - Ví dụ về tính thông lượng sử dụng công cụ `rwfilter`, `rwcount`, và `rwstats` trong SiLK

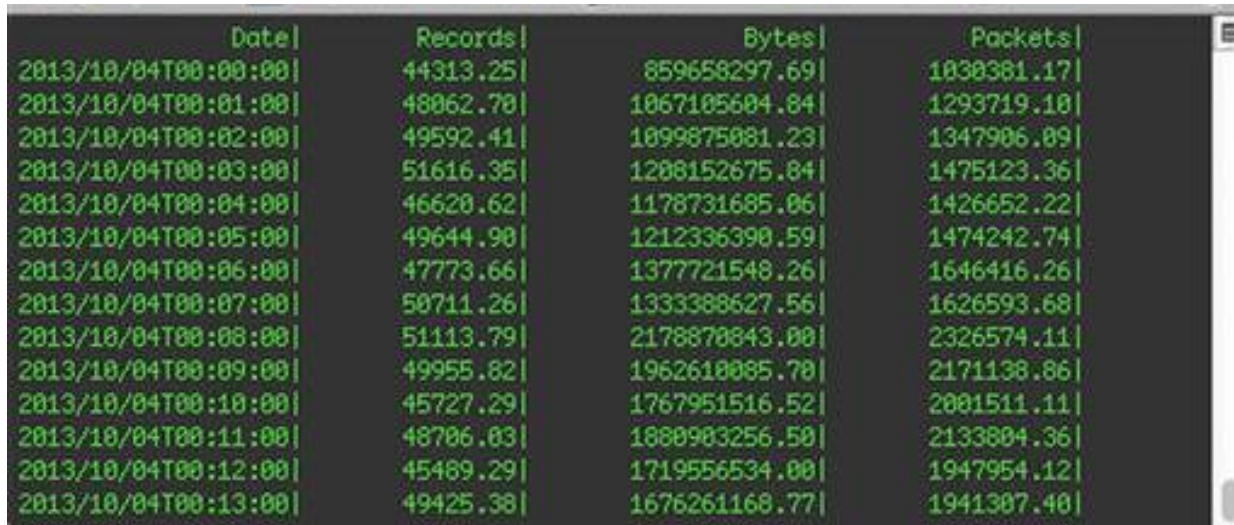
Bắt đầu, sử dụng `rwfilter` để chọn một khoảng thời gian cụ thể, ví dụ như 1 ngày, và lưu vào file `daily.rw`:

```
rwfilter --start-date = 2013/10/04 --proto = 0- --type = all --pass =  
daily.rw
```

Để xác định có bao nhiêu dữ liệu, dùng công cụ `rwcount`, với `bin-size` 1 phút:

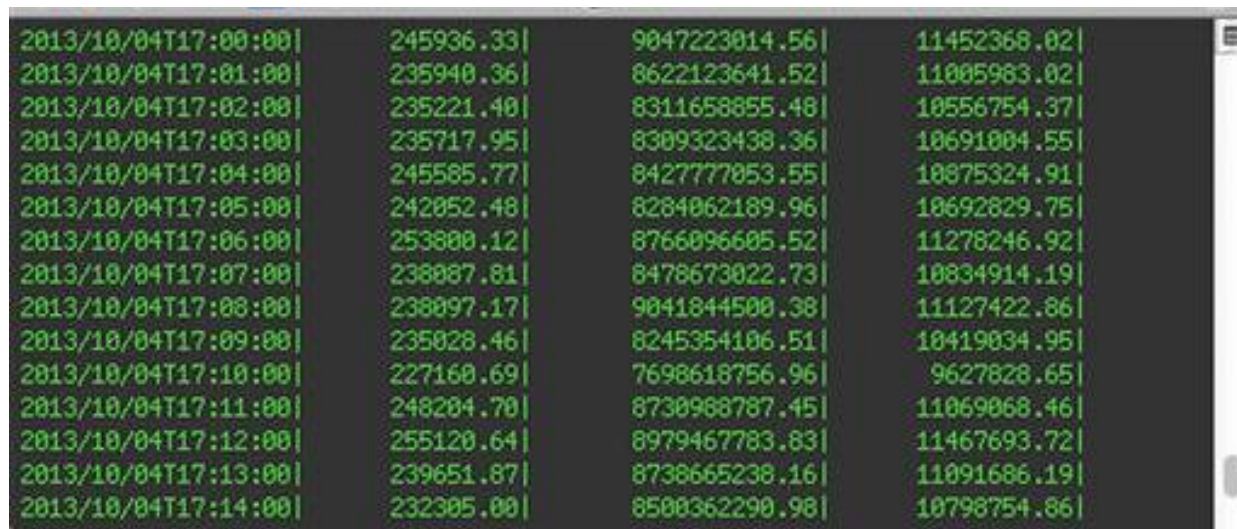
```
cat daily.rw | rwcount --bin-size = 60
```

