

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

-----  
**NGUYỄN NGỌC ĐIỆP**

**BÀI GIẢNG  
KỸ THUẬT THEO DÕI, GIÁM SÁT  
AN TOÀN MẠNG**

**HÀ NỘI, 2015**

## GIỚI THIỆU

Ngày nay, Internet đã phát triển vô cùng mạnh mẽ và có ảnh hưởng rất lớn đến hầu hết các hoạt động kinh tế, văn hóa và xã hội. Dường như không có sự khác biệt nhiều lắm giữa việc người dùng tham gia vào các hoạt động trong môi trường mạng Internet so với cuộc sống thực bên ngoài. Đi kèm đó, ngày càng xuất hiện nhiều hơn những nguy cơ liên quan đến việc bảo mật và bảo vệ tài sản của người dùng và các tổ chức, dẫn đến ngày càng có nhiều hơn những nhu cầu bảo vệ tài sản trên mạng. Trong hơn một thập kỷ qua, đi cùng với những nguy cơ và nhu cầu đó, khái niệm **giám sát an toàn mạng (network security monitoring, NSM)** đã ra đời và phát triển, thông qua chu trình giám sát an toàn mạng giúp các cá nhân và tổ chức nâng cao chất lượng bảo vệ toàn vẹn tài sản của họ.

Bài giảng “Kỹ thuật theo dõi, giám sát an toàn mạng” được biên soạn nhằm hỗ trợ cho sinh viên Đại học Công nghệ thông tin, đặc biệt là chuyên ngành An toàn thông tin, có được những kiến thức chuyên sâu về giám sát an toàn mạng, hoặc những người quan tâm đến lĩnh vực này có thể tham khảo. Bên cạnh những nội dung lý thuyết tập trung vào quy trình giám sát an toàn mạng, bài giảng còn đưa ra những ví dụ cụ thể trong thực tế và hướng dẫn các hoạt động thực hành, giúp người đọc nắm chắc được về mặt công nghệ và kỹ thuật liên quan. Nội dung trong bài giảng được tham khảo từ một số tài liệu chuyên ngành về NSM, đặc biệt là cuốn sách “Applied Network Security Monitoring” của các tác giả Chris Sanders và Jason Smith. Đây là một tài liệu được rất nhiều giáo viên và sinh viên sử dụng cho mục đích nghiên cứu và học tập.

Bài giảng được cấu trúc với bốn nội dung chính như sau:

Chương 1 giới thiệu các khái niệm về giám sát an toàn mạng và chu trình giám sát an toàn mạng, những thách thức đối với một hệ thống NSM, chuyên gia phân tích NSM, và giới thiệu bộ công cụ Security Onion, là bộ công cụ rất hữu dụng trong giảng dạy và học tập.

Các chương tiếp theo trong bài giảng sẽ trình bày cụ thể về các bước trong chu trình giám sát an toàn mạng, bao gồm thu thập dữ liệu, phát hiện xâm nhập và phân tích dữ liệu để có thể đưa ra những cảnh báo và các biện pháp đối phó với những nguy cơ an toàn mạng.

Chương 2 trình bày về các phương pháp thu thập dữ liệu, thiết bị thu thập dữ liệu là các cảm biến và các loại dữ liệu NSM: dữ liệu phiên, dữ liệu bắt gói tin đầy đủ và dữ liệu kiểu chuỗi trong gói tin.

Chương 3 thảo luận về các kỹ thuật phát hiện xâm nhập, dấu hiệu tấn công và chữ ký, và các phương pháp phát hiện xâm nhập: dựa trên danh tiếng, dựa trên chữ ký và dựa trên dữ liệu bắt thường thống kê, cùng với các công cụ thực hành cụ thể là Snort, Suricata và SiLK.

Cuối cùng, chương 4 trình bày về kỹ thuật phân tích gói tin với các công cụ như Tepdump và Wireshark. Chương này cũng trình bày về chu trình thu thập tri thức về nguy cơ bảo mật cho NSM và quá trình tạo tri thức về tài nguyên cần bảo vệ cũng như tri thức về nguy cơ bảo mật. Phần cuối chương trình bày về quy trình phân tích dữ liệu.

# MỤC LỤC

DANH MỤC CÁC TỪ VIẾT TẮT VÀ THUẬT NGỮ.....	v
<b>CHƯƠNG 1 GIỚI THIỆU VỀ GIÁM SÁT AN TOÀN MẠNG .....</b>	<b>1</b>
1.1 CÁC THUẬT NGỮ CHÍNH TRONG NSM .....	1
1.2 PHÁT HIỆN XÂM NHẬP .....	3
1.3 GIÁM SÁT AN TOÀN MẠNG (NSM).....	4
1.4 PHÒNG THỦ THEO LỖ HỒNG BẢO MẬT VÀ PHÒNG THỦ THEO NGUY CƠ.....	6
1.5 CHU TRÌNH GIÁM SÁT AN TOÀN MẠNG .....	7
1.6 THÁCH THỨC ĐỐI VỚI HỆ THỐNG NSM .....	8
1.7 CHUYÊN GIA PHÂN TÍCH CỦA HỆ THỐNG NSM.....	9
1.8 BỘ CÔNG CỤ SECURITY ONION .....	11
<b>CHƯƠNG 2 THU THẬP DỮ LIỆU .....</b>	<b>16</b>
2.1 PHƯƠNG PHÁP THU THẬP DỮ LIỆU .....	16
2.1.1 Giới thiệu phương pháp .....	16
2.1.2 Ví dụ tình huống: Cửa hàng bán lẻ .....	19
2.2 KIẾN TRÚC CẢM BIẾN.....	25
2.2.1 Các loại dữ liệu NSM .....	26
2.2.2 Các loại cảm biến .....	27
2.2.3 Phần cứng cảm biến.....	28
2.2.4 Hệ điều hành cảm biến .....	31
2.2.5 Vị trí đặt cảm biến .....	31
2.2.6 Bảo mật cho cảm biến .....	32
2.3 DỮ LIỆU PHIÊN .....	34
2.3.1 Luồng dữ liệu.....	34
2.3.2 Thu thập dữ liệu phiên.....	36
2.3.3 Thu thập và phân tích luồng dữ liệu với SiLK .....	37
2.3.4 Thu thập và phân tích luồng dữ liệu với Argus .....	40
2.3.5 Lưu trữ dữ liệu phiên.....	42
2.4 DỮ LIỆU BẮT GÓI TIN ĐẦY ĐỦ .....	42
2.4.1 Giới thiệu một số công cụ.....	43
2.4.2 Lựa chọn công cụ thu thập .....	47
2.4.3 Lập kế hoạch thu thập.....	47
2.4.4 Giảm tải cho lưu trữ dữ liệu .....	51
2.4.5 Quản lý dữ liệu thu thập .....	53
2.5 DỮ LIỆU KIỀU CHUỖI TRONG GÓI TIN .....	53
2.5.1 Định nghĩa .....	53
2.5.2 Thu thập dữ liệu.....	55
2.5.3 Xem dữ liệu .....	59
<b>CHƯƠNG 3 PHÁT HIỆN XÂM NHẬP .....</b>	<b>64</b>
3.1 CÁC KỸ THUẬT PHÁT HIỆN XÂM NHẬP, DẤU HIỆU TẤN CÔNG VÀ CHỮ KÝ .....	64
3.1.1 Kỹ thuật phát hiện xâm nhập .....	64
3.1.2 Dấu hiệu xâm nhập và chữ ký .....	65

3.1.3	Quản lý dấu hiệu tấn công và chữ ký .....	70
3.1.4	Các khung làm việc cho dấu hiệu tấn công và chữ ký.....	71
3.2	PHÁT HIỆN XÂM NHẬP DỰA TRÊN DANH TIẾNG .....	74
3.2.1	Danh sách danh tiếng công khai .....	74
3.2.2	Tự động phát hiện xâm nhập dựa trên danh tiếng .....	76
3.3	PHÁT HIỆN XÂM NHẬP DỰA TRÊN CHỮ KÝ VỚI SNORT VÀ SURICATA .....	80
3.3.1	Snort.....	80
3.3.2	Suricata .....	82
3.3.3	Thay đổi công cụ IDS trong Security Union .....	84
3.3.4	Khởi tạo Snort và Suricata cho việc phát hiện xâm nhập.....	85
3.3.5	Cấu hình Snort và Suricata .....	87
3.3.6	Các luật IDS.....	93
3.3.7	Xem các cảnh báo của Snort và Suricata.....	101
3.4	PHÁT HIỆN XÂM NHẬP DỰA TRÊN BẤT THƯỜNG VỚI DỮ LIỆU THỐNG KÊ...	103
3.4.1	Tạo danh sách thống kê với SiLK .....	103
3.4.2	Khám phá dịch vụ với SiLK.....	106
3.4.3	Tìm hiểu thêm về phát hiện xâm nhập dựa trên thống kê .....	110
3.4.4	Một số công cụ hiển thị thống kê .....	113
<b>CHƯƠNG 4</b>	<b>PHÂN TÍCH DỮ LIỆU.....</b>	<b>117</b>
4.1.	PHÂN TÍCH GÓI TIN .....	117
4.1.1	Xâm nhập vào gói tin.....	117
4.1.2	Một số khái niệm toán học liên quan.....	118
4.1.3	Phân tích chi tiết gói tin.....	121
4.1.4	Phân tích NSM với Tcpdump .....	125
4.1.5	Phân tích NSM với Wireshark.....	128
4.1.6	Lọc gói tin.....	135
4.2.	TRÌ THỨC VỀ NGUY CƠ BẢO MẬT VÀ TÀI NGUYÊN CẦN BẢO VỆ .....	137
4.2.1	Chu trình thu thập tri thức về nguy cơ bảo mật cho NSM .....	138
4.2.2	Tạo tri thức về tài nguyên cần bảo vệ.....	141
4.2.3	Tạo tri thức về nguy cơ bảo mật .....	147
4.3.	QUY TRÌNH PHÂN TÍCH .....	152
4.3.1	Các phương pháp phân tích .....	152
4.3.2	Các phương pháp quy chuẩn thực tiễn tốt nhất cho phân tích.....	162
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>165</b>	

## DANH MỤC CÁC TỪ VIẾT TẮT VÀ THUẬT NGỮ

Từ viết tắt/Thuật ngữ	Giải thích
ACF	Applied Collection Framework
BPF	Berkeley Packet Filter
C&C	Command and Control
CERT	Community Emergency Response Team
CIDR	Classless Inter-Domain Routing
CIS	Center for Internet Security
Client	máy khách
CND	Computer Network Defense
DNS	Domain Name System
FPC	Full Packet Capture
HIDS	Host-based IDS
Host	máy tính/máy trạm
HTTP	Hypertext Transfer Protocol
HTTPS	HTTP over SSL / HTTP Secure
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IOC	Indicators of Compromise
JRE	Java Runtime Environment
JSON	JavaScript Object Notation
MDL	Malware Domain List
NAT	Network Address Translation
NIDS	Network-based IDS
NSM	Network Security Monitoring
OSINT	Open-source intelligence
PCAP	Packet Capture
PRADS	Passive Real-time Asset Detection System
PSTR	Packet String
Request/response	yêu cầu/phản hồi
SAN	Storage Area Networking
server	máy chủ
SIEM	Security Information and Event Management
SMTP	Simple Mail Transfer Protocol
SO	Security Onion
SPAN	Switched Port Analyzer
STIX	Structured Threat Information eXpression
TCP	Transmission Control Protocol
TI	Threat Intelligence
VLAN	Virtual Local Area Network
VPN	Virtual Private Network

# CHƯƠNG 1

## GIỚI THIỆU VỀ GIÁM SÁT AN TOÀN MẠNG

Chương này trình bày các vấn đề cơ bản về giám sát an toàn mạng (NSM), bao gồm một số nội dung sau: các thuật ngữ chính dùng trong NSM, phát hiện xâm nhập mạng, giám sát an toàn mạng, chương trình giám sát an toàn mạng; phân biệt các khái niệm phòng thủ theo lỗ hổng bảo mật và phòng thủ theo nguy cơ; thách thức của một hệ thống NSM; các chuyên gia phân tích trong lĩnh vực NSM; và cuối cùng sẽ giới thiệu về bộ công cụ Security Onion (SO), là một bộ công cụ rất hữu ích trong giảng dạy và học tập trong lĩnh vực NSM.

### 1.1 CÁC THUẬT NGỮ CHÍNH TRONG NSM

Internet là một kho dữ liệu khổng lồ, là nơi mà tất cả những ai tham gia vào đều có thể cung cấp, lưu trữ và khai thác những thông tin, dữ liệu sẵn có trên hệ thống. Cùng với đó sẽ là rất nhiều các mối nguy cơ, đe dọa mà mọi người sẽ phải đối mặt, vượt qua khái niệm vùng lãnh thổ địa lý đến cấp độ toàn cầu. Những người có hành vi xấu (muốn đánh cắp thông tin/dữ liệu, muốn gây hại đến người dùng khác, muốn phá hủy các hệ thống quan trọng trong các tổ chức,..., có thể được gọi chung là tội phạm Internet) có thể hoạt động theo tổ chức hoặc đơn lẻ. Vậy câu hỏi đặt ra là làm thế nào để có thể đưa ra các luật (hay các quy tắc) và ép buộc tất cả mọi người thực thi các luật này? Câu trả lời là vô cùng khó khăn. Với thực tế này, trong những năm vừa qua, nhiều cá nhân, tổ chức đã tập trung tâm vào việc bảo vệ máy tính và dữ liệu của họ khỏi những kẻ tội phạm mạng bằng nhiều cách khác nhau. Đặc biệt, có một cách hiệu quả nhất để thực hiện việc này là thực thi giám sát an toàn mạng (network security monitoring - NSM).

NSM bao gồm việc thu thập dữ liệu, phát hiện xâm nhập và phân tích dữ liệu an ninh mạng. NSM được phân loại theo các miền sau:

- *Bảo vệ*: Tập trung vào việc ngăn chặn xâm nhập và khai thác trái phép vào hệ thống. Các chức năng bao gồm đánh giá lỗ hổng, đánh giá điểm yếu, quản lý chống phần mềm độc hại, đào tạo nâng cao nhận thức của người dùng, và các nhiệm vụ đảm bảo thông tin chung khác.
- *Dò tìm (phát hiện)*: Tập trung vào việc phát hiện ra tấn công đang xảy ra hoặc đã xảy ra trước đây. Chức năng bao gồm giám sát an ninh mạng, nhạy cảm với việc tấn công và cảnh báo.
- *Đáp ứng*: Tập trung vào việc phản ứng lại sau khi có một tấn công đã xảy ra. Chức năng bao gồm ngăn chặn sự cố, phân tích dựa trên máy chủ và mạng, phân tích phần mềm độc hại và báo cáo sự cố.
- *Duy trì*: Tập trung vào việc quản lý con người, các tiến trình và công nghệ liên quan đến việc bảo vệ mạng máy tính (Computer Network Defense - CND). Điều này bao gồm hợp đồng, biên chế và đào tạo, phát triển và triển khai công nghệ, và quản lý các hệ thống hỗ trợ.

Các thuật ngữ quan trọng trong giám sát an toàn mạng bao gồm: tài sản, nguy cơ (đe dọa), lỗ hổng, khai thác, điểm yếu, bất thường, sự cố, phát hiện xâm nhập, giám sát an toàn mạng (NSM), thu thập dữ liệu, phân tích dữ liệu, hệ thống phát hiện xâm nhập, chuyên gia phân tích, kỹ năng, cảnh báo, kẻ thù, bộ công cụ Security Onion.

- *Tài sản* (đè cập đến những gì thuộc phạm vi mạng tin cậy của một tổ chức): là bất cứ thứ gì có giá trị trong tổ chức, bao gồm máy tính, máy chủ, thiết bị mạng,... Ngoài ra, tài sản còn bao gồm dữ liệu, con người, quy trình, sở hữu trí tuệ và danh tiếng của tổ chức.
- *Nguy cơ (đe dọa)*: là một bên có khả năng và ý định khai thác một lỗ hổng trong một tài sản. Các nguy cơ có thể được chia thành 2 loại, là có cấu trúc và không có cấu trúc.
  - *Nguy cơ có cấu trúc*: sử dụng chiến thuật và thủ tục hành chính, và đã xác định được rõ mục tiêu. Điều này thường bao gồm các tội phạm có tổ chức, các nhóm tin tặc, cơ quan tình báo của chính phủ và quân đội. Nguy cơ có cấu trúc luôn theo đuổi mục tiêu đã lựa chọn, có một lý do và mục đích cụ thể.
  - *Nguy cơ không có cấu trúc*: không có động cơ, kỹ năng, chiến lược, hoặc kinh nghiệm như một nguy cơ có cấu trúc. Các nguy cơ này thường là do các cá nhân hoặc nhóm có tổ chức lỏng lẻo và nhỏ. Nguy cơ không có cấu trúc thường theo đuổi các mục tiêu tùy vào cơ hội, được lựa chọn khi tài sản mạng hiện diện với những lỗ hổng dễ dàng bị làm tổn hại.

Dù phạm vi và bản chất của các nguy cơ là như thế nào, thì tất cả chúng đều có một điểm chung là muôn đánh cắp một cái gì đó từ hệ thống của tổ chức, như tiền bạc, tài sản trí tuệ, danh tiếng, hoặc đơn giản là thời gian.

- *Lỗ hổng*: là một phần mềm, phần cứng, hoặc một điểm yếu thủ tục mà có thể hỗ trợ kẻ tấn công đạt được quyền truy cập trái phép vào một tài sản mạng. Ví dụ như một hệ thống xác thực không đúng cách sẽ có thể cho phép kẻ tấn công đoán ra tên đăng nhập của người dùng. Chú ý là con người cũng có thể được coi là một lỗ hổng.
- *Khai thác*: là phương pháp tấn công một lỗ hổng. Ví dụ, trong trường hợp khai thác phần mềm, đoạn mã khai thác có thể chứa payload (tải) cho phép kẻ tấn công thực hiện một số hành động trên hệ thống từ xa (như sinh ra lệnh shell); trong một ứng dụng web, lỗ hổng trong cách xử lý đầu vào và đầu ra có thể cho phép kẻ tấn công khai thác ứng dụng với SQL injection.
- *Điểm yếu (rủi ro)*: là khả năng có một mối đe dọa nhằm khai thác một lỗ hổng. Việc xác định định lượng rủi ro là một điều khó khăn vì nó liên quan đến việc đặt một giá trị trên mạng và tài sản dữ liệu. Vì vậy, người ta thường thảo luận về những việc có thể làm tăng hoặc giảm mức độ rủi ro đối với một tài sản, hơn là việc tính toán định lượng rủi ro.
- *Bất thường*: Là một sự kiện quan sát được trong hệ thống hoặc mạng được coi là khác thường. Ví dụ bất thường có thể là một hệ thống bị sập, các gói tin bị thay đổi, thấy có một liên hệ không bình thường với một máy chủ lạ, hoặc một số lượng lớn dữ liệu được chuyển

đi trong một khoảng thời gian ngắn,... Bất thường tạo ra các cảnh báo bởi các công cụ phát hiện, như hệ thống phát hiện xâm nhập trái phép, hoặc các ứng dụng xem xét nhật ký (log).

- *Sự cố*: Khi một sự kiện được xem xét điều tra, nó có thể được phân loại như là một phần của một sự cố. Một sự cố là sự vi phạm hoặc nguy cơ sắp xảy ra có liên quan đến các chính sách bảo mật máy tính, các chính sách sử dụng chấp nhận hoặc các chính sách bảo mật chuẩn. Đơn giản hơn có thể nói, sự cố là một điều xấu đã xảy ra, hoặc đang diễn ra trên mạng của tổ chức. Ví dụ, có một tấn công vào thư mục gốc của một máy tính, cài đặt phần mềm độc hại đơn giản, tấn công từ chối dịch vụ, hoặc thực thi thành công mã độc từ một email (thư điện tử) giả mạo. Chú ý là một sự cố bao gồm một hoặc nhiều sự kiện, nhưng hầu hết các sự kiện sẽ không trực tiếp đại diện cho một sự cố.

## 1.2 PHÁT HIỆN XÂM NHẬP

Trước khi sử dụng thuật ngữ NSM, lĩnh vực phát hiện thường được mô tả đơn giản là *phát hiện xâm nhập* (Intrusion Detection). Mặc dù NSM đã xuất hiện được khoảng mười năm, nhưng các thuật ngữ này vẫn thường được sử dụng thay thế cho nhau. Đây không phải là các từ đồng nghĩa, mà đúng hơn, phát hiện xâm nhập là một thành phần của NSM hiện đại.

Việc phát hiện được xây dựng xung quanh mô hình cũ của phát hiện xâm nhập, thường có một vài đặc điểm riêng biệt:

- *Bảo vệ (phòng thủ) lỗ hổng bảo mật*. Mô hình phổ biến nhất của những kẻ tấn công mạng máy tính là đột nhập vào mạng bằng cách khai thác một lỗ hổng phần mềm. Vì mô hình này rất đơn giản và dễ dàng bị loại bỏ, nên đây là phần mà hầu hết các chương trình phát hiện xâm nhập sớm được xây dựng xung quanh. Hệ thống phát hiện xâm nhập (IDS) được triển khai với mục tiêu phát hiện việc khai thác các lỗ hổng.
- *Phát hiện trong tập dữ liệu quan trọng*. Phần lớn các nỗ lực đặt trên lĩnh vực này nằm trong phạm vi của việc phát hiện. Tuy nhiên, việc thu thập dữ liệu thường không tập trung vào mối liên hệ giữa chiến lược thu thập và mục tiêu phát hiện. Việc này sẽ dẫn đến tình trạng coi thu thập "quá nhiều dữ liệu luôn luôn tốt hơn là không đủ" và sẽ "bắt giữ tất cả mọi thứ và sắp xếp lại sau".
- *Phần lớn dựa trên chữ ký*. Việc khai thác một lỗ hổng phần mềm thường là một hành động tương đối tĩnh và có thể được phát triển khá dễ dàng thành một chữ ký IDS. Như vậy, phát hiện xâm nhập theo cách truyền thống dựa vào những hiểu biết về tất cả các lỗ hổng được biết đến và phát triển chữ ký cho các phát hiện của họ.
- *Có gắng phân tích tự động hoàn toàn*. Mô hình phát hiện xâm nhập dựa trên lỗ hổng tin tưởng rằng hầu hết các cảnh báo IDS tạo ra là đáng tin cậy. Do vậy, mô hình này thường ít dựa vào việc phân tích của con người, và cố gắng để tự động phân tích càng nhiều càng tốt sau khi phát hiện có xâm nhập. Tuy nhiên, với thực trạng an ninh mạng hiện tại, việc phát hiện xâm nhập theo cách truyền thống là rất kém hiệu quả. Lý do chính là vì sự thất bại của

phòng vệ dựa trên lỗ hổng. Khi một lỗ hổng đã được tập trung bảo vệ, thì một khi kẻ tấn công đã quyết tâm nhắm vào mục tiêu cụ thể, hắn sẽ tìm các lỗ hổng khác để khai thác.

### 1.3 GIÁM SÁT AN TOÀN MẠNG (NSM)

NSM xuất phát và được ủng hộ bởi những người/tổ chức có tư duy phòng thủ, ví dụ như trong quân đội, nơi mà các hoạt động có tầm quan trọng và dữ liệu cần có tính bảo mật cao. Một số các hoạt động như sau:

- *Phá hủy*: Làm tổn thương hệ thống hoặc thực thể rất nặng đến mức không thể thực hiện bất kỳ chức năng nào hoặc không thể khôi phục được về trạng thái hữu dụng mà không phải xây dựng lại hoàn toàn.
- *Phá vỡ*: Phá vỡ hoặc làm gián đoạn luồng thông tin.
- *Làm suy giảm*: Làm giảm ảnh hưởng hoặc tác động từ lệnh của đối thủ, giảm kiểm soát, hoặc làm suy giảm các hệ thống thông tin liên lạc, đồng thời nỗ lực thu thập thông tin, phương tiện. Từ đó, có thể làm suy giảm tinh thần của đơn vị, giảm giá trị của mục tiêu, hoặc làm giảm chất lượng trong các quyết định và hành động của kẻ thù.
- *Tù chối*: Nhằm ngăn chặn kẻ thù truy cập và sử dụng các thông tin, hệ thống và dịch vụ quan trọng.
- *Đánh lừa*: Làm cho một người tin rằng đó là không phải sự thật. Tìm kiếm để đánh lừa những người ra quyết định bằng cách điều khiển nhận thức của họ về thực tại.
- *Khai thác*: Nhằm truy nhập được đến lệnh của kẻ thù và điều khiển hệ thống thu thập thông tin hoặc đưa ra những thông tin sai lệch.
- *Ảnh hưởng*: Làm cho người khác thực hiện hành vi theo cách thuận lợi hơn.
- *Bảo vệ*: Có hành động bảo vệ chống lại gián điệp hoặc bắt giữ các thiết bị và thông tin nhạy cảm.
- *Phát hiện*: Khám phá hay xác định được sự tồn tại, hiện diện, hoặc thực tại của một sự xâm nhập vào hệ thống thông tin.
- *Khôi phục*: Đưa thông tin và hệ thống thông tin trở về trạng thái ban đầu.
- *Ứng phó*: Phản ứng nhanh với kẻ thù hoặc những kẻ tấn công, xâm nhập khác.

Nhiều mục tiêu trong số này liên kết với nhau. Phần lớn các hệ thống NSM được dành riêng để phát hiện trong một nỗ lực nhằm ứng phó tốt hơn. NSM được coi là mô hình mới cho lĩnh vực phát hiện và đã xây dựng được một tập các đặc tính khác biệt hoàn toàn so với phát hiện xâm nhập truyền thống.

- *Phòng chống đến cùng dù thất bại*. Một thực tế khó khăn nhất là việc chấp nhận cuối cùng tài sản có thể bị mất. Mặc dù tất cả mọi thứ từ phòng thủ đến các bước phản ứng chủ động đã được thực hiện, nhưng cuối cùng kẻ tấn công vẫn có thể tìm thấy một con đường đi

được vào hệ thống. Thực tế, khi tổ chức xây dựng được một hệ thống phòng thủ tốt, thì kẻ tấn công cũng sẽ xây dựng được phương pháp tấn công mạnh hơn. Ví dụ, khi doanh nghiệp triển khai tường lửa và đảm bảo là các máy chủ đã được vá lỗi đầy đủ, thì kẻ tấn công sẽ sử dụng các cuộc tấn công bằng kỹ thuật lừa đảo để có được chỗ đứng trên mạng hoặc sử dụng khai thác zero-day để đạt được quyền truy nhập vào thư mục gốc trong máy chủ đã vá lỗi.

Do vậy, khi đã chấp nhận là cuối cùng tài sản có thể bị tổn hại, thì các tổ chức sẽ thay đổi cách bảo vệ tài sản của họ. Thay vì chỉ dựa vào phòng thủ, các tổ chức cần tập trung thêm vào việc phát hiện và phản ứng. Để làm được điều này, khi có tấn công lớn xảy ra, tổ chức cần được đặt vào đúng vị trí để có phản ứng hiệu quả và ngăn chặn tổn thất.

- *Tập trung vào tập dữ liệu.* Khi tất cả các nguồn dữ liệu có sẵn được thu thập và đặt vào một kho lưu trữ tập trung, thì sẽ dẫn đến những triển khai quản lý vô cùng tốn kém. Không những vậy, việc này còn không cung cấp được giá trị thực của dữ liệu bởi vì các loại quyền của dữ liệu không có sẵn và các công cụ phát hiện không thể có quy mô phù hợp với số lượng dữ liệu rất lớn mà họ phải đối mặt.

Để có được bất kỳ phát hiện hoặc phân tích nào thì đều cần phải có dữ liệu. Với cùng một mức độ phát hiện, trong trường hợp ít dữ liệu hơn thì có thể tiết kiệm được chu kỳ CPU và việc thực hiện hiệu quả hơn. Hơn nữa, nếu chỉ cung cấp cho các chuyên gia phân tích những dữ liệu mà họ cần thì họ có thể đưa ra quyết định nhanh và an toàn hơn nhiều.

- *Tiến trình theo chu trình.* Mô hình phát hiện xâm nhập cũ là một tiến trình tuyến tính. Khi người dùng nhận được cảnh báo, họ sẽ xác nhận cảnh báo, việc đáp ứng có thể được thực hiện nếu cần, và sau đó kết thúc. Tiến trình tuyến tính này rất đơn giản và thiếu trách nhiệm. Việc đặt sự cố an ninh mạng trong tiến trình này sẽ không hỗ trợ cho bất kỳ mục đích bảo vệ mạng nào. Mặc dù một số tấn công có thể chỉ diễn ra trong vài giây, nhưng những kẻ tấn công chuyên nghiệp thường thực hiện chậm và có phương pháp, đôi khi cần phải mất vài tháng để tấn công một mục tiêu cụ thể.

Như vậy có thể thấy rằng, tiến trình phát hiện và ứng phó với xâm nhập cần phải có tính chu trình. Nghĩa là, tập dữ liệu cần được cung cấp cho việc phát hiện xâm nhập, phát hiện xâm nhập cần được cung cấp cho việc phân tích dữ liệu, và kết quả phân tích nên được cung cấp quay trở lại cho tập dữ liệu. Điều này cho phép hệ thống bảo vệ tài sản mạng xây dựng được trí thông minh qua thời gian, và do đó có thể được sử dụng để phục vụ tốt hơn trong việc bảo vệ mạng.

- *Phòng thủ theo nguy cơ.* Trong khi phòng thủ theo lỗ hổng tập trung vào “làm thế nào”, thì phòng thủ theo nguy cơ tập trung vào “ai” và “tại sao”. Cụ thể, cần phải tự hỏi rằng ai muốn tấn công vào hệ thống mạng của tổ chức, và tại sao họ lại thực hiện hành động này?

Phòng thủ theo nguy cơ là công việc khá khó khăn, do hai nguyên nhân: (1) tầm nhìn sâu rộng vào hệ thống mạng của tổ chức và (2) khả năng thu thập và phân tích thông tin tình báo liên quan đến mục đích và khả năng của kẻ tấn công.

## 1.4 PHÒNG THỦ THEO LỖ HỒNG BẢO MẬT VÀ PHÒNG THỦ THEO NGUY CƠ

Phòng thủ theo lỗ hồng bảo mật và phòng thủ theo nguy cơ là hai phương pháp bảo mật mạng khác nhau. Có thể tưởng tượng chúng ta có một cầu môn cần bảo vệ, và sử dụng một trong hai cách bảo vệ, hoặc là xây một bức tường gạch, hoặc là dùng một thủ môn để bảo vệ. Phương pháp thứ nhất tương đương với việc phòng thủ theo lỗ hồng bảo mật, và phương pháp thứ hai tương đương với phòng thủ theo nguy cơ trong bảo mật mạng.

Mới đầu, thường suy nghĩ là tường gạch có vẻ là lựa chọn tốt hơn cả, do tường gạch vững chắc có thể bảo vệ được khá nhiều mục tiêu, và những kẻ tấn công chỉ có thể đạt được một mục tiêu khi vượt qua nó. Nhưng theo thời gian, tường gạch có thể bị phá hỏng dần dần. Biện pháp khắc phục là có thể thay thế từng viên gạch hỏng, nhưng thay thế xong viên này thì viên khác có thể lại tiếp tục bị hỏng.

Trong khi với trường hợp dùng thủ môn để bảo vệ, thì thủ môn sẽ được học các kỹ năng phòng thủ. Có thể có lần thủ môn sẽ bị đánh bại bởi một kẻ tấn công, nhưng thủ môn sẽ tích lũy được kinh nghiệm sau lần bị đánh bại đó, và lần sau sẽ khó có kẻ tấn công nào khác thực hiện chiến thuật tương tự mà vượt qua được.

Sự khác biệt chính là bức tường gạch không bao giờ biết được và thay đổi chiến thuật bảo vệ, trong khi thủ môn thì có thể nhận biết được thói quen của kẻ tấn công, họ sẽ học, thích nghi, và phát triển kỹ thuật. Đây chính cũng chính là khác biệt chính giữa hai phương pháp phòng thủ theo lỗ hồng và phòng thủ theo nguy cơ. Bảng 1.1. trình bày so sánh giữa hai phương pháp này.

**Bảng 1.1. So sánh giữa phòng thủ theo lỗ hồng bảo mật và phòng thủ theo nguy cơ**

Phòng thủ theo lỗ hồng bảo mật	Phòng thủ theo nguy cơ
<ul style="list-style-type: none"><li>Dựa vào kỹ thuật phòng chống</li><li>Tập trung vào phát hiện xâm nhập</li><li>Giả thiết có thể biết được tất cả các nguy cơ</li><li>Phân tích mỗi tấn công trong ngữ cảnh đơn giản</li><li>Phụ thuộc nhiều vào phát hiện dựa trên chữ ký</li><li>Ít khả năng phát hiện ra các nguy cơ chưa biết</li><li>Tiến trình tuyến tính</li></ul>	<ul style="list-style-type: none"><li>Biết rằng việc phòng chống cuối cùng sẽ thất bại</li><li>Tập trung vào tập dữ liệu</li><li>Biết rằng các nguy cơ sẽ sử dụng các công cụ, chiến thuật và thủ tục khác nhau</li><li>Kết hợp thông minh từ mọi tấn công</li><li>Sử dụng toàn bộ dữ liệu nguồn</li><li>Rất có khả năng phát hiện ra các hoạt động tấn công ngoài những dấu hiệu đã biết</li><li>Tiến trình theo chu trình</li></ul>

## 1.5 CHU TRÌNH GIÁM SÁT AN TOÀN MẠNG

Chu trình NSM bao gồm ba giai đoạn: thu thập dữ liệu, phát hiện xâm nhập, và phân tích dữ liệu. Hình 1.1 trình bày về chu trình của NSM.



*Hình 1.1 Chu trình giám sát an toàn mạng*

### Bước 1: Thu thập dữ liệu

Chu trình NSM bắt đầu với bước quan trọng nhất là thu thập dữ liệu. Việc thu thập dữ liệu được thực hiện với sự kết hợp của cả phần cứng và phần mềm trong việc tạo, sắp xếp và lưu trữ dữ liệu cho việc phát hiện xâm nhập và phân tích dữ liệu trong hệ thống NSM. Thu thập dữ liệu là phần quan trọng nhất của chu trình NSM bởi vì các bước thực hiện ở đây sẽ định hình khả năng của một tổ chức trong việc phát hiện xâm nhập và phân tích dữ liệu hiệu quả.

Có một số loại dữ liệu của NSM và tương ứng sẽ có một số phương pháp có thể thu thập được các loại dữ liệu này. Các loại dữ liệu phổ biến nhất của NSM bao gồm dữ liệu nội dung đầy đủ, dữ liệu phiên, dữ liệu thống kê, dữ liệu kiểu chuỗi trong gói tin và dữ liệu cảnh báo. Tùy thuộc vào nhu cầu của tổ chức, kiến trúc mạng và nguồn tài nguyên sẵn có, các kiểu dữ liệu này có thể được sử dụng chủ yếu để phát hiện xâm nhập, phân tích, hoặc cho dùng cho cả hai.

Khởi đầu, thu thập dữ liệu có thể là một trong những phần cần nhiều lao động nhất trong chu trình NSM. Để thu thập dữ liệu có hiệu quả đòi hỏi có một sự nỗ lực từ lãnh đạo tổ chức, đội ngũ an ninh thông tin, các nhóm mạng và các nhóm quản trị hệ thống.

Thu thập dữ liệu bao gồm các nhiệm vụ như sau:

- Xác định các vị trí có nhiều điểm yếu tồn tại trong tổ chức
- Xác định các nguy cơ ảnh hưởng đến mục tiêu tổ chức
- Xác định nguồn dữ liệu có liên quan
- Tinh chế nguồn dữ liệu thu thập được
- Cấu hình cổng SPAN để thu thập dữ liệu gói tin
- Xây dựng lưu trữ SAN cho lưu giữ nhật ký

- Cấu hình phần cứng và phần mềm thu thập dữ liệu

### **Bước 2: Phát hiện xâm nhập**

Phát hiện xâm nhập là quá trình mà qua đó dữ liệu thu thập được kiểm tra và cảnh báo sẽ được tạo ra dựa trên các sự kiện quan sát được và dữ liệu thu thập không được như mong đợi. Điều này thường được thực hiện thông qua một số hình thức chữ ký, sự bất thường, hoặc phát hiện dựa trên thống kê. Kết quả là tạo ra các dữ liệu cảnh báo.

Phát hiện xâm nhập thường là một chức năng của phần mềm với một số gói phần mềm phổ biến như Snort IDS và Bro IDS của một hệ thống phát hiện xâm nhập mạng (NIDS), và OSSEC, AIDE hoặc McAfee HIPS của một hệ thống phát hiện xâm nhập máy chủ (HIDS). Một số ứng dụng như Quản lý sự kiện và thông tin an ninh (Security Information and Event Management - SIEM) sẽ sử dụng cả dữ liệu dựa trên mạng và dữ liệu dựa trên máy chủ để phát hiện xâm nhập dựa trên các sự kiện liên quan.

### **Bước 3: Phân tích dữ liệu**

Phân tích là giai đoạn cuối cùng của chu trình NSM, và được thực hiện khi một người diễn giải và xem xét dữ liệu cảnh báo. Điều này thường sẽ liên quan đến việc xem xét thu thập dữ liệu bổ sung từ các nguồn dữ liệu khác. Phân tích dữ liệu có thể được thực hiện với các nhiệm vụ sau:

- Phân tích gói tin
- Phân tích mạng
- Phân tích máy chủ
- Phân tích phần mềm độc hại

Phân tích dữ liệu là phần tốn thời gian nhất trong chu trình NSM. Tại thời điểm này một sự kiện có thể được chính thức nâng lên thành sự cố, và có thể bắt đầu với các biện pháp ứng phó.

Chu trình NSM kết thúc bằng các bài học kinh nghiệm trong việc phát hiện xâm nhập và phân tích dữ liệu cho bất kỳ sự bất thường nào và tiếp tục hình thành các chiến lược thu thập dữ liệu cho tổ chức.

## **1.6 THÁCH THỨC ĐỐI VỚI HỆ THỐNG NSM**

Sự ra đời của NSM và phòng thủ theo nguy cơ được coi là một bước phát triển lớn trong an toàn thông tin mạng, tuy nhiên đây vẫn còn là một lĩnh vực mới nên còn mang nhiều khó khăn, thách thức. Trong khi có một số nỗ lực được đưa ra nhằm chuẩn hóa thuật ngữ và phương pháp, thì vẫn có một sự chênh lệch lớn giữa việc viết ra và những gì đang thực sự được thực hiện.

Với một vấn đề an ninh mạng cụ thể, nếu có vài ba người nói chuyện với nhau, họ sẽ có thể sử dụng các thuật ngữ khác nhau. Đây là vấn đề hạn chế từ góc độ đào tạo. Với một người muốn thành công trong công việc liên quan đến an ninh mạng, họ phải có một mức độ về kiến thức cơ

bản trước khi bước vào thực tế. Kiến thức ở đây bao gồm lý thuyết chung, thực hành và các yêu cầu cụ thể về một vấn đề.

Vấn đề về kỹ năng thực hành để có được hiệu quả tốt trong giám sát an toàn mạng là một vấn đề khá khó khăn. Nguồn nhân lực về NSM không đủ đáp ứng yêu cầu về kinh nghiệm và kiến thức cần thiết. NSM là một công việc đòi hỏi kinh nghiệm được thực hiện ở mức cấp cao để có thể hướng dẫn nhân viên cơ sở. Tuy nhiên, hầu hết các nhân viên từ mức trung đến mức cao đều thường khó khăn trong việc duy trì công việc, và họ kết thúc trong vai trò tư vấn hoặc một vị trí quản lý.

Vấn đề cuối cùng cần nhắc đến như là một thách thức lớn cho sự phát triển của NSM là chi phí cần thiết để thiết lập và duy trì một chương trình NSM. Chi phí bao gồm phần cứng cần thiết để thu thập và phân tích lượng dữ liệu lớn được tạo ra từ các chức năng NSM, phần lớn chi phí nữa là cho lực lượng lao động cần thiết làm phân tích NSM, và chi phí để hỗ trợ cơ sở hạ tầng NSM cho các chuyên gia phân tích.

## 1.7 CHUYÊN GIA PHÂN TÍCH CỦA HỆ THỐNG NSM

Thành phần quan trọng nhất của một hệ thống NSM là chuyên gia phân tích. Chuyên gia là một cá nhân, người sẽ diễn giải dữ liệu cảnh báo, phân tích và xem xét xem những dữ liệu nào có liên quan đến nhau, và xác định xem sự kiện xảy ra có phải là thật hay không, hay cần có những phân tích và điều tra thêm. Tùy thuộc vào kích thước và cấu trúc của một tổ chức, chuyên gia cũng có thể tham gia vào quá trình ứng phó sự cố hoặc thực hiện các nhiệm vụ khác như phân tích máy tính hoặc phân tích phần mềm độc hại.

Máu chốt của tổ chức chính là chuyên gia phân tích, là người có thể tìm thấy những bất thường trong dữ liệu thu thập được. Chuyên gia phân tích sẽ phải thường xuyên được cập nhật các công cụ, các chiến thuật và thủ tục mới nhất mà đối phương có thể sử dụng. Sự thật là sự an toàn hệ thống mạng của một tổ chức phụ thuộc vào khả năng làm việc hiệu quả của các chuyên gia phân tích.

### Kiến thức, kỹ năng cơ bản quan trọng mà chuyên gia phân tích cần phải có

- Phòng thủ theo nguy cơ, NSM, và chu trình NSM
- Chỗng giao thức TCP/IP
- Các giao thức tầng ứng dụng
- Phân tích gói tin
- Kiến trúc Windows
- Kiến trúc Linux
- Phân tích dữ liệu cơ bản (BASH, Grep, SED, AWK,...)
- Cách sử dụng IDS (Snort, Suricata,...)

- Chỉ dẫn tấn công và hiệu chỉnh chữ ký IDS
- Mã nguồn mở
- Phương pháp chẩn đoán phân tích cơ bản
- Phân tích phần mềm mã độc cơ bản

### **Một số kiến thức, kỹ năng chuyên ngành của một chuyên gia phân tích**

- *Chiến thuật tấn công.* Tập trung vào thử nghiệm thâm nhập và đánh giá an ninh. Các chuyên gia phân tích về lĩnh vực này sẽ cố gắng để đạt được quyền truy cập vào hệ thống mạng theo cùng cách mà kẻ tấn công sẽ thực hiện. Những loại bài tập là rất quan trọng để xác định các điểm yếu trong hệ thống. Ngoài ra, các chuyên gia phân tích có kiến thức về chiến thuật tấn công thường dễ nhận ra hoạt động tấn công hơn khi thực hiện phân tích NSM. Kiến thức và kỹ năng cần cho chuyên gia chiến thuật tấn công bao gồm trinh sát mạng, khai thác dịch vụ và phần mềm, backdoors, sử dụng phần mềm độc hại, và các kỹ thuật lọc dữ liệu.
- *Chiến thuật phòng thủ.* Chuyên gia chiến thuật phòng thủ là những người rất xuất sắc trong phát hiện xâm nhập và phân tích dữ liệu. Công việc thường liên quan đến các phương pháp phân tích và các công cụ phát triển mới, bao gồm cả việc đánh giá các công cụ để xem xét việc sử dụng chúng trong chương trình NSM của tổ chức. Kiến thức và kỹ năng cần cho chuyên gia chiến thuật phòng thủ bao gồm kiến thức sâu về truyền thông mạng, kiến thức rộng về kỹ thuật và hoạt động của IDS, chữ ký IDS và phát hiện xâm nhập dựa trên thống kê.
- *Lập trình.* Viết mã là một khả năng quan trọng trong hầu hết các lĩnh vực của công nghệ thông tin, đặc biệt là trong bảo mật thông tin và NSM. Một chuyên gia phân tích thông thạo lập trình sẽ có thể tùy chỉnh phát triển phát hiện xâm nhập và các giải pháp phân tích cho một đội NSM. Ngoài ra, người này thường rất giỏi phân tích dữ liệu lớn. Chuyên gia lập trình cho các mục đích của NSM cần phải có một sự hiểu biết rất lớn về môi trường BASH Linux, từ đó họ sẽ thành thạo một trong các ngôn ngữ diễn giải như Python hoặc PERL, thành thạo một ngôn ngữ lập trình như lập trình Java hoặc lập trình web PHP, và cuối cùng, thành thạo một ngôn ngữ biên dịch như C hoặc C++.
- *Quản trị hệ thống.* Một kỹ năng quan trọng của chuyên gia phân tích là quản trị hệ thống, như trong trường hợp thu thập dữ liệu về cấu hình IDS và di chuyển dữ liệu theo đúng cách của các gói phần mềm phát hiện khác nhau. Chuyên gia phân tích cũng có thể thực hiện dựa trên các cảm biến và phát triển việc thu thập dữ liệu máy chủ thông minh. Kiến thức và kỹ năng cần có với một chuyên gia quản trị hệ thống bao gồm hai nền tảng Windows và Linux, cùng với sự hiểu biết tinh thông về dữ liệu và thu thập dữ liệu nhật ký.

- *Phân tích phần mềm độc hại.* Chuyên gia phân tích cần có khả năng phân tích phần mềm độc hại và có thể thực hiện việc phân tích ở mức cao. Các kỹ năng cần có bao gồm cả kỹ năng phân tích tĩnh và động.
- *Phân tích dựa trên máy tính.* Từ việc phân tích máy tính đã từng bị xâm nhập, chuyên gia phân tích có thể nâng cao được tiến trình thu thập dữ liệu trong tổ chức. Kiến thức này cũng có thể được dùng để đánh giá và cài đặt cơ chế phát hiện xâm nhập dựa trên máy tính mới nhằm tạo ra các chỉ dẫn mới về tấn công dựa trên việc phân tích các thành phần dựa trên máy tính. Các kỹ năng cần thiết bao gồm phân tích ổ đĩa cứng và hệ thống tệp tin, phân tích bộ nhớ và tạo ra đường thời gian xảy ra sự cố.

### **Phân loại chuyên gia phân tích**

Có thể sử dụng một cách phân loại dựa trên ba cấp độ về khả năng.

- *Chuyên gia phân tích cấp 1 (L1).* Có một số kỹ năng cơ bản đã được liệt kê ở trên, nhưng họ không có khả năng giải quyết vấn đề liên quan đến chuyên môn đặc biệt. L1 điển hình sẽ dành phần lớn thời gian của họ để xem xét các cảnh báo IDS và thực hiện phân tích dựa trên những phát hiện của họ. Yếu tố quan trọng nhất có thể đóng góp vào sự thành công của L1 là kinh nghiệm. Trong hầu hết các tổ chức, phần lớn các chuyên gia phân tích thuộc loại L1.
- *Chuyên gia phân tích cấp 2 (L2).* Có nền tảng vững chắc về phần lớn các kỹ năng cơ bản. Thông thường, chuyên gia phân tích cấp 2 đã chọn được ít nhất một lĩnh vực chuyên môn và bắt đầu dành nhiều thời gian xem xét và cố gắng nâng cao kỹ năng của họ trong lĩnh vực đó. L2 như là một người cố vấn cho L1, và bắt đầu xác định "thực hành là tốt nhất" trong phạm vi chương trình NSM của tổ chức. L2 sẽ có thể tham gia vào việc hỗ trợ hình thành các tiến trình phát hiện xâm nhập trong nhóm bằng cách tạo chữ ký dựa trên các sự kiện mạng khác hoặc nghiên cứu OSINT. Chuyên gia phân tích L2 cũng phát triển khả năng tìm kiếm thông qua các nguồn dữ liệu khác nhau bằng tay để cố gắng tìm các sự kiện tiềm tàng thay vì chỉ dựa vào các công cụ phát hiện tự động.
- *Chuyên viên phân tích cấp 3 (L3).* Là các chuyên gia phân tích cấp cao nhất trong một tổ chức. Họ có nền tảng vững chắc về tất cả các kỹ năng cơ bản và thông thạo ít nhất một lĩnh vực chuyên môn. Các chuyên gia phân tích L3 thường được giao nhiệm vụ tư vấn cho các chuyên gia phân tích khác, phát triển và hỗ trợ đào tạo cũng như cung cấp các hướng dẫn về những điều tra phức tạp. Họ chủ yếu chịu trách nhiệm trong việc hỗ trợ để phát triển và tăng cường khả năng thu thập dữ liệu và phân tích xâm nhập cho tổ chức, trong đó có thể bao gồm tạo và phát triển các công cụ mới, cũng như đánh giá các công cụ hiện có.

### **1.8 BỘ CÔNG CỤ SECURITY ONION**

Security Onion (SO) là một bản phân phối của Linux được thiết kế để phát hiện xâm nhập và NSM. Bộ công cụ này dựa trên Xubuntu 10.04, gồm Snort, Suticata, Sguil, Squert, Snorby, Bro, NetworkMiner, Xplico, và nhiều các công cụ an toàn khác. Đây là bộ công cụ rất hữu dụng trong

giảng dạy và học tập, ngoài ra Security Onion còn được sử dụng cho các văn phòng và mạng lưới cá nhân. Với việc cài đặt khá đơn giản (khoảng gần 15 phút), người dùng có thể có một hệ thống NSM với đầy đủ các tính năng thu thập dữ liệu, phát hiện xâm nhập và phân tích dữ liệu.

## Cài đặt ban đầu

Ngoài việc được cài đặt để kiểm tra cho một hệ thống mạng thực tế, với mục đích thực hành, Security Onion hoàn toàn có thể được cài vào một hệ thống máy ảo, như VMWare Player hoặc VirtualBox.

Khi đã thiết lập phần mềm ảo, có thể tải các tập tin Security Onion ISO mới nhất từ: <http://securityonion.blogspot.com/>. Trang này cũng chứa rất nhiều các tài nguyên hữu ích cho việc cài đặt và cấu hình các lĩnh vực khác nhau của Security Onion. Khi đã hoàn tất việc tải xuống, thực hiện theo các bước sau để có được Security Onion và chạy:

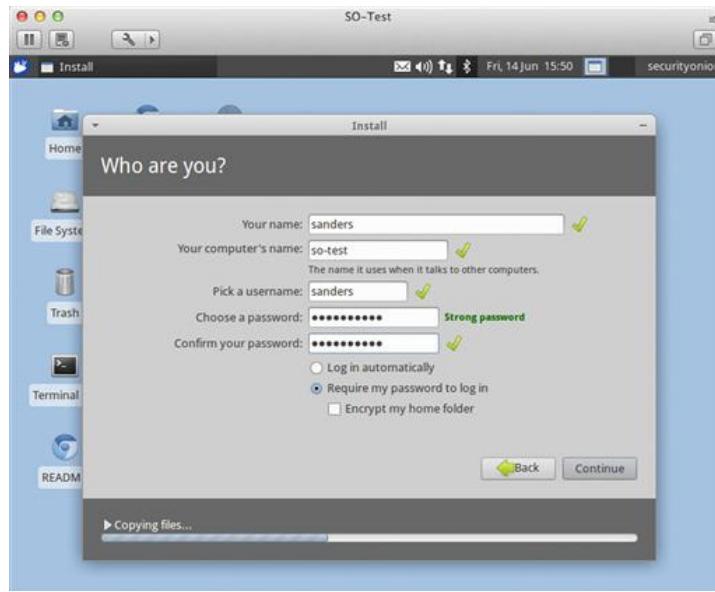
1. Tạo một máy ảo mới trên nền ảo đang sử dụng. Cần phải có ít nhất 1 GB RAM cho mỗi giao diện mạng giám sát, và tối thiểu là 2 GB cho toàn bộ. Chú ý đảm bảo các giao diện mạng được kết nối với các máy ảo tại thời điểm này.
2. Gán tập tin ISO đã tải về thành một ổ đĩa CD/DVD ảo trong phần mềm ảo.
3. Khi khởi động máy ảo, đặt ở chế độ khởi động đầy đủ trong hệ điều hành trực tiếp. Khi quá trình này hoàn thành, chọn biểu tượng "Install SecurityOnion" trên màn hình desktop để bắt đầu cài đặt hệ điều hành vào ổ ảo.
4. Thực hiện theo các hướng dẫn đã được giới thiệu bởi trình cài đặt Xubuntu. Trong khi cài đặt, một số mục sẽ được yêu cầu, bao gồm cách muôn cấu hình phân vùng đĩa, múi giờ, kết nối Internet, tên hệ thống, và tên người dùng và mật khẩu cho tài khoản (xem Hình 1.2). Các tùy chọn có thể được cấu hình theo ý muốn, tuy nhiên, điều quan trọng là không chọn tùy chọn mã hóa thư mục Home, và không kích hoạt tính năng tự động cập nhật. Các tùy chọn này được tắt theo mặc định. Khi đã hoàn tất cài đặt Xubuntu, hệ thống sẽ nhắc việc khởi động lại.

## Cập nhật Security Onion

Khi đã hoàn thành việc cài đặt hệ điều hành và khởi động lại máy, bước tiếp theo là đảm bảo Security Onion luôn được cập nhật. Ngay cả khi đã tải các tệp ISO, thì vẫn cần phải cập nhật các gói SO. Việc cập nhật có thể được bắt đầu bằng cách đặt lệnh sau:

```
sudo apt-get update && sudo apt-get dist-upgrade
```

Quá trình này có thể mất thời gian tùy thuộc vào số lượng các bản cập nhật kể từ khi ISO cuối cùng được tạo ra. Khi việc này hoàn thành, chúng ta sẽ có một bản cài đặt cập nhật của Security Onion.



**Hình 1.2 Câu hình thông tin người dùng trong cài đặt Security Onion**

### Cài đặt các dịch vụ NSM

Để có được các dịch vụ NSM và chạy trên Security Onion cần phải hoàn thành quá trình cài đặt tự động. Khi đã đăng nhập vào SO, thực hiện theo các bước sau:

1. Nhấn vào biểu tượng "Setup" trên desktop để bắt đầu quá trình cài đặt.
2. Sau khi nhập lại mật khẩu, sẽ được nhắc cấu hình /etc/network/interfaces. Chọn "Yes." Nếu có nhiều giao diện, hệ thống sẽ nhắc việc chọn một giao diện là giao diện quản lý, là giao diện sẽ sử dụng để truy cập vào hệ thống. Nếu chỉ có một giao diện duy nhất, giao diện sẽ được sử dụng để quản lý. Tiếp tục tiến trình này bằng cách chọn tùy chọn địa chỉ IP tĩnh và cấu hình địa chỉ IP của giao diện, subnet mask, default gateway, địa chỉ máy chủ DNS, và tên miền cục bộ. Sau khi các thông tin này được xác nhận, hệ thống sẽ được khởi động lại.  
Chú ý là nếu cấu hình giao diện bằng tay, thì nên để SO thực hiện bước này tự động vì nó sẽ có một số bước tối ưu hóa để đảm bảo các giao diện màn hình được cấu hình phù hợp nhằm bắt được tất cả lưu lượng mạng có thể.
3. Khởi tạo lại tiến trình cài đặt bằng cách nhấn vào biểu tượng "Setup" trên desktop.
4. Bỏ qua tiến trình cấu hình mạng vì việc này đã được hoàn thành.
5. Chọn "Quick Setup." (Có thể chọn cài đặt nâng cao, nhưng các cài đặt nhanh cũng đã là đủ cho các mục đích thực hành. Có thể khám phá những tùy chọn trong cài đặt nâng cao khi có thời gian.)
6. Nếu có nhiều giao diện, hệ thống sẽ nhắc việc chọn một giao diện giám sát. Có thể chọn một hoặc một vài giao diện phù hợp.