- ❑ Thông lượng dữ liệu trong một phút với Rwcount ngoài lúc cao điểm là khoảng 1,5 GB lúc 0h:



Date	Records	Bytes	Packets
2013/10/04T00:00:00	44313.25	859658297.69	1030381.17
2013/10/04T00:01:00	48062.70	1067105604.84	1293719.10
2013/10/04T00:02:00	49592.41	1099875081.23	1347906.09
2013/10/04T00:03:00	51616.35	1208152675.84	1475123.36
2013/10/04T00:04:00	46620.62	1178731685.06	1426652.22
2013/10/04T00:05:00	49644.90	1212336390.59	1474242.74
2013/10/04T00:06:00	47773.66	1377721548.26	1646416.26
2013/10/04T00:07:00	50711.26	1333388627.56	1626593.68
2013/10/04T00:08:00	51113.79	2178870843.00	2326574.11
2013/10/04T00:09:00	49955.82	1962610085.70	2171138.86
2013/10/04T00:10:00	45727.29	1767951516.52	2001511.11
2013/10/04T00:11:00	48706.03	1880903256.50	2133804.36
2013/10/04T00:12:00	45489.29	1719556534.00	1947954.12
2013/10/04T00:13:00	49425.30	1676261168.77	1941307.40

- ❑ Lưu lượng truy cập cao đến 8-9 GB mỗi phút trong giờ cao điểm, 17:00h:

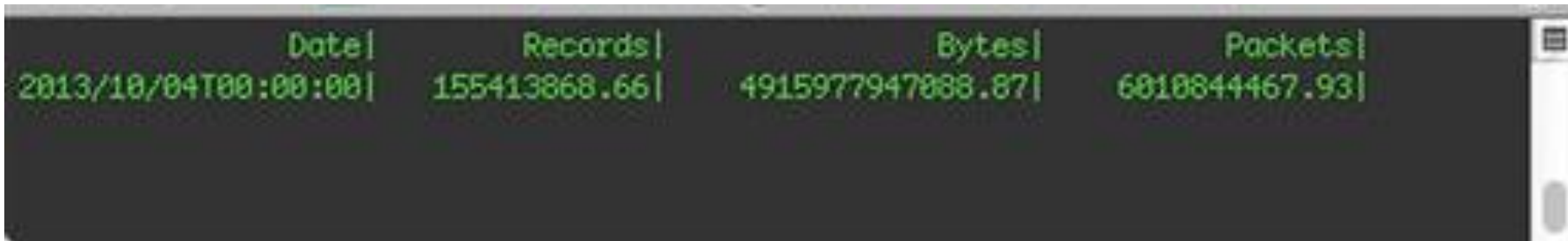


2013/10/04T17:00:00	245936.33	9047223014.56	11452368.02
2013/10/04T17:01:00	235940.36	8622123641.52	11005983.02
2013/10/04T17:02:00	235221.40	8311658855.48	10556754.37
2013/10/04T17:03:00	235717.95	8309323438.36	10691004.55
2013/10/04T17:04:00	245585.77	8427777053.55	10875324.91
2013/10/04T17:05:00	242052.48	8284062189.96	10692829.75
2013/10/04T17:06:00	253800.12	8766096605.52	11278246.92
2013/10/04T17:07:00	238007.81	8478673022.73	10834914.19
2013/10/04T17:08:00	238097.17	9041844500.38	11127422.86
2013/10/04T17:09:00	235028.46	8245354106.51	10419034.95
2013/10/04T17:10:00	227160.69	7698618756.96	9627828.65
2013/10/04T17:11:00	248204.70	8730988787.45	11069068.46
2013/10/04T17:12:00	255120.64	8979467783.83	11467693.72
2013/10/04T17:13:00	239651.87	8738665238.16	11091686.19
2013/10/04T17:14:00	232305.00	8500362290.98	10798754.86