7. Nhập tên và mật khẩu người dùng sử dụng các dịch vụ NSM.
8. Khi được nhắc kích hoạt ELSA, chọn "Yes".
9. Cuối cùng, hệ thống sẽ nhắc việc xác nhận cấu hình của các cảm biến (Hình 1.3). Chọn "Yes, proceed with the changes!" để hướng dẫn SO thay đổi.



**Hình 1.3 Xác nhận thay đổi cài đặt**

Khi đã hoàn thành cài đặt, Security Onion sẽ cung cấp vị trí của các tệp tin cấu hình và nhật ký quan trọng. Khi gặp bất kỳ vấn đề gì với cài đặt hoặc thông báo về một dịch vụ không được bắt đầu một cách chính xác, cần phải kiểm tra nhật ký cài đặt tại /var/log/nsm/sosetup.log.

### Kiểm tra Security Onion

Cách nhanh nhất để đảm bảo rằng các dịch vụ NSM trên Security Onion đang chạy là buộc Snort tạo ra một cảnh báo từ một trong các luật của nó. Trước khi làm điều này, chúng ta cần cập nhật các tập luật được sử dụng bởi Snort. Có thể thực hiện điều này bằng cách đặt lệnh sudo rule-update. Lệnh này sẽ sử dụng tiện ích PulledPork để tải về các tập luật mới nhất từ Emerging Threats, tạo ra một sid-map mới (được sử dụng để ánh xạ các tên luật thành định danh duy nhất) và khởi động lại Snort để các luật mới được áp dụng. Một phần kết quả của lệnh này được thể hiện trong Hình 1.4.

Để kiểm tra chức năng của các dịch vụ NSM, chạy Snorby bằng cách chọn biểu tượng Snorby trên desktop. Hệ thống sẽ nhắc việc đăng nhập với địa chỉ e-mail và mật khẩu đã được cung cấp trong quá trình cài đặt. Tiếp theo, nhấp vào tab "Events" ở phía trên cùng của màn hình. Tại thời điểm này, có khả năng cửa sổ này đang bị bỏ trống.

Để tạo ra một cảnh báo Snort, mở một tab khác trong cửa sổ trình duyệt và chọn đến <http://www.testmyids.com>.

```

Processing /etc/nsm/pulledpork/enablesid.conf....
    Modified 0 rules
    Done
Processing /etc/nsm/pulledpork/dropsid.conf....
    Modified 0 rules
    Done
Processing /etc/nsm/pulledpork/disablesid.conf....
    Modified 0 rules
    Done
Modifying Sids....
    Done!
Setting Flowbit State....
    Enabled 29 flowbits
    Done
Writing /etc/nsm/rules/downloaded.rules....
    Done
Writing /etc/nsm/rules/so_rules.rules....
    Done
Generating sid-msg.map....
    Done
Writing /etc/nsm/rules/sid-msg.map....
    Done
Writing /var/log/sid_changes.log....
    Done
Rule Stats...
    New:-----0
    Deleted:----0
    Enabled Rules:----13968
    Dropped Rules:----0
    Disabled Rules:---3324
    Total Rules:----17292
    Done
Please review /var/log/sid_changes.log for additional details
Fly Piggy Fly!
Restarting Barnyard2.
Restarting: so-test-eth0
  * stopping: barnyard2-1 (spooler, unified2 format) [ OK ]
  * starting: barnyard2-1 (spooler, unified2 format) [ OK ]
Restarting IDS Engine.
Restarting: so-test-eth0
  * stopping: snort-1 (alert data) [ OK ]
  * starting: snort-1 (alert data) [ OK ]

```

*Hình 1.4 Đầu ra của lệnh Cập nhật luật*

Lúc này, nếu chuyển về tab với Snorby mở và làm mới lại trang Events (sự kiện), sẽ thấy một cảnh báo được đưa ra với các chữ ký sự kiện " GPL ATTACK\_RESPONSE id check returned root" (Hình 1.5). Nếu nhìn thấy cảnh báo này, thì coi như việc kiểm tra đã hoàn tất. Chúng ta đã thiết lập môi trường NSM đầu tiên thành công với Security Onion! Có thể kiểm tra các cảnh báo bằng cách nhấp vào nó và xem các đầu ra trong Snorby.

Sensor	Source IP	Destination IP	Event Signature	Timestamp	Sessions
so-test-eth0:1	217.160.51.31	172.16.16.111	GPL ATTACK_RESPONSE id check returned root	9:32 AM	1

*Hình 1.5 Kiểm tra cảnh báo Snort được trình bày trong Snorby*

## CHƯƠNG 2

### THU THẬP DỮ LIỆU

Chương này trình bày các vấn đề liên quan đến bước thu thập dữ liệu trong chu trình giám sát an toàn mạng, bao gồm một số nội dung sau: phương pháp thu thập dữ liệu, thiết bị thu thập dữ liệu là các cảm biến, và các loại dữ liệu NSM như dữ liệu phiên, dữ liệu bắt gói tin đầy đủ và dữ liệu kiểu chuỗi trong gói tin, cùng với các phương pháp thu thập cụ thể, kỹ thuật và công cụ quản lý, lưu trữ các loại dữ liệu này.

#### 2.1 PHƯƠNG PHÁP THU THẬP DỮ LIỆU

Việc thu thập dữ liệu là sự kết hợp của cả phần cứng và phần mềm, trong đó sẽ tạo và thu thập dữ liệu để phát hiện xâm nhập và phân tích dữ liệu NSM. Một chuyên gia phân tích phải là bậc thầy về dữ liệu đang sở hữu, nghĩa là họ phải hiểu rất rõ các nguồn dữ liệu họ có, nơi lấy được dữ liệu, cách thu thập dữ liệu, lý do thu thập và những gì có thể làm với dữ liệu đó. Chuyên gia phân tích giỏi có thể làm cho dữ liệu xấu trở nên hữu dụng và dữ liệu tốt thì sẽ hữu dụng hơn nữa.

Liên quan đến vấn đề này, hầu hết các tổ chức thuộc vào một trong ba loại sau:

- Các tổ chức không có cơ sở hạ tầng NSM, ở đó họ mới chỉ đang bắt đầu xác định nhu cầu thu thập dữ liệu của họ.
- Các tổ chức đã thực hiện phát hiện xâm nhập, nhưng chưa bao giờ có một cái nhìn sâu sắc về những dữ liệu mà họ thu thập được.
- Các tổ chức đã đầu tư rất nhiều thời gian trong việc xác định chiến lược thu thập dữ liệu, và liên tục phát triển chiến lược đó như là một phần của chu trình NSM.

##### 2.1.1 Giới thiệu phương pháp

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. Họ không có cách tiếp cận có cấu trúc để xác định các nguy cơ có thể đến với tổ chức, mà hầu như chỉ đơn giản là nắm bắt lấy bất kỳ dữ liệu tùy biến nào có sẵn để xây dựng chương trình. Với lượng dữ liệu lớn có thể dẫn đến tình trạng không đủ tài nguyên để lưu trữ dữ liệu, và do vậy hoặc là phải sử dụng lọc dữ liệu bằng nhân công (với rất nhiều sự kiện và có thể sàng lọc nhầm), hoặc các công cụ phát hiện xâm nhập và phân tích dữ liệu không thể hoạt động hiệu quả với số lượng lớn dữ liệu đang cần phân tích.

ACF (Applied Collection Framework) là khung làm việc được xây dựng để làm giảm sự phức tạp của việc thu thập dữ liệu (Hình 2.1). ACF bao gồm các bước nhằm giúp tổ chức đánh giá các nguồn dữ liệu cần tập trung trong quá trình thu thập dữ liệu. ACF gồm có bốn giai đoạn: Xác định nguy cơ, định lượng rủi ro (điểm yếu), xác định nguồn dữ liệu phù hợp và chọn lọc dữ liệu.



**Hình 2.1 ACF (Applied Collection Framework)**

### Giai đoạn 1: Xác định nguy cơ

Để có thể bảo mật các mối nguy cơ trọng tâm, cần phải có khả năng xác định những nguy cơ sẽ phải đối mặt. Thay vì chỉ xác định các nguy cơ chung, như các công ty đối thủ, script kiddie, nhóm tin tặc,..., cần phải xác định nguy cơ cụ thể vào mục tiêu của tổ chức. Câu hỏi đầu tiên cần đặt ra là “Tình trạng xấu nhất liên quan đến khả năng sống còn của tổ chức là gì?”, và đây là lý do mà chuyên gia an ninh thông tin thường phải 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 bảo mật (ví dụ, thông tin bị lộ ra bên ngoài, tài sản trí tuệ bị đánh cắp,...), tính toàn vẹn (ví dụ, sự sai lệch về dữ liệu) hoặc tính sẵn sàng (ví dụ, làm gián đoạn hoạt động) của tổ chức. Trong thực tế, hầu hết các tổ chức sẽ có một số nguy cơ mà họ đang quan tâm.

Khi các nguy cơ được xác định, chuyên gia an ninh thông tin có thể nghiên cứu sâu hơn vào những nguy cơ này để xác định 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, thì có thể nghiên cứu sâu hơn bằng những câu hỏi như sau:

- 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?

Tùy thuộc vào các câu trả lời, chuyên gia an ninh thông tin sẽ có thể bắt đầu xây dựng một bức tranh về những tài sản quan trọng nhất trong hệ thống để bảo vệ. 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, như máy chủ web (web server), máy chủ cơ sở dữ liệu (database server), máy chủ lưu trữ tệp tin (file server),...

### Giai đoạn 2: Định lượng rủi ro (điểm yếu)

Khi xác định được một danh sách các nguy cơ, thì cần phải xác định xem nguy cơ nào cần được ưu tiên. Điều này có thể thực hiện được bằng cách tính toán rủi ro gây ra bởi các nguy cơ tiềm ẩn, theo phương trình sau:

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

*Ảnh hưởng* là tác động của nguy cơ đến tổ chức, được đo trên thang điểm từ 1 đến 5, với ý nghĩa 1 là nguy cơ có tác động ít nhất và 5 là nguy cơ tác động lớn nhất. Xác định mức độ ảnh hưởng liên quan đến các vấn đề như tài chính, khả năng phục hồi dữ liệu bị mất và thời gian cần để làm cho tổ chức có thể hoạt động lại bình thường.

*Xác suất* là khả năng nguy cơ xuất hiện, cũng được đo trên thang điểm từ 1 đến 5, với ý nghĩa 1 là có một xác suất rất thấp nguy cơ xuất hiện, và 5 là có một xác suất rất cao nguy cơ xuất hiện. Việc xác định xác suất có liên quan đến việc thấy được mối đe dọa đến tài sản, hoặc mức độ mà mạng có thể bị tấn công, hoặc thậm chí là khả năng mà một người nào đó có thể truy cập vật lý tới tài sản,... Theo thời gian, xác suất khai thác một lỗ hổng sẽ tăng lên. Như vậy, xác suất sẽ chỉ thể hiện tại từng thời điểm và nó cần được xác định lại theo thời gian.

Kết quả là *mức độ rủi ro* mà nguy cơ gây ra đối với sự an toàn của mạng, ảnh hưởng tới tổ chức. Kết quả này được đo trên thang điểm từ 1 đến 25, và được chia thành 3 loại như sau:

- 0-9: Rủi ro thấp
- 10-16: Rủi ro trung bình
- 17-25: Rủi ro cao

Mặc dù ảnh hưởng và xác suất cung cấp khả năng định lượng cho các số liệu liên quan đến các nguy cơ, nhưng các con số này vẫn còn mang tính chủ quan. Điều này là do các con số này được tạo ra bởi ủy ban và nhóm các cá nhân tham gia vào bảng xếp hạng các nguy cơ. Một số tổ chức chọn bốn thứ ba để giúp định lượng các rủi ro và cũng đã thực hiện thành công.

### Giai đoạn 3: Xác định nguồn dữ liệu

Giai đoạn tiếp theo của ACF liên quan đến việc xác định nguồn dữ liệu chính được dùng trong việc phát hiện xâm nhập và phân tích NSM. Dù từ nguy cơ có hệ số rủi ro cao nhất, chúng ta cần 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 được cấu trúc của máy chủ này, vị trí của nó trên mạng, người có quyền truy cập và đường dẫn mà dữ liệu đi vào. Dựa vào những thông tin này, có thể kiểm tra cả hai nguồn dữ liệu dựa trên mạng và dựa trên máy chủ. Danh sách các loại nguồn dữ liệu như sau:

- 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 bắt 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

#### **Giai đoạn 4: Chọn lọc dữ liệu**

Giai đoạn cuối cùng của ACF chọn lọc được dữ liệu. Việc này liên quan đến các bước kỹ thuật chiêm sâu và cần phải xem xét tất cả các nguồn dữ liệu riêng để xác định giá trị của nó. Với 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. Do vậy, các tổ chức sẽ phải thực hiện phân tích chi phí/lợi ích của các nguồn dữ liệu. Chi phí có thể liên quan đến tài nguyên phần cứng, phần mềm, nhân công, việc tổ chức và lưu trữ dữ liệu,...

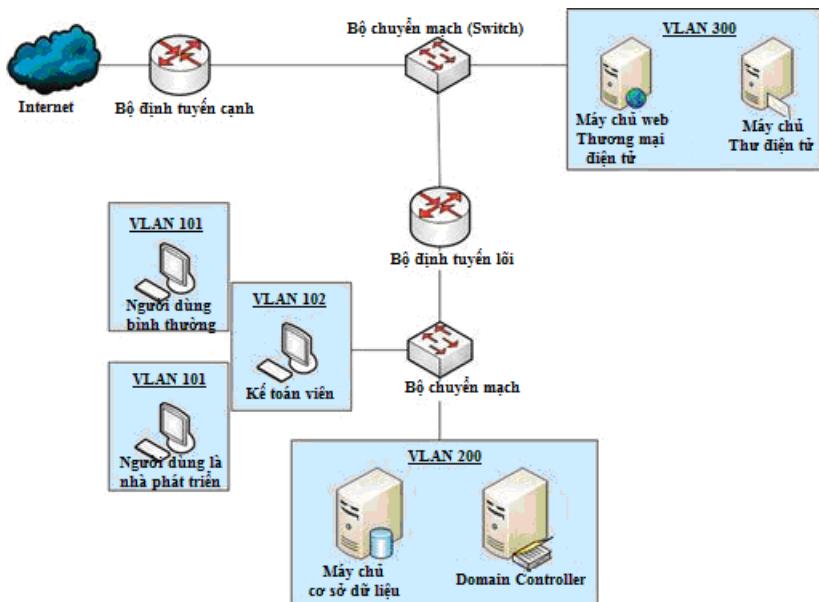
Số lượng dữ liệu và thời gian lưu trữ dữ liệu cũng cần phải được xác định. Cần phải giảm tối thiểu chi phí lưu trữ dữ liệu và tăng tối đa độ quan tâm về dữ liệu hữu ích dùng trong việc phân tích. Ví dụ, với dữ liệu bắt giữ gói tin đầy đủ (dựa trên mạng) cần quan tâm đến 2 loại dữ liệu là: (1) tất cả các số hiệu công và giao thức đến/đi từ máy chủ lưu trữ tệp tin, và (2) tất cả lưu lượng SMB được định tuyến đi ra ngoài VLAN.

Sau khi đã xác định được danh sách chi tiết các loại dữ liệu cần quan tâm, cần phải xây dựng cơ sở hạ tầng thích hợp cho việc thu thập dữ liệu. Tất nhiên, việc này có thể tốn kém về chi phí, nhưng đây là việc cần làm sau khi đã hoàn tất các giai đoạn ACF. Mục tiêu của mô hình này là xác định được những dữ liệu muốn thu thập và tầm quan trọng của các nguồn dữ liệu đó. Điều này có thể dựa trên phân tích chi phí/lợi ích, nghĩa là tầm quan trọng của dữ liệu thu thập về các nguy cơ đối với những mục tiêu của tổ chức.

Dữ liệu được tạo ra từ quá trình này với NSM là không bao giờ kết thúc. 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.

##### **2.1.2 Ví dụ tình huống: Cửa hàng bán lẻ**

Có một cửa hàng bán lẻ trực tuyến lần đầu tiên muốn thiết lập một hệ thống NSM. Cửa hàng sử dụng trang web để tiếp thị và bán hàng thủ công và các đồ lặt vặt từ các nhà cung cấp khác nhau. Toàn bộ doanh thu là từ việc bán hàng qua trang web. Hình 2.2 là một sơ đồ mạng của cửa hàng, với máy chủ truy nhập công khai trong một DMZ (Demilitarized Zone) nằm phía trong bộ định tuyến cạnh (edge router). Người dùng và máy chủ mạng nội bộ ở các VLAN khác nhau bên trong bộ định tuyến lõi (core router). Trong sơ đồ này chưa có bất kỳ cảm biến nào, là do chưa xác định được nhu cầu thu thập dữ liệu.



**Hình 2.2 Sơ đồ mạng của cửa hàng bán lẻ**

### Xác định nguy cơ của tổ chức

Cửa hàng là một đơn vị trung gian cho việc bán và phân phối các sản phẩm (do họ không tự sản xuất hàng hóa riêng). Các nguy cơ có thể được xác định như sau:

- **Tính bảo mật:** Trang web thương mại điện tử của cửa hàng thu thập và lưu trữ các thông tin của khách hàng, bao gồm cả thông tin về thẻ tín dụng của khách. Cơ sở dữ liệu này là không thể truy cập trực tiếp từ Internet. Tuy nhiên, kẻ tấn công có thể tấn công cơ sở dữ liệu lưu trữ thông tin này qua một lỗ hổng trong ứng dụng web mà nó kết nối, hoặc kẻ tấn công có thể truy cập thông tin này từ máy trạm của một nhân viên có quyền truy cập vào cơ sở dữ liệu.
- **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, ví dụ thực hiện tấn công từ chối dịch vụ. Kết quả sẽ làm gián đoạn hoạt động thương mại điện tử của trang web.
- **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, bao gồm cả việc mua sản phẩm mà không có giao dịch về tiền. Hoặc kẻ tấn công có thể khai thác một lỗ hổng trong ứng dụng web. Hoặc thậm chí, kẻ tấn công có thể thỏa hiệp với người dùng nội bộ để truy cập vào cơ sở dữ liệu back-end của trang web thương mại điện tử.

### Định lượng rủi ro

Với một danh sách các nguy cơ đối với tổ chức, có thể định lượng rủi ro như Bảng 2.1.

**Bảng 2.1 Định lượng rủi ro cho các nguy cơ của cửa hàng bán lẻ**

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 chú ý – tấn công ứng dụng web	2	4	8
Sử dụng dịch vụ thương mại điện tử không chú ý – tấn công tài sản nội mạng	2	1	2

Sắp xếp lại theo thứ tự định lượng rủi ro (ưu tiên những nguy cơ có rủi ro cao), như Bảng 2.2.

**Bảng 2.2 Sắp xếp định lượng rủi ro của cửa hàng bán lẻ**

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 chú 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 chú ý – tấn công tài sản nội mạng	2	1	2

Từ Bảng 2.2 có thể thấy nguy cơ đe dọa lớn nhất đối với tổ chức là sự gián đoạn của dịch vụ thương mại điện tử từ tấn công bên ngoài vào ứng dụng web, và nguy cơ đe dọa ít nhất là việc sử dụng dịch vụ thương mại điện tử không chú ý từ bên trong nội mạng dẫn đến tổn thất cho tài sản. Các thông tin này sẽ được dùng để định hình các lựa chọn trong bước tiếp theo.

### Xác định nguồn dữ liệu

Để xác định được các nguồn dữ liệu hữu ích cho việc phát hiện xâm nhập và phân tích NSM, phần này chỉ xem xét một số nguy cơ ở mức cao hơn.

#### Đánh cắp thông tin thẻ tín dụng của khách hàng – tấn công ứng dụng web

- 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. Để làm việc này, có thể đặt một bộ cảm biến ở cạnh mạng để thu thập dữ liệu bắt gói tin dày đặc, dữ liệu phiên, hoặc dữ liệu kiểu chuỗi. Trong trường hợp này có thể sử dụng NIDS dựa trên chữ ký hay dựa trên bất thường.
- Thu thập dữ liệu nhật ký ứng dụng cụ thể của các máy chủ web để xem xét các hoạt động của nó.

- Cần kiểm tra các giao dịch đến máy chủ cơ sở dữ liệu, do các ứng dụng web có thể hỗ trợ người dùng truy cập gián tiếp đến cơ sở dữ liệu back-end. Vì các máy chủ cơ sở dữ liệu nằm trong mạng nội bộ nên cần đặt một cảm biến thứ hai có khả năng hiển thị ở đây. Từ đó có thể thu thập dữ liệu bắt gói tin đầy đủ, dữ liệu phiên, và dữ liệu kiểu chuỗi, đồng thời cho phép sử dụng NIDS dựa trên chữ ký và NIDS dựa trên bát thường.
- 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.

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 bát 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 bát 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

#### **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 thương mại điện tử (với dịch vụ web được mở tại các cổng 80 và 443), và máy chủ thư điện tử của cửa hàng (được mở tại cổng 25 cho SMTP).
- Với cơ sở hạ tầng mạng hiện có, thì dữ liệu nhật ký tường lửa là nguồn dữ liệu điều tra rất hữu ích.
- Trong bối cảnh với các nguy cơ đã phân tích, cần có một cảm biến để thu thập dữ liệu qua giao diện mạng.
- Để có thể hỗ trợ cho việc phát hiện xâm nhập và phân tích NSM đầy đủ, 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.
- Để đảm bảo dữ liệu đầy đủ cho việc phát hiện và phân tích các sự kiện liên quan đến các loại tấn công, 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.

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

- Dữ liệu nhật ký tường lửa 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 bát thường, được thu thập qua cảm biến DMZ
- Dữ liệu nhật ký ứng dụng máy chủ cơ sở dữ liệu

- Dữ liệu nhặt ký ứng dụng máy chủ thư điện tử
- Dữ liệu nhặt 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

#### **Làm gián đoạn các dịch vụ thương mại điện tử – tấn công tài sản nội mạng**

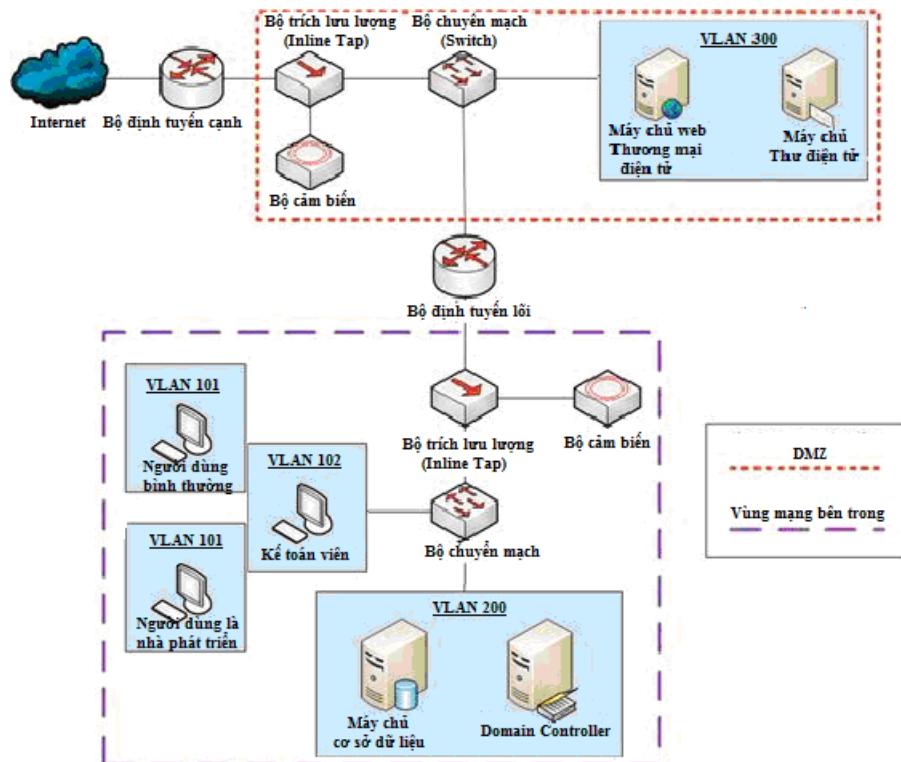
- Chỉ có các máy chủ trong VLAN 200 và những người dùng là nhà phát triển trong VLAN 103 là có quyền truy nhập vào DMZ từ bên trong mạng, do vậy, cần triển khai một cảm biến ở bên trong mạng để thu thập các dữ liệu từ các thiết bị này.
- Nếu kẻ tấn công chiếm được quyền sử dụng máy của người dùng là nhà phát triển trong nội mạng, hắn sẽ có quyền truy nhập đến DMZ. Do vậy cần thu thập dữ liệu của các hệ thống có liên quan và các nhặt ký bảo mật của máy tính nhà phát triển, cùng với dữ liệu cảnh báo HIDS và dữ liệu cảnh báo chống vi-rút. Đồng thời, để biết được những gì đi qua được các liên kết từ bên trong mạng DMZ, cần thu thập nhặt ký tường lửa từ các bộ định tuyến nội mạng.
- Khi kẻ tấn công đã tấn công được một máy trong nội mạng, thì hắn sẽ cố gắng để làm tăng vị trí trong mạng bằng cách tác động đến môi trường Active Directory Windows, do vậy, cần phải thu thập nhặt ký từ các bộ điều khiển miền (liên quan đến xử lý DNS).

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

- Dựa trên mạng:
  - Dữ liệu nhặt ký tường lửa bên cạnh mạng
  - Dữ liệu nhặt ký tường lửa bên trong mạng
  - Dữ liệu bắt gói tin đầy đủ, dữ liệu phiên, sử dụng NIDS dựa trên chữ ký và NIDS dựa trên bắt 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 bắt thường, được thu thập qua cảm biến nội mạng.
- Dựa trên máy chủ:
  - Nhặt ký dữ liệu máy chủ web, máy chủ cơ sở dữ liệu, và ứng dụng điều khiển miền.
  - Dữ liệu nhặt 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

Các danh mục nguồn dữ liệu được tạo ra từ những nguy cơ không phải là để dùng cho mọi kịch bản, nhưng những dữ liệu này sẽ là đại diện cho một số lượng hợp lý các tình huống phòng thủ tiềm năng.

Với việc xác định một loạt các nguồn dữ liệu có thể hữu ích cho việc phát hiện xâm nhập và phân tích NSM mang lại kết quả là một sơ đồ mạng thay đổi so với ban đầu, với các cảm biến và vị trí của chúng trong mạng (Hình 2.3).



**Hình 2.3 Sơ đồ mạng được cập nhật với vị trí của các cảm biến**

### Chọn lọc dữ liệu

Bước cuối cùng trong quá trình này là sàng lọc để có được những nguồn dữ liệu chính hữu ích trong nguồn dữ liệu đã thu thập. Có nhiều cách để thực hiện việc này, tuy nhiên ở đây, trong tình huống của cửa hàng bán lẻ, có thể lựa chọn dựa trên phân tích chi phí/lợi ích của họ.

Dựa trên mạng	Dựa trên máy chủ
<ul style="list-style-type: none"> <li>Dữ liệu nhặt ký từ lõi bên cạnh mạng           <ul style="list-style-type: none"> <li>Bên trong → Từ chối bên ngoài</li> </ul> </li> <li>Dữ liệu nhặt ký từ lõi bên trong (lõi mạng)           <ul style="list-style-type: none"> <li>Bên ngoài → Cho phép/Từ chối bên trong</li> <li>Bên trong → Từ chối bên ngoài</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Dữ liệu nhặt ký từ máy chủ thư điện tử, máy chủ web, máy chủ cơ sở dữ liệu và ứng dụng điều khiển miền           <ul style="list-style-type: none"> <li>Máy chủ thư điện tử – Tạo và sửa đổi tài khoản</li> <li>Máy chủ web – Các giao dịch từ miền con xử lý thanh toán</li> </ul> </li> </ul>

<ul style="list-style-type: none"> <li>• Cảm biến DMZ – Dữ liệu bắt gói tin đầy đủ           <ul style="list-style-type: none"> <li>▪ Bên ngoài → Các cổng web bên trong</li> <li>▪ Bên ngoài → Các cổng thu điện tử bên trong</li> <li>▪ Bên trong → Các cổng thu điện tử bên ngoài</li> </ul> </li> <li>• Cảm biến DMZ – Dữ liệu phiên           <ul style="list-style-type: none"> <li>▪ Tất cả các bản ghi</li> </ul> </li> <li>• Cảm biến DMZ – NIDS dựa trên chữ ký           <ul style="list-style-type: none"> <li>▪ Các luật tập trung vào tấn công ứng dụng web: SQL injection, XSS,...</li> <li>▪ Các luật tập trung vào tấn công máy chủ web</li> <li>▪ Các luật tập trung vào tấn công máy chủ thu điện tử</li> </ul> </li> <li>• Cảm biến DMZ – NIDS dựa trên bất thường           <ul style="list-style-type: none"> <li>▪ Các luật tập trung vào những bất thường trong nội dung thư và web</li> </ul> </li> <li>• Cảm biến nội mạng – Dữ liệu bắt gói tin đầy đủ           <ul style="list-style-type: none"> <li>▪ Bên trong → Các IP máy chủ web</li> <li>▪ Bên trong → Nhà phát triển VLAN 103</li> <li>▪ Bên ngoài → Máy chủ VLAN 200</li> </ul> </li> <li>• Cảm biến nội mạng – Dữ liệu phiên           <ul style="list-style-type: none"> <li>▪ Tất cả các bản ghi</li> </ul> </li> <li>• Cảm biến nội mạng – Dữ liệu kiểu chuỗi trong gói tin           <ul style="list-style-type: none"> <li>▪ Nhà phát triển VLAN 103 → Bên ngoài</li> </ul> </li> <li>• Cảm biến nội mạng – NIDS dựa trên chữ ký           <ul style="list-style-type: none"> <li>▪ Các luật tập trung vào tấn công cơ sở dữ liệu</li> <li>▪ Các luật tập trung vào tấn công và các hoạt động quản trị bộ điều khiển miền</li> <li>▪ Các luật phần mềm độc hại chung</li> </ul> </li> <li>• Cảm biến nội mạng – NIDS dựa trên bất thường           <ul style="list-style-type: none"> <li>▪ Các luật tập trung vào tương tác cơ sở dữ liệu bất thường</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>▪ Máy chủ web – Các giao dịch từ miền con quản trị</li> <li>▪ Máy chủ cơ sở dữ liệu – Tạo và sửa đổi tài khoản</li> <li>▪ Máy chủ cơ sở dữ liệu – Các giao dịch thanh toán</li> <li>▪ Máy chủ cơ sở dữ liệu – Các giao dịch quản trị</li> <li>▪ Bộ điều khiển miền – Tạo và sửa đổi tài khoản</li> <li>▪ Bộ điều khiển miền – Tạo và sửa đổi máy tính</li> <li>• Dữ liệu nhật ký bảo mật và hệ điều hành máy chủ thu điện tử, máy chủ web, VLAN 200 và VLAN 103           <ul style="list-style-type: none"> <li>▪ Tạo và sửa đổi tài khoản</li> <li>▪ Các thông báo phần mềm được cài đặt</li> <li>▪ Các thông báo cập nhật hệ thống</li> <li>▪ Thông báo khởi động lại hệ thống</li> </ul> </li> <li>• Dữ liệu cảnh báo chống vi-rút máy chủ thu điện tử, máy chủ web, VLAN 200 và VLAN 103           <ul style="list-style-type: none"> <li>▪ Tất cả dữ liệu cảnh báo</li> </ul> </li> <li>• Dữ liệu cảnh báo HIDS máy chủ thu điện tử, máy chủ web và VLAN 103 Alert Data           <ul style="list-style-type: none"> <li>▪ Tất cả các cảnh báo liên quan đến những thay đổi tệp tin hệ thống chính</li> <li>▪ Tất cả những thay đổi liên quan đến tạo/sửa đổi tài khoản.</li> </ul> </li> </ul>
--	--

## 2.2 KIẾN TRÚC CẢM BIẾN

Ngoài con người thì 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 lĩnh vực 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 chương trình NSM là thu thập dữ liệu, phát hiện xâm nhập và phân tích dữ liệu.

### 2.2.1 Các loại dữ liệu NSM

Phần sau sẽ trình bày chi tiết về các loại dữ liệu khác nhau của NSM. Trong phần này, để thảo luận về kiến trúc của cảm biến, chúng ta cần có một cái nhìn tổng quan về các loại dữ liệu NSM chính được thu thập để dùng trong phát hiện xâm nhập và phân tích.

**Dữ liệu bắt gói tin đầy đủ (dữ liệu FPC).** Cung cấp thông tin đầy đủ về tất cả các gói dữ liệu được truyền giữa hai điểm đầu cuối. Các loại dữ liệu FPC phổ biến nhất là theo định dạng dữ liệu PCAP. Loại dữ liệu này được sử dụng nhiều nhất và được đánh giá cao về giá trị do tính chất đầy đủ của nó, và rất phù hợp trong ngữ cảnh phân tích. Các loại dữ liệu khác, như dữ liệu thống kê hay dữ liệu chuỗi trong gói tin, thường bắt nguồn từ dữ liệu FPC.

**Dữ liệu phiên.** Là bản tóm tắt các thông tin giữa hai thiết bị mạng, là một trong những hình thức linh hoạt và hữu ích nhất của dữ liệu NSM. Tuy dữ liệu phiên không cung cấp mức độ chi tiết như dữ liệu FPC, nhưng với kích thước nhỏ nó sẽ được lưu lại trong khoảng thời gian dài hơn nhiều, và điều này là vô cùng quý giá khi thực hiện phân tích lại quá khứ.

**Dữ liệu thống kê.** Là dữ liệu tổ chức, phân tích, giải thích và biểu diễn các loại dữ liệu khác. Dữ liệu thống kê có thể bao gồm nhiều hình thức khác nhau.

**Dữ liệu kiểu chuỗi trong gói tin (PSTR).** Được lấy từ dữ liệu FPC, và tồn tại như một dạng dữ liệu trung gian giữa dữ liệu FPC và dữ liệu phiên. Định dạng dữ liệu này bao gồm các chuỗi văn bản rõ ràng từ tiêu đề (header) của các giao thức (ví dụ, dữ liệu trong phần tiêu đề của HTTP). Kết quả là có được dạng dữ liệu chi tiết gần giống với dữ liệu FPC và duy trì một kích thước dễ quản lý hơn và cho phép tăng lượng dữ liệu lưu trữ.

**Dữ liệu nhật ký.** Dữ liệu nhật ký là các tệp tin nhật ký thô được tạo ra từ các thiết bị, hệ thống hoặc ứng dụng, bao gồm nhật ký web-proxy, nhật ký tường lửa định tuyến, nhật ký chứng thực VPN, nhật ký bảo mật Windows và dữ liệu SYSLOG. Loại dữ liệu này thay đổi kích thước và tính hữu dụng của nó tùy thuộc vào nguồn gốc.

**Dữ liệu cảnh báo.** Khi công cụ phát hiện ra bất kỳ một bất thường nào trong dữ liệu mà nó kiểm tra, thì nó sẽ tạo ra một loại dữ liệu gọi là dữ liệu cảnh báo. Dữ liệu này thường chứa mô tả của các cảnh báo, và một con trỏ chỉ đến dữ liệu bất thường. Nói chung, kích thước của dữ liệu cảnh báo thường rất nhỏ, có khi chỉ là con trỏ chỉ đến dữ liệu khác. Việc phân tích NSM thường dựa trên các thể hệ của dữ liệu cảnh báo.

Khi xem xét một cách tổng thể về các loại dữ liệu, thường so sánh tính hữu ích với kích thước của nó. Dạng dữ liệu lớn nhất thường là FPC, tiếp theo là PSTR, và sau đó là dữ liệu phiên. Các loại dữ liệu nhật ký, cảnh báo, thống kê thường là rất nhỏ so với các loại dữ liệu khác, và có thể tùy biến theo các loại dữ liệu đang thu thập và nguồn dữ liệu đang sử dụng.

### 2.2.2 Các loại cảm biến

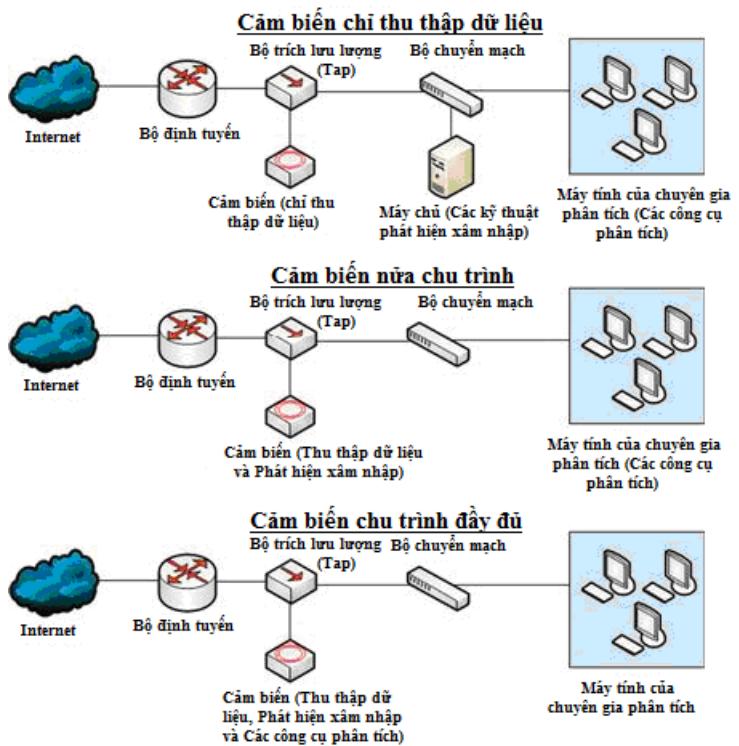
Tùy thuộc vào kích thước và các nguy cơ phải đối mặt của một hệ thống mạng, mà các cảm biến có thể có vai trò khác nhau trong các giai đoạn của chu trình NSM.

**Cảm biến chỉ thu thập dữ liệu (collection-only sensor).** Đơn giản là 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 (dữ liệu thống kê và PSTR) dựa trên những gì đã thu thập được. Loại cảm biến này thường được dùng trong các tổ chức lớn, nơi mà các công cụ phát hiện xâm nhập cần truy nhập dữ liệu thu thập từ xa để thực hiện xử lý. Việc phân tích cũng được thực hiện tách rời với cảm biến, vì dữ liệu phù hợp sẽ được đưa vào các thiết bị khác khi cần. Cảm biến chỉ thu thập dữ liệu là một khung tràn và không được cài đặt phần mềm mở rộng, vì vậy các chuyên gia phân tích cũng hiếm khi truy cập trực tiếp được.

**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, cảm biến nửa chu trình sẽ ghi dữ liệu PCAP vào ổ đĩa, nhưng cũng sẽ chạy một NIDS (như Snort) trong thời gian thực từ NIC, hoặc gần trong thời gian thực để chống lại việc ghi dữ liệu PCAP vào ổ đĩa. Khi thực hiện phân tích, dữ liệu sẽ được đưa trở lại thiết bị khác thay vì được phân tích trên chính cảm biến. Các loại cảm biến này được triển khai phổ biến nhất, với việc các chuyên gia phân tích có thể truy nhập trực tiếp vào cảm biến để tương tác với các công cụ phát hiện khác nhau.

**Cảm biến phát hiện chu trình đầy đủ (full cycle detection sensor).** Đây là loại cảm biến 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. Nghĩa là, ngoài các công cụ thu thập dữ liệu và phát hiện xâm nhập, cảm biến còn được bổ sung một bộ đầy đủ các công cụ phân tích dữ liệu, có thể bao gồm hồ sơ phân tích cá nhân của cảm biến, môi trường giao diện máy tính đồ họa, hoặc cài đặt giao diện ứng dụng đồ họa NIDS như là Snorby. Với cảm biến phát hiện chu trình đầy đủ, hầu hết các nhiệm vụ của NSM đều được thực hiện trên chính cảm biến. Loại cảm biến này thường được dùng trong các tổ chức rất nhỏ, nơi mà chỉ có duy nhất một cảm biến hoặc là tài nguyên phần cứng của tổ chức bị hạn chế.

Trong ba loại cảm biến trên, cảm biến nửa chu trình là loại được sử dụng nhiều hơn cả. Điều này là do có thể dễ dàng cài đặt các công cụ phát hiện trên cùng hệ thống mà dữ liệu được thu thập. Cảm biến nửa chu trình cũng an toàn hơn cho các chuyên gia phân tích khi họ lấy các bản sao dữ liệu về máy tính phân tích chuyên dụng của họ mà không tương tác với dữ liệu thật trên chính nó. Điều này ngăn cản dữ liệu bị xử lý sai mà có thể dẫn đến việc mất mát các thông tin quan trọng. Mặc dù các chuyên gia phân tích sẽ cần phải tương tác với các cảm biến theo một mức độ nào đó, nhưng họ cũng không nên sử dụng chúng như là một môi trường phân tích desktop trừ khi không có lựa chọn nào khác. Các cảm biến cần phải được bảo vệ như là một tài sản có tầm quan trọng rất lớn trong hệ thống mạng.



**Hình 2.4 Các loại cảm biến**

### 2.2.3 Phản ứng cảm biến

Một tổ chức muốn thực hiện giám sát an toàn mạng một cách thực sự nghiêm túc thì cần phải đầu tư vào phản ứng tin cậy, và phản ứng cho các cảm biến ở đây nên thuộc cấp độ của máy chủ (những loại khác thường chỉ được chấp nhận với kịch bản phòng thí nghiệm). Các yếu tố kỹ thuật cần thiết để xác định số lượng tài nguyên phản ứng cần thiết bao gồm các loại cảm biến được triển khai, số lượng dữ liệu được thu thập bởi các cảm biến, và số lượng dữ liệu cần được lưu giữ.

Có một cách được sử dụng để xác định các yêu cầu phản ứng quan trọng của cảm biến là thiết lập và cấu hình một cảm biến tạm thời. Cảm biến này có thể là một máy chủ, máy trạm hoặc thậm chí là một máy tính xách tay. Vị trí của cảm biến cũng cần được xác định trước khi các cảm biến này được cài đặt, bao gồm cả vị trí vật lý và vị trí logic để xác định những gì mà các cảm biến trên mạng cần phải theo dõi.

Sau khi các cảm biến tạm thời được cài đặt trên mạng, cần 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. Sau đó, có thể 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. Chú ý là không nhất thiết phải có một cảm biến tạm thời rất mạnh có khả năng xử lý tất cả những công cụ này khi chúng được kích hoạt đồng thời, mà chỉ cần kích hoạt các công cụ riêng để tính toán tải trọng hoạt động của nó, và sau đó sẽ tổng hợp kết quả từ tất cả các công cụ để đánh giá nhu cầu tổng thể.

**CPU.** Số lượng tài nguyên CPU yêu cầu chủ yếu sẽ phụ thuộc vào các loại cảm biến được triển khai. Nếu triển khai một bộ cảm biến chỉ thu thập dữ liệu thì có khả năng sẽ không cần quá nhiều sức mạnh xử lý, nhưng với cảm biến phát hiện xâm nhập thì lại cần rất nhiều CPU. Do vậy, trong trường hợp triển khai cảm biến nửa chu trình hoặc cảm biến chu trình đầy đủ, thì nên lập kế hoạch cho việc bổ sung CPU hoặc CPU nhiều nhân.

**Bộ nhớ.** Lượng bộ nhớ cần cho việc thu thập dữ liệu và phát hiện xâm nhập thường nhỏ hơn nhiều so với phân tích dữ liệu. Nói chung, các cảm biến cần phải có bộ nhớ lớn, việc này sẽ làm tăng đáng kể chi phí, đặc biệt là đối với cảm biến chu trình đầy đủ. Thường rất khó để lên kế hoạch cho bộ nhớ, nên tốt nhất là mua phần cứng có khe cắm bộ nhớ để có thể bổ sung khi cần.

**Ô cứng lưu trữ dữ liệu.** Quy hoạch ô cứng cũng là một công việc khá khó khăn, là do có quá nhiều yếu tố cần xem xét. Để có được kế hoạch hiệu quả cho nhu cầu lưu trữ dữ liệu, đòi hỏi phải xác định được vị trí của các cảm biến và các hoạt động của chúng. Thậm chí ngay cả khi triển khai cảm biến, nhu cầu lưu trữ cũng thường phải được đánh giá lại.

Các bước giúp đánh giá nhu cầu lưu trữ cho một cảm biến bao gồm:

1. *Tính toán lưu lượng thu thập:* Dùng cảm biến tạm thời để tính toán nhu cầu lưu trữ dữ liệu bằng cách xác định số lượng dữ liệu thu thập được qua NSM trong một khoảng thời gian nhất định. Ví dụ, thu thập ít nhất 24 giờ trong nhiều giai đoạn, với thời gian thu thập vào một ngày trong tuần và một ngày cuối tuần để có được mô tả chính xác về lưu lượng dữ liệu cả lúc cao điểm và thấp điểm. Sau khi thu thập được một số tập dữ liệu sẽ có thể lấy giá trị trung bình để xác định lượng dữ liệu tạo ra theo mỗi giờ.
2. *Xác định thời gian lưu trữ khả thi cho mỗi loại dữ liệu:* Mỗi tổ chức cần xác định một tập các hoạt động tối thiểu và thời gian lưu trữ lý tưởng cho các loại dữ liệu NSM. Các hoạt động tối thiểu là yêu cầu tối thiểu để thực hiện các dịch vụ NSM ở mức chấp nhận được. Các hoạt động lý tưởng được thiết lập như là một mục tiêu hợp lý để thực hiện NSM đến mức tốt nhất có thể. Việc xác định những con số này phụ thuộc vào sự nhạy cảm của các hoạt động của tổ chức và ngân sách có thể dành cho phần cứng cảm biến. Khi đã xác định được những con số này, có thể áp dụng chúng vào với lượng dữ liệu thu thập để xem cần bao nhiêu không gian cho việc đáp ứng các mục tiêu duy trì.
3. *Bổ sung nhu cầu lưu trữ cho các loại cảm biến:* Hầu hết trong các sản phẩm, thường hệ điều hành và các công cụ cảm biến sẽ nằm trên một ổ đĩa logic, và các dữ liệu được thu thập và lưu trữ bởi cảm biến sẽ nằm ở một ổ đĩa khác. Tuy nhiên, để tính toán tổng số không gian đĩa cần thiết, cần phải tính toán cho hệ điều hành và các công cụ sẽ được cài đặt. Đây là đối với trường hợp cảm biến chỉ thu thập dữ liệu. Có một hướng dẫn chung cho việc tính toán bổ sung nhu cầu lưu trữ dữ liệu cho các loại cảm biến khác, có thể thay đổi tùy thuộc vào mục tiêu của tổ chức và các mạng cá nhân, như sau. Nếu triển khai cảm biến nửa chu trình thì cần thêm 10% nhu cầu lưu trữ để phù hợp cho các công cụ phát hiện xâm nhập và các dữ liệu cảnh báo được tạo ra. Nếu triển khai cảm biến chu trình đầy đủ thì cần thêm 25% cho cả các công cụ phát hiện xâm nhập và phân tích dữ liệu. Sau đó,

cũng cần thêm 10-25% cho các yêu cầu đối với hệ điều hành và các công việc phát triển mạng lưới sau này.

**Giao diện mạng.** Card (thẻ) giao diện mạng (NIC) là thành phần phần cứng quan trọng nhất trong các cảm biến, vì NIC có trách nhiệm thu thập dữ liệu dùng cho tất cả ba giai đoạn của chương trình NSM. Một cảm biến nên luôn có tối thiểu hai NIC. Một NIC được sử dụng để truy cập vào máy chủ, hoặc là cho mục đích quản trị hoặc là cho mục đích phân tích dữ liệu. Các NIC khác được dành riêng cho các nhiệm vụ 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 được lựa chọn.

Để đánh giá chính xác những gì sẽ cần cho một NIC, cần có đánh giá về lưu lượng mạng sẽ thu thập. Phương pháp đơn giản nhất là đánh giá lượng truy cập vào một liên kết thông qua việc giám sát trên một bộ định tuyến hoặc một chuyển mạch. Hai thông số quan trọng nhất là: (1) đỉnh điểm của lưu lượng (đo bằng Mbps), và (2) băng thông trung bình (thông lượng) mỗi ngày (đo bằng Mbps).

**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, nhưng PF\_Ring của Luca Deri thì lại phù hợp. PF\_Ring tối ưu hóa hiệu suất socket mạng thông qua hai chế độ hoạt động: (1) từng gói tin luân chuyển theo vòng, hoặc (2) đảm bảo toàn bộ dòng lưu chuyển gói tin được chuyển giao cho một quá trình duy nhất hoặc đi đến cảm biến, do vậy quá trình thu thập dữ liệu sẽ nhanh và hiệu quả hơn. PF\_Ring không phải là lựa chọn duy nhất để sử dụng với các công cụ phát hiện xâm nhập thông thường như Bro, Snort, hoặc Suricata, nhưng nó là phổ biến nhất, và nó được hỗ trợ bởi tất cả các công cụ trên.

**Các cổng SPAN và bộ trích dữ liệu mạng (network tap).** Trong kiến trúc của cảm biến, cần phải có một thiết bị thu các gói tin đến các bộ cảm biến. Tùy thuộc vào nơi đặt cảm biến, thiết bị có thể được chọn để sử dụng có thể là một cổng SPAN hoặc một bộ trích dữ liệu mạng.

Cổng SPAN là cách đơn giản nhất để thu được các gói tin đến cảm biến. Cổng SPAN là một chức năng của một switch (cáp doanh nghiệp) cho phép sao chép một hoặc nhiều cổng switch vật lý đến một cổng khác. Để thực hiện điều này, trước hết phải xác định cổng có lưu lượng mong muốn đến đi cảm biến, ví dụ thông thường nhất là các cổng kết nối bộ định tuyến upstream tới chuyển mạch, hoặc cũng có thể là một số cổng cá nhân có tài sản quan trọng.

Bộ trích dữ liệu thường là giải pháp ưu tiên trong các kịch bản có hiệu suất cao và yêu cầu độ tin cậy lớn hơn. Chúng có thể đến trong tất cả các hình dạng, kích thước và có thể mở rộng lên đến mức độ hiệu suất cao. Những gì thu được qua thiết bị này là rất xứng đáng so với chi phí khi lựa chọn nó, và do vậy bộ trích dữ liệu thường được sử dụng trong các liên kết quan trọng.

#### 2.2.4 Hệ điều hành cảm biến

Việc triển khai cảm biến phổ biến nhất là trên Linux hoặc BSD. Mỗi loại đều có ưu điểm và nhược điểm, và việc triển khai thường tùy thuộc vào sở thích cá nhân. Ví dụ, những người có nền tảng về DoD thường thích một cái gì đó dựa trên Red Hat như CentOS hoặc Fedora, do DoD chủ yếu là sử dụng Red Hat Linux. Trong khi đó, một số nhà chuyên môn NSM khác lại thích FreeBSD hoặc OpenBSD do bản chất đơn giản của chúng.

Trong khi nền tảng hệ điều hành được chọn là không quá quan trọng, thì điều quan trọng hơn là việc sử dụng một cái gì đó dựa trên \*nix. Có nhiều lý do ở đây, nhưng lý do phổ biến nhất là 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. Trong năm 2013, Linux gần như là sự lựa chọn phổ biến nhất, vì các nhà sản xuất phần cứng cung cấp toàn bộ trình điều khiển Linux cho phần cứng của họ.

#### 2.2.5 Vị trí đặt cảm biến

Có lẽ 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 sẽ quyết định xem có thể bắt được dữ liệu gì, phát hiện nào có thể có được liên quan đến dữ liệu đó, và mức độ mở rộng cho việc phân tích được đến đâu. Mục đích của vị trí cảm biến là để đảm bảo việc hiển thị thích hợp trong nguồn cung cấp dữ liệu đã được thiết lập như là tiến trình quan trọng NSM trong tổ chức.

Không có phương pháp thử và sai đối với việc xác định nơi để đặt một bộ cảm biến tốt nhất trên mạng, nhưng có một số thủ thuật và thực hành tốt nhất có thể giúp tránh được những bẫy thông thường.

**Sử dụng các tài nguyên thích hợp.** Vị trí đặt cảm biến là mục tiêu của nhóm bảo mật, do vậy cần phải xác định cách tốt nhất cho việc tích hợp các thiết bị này vào hệ thống mạng. Với ý định này, nhóm nghiên cứu bảo mật 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.** Trong trường hợp lý tưởng, và khi các nguồn lực thích hợp có sẵn, 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.

**Tầm nhìn của địa chỉ Internet cục bộ.** Khi thực hiện phát hiện xâm nhập và phân tích dữ liệu, điều quan trọng là khả năng xác định thiết bị nội bộ nào là đối tượng chính của một cảnh báo. Nếu cảm biến được đặt ở phía trái của một thiết bị NAT là một bộ định tuyến thì có thể được bảo vệ từ các thông tin này.

**Đánh giá tài sản quan trọng.** Tổ chức cần phải có quy định tài sản nào là quan trọng nhất cần được bảo vệ. Với ý định này, khi chỉ có nguồn lực hạn chế và không thể đủ khả năng để thực hiện thu thập dữ liệu và phát hiện xâm nhập ở tất cả các điểm đi vào/đi ra của mạng, tổ chức vẫn 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.** Sơ đồ mạng (bao gồm vị trí của các cảm biến) là vô cùng 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. Tuy nhiên, không phải lúc nào các chuyên gia phân tích cũng cần thông tin về sơ đồ mạng vì nó có thể gây quá tải thông tin (nếu sơ đồ được thể hiện quá mức chi tiết; trong nhiều trường hợp, các chuyên gia phân tích chỉ cần một sơ đồ mạng đơn giản, phù hợp với nhu cầu thông tin của họ).

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.

### 2.2.6 Bảo mật cho cảm biến

Trong lĩnh vực NSM, sự an toàn của các cảm biến nên được coi là tối quan trọng. Khi cảm biến lưu trữ dữ liệu bắt gói tin đầy đủ, hoặc thậm chí là dữ liệu PSTR, thì rất có khả năng các tập tin này sẽ chứa các thông tin mạng vô cùng nhạy cảm. Trong khi, ngay cả một kẻ tấn công không có tay nghề cũng có thể sử dụng các tập tin này để trích xuất toàn bộ tệp tin, mật khẩu, hoặc các dữ liệu quan trọng khác. Thậm chí, kẻ tấn công có thể sử dụng một cảm biến chỉ lưu trữ dữ liệu phiên để lấy được thông tin về mạng và cho phép hắn nâng cấp vị trí trong hệ thống.

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:* Cần đảm bảo các phần mềm và hệ điều hành của cảm biến được cập nhật bản vá lỗi bảo mật mới nhất.
- *Bảo mật hệ điều hành:* Ngoài vấn đề quan trọng về cập nhật, hệ điều hành còn cần phải được cấu hình bảo mật tốt nhất trước khi phần mềm cảm biến được cài đặt. Có một số phương pháp để tiếp cận bảo mật hệ điều hành, như sử dụng các loại chuẩn chính thức (nếu tổ chức thuộc các loại chuẩn này) như HIPAA, NERC CIP, hoặc PCI, hoặc sử dụng một số nguồn công khai có sẵn như chuẩn CIS (Center for Internet Security, <http://benchmarks.cisecurity.org/>) hay NSA Security Guides for Operating Systems ([http://www.nsa.gov/ia/mitigation\\_guidance/](http://www.nsa.gov/ia/mitigation_guidance/) / [security\\_configuration\\_guides/](http://www.nsa.gov/ia/security_configuration_guides/) [operating\\_systems.shtml](http://www.nsa.gov/ia/operating_systems.shtml)).

- *Hạn chế truy cập Internet:* Trong hầu hết các trường hợp, cảm biến không nên có quyền truy cập Internet. Nếu các cảm biến bị tấn công thì kẻ tấn công sẽ dễ dàng có được các thông tin nhạy cảm từ đó. Thường chỉ cung cấp truy cập Internet cho cảm biến giới hạn trong các lĩnh vực quan trọng (như yêu cầu cập nhật phần mềm và hệ thống) với việc sử dụng proxy web nội bộ; cấu hình để tải chữ ký IDS tại các khoảng thời gian định kỳ; tiếp cận các nguồn thông tin tình báo;...
- *Tối thiểu hóa cài đặt phần mềm:* Chỉ cài đặt các phần mềm chuyên ngành cần thiết trên các cảm biến để thực hiện các công việc thu thập dữ liệu theo yêu cầu, phát hiện xâm nhập và phân tích dữ liệu. Cần vô hiệu hóa và gỡ bỏ tất cả các dịch vụ không cần thiết và những gói bổ sung không sử dụng cài đặt với hệ điều hành. Việc này sẽ làm tăng hiệu suất của cảm biến và làm giảm tối thiểu những tấn công tiềm tàng.
- *Phân đoạn VLAN:* Hầu hết các cảm biến có ít nhất hai kết nối mạng. Giao diện thứ nhất dùng để thu thập dữ liệu mạng, còn giao diện thứ hai sẽ được dùng trong việc quản lý các cảm biến (thường thông qua SSH). Trong khi giao diện thu thập dữ liệu không nên được gán một địa chỉ IP hoặc được phép giao tiếp, thì giao diện quản trị sẽ được yêu cầu tồn tại một cách logic tại một số vị trí trên mạng. Nếu môi trường mạng hỗ trợ các phân đoạn của lưu lượng với VLAN (Virtual Local Area Networks), thì sau đó sẽ có thể bị lợi dụng, và giao diện quản lý cảm biến được đặt vào bên trong VLAN bảo mật sẽ chỉ được truy nhập bởi người quản trị cảm biến.
- *IDS dựa trên máy chủ:* Việc cài đặt một số hình thức phát hiện xâm nhập trái phép dựa trên máy chủ (HIDS) trên cảm biến là rất quan trọng. Các hệ thống này cung cấp phát hiện xâm nhập từ những thay đổi của các máy chủ thông qua một loạt các phương tiện, bao gồm cả việc giám sát các bản ghi hệ thống và phát hiện việc sửa đổi tệp tin hệ thống. Có một số phần mềm HIDS thương mại, nhưng cũng có một số phần mềm miễn phí có sẵn, như OSSEC hoặc Mô hình phát hiện xâm nhập nâng cao (AIDE).
- *Hai yếu tố xác thực:* Với một kẻ tấn công, cảm biến NSM là một mục tiêu có giá trị. Các dữ liệu thô và đã được xử lý của mạng được tìm thấy trên một cảm biến có thể được sử dụng để dàn xếp hoặc tiếp tục một loạt các cuộc tấn công. Vì vậy, điều quan trọng là phải bảo vệ được quá trình xác thực khi truy cập vào các cảm biến. Nếu chỉ dùng một mật khẩu xác thực thì kẻ tấn công vẫn có thể vượt qua được từ nguồn khác và sau đó truy nhập vào cảm biến. Do vậy khuyến khích nên phải có hai hình thức xác thực cho các cảm biến.
- *IDS dựa trên mạng:* Điều quan trọng là giao diện quản trị của cảm biến được coi là một tài sản mạng có giá trị cao. Cách tốt nhất để thực hiện điều này là để giao diện quản trị thực hiện việc phát hiện NIDS cho phần còn lại của mạng. Tuy nhiên, phần mềm NIDS có thể chạy trên chính cảm biến, nhưng có giải pháp tốt hơn là tạo ra bản sao lưu lượng mạng của giao diện quản trị trên giao diện giám sát (việc này là một bước dễ dàng thực hiện, tuy nhiên thường bị bỏ qua).

Cách tốt nhất để đảm bảo các cảm biến không giao tiếp với bất kỳ máy chủ trái phép nào là phải xác định máy chủ nào được phép giao tiếp với các cảm biến và tạo ra các luật trong Snort cho việc phát hiện giao tiếp với các thiết bị khác.

## 2.3 DỮ LIỆU PHIÊN

Dữ liệu phiên là bản tóm tắt các thông tin liên lạc giữa hai thiết bị mạng. Loại dữ liệu này cũng được biết đến như là một cuộc hội thoại hoặc một luồng lưu lượng. Dữ liệu tóm tắt phiên là một trong những hình thức linh hoạt và hữu ích nhất của dữ liệu NSM. Mặc dù dữ liệu phiên không cung cấp mức độ chi tiết như trong dữ liệu bắt gói tin đầy đủ, nhưng nó có một số điểm mạnh duy nhất có thể cung cấp giá trị đáng kể cho các chuyên gia phân tích NSM.

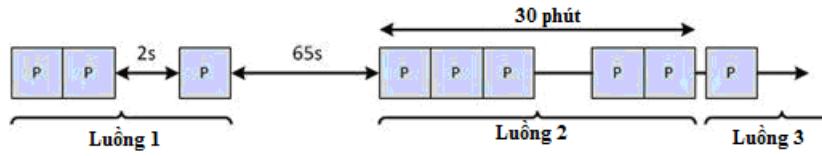
### 2.3.1 Luồng dữ liệu

Luồng dữ liệu là một bản ghi tổng hợp của các gói tin. Việc tổng hợp được diễn ra theo nhiều cách khác nhau, tùy thuộc vào công cụ đang được sử dụng để tạo và phân tích dữ liệu. Tài liệu này tập trung chủ yếu trên công cụ SiLK, do vậy, phần này sẽ mô tả cách SiLK tích hợp dữ liệu từ luồng dữ liệu.

Một luồng được xác định dựa trên 5 thuộc tính, tạo thành bộ-5 chuẩn (standard 5-tuple), gồm địa chỉ IP nguồn, cổng nguồn, địa chỉ IP đích, cổng đích và giao thức vận chuyển. Khi bộ tạo luồng phân tích gói tin, bộ 5 thuộc tính này sẽ được kiểm tra và ghi lại, và tạo ra một luồng mới với dữ liệu bộ-5, xác định loại lưu lượng đang sử dụng (NetFlow v5, NetFlow v9, IPFIX,...).

Khi một gói tin mới được phân tích và chứa cùng các giá trị thuộc tính bộ-5, thì sau đó dữ liệu sẽ được nối thêm vào luồng dữ liệu đã tồn tại. Dữ liệu sẽ được nối thêm vào luồng này cho đến khi nhìn thấy được các gói tin có cùng giá trị các thuộc tính bộ-5. Có ba điều kiện mà luồng dữ liệu có thể được kết thúc:

1. *Tự hết thời gian (Natural Timeout)*: Bất cứ khi nào giao tiếp tự kết thúc dựa trên đặc tả của giao thức. Việc này được theo dõi cho các giao thức hướng kết nối, và sẽ tìm kiếm các gói tin RST hoặc các số tuần tự FIN trong TCP.
2. *Hết thời gian chờ (Idle Timeout)*: Khi không có dữ liệu cho luồng nhận trong vòng ba mươi giây của gói tin cuối cùng thì luồng sẽ được kết thúc. Bất kỳ gói dữ liệu nào mới cùng với bộ-5 giá trị thuộc tính sau khoảng thời gian ba mươi giây đã trôi qua sẽ tạo ra một luồng mới. Đây là giá trị được cấu hình.
3. *Hết thời gian hoạt động (Active Timeout)*: Khi một luồng đã được mở trong khoảng thời gian ba mươi phút thì luồng sẽ được chấm dứt, và một luồng mới sẽ được tạo ra với cùng các giá trị thuộc tính bộ-5. Đây là giá trị được cấu hình.



**Hình 2.5 Kết thúc luồng chờ và luồng hoạt động**

Phần sau sẽ trình bày tóm tắt một số loại luồng chính.

### NetFlow

Được phát triển bởi Cisco vào năm 1990 và đã trải qua 9 phiên bản của NetFlow trong hơn 20 năm. Mục đích ban đầu là tinh giản quá trình định tuyến trên các thiết bị mạng. Trong mô tả ban đầu, luồng dữ liệu được tạo ra khi bộ định tuyến xác định được gói tin đầu tiên trong cuộc hội thoại mới trên mạng. Việc này giúp cho các cuộc đàm thoại cơ bản và cung cấp tài liệu tham khảo cho các bộ định tuyến để so sánh với các thiết bị và các dịch vụ khác trên mạng. Luồng dữ liệu cũng được sử dụng để xác định và tổng hợp số lượng lớn lưu lượng truy cập để đơn giản hóa nhiều quá trình, ví dụ như so sánh ACL. Với các phiên bản khác nhau thì tính năng cũng khác nhau rất nhiều, và được dùng vào những công việc khác nhau, từ hỗ trợ cơ sở hạ tầng đến phát triển ứng dụng bảo mật.

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. Luồng dữ liệu NetFlow v5 cung cấp thông tin theo chuẩn bô-5 cũng như tất cả các số liệu thống kê cần thiết để phân tích các gói tin. Không giống như NetFlow v9 và IPFIX, NetFlow v5 không hỗ trợ giao thức IPv6, do vậy sẽ bị hạn chế sử dụng trong một số môi trường nhất định.

NetFlow v9 có tất cả các tính năng của v5, và còn bổ sung thêm các tính năng khác. Trong khi, NetFlow v5 có 20 trường dữ liệu (trong đó có 2 trường còn để trống (padding)), thì NetFlow v9 có tới 104 trường dữ liệu, và do vậy, người quản trị có thể sử dụng NetFlow v9 để tạo ra luồng tương tự như luồng v5. NetFlow V9 cũng có hỗ trợ IPv6.

### IPFIX

Có rất 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 trong IPFIX là thông điệp. Một thông điệp có chứa một tiêu đề và một hoặc nhiều tập, trong đó có chứa các bản ghi. Một tập có thể là một mẫu mô tả các bản ghi dữ liệu trong tập đó. Sự khác biệt giữa NetFlow v9 và IPFIX là ở chức năng. Ví dụ, IPFIX cung cấp các trường dữ liệu có chiều dài thay đổi để xuất các thông tin tùy biến, nhưng NetFlow v9 thì không. 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, trong đó sử dụng lấy mẫu luồng để làm giảm tải cho CPU bằng cách chỉ dùng mẫu đại diện của dữ liệu trên liên kết. Các

biến thể của sFlow đang dần trở nên phổ biến với các nhà cung cấp. Bản thân sFlow cũng được tích hợp vào các thiết bị và các giải pháp phân cứng.

Ngoài ra, còn có Jflow được cung cấp bởi thiết bị Juniper; AppFlow được cung cấp bởi Citrix,...

### 2.3.2 Thu thập dữ liệu phiên

Dữ liệu phiên được thu thập theo một số cách khác nhau. Dù dùng phương pháp nào thì cũng cần 2 thành phần là một bộ sinh luồng và một bộ thu thập dữ liệu. Bộ sinh luồng là thành phần phân cứng hoặc phần mềm, có trách nhiệm tạo ra các luồng dữ liệu. Việc này được thực hiện bằng cách hoặc là 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.

Thông thường, khi thực hiện thu thập dữ liệu FPC sẽ chọn sinh luồng dữ liệu từ dữ liệu FPC này. Tuy nhiên, trong hầu hết các trường hợp, dữ liệu FPC đang thu thập sẽ được lọc, nghĩa là sẽ không thể sinh ra luồng dữ liệu cho lưu lượng mạng không bắt được. Hơn nữa, nếu gói tin bị mất trong quá trình bắt FPC, thì rất có thể sẽ bị mất dữ liệu luồng có giá trị. Mặc dù, việc lọc dữ liệu loại này rất có giá trị trong việc tối đa hóa sử dụng ô đĩa khi thu thập dữ liệu FPC, nhưng các dữ liệu luồng được kết hợp với lưu lượng này nên được giữ lại. Phương pháp sinh luồng dữ liệu này thường không được khuyến khích.

Phương pháp thích hợp hơn cho việc sinh dữ liệu phiên là bắt trực tiếp dữ liệu trên liên kết theo cùng cách mà dữ liệu FPC hoặc dữ liệu cảnh báo NIDS được tạo ra. Việc này có thể được thực hiện bằng phần mềm trên máy tính, hoặc thông qua một thiết bị mạng như bộ định tuyến. Trong phần này, việc sinh luồng được chia thành 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".

#### Sinh theo phần cứng

Trong nhiều tình huống, 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. Để cấu hình sinh NetFlow trên một bộ định tuyến của Cisco IOS, có thể tham khảo ở tài liệu sau của Cisco: [http://www.cisco.com/en/US/docs/ios/netflow/command/reference/nf\\_cr\\_book.pdf](http://www.cisco.com/en/US/docs/ios/netflow/command/reference/nf_cr_book.pdf).

#### Sinh theo phần mềm

Đa số các cài đặt NSM đều dựa trên sinh theo phần mềm. Việc sử dụng phần mềm cho sinh luồng 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 việc 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. Trong hầu hết các cấu hình, giao diện này sẽ tương tự như các giao diện mà các phần mềm thu thập dữ liệu và phát hiện xâm nhập khác sử dụng.

Hai 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 thông qua hầu hết các hệ thống quản lý gói phần mềm, như yum hay apt. YAF là một công cụ tạo luồng IPFIX. YAF được tạo ra bởi nhóm CERT NetSA (CERT Network Situation Awareness), để tạo ra các bản ghi IPFIX dùng cho SiLK.

### 2.3.3 Thu thập và phân tích luồng dữ liệu với SiLK

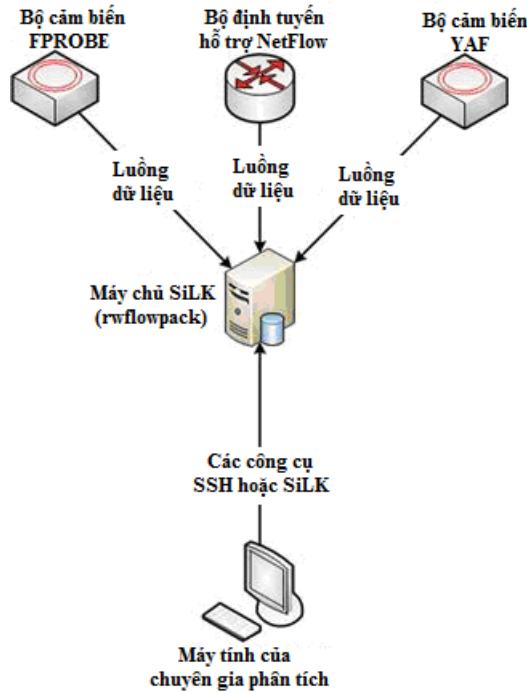
SiLK (System for Internet-Level Knowledge) là một bộ công cụ cho phép phân tích an toàn quản lý hiệu quả trên mạng. Đây là một bộ thu thập luồng, có thể dễ dàng, nhanh chóng lưu trữ, truy cập, phân tích, và hiển thị dữ liệu luồng. SiLK được phát triển bởi nhóm CERT NetSA. Nó có khả năng phân tích luồng nhanh chóng và hiệu quả, mà không cần lập kịch bản phức tạp, tốn tài nguyên CPU. SiLK là một tập hợp bao gồm các ngôn ngữ C, Python, và Perl, và do vậy, nó hoạt động trong hầu hết các môi trường dựa trên UNIX. Phần sau sẽ trình bày một kịch bản cơ bản sử dụng SiLK.

#### Bộ công cụ đóng gói SiLK

Các công cụ SiLK hoạt động thông qua 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. Thuật ngữ "đóng gói" (packing) đề cập đến khả năng của SiLK trong việc nén dữ liệu luồng sang một định dạng nhị phân hiệu quả về không gian để phân tích thông qua bộ phân tích SiLK. Bộ phân tích là một bộ công cụ thu thập dữ liệu dùng để lọc, hiển thị, sắp xếp, đếm,... dữ liệu. Đây là một bộ các công cụ dòng lệnh cung cấp một mức độ không giới hạn về tính linh hoạt. Trong khi chính mỗi công cụ hoạt động riêng cũng đã rất hiệu quả, thì chúng còn có thể được kết hợp với nhau theo dạng chuỗi liên tiếp với đầu ra của công cụ này sẽ là đầu vào hợp lý của một công cụ khác.

Để sử dụng các đặc tính của bộ thu thập và phân tích dữ liệu của SiLK, cần phải lấy được dữ liệu từ một bộ sinh luồng. Khi bộ thu thập nhận được luồng dữ liệu từ bộ sinh, thì các dữ liệu sẽ được tách ra một cách hợp lý theo các loại luồng. Các loại luồng sẽ được phân tích dựa trên một tệp tin cấu hình để xác định khi có dữ liệu bên ngoài-di vào-bên trong, dữ liệu bên trong-di ra-bên ngoài, dữ liệu bên trong đi đến nhau trong mối liên hệ theo kiến trúc mạng.

Trong SiLK, quá trình thu thập sử dụng một công cụ được gọi là rwflowpack. Rwflowpack phụ trách phân tích các loại luồng, xác định dữ liệu có được từ những cảm biến nào, và đặt các dữ liệu luồng đã tinh lọc vào cơ sở dữ liệu để có thể được phân tích bởi bất kỳ công cụ nào trong bộ công cụ phân tích. Việc này được thể hiện trong Hình 2.6.



**Hình 2.6 Luồng công việc của SiLK**

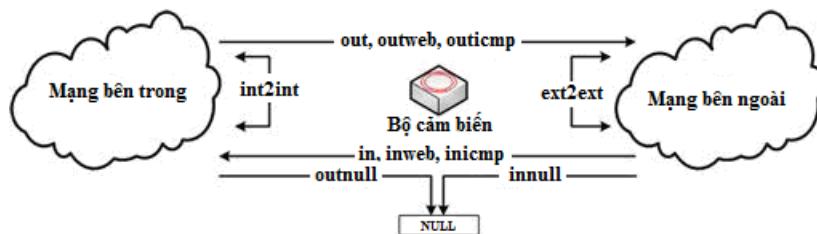
### Các loại luồng SiLK

SiLK phân các luồng thành một số loại để tiện cho việc lọc và sắp xếp các luồng dữ liệu. Việc này được xử lý dựa trên phạm vi mạng được cung cấp cho ipblocks nội mạng và ngoại mạng trong các tệp tin cấu hình sensor.conf được sử dụng bởi rwflowpack.

Có các loại luồng sau:

- In: di chuyển (inbound) đến một thiết bị trong nội bộ mạng (mạng bên trong)
- Out: di chuyển (outbound) đến một thiết bị bên ngoài mạng
- Int2int: từ một mạng nội bộ đến cùng mạng nội bộ, hoặc một mạng nội bộ khác
- Ext2ext: từ một mạng bên ngoài đến cùng mạng bên ngoài, hoặc một mạng bên ngoài khác
- Inweb: di chuyển (inbound) đến một thiết bị trong mạng nội bộ bằng cách sử dụng cổng 80, 443, hoặc 8080.
- Outweb: di chuyển (outbound) tới một thiết bị bên ngoài mạng bằng cách sử dụng cổng 80, 443, hoặc 8080.
- Inicmp: di chuyển (inbound) đến một thiết bị trong mạng nội bộ bằng cách sử dụng ICMP (giao thức IP 1)
- Outicmp: di chuyển (outbound) đến một thiết bị bên ngoài mạng bằng cách sử dụng ICMP (giao thức IP 1)

- Innull: di chuyển (inbound) luồng đã được lọc hoặc luồng bên trong mạng (inbound traffic) tới các null-ipblock cụ thể trong sensor.conf
- Outnull: di chuyển (outbound) luồng đã được lọc hoặc luồng bên ngoài mạng (outbound traffic) tới các null-ipblock cụ thể trong sensor.conf
- Khác: Nguồn không phải bên trong, cũng không phải bên ngoài, hoặc đích đến cũng không phải bên trong hoặc bên ngoài mạng.



**Hình 2.7 Các loại luồng SiLK**

### Các công cụ phân tích SiLK

Khi làm việc với dữ liệu luồng, thời gian phần lớn là làm trên các công cụ phân tích. Hiện có hơn 55 công cụ phân tích trong cài đặt của SiLK, tất cả đều rất hữu ích và một số được sử dụng khá thường xuyên. 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 rwfilter. Rwfilter 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. Do kích thước của luồng dữ liệu sẽ liên quan đến thời gian lưu trữ chúng, nên cần phải có cách thuận tiện để đưa dữ liệu vào bộ lọc nhằm lấy ra được dữ liệu phù hợp với từng nhiệm vụ cụ thể. 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. Rwidssqlery 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. Ngoài ra, thư viện PySiLK cho phép gọi các lời gọi hàm API từ Python.

Để có thể cài đặt SiLK trên Security Onion, xem thông tin tại <http://www.appliednsm.com/silk-on-security-onion/>.

### Lọc luồng dữ liệu với Rwfilter

Phạm vi rộng và tốc độ thu thập luồng dữ liệu làm cho SiLK trở nên mạnh mẽ trong vấn đề thu thập dữ liệu trong bất kỳ môi trường nào. Với SiLK, hầu như các chuyên gia phân tích có thể tập trung vào phạm vi của sự cố mạng nhanh hơn bất kỳ loại dữ liệu nào khác.

Tình huống ví dụ là cần kiểm tra mức độ quấy rối gây ra bởi một máy chủ vi phạm với một địa chỉ IP duy nhất, đây là hành động đầu tiên trong hầu hết các cuộc điều tra. Việc thu hẹp từ dữ liệu PCAP có thể vô cùng tốn kém thời gian. Với SiLK, quá trình này có thể bắt đầu bằng cách sử

dụng các lệnh rwfilter cùng với ít nhất một đầu vào, một đầu ra và một tùy chọn phân vùng. Đầu tiên là tùy chọn địa chỉ IP nào đó (any-address option), địa chỉ này sẽ được truy vấn đến tập dữ liệu trong tất cả các luồng để so khớp với địa chỉ IP đã xác định. Việc này có thể được kết hợp với tùy chọn ngày bắt đầu (start-date) và ngày kết thúc (end-date) để thu hẹp khung thời gian quan tâm. Thêm vào đó, có thể cung cấp rwfilter với type = tùy chọn all (muốn cả luồng đi vào (inbound) và luồng đi ra (outbound)), và pass = tùy chọn stdout (cho phép vượt qua đầu ra rwcutf (thông qua biểu tượng số 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 | rwcutf
```

(--start-date=2013/06/22:11 sẽ hiển thị toàn bộ dữ liệu lưu lượng lọc phù hợp lúc 11 giờ của ngày hôm đó).

Giả sử có một sự kiện là một IP đáng ngờ (6.6.6.6) đột nhiên nhận được lượng đáng kể dữ liệu mã hóa từ một máy chủ web an toàn ngay sau nửa đêm. Cách dễ nhất để đánh giá mức độ lưu lượng truy cập đáng ngờ là để chạy truy vấn SiLK diện rộng:

```
rwfilter --start-date=2013/06/22:00 --any-address=6.6.6.6 --type=all --pass=stdout | rwcutf
```

Nếu dữ liệu bị trại rộng quá, thì cách đơn giản là thêm vào trong các tùy chọn phân vùng "--aport = 443" để các bộ lọc trước thu hẹp tìm kiếm chỉ đến những sự kiện liên quan đến tương tác giữa các IP đáng ngờ và các máy chủ web an toàn. Lệnh --aport sẽ lọc dựa trên các cổng phù hợp với giá trị được cung cấp, trong trường hợp này là cổng 443 (cổng phổ biến nhất liên quan đến truyền thông HTTPS).

```
rwfilter --start-date=2016/06/22:00 --any-address=6.6.6.6 --aport=443 --type=all  
--pass=stdout | rwcutf
```

Sau khi xem xét dữ liệu, có thể nhận thấy các máy chủ web vi phạm đang liên lạc với một số máy chủ trên mạng nội bộ. Giả sử chúng ta không muốn truyền thông xảy ra từ một máy chủ nội bộ cụ thể (192.168.1.100) đến IP đáng ngờ, thì thay vì sử dụng các tùy chọn --any-address, có thể sử dụng các tùy chọn --saddress và --daddress, cho phép lọc địa chỉ nguồn và địa chỉ đích cụ thể tương ứng. Lệnh này như sau:

```
rwfilter --start-date=2013/06/22:00 --saddress=192.168.1.100 --daddress=6.6.6.6  
--aport=443 --type=all --pass=stdout | rwcutf
```

#### 2.3.4 Thu thập và phân tích luồng dữ liệu với Argus

Argus cũng là một 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. Argus bắt đầu được sử dụng vào năm 1989, trở thành bộ phân tích luồng dữ liệu mạng thời gian thực đầu tiên của thế giới. Đến năm 1991, Argus chính

thúc được hỗ trợ bởi CERT, và từ đó Argus có bước phát triển nhanh chóng cho đến năm 1995 khi nó được công bố công khai.

Argus xác định chính bản thân nó như là một giải pháp luồng hoàn toàn bao gồm nhiều luồng dữ liệu, nhưng thay vào đó, nó sẽ 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. Mặc dù Argus có nhiều đặc điểm như các giải pháp phân tích luồng IPFIX khác, nhưng nó lại có công cụ phân tích thống kê và kỹ thuật phát hiện/cảnh báo riêng (khác so với các giải pháp khác).

## Kiến trúc của Argus

Mặc dù Argus được đóng gói trong bộ công cụ Security Onion, nhưng điều quan trọng là phải hiểu được quy trình làm việc chung đằng sau việc thu thập và xác minh dữ liệu. Phần này trình bày kiến thức về việc triển khai Argus và cách thức mà các thành phần của nó làm việc với nhau để thực hiện phân tích luồng dữ liệu.

Nếu triển khai độc lập, Argus sẽ bao gồm hai phần chính nằm trong hai gói. Thành phần thứ nhất được gọi là "Argus" chung, có nhiệm vụ ghi lại lưu lượng dữ liệu thu được vào ổ đĩa qua một giao diện mạng của một thiết bị nào đó. Thành phần này có thể ghi dữ liệu vào ổ đĩa để truyền đi liên tục hoặc duy trì một kết nối socket đến máy chủ an toàn trung tâm để có thể truyền dữ liệu đi liên tục. Thông thường thành phần này sẽ nằm trên một cảm biến và sẽ truyền dữ liệu về một máy chủ nhật ký trung tâm.

Thành phần thứ hai của Argus là Argus Client (Máy khách Argus). Thành phần này, khi triển khai một cách chính xác, sẽ đọc từ các tệp tin nhật ký, thư mục, hoặc một kết nối socket liên tục để phân tích thời gian thực. Ngoài chức năng thu thập dữ liệu từ các bộ sinh bên ngoài, các công cụ của máy khách còn phục vụ như là công cụ phân tích chính trong suốt thời gian sử dụng Argus. Và do vậy, cần phải có các công cụ máy khách trên tất cả các thiết bị dùng Argus để phân tích luồng.

## Đặc điểm

Argus có nhiều tính năng được xây dựng hơn so với hầu hết các công cụ phân tích luồng khác. Khi triển khai Argus độc lập, nó vẫn có thể làm nhiều việc khác ngoài truy vấn luồng cơ bản và thống kê. Ví dụ, một phần của dữ liệu ứng dụng có thể được lấy ra bằng cách thu thập dữ liệu luồng IPFIX và đưa vào trong Argus, nên nó có thể thực hiện một số các nhiệm vụ như lọc dữ liệu dựa trên HTTP URL.

## Thu thập dữ liệu cơ bản

Thu thập dữ liệu trong các công cụ phân tích luồng khác nhau có thể tương tự nhau, nhưng cú pháp truy vấn và khả năng tạo ra số liệu thống kê của các công cụ là khác nhau. Ngay từ đầu, Argus có một cú pháp truy vấn cơ bản, nhưng độ phức tạp của cú pháp cũng dần tăng lên.

Công cụ hữu ích nhất trong bộ công cụ Argus Client là *ra*. Công cụ này 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. Các thông tin này có thể được cung cấp thông qua một tệp tin Argus từ các tùy chọn -r, từ đầu vào chuẩn, hoặc từ một nguồn cấp dữ liệu từ xa. Có thể tham khảo các thư mục lưu trữ cho các tệp tin Argus trong /nsm/sensor\_data/<interface>/argus/.

Một ví dụ lệnh dùng *ra* để xử lý chuẩn đầu vào và xuất chuẩn đầu ra vào một tệp tin như sau:

```
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
```

### 2.3.5 Lưu trữ dữ liệu phiên

So với các loại dữ liệu khác, dữ liệu phiên là khá nhỏ. Tuy nhiên, dữ liệu sẽ ngày càng tăng dần kích thước đến một mức không thể quản lý nếu không được kiểm soát. Không có khuyến nghị cụ thể về số lượng luồng dữ liệu cần lưu trữ, vì việc này phụ thuộc vào tầm quan trọng của dữ liệu đối với tổ chức và băng thông mà tổ chức có. Có thể không cần thiết hoặc không được phép thu thập dữ liệu FPC cho các giao thức nhất định, như đường hầm mã hóa GRE hoặc lưu lượng HTTPS, nhưng vẫn nên giữ các luồng dữ liệu về các phiên liên lạc.

Để ước lượng không gian lưu trữ cần thiết cho dữ liệu lưu lượng, đội CERT NetSA cung cấp một bảng tính dự phòng mà SiLK có thể hỗ trợ, tại: <http://tools.netsa.cert.org/releases/SiLK-Provisioning-v3.3.xlsx>.

Có nhiều cách để quản lý các bản ghi nhật ký mạng, nhưng cách đơn giản và dễ nhất để duy trì là 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. Có nhiều công cụ thu thập dữ liệu có tính năng rollover trong quá trình tạo ra dữ liệu, tuy nhiên vẫn có thể phải quản lý việc này bằng tay. Một giải pháp hạn chế dữ liệu là làm sạch các thư mục luồng dữ liệu định kỳ bằng cách xóa các tệp tin cũ hơn *X* ngày. Ví dụ cụ thể SiLK thực hiện việc định kỳ này như sau:

```
30 12 * * * find /data/silk/* -mtime + 29 -exec rm {}
```

Các tệp tin trong thư mục /data/silk/ sẽ bị xóa nếu đã được lưu từ 30 ngày trở lên. Việc này được thực hiện hàng ngày vào lúc 12:30 PM. Tuy nhiên, cần chú ý đảm bảo các tệp tin cấu hình không có trong các thư mục này.

Có thể lưu trữ dự phòng luồng dữ liệu, như di chuyển dữ liệu đến một thiết bị lưu trữ USB bên ngoài, như sau:

```
*/30 * * * * rsync --update -vr /data/silk/ /mnt/usb/data/silk/ &> /dev/null
```

Lệnh này sẽ sao chép tất cả các luồng tệp tin mới theo chu kỳ 2 phút.

Nhìn chung, các phương pháp định kỳ này được dùng tùy thuộc vào hệ điều hành hiện đang sử dụng.

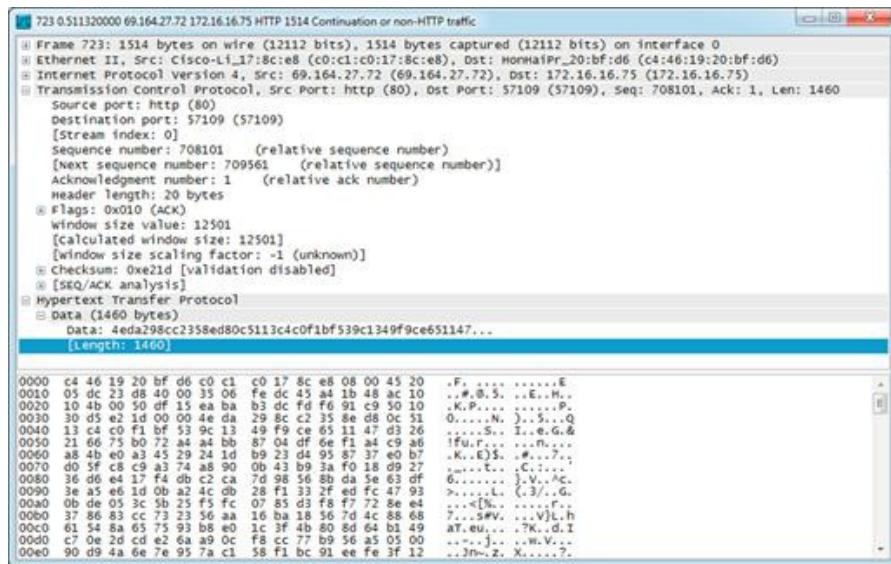
## 2.4 DỮ LIỆU BẮT GÓI TIN ĐẦY ĐỦ

Loại dữ liệu NSM có giá trị nội tại nhất với các chuyên gia phân tích là dữ liệu bắt gói tin đầy đủ (Full Packet Capture Data – FPC). Dữ liệu FPC cung cấp thông tin đầy đủ về tất cả các gói dữ

liệu được truyền giữa hai điểm đầu cuối. Phần này sẽ trình bày về tầm quan trọng của dữ liệu gói tin đầy đủ, trong đó sẽ xem xét một số công cụ bắt gói tin đầy đủ của dữ liệu PCAP như NetSniff-NG, Daemonlogger, và Dumpcap. Việc này sẽ hỗ trợ cho việc lập kế hoạch lưu trữ và duy trì dữ liệu FPC, bao gồm cả vấn đề "cắt tia" bớt số lượng dữ liệu FPC được lưu trữ.

### 2.4.1 Giới thiệu một số công cụ

Định dạng phổ biến nhất của dữ liệu FPC là PCAP (Hình 2.8). Định dạng PCAP được hỗ trợ bởi hầu hết các công cụ thu thập dữ liệu, phát hiện xâm nhập và phân tích dữ liệu mã nguồn mở và là "chuẩn vàng" cho dữ liệu FPC trong một khoảng thời gian. Có một số thư viện hiện có giúp tạo phần mềm sinh và tương tác với các tệp tin PCAP, nhưng phổ biến nhất là Libpcap. Đây là thư viện bắt gói tin mã nguồn mở cho phép ứng dụng tương tác với card giao diện mạng để bắt gói tin. Nhiều ứng dụng được dùng trong việc thu thập và phân tích gói tin sử dụng libpcap, như Dumpcap, Tcpdump, Wireshark,...



Hình 2.8 Ví dụ dữ liệu PCAP được nhìn thấy trong Wireshark

#### 2.4.1.1 Dumpcap

Một trong những cách dễ nhất để có được dữ liệu gói tin đầy đủ là sử dụng Dumpcap. Công cụ Dumpcap cũng đi kèm với Wireshark. Dumpcap là một công cụ đơn giản được thiết kế chỉ với mục đích bắt gói tin từ một giao diện mạng và ghi chúng vào đĩa. Dumpcap sử dụng thư viện libpcap để bắt các gói tin và viết chúng ở định dạng PCAP-NG.

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
```

Lệnh này sẽ bắt đầu bắt gói tin và ghi chúng vào một tệp tin có tên ngẫu nhiên trong thư mục làm việc hiện tại, và sẽ tiếp tục làm như vậy cho đến khi dừng lại. Dumpcap cung cấp một số tùy chọn hữu ích khác trong việc lưu trữ và bắt gói tin:

- -a <value>: Chỉ ra khi nào cần dừng ghi gói tin bắt được vào tệp tin.
- -b <options>: Yêu cầu Dumpcap ghi vào nhiều tệp tin theo các tiêu chí nhất định.
- -B <value>: Chỉ định kích thước bộ đệm, là dữ liệu được lưu trữ trước khi ghi vào đĩa.
- -f <filter>: các lệnh Berkeley Packet Filter (BPF) để lọc các tệp tin ghi các gói tin bắt được.
- -i <interface>: Bắt các gói tin từ giao diện mạng xác định
- -P: Lưu tệp tin định dạng PCAP thay vì PCAP-NG. Hữu ích khi yêu cầu khả năng tương thích ngược với các công cụ không hỗ trợ PCAP-NG.
- -w <filename>: Được sử dụng để xác định tên tệp tin đầu ra.

Dumpcap đơn giản và có những hạn chế. Đầu tiên, nó không phù hợp trong tình huống cần hiệu suất cao khi mức thông lượng cao, có thể dẫn đến các gói tin bị mất. Ngoài ra, sự đơn giản của công cụ này làm hạn chế tính linh hoạt của nó. Điều này là hiển nhiên do các tùy chọn cấu hình của công cụ tương đối ít.

Dumpcap là một giải pháp FPC khá tốt nếu chỉ cần một công cụ chạy nhanh chóng và ít mất công. Tuy nhiên, nếu yêu cầu mức độ linh hoạt cao hơn hoặc bắt gói tin trong trường hợp kết nối thông lượng cao, nên tìm một công cụ khác.

#### **2.4.1.2 Daemonlogger**

Daemonlogger được thiết kế bởi Marty Roesch, và cũng là nhà phát triển ban đầu của Snort IDS. Đây là một ứng dụng ghi log gói tin được thiết kế đặc biệt để sử dụng trong môi trường NSM. Công cụ này 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.

Lợi ích lớn nhất Daemonlogger cung cấp là sự đơn giản trong việc bắt gói tin, giống như Dumpcap. Để bắt đầu, chỉ cần gọi lệnh và chỉ định một giao diện.

```
daemonlogger -i eth1
```

Tùy chọn này, theo mặc định, sẽ bắt gói tin và ghi chúng vào thư mục làm việc hiện tại. Các gói tin sẽ được thu thập cho đến khi kích thước tệp tin đạt đến 2 GB, và sau đó một tệp tin mới sẽ được tạo ra. Việc này sẽ tiếp tục vô thời hạn cho đến khi quá trình này phải dừng lại.

Daemonlogger cung cấp một vài tùy chọn hữu ích để tùy biến cách lưu trữ gói tin. Một số tùy chọn này là:

- -d: Chạy như một daemon
- -f <filename>: Nạp Packet Filters Berkeley (BPF) từ các tệp tin xác định
- -g <group>: Chạy dưới quyền nhóm xác định

- **-i <interface>**: Bắt các gói tin từ giao diện xác định
- **-l <thư mục>**: Ghi dữ liệu vào một thư mục xác định
- **-M <pct>**: Trong chế độ đệm vòng (ring buffer), ghi dữ liệu với tỷ lệ phần trăm công suất nhất định. Để kích hoạt chế độ đệm vòng, cũng sẽ cần xác định tùy chọn **-r**.
- **-n <prefix>**: Thiết lập một tiền tố đặt tên cho các tệp tin đầu ra (hữu ích cho việc xác định tên cảm biến)
- **-r**: Kích hoạt chế độ đệm vòng
- **-t <value>**: chuyển qua các tệp tin log khác theo khoảng thời gian quy định
- **-u <user>**: Chạy với quyền của một người dùng xác định

Ví dụ:

```
daemonlogger -i eth1 -d -f filter.bpf -l /data/pcap/ -n NYC01
```

Khi lệnh này được gọi, Daemonlogger sẽ được thực thi như một daemon (-d) mà các bản ghi của các gói tin được bắt từ giao diện eth1 (-i eth1) được ghi lại trong tệp tin trong thư mục /data/pcap (-l /data/pcap). Những tập tin này sẽ được thêm vào phía trước với chuỗi NYC01 để chỉ ra các cảm biến chúng được thu thập (-n NYC01). Dữ liệu thu thập được sẽ được lọc dựa trên các mô tả BPF chứa trong tệp tin filter.bpf (-f filter.bpf).

Daemonlogger cũng có thiếu sót như Dumpcap khi nói đến hiệu suất. Trong khi 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.

Daemonlogger hiện lại nổi trội hơn do cung cấp một chế độ đệm vòng để loại bỏ sự cần thiết trong việc bảo trì các tệp tin PCAP một cách thủ công. Bằng cách xác định **-r -M <pct>** trong Daemonlogger, có thể yêu cầu công cụ này tự động loại bỏ các dữ liệu cũ khi việc lưu trữ PCAP vượt quá tỷ lệ quy định. Trong một số trường hợp tính năng này là rất cần thiết.

#### **2.4.1.3 Netsniff-NG**

Netsniff-NG là một công cụ bắt gói hiệu suất cao được thiết kế bởi Daniel Borkmann. Trong khi các tiện ích đã thảo luận trên đây đều dựa trên libpcap để bắt gói tin, thì Netsniff-NG sử dụng cơ chế zero-copy để làm việc này. Điều này được thực hiện với mục đích hỗ trợ bắt gói tin đầy đủ trên các liên kết thông lượng cao.

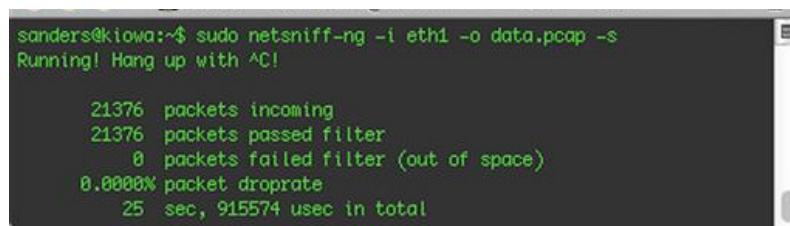
Một trong những tính năng thú vị của Netsniff-NG là nó không chỉ cho phép bắt gói với cơ chế RX\_RING zero-copy, mà còn truyền gói tin với TX\_RING. Điều này có nghĩa là nó có khả năng đọc các gói tin từ một giao diện và chuyển hướng chúng vào một giao diện khác. Tính năng này được thực hiện mạnh mẽ hơn với 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. Trong hầu hết các trường hợp, đầu vào sẽ là một giao diện mạng, và đầu ra sẽ là một tệp tin hoặc thư mục trên đĩa.

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

Lệnh này sẽ bắt gói tin từ giao diện eth0 (-i eth1) và ghi vào tệp tin data.pcap trong thư mục hiện tại (-o data.pcap) cho đến khi ứng dụng dừng. Nếu thực hiện lệnh này, cũng sẽ nhận thấy là màn hình bị lấp đầy với các nội dung của gói tin đang bắt. Để ngăn chặn việc này, cần đặt NetSniff-NG ở chế độ im lặng với cờ -s.

Khi chấm dứt quá trình bắt gói tin (có thể được thực hiện bằng cách nhấn tổ hợp phím Ctrl + C), NetSniff-NG sẽ tạo ra một số liệu thống kê cơ bản liên quan đến dữ liệu bắt được. Những thống kê này được trình bày trong Hình 2.9.



```
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
```

Hình 2.9 Đầu ra tiến trình NetSniff-NG

NetSniff-NG cung cấp rất nhiều tính năng. Một số tùy chọn của NetSniff-NG:

- -g <group>: Chạy với quyền nhóm xác định
- -f <tên file>: Nạp Packet Filters Berkeley (BPF) từ một tệp tin xác định
- -F <value>: Kích thước hoặc khoảng thời gian sử dụng để xác định thời điểm kết thúc bắt gói tin ở chế độ một tệp tin, hoặc chuyển qua các tệp tin kế tiếp trong chế độ đệm vòng.
- -H: Thiết lập tiến trình với độ ưu tiên cao
- -i <interface>: bắt gói tin từ giao diện xác định
- -o <file>: dữ liệu đầu ra ghi vào một tệp tin
- -t <type>: Chỉ xử lý các gói tin có kiểu được định nghĩa (host, host, multicast, outgoing)
- -P <tiền tố>: Thiết lập một tiền tố đặt tên cho các tệp tin đầu ra (hữu ích cho việc xác định tên cảm biến)
- -s: Chạy âm thầm. Không in các gói đã bắt ra màn hình.
- -u <user>: Chạy với quyền người dùng xác định

Ví dụ một lệnh như sau:

```
netsniff-ng -i eth1 -o / data / -F 60 -P "NYC01"
```

Lệnh này sẽ chạy NetSniff-NG ở chế độ đệm vòng, được ghi vào thư mục đầu ra thay vì một tệp tin trong tham số -o (-o /data/). Từ đó sẽ tạo ra một tệp tin PCAP mới mỗi chu kỳ 60 giây (-F 60), và tất cả các tệp tin sẽ được bắt đầu bằng tên cảm biến NYC01 (-P "NYC01").

Trong nhiều thử nghiệm, NetSniff-NG là công cụ FPC tốt nhất trong các công cụ được giới thiệu ở đây trong trường hợp liên kết có thông lượng cao. NetSniff-NG thực hiện rất tốt, là công cụ FPC chuẩn mực, và được kèm mặc định trong bộ công cụ SO.

#### 2.4.2 Lựa chọn công cụ thu thập

Phần trên đã mô tả ba giải pháp FPC cùng hiệu suất tổng thể của mỗi công cụ. Trong khi Dumpcap và Daemonlogger thường làm việc tốt trong hầu hết các tình huống có ít hoặc không mất gói tin, thì thực sự cần một công cụ như NetSniff-NG để hoạt động trong môi trường có tỷ lệ lưu lượng rất cao. Nếu không mở rộng quy mô các công cụ thu thập để đáp ứng yêu cầu về thông lượng, chúng ta sẽ lãng phí chu kỳ CPU và thu thập dữ liệu không đầy đủ. Không có gì khó chịu hơn cho một chuyên gia phân tích là phải cố gắng gộp lại một luồng dữ liệu mà chỉ để thấy rằng có hiện tượng mất mát gói tin và như vậy mọi nỗ lực đều là vô nghĩa.

Lịch sử của công cụ thu thập FPC chủ yếu xoay quanh việc tạo ra dữ liệu "tốt nhất". Ở đây không phải là có nhiều tính năng mà là giải pháp FPC mới được tạo ra để đáp ứng các yêu cầu mới hơn, các mạng nhanh hơn. Điều này không có nghĩa là các công cụ thu thập FPC tốt nhất là các công cụ có thể xử lý được dữ liệu nhanh nhất, mà là công cụ làm mất mát gói tin ít nhất trên cảm biến và cũng có đủ các tính năng để đảm bảo dữ liệu được lưu trữ theo một định dạng mà có thể truy nhập được bằng các công cụ phát hiện và phân tích hiện dùng.

Ba công cụ đề cập trên đây đều đã được chứng minh thực hiện tốt công việc trong các môi trường mạng khác nhau, và cũng là một trong số các giải pháp miễn phí nổi tiếng nhất và đã được triển khai rộng rãi. Với ý nghĩ đó, cần chọn công cụ phù hợp nhất cho tổ chức dựa trên các tiêu chí quan trọng nhất đối với chính tổ chức.

#### 2.4.3 Lập kế hoạch thu thập

Việc thu thập dữ liệu FPC nên được ưu tiên cao khi xây dựng kiến trúc cảm biến vì rất nhiều lý do. Một lý do trong đó là 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. Điều này có nghĩa là số lượng không gian đĩa cứng được dùng cho dữ liệu FPC sẽ vượt qua bất kỳ loại dữ liệu khác.

Một việc quan trọng phải lưu ý khi triển khai một giải pháp FPC là thông lượng, hoặc tỷ lệ trung bình của lưu lượng mạng qua giao diện đang theo dõi. Cần phải có một công 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.

##### a. Những cân nhắc khi lưu trữ

Khi tạo dữ liệu FPC lưu trữ, PCAP chiếm rất nhiều không gian so với tất cả các loại dữ liệu khác, vì vậy việc xác định số lượng dữ liệu FPC cần lưu trữ là rất quan trọng. Việc này được bắt đầu bằng cách lựa chọn một chiến lược duy trì dựa trên thời gian hoặc kích thước.

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 (giới hạn bởi một RAID array 10 TB). Với cả hai chiến lược, nên cố gắng xác định mức hoạt động tối thiểu và lý tưởng.

Khi lên kế hoạch cho thời gian lưu trữ dựa trên lý thuyết, việc đo thông lượng trung bình trên một giao diện có thể cho phép xác định cần bao nhiêu dữ liệu. Ví dụ, nếu xác định rằng giao diện mạng trung bình 100 MB/s và có 1 TB không gian ổ đĩa cứng dành riêng, thì có thể lưu trữ dữ liệu FPC (về mặt lý thuyết) hơn 24 ngày. Tuy nhiên, đây là một cái bẫy do chỉ dựa vào các phép đo thông lượng trung bình. Điều này là do việc đo lường này không tính đến các “gai” lưu lượng, là khoảng thời gian ngắn thông lượng có thể tăng đáng kể. “Gai” thông lượng có thể xảy ra do kết quả của các sự kiện thường xuyên theo lịch như sao lưu trang web hoặc cập nhật ứng dụng, hoặc các sự kiện ngẫu nhiên cao hơn so với hoạt động duyệt web trung bình. Vì FPC là một kiểu dữ liệu sơ cấp, các “gai” cũng có thể dẫn đến sự gia tăng của các dữ liệu khác có nguồn gốc từ nó. Sự kết hợp này làm tăng ảnh hưởng của các “gai” dữ liệu.

Do tính chất của lưu lượng mạng, rất khó để dự đoán mức thông lượng cao điểm. Vì vậy, kế hoạch duy trì dựa trên thời gian buộc phải lựa chọn một khoảng thời gian ngắn hơn đáng kể so với phần cứng có thể cung cấp.

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. Với phương pháp này, 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.

## b. Tính thông lượng giao diện cảm biến với NetSniff-NG và IFPPS

Phương pháp đầu tiên liên quan đến việc sử dụng một công cụ được gọi là ifpps, là một phần của NetSniff-NG. Trong SO, NetSniff-NG được cài đặt mà không có ifpps, vì vậy cần phải cài đặt nó bằng tay theo các bước sau:

1. Cài đặt libncurses-dev trước với APT  
`sudo apt-get install libncurses-dev`
2. Tải NetSniff-NG với GIT  
`git clone https://github.com/borkmann/net-sniff-NG.git`
3. Cấu hình, Make, và cài đặt chỉ ifpps  
`./configure`  
`make && sudo make install ifpps_install`

Sau khi cài đặt ifpps, chạy với tham số -h để xem các tham số, hoặc chỉ cần chạy lệnh sau để tạo ra một danh sách cập nhật liên tục dữ liệu thông kê mạng:

```
ifpps -d <interface>
```

Ifpps sẽ 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. Một ví dụ về đầu ra của lệnh này được thể hiện trong Hình 2.10.

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	288325912727 pkts	97888250 drops	7887 errors
TX:	0.000 Mib	3 pkts	0 drops	0 errors
SYS:	644657 cs/t	98.9% use	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	0.8% iow/t
CPU2:	29.2% usr/t	11.7% sys/t	59.8% idl/t	0.8% 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	0.8% iow/t
CPU6:	30.8% usr/t	1.8% sys/t	67.3% idl/t	0.8% iow/t
CPU7:	30.3% usr/t	1.8% sys/t	67.9% idl/t	0.8% iow/t

Hình 2.10 Tạo thống kê mạng với ifpps

Ifpps có giới hạn là không cung cấp bất kỳ chức năng nào để áp dụng một bộ lọc tới giao diện đang bắt gói tin, vì vậy nếu muốn giảm bớt FPC thì ifpps có thể không phải là công cụ tốt nhất.

### c. Tính thông lượng giao diện cảm biến với dữ liệu phiên

Có lẽ cách linh hoạt nhất để tính toán thống kê thông lượng là tham khảo dữ liệu phiên. Xét một ví dụ về tính thông lượng sử dụng công cụ rwfilter, rwcoun, và rwstats trong SiLK.

Bắt đầu, sử dụng rwfilter để chọn một khoảng thời gian cụ thể, ví dụ như trong 1 ngày:

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

Bộ lọc cơ bản này sẽ chọn tất cả các dữ liệu phiên đã thu thập trong ngày 4/10 và lưu nó vào tệp tin daily.rw.

Để xác định có bao nhiêu dữ liệu, dùng công cụ rwcoun:

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

Lệnh này sẽ cung cấp dữ liệu cho rwcoun, để tạo ra bản tóm tắt lượng dữ liệu mỗi phút đi qua cảm biến. Khoảng thời gian này được xác định bởi thiết lập --bin-size, cho rwcoun biết cần nhóm vào bin theo chu kỳ 60 giây. Kết quả được thể hiện trong Hình 2.11.