❑ Để tính toán thông lượng trung bình trong ngày, có thể tăng kích thước bin trong lệnh `rwcount` để đếm tổng số dữ liệu cho một ngày, là 86.400 giây.

```
cat daily.rw | rwcount --bin-size = 86400
```

Khi đó tổng dữ liệu là **4578.36 GB**



Date	Records	Bytes	Packets
2013/10/04T00:00:00	155413868.66	4915977947088.87	6010844467.93

4.4. Giảm tải cho lưu trữ dữ liệu

❑ Dữ liệu quá lớn sẽ gây ảnh hưởng tới hệ thống lưu trữ. Có một số cách giảm tải dữ liệu.

❑ Loại bỏ dịch vụ:

- Loại bỏ lưu lượng được tạo ra bởi các dịch vụ riêng lẻ
- Xác định các dịch vụ thích hợp trong chiến lược này nhờ sử dụng `rwstats`

Sử dụng `rwstats` để xác định cổng chịu trách nhiệm về lưu lượng đi vào nhiều nhất trong mạng: xác định top 5 cổng lưu lượng theo dịch vụ. Rồi bỏ lưu lượng HTTPS đi → giảm ~21% lưu lượng mỗi ngày.

`cat daily.rw | rwstats --fields = sport --top --count = 5 --value = bytes`

```
INPUT: 155426371 Records for 65456 Bins and 4922925492806 Total Bytes
OUTPUT: Top 5 Bins by Bytes
sport|          Bytes|      %Bytes|   cumul_%|
 80|      2201459528713|  44.718522|  44.718522|
 443|      806016408749|  16.372712|  61.091234|
 445|      746044768087|  15.154500|  76.245735|
1935|      150008592677|   3.047143|  79.292878|
25873|      82746434776|   1.680839|  80.973717|
```

Giảm tải cho lưu trữ dữ liệu

❑ Loại bỏ lưu lượng host tới host:

- Là loại bỏ các liên lạc giữa các host cụ thể
- Sử dụng rstat để xác định các cặp IP có lưu lượng lớn nhất

```
INPUT: 105261826 Records for 2851556 Bins and 3897156767840 Total Bytes
OUTPUT: Top 5 Bins by Bytes
      sIP|          dIP|          Bytes|      %Bytes|      cumul_%|
141.239.24.49| 200.7.118.91| 740741493131| 19.007229| 19.007229|
141.239.194.40| 200.133.46.253| 165113761732| 4.236775| 23.244003|
141.239.108.35| 9.255.76.74| 29628207147| 0.760252| 24.004255|
247.76.249.129| 141.239.146.71| 22853505245| 0.586415| 24.590670|
200.7.214.240| 141.239.24.254| 22529483409| 0.578101| 25.168771|
```

Kiểm tra lưu lượng giữa các máy tính, lưu lượng cổng 22:

```
INPUT: 55426371 Records for 5456 Bins and 740741493131 Total Bytes
OUTPUT: Top 5 Bins by Bytes
sPort|          Bytes|      %Bytes|      cumul_%|
22| 740741493131| 100.000000| 100.000000|
```

Sử dụng chiến lược như trên có thể giảm số lượng dữ liệu được lưu trữ 40%.

4.5. Quản lý dữ liệu thu thập

❑ Quản lý dữ liệu FPC chủ yếu là thanh lọc dữ liệu cũ, với 2 chiến lược:

- Dựa trên thời gian

- Dễ dàng cho quản lý tự động
- Ví dụ, để tìm tệp tin cũ hơn 60 phút trong thư mục /data/pcap:

```
find /data/pcap -type f -mtime + 60
```

- Dựa trên kích thước

- Khó khăn hơn
- Xóa tệp tin PCAP lưu cũ nhất khi khối lượng lưu trữ vượt quá một tỷ lệ phần trăm nào đó đã sử dụng trên không gian đĩa
- Có thể sử dụng Daemonlogger để thực hiện