Date	Records	Bytes	Packets
2013/10/04T00:00:00	44313.25	859658297.69	1830381.17
2013/10/04T00:01:00	48862.78	1067105604.84	1293719.18
2013/10/04T00:02:00	49592.41	1099875881.23	1347906.89
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	49644.98	1212336398.59	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	1962618085.70	2171138.86
2013/10/04T00:10:00	45727.29	1767951516.52	2001511.11
2013/10/04T00:11:00	48706.03	1888903256.50	2133804.36
2013/10/04T00:12:00	45489.29	1749556534.00	1947954.12
2013/10/04T00:13:00	49425.38	1676261168.77	1941307.40

**Hình 2.11 Thông lượng dữ liệu trong một phút với Rwcount ngoài lúc cao điểm**

Hình 2.11 cho thấy khoảng 1,5 GB dữ liệu trong một phút đi qua các cảm biến ngoài khoảng thời gian cao điểm, lúc 00:00 UTC (là 08:00 EST). Tuy nhiên, nếu nhìn vào Hình 2.12 sẽ thấy lưu lượng truy cập cao đến 8-9 GB mỗi phút trong giờ cao điểm, 17:00 UTC (là 03:00 EST).

Date	Records	Bytes	Packets
2013/10/04T17:00:00	245936.33	9047223014.56	11452368.02
2013/10/04T17:01:00	235940.36	8622123641.52	11005963.02
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	842777053.55	10875324.91
2013/10/04T17:05:00	242852.48	8284062189.96	10692829.75
2013/10/04T17:06:00	253800.12	8766096605.52	11278246.92
2013/10/04T17:07:00	238887.81	8478673022.73	10834914.19
2013/10/04T17:08:00	238807.17	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	248204.78	8730988787.45	11069068.46
2013/10/04T17:12:00	255120.64	8979467783.83	11467693.72
2013/10/04T17:13:00	239651.87	8738665238.16	11891686.19
2013/10/04T17:14:00	232305.00	8500362290.98	10798754.86

**Hình 2.12 Thông lượng mỗi phút với Rwcount lúc cao điểm**

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
```

Kết quả có được thể hiện trong Hình 2.13.

Date	Records	Bytes	Packets
2013/10/04T00:00:00	155413868.66	4915977947888.87	601884467.93

**Hình 2.13 Tính thông lượng trung bình mỗi ngày**

Sử dụng tính toán này sẽ tính được tổng dữ liệu là 4578.36 GB.

#### 2.4.4 Giảm tải cho lưu trữ dữ liệu

Do lượng dữ liệu quá lớn sẽ gây ảnh hưởng tới hệ thống lưu trữ, từ đó cần hạn chế lượng dữ liệu thu thập. Có một số cách giảm tải dữ liệu như sau.

##### a. Loại bỏ dịch vụ

Phương pháp đơn giản nhất để giảm bớt số lượng dữ liệu FPC thu thập là loại bỏ lưu lượng được tạo ra bởi các dịch vụ riêng lẻ. Có thể xác định các dịch vụ thích hợp trong chiến lược này nhờ sử dụng rwstats.

Đầu tiên sử dụng rwstats để xác định cổng chịu trách nhiệm về lưu lượng đi vào nhiều nhất trong mạng.

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

Lệnh này lấy tập dữ liệu ban đầu đưa tới rwstats, tính nhóm 5 cổng nguồn (--top --count = 5) (--fields = sport) có lượng truyền dữ liệu lớn nhất, theo byte (- -value== byte). Kết quả được thể hiện trong Hình 2.14.

sPort	Bytes	%bytes	cumul_%
80	2201459528713	44.718522	44.718522
443	8060164087491	16.372712	61.091234
445	7468447688871	15.154500	76.245735
1935	1500085926771	3.847143	79.292878
25873	827464347761	1.680039	80.973717

Hình 2.14 Nhóm cổng nguồn có lượng dữ liệu trao đổi cao nhất

Có 44% lưu lượng quan sát được là HTTP. Tuy nhiên, trong các dòng tiếp theo, hơn 16% của tổng số dữ liệu xuất phát từ cổng nguồn 443. Để có bức tranh hoàn chỉnh, hãy nhìn nhóm 5 cổng đích có lưu lượng cao nhất để hiểu về lưu lượng đi ra ngoài:

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

Đầu ra của lệnh này được thể hiện trong Hình 2.15.

dPort	Bytes	%bytes	cumul_%
443	2197523546731	4.463857	4.463857
1508	1655943924881	3.363748	7.827597
80	1377874599801	2.797269	10.624865
25	913824775221	1.856264	12.481129
10001	38103250351	0.773996	13.255125

Hình 2.15 Nhóm 5 cổng đích có lưu lượng cao

Hơn 4% lưu lượng đi tới TCP/443, dẫn đến kết luận rằng vào một ngày nào đó, sẽ có khoảng 20,9% lưu lượng truy cập qua các giao diện giám sát của cảm biến thuộc TCP/443. Do đó, lọc lưu lượng TCP/443 ra sẽ giúp tiết kiệm không gian lưu trữ dữ liệu FPC.

Thông thường các tổ chức sẽ loại bỏ các lưu lượng dữ liệu mã hoá thuộc HTTPS. Hãy xem tổng lưu lượng khi loại bỏ TCP/443 như thế nào:

```
cat daily.rw | rwfilter --input-pipe = stdin --aport = 443 --fail = stdout| rwcount --bin-size = 86400
```

Số liệu thống kê cho thấy tổng số dữ liệu đi qua các cảm biến dựa trên bộ lọc mới (Hình 2.16).

Date	Records	Bytes	Packets
2013/10/04T00:00:00	105255664.53	3890585380766.37	4449389688.93

**Hình 2.16 Thông kê thông lượng cho cùng ngày không bao gồm lưu lượng TCP/443**

Số liệu thống kê là 2.52 GB mỗi phút, hoặc 42,9 MB mỗi giây, nghĩa là giảm được ~20,9% trong một ngày.

### b. Loại bỏ lưu lượng host tới host

Một cách khác để giảm số lượng dữ liệu FPC được lưu trữ là loại bỏ các liên lạc giữa các host cụ thể. Ví dụ sau xem xét nhóm các địa chỉ IP nguồn và đích sau khi loại bỏ cổng 443:

```
cat daily.rw | rwfilter --input-pipe = stdin --aport = 443 --fail = stdout| rwstats --fields = sip,dip --top --count = 5 --value = bytes
```

Lệnh này sẽ gửi dữ liệu hiện có tới rwfilter để loại bỏ bất kỳ lưu lượng nào dùng cổng TCP/443. Những thông tin này sau đó được đưa tới rwstats, để tạo ra nhóm các cặp IP nguồn và đích (--fields = sip,dip) theo tổng số byte (--value = byte). Kết quả được thể hiện trong Hình 2.17.

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	0.750252	24.004255	
247.76.249.129	141.239.146.71	22853505245	0.586415	24.590678	
200.7.214.240	141.239.24.254	22529483409	0.578101	25.168771	

**Hình 2.17 Xác định các cặp IP có lưu lượng lớn nhất**

19% lưu lượng trên đoạn mạng này là giữa hai máy tính có địa chỉ 141.239.24.49 và 200.7.118.91. Để xác định xem liệu lưu lượng này có thể loại trừ khỏi dữ liệu FPC hay không, cần phải xem xét kỹ hơn với lệnh:

```
cat daily.rw | rwfilter --input-pipe = stdin --saddress = 141.239.24.49 --daddress = 200.7.118.91 --pass = stdout| rwstats --fields = sport --top --count = 10 --value = bytes
```

Các kết quả của truy vấn này được thể hiện trong Hình 2.18.

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

Hình 2.18 Kiểm tra lưu lượng giữa các máy tính

Có vẻ như tất cả các lưu lượng này xuất hiện trên cổng 22. Giả sử đây là một kết nối hợp pháp, thì điều này thể hiện một hình thức SSH VPN tồn tại giữa hai thiết bị. Nếu không thể giải mã và theo dõi lưu lượng truy cập này (chẳng hạn như thông qua một proxy trung gian), thì đây có lẽ là lưu lượng có thể loại trừ. Quá trình này có thể được lặp lại để giảm bớt lượng dữ liệu thu thập. Sử dụng chiến lược như trên có thể giảm số lượng dữ liệu được lưu trữ FPC khoảng 40%.

#### 2.4.5 Quản lý dữ liệu thu thập

Việc quản lý dữ liệu FPC chủ yếu là thanh lọc dữ liệu cũ. Ở đây cũng theo hai chiến lược tương tự như sử dụng để lưu trữ dữ liệu FPC: dựa trên thời gian và kích thước. Các phương pháp để quản lý hai chiến lược sẽ khác nhau.

##### a. Quản lý dữ liệu thu thập dựa trên thời gian

Chiến lược duy trì dựa trên thời gian khá dễ dàng cho quản lý tự động. Tiện ích find trong linux có thể dễ dàng tìm kiếm các tệp tin theo thời gian thay đổi nhất định. Ví dụ, để tìm tệp tin cũ hơn 60 phút trong thư mục /data/pcap/, chỉ cần chạy lệnh sau:

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

Từ đó có thể tạo ra một danh sách tệp tin PCAP muôn xóa. Lệnh này có thể được sửa đổi bằng cách ghép nối nó với xargs để xóa dữ liệu đáp ứng tiêu chí này.

```
find /data/pcap -type f -mtime + 60 | xargs -i rm {}
```

##### b. Quản lý dữ liệu thu thập dựa trên kích thước

Quản lý dữ liệu FPC sử dụng chiến lược duy trì dựa trên kích thước có khó khăn hơn. Phương pháp này 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. Tùy thuộc vào việc triển khai thu thập FPC, phương pháp này có thể có những khó khăn khác nhau, tuy nhiên có thể sử dụng Daemonlogger để thực hiện điều này thuận tiện hơn.

## 2.5 DỮ LIỆU KIỂU CHUỖI TRONG GÓI TIN

### 2.5.1 Định nghĩa

Dữ liệu kiểu chuỗi trong gói tin (Packet String Data - PSTR) là một thuật ngữ được xác định theo cách chọn để sử dụng nó. Đơn giản, nó là một lựa chọn dữ liệu mà con người có thể đọc được, lấy từ dữ liệu FPC. Dữ liệu này có thể xuất hiện dưới nhiều hình thức khác nhau. Ví dụ, tạo ra dữ liệu PSTR với định dạng cụ thể để diễn tả tiêu đề dữ liệu từ các giao thức tầng ứng dụng phổ biến (như HTTP hoặc SMTP), mà không có tải dữ liệu. Một ví dụ của loại hình dữ liệu PSTR được thể hiện trong Hình 2.19.

```

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.appliednsn.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.

```

**Hình 2.19 Log dữ liệu kiểu PSTR cho HTTP request và response**

Hình 2.20 là ví dụ chỉ có một trường duy nhất được lưu trữ.

```

sonders@osprey:~/ch6$ sudo justniffer -f packets.pcap -p "tcp port 80" -u -l "%request.time% %source.ip% -> %dest.ip% - %request.header.host% %request.url%"
estamp - %source.ip% -> %dest.ip% - %request.header.host% %request.url%
09/22/13 23:41:02 - 10.10.10.3 -> 67.205.2.30 - www.appliednsn.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.appliednsn.com/
09/22/13 23:41:47 - 10.10.10.3 -> 67.205.2.30 - www.appliednsn.com/contributors
09/22/13 23:41:57 - 10.10.10.3 -> 67.205.2.30 - www.appliednsn.com/about-the-book

```

**Hình 2.20 Log dữ liệu kiểu PSTR chỉ ra một HTTP URL được yêu cầu**

Trong ví dụ này, dữ liệu PSTR chỉ chứa các yêu cầu HTTP URL. Trong khi nhiều tổ chức lựa chọn lưu trữ dữ liệu PSTR cho phân tích hồi cứu, ví dụ này đại diện cho dữ liệu được thu thập trên cơ sở thời gian thực. Điều này cho phép dữ liệu được sử dụng nhiều hơn, bao gồm cả việc sử dụng hiệu quả hơn theo cơ chế phát hiện danh tiếng tự động (được thảo luận trong chương sau).

Một hình thức khác của dữ liệu PSTR là tập trung vào tải của gói tin sau tiêu đề của giao thức ứng dụng. Những thông tin này bao gồm một số lượng giới hạn các byte không phải là nhị phân từ tải của gói tin, có thể cho biết mục đích của gói tin. Hình 2.21 là một ví dụ của kiểu dữ liệu.

Các số liệu trong hình 2.21 thể hiện bản sao của dữ liệu có thể đọc, lấy từ trình duyệt web của người dùng. Cụ thể, có thể xem nội dung của trang web được truy cập mà không cần quá nhiều chi tiết bổ sung. Cách này hiệu quả cho việc lưu trữ dữ liệu vì không cần phải lưu trữ ký tự không đọc được. Bất lợi của việc sử dụng dữ liệu PSTR kiểu tải là chi phí cần thiết để tạo ra nó. Ngoài ra cũng cần có một lượng hợp lý các dữ liệu thừa đi cùng với nó.

```

16:15:31.686876 IP 69.172.216.55.80 . 192.168.146.136.50505: tcp 1831
E.E.7.P.lqN.I.XAP.400
var adsafeVisParams = {
    mode : 'jss',
    jsrcref : 'http://imp.bid.ace.advertising.com/site.850222.size.160600.u.2.bnum.99795581.vkhr.160
    .hr.16.hi.2.scores.5.svh.1440x900.title.2.f.2.r.i.optn.1.fv.11.dolexp
    .1.togs.1.dref.http.253A.252F.252Fwww.autoblog.com.252F.
    adsafeSrc : 'http://pixel.adsafeprotected.com/rfv.st.19024.1214081.skeleton.js',
    adsafeSep : '',
    requrl : '',
    requrey : '',
    debug : false,
    allowEngagement : true,
    trackMouse : true,
    jsFeatures : 'mousetrack.viewabilityready.consecutive.cachebust:8.forcecocoa:10.rattie:100.ex
    ch.recordalternate:100.cocoapuffs.nextcocoa.usedtdomain:8',
    engagementDelay : 1-5-15,
    useAdTalk : true,
    adTalkDtCall : true,
    killPhrases : '',
    asid : '8d5913c6-1d93-11e3-bcb0-0025904ea2d8',
    adWidth : 160,
    adHeight : 600,
    adHeight : 600,
    minimizeCalls : false,
    exclList : 'e1::nqzryq.e2::tbbtyrnqf.t.qphoyrpyvpx.e3::ehovpbacebwpg.e4::chozngvp.e5::b
    crak.e6::nqoevgr.pbz.e7::tynz.pbz.e8::lvryqzqnanre.pbz.e9::yvvvg.e10::'
}
16:15:31.686885 IP 192.168.146.136.50505 . 69.172.216.55.80: tcp 8
E.(.K.-p.E.7.I.P.XAqN.sP.Dwq.

```

**Hình 2.21 Dữ liệu PSTR kiểu tải của gói tin**

### 2.5.2 Thu thập dữ liệu

Đầu tiên, cần xem xét mức độ của các dữ liệu PSTR muôn thu thập. Giải pháp lý tưởng là tập trung vào việc thu thập dữ liệu tầng ứng dụng cần thiết, càng nhiều từ các giao thức văn bản rõ ràng tốt. Vì có nhiều biến thể của dữ liệu PSTR có thể được thu thập nên không lưu trữ dữ liệu sẽ biến đổi rất lớn. Vì vậy, nên sử dụng một số phương pháp thảo luận ở phần trước để xác định có bao nhiêu không lưu trữ để sử dụng cho dữ liệu PSTR.

Song song với việc xác định loại dữ liệu PSTR sẽ tạo ra, cũng 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, trong khi 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 lập đầy khoảng trống giữa FPC và dữ liệu phiên.

Khi đánh giá các nhu cầu lưu trữ dữ liệu PSTR, nên chú ý là sẽ có sự biến đổi rất lớn. Ví dụ, trong thời gian ăn trưa, lưu lượng HTTP là ở đỉnh cao và lưu lượng được tạo ra từ các giao thức khác có liên quan chặt chẽ tới các quy trình kinh doanh giảm xuống. Điều này có thể không ảnh hưởng đến tổng lượng dữ liệu PCAP được thu thập trong khoảng thời gian này, nhưng nó sẽ làm tăng lượng dữ liệu PSTR được thu thập.

Có một số ứng dụng mã nguồn mở miễn phí có thể thực hiện cả hai việc: thu thập dữ liệu PSTR từ mạng và tạo ra từ dữ liệu FPC. Phần sau sẽ xem xét một số công cụ.

### a. Tạo dữ liệu PSTR thủ công

Trước khi xem xét một số công cụ có thể được sử dụng để tự động tạo ra dữ liệu PSTR, hãy tìm hiểu một số phương pháp khác nhau để tạo ra dữ liệu PSTR bằng cách sử dụng công cụ trong môi trường BASH Linux. Xét các dữ liệu ASCII từ một tệp tin PCAP. Với dữ liệu PSTR, dữ liệu duy nhất cần quan tâm là tập hợp các ký tự đọc được, vì vậy chỉ cần giới hạn kết quả bằng dữ liệu thông qua chuỗi các tiện ích Linux.

Script theo kiểu log dưới đây sẽ tạo ra dữ liệu tương tự như trong Hình 2.20, với các bản ghi theo dòng mô tả chi tiết URI gắn với yêu cầu của người sử dụng.

```
#!/bin/bash
#Send the ASCII from the full packet capture to stdout
/usr/sbin/tcpdump -qnnS 0 -A -r test.pcap |
#Normalizes the PCAP
strings |
#Parse out all timestamp headers and Host fields
grep -e '[09][09]:[09][09]:[09][09].[09]\'7b6\'7dHost:'| grep -B1 "Host:" |
#Clean up the results
grep -v -- "--" | sed 's/Host.*$/g'
tr "" "-" | sed 's/-//g'| sed 's/-Host:/ -/g'
```

Các giải pháp thủ công tuy chậm trong xử lý dữ liệu nhưng sẽ rất linh hoạt.

### b. URLSnarf

URLSnarf thu thập dữ liệu yêu cầu HTTP một cách thụ động và lưu chúng dưới định dạng log chung (common log format - CLF). URLSnarf có trong bộ công cụ dsniff, không được cài đặt mặc định trên SO, vì vậy nếu muốn sử dụng nó, cần phải cài đặt:

```
sudo apt-get install dsniff
```

Khi chạy không có tham số, URLSnarf sẽ thụ động lắng nghe trên một giao diện và xuất dữ liệu thu thập được tới đầu ra chuẩn. Mặc định, nó sẽ lắng nghe trên giao diện eth0 và bắt lưu lượng trên cổng TCP 80, 8080 và 3128.

URLSnarf chỉ có 4 tùy chọn:

- -p: Cho phép người dùng chạy URLSnarf trên một tệp tin PCAP đã có gói tin thu thập
- -i: Xác định một giao diện mạng
- -n: Phân tích dữ liệu mà không phân giải địa chỉ DNS
- -v <expression>: Mặc định, có thể xác định một URL cụ thể, được coi là một biểu thức trong thời gian chạy để chỉ hiển thị các URL phù hợp với biểu thức đó. Tùy chọn -v cho phép xác định một biểu thức để hiển thị tất cả kết quả không khớp với URL đã nêu.

Do đầu ra là định dạng log chuẩn, phân tích đầu ra bằng cách chuyên (pipe) qua các công cụ dòng lệnh BASH như grep, cut, và awk để hơn là chỉ định biểu thức với tùy chọn -v. Trong Hình 2.22 dưới đây, đầu tiên sẽ bắt lưu lượng truy cập bằng tcpdump và sau đó truyền qua URLsnarf với tùy chọn -p.

```
sanders@osprey:~/ch6$ urlsnarf -p packets.pcap
urlsnarf: using packets.pcap [tcp port 80 or port 8080 or port 3128]
10.10.10.3 - [22/Sep/2013:23:41:02 +0000] "GET http://www.appliedns.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 librtmp/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 librtmp/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 librtmp/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 librtmp/2.3"
10.10.10.3 - [22/Sep/2013:23:41:42 +0000] "GET http://www.appliednsm.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 librtmp/2.3"
10.10.10.3 - [22/Sep/2013:23:41:47 +0000] "GET http://www.appliednsa.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 librtmp/2.3"
10.10.10.3 - [22/Sep/2013:23:41:57 +0000] "GET http://www.appliednsa.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 librtmp/2.3"
```

*Hình 2.22 Dữ liệu mẫu từ URLsnarf*

Đầu ra thể hiện trong Hình 2.22 là một tập hợp các log tiêu chuẩn mô tả chi tiết các yêu cầu HTTP khi truy cập trang web appliednsm.com.

### c. Httpry

Httpry là một công cụ bắt gói tin chuyên để hiển thị và ghi lại lưu lượng HTTP. Httpry chỉ có thể phân tích lưu lượng HTTP. Tuy nhiên, không giống như URLsnarf, Httpry 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. Khả năng tùy chỉnh đầu ra theo các công cụ khác làm cho httpry hữu ích trong việc tạo dữ liệu PSTR hữu ích. Httpry không có sẵn trong SO, nhưng có thể được cài đặt khá dễ dàng như sau.

1. Cài đặt thư viện phát triển libpcap  
sudo apt-get install libpcap-dev
2. Tải về tarball từ website Httpry của Jason Bittel  
wget <http://dumpsterventures.com/jason/httpry/httpry-0.1.7.tar.gz>
3. Giải nén  
tar -zxf httpry-0.1.7.tar.gz
4. Vào thư mục Httpry và thực hiện cài đặt ứng dụng  
make && sudo make install

Sau khi cài đặt hoàn tất, có thể chạy chương trình không có tham số để bắt đầu thu thập lưu lượng HTTP từ cổng 80 trên giao diện mạng đánh số thấp nhất. Hình 2.23 hiển thị httpry đọc lưu lượng từ một tệp tin bằng cách sử dụng tham số -r và xuất ra đầu ra.

```

monders@osprey:~/ch6$ httpry -r packets.pcap
httpry version 0.1.2 -- 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.appliednames.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 2608:1000:0001:0009:98be:00a1:c57e:6cf0:2607:f008:4000:c83:167 > GET www.google.com HTTP/1.1 -
2013-09-22 23:41:14 2607:f008:4000:c83:167:2608:1000:0001:0009:98be:00a1:c57e:6cf0 < - - - - - HTTP/1.1 200 OK
2013-09-22 23:41:17 10.10.10.3 157.156.248.13 > GET www.cnn.com / HTTP/1.1 -
2013-09-22 23:41:17 157.156.248.13 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.appliednames.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.appliednames.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.appliednames.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

```

**Hình 2.23 Ví dụ dữ liệu Httpry**

Httpry cung cấp một số tham số dòng lệnh, và sau đây là một vài tham số hữu ích nhất để bắt đầu:

- **-r <file>**: Đọc từ một tệp tin PCAP đầu vào thay vì thực hiện bắt gói tin trực tiếp
- **-o <file>**: Ghi ra một tệp tin httpry log (cần thiết cho các script phân tích)
- **-i <interface>**: Bắt dữ liệu từ một giao diện xác định
- **-d**: Chạy httpry như một daemon
- **-q**: Chạy trong chế độ im lặng, loại dữ liệu đầu ra không quan trọng như banner và thông kê.

Httpry có sẵn một số kịch bản có thể xử lý đầu ra cho phép phân tích dữ liệu đầu ra tốt hơn. Bằng cách sử dụng tham số **-o**, có thể buộc các dữ liệu được thu thập bởi httpry được xuất ra bởi một trong các plugin. Một số các plugin có khả năng xuất ra các thông kê tên máy, thông tin tóm lược về HTTP log, và khả năng chuyển đổi các định dạng đầu ra thành định dạng log chung, từ đó cho phép tạo ra kết quả tương tự như những gì URLsnarl có.

Khả năng tạo ra kịch bản phân tích cho phép tích hợp các plugin giúp tự động hóa một giải pháp dữ liệu PSTR dựa trên httpry. Việc chuyển đổi đòi hỏi một script gọi là `parse_log.pl`. Script này nằm trong thư mục của httpry `scripts/plugins/`, và hoạt động bằng cách sử dụng các plugin có trong thư mục đó. Ví dụ, các lệnh dưới đây có thể được sử dụng cho một script phân tích. Trong trường hợp này, sử dụng định dạng nhật ký chung khi tạo ra dữ liệu httpry trong một định dạng linh hoạt cho việc phân tích bằng các công cụ phát hiện và phân tích.

1. Chạy Httpry và hướng đầu ra vào một tệp tin  
`httpry -o test.txt`
2. Phân tích đầu ra  
`perl scripts/parse_log.pl -p scripts/plugins/common_log.pm test.txt`

Đầu ra httpry đầu tiên phải được ghi vào một tệp tin với tùy chọn **-o**. Sau đó, script `parse_log.pl` có thể được thực hiện để phân tích dữ liệu. Ví dụ về đầu ra này được thể hiện trong Hình 2.24.

```

10.10.10.3 www.cnn.com - [22/Sep/2013:23:41:17 +0000] "GET / HTTP/1.1" 200 -
10.10.10.3 www.foxnews.com - [22/Sep/2013:23:41:22 +0000] "GET / HTTP/1.1" 200 -
10.10.10.3 www.espn.com - [22/Sep/2013:23:41:27 +0000] "GET / HTTP/1.1" 301 -
10.10.10.3 www.appliednsa.com - [22/Sep/2013:23:41:42 +0000] "GET / HTTP/1.1" 200 -
10.10.10.3 www.appliednsa.com - [22/Sep/2013:23:41:47 +0000] "GET /contributors HTTP/1.1" 301 -
10.10.10.3 www.appliednsa.com - [22/Sep/2013:23:41:57 +0000] "GET /about-the-book HTTP/1.1" 301 -

```

**Hình 2.24 Ví dụ đầu ra Httpry được phân tích thành định dạng chung**

Tạo ra dữ liệu PSTR với httpry thường nhanh hơn đáng kể so với URLsnarf.

### 2.5.3 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. Công cụ này miễn phí và theo mã nguồn mở, vô cùng mạnh mẽ và tương đối dễ dàng thiết lập trong môi trường lớn. Phần sau sẽ trình bày cấu hình Logstash để phân tích log đang thu thập với URLsnarf. Logstash phiên bản 1.2.1 có giao diện Kibana để xem log, vì vậy phần này cũng sẽ thảo luận một số tính năng có thể được sử dụng để truy vấn dữ liệu.

Logstash không có trong SO, cần phải tải về từ trang web của dự án tại [www.logstash.net](http://www.logstash.net). Logstash được chứa hoàn toàn trong một gói java, vì vậy sẽ cần cài đặt Java Runtime Environment (JRE).

Để phân tích bất kỳ loại dữ liệu nào, Logstash đòi hỏi một tệp tin cấu hình để định nghĩa làm thế nào nhận được dữ liệu đó. Ví dụ sau sẽ xem xét việc dữ liệu được ghi vào một vị trí cụ thể. Gọi tệp tin cấu hình là urlsnarf-parse.conf. Đây là một cấu hình rất đơn giản:

```

input {
    file {
        type => "urlsnarf"
        path => "/home/idsusr/urlsnarf.log"
    }
}
output {
    elasticsearch { embedded => true }
}

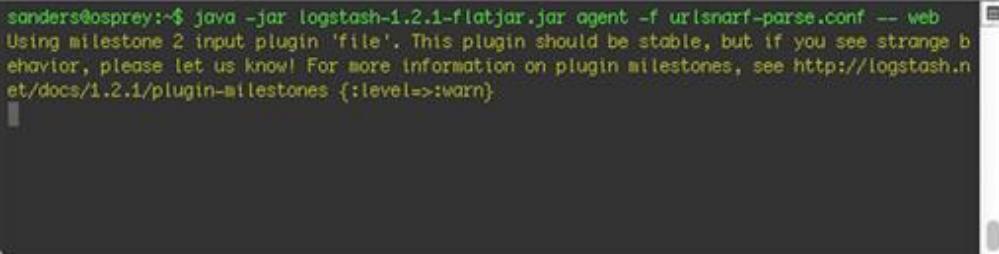
```

Cấu hình này báo cho Logstash cần lắng nghe các dữ liệu được ghi vào tệp tin /home/idsusr/urlsnarf.log và xem xét các log đã ghi vào tệp tin đó là một kiểu log "urlsnarf", loại log đã định nghĩa. Phần đầu ra của tệp tin cấu hình này bắt đầu một thực thể Elasticsearch bên trong Logstash cho phép lập chỉ mục và tìm kiếm các dữ liệu nhận được.

Khi có tệp tin cấu hình, có thể khởi tạo Logstash để bắt đầu nghe log cho đến khi bắt đầu sinh dữ liệu. Để chạy Logstash với Kibana, thực hiện lệnh:

```
java -jar logstash-1.2.1-flatjar.jar agent -f urlsnarf-parse.conf -- web
```

Đầu ra của lệnh này được hiển thị trong Hình 2.25.



```
sanders@osprey:~$ java -jar logstash-1.2.1-flatjar.jar agent -f urlsnarf-parse.conf -- web
Using milestone 2 input plugin 'file'. This plugin should be stable, but if you see strange behavior, please let us know! For more information on plugin milestones, see http://logstash.net/docs/1.2.1/plugin-milestones {:level=>:warn}
```

Hình 2.25 Chạy Logstash

Lệnh này sẽ khởi tạo agent, chỉ ra urlsnarf-parse.conf với tùy chọn -f. Kết thúc lệnh với "-web" sẽ đảm bảo là Kibana được chạy cùng. Việc khởi động lần đầu tiên có thể mất một phút. Sau đó, truy cập <http://127.0.0.1:9292> trong trình duyệt web để vào Kibana.

Nếu đã cài đặt Logstash trên SO và muốn truy cập vào giao diện web của Kibana từ một hệ thống khác thì sẽ không thể được. Điều này là do tường lửa được kích hoạt trên hệ thống. Có thể thêm một ngoại lệ cho tường lửa với lệnh sau:

```
sudo ufw allow 9292/tcp
```

Để tạo dữ liệu phân tích, có thể chạy URLsnarf và xuất dữ liệu đầu ra của nó vào một tệp tin.

```
sudo urlsnarf > /home/idsusr/urlsnarf.log
```

Sau đó, mở trình duyệt web (hoặc sử dụng curl từ dòng lệnh) và đi đến một vài trang web để tạo ra một số dữ liệu. Khi hoàn tất, sử dụng Ctrl + C để kết thúc URLsnarf. Sau khi dừng thu thập dữ liệu, quay trở lại Kibana và xác nhận rằng các bản ghi đang đến trong trình duyệt. Nếu có, sẽ thấy một số dữ liệu hiển thị trên màn hình, tương tự như Hình 2.26.



Hình 2.26 Xem dữ liệu log trong Kibana

Hình 2.26 chỉ mô tả tệp tin log theo mức "thô" và hầu hết là chưa phân tích. Cần phải định nghĩa các bộ lọc tùy chỉnh để tạo ra các thông tin trạng thái, để Kibana thực sự có ích. Logstash 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. GROK là một ngôn ngữ mạnh mẽ được sử dụng bởi Logstash để

làm cho việc phân tích dễ dàng hơn so với lúc sử dụng biểu thức thông thường. Xét một ví dụ trong đó sẽ tạo ra một bộ lọc để so khớp các trường văn bản trong log đã tạo với Justniffer trong Hình 2.27, nhưng có bổ sung “sensor name” ở cuối. (Justsniffer là một công cụ thu thập dữ liệu mạnh, có thể được dùng cho dữ liệu PSTR. Tham khảo chi tiết ở tài liệu [1]).

```

09/22/13 23:41:02 - 10.10.10.3 -> 67.205.2.38 - www.appliednsis.com/ - IDS1 SENSOR
09/22/13 23:41:17 - 10.10.10.3 -> 157.166.240.43 - www.cnn.com/ - IDS1 SENSOR
09/22/13 23:41:22 - 10.10.10.3 -> 23.66.230.66 - www.faxnews.com/ - IDS1 SENSOR
09/22/13 23:41:27 - 10.10.10.3 -> 199.181.132.268 - www.espn.com/ - IDS1 SENSOR
09/22/13 23:41:42 - 10.10.10.3 -> 67.205.2.38 - www.appliednsis.com/ - IDS1 SENSOR
09/22/13 23:41:47 - 10.10.10.3 -> 67.205.2.38 - www.appliednsis.com/contributors/ - IDS1 SENSOR
09/22/13 23:41:57 - 10.10.10.3 -> 67.205.2.38 - www.appliednsis.com/about-the-book/ - IDS1 SENSOR

```

**Hình 2.27 Tuỳ chỉnh dữ liệu Justniffer với một tên cảm biến để được phân tích**

Để cho thấy cách Logstash xử lý việc so khớp cơ bản so với các mẫu xây dựng sẵn, sử dụng một bộ lọc "match" trong cấu hình.

```

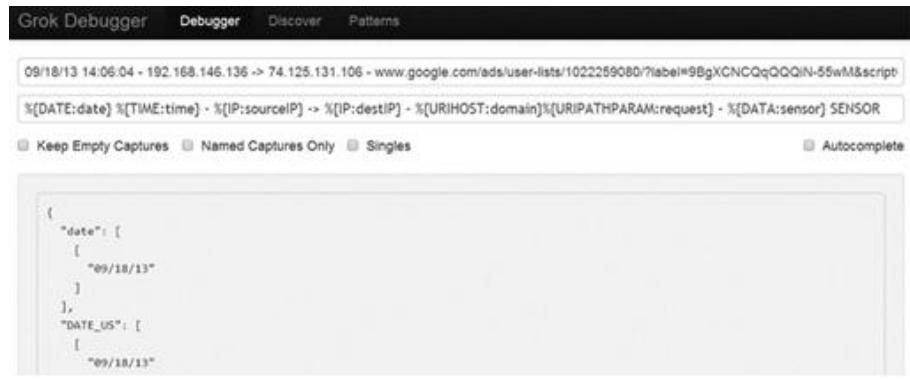
input {
    file {
        type => "Justniffer-Logs"
        path => "/home/idsusr/justniffer.log"
    }
}
filter {
    grok {
        type => "Justniffer-Logs"
        match => [ "message", "insertfilterhere" ]
    }
}
output {
    elasticsearch { embedded => true }
}

```

Dùng các mẫu GROK được xây dựng sẵn để tạo ra các dữ liệu cần cho cấu hình, gọi là justniffer-parse.conf. Những mẫu này có thể được tìm thấy tại <https://github.com/logstash/logstash/blob/master/patterns/grok-patterns>. Nhưng trước khi kiểm tra các mẫu cần gắn với nhau, điều đầu tiên cần làm là nhìn vào định dạng log và định nghĩa những trường muốn xác định. Định dạng dữ liệu này như sau:

datestamp timestamp – IP -> IP – domain/path – sensorname SENSOR

Tiếp theo cần phải dịch định dạng dữ liệu này vào GROK, và cần bộ gỡ lỗi GROK. Bộ gỡ lỗi nằm ở <http://grokdebug.herokuapp.com/>. Ở đây chỉ cần đặt chuỗi log cần so khớp tại dòng trên cùng, và trong dòng mẫu, nhập mẫu GROK phù hợp. Ứng dụng này sẽ cho thấy dữ liệu nào phù hợp. Điều quan trọng khi phát triển các chuỗi định dạng GROK là bắt đầu với các mẫu nhỏ và mở rộng chúng dần dần để phù hợp với toàn bộ dòng log (Hình 2.28).



**Hình 2.28 Sử dụng trình gõ lỗi GROK**

Để phù hợp với dòng log, sẽ sử dụng mẫu:

```
%{DATE:date} %{TIME:time} - %{IP:sourceIP} -> %{IP:destIP} -
%{URIHOST:domain}%{URIPATHPARAM:request} - %{DATA:sensor} SENSOR
```

Trong này đã bao gồm các nhãn trường theo sau mỗi trường. Áp dụng bộ lọc cho toàn bộ tệp tin cấu hình sẽ cho một cấu hình hoàn chỉnh, để phân tích tất cả các log Justniffer đi đến phù hợp với các định dạng xác định trước đó. Đây là tệp tin cấu hình kết quả:

```
input {
    file {
        type => "Justniffer-Logs"
        path => "/home/idsusr/justniffer.log"
    }
}
filter {
    grok {
        type => "Justniffer-Logs"
        match => [ "message", "%{DATE:date} %{TIME:time} -
%{IP:sourceIP} -> %{IP:destIP} -
%{URIHOST:domain}%{URIPATHPARAM:request} -
%{DATA:sensor} SENSOR" ]
    }
}
output {
    elasticsearch { embedded => true }
}
```

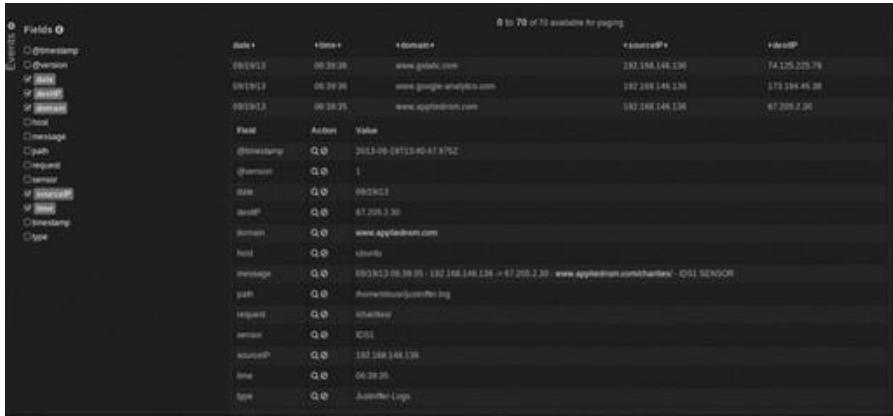
Sau khi có được cấu hình, có thể bắt đầu bộ thu thập dữ liệu Logstash sử dụng tệp tin cấu hình mới:

```
java -jar logstash-1.2.1-flatjar.jar agent -f justniffer-parse.conf --web
```

Khi Logstash chạy, có thể bắt đầu thu thập dữ liệu với lệnh Justniffer để tạo ra dữ liệu log theo định dạng phù hợp với cấu hình vừa tạo:

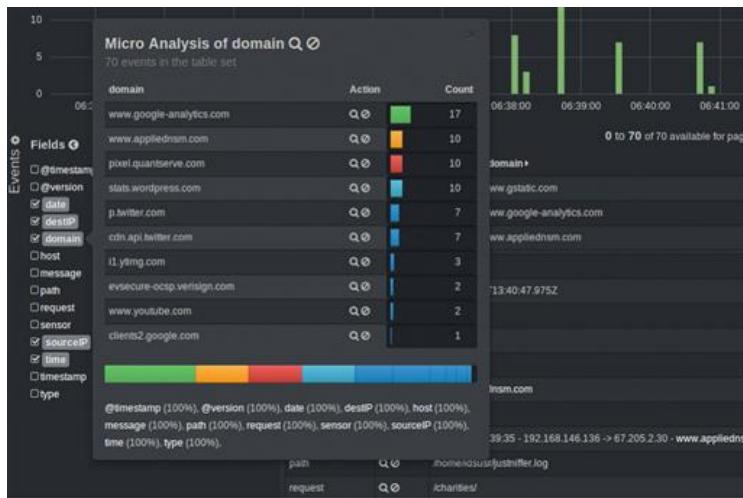
```
sudo justniffer -p "tcp port 80" -u -l "%request.timestamp - %source.ip - > %dest.ip - %request.header.host%request.url - IDS1 SENSOR" >> /home/idsusr/justniffer.log
```

Khi chạy, lại cần phải duyệt một vài trang web để tạo ra log. Sau khi thu thập dữ liệu, kiểm tra lại Kibana và xem liệu log có được hiển thị lên hay không. Nếu mọi thứ chính xác, sẽ phải có được phân tích đầy đủ các log tùy chỉnh.



*Hình 2.29 Kiểm tra bản ghi cá nhân trong Kibana*

Có thể kiểm tra thang đo cho một trường bằng cách nhấp vào tên trường trong danh sách ở phía bên trái màn hình. Hình 2.30 là thang đo trường trong trường Host, dùng để hiển thị tất cả các máy đã truy cập trong các bản ghi hiện hành.



**Hình 2.30 Kiểm tra thang đo trường trong Kibana**

Như đã thấy, sự kết hợp của Logstash, Kibana, và GROK làm thành một bộ ba mạnh mẽ, thuận tiện cho việc phân tích các log giống như log được tạo ra bởi dữ liệu PSTR. Nếu muốn tìm hiểu thêm về những công cụ này, truy cập vào trang web Logstash tại <http://logstash.net/>.

## CHƯƠNG 3

# PHÁT HIỆN XÂM NHẬP

Chương này trình bày các vấn đề liên quan đến bước phát hiện xâm nhập trong chu trình giám sát an toàn mạng, bao gồm một số nội dung sau: các kỹ thuật phát hiện xâm nhập, dấu hiệu tấn công và chữ ký, các phương pháp phát hiện xâm nhập như phương pháp phát hiện xâm nhập dựa trên danh tiếng, phương pháp phát hiện xâm nhập dựa trên chữ ký và phương pháp phát hiện xâm nhập dựa trên dữ liệu bất thường thống kê, và các công cụ thực hành cụ thể là Snort, Suricata và SiLK.

### 3.1 CÁC KỸ THUẬT PHÁT HIỆN XÂM NHẬP, DẤU HIỆU TẤN CÔNG VÀ CHỮ KÝ

#### 3.1.1 Kỹ thuật phát hiện xâm nhập

Phát hiện xâm nhập là một chức năng của phần mềm thực hiện phân tích các dữ liệu thu thập được để tạo ra dữ liệu cảnh báo. Dữ liệu cảnh báo được tạo ra bởi các cơ chế phát hiện được chuyển tới chuyên gia phân tích, và đó là khi việc phân tích bắt đầu. Để phát hiện thành công, cần chú ý đến việc lựa chọn cơ chế phát hiện và điều vào thích hợp.

Phần lớn các cơ chế phát hiện xâm nhập được thảo luận trong tài liệu này là những hệ thống phát hiện xâm nhập dựa trên mạng (NIDS), bao gồm hai loại chính là dựa trên chữ ký và dựa trên phát hiện bất thường.

Phát hiện dựa trên chữ ký là hình thức lâu đời nhất của phát hiện xâm nhập. Phương pháp này thực hiện bằng cách duyệt qua dữ liệu để tìm các kết quả khớp với các mẫu đã biết. Ví dụ đơn giản của mô hình này là một địa chỉ IP hoặc một chuỗi văn bản; ví dụ về mô hình phức tạp hơn là số lượng byte null (byte rỗng) xuất hiện sau một chuỗi xác định khi sử dụng một giao thức nào đó. Khi các mẫu được chia thành các mẫu nhỏ độc lập với nền tảng hoạt động, chúng trở thành dấu hiệu của tấn công. Khi được mô tả bằng ngôn ngữ cụ thể trong nền tảng của một cơ chế phát hiện xâm nhập, chúng trở thành chữ ký. Có hai cơ chế phát hiện dựa trên chữ ký phổ biến là Snort và Suricata (sẽ được giới thiệu trong phần sau).

Một tập con các phát hiện dựa trên chữ ký là phát hiện dựa trên danh tiếng. Cơ chế phát hiện này cố gắng phát hiện thông tin liên lạc giữa các máy tính được bảo vệ trong mạng và các máy tính trên Internet có thể bị nhiễm độc do đã từng tham gia vào các hành động độc hại trước đó. Kết quả phát hiện dựa trên các chữ ký đơn giản như địa chỉ IP hoặc tên miền.

Phát hiện dựa trên bất thường là một hình thức mới của phát hiện xâm nhập, được phổ biến nhanh chóng nhờ các công cụ như Bro. Phát hiện dựa trên bất thường dựa vào quan sát sự cố mạng và nhận biết lưu lượng bất thường thông qua các chẩn đoán và thống kê. Thay vì chỉ đơn giản là cảnh báo bất cứ khi nào phát hiện ra mẫu tấn công, cơ chế phát hiện dựa trên bất thường có khả năng nhận ra các mẫu tấn công khác biệt với hành vi mạng thông thường. Đây là cơ chế

phát hiện rất tốt nhưng khó thực hiện. Ví dụ, Bro là một cơ chế phát hiện bất thường, và thực hiện phát hiện bất thường dựa trên thông kê.

Một tập con mới được phát triển của phát hiện dựa trên bất thường là sử dụng cơ chế phát hiện dựa trên Honeypot. Honeypot đã được sử dụng trong nhiều năm để thu thập phần mềm độc hại và các mẫu tấn công cho mục đích nghiên cứu. Nhưng chúng cũng có thể được ứng dụng tốt trong phát hiện xâm nhập bằng cách cấu hình hệ thống. Honeypot thường chứa các lỗ hổng đã được biết đến, nhưng chúng không có dữ liệu bí mật thực tế. Thay vào đó, chúng được cấu hình cho việc ghi lại dữ liệu, và thường được kết hợp với các loại khác của NIDS hoặc HIDS.

### 3.1.2 Dấu hiệu xâm nhập và chữ ký

Các cơ chế phát hiện sẽ không hiệu quả nếu dữ liệu đầu vào không được chuẩn bị kỹ càng và hợp lý. Điều này liên quan đến sự phát triển, duy trì và thực hiện các dấu hiệu xâm nhập (Indicators of Compromise - IOC) và chữ ký.

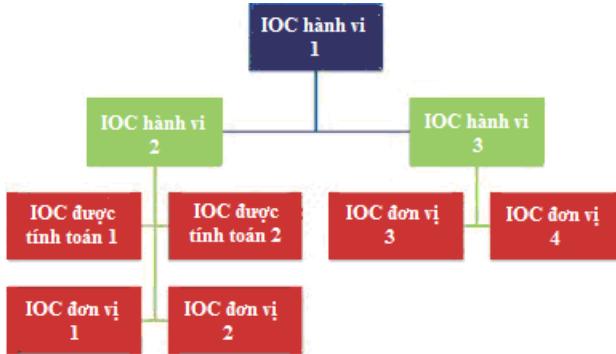
IOC là những thông tin được sử dụng để mô tả khách quan một xâm nhập mạng, độc lập về nền tảng. Các thông tin có thể bao gồm một dấu hiệu đơn giản như địa chỉ IP của máy chủ chỉ huy và kiểm soát (command and control server - C&C hay C2), hoặc một tập hợp phức tạp các hành vi chỉ ra rằng máy chủ thư điện tử đã bị sử dụng như là một SMTP relay độc hại. IOC có thể được trình bày theo nhiều cách thức và định dạng khác nhau để có thể được sử dụng bởi các cơ chế phát hiện khác nhau. Ví dụ, một công cụ có thể có thể phân tích các địa chỉ IP trong một danh sách phân cách bởi dấu phẩy, một công cụ khác có thể yêu cầu chúng phải được đưa vào một cơ sở dữ liệu SQL. Mặc dù biểu diễn của IOC đã được thay đổi, nhưng nó vẫn còn phù hợp. Hơn nữa, một IOC hành vi có thể có được chia nhỏ thành nhiều thành phần riêng lẻ và được triển khai cho nhiều cơ chế phát hiện để hoạt động được trên mạng. Khi một IOC được thực hiện và được sử dụng trong một ngôn ngữ hoặc định dạng cụ thể, chẳng hạn như một luật Snort hoặc một tệp tin theo định dạng Bro, nó sẽ trở thành một phần của một chữ ký. Một chữ ký có thể chứa một hoặc nhiều IOC.

#### IOC cho mạng và máy tính

Có 2 dạng phổ biến nhất của IOC là dựa trên mạng và máy tính. IOC cho máy tính là một mẫu thông tin được tìm thấy trên một máy tính, mô tả khách quan một xâm nhập. Một số IOC cho máy tính thông thường là tài khoản người dùng, đường dẫn thư mục, tên tiến trình, tên tệp tin, khóa đăng ký (registry), ... IOC cho mạng là một mẫu thông tin có thể được bắt trên kết nối mạng giữa các máy chủ, mô tả khách quan một xâm nhập. Một số IOC cho mạng phổ biến là địa chỉ IPv4, địa chỉ IPv6, tên miền, chuỗi văn bản, giao thức truyền thông,...

#### IOC tĩnh

IOC tĩnh là những IOC mà giá trị được định nghĩa một cách rõ ràng. Có ba biến thể của IOC tĩnh là đơn vị (hay còn gọi là nguyên tố), được tính toán, và hành vi (Hình 3.1).



**Hình 3.1 IOC đơn vị, được tính toán và hành vi**

IOC đơn vị là các IOC cụ thể và nhỏ mà không thể chia được tiếp thành các thành phần nhỏ hơn nữa, nhưng vẫn có ý nghĩa trong tình huống một xâm nhập. IOC đơn vị có thể là địa chỉ IP, chuỗi văn bản, tên máy, địa chỉ thư điện tử, và tên tệp tin. IOC được tính toán có nguồn gốc từ dữ liệu sự cố, bao gồm giá trị băm, các biểu thức thông thường, và các thống kê. IOC hành vi là tập các IOC đơn vị và IOC được tính toán được kết hợp với nhau theo một số hình thức logic, dùng để cung cấp cho một số tình huống hữu dụng. IOC hành vi có thể bao gồm một tập các dữ liệu chứa tên tệp tin và các giá trị băm tương ứng, hoặc một sự kết hợp của một chuỗi văn bản và một biểu thức thông thường.

Xem xét một kịch bản đã xác định là có một thiết bị trên mạng bị xâm nhập. Một phân tích từ dữ liệu NSM và dữ liệu phân tích dựa trên máy chủ giúp xác định chuỗi sự kiện xảy ra như sau:

1. Người dùng nhận được một e-mail từ chris@appliednsm.com với chủ đề "Thông tin tiền lương" và một tệp PDF đính kèm là "Payroll.pdf". Tệp PDF có một giá trị băm MD5 là e0b359e171288512501f4c18ee64a6bd.
2. Người dùng mở tệp PDF, kích hoạt việc tải một tệp tin gọi là kerndel32.dll với MD5 là da7140584983eccde51ab82404ba40db. Tệp tin được tải về từ <http://www.appliednsm.com/kernel32.dll>.
3. Tệp tin được dùng để ghi đè lên C:/Windows/System32/kernel32.dll.
4. Mã trong DLL được thực thi, và một kết nối SSH được thiết lập tới một máy chủ có địa chỉ IP là 192.0.2.75 trên cổng 9966.
5. Khi kết nối này được thiết lập, phần mềm độc hại tìm kiếm mọi tệp DOC, DOCX, hoặc PDF từ máy chủ và truyền nó qua kết nối SSH đến máy chủ nguy hiểm.

Trong bối cảnh NSM, các mô tả trên quá phức tạp. Để các cơ chế phát hiện có hiệu quả, đầu tiên cần phải phân tích các dấu hiệu thành các phần nhỏ có ích hơn, mà vẫn đảm bảo phù hợp với bối cảnh. Có thể tạo ra các IOC hành vi (B) như sau:

- B-1: Người dùng nhận được một e-mail từ chris@appliednsm.com với chủ đề "Thông tin tiền lương" và một tệp PDF đính kèm là "Payroll.pdf", có một giá trị băm MD5 là e0b359e171288512501f4c18ee64a6bd.
- B-2: Tệp tin kernel32.dll với hàm băm MD5 da7140584983eccde51ab82404ba40db được tải về từ <http://www.appliednsm.com/kernel32.dll>.
- B-3: Tệp tin C:/Windows/System32/Kernel32.dll bị ghi đè bởi một tệp tin độc hại cùng tên với giá trị hàm băm MD5 da7140584983eccde51ab82404ba40db.
- B-4: Máy tính nạn nhân có gắng kết nối qua SSH tới máy tính nguy hiểm bên ngoài 192.0.2.75 trên cổng 9966.
- B-5: Các tệp tin DOC, DOCX, và PDF được truyền tới 192.0.2.75 trên cổng 9966 thông qua một kết nối được mã hóa.

Tiếp theo, tiếp tục phân tích IOC hành vi thành các IOC đơn vị (A) và IOC được tính toán (C). Sau đây là một kết quả:

- C-1: MD5 Hash e0b359e171288512501f4c18ee64a6bd
- C-2: MD5 Hash da7140584983eccde51ab82404ba40db
- A-1: Tên miền nguy hiểm: appliednsm.com
- A-2: Địa chỉ e-mail: chris@appliednsm.com
- A-3: Tiêu đề thư: "Thông tin tiền lương"
- A-4: Tên file: Payroll.pdf
- A-5: Tên file: Kernel32.dll
- A-6: IP nguy hiểm 192.0.2.75
- A-7: Cổng 9966
- A-8: Giao thức SSH
- A-9: Kiểu file DOC, DOCX, PDF
- A-10: Tên file Kernel32.dll

Tổng cộng có 5 IOC hành vi, 1 IOC được tính toán, và 10 IOC đơn vị có thể được đưa vào hệ thống phát hiện xâm nhập. Từ đó dẫn đến các IOC được chuyển đổi thành các chữ ký để sử dụng trong một loạt các cơ chế phát hiện, chẳng hạn như trong ví dụ sau:

- C-1/2: Chữ ký chống vi-rút để phát hiện sự tồn tại của giá trị băm
- A-1: Chữ ký Snort/Suricata để phát hiện kết nối với tên miền nguy hiểm
- A-2: Chữ ký Snort/Suricata để phát hiện thư nhận được từ địa chỉ e-mail nguy hiểm

- A-3: Chữ ký Snort/Suricata để phát hiện dòng chủ đề
- A-3: Bro script để phát hiện dòng chủ đề
- A-4/C-1: Bro script để phát hiện tên tệp tin hay giá trị băm MD5 được truyền trên mạng
- A-5/C-2: Bro script để dò tìm tệp tin có tên là Kernel32.dll hoặc tệp tin với giá trị băm MD5 truyền qua mạng
- A-6: Chữ ký Snort/Suricata để phát hiện thông tin liên lạc với địa chỉ IP
- A-7/A-8: Chữ ký Snort/Suricata để phát hiện thông tin liên lạc SSH đến cổng 9966
- A-10: Luật HIDS để phát hiện những thay đổi của Kernel32.dll

Như vậy có thể thấy rằng, có nhiều phương pháp khác nhau để phát hiện ra các dấu hiệu khác nhau từ một sự cố đơn lẻ. Nếu chi tiết hơn, tình huống này thậm chí có thể mô tả một kịch bản với nhiều tiềm năng, chẳng hạn như khả năng phát hiện một số lời gọi đổi tượng độc hại ngay chính bên trong tệp tin PDF, hoặc các đặc tính của giao thức sửa đổi có thể được sử dụng. Tùy thuộc vào kiến trúc mạng đang bảo vệ, có thể có nhiều cơ chế phát hiện được dùng để tạo ra chữ ký cho một dấu hiệu riêng lẻ, hoặc không có khả năng phát hiện ra dấu hiệu nào. Quyết định phương pháp nào là tốt nhất cho việc phát hiện một IOC phụ thuộc vào cơ sở hạ tầng của mạng, đặc tính của phương pháp phát hiện, và bản chất của tri thức liên quan đến IOC.

## Các biến IOC

Nếu các cơ chế phát hiện sử dụng trong mạng chỉ được cấu hình để phát hiện các cuộc tấn công đã biết, thì sẽ có khả năng bỏ lỡ phát hiện những nguy hiểm khác. Cần phải coi IOC là các biến, trong đó có những dấu hiệu chưa biết giá trị. Việc này được thực hiện bằng cách tạo ra một chuỗi các sự kiện mà một cuộc tấn công có thể tạo ra (tạo thành một IOC hành vi), và xác định nơi các biến tồn tại. Về cơ bản, cần xem xét một cuộc tấn công trên lý thuyết, chứ không phải một tấn công đã xảy ra. Các biến IOC không hoàn toàn hữu ích cho việc triển khai cơ chế phát hiện dựa trên chữ ký, nhưng lại hữu ích trong các giải pháp như Bro.

Ví dụ về việc tạo các biến IOC như sau: thay vì dựa vào kịch bản là cuộc tấn công đó đã thực sự xảy ra, ở đây sẽ dựa trên một cuộc tấn công lý thuyết (giả định). Kịch bản tấn công ở trên sẽ diễn ra như sau:

1. Người dùng nhận được một e-mail với một tệp tin đính kèm độc hại.
2. Người dùng mở tệp tin đính kèm, kích hoạt việc tải tệp tin từ một tên miền độc hại.
3. Tệp tin được dùng để ghi đè lên một tệp tin hệ thống với phiên bản mã độc của tệp tin đó.
4. Mã trong các tệp tin độc hại thực thi, gây ra một kết nối mã hóa đến một máy chủ độc hại.
5. Sau khi kết nối được thiết lập, một số lượng lớn dữ liệu sẽ bị rò rỉ từ hệ thống.

Các bước này diễn tả các IOC hành vi có chứa nhiều các IOC đơn vị và biến. Một số IOC:

- VB-1: Một người dùng nhận được một e-mail với một tệp tin đính kèm độc hại.

- VA-1: Địa chỉ e-mail
- VA-2: Tiêu đề e-mail
- VA-3: Tên miền nguồn của e-mail độc hại
- VA-4: Địa chỉ IP nguồn của e-mail
- VA-5: Tên tệp tin đính kèm độc hại
- VC-1: Tệp tin đính kèm độc hại với giá trị băm MD5
- VB-2: Người dùng mở tệp tin đính kèm, kích hoạt việc tải một tệp tin từ một tên miền độc hại.
- VA-6: Tên miền/IP chuyển hướng độc hại
- VA-7: Tên tệp tin độc hại đã tải
- VC-2: Giá trị băm MD5 của tệp tin độc hại đã tải
- VB-3: Tệp tin được sử dụng để ghi đè lên một tệp tin hệ thống với phiên bản mã độc của tệp tin đó.
- VB-4: Thực thi mã trong tệp tin độc hại, tạo ra một kết nối mã hóa đến một máy chủ độc hại trên một cổng không chuẩn.
- VA-8: Địa chỉ IP C2 ngoài
- VA-9: Cổng C2 ngoài
- VA-10: Giao thức C2 ngoài
- VB-5: Sau khi kết nối được thiết lập, một số lượng lớn các dữ liệu đã bị rò rỉ từ hệ thống.

Trong ví dụ này, V trong tên IOC mô tả một thành phần biến của IOC. Như vậy, có 10 biến IOC đơn vị, 2 biến IOC được tính toán, và 5 biến IOC hành vi. Bây giờ, có thể đưa ra giả thuyết về các phương pháp mà các IOC này có thể được tạo thành chữ ký để ghép nối với cơ chế phát hiện. Biến IOC thường sẽ được tái sử dụng và kết hợp để có thể dùng phát hiện trong các kịch bản tấn công rộng hơn.

- VB-1 (VA-3/VA-4) VB-2 (VA-6) VB-4 (VA-8) VB-5 (VA-8): Luật Snort/Suricata để phát hiện các liên lạc với danh tiếng xấu theo địa chỉ IP và tên miền.
- VB-1 (VA-5/VC-1) VB-2 (VA-7/VC-2): Bro script để kéo các tệp tin từ đường truyền và so sánh tên của chúng và các giá trị băm MD5 với một danh sách các tên tệp tin danh tiếng xấu được biết đến và các giá trị băm MD5.
- VB-1 (VA-5/VC-1) VB-2 (VA-7/VC-2): Bro script để lấy các tệp tin từ đường truyền và đặt chúng vào trong thử nghiệm phân tích phần mềm độc hại sơ bộ.

- VB-2 (VA-6/VA-7/VC-2): chữ ký HIDS để phát hiện các trình duyệt đang được gọi từ một tài liệu.
- VB-3: chữ ký HIDS để phát hiện một tệp tin hệ thống đang bị ghi đè
- VB-4 (VA-9/VA-10) VB-5: Bro script để phát hiện mã hóa lưu lượng đang xảy ra trên một cổng không chuẩn
- VB-4 (VA-9/VA-10) VB-5: một luật Snort/Suricata để phát hiện mã hóa lưu lượng đang xảy ra trên một cổng không chuẩn
- VB-5: script tự viết sử dụng thống kê dữ liệu phiên để phát hiện khôi lưu lượng lớn lưu lượng gửi đi từ máy trạm.

### 3.1.3 Quản lý dấu hiệu tấn công và chữ ký

Số lượng các IOC và chữ ký được quản lý bởi một tổ chức có thể phát triển nhanh trong một thời gian ngắn. Điều quan trọng là cần phải có chiến lược lưu trữ, truy cập và chia sẻ chúng. Hầu hết các tổ chức chỉ có xu hướng lưu trữ IOC và chữ ký trong cơ chế phát hiện mà họ đang sử dụng. Ví dụ, nếu đang sử dụng Snort để phát hiện và ghi nhật ký các truy cập vào một tên miền độc hại (một IOC đơn vị), thì sau đó các IOC sẽ được lưu thành chữ ký Snort, được truy cập trực tiếp bởi Snort. Đây là cách đơn giản nhất nhưng lại làm hạn chế khả năng tương tác và tham khảo chung. Ngoài ra, cách này cũng ngăn cản sự chia sẻ hoặc chuyển đổi IOC sang chữ ký được thiết kế cho một cơ chế phát hiện khác. Do vậy, để có thể quản lý được các IOC và chữ ký tốt nhất, cần tuân theo các nguyên tắc sau:

**Định dạng dữ liệu thô.** Đơn giản nhất là làm việc với các IOC trong hình thức nguyên bản. Luôn luôn có thể truy nhập vào một IOC mà không cần bất kỳ nội dung bổ sung hoặc xử lý nào không liên quan. Điều này đảm bảo cho IOC tính linh động và có thể được phân tích một cách dễ dàng bằng các công cụ tự động hay tùy chỉnh, cho phép chúng được triển khai thành các chữ ký duy nhất cho các cơ chế phát hiện khác nhau. Ví dụ, các địa chỉ IP và chuỗi băm của tệp tin nên là văn bản bình thường, còn dữ liệu nhị phân nên tồn tại ở dạng nhị phân.

**Dễ dàng tiếp cận.** Các chuyên gia phân tích có thể truy cập và chỉnh sửa IOC và chữ ký dễ dàng. Nếu họ phải trải qua nhiều bước bất tiện để thêm những cái mới hoặc tìm ra nguồn gốc của một IOC, thì sẽ làm lãng phí thêm thời gian của họ và có thể làm nản lòng các chuyên gia phân tích khi tương tác với IOC và chữ ký.

**Dễ dàng tìm kiếm.** Để tạo điều kiện cho các chuyên gia phân tích nhanh chóng kiểm tra IOC và chữ ký, chúng nên tồn tại trong một định dạng dễ tìm kiếm, bao gồm khả năng tìm kiếm các chỉ số hoặc ký tự, cùng với những dữ liệu theo ngữ cảnh được lưu trữ và ngày được thêm vào, mã nguồn, hoặc kiểu. Nếu được lưu trữ trong cơ sở dữ liệu, tìm kiếm có thể được thực hiện với việc truy nhập cơ sở dữ liệu hoặc truy nhập qua một trang web đơn giản. Nếu được lưu trữ trong các tệp tin phẳng, có thể sử dụng các công cụ dòng lệnh Linux như grep.

**Theo dõi sửa đổi.** Sửa đổi lại chữ ký là việc bình thường. Nó có thể xảy ra khi một chữ ký dẫn đến quá nhiều lỗi, hoặc khi nó không phát hiện ra các hoạt động mong muốn, có kết quả âm tính giả. Chữ ký cũng được sửa đổi để phản ánh những thay đổi trong chiến lược hoặc các kỹ thuật tấn công đối lập. Bất cứ khi nào điều này xảy ra, việc sửa đổi, người đã thực hiện các thay đổi, và ngày thay đổi cần phải được ghi lại để tất cả các vấn đề phát sinh từ việc sửa đổi có thể được giải quyết. Trường hợp lý tưởng nhất, lý do của sự thay đổi cũng nên được ghi lại.

**Theo dõi việc triển khai.** Mục đích cuối cùng của một IOC là có thể được sử dụng trong một chữ ký cùng với một cơ chế phát hiện. Khi điều này xảy ra, các cặp IOC và cơ chế phát hiện cần được quan tâm. Điều này sẽ giúp chuyên gia phân tích hiểu cách mà hạ tầng NSM đang được sử dụng với một IOC, đồng thời cũng tránh trùng lặp cho IOC trên một cơ chế phát hiện. Có thể thực hiện việc ánh xạ đơn giản từ một IOC GUID thành một SID.

**Sao lưu dữ liệu.** IOC và chữ ký rất quan trọng cho sự thành công của NSM, và cần được sao lưu phù hợp. Do vậy, cần có một nơi sao lưu dữ liệu trong trường hợp có thảm họa xảy ra ảnh hưởng đến hoạt động của công ty.

Trong thực tế, có thể quản lý các chữ ký và IOC với tệp tin CSV (như Bảng 3.1).

**Bảng 3.1 Danh sách chữ ký/IOC chính**

GUID	Tác giả	Ngày tạo	Ngày sửa đổi	Phiên bản	Nguồn	Phân loại	Loại	Giai đoạn trong chữ ký sống	Độ tin tưởng	IOC	Triển khai
10001	Sanders	3/17/2013	3/20/2013	2	Case # 1492	MD5	Computed/Static	Mature	Very High	e0b359e1712 88512501f4c 18ee64a6bd <a href="http://appliednsm.com">appliednsm.com</a>	Antivirus Signature 42039
10002	Smith	3/18/2013	3/18/2013	1	Malware Domain List	Domain	Atomic/Static	Mature	Moderate		Snort Signature 7100031
10003	Sanders	3/18/2013	3/18/2013	1	Case # 1498	E-Mail Address	Atomic/Static	Mature	Very High	<a href="mailto:chris@appliednsm.com">chris@appliednsm.com</a>	Snort Signature 7100032
10004	Sanders	3/19/2013	3/19/2013	1	Zeus Tracker	IP	Atomic/Static	Mature	High	192.0.2.99	Custom SiLK Script
10005	Randall	3/20/2013	3/24/2013	4	Analyst	Protocol/Port	Behavioral/Variable	Immature	Moderate	Encrypted Traffic over Non-Standard Port	Bro Script
10006	Sanders	3/20/2013	3/20/2013	1	RSS Feed	Protocol/Port	Behavioral/Static	Mature	Moderate	SSH/9966	Suricata Signature 7100038
10007	Sanders	3/21/2013	3/24/2013	3	Internal Discussion	Statistical	Behavioral/Variable	Immature	Low	Outbound Traffic Volume Ratio Greater than 4:1	Custom SiLK Script

### 3.1.4 Các khung làm việc cho dấu hiệu tấn công và chữ ký

Một trong những vấn đề lớn nhất đối với an ninh thông tin và cộng đồng tạo tri thức về các nguy cơ và bảo mật nói chung là thiếu một khung làm việc chung cho việc tạo lập, quản lý và phân phối IOC và chữ ký. Hầu hết những người sử dụng chúng đều có xu hướng sử dụng các phương pháp cá nhân của họ để tổ chức và lưu trữ dữ liệu. Do vậy, IOC và chữ ký không phải là thành phần mở và không thể dễ dàng được chia sẻ với các tổ chức khác. Trong khi việc chia sẻ chính các dữ liệu thường có thể được thực hiện khá dễ dàng, ví dụ như danh sách các địa chỉ IP, thì chia sẻ thông tin theo ngữ cảnh là điều thực sự khó khăn.

Trong những năm gần đây, một số tổ chức đã nỗ lực tạo ra các khung làm việc cho việc chia sẻ dữ liệu IOC và chữ ký.

## OpenIOC

Một trong những cải tiến lớn nhất hướng tới một khung làm việc chung cho tri thức về các nguy cơ và bảo mật (threat intelligence - TI) là dự án OpenIOC của Mandiant. Dự án này ban đầu được thiết kế nhằm cho phép sản phẩm của Mandiant có thể hệ thống hóa một cách thông minh để nhanh chóng tìm kiếm các lỗ hổng bảo mật tiềm tàng, và được phát hành vào năm 2010 để thành một lược đồ nguồn mở cho thông tin về TI.

OpenIOC chỉ là một lược đồ XML được sử dụng để mô tả các đặc điểm kỹ thuật xác định các hoạt động tấn công. OpenIOC cho phép quản lý các IOC với rất nhiều các thông tin theo ngữ cảnh cần thiết để sử dụng hiệu quả các IOC. Một ví dụ về OpenIOC được thể hiện trong Hình 3.2.