5. Dữ liệu kiểu chuỗi trong gói tin

5.1. Định nghĩa

❑ Dữ liệu kiểu chuỗi trong gói tin: Packet String Data – PSTR

- Là một lựa chọn dữ liệu FPC
- Có thể xuất hiện dưới dạng:
 - Ví dụ, tạo ra dữ liệu giao thức tầng ứng dụng

```
-----
09/22/13 23:33:01 - 10.10.10.3 -> 67.205.2.30
GET / HTTP/1.1.
User-Agent: Wget/1.13.4 (linux-gnu).
Accept: */*.
Host: www.appliednsm.com.
Connection: Keep-Alive.

09/22/13 23:33:02 -
HTTP/1.1 200 OK.
Date: Sun, 22 Sep 2013 23:33:01 GMT.
Server: Apache.
Accept-Ranges: bytes.
X-Mod-Pagespeed: 1.1.23.1-2169.
Cache-Control: max-age=0, no-cache.
Vary: Accept-Encoding, Cookie.
Content-Length: 71248.
Keep-Alive: timeout=2, max=100.
Connection: Keep-Alive.
Content-Type: text/html; charset=UTF-8.
```

Dữ liệu kiểu chuỗi trong gói tin

- ❑ Log dữ liệu kiểu PSTR chỉ ra một HTTP URL được yêu cầu:
 - Dữ liệu PSTR chỉ chứa các yêu cầu HTTP URL theo thời gian thực, có thể ứng dụng trọng cơ chế phát hiện danh tiếng tự động

```
sanders@osprey:~/ch6$ sudo justniffer -f packets.pcap -p "tcp port 80" -u -l "%request.timestamp - %source.ip -> %dest.ip - %request.header.host%request.url"
09/22/13 23:41:02 - 10.10.10.3 -> 67.205.2.30 - www.appliednsm.com/
09/22/13 23:41:17 - 10.10.10.3 -> 157.166.240.13 - www.cnn.com/
09/22/13 23:41:22 - 10.10.10.3 -> 23.66.230.66 - www.foxnews.com/
09/22/13 23:41:27 - 10.10.10.3 -> 199.181.132.250 - www.espn.com/
09/22/13 23:41:42 - 10.10.10.3 -> 67.205.2.30 - www.appliednsm.com/
09/22/13 23:41:47 - 10.10.10.3 -> 67.205.2.30 - www.appliednsm.com/contributors
09/22/13 23:41:57 - 10.10.10.3 -> 67.205.2.30 - www.appliednsm.com/about-the-book
```


Dữ liệu kiểu chuỗi trong gói tin

□ Tập trung
ứng dụng:

- Gồm một số
của gói tin
- Có thể đi k

```
-----
16:15:31.686876 IP 69.172.216.55.88 , 192.168.146.136.50505: tcp 1831
E.E.7.P.IqN.I.XAP.400
var adsafeVisParams . .
  mode : .jss.
  jsref : .http://imp.bid.ace.advertising.com/site.858222.size.160688.u.2.bnum.99795581.wkhr.168
.hr.16.hi.2.screens.5.swh.1440x900.tile.2.f.2.r.1.optn.1.fv.11.oolexp
.1.tags.1.dref.http.253A.252F.252Fwww.autoblog.com.252F.
  adsafeSrc : .http://pixel.adsafeprotected.com/rfv.st.19824.1214881.skeleton.js.
  adsafeSep : .
  requir : .
  requery : .
  debug : .false.
  allowEngagement : .true.
  trackHouse : .true.
  jsFeatures : .mousetrack.viewabilityready.consecutive.cochebust:8.forcecocoa:10.rattle:100.ex
ch.recordalternate:100.cocoapuffs.nextcocoa.usedtdomain:8.
  engagementDelay : .1-5-15.
  useAdTalk : .true.
  adTalkDtCall : .true.
  killPhrases : .
  asid : .8d5913c6-1d93-11e3-bcb0-8825904ea2d8.
  adWidth : .160.
  adHeight : .600.
  adHeight : .600.
  minimizeCalls : .false.
  exchList : .e1.:.nqzryq.e2.:.tbbtyrnqf.t.qbhoyrpyvpx.e3.:.ehovpbacebwrpg.e4.:.choznqvp.e5.:.b
crak.e6.:.nqoevgr.pbz.e7.:.tynz.pbz.e8.:.lvryqznantre.pbz.e9.:.yvvvg.e10.:.
-----
16:15:31.686885 IP 192.168.146.136.50505 , 69.172.216.55.88: tcp 0
E.(.K.-p.E.7.I.P.XAqN.zP.Duq.
```

5.2. Thu thập dữ liệu PSTR

□ Đầu tiên, cần xem xét mức độ của các dữ liệu PSTR muốn thu thập

- Lý tưởng là tập trung vào việc thu thập dữ liệu tăng ứng dụng cần thiết, càng nhiều từ các giao thức văn bản rõ càng tốt
- Vì có nhiều biến thể của dữ liệu PSTR có thể được thu thập nên không gian lưu trữ dữ liệu sẽ biến đổi rất lớn
- Nên sử dụng một số phương pháp thảo luận ở phần trước để xác định có bao nhiêu không gian lưu trữ để sử dụng cho dữ liệu PSTR
- Nên xem xét các khoảng thời gian dữ liệu được lưu lại
 - Việc lưu dữ liệu FPC thường được xem xét theo chu kỳ vài giờ hoặc vài ngày
 - Duy trì dữ liệu phiên cần xem xét theo chu kỳ quý hoặc năm
 - Dữ liệu PSTR nên theo chu kỳ tuần hoặc tháng để lấp đầy khoảng trống giữa FPC và dữ liệu phiên

□ Chú ý là sẽ có sự biến đổi rất lớn khi đánh giá các nhu cầu lưu trữ dữ liệu PSTR, phụ thuộc vào việc kinh doanh

Thu thập dữ liệu PSTR

- ❑ Thu thập dữ liệu PSTR từ mạng và thu thập từ dữ liệu FPC
- ❑ Tự động tạo ra dữ liệu PSTR hoặc thủ công
 - ✓ Các giải pháp thủ công tuy chậm trong xử lý dữ liệu nhưng linh hoạt
- ❑ Thu thập dữ liệu với URLSnarf
 - Thu thập dữ liệu yêu cầu HTTP một cách thụ động và lưu chúng dưới định dạng log chung CLF
 - Ví dụ: bắt lưu lượng truy cập bằng tcpdump và sau đó truyền qua URLsnarf với tùy chọn -p

```
sanders@osprey:~/ch6$ urlsnarf -p pockets.pcap
urlsnarf: using pockets.pcap [tcp port 80 or port 8080 or port 3126]
10.10.10.3 - - [22/Sep/2013:23:41:02 +0000] "GET http://www.opplinedns.com/ HTTP/1.1" - - "-" "curl/7.22.0 (x86_64-pc-linux-gnu) libcurl/7.22.0 OpenSSL/1.0.1 zlib/1.2.3.4 libidn/1.23 librtap/2.3"
10.10.10.3 - - [22/Sep/2013:23:41:17 +0000] "GET http://www.cnn.com/ HTTP/1.1" - - "-" "curl/7.22.0 (x86_64-pc-linux-gnu) libcurl/7.22.0 OpenSSL/1.0.1 zlib/1.2.3.4 libidn/1.23 librtap/2.3"
10.10.10.3 - - [22/Sep/2013:23:41:22 +0000] "GET http://www.foxnews.com/ HTTP/1.1" - - "-" "curl/7.22.0 (x86_64-pc-linux-gnu) libcurl/7.22.0 OpenSSL/1.0.1 zlib/1.2.3.4 libidn/1.23 librtap/2.3"
10.10.10.3 - - [22/Sep/2013:23:41:27 +0000] "GET http://www.espn.com/ HTTP/1.1" - - "-" "curl/7.22.0 (x86_64-pc-linux-gnu) libcurl/7.22.0 OpenSSL/1.0.1 zlib/1.2.3.4 libidn/1.23 librtap/2.3"
10.10.10.3 - - [22/Sep/2013:23:41:42 +0000] "GET http://www.opplinedns.com/ HTTP/1.1" - - "-" "curl/7.22.0 (x86_64-pc-linux-gnu) libcurl/7.22.0 OpenSSL/1.0.1 zlib/1.2.3.4 libidn/1.23 librtap/2.3"
10.10.10.3 - - [22/Sep/2013:23:41:47 +0000] "GET http://www.opplinedns.com/contributors HTTP/1.1" - - "-" "curl/7.22.0 (x86_64-pc-linux-gnu) libcurl/7.22.0 OpenSSL/1.0.1 zlib/1.2.3.4 libidn/1.23 librtap/2.3"
10.10.10.3 - - [22/Sep/2013:23:41:57 +0000] "GET http://www.opplinedns.com/about-the-book HTTP/1.1" - - "-" "curl/7.22.0 (x86_64-pc-linux-gnu) libcurl/7.22.0 OpenSSL/1.0.1 zlib/1.2.3.4 libidn/1.23 librtap/2.3"
```