```
<?xml version="1.0" encoding="us-ascii"?>
<ioc xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema" id="9ad0ddec-dc4e-4432-9687-
      b7002806dcf8" last-modified="2013-02-20T02:02:40" xmlns="http://
      schemas.mandiant.com/2010/ioc">
    <short_description>PHISH-UPS-218934</short_description>
    <description>Part of the UPS Phishing scheme reported on 12/4.</description>
    <authored_by>Chris Sanders</authored_by>
    <authored_date>2013-02-20T02:02:00</authored_date>
    <links>
        <link rel="Source">http://www.appliednsa.com/</link>
        <link rel="Stage">Mature</link>
    </links>
    <definition>
        <Indicator operator="OR" id="bea0030a-dddc-440b-9dd4-400cacc0e13d">
            <IndicatorItem id="16b2873b-a491-46b1-9f3e-895dd49f3cb0"
            condition="contains">
                <Content document="Email" search="Email/Subject" type="mir" />
                <Content type="string">UPS Alert: Shipment Delayed</Content>
            </IndicatorItem>
        </Indicator>
    </definition>
</ioc>
```

Hình 3.2 Một IOC đơn giản trong định dạng XML OpenIOC

Nếu sử dụng Windows, có thể làm việc với định dạng này trong công cụ OpenIOC Editor miễn phí của Mandiant.

## STIX

STIX (Structured Threat Information eXpression) là một dự án dựa vào cộng đồng mã nguồn mở được phát triển bởi MITRE cho US Department of Homeland Security. STIX được thiết kế để chuẩn hoá thông tin TI, và được phổ biến trong chính phủ và quân đội.

Kiến trúc STIX dựa trên cấu trúc độc lập và các mối liên quan (Hình 3.3).



**Hình 3.3 Kiến trúc STIX**

Cốt lõi của kiến trúc này là các đối tượng quan sát được, được định nghĩa là các thuộc tính có trạng thái hoặc các sự kiện đo được, thích hợp cho các hoạt động của máy tính và mạng. Đây có thể là một dịch vụ đang dừng, tên tệp tin, một sự kiện khởi động lại hệ thống, hoặc một thiết lập kết nối. Các đối tượng được lưu trong định dạng XML, và được mô tả bằng cách sử dụng ngôn ngữ CybOX. Một ví dụ về đối tượng có thể quan sát được thể hiện trong Hình 3.4. Đối tượng này đại diện cho một địa chỉ IPv4, với một vài đối tượng liên quan. Các đối tượng được liên kết với nhau thông qua việc sử dụng các định danh duy nhất toàn cầu GUID.

```

<?xml version="1.0" encoding="UTF-8"?>
<cybox:Observable id="cybox:observable-6f45f992-38c8-11e2-8011-000c291a73d5" type="IP-Address">
    <cybox:Stateful_Measure>
        <cybox:AddressObj id="cybox:addressobj-6ec8ffaa-38c8-11e2-8011-000c291a73d5" category="ipv4-addr">
            <AddressObj>
                <Address_Value>192.0.2.182</Address_Value>
            </AddressObj>
        </cybox:AddressObj>
        <cybox:Related_Objects>
            <cybox:Related_Object idref="cybox:guid-6ec8ffaa-38c8-11e2-8011-000c291a73d5" type="URI" relationship="Resolved_To"/>
            <cybox:Related_Object idref="cybox:guid-6ec1cdf2-38c9-11e2-8011-000c291a73d5" type="DNS_Query" relationship="Contained_Within"/>
            <cybox:Related_Objects>
                <cybox:Related_Object idref="cybox:guid-6ec8ffaa-38c8-11e2-8011-000c291a73d5" type="DNS_Record" relationship="Contained_Within"/>
            </cybox:Related_Objects>
        </cybox:Related_Objects>
    </cybox:Stateful_Measure>
</cybox:Observable>

```

**Hình 3.4 Một đối tượng STIX đại diện cho một địa chỉ IP với các đối tượng liên quan**

Trong khung làm việc STIX, đối tượng có thể quan sát được có thể gắn với các dấu hiệu, sự cố, các nguy cơ cụ thể, chiến dịch thù địch, mục tiêu cụ thể, những mảng dữ liệu, và hàng loạt các hành động. Những thực thể này được kết hợp với nhau để hình thành một hệ thống lớn hơn một hệ thống quản lý đơn giản, hay đúng hơn, chúng tạo thành một hệ thống quản lý TI đầy đủ.

Có thể tìm hiểu thêm về STIX tại <http://stix.mitre.org>.

## **3.2 PHÁT HIỆN XÂM NHẬP DỰA TRÊN DANH TIẾNG**

### **3.2.1 Danh sách danh tiếng công khai**

Trong thực tế, hầu hết các tổ chức thực hiện phát hiện dựa trên danh tiếng bằng cách sử dụng danh sách công khai của các IOC đơn vị (phổ biến nhất là các địa chỉ IP và tên miền) với danh tiếng xấu. Những danh sách đen này sau đó được đưa vào một số loại cơ chế phát hiện để các chuyên gia phân tích thông báo khi một máy tính được bảo vệ xuất hiện để giao tiếp với một thiết bị bên ngoài thuộc một trong những danh sách này.

Ngoài ra còn có một số khía cạnh tiêu cực khi sử dụng danh sách danh tiếng. Trong rất nhiều trường hợp, các nhà bảo trì các danh sách này không thường xuyên cung cấp bối cảnh cho các địa chỉ IP cá nhân, tên miền trên danh sách. Khi một cảnh báo được tạo ra dựa trên thông tin liên lạc với một máy chủ trên một trong những danh sách này, chúng ta không thực sự biết lý do tại sao máy chủ có danh tiếng xấu.

Tuy nhiên, rõ ràng những mặt tích cực của các danh sách công khai lớn hơn những mặt tiêu cực. Do vậy, cần đảm bảo là danh sách đã chọn để kết hợp vào kiến trúc phát hiện phải phù hợp với mục tiêu của tổ chức.

Phản sau trình bày các danh sách danh tiếng công khai phổ biến.

#### **Danh sách tên miền có mã độc**

Một trong những cách dễ nhất để phát hiện phần mềm độc hại ở mức độ mạng là sử dụng danh sách danh tiếng nào có chứa địa chỉ IP và tên miền gắn với những liên lạc liên quan tới các phần mềm độc hại.

Danh sách tên miền có mã độc (Malware Domain List - MDL) là một dự án cộng đồng phi thương mại duy trì danh mục các tên miền và các địa chỉ IP độc hại. MDL cho phép truy vấn danh sách một cách lân lượt, hoặc tải về danh sách trong một loạt các định dạng, bao gồm CSV, nguồn cấp dữ liệu RSS, và danh sách đã được định dạng hosts.txt. MDL là một trong những danh sách có danh tiếng lớn nhất và được sử dụng nhiều nhất hiện nay.

Tuy nhiên, có thể có sai lầm khi sử dụng MDL. Vì vậy một cảnh báo được tạo ra từ một máy chủ được bảo vệ liên lạc với một mục tìm thấy trên MDL là không đủ. Khi một cảnh báo được tạo ra, nên điều tra các nguồn dữ liệu khác và các liên lạc khác từ những gì được bảo vệ để cố gắng xác định xem có những dấu hiệu khác của nhiễm độc hoặc tấn công hay không.

Có thể tìm hiểu thêm về MDL tại <http://www.malwaredomainlist.com>.

#### **Abuse.ch Zeus và SpyEye Trackers**

Zeus và SpyEye là các bộ công cụ tội phạm cực kỳ phổ biến được sử dụng bởi những kẻ tấn công để lây nhiễm độc cho hệ thống và thực hiện một loạt các hành động nguy hiểm (Hình 3.5). Các bộ dụng cụ cung cấp khả năng để tạo ra phần mềm độc hại lây nhiễm vào máy thông qua các hình thức tải về, cuối cùng gia nhập vào một botnet mà các bộ công cụ trên có thể sử dụng để

kiểm soát. Có khoảng thời gian Zeus là botnet lớn nhất thế giới, còn SpyEye là một trong những đối thủ cạnh tranh lớn nhất.

The screenshot shows a web browser window titled "Zeus Tracker :: Monitor" at the URL <https://zeustracker.abuse.ch/monitor.php>. The page contains the following information:

- Zeus Tracker :: Monitor**: A list of all Zeus C&Cs and Fake URLs known to Zeus Tracker.
- Description of Levels** (Color-coded legend):
  - Level 1: Bulletproof host
  - Level 2: Hacked webserver
  - Level 3: Free hosting service
  - Level 4: Unknown
  - Level 5: FastFlux hosted
- Additional Categories** (Icons):
  - Hosts tagged as are Zeus Command&Control servers
  - Hosts tagged as are referenced by Zeus as FakeURLs
- Search Function**: You can search the Zeus Tracker for domains, IPs, urls, MD5 hashes or FakeURLs.
- Browse Options**: ZeuS Binary/Bin | ZeuS Config/Bin | ZeuS Dropzones
- Malware family filter**: ZeuS | Ice IX | Citadel
- Filter Options**: Remove filter (Show all) | online ZeuS hosts | offline ZeuS hosts | ZeuS hosts with files online | order by lastupdated | Filter C&C server tagged with level: Level 1 (Bulletproof) | Level 2 (Hacked sites) | Level 3 (free webhosting) | Level 4 (Unknown) | Level 5 (FastFlux hosted)
- Subscribe**: Subscribe this list via RSS feed
- Data Table** (List of hosts):
 

Date added	Malware	Host	IP address	Level	Status	Files Online	SBL	Country	AS number	Uptime
2013-06-28	ZeuS	shanghairegistry.com	173.208.107.195	2	online	2	Not listed	US	AS15003	00:14:29
2013-06-28	Citadel	gnacislenivdunlo.ru		FastFlux Botnet	online	1	Not listed	-	-	05:49:14
2013-06-26	Citadel	astarts.ru		FastFlux Botnet	online	1	Not listed	-	-	16:41:24
2013-06-26	Citadel	centow.ru		FastFlux Botnet	online	1	Not listed	-	-	20:42:03
2013-06-26	Citadel	pekin34.ru	31.31.199.159	2	online	1	Not listed	RU	AS39792	00:43:03
2013-06-26	ZeuS	cardpalooza.su		FastFlux Botnet	online	1	Not listed	-	-	00:36:22
2013-06-26	Citadel	volgpromotion.su		FastFlux Botnet	online	2	Not listed	-	-	00:49:48
2013-06-26	Citadel	lomomo.com	46.119.217.55	4	online	2	Not listed	ES	AS15895	21:00:09
2013-06-26	ZeuS	eltrijuno.com.mx	216.139.244.210	2	online	2	Not listed	MX	AS32400	23:24:53

Hình 3.5 Zeus Tracker

Zeus và SpyEye Tracker Tracker là những dự án theo dõi và kiểm soát các máy chủ ra lệnh trên Internet được sử dụng để kiểm soát máy tính bị nhiễm Zeus và SpyEye. Thêm vào đó, các dịch vụ cũng theo dõi các máy chủ bị nhiễm Zeus và SpyEye. Có thể tìm hiểu thêm về các Tracker Zeus ở <https://zeustracker.abuse.ch/>, và SpyEye tracker tại <https://spyeyetracker.abuse.ch/>.

## PhishTank

Một lượng lớn các cuộc tấn công đã nhắm tới mục tiêu được bắt đầu với một số loại lừa đảo. Hầu hết các tổ chức thành công phát hiện các loại tấn công sau giai đoạn đầu này, tuy nhiên, khả năng nhận biết thời điểm người dùng đang được chuyển đến trang web lừa đảo đã biết có thể có ích cho việc phát hiện sớm sự cố đang xảy ra, hoặc cho một cuộc điều tra hồi cứu của một sự cố đã xảy ra.

PhishTank, dịch vụ của OpenDNS, là một trang web dựa vào cộng đồng, miễn phí, cho phép chia sẻ các dữ liệu liên quan đến lừa đảo. Sau khi đăng ký, người dùng có thể gửi liên kết (link) họ đã tìm thấy để báo nếu chúng liên quan với các kiểu lừa đảo. PhishTank là duy nhất bởi vì nó dựa vào việc gửi và xác minh dựa trên cộng đồng. Để bắt kỳ URL nào đó xuất hiện trên danh sách của nó, URL phải được xác nhận bởi một số lượng nhất định những người dùng PhishTank đã đăng ký. Có thể tìm hiểu thêm về PhishTank tại <http://www.phishtank.com/>.

## Các danh sách khác

Một loạt các danh sách danh tiếng IP và tên miền khác có sẵn, bao gồm:

- Tor Exit Node
- Spamhaus
- AlientVault Labs IP Reputation Database:  
<http://labs.alienvault.com/labs/index.php/projects/open-source-ip-reputation-portal/>
- MalC0de Database: <http://malc0de.com/database/>
- SRI Malware Threat Center  
[http://www.mtc.sri.com/live\\_data/attackers/](http://www.mtc.sri.com/live_data/attackers/)
- Project Honeypot: [https://www.projecthoneypot.org/list\\_of\\_ips.php](https://www.projecthoneypot.org/list_of_ips.php)
- Emerging Threats Rules: <http://www.emergingthreats.net/open-source/etopen-ruleset/>

### 3.2.2 Tự động phát hiện xâm nhập dựa trên danh tiếng

Để thực hiện phát hiện dựa trên danh tiếng cần có hai thành phần. Đầu tiên, cần ít nhất một danh sách các IP hoặc tên miền với danh tiếng xấu. Sau khi có ít nhất một danh sách, cần đưa nội dung của danh sách vào một số loại hình cơ chế phát hiện xâm nhập dựa trên các mục trong danh sách. Có một số tùy chọn cho việc tự động hóa các nhiệm vụ này.

#### Phát hiện danh tiếng IP với Snort

Trong quá khứ, phát hiện dựa trên danh tiếng cho các địa chỉ IP với Snort được thực hiện với các luật chuẩn. Để giải quyết được vấn đề này, tiền xử lý danh tiếng đã được phát triển. Tiền xử lý này chạy trước tất cả các tiền xử lý khác một cách có hiệu quả.

Tiền xử lý danh tiếng được kích hoạt trong Snort trên Security Onion, nhưng cảnh báo cho nó chưa được kích hoạt. Trước khi thêm các mục vào danh sách đen của tiền xử lý danh tiếng, chúng ta nên cho phép cảnh báo. Để làm được điều này, trước tiên cần tạo ra một tệp tin gọi là `preprocessor_rules` trong `/etc/NSM/rules` của bộ cảm biến SO. Tập luật này nên chứa các luật như sau để cho phép để cảnh báo các sự kiện tiền xử lý danh tiếng:

```
alert ( msg: "REPUTATION_EVENT_BLACKLIST"; sid: 1; gid: 136; rev: 1; metadata: rule-type preproc ; classtype:bad-unknown; )
```

Tiếp theo, cấu hình Snort phải được sửa đổi để cho phép phân tích các tệp tin luật tiền xử lý mà vừa tạo ra. Điều này được thực hiện bằng cách chỉnh sửa: `/etc/nsm/sensor_name/snort.conf`, và bỏ ghi chú (bỏ comment) dòng này:

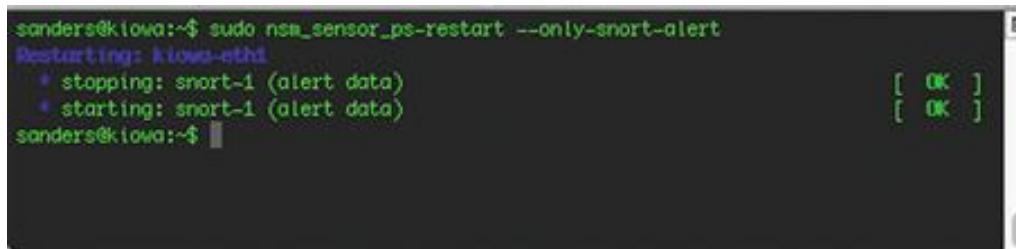
```
include $PREPROC_RULE_PATH/preprocessor.rules
```

Bây giờ, điều duy nhất còn lại là bổ sung thêm các địa chỉ IP vào danh sách đen tiền xử lý danh tiếng. Tập tin này có thể được tìm thấy tại `/etc/nsm/rules/black_list.rules`. Các tệp tin chấp

nhận cả các địa chỉ IP riêng lẻ, và các dãy địa chỉ IP trong CIDR. Để kiểm tra các tiền xử lý, có thể thêm mục sau đây:

```
192.0.2.75 # Test Address
```

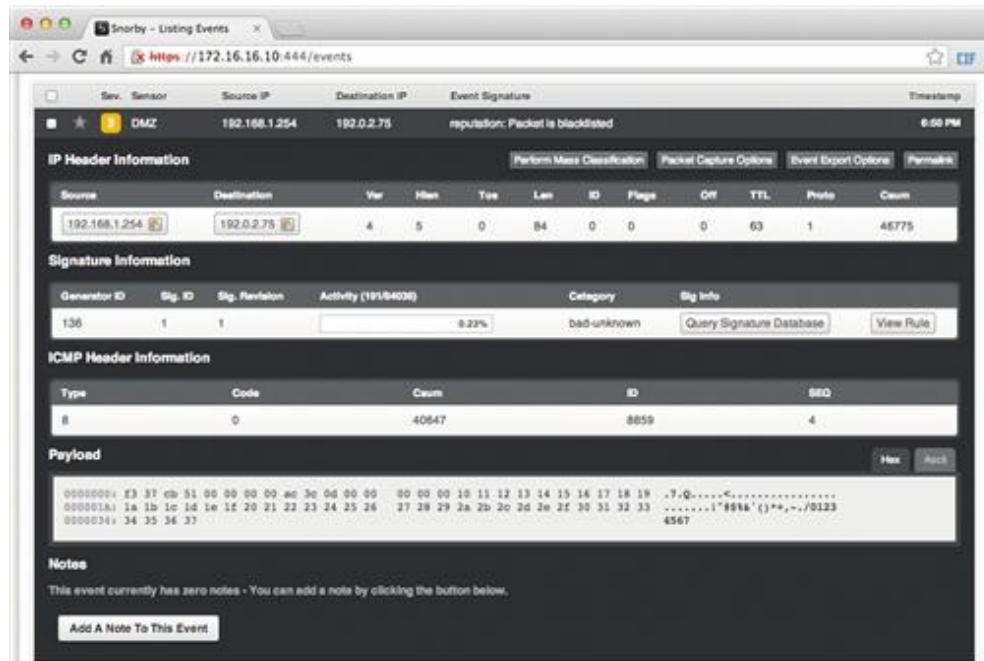
Để cho các thay đổi có hiệu lực, cần khởi động lại Snort trên cảm biến, như trong Hình 3.6.



```
sanders@k1owa:~$ sudo nsm_sensor_ps-restart --only-snort-alert
Restarting: k1owa-eth1
  * stopping: snort-1 (alert data)
  * starting: snort-1 (alert data)
sanders@k1owa:~$
```

**Hình 3.6 Khởi động lại trình Snort**

Để kiểm tra các luật mới được tạo ra, ping đến địa chỉ 192.0.2.75 từ chính Security Onion, hoặc từ một thiết bị khác mà nó đang theo dõi. Hình 3.7 là một ví dụ về luật Snort tạo ra một cảnh báo.



**Hình 3.7 Một cảnh báo được tạo ra bởi các tiền xử lý danh tiếng**

### Phát hiện danh tiếng IP với Suricata

Suricata nhanh chóng phổ biến như một thay thế cho Snort trong việc phát hiện xâm nhập dựa trên chữ ký. Điều này chủ yếu là do khả năng kiểm tra lưu lượng truy cập đa luồng, làm cho

nó thích hợp hơn khi giám sát kết nối thông lượng cao. Suricata cũng sử dụng cú pháp luật tương tự như Snort, nên các luật có thể được sử dụng bởi cả hai công cụ này.

Chức năng phát hiện dựa trên danh tiếng của Suricata được thiết kế để tối ưu hóa xử lý một số lượng lớn các mục, bằng cách sử dụng cùng các API để gắn thẻ và xác định người. Để kích hoạt chức năng này, đầu tiên cần phải thay đổi tệp tin cấu hình Suricata.yaml. Phần sau được sử dụng để phát hiện danh tiếng IP:

```
# IP Reputation
  reputation-categories-file: /etc/nsm/sensor-name/ipmap/categories.txt
  default-reputation-path: /etc/nsm/rules
  reputation-files:
    - zeustracker.list
    - spyeyetracker.list
    - mdl.list
    - watch.list
```

Mục đầu tiên trong cấu hình này là các tệp tin mô tả loại danh tiếng. Danh mục cho phép tổ chức các danh sách và thông báo của chúng thành các đơn vị quản lý được. Các tệp tin danh mục yêu cầu chỉ định một số id duy nhất cho mỗi thẻ loại, tên thẻ loại và mô tả. Thông thường, mỗi loại sẽ được tổ chức theo danh sách nguồn và theo định dạng sau:

<id>, <tên ngắn>, <mô tả>

Ví dụ một tệp tin mô tả loại có thể như sau:

```
1,ZeusTracker,Zeustracker IP Addresses
2,SpyEyeTracker,SpyEye Tracker IP Addresses
3,MDL,Malware Domain List IP Addresses
4,Watchlist,Internal Watch List IP Addresses
```

Tiếp theo, cần phải xác định đường dẫn danh tiếng mặc định, đó là thư mục có chứa danh sách các tệp tin danh tiếng. Trong trường hợp ví dụ trên, đã chọn đặt các tệp tin trong cùng một thư mục cùng các luật của Suricata/Snort.

Các mục cấu hình cuối cùng là xác định các tệp tin danh sách thực tế được phân tích bằng Suricata. Những tập tin này phải tồn tại bên trong đường dẫn danh tiếng mặc định. Các mục trong những tập tin này phải phù hợp với các định dạng:

<IP>, <category>, <tin>

Định dạng này yêu cầu các địa chỉ IP đúng chuẩn.Thêm vào đó, các loại số hiệu quy định phải tồn tại trong tập các loại được đề cập trước đó. Cuối cùng, cần phải có một giá trị độ tin cậy.

Ví dụ, tệp tin danh sách danh tiếng như sau:

192.0.2.1,1,65

192.0.2.2,1,50

192.0.2.3,2,95

Sau khi cấu hình danh tiếng IP, phần còn lại là nhằm tạo ra cảnh báo để các chuyên gia phân tích có thể được thông báo bất cứ khi nào phát hiện có liên lạc với một trong các địa chỉ IP. Điều này được thực hiện bằng cách thêm một luật sử dụng các chỉ thị *iprep*. Chỉ thị *iprep* có bốn lựa chọn:

- Hướng liên lạc (any/src/dst/both): Được sử dụng để xác định hướng của lưu lượng truy cập đến/từ các IP.
- Loại (tên ngắn gọn): Tên viết tắt của các thẻ loại phù hợp. Tên ngắn phải phù hợp và chính xác với những gì được liệt kê trong tệp tin về thẻ loại.
- Toán tử (>, <, =): sử dụng kết hợp với các giá trị danh tiếng được xác định.
- Giá trị tin cậy (1-127): hạn chế các kết quả tìm thấy để chỉ lấy những địa chỉ có độ tin cậy phù hợp với các toán tử và các giá trị xác định.

Ví dụ về một luật rất cơ bản chỉ có IP, như sau:

```
alert ip any any -> any any (msg:"IPREP Malware Domain List – High Confidence";  
iprep:dst,MDL,>,75; sid:1; rev:1);
```

Luật này sẽ tạo ra một cảnh báo bất cứ khi nào phát hiện có liên lạc ra ngoài tới một địa chỉ IP được liệt kê trên danh sách MDL, có độ tin cậy lớn hơn 75. Một ví dụ cảnh báo được tạo ra bởi luật này được thể hiện trong Hình 3.8.

The screenshot shows the Snorby web interface with the title "Snorby - Listing Sessions". The URL is https://172.16.16.123:444/events/sessions. The main content area displays an event titled "IPREP Malware Domain List - High Confidence" with timestamp "1:06 AM". The event details are as follows:

- IP Header Information:** Shows a single entry with Source: 172.16.16.123 and Destination: 192.0.2.1.
- Signature Information:** Shows one signature entry with Generator ID: 1, Sig. ID: 1234567, and Activity: 4.44%.
- ICMP Header Information:** Shows one ICMP entry with Type: 8, Code: 0, and Csum: 56426.
- Payload:** Displays the raw hex and ASCII data of the captured packet.
- Notes:** A note states "This event currently has zero notes - You can add a note by clicking the button below." with a "Add A Note To This Event" button.

Hình 3.8 Một cảnh báo được tạo ra bởi chỉ thị Suricata Iprep

Suricata có khả năng phân tích một số lượng lớn các địa chỉ IP bằng cách sử dụng phương pháp này (lên tới hàng triệu). Hệ thống này là một sự lựa chọn vững chắc và hiệu quả để phát hiện dựa trên danh tiếng của địa chỉ IP.

### Phát hiện danh tiếng IP với Bro

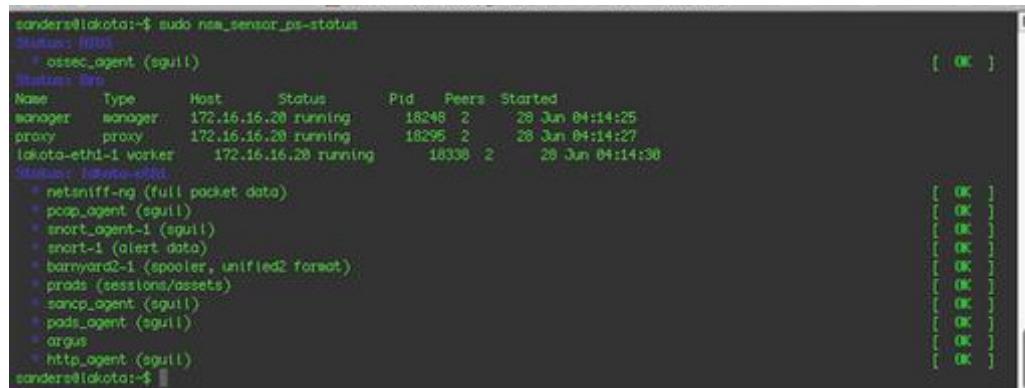
Bro IDS là một trong những công cụ phát hiện NSM mạnh mẽ và linh hoạt nhất hiện nay. Bro là rất thích hợp cho việc phát hiện một số loại IOC, chẳng hạn như địa chỉ IP, tên miền, địa chỉ thư điện tử và chứng chỉ SSL nhờ sử dụng các tính năng xử lý thông minh có sẵn được gọi là *intel framework*. Để biết được cách thức chi tiết sử dụng Bro để phát hiện danh tiếng, có thể tham khảo tới tài liệu [1].

## 3.3 PHÁT HIỆN XÂM NHẬP DỰA TRÊN CHỮ KÝ VỚI SNORT VÀ SURICATA

### 3.3.1 Snort

Snort là một trong các IDS phổ biến nhất trong thế giới do có nhiều tính năng mạnh mẽ và linh hoạt, nhiều tính năng đã trở thành tiêu chuẩn cho ngành công nghiệp IDS.

Snort được cài đặt mặc định trên Security Onion. Nếu đã chọn Snort là IDS mặc định trên Security Onion, chúng ta có thể xác minh bằng cách chạy lệnh sudo nsm\_sensor\_ps-status. Trong đầu ra trong Hình 3.9, sẽ thấy snort-1 (dữ liệu cảnh báo) được báo [OK].



```
sanders@lakota:~$ sudo nsm_sensor_ps-status
Status: [OK]
  ossec_agent (sgull)
Status: [OK]
Name      Type      Host      Status      Pid  Peers Started
snorger   manager   172.16.16.20 running    18248 2     28 Jun 04:14:25
proxy     proxy     172.16.16.20 running    18295 2     28 Jun 04:14:27
lakota-eth1-worker  worker   172.16.16.20 running    18330 2     28 Jun 04:14:30
Status: [OK]
  netsniff-ing (full packet data)
  psop_agent (sgull)
  snort_agent-1 (sgull)
  snort-1 (alert data)
  barnyard2-1 (spooler, unified2 format)
  probs (sessions/assets)
  sanqp_agent (sgull)
  pads_agent (sgull)
  argus
  http_agent (sgull)
sanders@lakota:~$
```

Hình 3.9 Kiểm tra tình trạng cảm biến

### Kiến trúc Snort

Chức năng của Snort sẽ phụ thuộc vào chế độ hoạt động được quy định tại thời điểm chạy. Snort có ba chế độ hoạt động chính: chế độ sniffer, chế độ log gói tin, và chế độ NIDS.

Chế độ sniffer cho phép Snort bắt các gói tin và xuất chúng ra màn hình trong một định dạng có thể đọc được, cũng giống như *tcpdump*. Tuy nhiên, đầu ra của nó đẹp hơn một chút so với *tcpdump*. Ví dụ về dữ liệu gói có thể thấy trong Hình 3.10.

```

=====
07/22-16:53:26.591784 20:09:00:BA:63:FB -> 00:0C:29:15:4A:1F type:0x800 len:0x42
172.16.16.113:58934 -> 172.16.16.10:22 TCP TTL:64 TOS:0x0 ID:6760 IpLen:28 DgmLen:52 DF
***A*** Seq: 0xF561F563 Ack: 0xBA2D9ABA Win: 0x205C TcpLen: 32
TCP Options (3) => NOP NOP TS: 997547882 531097969

=====
07/22-16:53:26.591742 20:09:00:BA:63:FB -> 00:0C:29:15:4A:1F type:0x800 len:0x42
172.16.16.113:58934 -> 172.16.16.10:22 TCP TTL:64 TOS:0x0 ID:12860 IpLen:28 DgmLen:52 DF
***A*** Seq: 0xF561F563 Ack: 0xBA2D9C2A Win: 0x2045 TcpLen: 32
TCP Options (3) => NOP NOP TS: 997547882 531097969

=====
07/22-16:53:26.592019 20:09:00:BA:63:FB -> 00:0C:29:15:4A:1F type:0x800 len:0x42
172.16.16.113:58934 -> 172.16.16.10:22 TCP TTL:64 TOS:0x0 ID:13341 IpLen:28 DgmLen:52 DF
***A*** Seq: 0xF561F563 Ack: 0xBA2D9CAA Win: 0x2054 TcpLen: 32
TCP Options (3) => NOP NOP TS: 997547882 531097969

=====
07/22-16:53:26.592688 20:09:00:BA:63:FB -> 00:0C:29:15:4A:1F type:0x800 len:0x42
172.16.16.113:58934 -> 172.16.16.10:22 TCP TTL:64 TOS:0x0 ID:48737 IpLen:28 DgmLen:52 DF
***A*** Seq: 0xF561F563 Ack: 0xBA2D9CDA Win: 0x2059 TcpLen: 32
TCP Options (3) => NOP NOP TS: 997547882 531097969

=====
07/22-16:53:26.592790 20:09:00:BA:63:FB -> 00:0C:29:15:4A:1F type:0x800 len:0x42
172.16.16.113:58934 -> 172.16.16.10:22 TCP TTL:64 TOS:0x0 ID:39689 IpLen:28 DgmLen:52 DF
***A*** Seq: 0xF561F563 Ack: 0xBA2D905A Win: 0x2054 TcpLen: 32
TCP Options (3) => NOP NOP TS: 997547883 531097969
=====
```

**Hình 3.10 Đầu ra của Snort trong chế độ Sniffer**

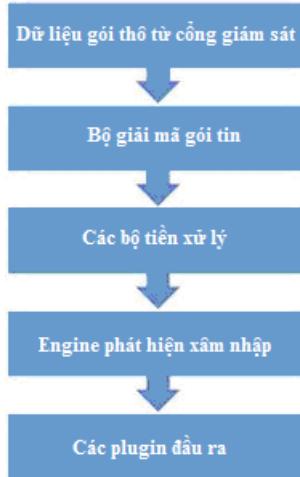
Chế độ sniffer là chế độ mặc định, vì vậy có thể chạy Snort ở chế độ này bằng cách đơn giản là xác định một giao diện mạng với lệnh snort -i <interface>.

Chế độ log gói tin cũng tương tự như chế độ sniffer, chỉ ghi các gói tin vào một tệp tin chứ không phải là màn hình. Những thông tin này thường được ghi ở định dạng PCAP nhị phân. Chế độ hoạt động được kích hoạt bằng cách xác định các thư mục log với việc bổ sung tham số như sau: snort -l <thư mục log>. Khi cần đọc tệp tin PCAP, thực hiện gọi Snort với lệnh: snort -r <pcap file>.

Chế độ quan tâm nhất là chế độ NIDS, được thiết kế để đọc dữ liệu bắt giữ từ mạng, với mục tiêu cuối cùng là đưa ra cảnh báo. Để làm được điều này, dữ liệu gói đi qua các giai đoạn khác nhau của kiến trúc của Snort, thể hiện trong Hình 3.11.

Snort có thể nhận dữ liệu bằng cách phân tích một tệp tin PCAP hoặc bằng cách lấy từ một giao diện mạng đang giám sát của cảm biến. Khi Snort nhận dữ liệu này, bước đầu tiên là phân tích bằng các bộ giải mã gói tin. Trong thực tế, đây là một loạt các bộ giải mã phân tích dữ liệu gói và bình thường hóa dữ liệu để thích hợp cho việc phân tích bởi các tiền xử lý và các công cụ phát hiện.

Khi dữ liệu đã được xử lý bởi các bộ giải mã gói tin, nó được gửi đến các tiền xử lý trong Snort. Có hai loại tiền xử lý. Loại đầu tiên được sử dụng cho mục đích phát hiện xâm nhập. Loại thứ hai bao gồm những tiền xử lý được sử dụng để sửa đổi dữ liệu gói, sao cho có thể được phân tích tốt hơn bởi các công cụ phát hiện.



**Hình 3.11 Kiến trúc Snort NIDS**

Sau khi kết thúc tiền xử lý, dữ liệu được chuyển tới engine phát hiện trong kiến trúc Snort. Engine phát hiện có trách nhiệm phân tích cú pháp các luật và xác định liệu các điều kiện xác định trong các luật phù hợp với lưu lượng đang được phân tích hay không.

Khi engine phát hiện xác định rằng lưu lượng phù hợp với luật, nó chuyển dữ liệu qua các plugin đầu ra xác định trong tệp tin cấu hình Snort, để từ đó một chuyên gia phân tích có thể được thông báo về các cảnh báo. Snort có thể ghi log lại theo nhiều định dạng, bao gồm cả các thông báo đơn dòng trong một tệp văn bản, tệp tin CSV, định dạng PCAP chứa lưu lượng phù hợp với các luật, định dạng XML, Syslog,... Trong nhiều môi trường sản xuất, Snort được cấu hình để ghi log lại theo định dạng Unified2, một định dạng mở có thể được đọc bởi các công cụ như Barnyard2 hoặc Pigsty, sau đó có thể được sử dụng cho các định dạng đầu ra linh hoạt hơn như đầu ra trực tiếp đến một cơ sở dữ liệu.

### 3.3.2 Suricata

Trong khi Snort là IDS phổ biến nhất dựa trên chữ ký ngày nay, một công cụ khác cũng phổ biến không kém là Suricata, là một IDS mã nguồn mở được phát triển bởi OISF. Điều này chủ yếu là do sự hiệu quả của nó, trong việc thiết kế thực hiện đa luồng. Chức năng của Suricata tương tự như Snort.

Nếu đã thiết lập Security Onion và chọn Suricata làm IDS, có thể xác minh bằng cách chạy lệnh sudo nsm\_sensor\_ps-status. Trong đầu ra thể hiện trong Hình 3.12, sẽ thấy Suricata (dữ liệu cảnh báo) được báo là [OK].

```

sander@so-suricata:~$ sudo nsm_sensor_ps-status
Status: OK
  * ossec_agent (sgui)
Status: OK
Name      Type      Host      Status      Pid  Peers Started
manager   manager   172.16.16.123 running    3485  2   01 Aug 19:27:17
proxy     proxy     172.16.16.123 running    3656  2   01 Aug 19:27:19
so-suricata-eth0-1 worker  172.16.16.123 running    3823  2   01 Aug 19:27:21
Status: OK
  * netsniff-ng (full packet data)
  * pcap_agent (sgui)
  * snort_agent (sgui)
  * suricata (alert data)
  * barnyard2 (spooler, unified2 format)
  * prods (sessions/assets)
  * soncp_agent (sgui)
  * pods_agent (sgui)
  * argus
  * http_agent (sgui)
Status: OK

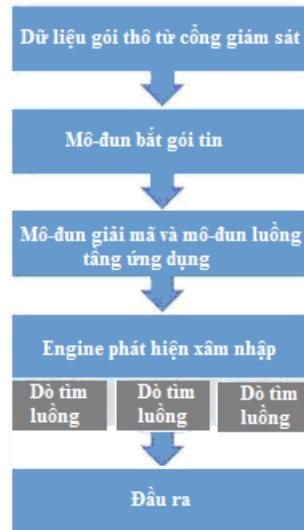
```

**Hình 3.12 Kiểm tra tình trạng cảm biến**

### Kiến trúc Suricata

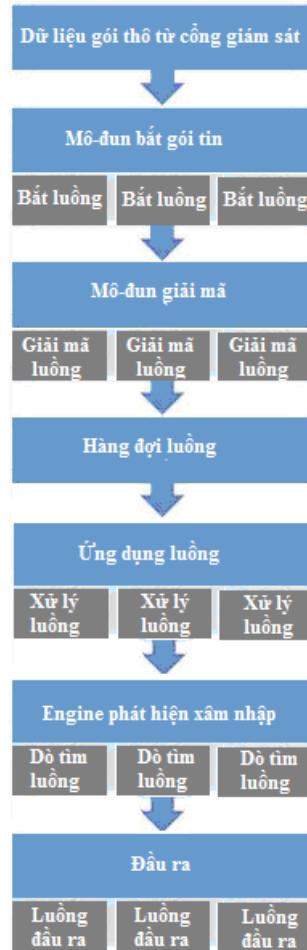
Suricata được tạo bởi một số mô-đun có thể tương tác khác nhau tùy thuộc vào cách Suricata được khởi tạo. Cách thức mà các mô-đun và các luồng (thread) cùng hàng đợi liên kết với chúng được bố trí được gọi là *runmode* của Suricata. *Runmode* này được lựa chọn dựa trên mức độ ưu tiên xử lý của Suricata được thiết lập.

*Runmode* mặc định được tối ưu hóa cho việc phát hiện xâm nhập, và đây là mô-đun cần nhiều tài nguyên nhất. *Runmode* này được mô tả trong Hình 3.13.



**Hình 3.13 Runmode mặc định của Suricata**

Trong một *runmode* khác, *pfring* được sử dụng để tối ưu hóa việc bắt gói tin và giải mã cho các liên kết thông lượng cao. *Runmode* này được thể hiện trong Hình 3.14.



**Hình 3.14 Pfring Suricata Runmode**

Với bất kỳ *runmode* nào được sử dụng, thì bước đầu tiên Suricata thực hiện là thu thập các gói tin với mô-đun bắt gói tin (Packet Acquisition). Mô-đun này tập hợp các gói tin từ giao diện mạng và đưa chúng vào giải mã gói tin. Sau khi hoàn thành, dữ liệu được truyền cho mô-đun luồng. Mô-đun luồng chủ yếu chịu trách nhiệm theo dõi các giao thức phiên (như TCP) và nối ghép dữ liệu gói theo thứ tự thích hợp.Thêm vào đó, mô-đun luồng cũng thực hiện một số xử lý và nối dữ liệu lại từ các giao thức tầng ứng dụng như HTTP. Tất cả các dữ liệu được đưa vào mô-đun phát hiện xâm nhập. Khi cảnh báo được tạo ra, chúng có thể được gửi tới mô-đun đầu ra để xuất theo một số định dạng.

### 3.3.3 Thay đổi công cụ IDS trong Security Union

Nếu hoàn thành quá trình cài đặt Security Onion và bước đầu đã chọn một trong hai công cụ IDS là Snort hoặc Suricata, khi muốn thử các công cụ khác mà không cần cài đặt lại Security Onion, có thể thực hiện với một vài thay đổi nhanh chóng như sau.

1. Dừng quá trình cảm biến NSM:  
`sudo nsm_sensor_ps-stop`
2. Sửa đổi tệp tin SO cấu hình chính:

Chuyển từ Snort tới Suricata:

```
sudo -i sed 's | ENGINE = snort | ENGINE = suricata | g' /etc/nsm/securityonion.conf
```

Chuyển từ Suricata tới Snort:

```
sudo -i sed 's | ENGINE = suricata | ENGINE = snort | g' /etc/nsm/securityonion.conf
```

3. Cập nhật tập luật cảm biến cho công cụ IDS thích hợp:

Cập nhật luật sudo

4. Bắt đầu các tiến trình cảm biến NSM:

```
sudo nsm_sensor_ps-start
```

Nếu đã phát triển những luật khác cho cảm biến riêng thì hãy chắc chắn là chúng phù hợp với các công cụ IDS mà sẽ chuyển sang để dự đoán được bất kỳ vấn đề nào có thể ngăn các IDS khỏi tạo.

### 3.3.4 Khởi tạo Snort và Suricata cho việc phát hiện xâm nhập

Để chạy Snort hoặc Suricata với mục đích phát hiện xâm nhập, tất cả những gì cần làm là xác định vị trí của một tệp tin cấu hình hợp lệ với tùy chọn dòng lệnh -c và một giao diện giám sát với tùy chọn -i.

Snort:        sudo snort -c snort.conf -i eth1

Suricata:     sudo suricata -c suricata.yaml -i eth1

Trước khi thực hiện việc này, điều quan trọng là xác minh các tệp tin cấu hình là hợp lệ bằng cách thêm tham số -T, để chạy công cụ IDS với các tệp tin cấu hình được cung cấp nhằm đảm bảo là chúng có thể khởi động thành công với cấu hình được cung cấp.

Snort:        sudo snort -Tc snort.conf -i eth1

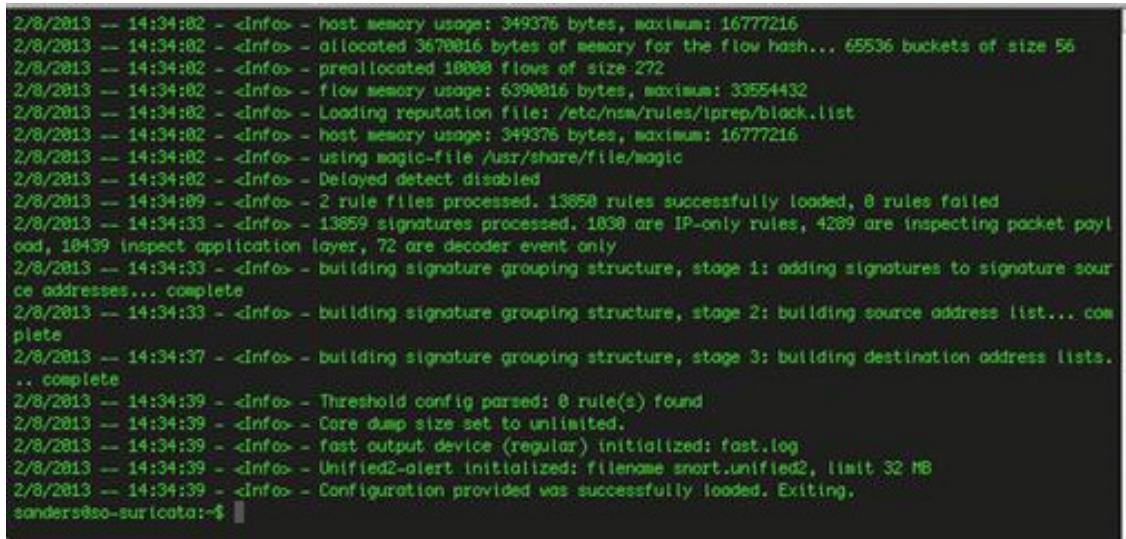
Suricata:     sudo suricata -Tc suricata.yaml -i eth1

```
~*> Snort! <*~  
Version 2.9.4.6 GRE (Build 73)  
By Martin Roesch & The Snort Team: http://www.snort.org/snort/snort-team  
Copyright (C) 1998-2013 Sourcefire, Inc., et al.  
Using libpcap version 1.1.1  
Using PCRE version: 8.12 2011-01-15  
Using ZLIB version: 1.2.3.4  
  
Rules Engine: SF_SNORT_DETECTION_ENGINE Version 1.17 <Build 18>  
Preprocessor Object: SF_SOF Version 1.1 <Build 1>  
Preprocessor Object: SF_DNS Version 1.1 <Build 4>  
Preprocessor Object: SF_IMAP Version 1.0 <Build 1>  
Preprocessor Object: SF_REPUTATION Version 1.1 <Build 1>  
Preprocessor Object: SF_MODBUS Version 1.1 <Build 1>  
Preprocessor Object: SF_SIP Version 1.1 <Build 1>  
Preprocessor Object: SF_DNP3 Version 1.1 <Build 1>  
Preprocessor Object: SF_FTPTELNET Version 1.2 <Build 13>  
Preprocessor Object: SF_SSH Version 1.1 <Build 3>  
Preprocessor Object: SF_POP Version 1.0 <Build 1>  
Preprocessor Object: SF_SSLPP Version 1.1 <Build 4>  
Preprocessor Object: SF_GTP Version 1.1 <Build 1>  
Preprocessor Object: SF_SMTP Version 1.1 <Build 9>  
Preprocessor Object: SF_DCERPC2 Version 1.0 <Build 3>  
  
Snort successfully validated the configuration!  
Snort exiting
```

Hình 3.15 Snort thành công kiểm tra một tệp tin cấu hình trong chế độ NIDS

Nếu tất cả mọi thứ được kiểm tra với Snort, sẽ thấy một thông báo là nó đã xác nhận thành công cấu hình, như thể hiện trong Hình 3.15. Snort sẽ thoát ra khi kiểm tra này được hoàn thành.

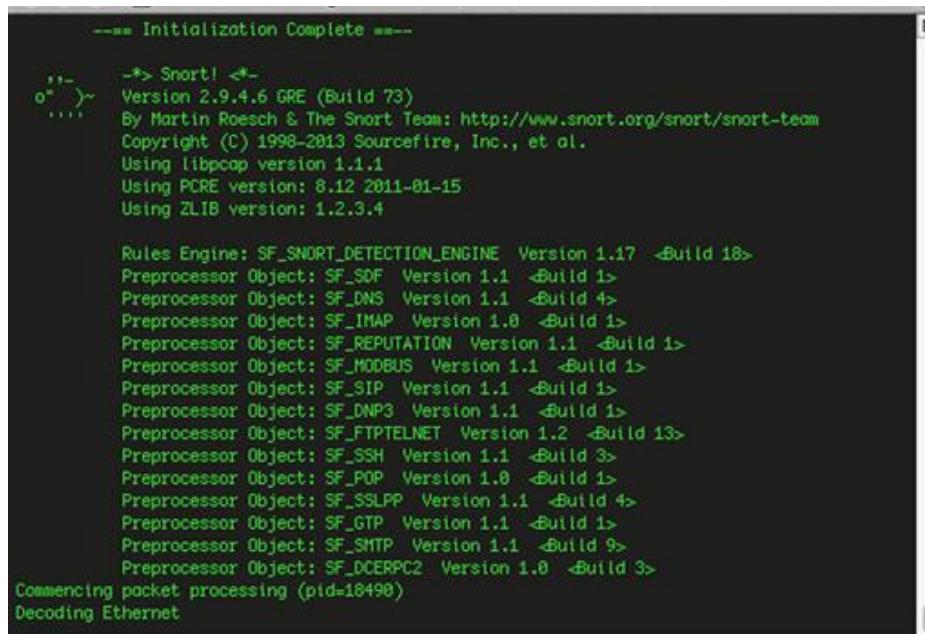
Nếu Suricata khởi tạo thành công, sẽ thấy một thông báo là các cấu hình cung cấp đã được nạp thành công, như thể hiện trong Hình 3.16. Suricata sẽ thoát ra khi thử nghiệm này được hoàn thành.



```
2/8/2013 -- 14:34:02 - <Info> - host memory usage: 349376 bytes, maximum: 16777216
2/8/2013 -- 14:34:02 - <Info> - allocated 3670016 bytes of memory for the flow hash... 65536 buckets of size 56
2/8/2013 -- 14:34:02 - <Info> - preallocated 10000 flows of size 272
2/8/2013 -- 14:34:02 - <Info> - flow memory usage: 6390016 bytes, maximum: 33554432
2/8/2013 -- 14:34:02 - <Info> - Loading reputation file: /etc/nsm/rules/ipmap.block.list
2/8/2013 -- 14:34:02 - <Info> - host memory usage: 349376 bytes, maximum: 16777216
2/8/2013 -- 14:34:02 - <Info> - using magic-file /usr/share/file/magic
2/8/2013 -- 14:34:02 - <Info> - Delayed detect disabled
2/8/2013 -- 14:34:09 - <Info> - 2 rule files processed. 13850 rules successfully loaded, 0 rules failed
2/8/2013 -- 14:34:33 - <Info> - 13859 signatures processed. 1030 are IP-only rules, 4209 are inspecting packet payload, 10439 inspect application layer, 72 are decoder event only
2/8/2013 -- 14:34:33 - <Info> - building signature grouping structure, stage 1: adding signatures to signature source addresses... complete
2/8/2013 -- 14:34:33 - <Info> - building signature grouping structure, stage 2: building source address list... complete
2/8/2013 -- 14:34:37 - <Info> - building signature grouping structure, stage 3: building destination address lists... complete
2/8/2013 -- 14:34:39 - <Info> - Threshold config parsed: 0 rule(s) found
2/8/2013 -- 14:34:39 - <Info> - Core dump size set to unlimited.
2/8/2013 -- 14:34:39 - <Info> - Fast output device (regular) initialized: fast.log
2/8/2013 -- 14:34:39 - <Info> - Unified2-alert initialized: filename snort.unified2, limit 32 MB
2/8/2013 -- 14:34:39 - <Info> - Configuration provided was successfully loaded. Exiting.
sonders@eo-suricata:~$
```

Hình 3.16 Suricata thành công kiểm tra một tệp tin cấu hình trong Runmode mặc định

Nếu Snort khởi động ở chế độ NIDS thành công, sẽ nhận được thông báo là Snort bắt đầu xử lý gói tin, cùng với PID, như thể hiện trong Hình 3.17.



```
---- Initialization Complete ----

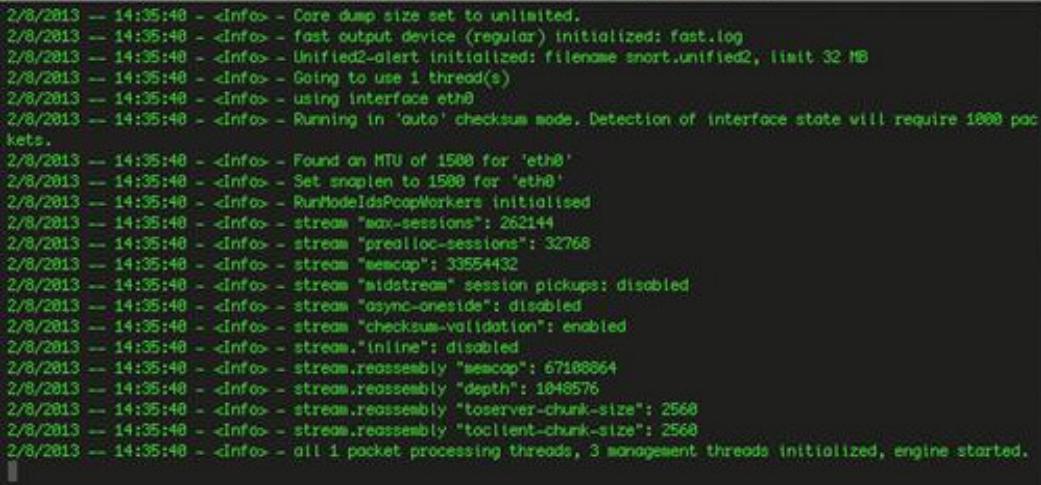
->> Snort! <-
o" >>
.... Version 2.9.4.6 GRE (Build 73)
By Martin Roesch & The Snort Team: http://www.snort.org/snort/snort-team
Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using libpcap version 1.1.1
Using PCRE version: 8.12 2011-01-15
Using ZLIB version: 1.2.3.4

Rules Engine: SF_SNORT_DETECTION_ENGINE Version 1.17 <Build 18>
Preprocessor Object: SF_SDF Version 1.1 <Build 1>
Preprocessor Object: SF_DNS Version 1.1 <Build 4>
Preprocessor Object: SF_IMAP Version 1.0 <Build 1>
Preprocessor Object: SF_REPUTATION Version 1.1 <Build 1>
Preprocessor Object: SF_MODBUS Version 1.1 <Build 1>
Preprocessor Object: SF_SIP Version 1.1 <Build 1>
Preprocessor Object: SF_DNP3 Version 1.1 <Build 1>
Preprocessor Object: SF_FTPTELNET Version 1.2 <Build 13>
Preprocessor Object: SF_SSH Version 1.1 <Build 3>
Preprocessor Object: SF_POP Version 1.0 <Build 1>
Preprocessor Object: SF_SSLPP Version 1.1 <Build 4>
Preprocessor Object: SF_GTP Version 1.1 <Build 1>
Preprocessor Object: SF_SMTP Version 1.1 <Build 9>
Preprocessor Object: SF_DCERPC2 Version 1.0 <Build 3>

Commencing packet processing (pid=18498)
Decoding Ethernet
```

Hình 3.17 Chạy thành công Snort ở chế độ NIDS

Nếu Suricata chạy thành công, sẽ được thông báo rằng các luồng đã được khởi tạo, và engine đã được bắt đầu, như thể hiện trong Hình 3.18.



```
2/8/2013 -- 14:35:40 - <Info> - Core dump size set to unlimited.
2/8/2013 -- 14:35:40 - <Info> - fast output device (regular) initialized: fast.log
2/8/2013 -- 14:35:40 - <Info> - Unified2-alert initialized: filename snort.unified2, limit 32 MB
2/8/2013 -- 14:35:40 - <Info> - Going to use 1 thread(s)
2/8/2013 -- 14:35:40 - <Info> - using interface eth0
2/8/2013 -- 14:35:40 - <Info> - Running in 'auto' checksum mode. Detection of interface state will require 1000 packets.
2/8/2013 -- 14:35:40 - <Info> - Found an MTU of 1500 for 'eth0'
2/8/2013 -- 14:35:40 - <Info> - Set snaplen to 1500 for 'eth0'
2/8/2013 -- 14:35:40 - <Info> - RunModeIdPcapWorkers initialised
2/8/2013 -- 14:35:40 - <Info> - stream "max-sessions": 262144
2/8/2013 -- 14:35:40 - <Info> - stream "prealloc-sessions": 32768
2/8/2013 -- 14:35:40 - <Info> - stream "memcap": 30554432
2/8/2013 -- 14:35:40 - <Info> - stream "midstream" session pickups: disabled
2/8/2013 -- 14:35:40 - <Info> - stream "async-one-sided": disabled
2/8/2013 -- 14:35:40 - <Info> - stream "checksum-validation": enabled
2/8/2013 -- 14:35:40 - <Info> - stream."(inline)": disabled
2/8/2013 -- 14:35:40 - <Info> - stream.reassembly "memcap": 67188664
2/8/2013 -- 14:35:40 - <Info> - stream.reassembly "depth": 1048576
2/8/2013 -- 14:35:40 - <Info> - stream.reassembly "tosever-chunk-size": 2568
2/8/2013 -- 14:35:40 - <Info> - stream.reassembly "toclient-chunk-size": 2568
2/8/2013 -- 14:35:40 - <Info> - all 1 packet processing threads, 3 management threads initialized, engine started.
```

Hình 3.18 Chạy thành công Suricata trong Runmode mặc định

Trong Security Onion, Snort và Suricata có thể được bắt đầu bằng cách sử dụng script nsm\_sensor\_ps-start (tham khảo [1]).

### 3.3.5 Cấu hình Snort và Suricata

Snort và Suricata dựa vào các tệp tin cấu hình và/hoặc tham số dòng lệnh để kiểm soát cách chúng hoạt động. Snort sử dụng một tệp tin gọi là snort.conf, và Suricata sử dụng suricata.yaml. Những tập tin này có thể được sử dụng để kiểm soát và tinh chỉnh hầu như mọi hành vi trong các ứng dụng, bao gồm cả đặc điểm của các công cụ phát hiện, vị trí của tệp tin luật, và việc kê khai của các biến được sử dụng trong các luật đó. Nếu đang sử dụng Security Onion, những tập tin này được đặt tại /etc/NSM/<sensor-interface>/. Phần tiếp theo sẽ bắt đầu xem xét một số mục cấu hình phổ biến được áp dụng cho cả hai công cụ.

#### 3.3.5.1. Các biến

Snort và Suricata sử dụng các biến trong các cấu hình tương ứng của chúng để bổ sung sự linh hoạt cho các luật IDS, dễ dàng tạo và duy trì chúng. Snort cũng sử dụng các biến trong tệp tin cấu hình của nó để chỉ đường dẫn chung. Một biến chỉ được quy định một lần để nó được nạp khi Snort được thực thi, và sau đó nó có thể được thay thế tại bất kỳ thời điểm nào trong các tệp tin cấu hình hoặc trong luật Snort. Có ba loại biến được sử dụng: biến IP, biến cổng, và các biến chuẩn.

#### Biến IP

Biến IP được sử dụng để xác định địa chỉ mạng hoặc một dải địa chỉ để sử dụng trong luật IDS khi đề cập đến các nguồn hoặc đích của lưu lượng đang được kiểm tra. Bằng cách sử dụng

các biến để xác định phạm vi IP thường xuyên tham chiếu, chúng ta chỉ cần cập nhật các biến một lần để áp dụng thay đổi bất kỳ luật tham chiếu phạm vi đó.

Với Snort, một biến IP được xác định trong snort.conf bởi từ khóa ipvar, theo sau là tên biến và các địa chỉ IP bao gồm các biến. Ví dụ, có thể chỉ định các biến sau đây để xác định một máy chủ DNS trên mạng:

```
ipvar DNS_SERVERS 192.168.1.10
```

Có thể chỉ định nhiều địa chỉ IP bằng cách kèm theo các địa chỉ trong dấu ngoặc vuông và tách chúng bằng dấu phẩy. Ở đây, thực hiện việc này để xác định một số máy chủ mail SMTP:

```
SMTP_SERVERS ipvar [192.168.1.75,192.168.1.76,192.168.1.77]
```

Có thể chỉ định các dải địa chỉ sử dụng ký hiệu CIDR. Ở đây, xác định hai mạng con có chứa các máy chủ web:

```
ipvar HTTP_SERVERS [192.168.2.0/24,192.168.12.0/24]
```

Suricata không sử dụng một từ khóa cụ thể để xác định các biến; thay vào đó, nó đòi hỏi biến của các kiểu cụ thể được xác định trong các phần chỉ định của suricata.yaml. Cụ thể, phải xác định tất cả các biến dưới tiêu đề vars, và các biến IP theo tiêu đề con address-groups.

vars:

address-groups:

```
DNS_SERVERS 192.168.1.10
```

```
SMTP_SERVERS [192.168.1.75,192.168.1.76,192.168.1.77]
```

```
HTTP_SERVERS [192.168.2.0/24,192.168.12.0/24]
```

Để sử dụng một biến IP trong một luật cần sử dụng dấu đô la (\$) và theo sau là tên biến. Trong trường hợp các luật dưới đây, cả hai biến \$SMTP\_SERVERS và \$EXTERNAL\_NET được sử dụng để phát hiện SMTP AUTH LOGON bruteforce.

```
alert tcp $SMTP_SERVERS 25 -> $EXTERNAL_NET any (msg:"GPL SMTP AUTH  
LOGON brute force attempt"; flow:from_server,established; content:"Authentication  
unsuccessful"; offset:54; nocase; threshold:type threshold, track by_dst, count 5, seconds  
60; classtype:suspicious-login; sid:2102275; rev:3;)
```

Hai biến mạng quan trọng nhất là \$HOME\_NET và \$EXTERNAL\_NET. Biến \$HOME\_NET được sử dụng để xác định khoảng địa chỉ IP mà Snort/Suricata cần bảo vệ. Ví dụ về khai báo \$HOME\_NET như sau:

Snort:

```
ipvar HOME_NET [192.168.0.0/16,10.0.0.0/8,172.16.0.0/12]
```

Suricata:

vars:

address-groups:

```
HOME_NET [192.168.0.0/16,10.0.0.0/8,172.16.0.0/12]
```

Biến \$EXTERNAL\_NET được sử dụng để xác định phạm vi địa chỉ IP không được bảo vệ bởi Snort/Suricata.

Snort:

```
ipvar EXTERNAL_NET! $ HOME_NET
```

Suricata:

```
vars:
```

```
address-groups:
```

```
EXTERNAL_NET! $HOME_NET
```

Ngoài ra, một số biến khác cũng thường được dùng như:

- \$HTTP\_SERVERS - hữu ích cho việc tạo và triển khai các luật liên quan đến khai thác web phía máy chủ hoặc máy khách.
- \$DNS\_SERVERS - hữu ích cho việc tạo và triển khai các luật liên quan đến danh tiếng tên miền hoặc C&C mã độc.
- \$SMTP\_SERVERS - hữu ích cho việc tạo và triển khai các luật liên quan đến thư rác hay tệp tin đính kèm độc hại.
- \$SSH\_SERVERS - hữu ích cho hoạt động log gói tin liên quan đến việc quản lý các thiết bị chuyển mạch, định tuyến, và các thiết bị mạng khác thông qua giao thức SSH.

Người dùng cũng có thể tạo ra các biến riêng bằng cách sử dụng cú pháp này.

## Biến cổng

Biến cổng xác định một cổng tầng 4 (tầng giao vận) hoặc dải cổng dùng trong các luật IDS khi đề cập đến các cổng nguồn hoặc đích của lưu lượng đang kiểm tra.

Với Snort, các biến này được tạo ra bằng cách sử dụng từ khóa *portvar* trong snort.conf. Sau đây là một số ví dụ:

Xác định một cổng duy nhất mà được sử dụng bởi dịch vụ SMTP:

```
portvar SMTP_PORTS 25
```

Xác định hai cổng được sử dụng phỏ biến bởi các dịch vụ FTP:

```
FTP_PORTS portvar 20:21
```

Khai báo một số cổng mà có thể được sử dụng để giao tiếp HTTP:

```
portvar HTTP_PORTS [80,81,82,83,84,85,86,87,88,89,311,383,591,593,631,  
901,1220,1414,1741,1830,2301,2381,2809,3037,3057,3128,3702,4343,4848,5250,6080,  
6988,7000,7001,7144,7145,7510,7777,7779,8000,8008,8014,8028,8080,  
8085,8088,8090,8118,8123,8180,8181,8222,8243,8280,8300,8500,8800,8888,  
8899,9000,9060,9080,9090,9091,9443,9999,10000,11371,34443,34444,41080,  
50002,55555]
```

## Biến chuẩn

Biến chuẩn là loại biến chỉ được sử dụng bởi Snort. Các biến này được tạo ra bằng cách sử dụng các từ khóa `var`, và thường được sử dụng để chỉ định thư mục. Ví dụ, để xác định các thư mục có chứa các loại khác nhau của các quy tắc Snort:

```
var RULE_PATH /etc/NSM/rules  
var SO_RULE_PATH /etc/NSM/rules  
var PREPROC_RULE_PATH /etc/NSM/rules
```

Phần lớn các khai báo biến có thể được tìm thấy trong phần đầu tiên của snort.conf.

### 3.3.5.2 Xác định các tập luật

Đối với Snort hoặc Suricata, để kiểm tra lưu lượng mạng cho IOC, cần phải có các luật. Các luật trong Snort và Suricata là phương pháp xây dựng các IOC theo nền tảng cụ thể (platform-specific). Các luật chỉ dẫn cho công cụ phát hiện cách thức xác định vị trí IOC trong lưu lượng mạng.

Các luật được ghi trong các tệp tin luật, là tệp tin văn bản có chứa luật theo một định dạng phân cách theo từng dòng. Để cho Snort hoặc Suricata phân tích được luật, chúng phải được chỉ dẫn trong các tệp tin cấu hình tương ứng.

#### Xác định tệp tin luật Snort

Trong snort.conf, phần cuối của tệp tin cấu hình là nơi khai báo các luật. Chúng ta cần chỉ định một thư mục luật và đường dẫn cùng tên tệp tin luật. Ví dụ:

```
include $RULE_PATH/emerging-exploit.rules
```

Snort cũng cho phép sử dụng các loại luật không tiêu chuẩn. Bao gồm:

- Luật tiền xử lý: Các luật này phụ thuộc vào chức năng cung cấp bởi các bộ tiền xử lý, và được phân tích trước các luật được phân tích bởi engine phát hiện.
- Luật đối tượng chia sẻ: Các luật này được biên dịch thay vì được thông dịch từ một dòng văn bản. Chúng rất có ích trong việc tạo ra các luật tiên tiến, hoặc triển khai các luật mà không tiết lộ các chi tiết của các IOC trong luật.

Những luật này có thể được đặt tại các vị trí khác nhau, do đó, chúng có biến đường dẫn luật riêng. Tệp tin luật có thể được khai báo sử dụng các biến này:

```
include $PREPROC_RULE_PATH/preproc.rules  
include $SO_RULE_PATH/sharedobj.rules
```

#### Xác định tệp tin luật Suricata

Với Suricata, tệp tin luật được xác định bằng cách đặt chúng vào phần thích hợp của suricata.yaml. Để làm điều này, đường dẫn luật mặc định phải được xác định, sau đó các tệp tin

luật có thể được liệt kê dưới tiêu đề *rule-files*, với mỗi tệp tin được xác định trên một dòng mới với một dấu gạch ngang.

```
default-rule-path: /etc/nsm/rules/
rule-files:
- local.rules
- downloaded.rules
```

### Các nguồn luật công cộng

Luật có thể tự tạo một cách thủ công, chia sẻ giữa các tổ chức, hoặc lấy từ các nguồn công cộng. Có hai nguồn chính cho luật trong Snort và Suricata: Emerging Threats (ET) và Sourcefire VRT.

ET ban đầu được gọi là Bleeding Snort, ban đầu được đưa ra vào năm 2003 bởi Matt Jonkman, và được thiết kế thành một cộng đồng mã nguồn mở chia sẻ chữ ký IDS. Hiện nay, cộng đồng ET mạnh mẽ hơn bao giờ hết và cung cấp các bộ quy tắc cho cả Snort và Suricata.

VRT, từ cùng một công ty tạo ra Snort, có trách nhiệm cho sự phát triển và duy trì các tập luật chính thức trong Snort.org.

Trong khi Sourcefire VRT không cung cấp một tập luật Suricata cụ thể nào thì một số luật trong đó sẽ vẫn hoạt động với Suricata. Tuy nhiên, Suricata không hỗ trợ nhiều tùy chọn luật được cung cấp bởi các bộ tiền xử lý Snort.

Có thể tải về các luật Snort VRT ở <http://www.snort.org/snort-rules/>.

### Quản lý cập nhật luật với PulledPork

Các nguồn như VTR phát hành luật mới gần như mỗi ngày. Nhiệm vụ kiểm tra các bản cập nhật luật mới, tải những bản cập nhật, đặt chúng trong thư mục thích hợp, và đảm bảo rằng chúng được đưa vào hoạt động có thể rất tẻ nhạt nếu được thực hiện bằng tay.

PulledPork đã được tạo ra để tự động hóa quá trình này, và nó có thể được sử dụng để đảm bảo rằng các luật luôn được cập nhật. Nó cung cấp một loạt các tính năng rất hữu ích cho một số tình huống. Trong đó bao gồm việc cung cấp cơ chế tải bản cập nhật luật, khả năng quản lý và phân phối các tệp tin luật tùy chỉnh, và khả năng để theo dõi các thay đổi luật. Có thể đọc thêm thông tin tại <https://code.google.com/p/pulledpork/>.

### Quản lý luật trong Security Onion

Theo mặc định, các luật trong Security Onion được đặt trong */etc/NSM/rules/*. Các luật được tải về từ các nguồn công cộng như Sourcefire VRT được đặt vào tệp tin *downloaded.rules*, và luật được tùy chỉnh nên được đặt vào tệp tin *local.rules*. Tệp tin luật bổ sung có thể được sử dụng, nhưng lần đầu phải được quy định trong *snort.conf* hoặc *suricata.yaml*.

Nếu đang sử dụng SO làm nền tảng NSM, ta nên tránh việc cập nhật các quy tắc bằng cách sử dụng các phương pháp đã đề cập trước đó, và thay vào đó sử dụng script rule-update. Script

này thực hiện các nhiệm vụ theo yêu cầu của các công cụ khác như Barnyard2 và PulledPork. Các script được chạy như sau:

```
sudo rule-update
```

Có hai tệp tin đặc biệt quan trọng đối với việc duy trì các luật trong SO là `disablesid.conf` và `modifysid.conf`. Những tập tin này là một phần của PulledPork. Tệp tin `disablesid.conf` được sử dụng để vô hiệu hóa các quy tắc không muốn sử dụng. Điều này đặc biệt quan trọng khi tương tác với các luật công cộng vì chúng được cập nhật liên tục. Các tệp tin `modifysid.conf` được sử dụng để liên tục thay đổi các luật được lấy từ các nguồn công cộng. Cũng như với các luật đã xóa, nếu sửa một luật lấy từ một nguồn nào đó, bản cập nhật PulledPork hàng đêm sẽ phục vụ để thay thế tệp tin luật và loại bỏ bất kỳ thay đổi đã được thực hiện. Bởi vậy, PulledPork phân tích `modifysid.conf` sau mỗi lần cập nhật luật để có thể quay trở lại và áp dụng các sửa đổi tới các luật đã được chỉnh sửa.

### 3.3.5.3 Đầu ra cảnh báo

Snort và Suricata đều rất linh hoạt trong cách xử lý dữ liệu cảnh báo có thể được xuất ra để phân tích, do đó hữu ích cho việc áp dụng chúng vào một loạt các tình huống.

Trong Snort, đầu ra cảnh báo được kiểm soát trong phần plugin đầu ra của `snort.conf`. Để chỉ định một plugin đầu ra cụ thể chúng ta có thể sử dụng các từ khóa đầu ra, theo sau là tên của plugin và các tùy chọn.

```
output < plugin name >: < options >
```

Nếu không được quy định tại thời gian chạy với tham số `-l`, thư mục log mặc định của Snort sẽ là `/var/log/snort`.

Trong Suricata, đầu ra cảnh báo được kiểm soát trong phần kết quả đầu ra của `Suricata.yaml`. Bên dưới tiêu đề `outputs`, mỗi tùy chọn đầu ra được liệt kê, cùng với các tùy chọn liên quan tương ứng.

`outputs:`

```
- < output type >:  
  < options >
```

Nếu không được quy định tại thời gian chạy với tham số `-l`, thư mục log mặc định của Suricata sẽ là `/var/log/suricata`.

### 3.3.5.4 Các bộ tiền xử lý của Snort

Trong khi phần lớn các tính năng của Suricata được xây dựng với kiến trúc cốt lõi của nó, đa phần trong số các tính năng được cung cấp bởi Snort được xây dựng bằng cách sử dụng các bộ tiền xử lý riêng rẽ. Trong kiến trúc Snort, tiền xử lý có hai loại và có thể được sử dụng cho dữ liệu đã chuẩn hóa, trước khi nó được phân tích bởi các công cụ phát hiện, có thể được sử dụng để cung cấp thêm tính linh hoạt cho các luật Snort sử dụng bởi các công cụ phát hiện. Cả hai loại đều có thể được cấu hình trong `snort.conf`. Một bộ tiền xử lý được xác định bởi các từ khóa tiền xử lý,

tiếp theo là tên tiền xử lý và sau đó là các lựa chọn liên quan. Một số tiền xử lý như phát hiện portscan, chỉ có một vài tùy chọn cấu hình.

```
# Portscan detection. For more information, see README.sfportscan  
# preprocessor sfportscan: proto { all } memcap { 10000000 } sense_level { low }
```

Những cái khác, chẳng hạn như tiền xử lý phát hiện SSH bất thường, có nhiều lựa chọn:

```
# SSH anomaly detection. For more information, see README.ssh  
preprocessor ssh: server_ports {22}  
autodetect  
max_client_bytes 19600  
max_encrypted_packets 20  
max_server_version_len 100  
enable_respoverflow enable_ssh1crc32  
enable_srvoverflow enable_protomismatch
```

Điều quan trọng là các bộ tiền xử lý liệt kê trong tệp tin cấu hình được thực hiện theo thứ tự. Nên tận dụng lợi thế của các bộ tiền xử lý vì có lúc có thể thấy một bộ tiền xử lý sẽ làm cho việc viết một luật phức tạp trở nên đơn giản hơn nhiều. Ví dụ:

- Danh tiếng: Được sử dụng để phát hiện và ngăn chặn các liên lạc với các địa chỉ IP nhất định dựa trên danh tiếng.
- arpspoof: Được thiết kế để phát hiện sự xuất hiện của ARP spoofing.
- SFportscan: Phát hiện quét trinh sát.
- Frag3: Thực hiện chống phân mảnh gói tin IP và ngăn chặn việc tránh né IDS.
- Stream5: Cho phép theo dõi trạng thái của các kết nối TCP và tạo ra các luật có trạng thái.
- HTTP\_Inspect: chuẩn hóa lưu lượng HTTP để nó có thể được phân tích đúng đắn với các công cụ phát hiện. Cung cấp một số chỉ dẫn có thể sử dụng được trong luật của Snort.

### 3.3.6 Các luật IDS

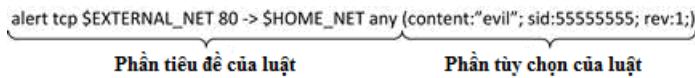
#### 3.3.6.1 Phân tích về luật

Cú pháp sử dụng bởi các luật Snort và Suricata là vô cùng linh hoạt, nhưng nó đòi hỏi các thông lệ nhất định phải được tuân thủ. Ví dụ về một luật rất đơn giản:

```
alert tcp $EXTERNAL_NET 80 -> $HOME_NET any (msg:"Users Downloading Evil";  
content:"evil"; sid:55555555; rev:1;)
```

Luật này là rất cơ bản là sẽ tạo ra một cảnh báo nếu một người dùng trên mạng nội bộ tải dữ liệu từ một máy chủ web có chứa từ "evil". Tuy nhiên, việc phát hiện khi người dùng tải về những dữ liệu xấu từ Internet không phải là dễ dàng.

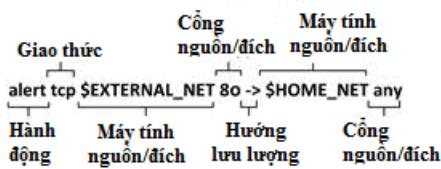
Trước khi kiểm tra từng thành phần cụ thể của luật này, cần thấy rằng các luật có hai phần riêng biệt: tiêu đề luật và các tùy chọn. Các tiêu đề luật là các mục trước dấu ngoặc, và các tùy chọn là các mục trong dấu ngoặc đơn (xem Hình 3.19).



**Hình 3.19 Phân tích luật cơ bản**

### a. Tiêu đề luật

Tiêu đề luật luôn luôn là phần đầu tiên của luật và nó là phần bắt buộc của luật. Tiêu đề có nhiệm vụ xác định "ai" có liên quan tới mẫu lưu lượng. Tất cả mọi thứ được định nghĩa trong tiêu đề luật có thể được tìm thấy trong tiêu đề của gói tin. Chúng rất quan trọng trong việc phân tích các luật này. Các thành phần trong tiêu đề luật được thể hiện trong Hình 3.20.



**Hình 3.20 Phân tiêu đề luật IDS**

Tiêu đề luật thường bao gồm các thành phần tương tự nhau: hành động của luật, giao thức, máy tính nguồn/đích, cổng nguồn/đích, và hướng lưu lượng.

**Hành động của luật.** Phần đầu tiên của bất kỳ luật nào là khai báo hành động để báo cho IDS engine phải làm gì khi có cảnh báo. Có ba hành động có thể có:

- Cảnh báo: Báo cho IDS engine ghi log các luật tìm thấy kết quả phù hợp, và các dữ liệu gói tin liên quan kết quả. Đây là hành động luật phổ biến nhất.
- Ghi log: Báo cho IDS engine ghi log các luật tìm thấy kết quả phù hợp, nhưng không log lại dữ liệu gói tin liên quan.
- Bỏ qua: Báo cho IDS engine không xử lý các gói tin.

**Giao thức.** Trường này báo cho IDS engine về giao thức luật sẽ áp dụng cho. Các lựa chọn hợp lệ bao gồm tcp, idp, icmp, ip, và any. Lưu ý rằng chỉ có một trong những lựa chọn đó được sử dụng, vì vậy nếu muốn viết một luật áp dụng cho cả hai giao thức TCP và UDP, hãy sử dụng tùy chọn giao thức IP trong tiêu đề luật.

**Máy tính nguồn và đích.** Để tạo ra một luật, có thể chỉ định các máy tính nguồn và đích cho các mẫu lưu lượng đang cần so sánh. Các máy tính này phải được xác định bằng địa chỉ IP. Có thể dùng any để chỉ địa chỉ IP bất kỳ.

**Cổng nguồn và đích.** Cùng với xác định các máy tính, cũng có thể chỉ định cụ thể cổng ở tầng bốn. Hãy nhớ rằng, chúng có thể được quy định như các cổng riêng lẻ, theo danh sách, hoặc theo phạm vi. Trong trường hợp khi không có cổng cụ thể được chỉ ra, từ khóa "any" có thể được sử dụng để phù hợp với bất kỳ cổng nào.

**Hướng lưu lượng.** Cuối cùng là xác định đích đến của lưu lượng. Chỉ có hai tùy chọn ở đây:

- ->: Xác định lưu lượng đơn hướng từ nguồn tới đích
- <>: Xác định lưu lượng hai chiều

Vì chỉ có hai lựa chọn ở đây, khi viết các luật cần phải xem xem liệu hướng của lưu lượng có phải là vấn đề hay không. Nếu hướng của lưu lượng không quan trọng thì thứ tự của máy tính nguồn và đích cùng với số hiệu cổng trong tiêu đề là không quan trọng. Tuy nhiên, nếu hướng là cần thiết thì máy tính và số hiệu cổng nguồn nên được liệt kê đầu tiên.

## b. Các tùy chọn của luật

Trong khi phần tiêu đề luật là chịu trách nhiệm cho "ai", phần tùy chọn luật có trách nhiệm khai báo "cái gì." Phần này cho các công cụ IDS biết chính xác những gì nó tìm kiếm trong các gói được kiểm tra, và làm thế nào để tìm thấy nó. Nội dung của phần tùy chọn của luật đều có thể thay đổi và có thể bao gồm một số, nhưng cho dù chọn bất kỳ cái gì để khai báo, phần lựa chọn phải luôn luôn được đặt trong ngoặc đơn. Trong các dấu ngoặc đơn, tùy chọn riêng lẻ có dạng:

<Option>: <giá trị tùy chọn>;

Tên tùy chọn và giá trị của nó được phân cách bằng dấu hai chấm (:) và các giá trị tùy chọn đều được kết thúc bằng một dấu chấm phẩy (;). Nếu các giá trị tùy chọn chứa các khoảng trống, những giá trị đó phải được đặt trong dấu ngoặc kép.

Trong một số trường hợp, tùy chọn này sẽ không có giá trị, và chỉ đơn giản là gọi như sau:

<Option>;

Chú ý rằng tên các tùy chọn được kết thúc bằng một dấu chấm phẩy (;). Nếu không đặt dấu hai chấm hoặc dấu chấm phẩy theo yêu cầu, các công cụ IDS đang sử dụng sẽ không chạy khi phân tích luật đó.

Phần sau sẽ xem xét một số tùy chọn phổ biến trong luật.

### Các tùy chọn thông tin sự kiện

Các tùy chọn thông tin sự kiện được sử dụng để cung cấp thông tin theo ngữ cảnh về một luật. Thông tin sự kiện càng chi tiết thì càng hiệu quả khi phân tích điều tra dữ liệu kết hợp với cảnh báo. Một số tùy chọn như sau:

**Message (msg).** Đoạn mô tả kết hợp với luật. Ví dụ:

- ET POLICY Outgoing Basic Auth Base64 HTTP Password detected unencrypted
- OS-WINDOWS SMB NTLM NULL session attempt

- EXPLOIT-KIT Blackholev2 exploit kit jar file downloaded

**Nhận dạng chữ ký (sid).** Được sử dụng để nhận diện ra các luật. Mỗi luật phải có một SID duy nhất, mà chỉ đơn giản là một giá trị số. Điều quan trọng là cần lưu ý là một số phạm vi được coi là dành riêng.

- 0-1000000: Dành cho các Sourcefire VRT
- 2000001-2999999: Được sử dụng bởi các nguy cơ mới
- 3000000 +: Để sử dụng công cộng

**Revision (rev).** Tùy chọn phiên bản được sử dụng để biểu thị khi một luật đã được thay đổi. Khi một luật mới được tạo ra, nó phải được chỉ định rev: 1; để chỉ ra rằng nó là phiên bản đầu tiên của luật. Thay vì tạo ra một SID mới mỗi lần một luật được thay đổi, nên giữ lại cùng một SID và tăng số phiên bản. Trong trường hợp Snort hoặc Suricata gặp phải một bản sao SID, chúng sẽ sử dụng các luật với số phiên bản cao hơn.

**Reference.** Từ khóa tham khảo cung cấp khả năng liên kết đến các nguồn thông tin bên ngoài để cung cấp bối cảnh bổ sung cho các luật. Cách phổ biến nhất để làm điều này là để chỉ đơn giản bao gồm một tham chiếu đến một URL, như thể hiện trong luật sau:

```
alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS (msg:"ET
CURRENT_EVENTS FakeAlert/FraudPack/FakeAV/Guzz/Dload/Vobfus/ZPack HTTP
Post 2"; flow:established,to_server; content:"POST"; http_method; content:"/perce/"; nocase; http_uri; content:"/qwerce.gif"; nocase; http_uri; content:"data = "; nocase; reference:url,threatinfo.trendmicro.com/vinfo/virusencyclo/default5.asp?VName = TROJ_AGENT.GUZZ&VSect = T; reference:url,www.threatexpert.com/threats/trojan-fraudpack-sd6.html; reference:url,vil.nai.com/vil/content/v_157489.htm; reference:url,doc.emergingthreats.net/2010235; classtype:trojan-activity; sid:2010235; rev:6;)
```

Luật trên được sử dụng để phát hiện sự hiện diện của một số phần mềm độc hại. Trong trường hợp này, luật tham chiếu bốn tài liệu tham khảo:

- reference:url,threatinfo.trendmicro.com/vinfo/virusencyclo/default5.asp?VName = TROJ\_AGENT.GUZZ&VSect = T;
- reference:url,www.threatexpert.com/threats/trojan-fraudpack-sd6.html;
- reference:url,vil.nai.com/vil/content/v\_157489.htm;
- reference:url,doc.emergingthreats.net/2010235;

Lưu ý rằng tài liệu tham khảo theo định dạng sau:

reference: <reference name>,<reference>;

Kiểu tham chiếu được định nghĩa trong tệp tin reference.config được sử dụng bởi Snort và Suricata. Tên và vị trí của tập tin này được cấu hình trong snort.conf và suricata.yaml. Trong SO, nó nằm trong /etc/NSM/<tên cảm biến>/reference.config.

### Kiểm tra nội dung

Các hành động cơ bản nhất có thể được thực hiện trong phần tùy chọn của một luật IDS là thực hiện so sánh nội dung cơ bản. Sử dụng từ khóa nội dung, có thể hướng dẫn công cụ IDS kiểm tra nội dung (payload) của một gói tin trên tầng ứng dụng với các dữ liệu chỉ định.

Ví dụ, nếu muốn kiểm tra nội dung của một gói tin có chuỗi "evilliveshere", thì có thể chỉ định: content:

```
"evilliveshere";
```

Cũng có thể chỉ định nhiều so sánh nội dung trong một luật duy nhất:

```
content:"evillives"; content:"here";
```

Dữ liệu nhị phân cũng có thể được so sánh bằng cách diễn tả dữ liệu nhị phân như là các ký tự hệ cơ số 16 được bao quanh bởi các ký hiệu số thăng (|). Nếu muốn kiểm tra dữ liệu gói tin cho sự tồn tại của các tệp tin JPEG, cần tìm số hệ cơ số 16 đại diện cho JPEG:

```
content:"|FF D8|";
```

Dữ liệu chuỗi và dữ liệu nhị phân có thể được kết hợp trong các tìm kiếm nội dung. Trong ví dụ sau, tìm kiếm ba dấu hai chấm, tiếp theo là văn bản "evilliveshere", tiếp theo là ba byte rỗng (null):

```
content:"|3A 3A 3A|evilliveshere|00 00 00|";
```

### Sửa đổi kiểm tra nội dung

Có một số sửa đổi kiểm tra nội dung có thể được áp dụng cho các so sánh nội dung bằng cách đặt chúng sau khi nội dung được xuất hiện. Nó cho phép xác định chính xác cách mà các công cụ IDS sẽ tìm kiếm so sánh nội dung bên trong dữ liệu mạng. Các bộ sửa đổi giúp tăng độ chính xác của so sánh nội dung trong luật, và cũng giúp tăng hiệu suất của quá trình phát hiện trong công cụ IDS, bởi vì chúng cho phép các công cụ tìm trong một vị trí cụ thể cho các nội dung cụ thể hơn là phải kiểm tra hoàn toàn payload của mỗi gói.

Để áp dụng một bộ sửa đổi nội dung vào một so sánh nội dung, nên đặt trực tiếp sau các so sánh nội dung trong luật. Có một số bộ sửa đổi sau đây:

**Nocase.** So sánh không ảnh hưởng chữ thường/hoa.

**Offset và độ sâu.** Các sửa đổi offset được sử dụng để phù hợp với nội dung xảy ra ở một vị trí cụ thể với một payload của gói tin, bắt đầu từ byte đầu tiên của payload. Lưu ý rằng các payload bắt đầu tại byte 0, chứ không phải là byte 1. Vì vậy, nếu chỉ rõ offset 0, các công cụ phát hiện sẽ tìm kiếm nội dung để bắt đầu vào phần đầu của payload. Ví dụ, kiểm tra các gói dữ liệu FTP như sau:

```
14:51:44.824713 IP 172.16.16.139.57517 > 67.205.2.30.21: Flags [P.], seq 1:15, ack 27,  
win 16421, length 14  
0x0000: 4510 0036 efe4 4000 4006 4847 ac10 108b E..@.@@.HG....  
0x0010: 43cd 021e e0ad 0015 0bcb 6f30 fcb2 e53c C.....o0...<  
0x0020: 5018 4025 2efb 0000 5553 4552 2073 616e P.%....USER.san  
0x0030: 6465 7273 0d0a ders..
```

Nếu muốn viết một luật kết hợp nội dung, được phát hiện bát cứ khi một người dùng có gắng đăng nhập vào máy chủ FTP bên ngoài với tên người dùng này, thì có thể bắt đầu với một luật như sau:

```
alert tcp $HOME_NET any - > 67.205.2.30 21 (msg:"Suspicious FTP Login";  
content:"sanderson"; sid:5000000; rev:1;)
```

Luật này chắc chắn sẽ tạo ra một cảnh báo cho các gói tin được đưa ra, nhưng nó cũng dễ bị dương tính giả. Ví dụ, nếu ai đó đăng nhập vào một tài khoản khác trên máy chủ FTP và duyệt đến một thư mục có tên "sanderson", đó cũng sẽ tạo ra một cảnh báo.

Có thể thu hẹp phạm vi của luật này bằng cách xác định offset nơi tên người dùng xuất hiện trong payload của gói tin. Trong trường hợp này, các byte đầu tiên của payload gói tin là 0x55. Các ký tự đầu tiên của tên người dùng thực tế xuất hiện tại offset 5 (0x73). Chú ý là đang tính bắt đầu từ số không. Như vậy, có thể viết lại các luật để bắt đầu phù hợp với chuỗi nội dung tại offset đó:

```
alert tcp $HOME_NET any - > 67.205.2.30 21 (msg:"Suspicious FTP Login"  
content:"sanderson"; offset:5; sid:5000000; rev:1;)
```

Không chỉ luật ít sai, mà đồng thời cũng sẽ thực hiện nhanh hơn vì nó hạn chế số lượng các byte mà công cụ IDS phải kiểm tra.

Trong khi các sửa đổi offset có thể được sử dụng để chỉ định vị trí công cụ IDS bắt đầu tìm kiếm một so sánh nội dung, sửa đổi độ sâu có thể được sử dụng để xác định vị trí ngừng tìm kiếm một so sánh nội dung. Điều này được thực hiện bằng cách xác định các byte liên quan đến nội dung byte payload đầu tiên được kiểm tra. Nếu không sử dụng các sửa đổi offset, độ sâu sẽ được tính tương đối với byte đầu tiên của payload gói tin. Nếu đang sử dụng sửa đổi offset, độ sâu sẽ được tính tương đối đến các byte được chỉ định trong các giá trị offset.

Nếu xem xét luật Snort tạo ra trong ví dụ đăng nhập FTP trước đó, chúng ta có thể làm cho nó thậm chí còn hiệu quả hơn bằng cách hạn chế độ sâu. Ở đây, đã giới hạn độ sâu đến 6 byte, là độ dài của chuỗi đang cố gắng để so sánh (chú ý một lần nữa là đang đếm từ 0). Trong trường hợp này, đã kết hợp các sửa đổi offset và độ sâu để xác định vị trí tuyệt đối của nội dung đang cố gắng so sánh.