Thu thập dữ liệu PSTR

❑ Thu thập dữ liệu với Httpry

- Là một công cụ bắt gói tin chuyên để hiển thị và ghi lại lưu lượng HTTP
- Có rất nhiều tùy chọn khi xử lý các dữ liệu đã thu thập, cho phép bắt và xuất thông tin về tiêu đề HTTP theo bất kỳ thứ tự nào

```
wonders@osprey:~/ch6$ httpry -r packets.pcap
httpry version 0.1.7 -- HTTP logging and information retrieval tool
Copyright (c) 2005-2012 Jason Bittel <jason.bittel@gmail.com>
2013-09-22 23:41:02 10.10.10.3 67.205.2.30 > GET www.appliednse.com / HTTP/1.1 -
2013-09-22 23:41:02 67.205.2.30 10.10.10.3 < - - HTTP/1.1 200 OK
2013-09-22 23:41:13 2600:1000:5003:0469:94be:80a1:c57e:6cf0 2607:f8b0:400a:sc83::67 > GET www.google.com HTTP/1.1 -
2013-09-22 23:41:14 2607:f8b0:400a:sc83::67 2600:1000:b003:0d69:94be:80a1:c57e:6cf0 < - - HTTP/1.1 200 OK
2013-09-22 23:41:17 10.10.10.3 157.166.248.13 > GET www.cnn.com / HTTP/1.1 -
2013-09-22 23:41:17 157.166.248.13 10.10.10.3 < - - HTTP/1.1 200 OK
2013-09-22 23:41:22 10.10.10.3 23.66.230.66 > GET www.foxnews.com / HTTP/1.1 -
2013-09-22 23:41:22 23.66.230.66 10.10.10.3 < - - HTTP/1.1 200 OK
2013-09-22 23:41:27 10.10.10.3 199.181.132.250 > GET www.espn.com / HTTP/1.1 -
2013-09-22 23:41:27 199.181.132.250 10.10.10.3 < - - HTTP/1.1 301 Moved Permanently
2013-09-22 23:41:42 10.10.10.3 67.205.2.30 > GET www.appliednse.com / HTTP/1.1 -
2013-09-22 23:41:42 67.205.2.30 10.10.10.3 < - - HTTP/1.1 200 OK
2013-09-22 23:41:47 10.10.10.3 67.205.2.30 > GET www.appliednse.com /contributors HTTP/1.1 -
2013-09-22 23:41:51 67.205.2.30 10.10.10.3 < - - HTTP/1.1 301 Moved Permanently
2013-09-22 23:41:57 10.10.10.3 67.205.2.30 > GET www.appliednse.com /about-the-book HTTP/1.1 -
2013-09-22 23:41:59 67.205.2.30 10.10.10.3 < - - HTTP/1.1 301 Moved Permanently
16 http packets parsed
```

5.3. Xem dữ liệu

❑ Logstash

- Là một công cụ phân tích log phổ biến dùng cho cả log đơn dòng và đa dòng theo nhiều định dạng, bao gồm định dạng phổ biến như syslog và các log có định dạng JSON, cũng như khả năng phân tích các log tùy chỉnh
- Miễn phí và theo mã nguồn mở, mạnh mẽ và tương đối dễ dàng thiết lập trong môi trường lớn
- Logstash phiên bản 1.2.1 có giao diện Kibana để xem log
- Ứng dụng Elasticsearch bên trong Logstash cho phép lập chỉ mục và tìm kiếm các dữ liệu nhận được
- Sử dụng GROK để kết hợp các mẫu văn bản và biểu thức thông thường nhằm so khớp với văn bản trong log thứ tự mong muốn
➔ phân tích dễ dàng hơn so với lúc sử dụng biểu thức thông thường