```
alert tcp $HOME_NET any - > 67.205.2.30 21 (msg:"Suspicious FTP Login"  
content:"sanderson"; offset:5; depth:7; sid:5000000; rev:1;)
```

## Luồng thông tin

Để hiểu được tùy chọn về luồng làm việc như thế nào và tại sao chúng lại quan trọng, cần nhớ lại những gì tạo nên một phiên TCP. Trong một phiên TCP bình thường, có một máy khách và máy chủ giao tiếp với nhau. Máy khách là thiết bị bắt đầu kết nối đến máy chủ bằng cách gửi một gói tin SYN đến máy chủ trên một cổng đang lắng nghe. Máy chủ tại thời điểm đó sẽ phản hồi cho máy khách với một gói tin SYN/ACK. Khi nhận được, máy khách sẽ phản hồi lại cho máy chủ một gói tin ACK. Tại thời điểm này, giao thức bắt tay ba bước đã được hoàn tất, máy khách và máy chủ có thể trao đổi cho đến khi một trong số chúng chấm dứt kết nối theo một trong hai cách, hoặc đột ngột ngắt với một gói tin RST, hoặc bình thường hơn với một loạt các gói tin FIN. Đây là tiền đề cơ bản của những gì làm nên một phiên TCP.

Như vậy, tùy chọn luồng cho luật có một vài lựa chọn của riêng mình. Chúng được chia thành ba loại: tùy chọn trạng thái, tùy chọn hướng, và trạng thái mô hình hóa lưu lượng. Các tùy chọn này được cấu hình bằng cách sử dụng định dạng sau. Cần ít nhất một tùy chọn và bổ sung khác là tùy ý:

flow: <option>, <option>, <option>;

Hai tùy chọn trạng thái là sẵn sàng được thiết lập và không có trạng thái. Tùy chọn được thiết lập sẽ chỉ phù hợp với lưu lượng có sự tồn tại của phiên TCP. Tùy chọn không trạng thái sẽ thỏa mãn dù có kết nối thiết lập hay không.

Có bốn tùy chọn hướng:

- to\_server: lưu lượng từ máy khách đến máy chủ
- from\_server: lưu lượng từ máy chủ đến máy khách
- to\_client: lưu lượng từ máy chủ đến máy khách
- from\_client: lưu lượng từ máy khách đến máy chủ

## Các tùy chọn phát hiện tiêu đề giao thức

Snort và Suricata cung cấp khả năng phát hiện các giá trị trong các tiêu đề của các gói dữ liệu được kiểm tra. Nó bao gồm hầu hết các giá trị trong các tiêu đề ICMP, IP, TCP, UDP. Một số các giá trị được sử dụng nhiều nhất gồm:

- TTL: so sánh với một giá trị TTL được xác định. Có thể là một giá trị chính xác (=) hoặc sử dụng một toán tử quan hệ (>, >=, <, <=). Rất hữu ích cho việc phát hiện một số loại hệ điều hành dựa trên giá trị TTL ban đầu của chúng.
- dsize: so sánh một gói tin với một kích thước payload cụ thể. Có thể được xác định với một giá trị chính xác (=) hoặc sử dụng một toán tử quan hệ (>, <). Rất hữu ích để tăng hiệu suất luật bằng cách kết hợp nó với các luật so sánh nội dung.
- itype: so sánh với một giá trị kiểu ICMP cụ thể.

- icode: so sánh với một giá trị mã ICMP cụ thể.
- ip\_proto: so sánh với một giao thức IP cụ thể. Có thể được xác định theo tên (IGMP, GRE, vv) hoặc số hiệu giao thức.

### 3.3.6.2 Tinh chỉnh luật

Phần này giới thiệu một số phương pháp được sử dụng để tăng hiệu suất của luật.

#### Lọc sự kiện

Đôi khi các luật có thể tạo ra một số lượng lớn các cảnh báo, ví dụ như luật phát hiện một loại tấn công từ chối dịch vụ (DoS) cụ thể. Điều quan trọng để có thể phát hiện kiểu tấn công này là cần phải viết các luật phù hợp với tất cả các gói DoS gửi tới, và khi nhận được hàng ngàn những gói dữ liệu mỗi giây, thì sau đó sẽ nhận được hàng ngàn cảnh báo mỗi giây. Và các cảnh báo này sẽ lấn át công cụ IDS hoặc các chuyên gia phân tích. Các tùy chọn lọc sự kiện được cung cấp bởi Snort và Suricata cho phép áp dụng các ngưỡng vào luật để ngăn chặn loại cảnh báo tràn ngập kiểu này.

Các bộ lọc sự kiện được thiết kế để được đặt trong tệp tin threshold.conf. Tên và vị trí của tập tin này được cấu hình trong snort.conf và suricata.yaml khi cần. Trong SO, tập tin này được lưu trữ tại /etc/NSM/<tên cảm biến>/threshold.conf.

#### Ngăn chặn cảnh báo

Tính năng này cho phép xác định một luật và một địa chỉ IP (hoặc nhóm các địa chỉ IP từ một biến), và ngăn chặn các thông báo sẽ được tạo ra từ những máy có liên quan với một luật.

Các mục ngăn chặn cảnh báo cũng được khai báo trong tệp tin threshold.conf, và có cú pháp như sau:

```
suppress gen_id < value >,sig_id < value >,track < by_s
rc|by_dst >,ip < value >
```

Các mục sau đây sẽ được sử dụng để ngăn chặn bất kỳ cảnh báo nào được tạo ra bởi SID 5000000 với các địa chỉ IP nguồn là 192.168.1.100:

```
suppress gen_id 1, sig_id 5000000, track by_src, ip 192.168.1.100
```

#### Bộ lọc phát hiện cảnh báo

Snort và Suricata cung cấp khả năng sử dụng các bộ lọc phát hiện để thiết lập một ngưỡng trên một số so sánh luật cần phải xảy ra trước khi cảnh báo được tạo ra. Một bộ lọc phát hiện có thể được áp dụng cho một luật dựa trên địa chỉ nguồn hoặc địa chỉ đích của lưu lượng, và có thể áp dụng ngưỡng của nó dựa trên số lượng các so sánh luật được phát hiện trong một khoảng thời gian xác định.

Các tùy chọn bộ lọc phát hiện được áp dụng phù hợp với luật và có định dạng sau:

```
detection_filter: track < by_src|by_dst >, count < value >, seconds < value >;
```

Ví dụ về bộ lọc phát hiện trong thực tế như sau:

```
alert tcp $EXTERNAL_NET any ->$HTTP_SERVERS $HTTP_PORTS (msg:"ET  
SCAN Sqlmap SQL Injection Scan"; flow:to_server,established; content:"User-  
Agent|3a| sqlmap"; fast_pattern:only; http_header; detection_filter:track by_dst, count  
4, seconds 20; reference:url,sqlmap.sourceforge.net;  
reference:url,doc.emergingthreats.net/2008538; classtype:attempted-recon;  
sid:2008538; rev:8;)
```

Luật này được sử dụng để phát hiện hoạt động quét dùng công cụ Sqlmap, được sử dụng để phát hiện và dàn xếp các cuộc tấn công SQL injection. Trong trường hợp này, các luật phù hợp với nội dung liên quan đến các user agent được sử dụng bởi Sqlmap. Nói chung, việc nhìn thấy user agent này chỉ một hoặc hai lần có thể không chỉ ra bất kỳ loại hoạt động quét nào do thông tin Sqlmap thường chi tiết hơn. Và như vậy, việc tạo ra một cảnh báo mỗi khi sử dụng user agent này được cho là có thể tạo ra một số lượng đáng kể trường hợp dương tính giả. Vậy nên, các luật cần được cấu hình với các bộ lọc phát hiện như sau:

```
detection_filter:track by_dst, count 4, seconds 20;
```

Bộ lọc phát hiện này yêu cầu đáp ứng một ngưỡng xác định trước khi bộ phát hiện tạo ra một cảnh báo từ luật. Cụ thể, các công cụ phát hiện sẽ theo dõi số lượng các luật phù hợp với địa chỉ đích, và khi con số này vượt quá bốn trong khoảng thời gian hai mươi giây, nó sẽ tạo ra một cảnh báo.

### Các phương pháp khác

Một số phương pháp tinh chỉnh khác bao gồm:

- Loại bỏ lưu lượng không mong muốn
- Nhắm tới lỗ hổng
- Ghép cặp cho PCRE và các so sánh nội dung
- So sánh mẫu nhanh
- Kiểm tra các luật theo cách thủ công

Tuy nhiên do không gian có hạn nên không thể giới thiệu các phương pháp này chi tiết ở đây, có thể tham khảo thêm trong tài liệu [1].

#### 3.3.7 Xem các cảnh báo của Snort và Suricata

Các cảnh báo của IDS có thể được đọc trực tiếp từ các cảm biến và các tệp tin do Snort và Suricata tạo ra. Ngoài ra có thể sử dụng công cụ đồ họa của bên thứ ba để trợ giúp quá trình này.

#### Snorby

Snorby là một giao diện điều khiển quản lý cảnh báo mới được viết bằng *ruby on rails* và hoạt động trong các trình duyệt web. Snorby đã được tạo ra bởi Dustin Weber. Mục tiêu tổng thể

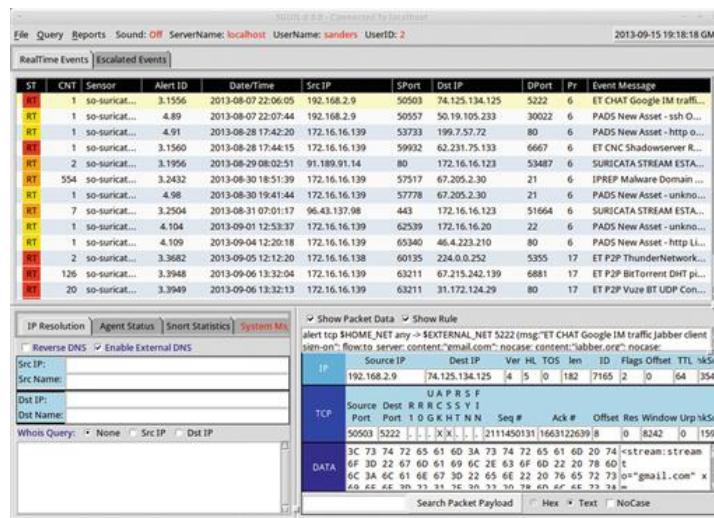
của Snorby là cung cấp cho các chuyên gia phân tích phương tiện để xem xét và phân tích cảnh báo theo cách đơn giản nhưng hiệu quả.

Nếu đang sử dụng SO, có thể truy cập Snorby bằng cách nhấn vào biểu tượng Snorby trên màn hình desktop, hoặc bằng cách truy cập [https://<Security\\_Other\\_IP>:444/](https://<Security_Other_IP>:444/). Hình 3.21 là bảng điều khiển chính của Snorby.



Hình 3.21 Bảng điều khiển của Snorby

## Sguil



Hình 3.22 Giao diện chính của Sguil

Sguil là giao diện điều khiển (console) quản lý cảnh báo chuẩn cho chuyên gia phân tích NSM trong nhiều năm. Không giống như Snorby, Sguil hoạt động dưới dạng ứng dụng máy tính để bàn kết nối với một nguồn dữ liệu trung tâm. Sguil được viết bởi Bamm Visscher, và được duy

trì là một ứng dụng mã nguồn mở miễn phí. Nó được cài đặt trên SO theo chế độ mặc định, và có thể được truy cập bằng cách nhấp vào biểu tượng Sguil. Hình 3.22 là giao diện chính của Sguil.

## 3.4 PHÁT HIỆN XÂM NHẬP DỰA TRÊN BẤT THƯỜNG VỚI DỮ LIỆU THÔNG KÊ

### 3.4.1 Tạo danh sách thống kê với SiLK

Ví dụ đơn giản của dữ liệu thống kê là một danh sách các máy tính giao tiếp trên mạng nội bộ xác định các thiết bị được bảo vệ mà có lưu lượng dữ liệu liên lạc lớn nhất trên một đoạn mạng được giám sát. Đội NSM có thể sử dụng số liệu thống kê này để xác định một số thông tin như các thiết bị có lưu lượng gửi đi đến máy chủ bên ngoài lớn đáng ngờ, hoặc có thể là máy tính được bảo vệ nhưng bị nhiễm phần mềm độc hại kết nối với một lượng lớn các địa chỉ IP bên ngoài đáng ngờ. Đây là một trường hợp mà chữ ký không thể phát hiện ra, bởi vì đây là một bất thường thật sự của mạng.

Khả năng để tạo ra một danh sách như trên có thể là một thách thức khi không có công cụ phù hợp và truy cập được dữ liệu mạng. Tuy nhiên, các công cụ phân tích dữ liệu về phiên như SiLK và Argus làm điều này một cách dễ dàng.

Trong các phần trước, đã thảo luận các phương pháp khác nhau để thu thập dữ liệu phiên và phương pháp cơ bản để phân tích loại dữ liệu này. Phần này sẽ xem xét SiLK, một công cụ được sử dụng hiệu quả cho việc thu thập, lưu trữ và phân tích dữ liệu luồng. Trong SiLK có một số công cụ hữu ích để tạo ra các số liệu thống kê và số liệu cho nhiều tình huống. SiLK hoạt động bằng cách yêu cầu người dùng xác định dữ liệu họ muốn sử dụng làm nguồn cho tập dữ liệu, sau đó cho phép người dùng lựa chọn từ một số các công cụ để hiển thị, sắp xếp, đếm, phân nhóm, và ghép dữ liệu lấy từ tập dữ liệu trên. Trong các công cụ này, có thể sử dụng rwstats và rwcount để tạo ra một danh sách thống kê lưu lượng.

Trong khi nhiều người sử dụng SiLK để trực tiếp xem dữ liệu lưu lượng, rwstats là một trong những cách mạnh mẽ nhất để thực sự sử dụng dữ liệu phiên nhằm có được sự hiểu biết tốt hơn về môi trường của tổ chức, tiến hành phản ứng với sự cố, và tìm kiếm dữ liệu. Trong mọi môi trường SiLK được triển khai, rwstats luôn là nguồn dữ liệu thống kê thường xuyên nhất được sử dụng. Phần này sẽ bắt đầu bằng cách sử dụng rwstats để tạo ra một danh sách thống kê lưu lượng theo thiết bị.

Với SiLK, nên bắt đầu bằng cách tạo ra một lệnh rwfilter để có thể xác minh tập dữ liệu sử dụng để tạo ra số liệu thống kê. Nói chung, có thể dễ dàng tạo ra bộ lọc và gửi dữ liệu kết quả đến rwcut. Nó sẽ hiển thị kết quả của lệnh rwfilter để có thể chắc chắn tập dữ liệu đang sử dụng là hợp lệ. Phần lớn các ví dụ này sẽ sử dụng ví dụ cơ bản của các truy vấn rwfilter để bắt cứ ai cũng có thể làm theo.

Rwstats chỉ yêu cầu xác định ba thành phần: một tham số đầu vào, một tập các trường dùng để tạo ra số liệu thống kê, và các điều kiện dừng khi muốn giới hạn kết quả. Các tham số đầu vào có thể là một tên tệp tin được liệt kê trên dòng lệnh, hoặc trong trường hợp phổ biến hơn, là dữ liệu được đọc từ đầu vào tiêu chuẩn, từ kết quả của một lệnh rwfilter. Đầu vào này cần được lấy

trực tiếp từ rwfilter và không được phân tích bởi lệnh rwcutf. Tập các trường đã xác định đại diện cho một khóa được người dùng định nghĩa, từ đó các bản ghi luồng của SiLK được nhóm lại. Dữ liệu phù hợp với khóa đã được lưu trong các nhóm (bins) cho mỗi kết quả so sánh duy nhất. Sau đó tổng số của các nhóm này (có thể là tổng số byte, bản ghi, gói dữ liệu, hay số lượng bản ghi giao tiếp riêng biệt) được sử dụng để tạo ra một danh sách sắp xếp theo kích thước từ trên xuống dưới (mặc định), hoặc từ dưới lên trên, tùy thuộc vào lựa chọn của người sử dụng. Các điều kiện dừng được sử dụng để hạn chế các kết quả tạo ra và có thể được hạn chế bằng cách xác định theo tổng số (20 bin), theo một ngưỡng giá trị (các bin có số byte ít hơn 400), hoặc tỷ lệ phần trăm của tổng số (bin có chứa ít nhất 10% của tất cả các gói).

Lệnh thực hiện quá trình trên như sau:

```
rwfilter --start-date = 2013/08/26:14 --any-address = 102.123.0.0/16 --type = all --
pass = stdout | rwstats --top --count = 20 --fields = sip,dip --value = bytes
```

Trong ví dụ này, lệnh rwfilter tập hợp tất cả các bản ghi lưu lượng thu thập trong 1400 giờ ngày 8 tháng 8, và chỉ kiểm tra lưu lượng trong phạm vi IP 102.123.0.0/16. Dữ liệu đó được chuyển tới rwstats, để tạo ra một danh sách top 20 (--count = 20) kết hợp địa chỉ IP nguồn và đích (--fields = sip, dip) cho dữ liệu trong bộ lọc, sắp xếp theo byte (--value = bytes).

Một cách khác để đạt được cùng một kết quả là phải truyền các kết quả của lệnh rwfilter vào một tệp tin, và sau đó sử dụng rwstats để phân tích tệp tin đó. Hai lệnh sau sẽ thực hiện việc này, sử dụng một tệp tin có tên test.rwf:

```
rwfilter --start-date = 2013/08/26:14 --any-address = 102.123.0.0/16 --type = all --
pass = stdout > test.rwf
rwstats test.rwf --top --count = 20 --fields = sip,dip --value = bytes
```

Kết quả từ các lệnh này được thể hiện trong Hình 3.23.

INPUT: 14258003 Records for 1442095 Bins and 359838034159 Total Bytes			
OUTPUT: Top 20 Bins by Bytes			
sIP	dIP	Bytes	%Bytes
102.123.222.245	168.59.76.107	30038339439	8.347739
173.221.197.93	[102.123.155.234]	29118445814	8.092098
173.221.45.238	[102.123.168.43]	6281460707	1.723403
102.123.178.162	[119.104.73.89]	6196314721	1.721973
102.123.242.126	[79.86.35.244]	6001082411	1.667718
173.12.240.132	[102.123.168.43]	5633491224	1.565563
102.123.73.19	[173.184.168.79]	4815223824	1.338164
102.123.73.19	[173.184.168.42]	4791526426	1.331579
102.123.168.248	[102.123.112.7]	3817497327	1.060693
102.123.142.81	[102.123.155.160]	2335601387	0.649078
168.59.73.168	[102.123.25.93]	2329652425	0.647417
173.184.210.142	[102.123.73.31]	2119689249	0.589068
71.9.97.121	[102.123.175.66]	2102937605	0.584412
102.123.168.169	[168.59.121.15]	1950040136	0.541922
168.59.73.132	[102.123.25.53]	1903135121	0.528887
102.123.142.136	[184.201.124.129]	1890621864	0.525409
102.123.142.203	[102.123.155.160]	1845292902	0.512812
102.123.155.25	[119.104.73.49]	1831162761	0.508885
168.59.73.168	[102.123.25.53]	1795540049	0.498986
102.123.142.79	[184.201.124.129]	1756983568	0.488271

Hình 3.23 Nhóm danh sách lưu lượng liên lạc cao được tạo ra bởi Rwstats

Trong số liệu thể hiện trong Hình 3.23, có một số thiết bị vô cùng bận rộn trong mạng cục bộ là máy tính 102.123.222.245. Có thể tạo thêm số liệu thống kê để giúp thu hẹp giao tiếp của nó.

Bằng cách chạy lệnh tương tự, nhưng thay phạm vi CIDR cho địa chỉ IP của nhóm máy tính cao trong danh sách trong lệnh rwfilter, có thể thấy các máy tính nó đang giao tiếp đã tạo ra tất cả các lưu lượng truy cập này.

```
rwfilter --start-date = 2013/08/26:14 --any-address = 102.123.222.245 --type = all --
pass = stdout | rwstats --top --count = 5 --fields = sip,dip --value = bytes
```

Các số liệu thống kê được tạo ra bằng truy vấn này được thể hiện trong Hình 3.24.

INPUT: 128837 Records for 8511 Bins and 32059683554 Total Bytes					
OUTPUT: Top 5 Bins by Bytes					
sIP	dIP	Bytes	%Bytes	cumul_%	
102.123.222.245	168.59.76.107	3008339439	93.695059	93.695059	
168.59.76.107	102.123.222.245	418987922	1.306658	95.001789	
102.123.222.245	102.123.231.154	81795975	0.255137	95.256846	
102.123.222.245	102.123.168.62	60875215	0.189881	95.446727	
102.123.168.62	102.123.222.245	24694805	0.077028	95.523754	

Hình 3.24 Tập trung vào nhóm các đối tác liên lạc thường xuyên của một máy tính đơn lẻ

Điều này giúp xác định "ai" liên quan đến lưu lượng cao bất thường. Có thể xác định được "cái gì" bằng cách thay đổi tiêu chí tìm kiếm để có gắng xác định các dịch vụ phổ biến quan sát được từ việc giao tiếp giữa các thiết bị này.

```
rwfilter --start-date = 2013/08/26:14 --any-address = 102.123.222.245 --type = all --
pass = stdout | rwstats --top --count = 5 --fields = sip,sport,dip --value = bytes
```

Trong lệnh này, sử dụng cùng một tập dữ liệu, nhưng cho rwstats sử dụng trường cổng nguồn làm tiêu chí cho việc tạo thống kê. Hình 3.25 cho chúng ta thấy kết quả của truy vấn này.

INPUT: 128837 Records for 32092 Bins and 32059683554 Total Bytes					
OUTPUT: Top 5 Bins by Bytes					
sIP	sPort	dIP	Bytes	%Bytes	cumul_%
102.123.222.245	22	168.59.76.107	15257159577	47.589863	47.589863

Hình 3.25 Sử dụng thống kê để xác định lượng sử dụng dịch vụ

Có thể thấy thủ phạm liên quan tới một số kiểu kết nối SSH, nhưng trước khi kiểm tra thông qua người dùng hoặc một nguồn dữ liệu, hãy tạo thêm một số thống kê có thể giúp xác định "khi nào" là thời điểm của liên lạc này. Để làm được điều này, cần sử dụng rwcount để xác định khoảng thời gian giao tiếp diễn ra. Rwcount là một công cụ trong gói phân tích SiLK giúp tóm tắt bản ghi SiLK theo thời gian. Việc này được thực hiện bằng cách đếm các bản ghi trong các dòng đầu vào và nhóm các byte và tổng số gói vào bin theo thời gian. Theo mặc định, truyền một lệnh rwfilter trực tiếp đến rwcount sẽ cung cấp một bảng diễn tả khói lượng các bản ghi, byte, và các gói nhìn thấy trong mỗi khoảng thời gian 30 trong kết quả rwfilter. Sử dụng tùy chọn --bin-size

cho phép thay đổi điều đó bằng cách xác định một giá trị giây khác. Kết quả là sẽ sử dụng lệnh sau:

```
rwfilter --start-date = 2013/08/26:14 --any-address = 102.123.222.245 --sport = 22 --
type = all --pass = stdout | rwdump --bin-size = 600
```

Do đang cố gắng xác định thời điểm lưu lượng truy cập cổng 22 này xảy ra, cần chỉnh sửa rwfilter sử dụng tùy chọn --sport = 22, và thay thế rwdump với rwdump để đánh giá các đơn vị thời gian mà trong khoảng đó dữ liệu tồn tại. Sử dụng các tùy chọn --bin-size để kiểm tra lưu lượng với các bin 10 phút (600 giây). Kết quả của lệnh này được hiển thị trong Hình 3.26.

Date	Records	Bytes	Packets
2013/08/26T14:30:00	0.26	405758958.97	271885.01
2013/08/26T14:40:00	1.00	1553551544.63	1040980.97
2013/08/26T14:50:00	1.00	1553551544.63	1040980.97
2013/08/26T15:00:00	1.00	1516348807.05	1016170.31
2013/08/26T15:10:00	1.00	1411920652.84	946529.39
2013/08/26T15:20:00	1.00	1411920652.84	946529.39
2013/08/26T15:30:00	1.00	1596803663.96	1070176.01
2013/08/26T15:40:00	1.00	2120740976.27	1428577.88
2013/08/26T15:50:00	1.00	2120740976.27	1428577.88
2013/08/26T16:00:00	0.74	1565821801.15	1048865.34

Hình 3.26 Kết quả Rwdump chi tiết khi có liên lạc

Có thể thấy việc truyền dữ liệu tương đối nhất quán theo thời gian. Như vậy, đường hầm SSH có thể được sử dụng để chuyển một lượng lớn dữ liệu. Đây có thể là một nguy cơ như rò rỉ dữ liệu, hoặc một cái gì đó đơn giản như một người sử dụng công cụ SCP để chuyển một cái gì đó tới hệ thống khác với mục đích sao lưu. Xác định câu trả lời đúng cho câu hỏi này sẽ đòi hỏi việc phân tích các nguồn dữ liệu bổ sung, nhưng với các số liệu thống kê đã tạo ra ở đây sẽ cung cấp cho các chuyên gia phân tích một ý tưởng về vị trí cần xem xét tiếp theo.

### 3.4.2 Khám phá dịch vụ với SiLK

Rwstats cũng có thể được sử dụng để thực hiện các hoạt động phát hiện các tài sản được bảo vệ trong mạng nội bộ. Ví dụ sau sẽ xác định được một số máy chủ quan trọng thường xuyên giao tiếp với các thiết bị bên ngoài của mạng nội bộ. Quá trình này bắt đầu với việc tạo ra một rwfilter để thu thập tập dữ liệu để từ đó tạo ra số liệu thống kê. Trong một kịch bản lý tưởng, loại truy vấn này được chạy định kỳ và được kiểm tra liên tục. Điều này có thể trợ giúp trong việc bắt máy chủ giả mạo được mở ra tạm thời.

Ở đây, sẽ làm việc với một tệp tin được tạo bởi rwfilter để có thể đơn giản truyền nó qua rwstats. Để thực hiện việc này, có thể sử dụng một bộ lọc như thế để tạo ra một tập dữ liệu dựa trên tất cả các lưu lượng truy cập trong một khoảng thời gian cụ thể và truyền dữ liệu đó vào một tệp tin gọi là sample.rw.

```
rwfilter --start-date = 2013/08/28:00 --end-date = 2013/08/28:23 --type = all --
protocol = 0- --pass = sample.rw
```

Với một tập dữ liệu sẵn sàng cho phân tích cú pháp, bây giờ cần phải xác định những số liệu thống kê muốn tạo ra. Sau khi tạo ra số liệu thống kê, có thể viết ra các câu hỏi và "chuyển" nó

vào cú pháp rwstats để đạt được dữ liệu đang cần tìm kiếm. Ví dụ hãy tự hỏi: "Các thiết bị trong mạng nội bộ trao đổi cái gì nhiều nhất tại các cổng phổ biến, 1-1024?"

Câu hỏi này gợi ra nhiều vấn đề. Việc phân định về nhóm cao nhất trong danh sách có thể là yếu tố quyết định cho các kết quả thực sự cần. Nếu cần tra 20 thiết bị trong danh sách nhóm cao nhất với câu hỏi trên, có thể thực hiện lệnh rwstats như sau:

```
rwfilter sample.rw --type = out,outweb --sport = 1-1024 --pass = stdout | rwstats --  
fields = sip,sport --count = 20 --value = dip-distinct
```

Các kết quả của truy vấn này được thể hiện trong Hình 3.27.

sIP	sPort	dIP-Distinct	%dIP-Distinct	cumul_%
219.15.129.211	53	41896	?	?
184.226.35.112	25	28637	?	?
184.226.79.198	25	16328	?	?
184.226.79.199	53	6155	?	?
184.226.79.216	53	6134	?	?
219.15.128.211	53	4066	?	?
219.15.128.242	53	4062	?	?
219.15.165.211	25	1458	?	?
184.226.19.89	25	387	?	?
184.226.60.116	25	357	?	?
184.226.60.145	25	318	?	?
184.226.60.43	25	315	?	?
219.15.4.152	25	233	?	?
219.15.178.3	992	194	?	?
184.226.35.215	25	113	?	?
120.140.239.134	500	100	?	?
184.226.127.254	500	95	?	?
219.15.155.69	21	88	?	?
219.15.156.80	25	86	?	?
184.226.94.102	500	65	?	?

Hình 3.27 Nhóm các cổng máy chủ giao tiếp nhiều nhất

Kết quả của truy vấn này cho thấy 20 máy tính đầu (--count = 20) theo địa chỉ IP và cổng nguồn (--fields = sip, sport) được xác định bằng cách xem xét số thiết bị bên ngoài các máy tính liên lạc với (--value = dip-distinct). Tập dữ liệu sử dụng cho truy vấn này được giới hạn bằng cách chạy rwfilter trên tập dữ liệu sample.rw đã tạo ra, và chỉ truyền lưu lượng gửi ra ngoài từ các cổng 1-1024 đến rwstats (--type = out, outweb --sport = 1-1024).

Điều này cho chúng ta một số dữ liệu cần thiết, nhưng còn những thông tin khác về lưu lượng truy cập máy chủ mà không liên quan đến thông tin liên lạc từ các host bên ngoài? Nếu bộ thu thập luồng (một bộ cảm biến hoặc một bộ định tuyến) là vị trí thu thập luồng bên trong nội bộ, thì có thể thêm tùy chọn kiểu int2int.

Điều có thể thực hiện để nâng cao chất lượng của dữ liệu và làm tốt hơn cho dữ liệu thống kê sẽ tạo ra sau này, là hạn chế nó chỉ đến nguồn có địa chỉ IP trong phạm vi mạng mà có trách nhiệm bảo vệ. Điều này thường sẽ là các giá trị được định nghĩa trong tệp tin SiLK sensor.conf

như các khối (block) IP nội bộ. Cách tốt nhất để xử lý là tạo ra một tập hợp gồm những khối IP nội bộ. Rwtools SiLK sử dụng thiết lập các tệp tin để tham khảo nhóm các địa chỉ IP. Để tạo ra một tệp tin thiết lập, cách đơn giản là đặt tất cả các địa chỉ IP (bao gồm cả phạm vi CIDR) trong một tệp tin văn bản, và sau đó chuyển đổi nó vào một tệp tin bằng cách sử dụng tập lệnh rwsetbuild:

```
rwsetbuild local.txt local.set
```

Ở đây, rwsetbuild mất trong danh sách các khối IP quy định trong tệp tin local.txt, và đầu ra các tệp tin thiết lập tên local.set. Với các tệp tin thiết lập tạo ra, có thể sử dụng lệnh sau để có được các dữ liệu mong muốn:

```
rwfilter sample.rw --sipset = local.set --type = int2int,out,outweb --sport = 1-1024 --pass = stdout | rwstats --fields = sip,sport --count = 20 --value = dip-distinct
```

Chú ý ở đây là các tùy chọn --sipset được sử dụng với các lệnh rwfilter giới hạn dữ liệu một cách thích hợp để nguồn chỉ có địa chỉ IP thuộc phạm vi của mạng có trách nhiệm bảo vệ.

Với những thay đổi tối thiểu các phương pháp đã sử dụng, có thể thu hẹp các lệnh này để phù hợp với môi trường của riêng tổ chức với độ chính xác cao. Ví dụ, trong khi chỉ kiểm tra 20 so sánh với mỗi truy vấn, có thể xác định bất kỳ thiết bị tương tự nào như một máy chủ cần được xem xét trong truy vấn nếu nó liên lạc với ít nhất 10 thiết bị lạ. Để có được một danh sách các thiết bị phù hợp với tiêu chí đó, chỉ cần thay đổi --count = 20 đến --threshold = 10. Có thể thao tác để cho kết quả tốt hơn bằng cách tập trung vào dài cổng hoặc tập trung vào các tập tệp tin mới. Điều quan trọng cần lưu ý ở đây là đang tìm kiếm tập trung vào các dịch vụ, và cụ thể là --fields = sip,sport, có nghĩa là đang hiển thị nhóm danh sách kết hợp các địa chỉ nguồn và cổng nguồn. Nếu ý tưởng ở đây là xác định các máy chủ giao tiếp bằng tổng số lượng địa chỉ IP đích riêng biệt khác nhau, thì điều quan trọng là phải loại bỏ các trường sport trong rwstats để tổng hợp tổng số các kết nối với mỗi thiết bị cụ thể, như sau:

```
rwfilter sample.rw --sipset = local.set --type = all --sport = 1-1024 --pass = stdout| rwstats --fields = sip --count = 20 --value = dip-distinct
```

Lấy kết quả của truy vấn này và thực hiện thêm lệnh rwstats để đi sâu vào các địa chỉ cụ thể (như trong ví dụ trước) sẽ mang lại nhiều thông tin liên quan đến những dịch vụ đang chạy trên các thiết bị xuất hiện trong danh sách đã tạo ra. Ví dụ, nếu muốn đi sâu vào các dịch vụ chạy trên 192.168.1.21, có thể xác định một "dịch vụ" của cổng nguồn cá nhân có ít nhất 10 lưu lượng đi ra ngoài lề. Có thể thu hẹp rwfilter bao gồm địa chỉ này và thay đổi lệnh rwstats để xem xét thông số nguồn:

```
rwfilter sample.rw --saddress = 192.168.1.21 --type = all --pass = stdout| rwstats --fields = sport --threshold = 10 --value = dip-distinct
```

Đầu ra của lệnh này được hiển thị trong Hình 3.28.

sPort	dIP-DistIn	%dIP-DistIn	cumul_%
443	35	?	?
9292	15	?	?
9200	15	?	?
22	14	?	?
3389	10	?	?

**Hình 3.28** *Đi sâu vào dịch vụ chạy trên một thiết bị cụ thể*

Một số ví dụ để có được chi tiết về tổ chức dịch vụ mạng của tổ chức (tham khảo trong bài viết "Network Profiling using Flow" của hai tác giả Whisnant và Faber):

### Máy chủ Web

```
rwfilter sample.rw --type = outweb --sport = 80,443,8080 --protocol = 6 --packets = 4- --ack-flag = 1 --pass = stdout|rwstats --fields = sip --percentage = 1 --bytes --no-titles|cut -f 1 -d "| "|rwsetbuild > web_servers.set ; echo Potential Web Servers:;rwfilter sample.rw --type = outweb --sport = 80,443,8080 --protocol = 6 --packets = 4- --ack-flag = 1 --sipset = web_servers.set --pass = stdout|rwuniq --fields = sip,sport --bytes --sort-output
```

### Máy chủ thư điện tử

```
echo Potential SMTP servers ;rwfilter sample.rw --type = out --sport = 25,465,110,995,143,993 --protocol = 6 --packets = 4- --ack-flag = 1 --pass = stdout|rwset --sip-file = smtpservers.set ;rwfilter sample.rw --type = out --sport = 25,465,110,995,143,993 --sipset = smtpservers.set --protocol = 6 --packets = 4- --ack-flag = 1 --pass = stdout|rwuniq --fields = sip --bytes --sort-output
```

### Máy chủ DNS

```
echo DNS Servers: ;rwfilter sample.rw --type = out --sport = 53 --protocol = 17 --pass = stdout|rwstats --fields = sip --percentage = 1 --packets --no-titles|cut -f 1 -d "| "|rwsetbuild > dns_servers.set ;rwsetcat dns_servers.set
```

### Máy chủ VPN

```
echo Potential VPNs: ;rwfilter sample.rw --type = out --protocol = 47,50,51 --pass = stdout|rwuniq --fields = sip --no-titles|cut -f 1 -d "| "|rwsetbuild > vpn.set ;rwfilter sample.rw --type = out --sipset = vpn.set --pass = stdout|rwuniq --fields = sip,protocol --bytes --sort-output
```

### Máy chủ FTP

```
echo -e "FTP Servers"; rwfilter sample.rw --type = out --protocol = 6 --packets = 4- --ack-flag = 1 --sport = 21 --pass = stdout|rwstats --fields = sip --percentage = 1 --bytes --no-titles|cut -f 1 -d "| "|rwsetbuild > ftpservers.set ;rwsetcat ftpservers.set ; echo FTP Servers making active connections: ;rwfilter sample.rw --type = out --sipset = ftpservers.set --sport = 20 --flags-initial = S/SAFR --pass = stdout|rwuniq --fields = sip
```

## Máy chủ SSH

```
echo -e "SSH Servers"; rwfilter sample.rw --type = out --protocol = 6 --packets = 4- --ack-flag = 1 --sport = 22 --pass = stdout|rwstats --fields = sip --percentage = 1 --bytes --no-titles|cut -f 1 -d ";"|rwsetbuild > ssh_servers.set ;rwsetcat ssh_servers.set
```

## Máy chủ TELNET

```
echo -e "Telnet Servers"; rwfilter sample.rw --type = out --protocol = 6 --packets = 4- --ack-flag = 1 --sport = 23 --pass = stdout|rwstats --fields = sip --percentage = 1 --bytes --no-titles|cut -f 1 -d ";"|rwsetbuild > telnet_servers.set ;rwsetcat telnet_servers.set
```

## Máy chủ Leftover

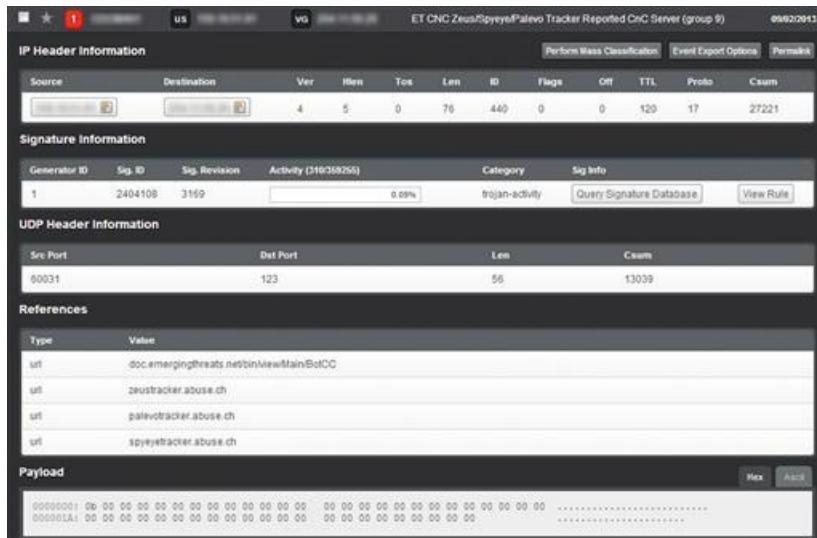
```
echo Leftover Servers: ;rwfilter sample.rw --type = out --sport = 1-19,24,26-52,54-499,501-1023 --pass = stdout|rwstats --fields = sport --percentage = 1
```

Trong kịch bản phát hiện xâm nhập, các lệnh này sẽ được chạy thường xuyên. Kết quả của mỗi lần chạy nên được so sánh với lần chạy trước, và khi một thiết bị mới chạy như một máy chủ bật lên (pops up), nó cần phải được chú ý xem xét.

### 3.4.3 Tìm hiểu thêm về phát hiện xâm nhập dựa trên thông kê

Đối với hầu hết các tổ chức, dữ liệu cảnh báo và phân tích thời gian thực cung cấp phần lớn các sự cố phải báo cáo trên mạng. Khi một cảnh báo mới được tạo ra, có thể hữu ích khi tạo ra các truy vấn thông kê sử dụng dữ liệu phiên để giúp phát hiện sự tồn tại của các IOC tương tự trên các máy khác.

Ví dụ, xem xét các cảnh báo được hiển thị trong Hình 3.29.



Hình 3.29 Một cảnh báo về Zeus tạo ra bởi Snort

Cảnh báo này được tạo ra do chứng cứ về liên lạc với một thiết bị có liên kết với lệnh máy chủ C&C của botnet Zeus. Nhìn sơ qua, lưu lượng này có vẻ như lưu lượng NTP do các kết nối xuất hiện giống như lưu lượng UDP, qua cổng 123.

Nếu không có quyền truy cập vào tài của gói tin, sự kiện này có thể bị che đậy bởi một số nhà phân tích vì không có các dấu hiệu về sự nhiễm mã độc nào khác ngay khi đó. Có khả năng lưu lượng này chỉ đơn thuần che dấu giao tiếp của mình bằng cách sử dụng cổng NTP chung. Tuy nhiên, nếu không có thêm chi tiết nào khác thì cũng không thể chắc chắn nhận định này. Muốn biết rõ hơn, cần phải xem thêm các liên lạc khác của máy tính. Để làm điều này, chỉ cần tìm các chi tiết duy nhất về cảnh báo và xác định xem máy tính đang liên lạc với "các máy chủ NTP" khác nữa mà có thể có dấu hiệu đáng ngờ. Cần thêm trường mã quốc gia vào truy vấn này, vì ở đây chỉ muốn các thiết bị trong mạng giao tiếp với máy chủ NTP tại Mỹ. Lệnh được thực hiện như sau:

```
rwfilter --start-date = 2013/09/02 --end-date = 2013/09/02 --any-address = 192.168.1.17 --
aport = 123 --proto = 17 --type = all --pass = stdout | rwstats --top --fields = dip,dcc,dport
--count = 20
```

Lệnh này sử dụng rwstats để hiển thị các thiết bị mà 192.168.1.17 giao tiếp trên cổng 123. Các kết quả được trình bày trong Hình 3.30. Trong hình này, một số địa chỉ IP đã được ẩn danh.

INPUT: 2042 Records for 44 Bins and 2042 Total Records			
OUTPUT: Top 20 Bins by Records			
dIP	[dcc]	dPort	Records
192.5.xxxxxx  us  123  128			6.268364
192.36.xxxxxx  sel  123  86			4.211557
192.36.xxxxxx  sel  123  85			4.162586
192.36.xxxxxx  sel  123  84			4.113614
158.254.xxxxxx  pl  123  84			22.869736
129.242xxxxx  no  123  83			4.064643
192.36.xxxxxx  sel  123  81			3.966699
62.119xxxxx  sel  123  81			3.966699
62.119xxxxx  sel  123  80			34.867777
192.36.xxxxxx  sel  123  77			3.917726
192.36.xxxxxx  sel  123  75			3.779813
192.36.1xxxxx  sel  123  75			42.556317
203.117.xxxxxx  sgl  123  75			3.672878
192.36.xxxxxx  sel  123  73			46.229187
193xxxxxx  sil  123  72			3.672878
284.11xxxxx  vgl  123  67			49.902057
192.36.1xxxxx  sel  123  67			53.476983
128.9.xxxxxx  us  123  64			3.574927
192.16xxxxxx  us  60059  60			60.284835
193.62xxxxx  gbl  123  59			63.565132
192.16xxxxxx  us  60060  58			66.699314
			69.637610
			72.526934
			75.367287

Hình 3.30: Máy tính được bảo vệ giao tiếp với nhiều máy tính khác trên cổng 123

Như thấy trên hình 3.30, máy tính nội bộ trong câu hỏi dường như đang giao tiếp với nhiều máy tính bên ngoài qua cổng 123. Các bản ghi gắn với mỗi máy tính và số lượng các liên lạc cho thấy rằng có một cái gì đó nguy hại đang xảy ra ở đây, hoặc ít nhất là đây không thực sự lưu lượng NTP. Trong một tình huống thông thường, một máy tính sẽ chỉ đồng bộ hóa các thiết lập NTP với một hoặc một vài máy tính dễ nhận biết. Sự nguy hại của lưu lượng truy cập này có thể được xác nhận bởi sự tồn tại của các kết quả ở nước ngoài (Non-US), đây không phải là điển hình của việc đồng bộ NTP với một máy chủ / mạng có trụ sở tại Mỹ.

Tại thời điểm này, có một sự cố có thể leo thang. Như trước đây, cần phải đánh giá những gì cần trước khi xem xét dữ liệu trước. Trong sự kiện này, chúng ta đã tìm kiếm tất cả dữ liệu phiên, nơi 192.168.1.17 là địa chỉ nguồn và là nơi giao tiếp đã xảy ra, trên cổng UDP 123. Số lượng áp

đảo của các lưu lượng UDP / 123 truy cập vào rất nhiều máy chủ bên ngoài đã dẫn đến kết luận rằng có hành vi nguy hại xảy ra. Có thể tạo ra một bộ lọc phù hợp với những đặc điểm này cho bất kỳ địa chỉ cụ bộ nào. Bộ lọc sẽ tương tự như thế này:

```
rwfilter --start-date = 2013/09/02 --end-date = 2013/09/02 --not-dipset = local.set --
dport = 123 --proto = 17 --type = all --pass = stdout | rwstats --top --fields = sip --
count = 20 --value = dip-distinct
```

Lệnh trên cho biết chỉ kiểm tra dữ liệu: từ 02/09/2013, không dành cho mạng nội bộ, và những gì đến cổng 123 sử dụng giao thức UDP. Những thông tin này sẽ được gửi đến rwstats, tạo ra số liệu thống kê cho nhóm đứng đầu gồm 20 địa chỉ IP khác nhau đáp ứng các tiêu chí (Hình 3.31).

INPUT: 19279 Records for 95 Bins			
OUTPUT: Top 20 Bins by dIP-Distinct			
sIP dIP-Distinct	%dIP-Distinct	cumul_%	
192.16.xxxxxx  596  ?  ?			
192.16xxxxxx  508  ?  ?			
192.16.1xxxxxx  471  ?  ?			
192.16.xxxxxx  138  ?  ?			
192.26xxxxxx  93  ?  ?			
10.20.xxxxxx  46  ?  ?			
192.16xxxxxx  22  ?  ?			
10.49xxxxxx  21  ?  ?			
10.49xxxxxx  21  ?  ?			
205.204xxxxxx  12  ?  ?			
192.24.1xxxxxx  18  ?  ?			
192.25xxxxxx  7  ?  ?			
10.242.xxxxxx  5  ?  ?			
10.240.xxxxxx  5  ?  ?			
192.16xxxxxx  3  ?  ?			
192.24.xxxxxx  3  ?  ?			
192.24.xxxxxx  3  ?  ?			
192.24.1xxxxxx  3  ?  ?			
10.240.xxxxxx  3  ?  ?			
10.43.1xxxxxx  3  ?  ?			

**Hình 3.31 Hiển thị nhiều thiết bị có các mẫu liên lạc tương tự**

Có thể thu hẹp bộ lọc này một chút bằng cách cho thêm điều kiện là: chỉ có lưu lượng UDP/123 đi ra các máy tính ngoài nước Mỹ (là một tiêu chí chỉ ra các liên lạc ban đầu đáng ngờ). Truy vấn này được xây dựng trên phần trước đó, rồi truyền đầu ra của rwfilter đầu tiên đến rwfilter thứ hai thực hiện loại bất kỳ bản ghi có chứa mã điểm đến là "us", đảm bảo là chỉ xem xét các dữ liệu đi ra nước ngoài.

```
rwfilter --start-date = 2013/09/02 --end-date = 2013/09/02 --not-dipset = local.set --
dport = 123 --proto = 17 --type = all --pass = stdout | rwfilter --input-pipe = stdin --
dcc = us --fail = stdout | rwstats --top --fields = sip --count = 20 --value = dip-distinct
```

Kiểm tra kỹ hơn những kết quả này có thể dẫn đến việc phát hiện thành công các logic độc hại trên hệ thống khác có một hành vi tương tự như cảnh báo IDS ban đầu. Trong khi một cảnh báo IDS có thể bắt được một số trường hợp độc hại, nó sẽ bắt tất cả chúng, khi đó phân tích thống kê có thể có ích. Ví dụ thể hiện ở đây được lấy từ một cuộc điều tra thực tế, khi đó các phân tích cho thấy 9 máy chủ bị nhiễm khác mà cảnh báo IDS ban đầu đã không chỉ ra được.

### 3.4.4 Một số công cụ hiển thị thống kê

Phần này trình bày về một số công cụ hiển thị thống kê.

#### 3.5.4.1 Hiển thị thống kê với Gnuplot

Tính năng hiển thị thống kê trực quan cung cấp cái nhìn sâu sắc và hữu ích về số liệu. Hiển thị thống kê có thể dùng để phát hiện bất thường và đưa ra các cảnh báo cho chuyên gia phân tích khi một thiết bị tạo ra hoặc nhận được lưu lượng lớn hơn đáng kể so với lưu lượng trung bình thường. Điều này có thể hữu ích cho việc phát hiện dữ liệu rò rỉ ra bên ngoài, một máy chủ nội bộ được sử dụng để phục vụ cho các phần mềm độc hại trên Internet, hoặc một tấn công từ chối dịch vụ nội mạng (inbound). Đồ thị thống kê cũng có thể giúp các chuyên gia phân tích thu hẹp các truy vấn dữ liệu của họ trong một khoảng thời gian dễ quản lý hơn, và cuối cùng là giúp đầy mạnh quá trình phân tích.

Một trong những công cụ hữu ích hơn để tổng kết dữ liệu qua các khoảng thời gian cụ thể và tạo ra số liệu thống kê liên quan là rwcount. Công cụ này có thể được sử dụng để xem có bao nhiêu dữ liệu tồn tại trong các chuỗi giao tiếp. Ví dụ đơn giản nhất là xem có bao nhiêu dữ liệu đi qua một đoạn mạng được giám sát trong một ngày nhất định. Với hầu như tất cả các truy vấn SiLK, điều này sẽ bắt đầu với một lệnh rwfilter để tập trung vào chỉ khoảng thời gian đang quan tâm. Trong trường hợp này, sẽ đẩy dữ liệu vào rwcount để chia các dữ liệu vào các khoảng thời gian quy định trong vài giây. Ví dụ, để kiểm tra tổng số bản ghi, byte, và gói mỗi phút (--bin-size = 60) đi qua giao diện mạng trong một giờ nhất định, có thể sử dụng lệnh sau:

```
rwfilter --start-date = 2013/09/02: 14 --proto = 0- --pass = stdout --type = all | rwcount --bin-size = 60
```

Nhìn lại các ví dụ trong phần trước với lưu lượng NTP đáng ngờ. Nếu đào sâu hơn vào kết quả thể hiện trong Hình 3.30 bằng cách sử dụng rwcount như trên, có thể thấy rằng nhiều địa chỉ IP bên ngoài trong phạm vi IP 204.2.134.0/24 cũng yêu cầu các kết nối NTP client, từ đó có thể chỉ ra được những thiết bị gây rối được cấu hình để sử dụng các máy chủ NTP không nằm trong mạng cục bộ. Nếu tìm hiểu sâu hơn nữa, kiểm tra lưu lượng trong ngày, chỉ thấy một lượng dữ liệu trong mỗi phút (Hình 3.32); ở đây không đưa ra nhiều hỗ trợ trong việc giải thích lưu lượng.

Date	Records	Bytes	Packets
2013/09/02T00:20:00	2848.12	3881203.67	28537.12
2013/09/02T00:21:00	3565.92	5550209.88	41936.87
2013/09/02T00:22:00	3371.62	5296400.96	39838.18
2013/09/02T00:23:00	3014.78	4587246.46	34867.81
2013/09/02T00:24:00	2963.83	4713578.12	36133.11
2013/09/02T00:25:00	398.33	663388.48	5114.80
2013/09/02T00:26:00	1.41	1714.79	6.62
2013/09/02T00:27:00	0.96	847.26	3.95
2013/09/02T00:28:00	0.61	152.66	1.82
2013/09/02T00:29:00	0.51	128.79	1.53

Hình 3.32 Rwcount hiển thị dữ liệu trải đều trên các khoảng thời gian

Để thực sự hiển thị dữ liệu này trên một dải rộng hơn, có thể vẽ những đồ thị lớn hơn. Nhưng do SiLK không có khả năng làm điều này, nên sẽ cần xử lý các kết quả của truy vấn SiLK và đưa tới gnuplot để vẽ đồ thị.

Để làm cho các dữ liệu hiển thị ở trên hữu ích hơn, mục tiêu của chúng ta là xây dựng một đồ thị đại diện cho khối lượng các byte cho mỗi giờ cho phiên dữ liệu chứa bất kỳ địa chỉ từ dãy IP 204.2.134.0/24. Việc này bắt đầu bằng cách sử dụng các lệnh rwcount tương tự như trên, nhưng với kích thước của bin 3600 để mang lại "một giờ" kết quả. Các đầu ra của lệnh rwcount được gửi thông qua một số dòng lệnh sửa đổi để tạo ra một tệp tin CSV chỉ chứa các dấu thời gian và giá trị byte cho mỗi dấu thời gian. Lệnh được viết như sau:

```
rwfilter --start-date = 2013/09/02 --any-address = 204.2.134.0/24 --proto = 0- --
pass = stdout --type = all | rwcount --bin-size = 3600 --delimited =, --no-titles| cut -d "," -
f1,3 > hourly.csv
```

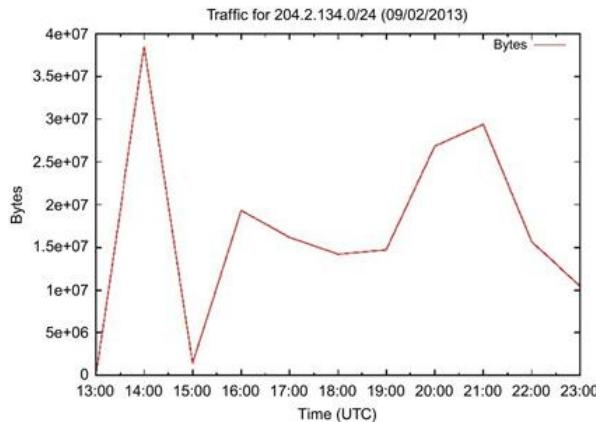
Kết quả dữ liệu như sau:

```
2013/09 / 02T13: 00: 00,146847.07
2013/09 / 02T14: 00: 00,38546884.51
2013/09 / 02T15: 00: 00,1420679.53
...
...
```

Tiếp theo, cần sử dụng Gnuplot để vẽ đồ thị các số liệu thống kê. Điều này được thực hiện bằng cách tạo ra một script Gnuplot.