Xem dữ liệu

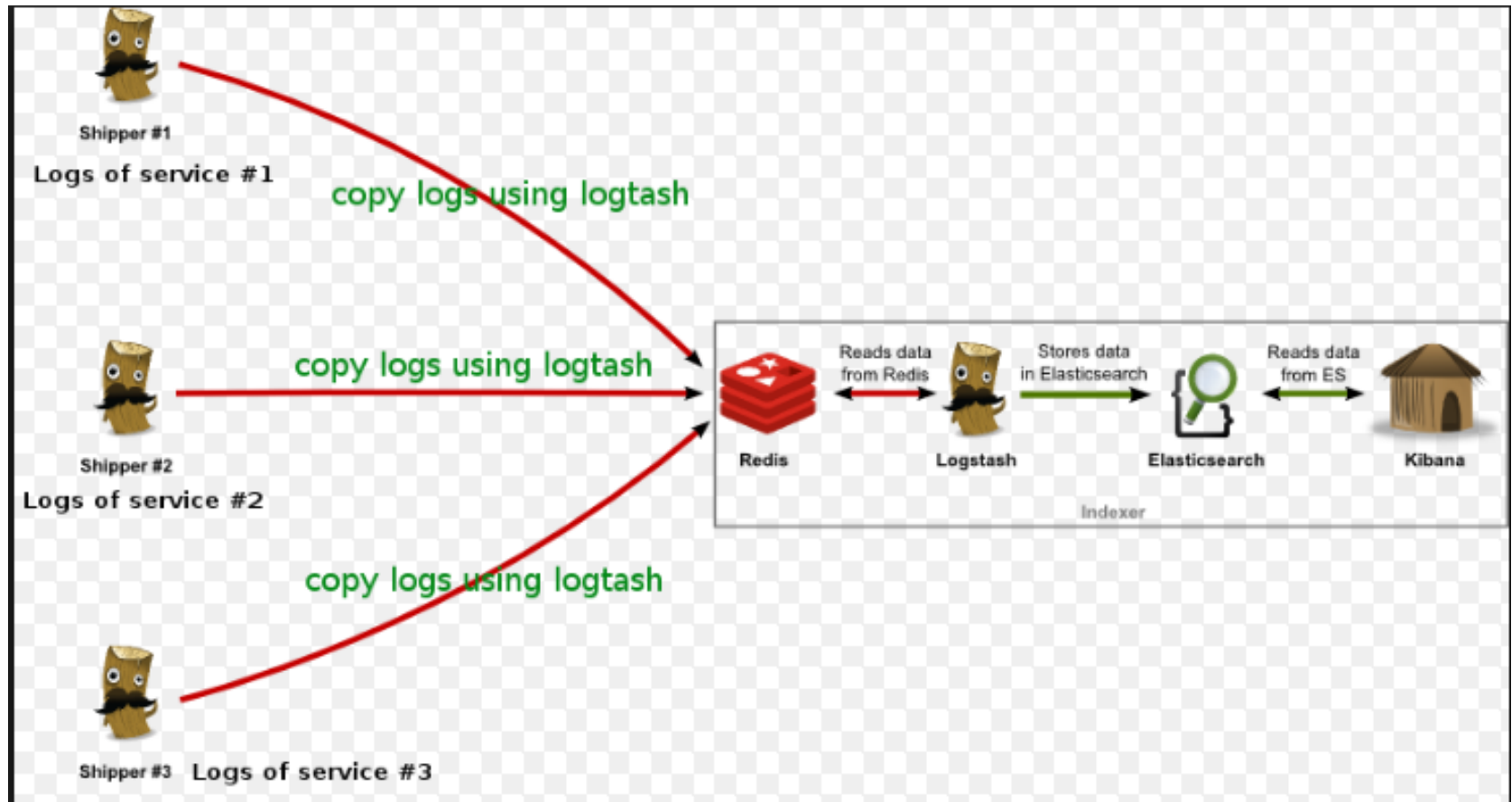
□ Nói thêm về Elasticsearch:

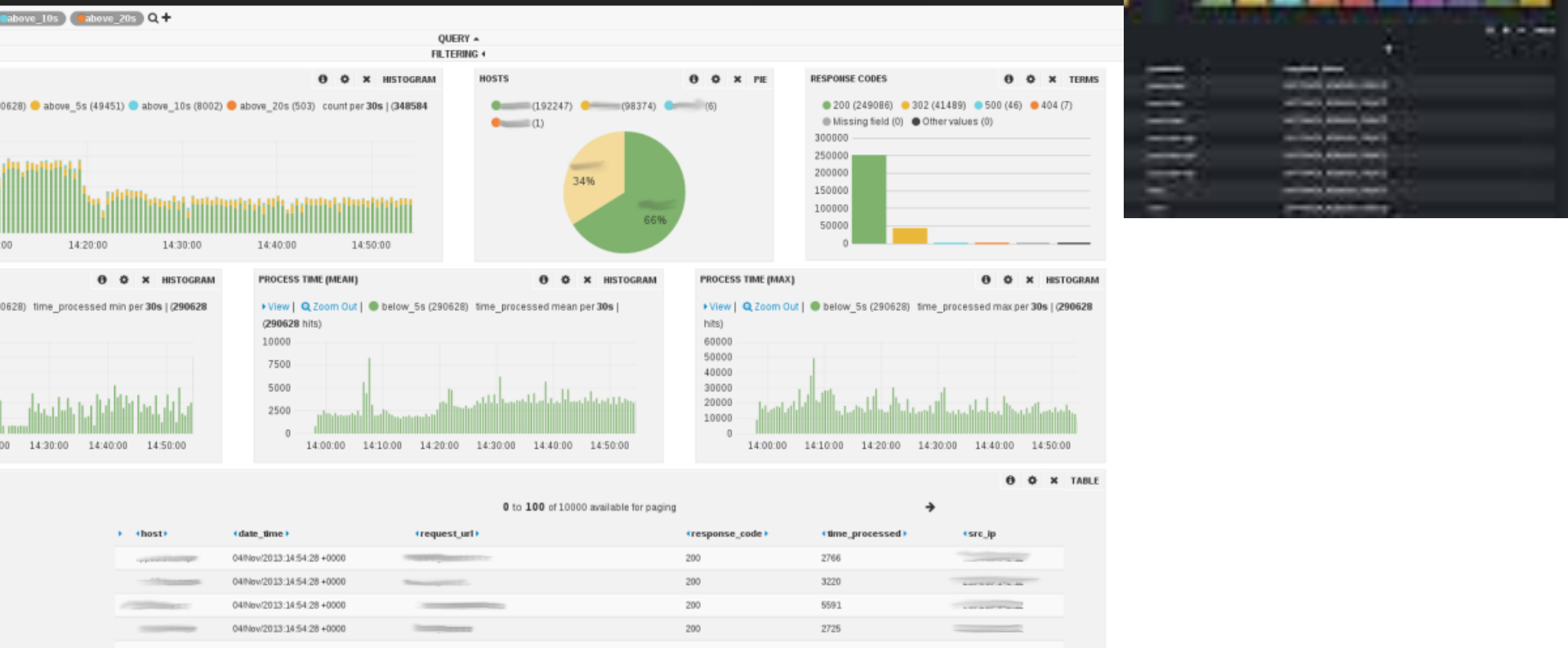
- Là một công cụ tìm kiếm cấp doanh nghiệp (enterprise-level search engine).
- Mục tiêu của nó là tạo ra một công cụ, nền tảng hay kỹ thuật tìm kiếm và phân tích trong thời gian thực
- Có thể áp dụng hay triển khai một cách dễ dàng vào nguồn dữ liệu (data sources) khác nhau: bao gồm các cơ sở dữ liệu nổi tiếng như MS SQL, PostgreSQL, MySQL,... văn bản (text), thư điện tử (email), pdf,... mọi thứ liên quan tới dữ liệu có văn bản



elasticsearch.

Xem dữ liệu





BÀI TẬP

- Tìm hiểu và so sánh định dạng file pcap và pcapng.
- Trình bày hiểu biết về các protocol hiển thị trong file pcapng: MDNS, NBNS, DHCP, ARP, LLMNR, SSDP, ICMPv6, QUIC, IGMP...