```
#!/usr/bin/gnuplot
set terminal postscript enhanced color solid
set output "hourly.ps"
set title "Traffic for 204.2.134.0/24 (09/02/2013)"
set xlabel "Time (UTC)"
set ylabel "Bytes"
set datafile separator ","
set timefmt '%Y/%m/%dT%H:%M:%S'
set xdata time
plot 'hourly.csv' using 1:2 with lines title "Bytes"
```

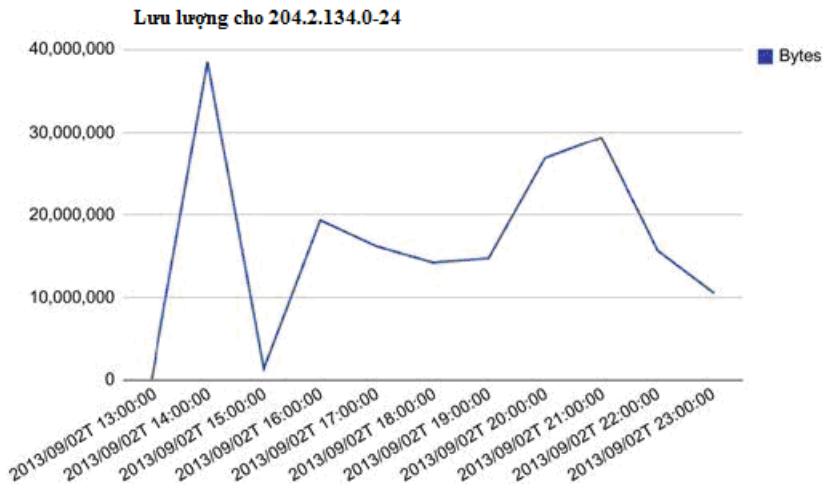
Cuối cùng, có một đồ thị thông lượng trong Gnuplot, thể hiện trong Hình 3.33.



Hình 3.33 Một đồ thị thông lượng Gnuplot

### 3.5.4.2 Hiển thị thống kê với Google Chart

Một cách khác để hiển thị dữ liệu thông lượng nhiều hơn nữa là tận dụng Google Chart API của Google (<https://developers.google.com/chart/>). Hầu hết các bảng biểu được tạo ra với các Google Chart API là tương thích với các trình duyệt và 100% miễn phí. Thêm vào đó, cú pháp của nó là tương đối đơn giản. Không mô tả cách làm chi tiết ở đây (tham khảo thêm trong tài liệu [1]) nhưng nếu sử dụng các API này, sẽ được một hình mô tả thông lượng theo thời gian như Hình 3.34:

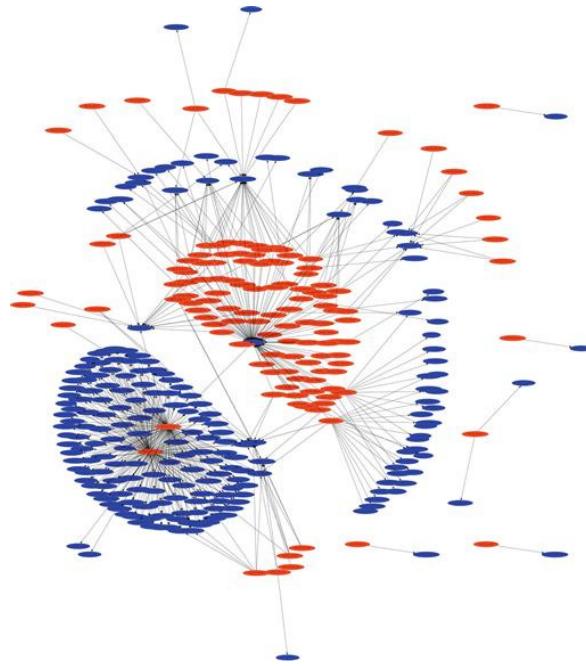


Hình 3.34 Một biểu đồ thông lượng sử dụng Google Chart API

### 3.5.4.3 Hiển thị thống kê với Afterglow

Afterglow là một công cụ Perl cho phép tạo ra một đồ thị các liên kết để có thể thấy được mô tả toàn cảnh về các thành phần liên quan đến nhau. Afterglow nhận tệp tin CSV có 2 hoặc 3 cột làm đầu vào và tạo ra tệp tin ngôn ngữ đồ thị có thuộc tính (theo yêu cầu của các thư viện graphviz) hoặc một tệp tin GDF có thể được phân tích bằng Gephi. Cần lưu ý là Afterglow lấy dữ liệu đầu vào và sinh dữ liệu có thể được sử dụng để tạo ra các đồ thị liên kết. Có rất nhiều ví dụ về cách sử dụng Afterglow để tìm các mối quan hệ trong một số bộ dữ liệu trên Internet; Pcap và Sendmail là những ví dụ hiển thị trên trang web chính của Afterglow.

Ở đây không đi sâu vào chi tiết về cách xây dựng đồ thị mà chỉ mô tả về ví dụ kết quả có được như trong Hình 3.35 với tệp tin test.gif. Chi tiết về script và cách cấu hình có thể tham khảo trong [1].



**Hình 3.35** Một đồ thị liên kết tạo từ dữ liệu NetFlow

Sức mạnh và sự linh hoạt mà Afterglow cung cấp cho phép tạo ra một số đồ thị liên kết rất hữu ích trong một loạt các tình huống, khi mà cần xem xét mối quan hệ giữa các thực thể.

## CHƯƠNG 4

# PHÂN TÍCH DỮ LIỆU

Chương này trình bày các vấn đề liên quan đến bước cuối cùng trong chu trình giám sát an toàn mạng là phân tích dữ liệu, bao gồm một số nội dung liên quan như sau: kỹ thuật phân tích gói tin với các công cụ như Tcpdump và Wireshark, chu trình thu thập tri thức về nguy cơ bảo mật cho NSM và quá trình tạo tri thức về tài nguyên cần bảo vệ cũng như tri thức về nguy cơ bảo mật. Phần cuối chương trình bày về quy trình phân tích dữ liệu.

### 4.1. PHÂN TÍCH GÓI TIN

#### 4.1.1 Xâm nhập vào gói tin

Gói tin là một minh họa của giao thức trong thực tế vì chúng được tạo ra phù hợp với tiêu chuẩn mô tả trong giao thức. Gói tin là một đơn vị dữ liệu được định dạng và truyền qua mạng từ thiết bị này tới thiết bị khác. Các gói tin là những đơn vị cơ bản nhất để tạo ra kết nối giữa các máy tính và do đó chúng cũng chính là bản chất của giám sát an toàn mạng.

Để tạo ra được gói tin cần kết hợp các dữ liệu từ nhiều giao thức. Ví dụ: một yêu cầu HTTP GET thông thường trong thực tế cần dùng ít nhất bốn giao thức để đảm bảo yêu cầu được truyền từ trình duyệt tới máy chủ web (HTTP, TCP, IP và Ethernet). Khi nhìn vào các gói tin trước đó sử dụng phần mềm Wireshark, có thể thấy các gói tin được hiển thị trong định dạng giống như mô tả trong Hình 4.1.



Hình 4.1 Một gói tin yêu cầu HTTP GET đơn giản hiển thị trong Wireshark

Wireshark là một công cụ tốt để giúp người dùng tương tác với gói tin và phân tích chúng. Nhưng nếu muốn hiểu được các gói tin ở mức cơ bản nhất, cần làm việc với một công cụ nền tảng

là tcpdump (hay Windump trong Windows). Khác với Wireshark, tcpdump không cung cấp giao diện đồ họa người dùng và các tiện ích kèm theo khi phân tích gói tin mà nó phải dựa vào người dùng để thực hiện phân tích cho từng gói dữ liệu riêng lẻ. Để làm việc được với tcpdump, người dùng phải nghĩ nhiều hơn về các gói tin đang phân tích, hiểu được và áp dụng được các tri thức của mình về gói tin. Khi đó, việc phân tích sẽ mang lại kết quả tốt hơn nhiều so với khi sử dụng bất kỳ công cụ phân tích tốt nhất nào.

Với tcpdump, có thể xem gói tin yêu cầu HTTP GET ở hình trên dưới dạng hệ cơ số 16 bằng lệnh sau:

```
tcpdump -nnxr ansm-13-httppget.pcapng
```

```
maverick:~ chris$ tcpdump -nnxr ansm-13-httppget.pcapng
reading from file ansm-13-httppget.pcapng, link-type Ethernet (Ethernet)
10:21:55.026674 08:c9:d0:ba:63:fb > c8:c1:c8:17:8c:e8, ethertype IPv4 (0x0800), length 199: 172.16.16.128.68804 >
67.205.2.30.80: Flags [P.], seq 438817104:438817249, ack 2543215998, win 16384, length 145
0x0000: c8c1 c817 8ce8 28e9 d8ba 63fb 0800 4508
0x0010: 00b9 3d2a 4008 4086 fa99 ac10 1000 43cd
0x0020: 021e edb4 0050 19ad bf58 9796 6576 5018
0x0030: 4009 cd7c 0000 4745 5428 2128 4854 5458
0x0040: 2f31 2e31 0d0a 5573 6572 2d11 6765 6e74
0x0050: 3d28 6375 726c 2137 2e32 342e 3829 2878
0x0060: 3836 5f36 342d 6170 706c 652d 6461 7277
0x0070: 696e 3132 2e30 2928 6c69 6263 7572 6c2f
0x0080: 372e 3234 2e38 284f 7065 6e53 534c 2f38
0x0090: 2e39 2e38 7620 706c 6962 2f31 2e32 2e35
0x00a0: 0d0a 48ef 7374 3a20 6178 706c 6965 646e
0x00b0: 736d 2e63 6f6d 0d0a 4163 6365 7874 3a20
0x00c0: 2a2f 2a0d 808d 0a
```

**Hình 4.2 Một gói tin yêu cầu HTTP GET đơn giản hiển thị trong tcpdump**

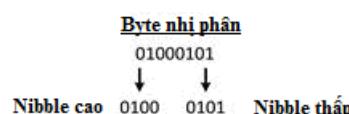
Nếu chỉ nhìn vào dữ liệu dạng cơ số 16 (hex) trong Hình 4.2, có thể thấy khó hiểu. Tuy nhiên, nếu áp dụng các giao thức để phân tích thì việc này sẽ không khó. Để bổ sung thông tin, phần sau sẽ mô tả một số khái niệm toán học cần dùng để hỗ trợ cho việc phân tích gói tin.

### 4.1.2 Một số khái niệm toán học liên quan

#### 4.1.2.1 Hiểu các byte diễn tả bằng hệ cơ số 16

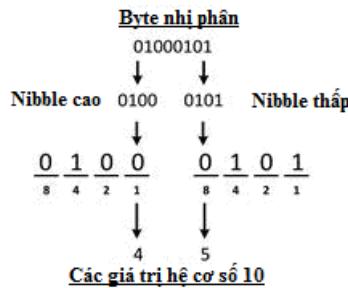
Khi xem xét các gói tin ở mức thấp với tcpdump, sẽ phải thường xuyên làm việc với dữ liệu gói hiển thị ở dạng thập lục phân (hệ cơ số 16, hex). Định dạng thập lục phân có được từ các diễn tả nhị phân của một byte. Một byte được tạo thành từ 8 bit, mỗi bit có thể là 0 hoặc 1. Ví dụ có một byte như sau: 01000101.

Để dễ đọc hơn, chúng ta chuyển nó sang dạng cơ số 16. Đầu tiên, tách byte thành 2 nửa, gọi là các nibble (Hình 4.3). Bốn bit đầu được gọi là nibble cao vì nó đại diện cho phần giá trị lớn trong byte. Bốn bit sau gọi là nibble thấp vì nó đại diện cho phần giá trị nhỏ hơn của byte.



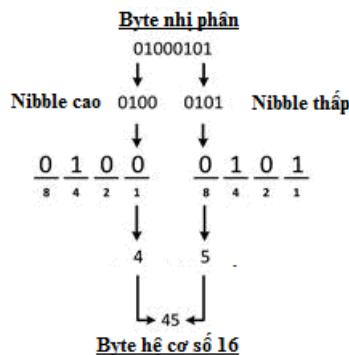
**Hình 4.3 Một byte chia thành các nibble**

Mỗi nibble của byte được chuyển thành một ký tự hệ cơ số 16 để tạo thành một byte 2 ký tự. Để tính toán các giá trị hệ cơ số 16 của một byte, trước tiên có thể tính các giá trị thập phân của mỗi nibble như trong Hình 4.4.



**Hình 4.4 Tính giá trị thập phân của mỗi nibble**

Mỗi ký tự hệ cơ số 16 có thể nhận giá trị 0-F với 0-9 tương đương 0-9 trong số thập phân và A-F bằng 10-15 trong số thập phân. Như vậy, 4 và 5 trong số thập phân tương đương 4 và 5 trong hệ cơ số 16, nghĩa là 45 là biểu diễn hệ cơ số 16 chính xác của byte 01000101. Toàn bộ quá trình được minh họa trong Hình 4.5.



**Hình 4.5 Chuyển đổi một byte dạng nhị phân sang hệ cơ số 16**

#### 4.1.2.2 Đếm số byte

Khi xem xét các gói tin ở hệ cơ số 16, cần dành nhiều thời gian cho việc đếm số byte. Có một số chú ý khi đếm byte trong gói tin.

Thứ nhất, đếm byte bắt đầu từ 0 chứ không phải 1. Điều này là do đang cần đếm từ một vị trí tương đối với offset. Đây là vị trí tương đối tới byte 0 trong tiêu đề (header) của giao thức hiện tại, chứ không phải là byte 0 bắt đầu gói tin. Xem ví dụ trong hình sau.

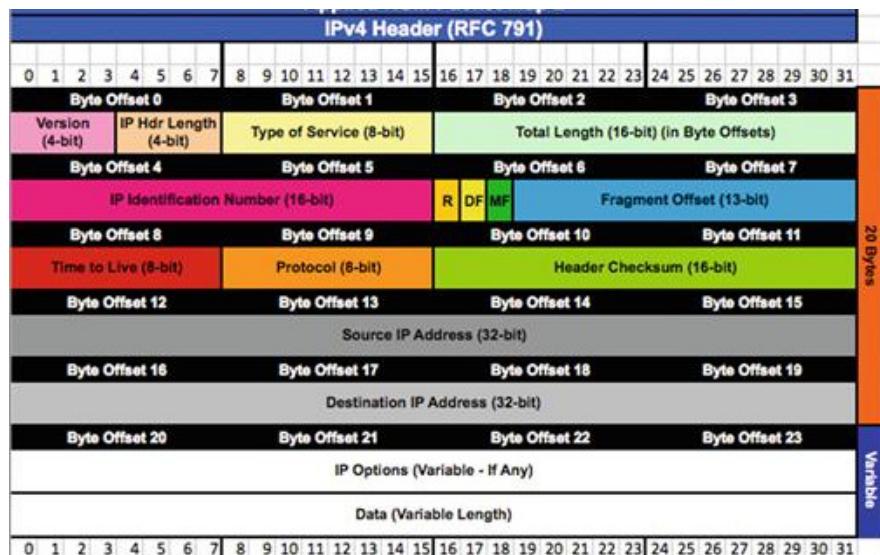
```

45 00 00 40 fd 0d 40 00 40 06 3b 2f ac 10 10 80
43 cd 02 1e ed 84 00 50 19 ad bf 4f 00 00 00 00
b0 02 ff ff 45 1f 00 00 02 04 05 b4 01 03 03 04
01 01 08 0a 31 4e f7 43 00 00 00 00 04 02 00 00

```

#### **Hình 4.6 Một gói tin IP dưới dạng hệ cơ số 16**

Hình 4.6 biểu diễn một gói tin IP cơ bản dưới dạng hệ cơ số 16 (đã được thêm các khoảng trống cho dễ đọc). Để hiểu được gói tin, cần đánh giá các giá trị trong các trường của gói tin. Việc này được thực hiện bằng cách so sánh với “bản đồ” (định dạng) các trường trong giao thức có trong gói tin. Trường hợp này chúng ta đã biết là gói tin IP nên cần xem xét một số giá trị trong phần tiêu đề của gói tin IP (như trong Hình 4.7).



#### **Hình 4.7 Định dạng tiêu đề gói tin IP**

Một thông tin hữu dụng giúp chúng ta có thể phân tích gói tin sâu hơn là giao thức được đóng gói trong IP. Giá trị này nằm trong phần đầu của IP header. Nhìn vào định dạng giao thức IP, thấy rằng giá trị này nằm tại byte số 9. Nếu đếm byte trong gói tin bắt đầu từ 1, sẽ xác định giá trị của trường giao thức đóng gói trong IP là 40. Nhưng như vậy là sai vì phải coi số này là tính từ offset, bắt đầu từ 0. Số đúng có được sẽ là 06 khi chúng ta đếm từ 0. Giá trị giao thức này là chỉ TCP.

```

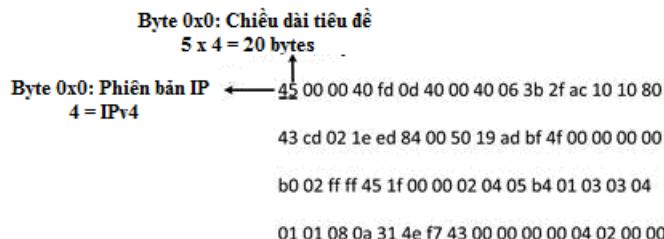
Byte 0x9: Giao thức
06=TCP
↑
45 00 00 40 fd 0d 40 00 40 06 3b 2f ac 10 10 80
43 cd 02 1e ed 84 00 50 19 ad bf 4f 00 00 00 00
b0 02 ff ff 45 1f 00 00 02 04 05 b4 01 03 03 04
01 01 08 0a 31 4e f7 43 00 00 00 00 04 02 00 00

```

#### **Hình 4.8 Xác định trường giao thức trong phần tiêu đề gói tin IP tại vị trí 0x9**

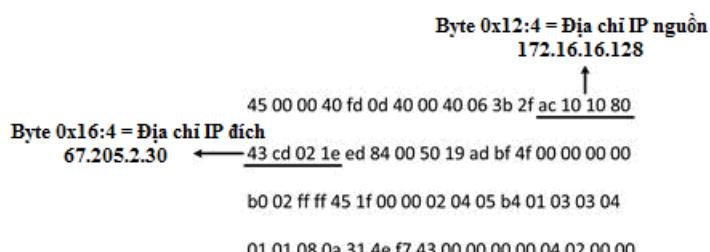
Tiếp tục áp dụng cách này để xác định trường khác như Time-to-Live (TTL), đó là byte offset thứ 8 kể từ 0, và giá trị của trường này là 40 trong hệ cơ số 16 hay 0x40, hoặc 64 trong hệ thập phân.

Chú ý là một vài trường có độ dài nhỏ hơn một byte. Ví dụ như byte 0x0 có 2 trường là phiên bản IP và độ dài tiêu đề gói tin IP. Trong ví dụ này, trường phiên bản IP chỉ là phần nibble cao của byte và độ dài tiêu đề gói tin IP là phần nibble thấp của byte. Do đó, phiên bản của IP là 4. Cũng thấy độ dài tiêu đề của IP là 5, nhưng giá trị thực không phải vậy. Độ dài tiêu đề tính bằng giá trị trong trường tiêu đề nhân với 4, tức là trong trường hợp này có 20 byte. Để dàng suy luận ra rằng độ dài tiêu đề IP lớn nhất là 60 byte do giá trị lớn nhất của trường độ dài tiêu đề IP là 0xF (15 hệ trong thập phân) và  $15 \times 4 = 60$  byte.



**Hình 4.9 Trường phiên bản IP và độ dài tiêu đề IP tại vị trí 0x0**

Ngoài ra, một số trường cũng chiếm nhiều hơn 1 byte như trường địa chỉ nguồn và đích IP với độ dài 4 byte, tại vị trí lần lượt là 0x12 và 0x16. Trong ví dụ, thấy được trường địa chỉ IP nguồn này là ac 10 10 80 (172.16.16.128 trong hệ thập phân), và địa chỉ IP đích là 43 cd 02 1e (67.205.2.30 trong hệ thập phân) (xem Hình 4.10).



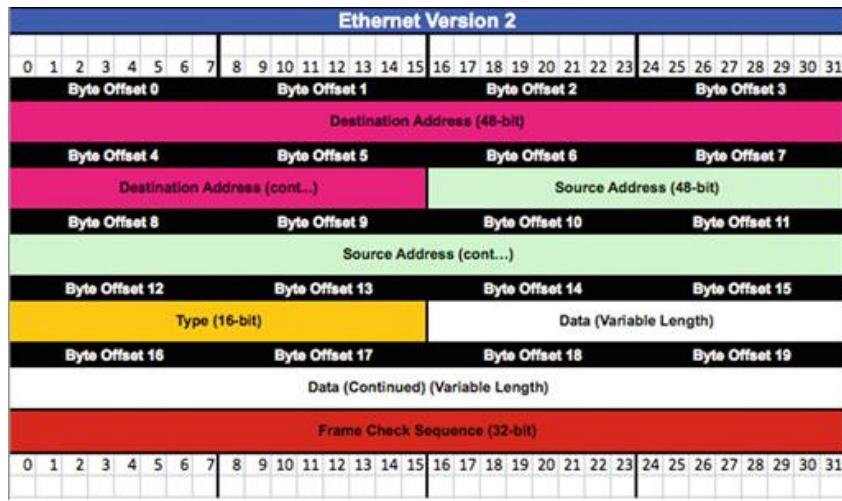
**Hình 4.10 Trường địa chỉ nguồn và đích IP tại vị trí 0x12 và 0x16**

Chú ý rằng khi kí hiệu byte 0x16:4 tức là bắt đầu từ offset byte thứ 16 tính từ 0, và sau đó lấy 4 byte kể từ đây.

#### 4.1.3 Phân tích chi tiết gói tin

Sử dụng một số khái niệm toán học trên, chúng ta tiếp tục phân tích sâu vào gói tin theo từng giao thức. Biết rằng một gói tin được xây dựng bắt đầu với các dữ liệu tầng ứng dụng, rồi các tiêu đề của các giao thức hoạt động ở các tầng thấp hơn được thêm vào, từ trên xuống dưới. Điều này có nghĩa là tiêu đề giao thức cuối cùng được thêm vào là của tầng liên kết dữ liệu. Đây cũng

chính là phần thâý đầu tiên trong gói tin. Giao thức liên kết dữ liệu phổ biến nhất là Ethernet. Tuy nhiên để xác định được thực sự đang nhìn thấy lưu lượng truy cập Ethernet, cần so sánh phần tiêu đề giao thức Ethernet với phần đầu gói tin có được.



*Hình 4.11 Sơ đồ gói tin cho phần tiêu đề Ethernet*

Nhìn vào định dạng tiêu đề Ethernet, thấy rằng 6 byte đầu tiên của gói tin dành riêng cho các địa chỉ MAC đích, và 6 byte tiếp theo bắt đầu từ 0x6 được dành riêng cho các địa chỉ MAC nguồn. Chú ý rằng trường Type (kiểu) 2 byte tại 0x12 được dùng để chỉ ra giao thức sau trường tiêu đề Ethernet. Giá trị hệ cơ số 16 của nó là 08 00, nghĩa là giao thức đóng gói bên trong là IP. Độ dài của tiêu đề Ethernet là 14 byte, do đó 00 là byte cuối của phần tiêu đề.

Thông thường, mặc định là tcpdump không hiển thị phần tiêu đề Ethernet. Do đó cần bắt đầu với giao thức tầng mạng.

Tiêu đề Ethernet	c0 c1 c0 17 8c e8 20 c9 d0 ba 63 fb 08 00 45 00  00 b9 3d 2a 40 00 40 06 fa 99 ac 10 10 80 43 cd  02 1e ed 84 00 50 19 ad bf 50 97 96 65 76 50 18  40 00 cd 7c 00 00 47 45 54 20 2f 20 48 54 54 50  2f 31 2e 31 0d 0a 55 73 65 72 2d 41 67 65 6e 74  3a 20 63 75 72 6c 2f 37 2e 32 34 2e 30 20 28 78  38 36 5f 36 34 2d 61 70 70 6c 65 2d 64 61 72 77  69 6e 31 32 2e 30 29 20 6c 69 62 63 75 72 6c 2f  37 2e 32 34 2e 30 20 4f 70 65 6e 53 53 4c 2f 30  2e 39 2e 38 78 20 7a 6c 69 62 2f 31 2e 32 2e 35  0d 0a 48 6f 73 74 3a 20 61 70 70 6c 69 65 64 6e  73 6d 2e 63 6f 6d 0d 0a 41 63 63 65 70 74 3a 20  2a 2f 2a 0d 0a 0d 0a
------------------	--

*Hình 4.12. Ví dụ về trường tiêu đề Ethernet*

Từ phần tiêu đề Ethernet có thể thấy được phần tiếp theo cần phân tích tiếp là tiêu đề IP (IP header). Do đó cần áp dụng kiến thức về cấu trúc tiêu đề IP cho phần tiếp theo của gói tin. Để tiếp tục phân tích sâu gói tin theo từng giao thức, chúng ta không quan tâm đến từng giá trị đơn lẻ trong tiêu đề này, ngoại trừ một số giá trị như độ dài tiêu đề IP và giao thức tiếp theo cần tìm ra. Trước tiên cần xác định phiên bản IP đang được sử dụng. Nó được chỉ ra bởi các nibble cao của byte 0x0 trong tiêu đề IP, và giá trị có được là IPv4.

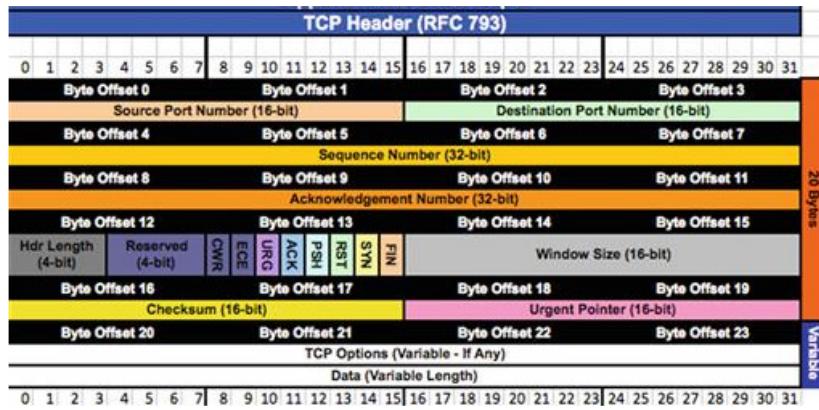
Phần tiêu đề của IP thay đổi tùy vào tập các tùy chọn hỗ trợ. Do đó tiếp theo cần xác định chiều dài phần tiêu đề IP, được chứa trong nibble thấp của byte 0x0 trong tiêu đề IP. Giá trị thấy được là 4, nhân với 5 để có được độ dài tiêu đề là 20 byte. Tức là 2 byte cuối của phần tiêu đề IP là 02 1e.

Giao thức tiếp theo được sử dụng trong gói tin được chỉ ra trong phần tiêu đề IP tại byte 0x9. Ở đây giá trị này là 06, là giá trị diễn tả giao thức TCP (Hình 4.13).

Tiêu đề Ethernet	c0 c1 c0 17 8c e8 20 c9 d0 ba 63 fb 08 00 45 00
Tiêu đề IP	00 b9 3d 2a 40 00 40 06 fa 99 ac 10 10 80 43 cd
	02 1e ed 84 00 50 19 ad bf 50 97 96 65 76 50 18
	40 00 cd 7c 00 00 47 45 54 20 2f 20 48 54 54 50
	2f 31 2e 31 0d 0a 55 73 65 72 2d 41 67 65 6e 74
	3a 20 63 75 72 6c 2f 37 2e 32 34 2e 30 20 28 78
	38 36 5f 36 34 2d 61 70 70 6c 65 2d 64 61 72 77
	69 6e 31 32 2e 30 29 20 6c 69 62 63 75 72 6c 2f
	37 2e 32 34 2e 30 20 4f 70 65 6e 53 53 4c 2f 30
	2e 39 2e 38 78 20 7a 6c 69 62 2f 31 2e 32 2e 35
	0d 0a 48 6f 73 74 3a 20 61 70 70 6c 69 65 64 6e
	73 6d 2e 63 6f 6d 0d 0a 41 63 63 65 70 74 3a 20
	2a 2f 2a 0d 0a 0d 0a

**Hình 4.13 Phân tiêu đề IP**

Sau khi biết được giao thức đóng gói trong IP là TCP, cần tiếp tục kiểm tra xem liệu có dữ liệu của tầng ứng dụng hay không. Để làm được việc này, độ dài của phần tiêu đề TCP cần được xác định bởi vì giống như IP header, độ dài của TCP header phụ thuộc vào tập tùy chọn hỗ trợ.



Hình 4.14 Mô tả định dạng của tiêu đề gói tin TCP

Độ dài phần tiêu đề gói tin TCP xác định bằng cách kiểm tra giá trị trường offset dữ liệu TCP tại nibble cao của byte thứ 0x12. Giá trị trường này là 5, nhân với 4 được độ dài là 20 byte. Nếu đếm quá 20 byte từ chỗ bắt đầu TCP header sẽ thấy vẫn còn dữ liệu sau đó. Đây chính là dữ liệu tầng ứng dụng. Tuy nhiên, không có trường nào trong TCP header mô tả giao thức sử dụng tại tầng ứng dụng. Một cách khác có thể sử dụng để xác định là kiểm tra trường cổng đích (giả sử rằng đây là lưu lượng từ client tới server) tại byte 0x2:2 trong TCP header. Giá trị trường này là 00 50, tức là 80 trong hệ thập phân. Do cổng 80 thường được dùng bởi giao thức HTTP nên có thể dữ liệu này là dữ liệu HTTP. Có thể kiểm tra lại bằng cách so sánh dữ liệu hệ cơ số 16 với định dạng giao thức HTTP, hoặc tách phần dữ liệu từ cuối TCP header đến hết gói tin và chuyển sang dạng ký tự ASCII (Hình 4.15).

Tiêu đề Ethernet	c0 c1 c0 17 8c e8 20 c9 d0 ba 63 fb 08 00 45 00.
Tiêu đề IP	00 b9 3d 2a 40 00 40 06 fa 99 ac 10 10 80 43 cd
Tiêu đề TCP	02 1e ed 84 00 50 19 ad bf 50 97 96 65 76 50 18.
	40 00 cd 7c 00 00 47 45 54 20 2f 20 48 54 54 50
	2f 31 2e 31 0d 0a 55 73 65 72 2d 41 67 65 6e 74
	3a 20 63 75 72 6c 2f 37 2e 32 34 2e 30 20 28 78
	38 36 5f 36 34 2d 61 70 70 6c 65 2d 64 61 72 77
	69 6e 31 32 2e 30 29 20 6c 69 62 63 75 72 6c 2f
	37 2e 32 34 2e 30 20 4f 70 65 6e 53 53 4c 2f 30
	2e 39 2e 38 78 20 7a 6c 69 62 2f 31 2e 32 2e 35
	0d 0a 48 6f 73 74 3a 20 61 70 70 6c 69 65 64 6e
	73 6d 2e 63 6f 6d 0d 0a 41 63 63 65 70 74 3a 20
	2a 2f 2a 0d 0a 0d 0a

Hình 4.15 Phần tiêu đề TCP

Các mức giao thức đã phân tích được mô tả trong Hình 4.16.

<u>Tiêu đề Ethernet</u>	c0 c1 c0 17 8c e8 20 c9 d0 ba 63 fb 08 00 45 00.
<u>Tiêu đề IP</u>	00 b9 3d 2a 40 00 40 06 fa 99 ac 10 10 80 43 cd
<u>Tiêu đề TCP</u>	02 1e ed 84 00 50 19 ad bf 50 97 96 65 76 50 18
<u>Dữ liệu HTTP</u>	40 00 cd 7c 00 00 47 45 54 20 2f 20 48 54 54 50 2f 31 2e 31 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 63 75 72 6c 2f 37 2e 32 34 2e 30 20 28 78 38 36 5f 36 34 2d 61 70 70 6c 65 2d 64 61 72 77 69 6e 31 32 2e 30 29 20 6c 69 62 63 75 72 6c 2f 37 2e 32 34 2e 30 20 4f 70 65 6e 53 53 4c 2f 30 2e 39 2e 38 78 20 7a 6c 69 62 2f 31 2e 32 2e 35 0d 0a 48 6f 73 74 3a 20 61 70 70 6c 69 65 64 6e 73 6d 2e 63 6f 6d 0d 0a 41 63 63 65 70 74 3a 20 2a 2f 2a 0d 0a 0d 0a

**Hình 4.16 Các mức giao thức đã phân tích của một gói tin HTTP**

#### 4.1.4 Phân tích NSM với Tcpdump

Tcpdump là một công cụ bắt và phân tích gói tin tiêu chuẩn trong môi trường Unix, thực hiện phân tích gói tin bằng dòng lệnh. Công cụ này rất hữu ích vì giúp người dùng luôn lấy được dữ liệu khi cần mà không có bất kỳ phiền phức nào. Điều này làm cho tcpdump là công cụ lý tưởng để xem xét các gói tin hay luồng dữ liệu trao đổi. Nhược điểm ở đây cũng chính là sự đơn giản của nó, tức là thiếu các tính năng phân tích bổ trợ như môi trường làm việc đồ họa giống Wireshark. Đồng thời không có khái niệm về trạng thái cũng như khả năng diễn dịch các giao thức ở tầng ứng dụng.

Phần này sẽ mô tả một số kiến thức cơ bản cần khi sử dụng tcpdump. Đầu tiên là khả năng bắt gói tin trực tiếp từ đường truyền với việc chạy trực tiếp tcpdump không cần tham số, tương đương với việc chạy tcpdump để bắt gói tin từ giao diện mạng đánh số thứ tự thấp nhất. Tcpdump mô tả mỗi gói tin bằng cách hiển thị một dòng các thông tin tóm lược ra màn hình. Nếu sử dụng tham số *-i*, sẽ chỉ ra giao diện mạng cần bắt gói tin, và tham số *-nn* để tắt phân giải tên giao thức và host.

Khi bắt gói tin, tốt nhất là càng che giấu (stealthy) càng tốt. Điều này là để giảm các lưu lượng được tạo thêm không liên quan tới sự kiện đang cần xem xét. Do đó, nên dùng tham số *-n* để ngăn chặn việc phân giải tên bởi vì có thể gây ra việc sinh thêm gói tin trên mạng khi thực hiện tiến trình phân giải DNS.

Nếu cần lưu lại gói tin đã bắt giữ để phân tích sau đó, có thể sử dụng tham số *-w* để chỉ ra tên tệp cần lưu. Ví dụ câu lệnh tcpdump như sau:

```
sudo tcpdump -nni eth1 -w packets.pcap
```

Nếu muốn đọc tập tin này, sử dụng tham số *-r* với tên tệp như Hình 4.17.

```

root@kali:~# tcpdump -nrn packets.pcap
reading from file packets.pcap, link-type EN10MB (Ethernet)
09:27:03.037982 IP 172.16.16.128.55957 > 67.205.2.38.88: Flags [S], seq 458297982, win 65535, options [mss 1460,nop,wscale 4,nop,nop,
TS val 884542884 ecr 0,sockOK,eol], length 8
09:27:03.136870 IP 67.205.2.38.88 > 172.16.16.128.55957: Flags [S.], seq 766588929, ack 458297983, win 5848, options [mss 1460,nop,no
p,sockOK,nop,wscale 9], length 8
09:27:03.136967 IP 172.16.16.128.55957 > 67.205.2.38.88: Flags [.], ack 1, win 16384, length 8
09:27:03.137005 IP 67.205.2.38.88 > 172.16.16.128.55957: Flags [P.], seq 1:146, ack 1, win 16384, length 145
09:27:03.239555 IP 67.205.2.38.88 > 172.16.16.128.55957: Flags [.], ack 146, win 14, length 8

```

**Hình 4.17 Đọc gói tin từ một tệp với tcpdump**

Dữ liệu đầu ra của tcpdump mặc định đưa ra một số thông tin cơ bản về mỗi gói tin. Định dạng đầu ra có thể khác nhau tùy thuộc giao thức đang sử dụng, nhưng định dạng phổ biến nhất là:

TCP:

[Timestamp] [Layer 3 Protocol] [Source IP].[Source Port] > [Destination IP].[Destination Port]: [TCP Flags], [TCP Sequence Number], [TCP Acknowledgement Number], [TCP Windows Size], [Data Length]

UDP:

[Timestamp] [Layer 3 Protocol] [Source IP].[Source Port] > [Destination IP].[Destination Port]: [Layer 4 Protocol], [Data Length]

Có thể yêu cầu tcpdump hiển thị thêm thông tin trong phần tóm lược bằng cách thêm tham số “-v”. Số lượng thông tin thêm vào tỷ lệ với số tham số v được thêm, nhiều nhất là 3. Hình 4.18 thể hiện cùng gói tin như trên nhưng với tham số -vvv.

```

root@kali:~# tcpdump -nnvvv packets.pcap
reading from file packets.pcap, link-type EN10MB (Ethernet)
09:27:03.037982 IP (tos 0x0, ttl 64, id 58578, offset 8, flags [DF], proto TCP (6), length 64)
    172.16.16.128.55957 > 67.205.2.38.88: Flags [S], cksum 0xed30 (correct), seq 458297982, win 65535, options [mss 1460,nop,wscale 4
,nop,nop,TS val 884542884 ecr 0,sockOK,eol], length 8
09:27:03.136870 IP (tos 0x0, ttl 64, id 884542884, offset 8, flags [DF], proto TCP (6), length 52)
    67.205.2.38.88 > 172.16.16.128.55957: Flags [S.], cksum 8ce790 (correct), seq 766588929, ack 458297983, win 5848, options [mss 14
60,nop,nop,sockOK,nop,wscale 9], length 8
09:27:03.136967 IP (tos 0x0, ttl 64, id 17335, offset 8, flags [DF], proto TCP (6), length 48)
    172.16.16.128.55957 > 67.205.2.38.88: Flags [.], cksum 0x7ff3 (correct), seq 1, ack 1, win 16384, length 8
09:27:03.137005 IP (tos 0x0, ttl 64, id 9462, offset 8, flags [DF], proto TCP (6), length 105)
    67.205.2.38.88 > 172.16.16.128.55957: Flags [P.], cksum 0x2343 (correct), seq 1:146, ack 1, win 16384, length 145
09:27:03.239555 IP (tos 0x0, ttl 51, id 53997, offset 8, flags [DF], proto TCP (6), length 40)
    67.205.2.38.88 > 172.16.16.128.55957: Flags [.], cksum 0x3eaf (correct), seq 1, ack 146, win 14, length 8

```

**Hình 4.18 Đọc gói tin với nhiều thông tin hơn**

Các dữ liệu này rất hữu dụng nhưng chúng vẫn chưa mô tả hoàn toàn đầy đủ thông tin. Để hiển thị tất cả thông tin về gói tin cần chạy tcpdump với tham số -x, các thông tin được hiển thị dưới dạng hệ cơ số 16 (xem Hình 4.19).

```
maverickz:~ chris$ tcpdump -nnnc packets.pop
reading from file packets.pop, link-type EN10MB (Ethernet)
09:27:03.037982 IP 67.205.2.38.55957 > 67.205.2.38.88: Flags [S], seq 458297982, win 65535, options [mss 1468,nop,vscale 4,nop,nop,
TS val 894542894 ecn 0,sockOK,esi], length 0
0x0000: 4580 0040 c592 4000 4000 720x oct10 1000
0x0010: 43cd 021e dd95 8050 1b51 122e 0000 0000
0x0020: b002 ffff ed38 0000 0204 05d4 0183 0394
0x0030: 0181 000a 346f 0034 0000 0000 0462 0000
09:27:03.136870 IP 67.205.2.38.88 > 172.16.16.128.55957: Flags [S.], seq 766588929, ack 458297983, win 5840, options [mss 1468,nop,no
p,sockOK,vscale 9], length 0
0x0000: 4580 0034 0000 4000 3306 4529 43cd 021e
0x0010: oct10 1000 0050 dd95 2b01 3b01 1b51 122f
0x0020: 0012 1600 e790 0000 0204 05d4 0181 0402
0x0030: 0181 0309
09:27:03.136870 IP 172.16.16.128.55957 > 67.205.2.38.88: Flags [S.], seq 11146, ack 1, win 16384, length 0
0x0000: 4580 0028 43d7 4000 4000 f494 oct10 1000
0x0010: 43cd 021e dd95 0000 1b51 122f 2b01 3b02
0x0020: 5010 4000 f137 0000
09:27:03.137685 IP 172.16.16.128.55957 > 67.205.2.38.88: Flags [P.], seq 11146, ack 1, win 16384, length 145
0x0000: 4580 0009 24f6 4000 4000 12ce oct10 1000
0x0010: 43cd 021e dd95 0000 1b51 122f 2b01 3b02
0x0020: 5010 4000 2343 0000 4745 5428 2720 4054
0x0030: 5458 2731 2b31 0000 5573 6572 2b01 6765
0x0040: 6674 3d20 6375 725c 2137 2b32 342e 3d20
0x0050: 2070 3d06 5f36 3d20 6178 790c 652d 6464
0x0060: 7277 696e 3103 2630 2920 6e09 6263 7972
0x0070: 622f 372e 3234 2630 2041 7065 6453 5340
0x0080: 2138 2639 2e30 7828 7060 6962 2731 2e32
0x0090: 2635 0000 4000 2724 3d20 6170 7060 6965
0x00a0: 6464 23d0 2e03 6f6d 0000 4163 6365 7074
0x00b0: 3d20 2a2f 2b04 8000 80
09:27:03.239655 IP 67.205.2.38.88 > 172.16.16.128.55957: Flags [S.], seq 766588929, ack 458297983, win 14, length 0
0x0000: 4580 0028 43d7 4000 3306 7247 43cd 021e
0x0010: oct10 1000 0050 dd95 2b01 3b02 1b51 1200
0x0020: 5010 000e 3eaf 0000 0000 4328 1d59
```

**Hình 4.19 Xem đầy đủ các gói tin dưới dạng hệ cơ số 16**

Một cách khác là hiển thị các gói tin dưới dạng ASCII, với tham số  $-A$  (Hình 4.20).

```
maverickz:~ chris$ tcpdump -nnnc packets.pop
reading from file packets.pop, link-type EN10MB (Ethernet)
09:27:03.037982 IP 67.205.2.38.55957 > 67.205.2.38.88: Flags [S], seq 458297982, win 65535, options [mss 1468,nop,vscale 4,nop,nop,
TS val 894542894 ecn 0,sockOK,esi], length 0
E..4..4.8.r....C.....P.0.....0. .....
4.
4.....
09:27:03.136870 IP 172.16.16.128.55957 > 67.205.2.38.88: Flags [S.], seq 766588929, ack 458297983, win 5840, options [mss 1468,nop,no
p,sockOK,vscale 9], length 0
E..4..4.3.EYC.....P.0...0.0. .....
09:27:03.136870 IP 172.16.16.128.55957 > 67.205.2.38.88: Flags [S.], seq 11146, ack 1, win 16384, length 0
E..(0.9.4. .....
C.....P.Q./-8.P.0...0. .....
09:27:03.137685 IP 172.16.16.128.55957 > 67.205.2.38.88: Flags [P.], seq 11146, ack 1, win 16384, length 145
E..4..8. .....
C.....P.Q./-8.P.0...0. .....
User-Agent: curl/7.24.0 (x86_64-apple-darwin12.0) libcurl/7.24.0 OpenSSL/0.9.8x zlib/1.2.5
Host: www.apple.com
Accept: */*
09:27:03.239655 IP 67.205.2.38.88 > 172.16.16.128.55957: Flags [S.], seq 11146, ack 146, win 14, length 0
E..(0.9.3.rC.....P. .....
Q. .....
C.....P. .....
C.....Q. .....
C.....Q. .....
maverickz:~ chris$
```

**Hình 4.20 Xem đầy đủ các gói tin dưới dạng ASCII**

Một tham số hữu ích khác là  $-X$ , được dung để hiển thị gói tin dưới cả 2 dạng hệ cơ số 16 và ASCII cạnh nhau (Hình 4.21).

```

root@localhost: ~# tcpdump -nnr packets.pcap
reading from file packets.pcap, link-type EN10MB (Ethernet)
09:27:03.037982 IP 172.16.16.128.55957 -> 172.26.2.30.88: Flags [S], seq 456297982, win 65535, options [mss 1468,nop,wscale 4,nop,nop,
TS val 89452804 ec 0,sock0,noe], length 0
0x0000: 4520 0040 c092 0000 4006 7202 0c18 1000 E..4..4.r.....
0x0001: 43cd 0216 0d95 0050 1051 1226 0000 0000 C.....P.Q.....
0x0002: b682 ffff ed20 0000 0204 0524 0103 0304 .....8.....
0x0003: 0101 0000 3409 0034 0000 0008 0002 0000 ....4.....
09:27:03.136671 IP 67.26.2.30.88 -> 172.16.16.128.55957: Flags [D], seq 766589929, ack 456297983, win 5040, options [mss 1468,nop,no
o,soo,wscale 9], length 0
0x0000: 4520 0040 c092 0000 4006 7202 0c18 1000 E..4..4.r.....
0x0001: 0c18 1000 0050 2811 3001 1051 1227 .....P...8.Q./
0x0002: 0012 1608 e790 0000 0204 0524 0101 0402 .....P.....
0x0003: 0103 0309
09:27:03.136907 IP 172.16.16.128.55957 -> 172.26.2.30.88: Flags [S], seq 1, win 16384, length 0
0x0000: 4520 0029 0d27 0006 4006 f493 0c18 1000 E..(C.4.4.....
0x0001: 43cd 0216 0d95 0050 1051 1227 2801 3002 C.....P.Q./-0.
0x0002: 5010 0000 0f31 0000 .....P.Q.?
09:27:03.137895 IP 172.16.16.128.55957 -> 172.26.2.30.88: Flags [P], seq 1:146, ack 1, win 16384, length 146
0x0000: 4520 0029 0d16 0000 4006 1226 0c18 1000 E..$4.4.....
0x0001: 43cd 0216 0d95 0050 1051 1227 2801 3002 C.....P.Q./-0.
0x0002: 5010 0029 0d41 0000 4745 5428 2128 4054 P..4..4..ET-/H
0x0003: 5450 2731 2601 0000 5573 6572 2411 6765 T/1.1..User-Age
0x0004: 6e74 3420 6375 726 2137 2632 3426 3829 nt1.cur1(7.24.8.
0x0005: 2070 3000 513 3420 6179 7060 6220 6462 (<0.54..apple-de
0x0006: 7277 6966 1137 2810 2920 5057 5263 7572 rwin12.0..1.1bcut
0x0007: 6254 772 823 2041 7065 6453 5340 0/7.24.8..OpenSS
0x0008: 2730 2457 2428 7620 7650 6962 2131 2022 />9.8..11b/4.2
0x0009: 2005 0000 0f6f 7374 3020 6170 7060 6965 >..Host-Applic
0x000a: 6464 7364 2603 6664 0000 4163 6395 7074 dnsm.con..Accept
0x000b: 3420 2021 2089 800 80 1.4%...
09:27:03.239556 IP 67.26.2.30.88 -> 172.16.16.128.55957: Flags [F], seq 146, win 14, length 0
0x0000: 4520 0029 0d27 0006 3396 7247 0c18 0104 E..(.,4.4.rC...
0x0001: 0c10 1008 0050 2801 3002 1051 1208 .....P...8.Q./
0x0002: 5010 0000 0f61 0000 0000 4120 F.....P.....

```

**Hình 4.21 Xem đầy đủ các gói tin dưới dạng ASCII và hệ cơ số 16**

Trong nhiều trường hợp phải làm việc với các tệp tin PCAP lớn hơn, cần phải sử dụng các bộ lọc để chỉ hiện thị các dữ liệu cần xem xét, hoặc lọc các dữ liệu không có giá trị. Tcpdump sử dụng định dạng BPF (Berkeley Packet Filter). Có thể gọi bộ lọc bằng cách thêm vào cuối câu lệnh, đặt trong dấu ngoặc đơn cho dễ nhìn. Ví dụ nếu chỉ muốn xem các gói tin với cổng đích TCP là 8080, có thể gọi lệnh sau:

```
tcpdump -nnr packets.pcap 'tcp dst port 8080'
```

Hoặc sử dụng tham số `-w` để tạo tệp tin mới chỉ chứa các gói tin phù hợp với bộ lọc này:

```
tcpdump -nnr packets.pcap 'tcp dst port 8080' -w packets_tcp8080.pcap
```

Trường hợp cần làm việc với một số lượng lớn các tham số tùy chọn khi phân tích các gói tin bắt giữ, để dễ dàng cho việc chỉnh sửa, tcpdump cho phép sử dụng tham số `-F` để xác định một tệp tin lọc chứa các tham số BPF.

Ngoài ra, để biết thêm các cách sử dụng tham số khác, có thể tham khảo tại trang web <http://www.tcpdump.org>. Hoặc tham khảo thêm công cụ Tshark tương tự tcpdump nhưng có thêm một số tính năng hữu ích như thống kê và lọc theo giao thức tầng ứng dụng.

#### 4.1.5 Phân tích NSM với Wireshark

Phân tích gói tin bằng dòng lệnh hữu ích khi tương tác với gói tin ở mức cơ bản nhưng thiếu các tính năng phân tích cao cấp như các ứng dụng phân tích gói tin có môi trường đồ họa kiểu như Wireshark. Tên trước đó của Wireshark là Ethereal. Wireshark được cài sẵn trong Security Onion.

##### 4.1.5.1 Bắt gói tin

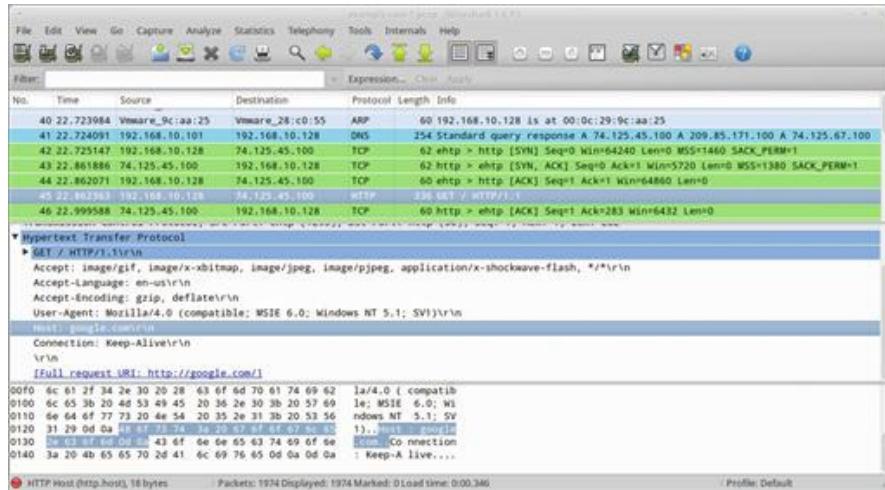
Để bắt gói tin từ đường truyền, chọn Capture -> Interfaces từ menu để hiển thị tất cả các giao diện mạng (Hình 4.22). Có thể chọn bắt gói tin từ một giao diện cảm biến hay giao diện khác bằng cách nhấn nút Start bên cạnh. Chú ý là Wireshark đọc tất cả các gói tin vào bộ nhớ, do đó

khi làm việc với lượng dữ liệu mạng lớn, nên bắt với các công cụ chạy trên dòng lệnh, ghi vào tệp và làm việc với Wireshark sau đó qua tệp.



**Hình 4.22** *Bắt gói tin với Wireshark*

Kết thúc việc bắt gói tin bằng cách nhấn nút Stop dưới menu Capture. Các gói tin đã bắt hiển thị trong chương trình giống như Hình 4.23.



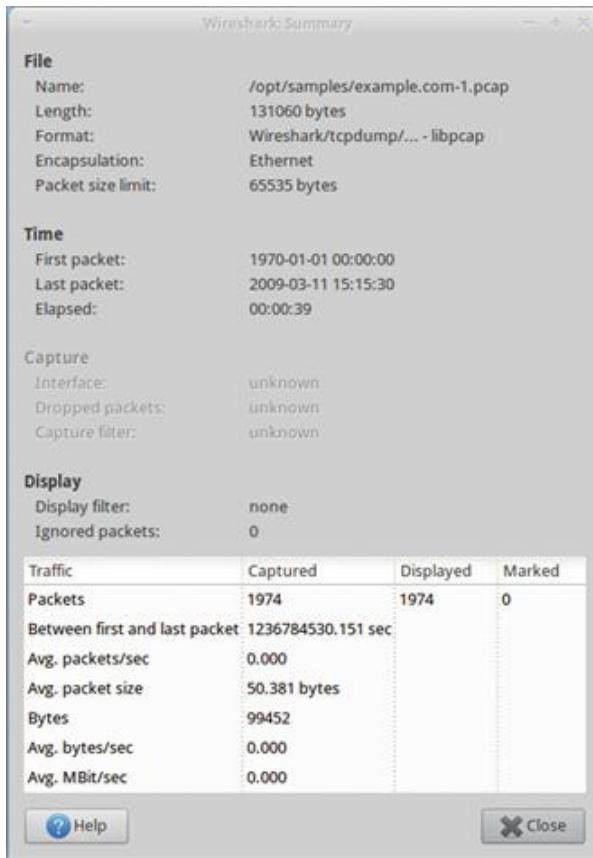
**Hình 4.23 Xem gói tin trong Wireshark**

Xem hình trên thấy rằng màn hình Wireshark chia ra làm 3 phần. Phần trên là danh sách các gói tin, được dùng để hiển thị tóm lược về thông tin gói tin, mỗi dòng một gói. Phần giữa hiển thị chi tiết về gói tin với từng trường dữ liệu trong gói. Phần cuối hiển thị chi tiết đến từng byte trong gói tin, dưới hệ cơ số 16 và định dạng ASCII, tương tự tham số  $-X$  trong tcpdump.

Wireshark có rất nhiều tính năng hữu ích cho phân tích gói tin. Tài liệu này chỉ mô tả các tính năng cơ bản và cần thiết nhất. Để biết đầy đủ các tính năng, có thể tham khảo trong một số tài liệu tiếng Anh như “Practical Packet Analysis” hay “Wireshark Network Analysis”.

**a. Xem tóm tắt**

Tính năng này hiển thị bản tóm lược các thông tin và thống kê về gói tin đã bắt giữ cùng dữ liệu trong đó (Hình 4.24).



Hình 4.24 Cửa sổ tóm lược của Wireshark

Các thông tin có được bao gồm:

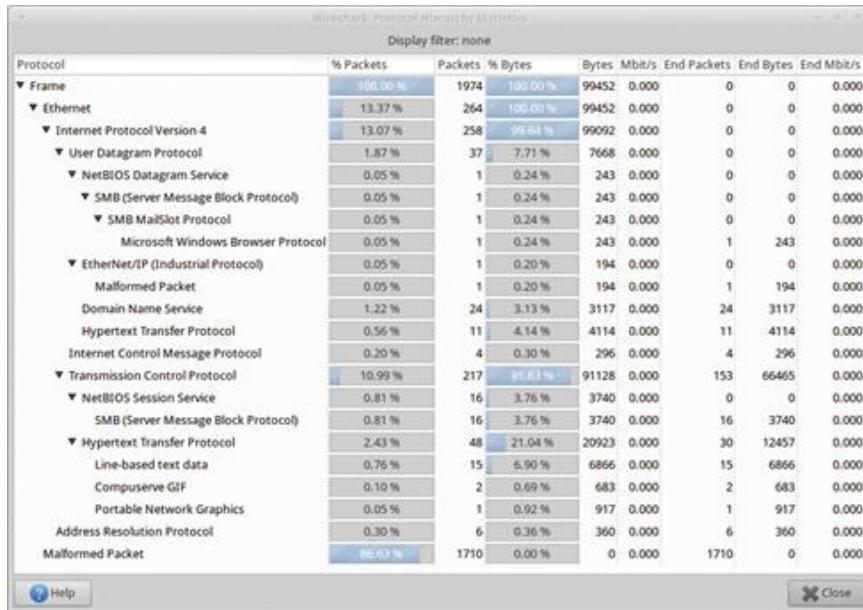
- Định dạng của file
- Thời gian bắt gói tin trong file
- Kích cỡ dữ liệu theo byte
- Kích thước trung bình gói tin
- Số byte trung bình/ giây và số Mbit/giây

### b. Cây giao thức

Màn hình cây giao thức được hiển thị khi bấm vào Statistics -> Protocol Hierarchy trong menu chính. Cây giao thức cung cấp thông tin về các giao thức trong tệp gói tin bắt giữ cùng với các thông tin thống kê về phân trăm lưu lượng theo từng giao thức.

Tính năng thống kê này rất quan trọng khi phân tích gói tin bởi vì nhờ nó mà người dùng có thể nhanh chóng xác định được các giao thức không mong muốn hoặc bất thường. Chẳng hạn như khi muốn phân tích lưu lượng của SMB nhưng lại không thấy máy tính có Windows hay dịch vụ Samba trên mạng. Hơn nữa, người dùng cũng có thể sử dụng tính năng này để xác định tỷ lệ phân bố không hợp lý của các giao thức đang cần xem xét. Ví dụ như nếu các gói tin bắt giữ có số lượng thuộc giao thức DNS hay ICMP, cần phải xem xét kỹ càng hơn về lưu lượng này.

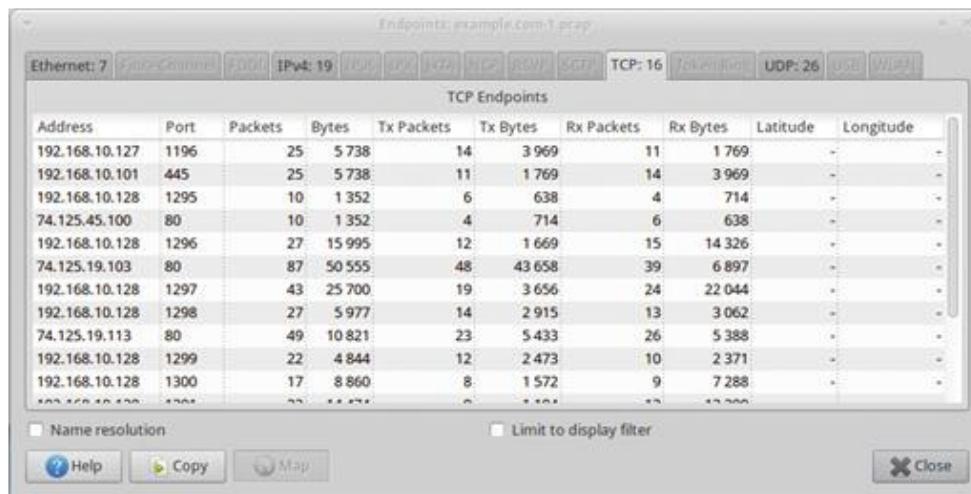
Cũng có thể tạo bộ lọc hiển thị trực tiếp từ cửa sổ thông qua bấm chuột phải vào một giao thức rồi chọn “Apply as filter”.



Hình 4.25 Cửa sổ cây giao thức của Wireshark

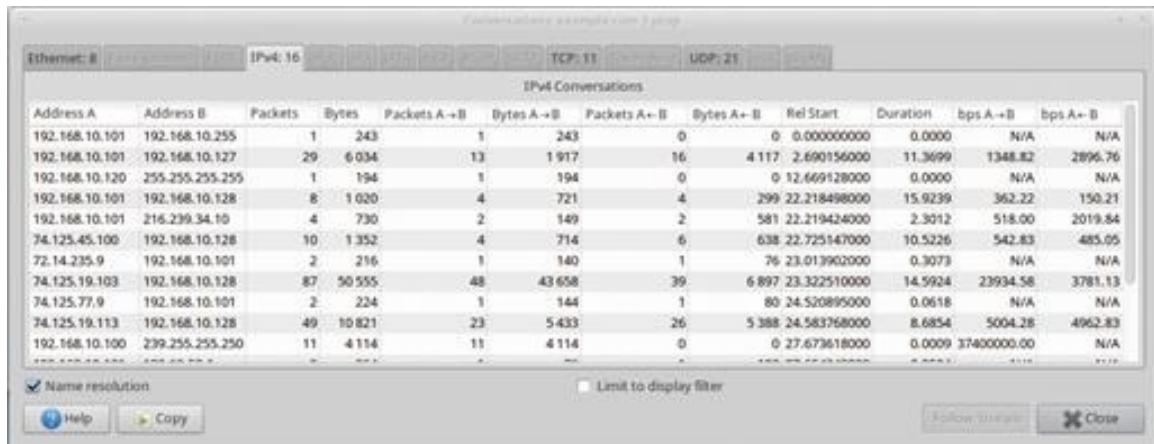
### c. Các thiết bị đầu cuối và các lưu lượng hội thoại

Trong Wireshark, một thiết bị trao đổi trên mạng gọi là thiết bị đầu cuối. Khi hai thiết bị đầu cuối truyền tin cho nhau thì gọi đó là một cuộc hội thoại. Trong Wireshark, để xem thống kê về lưu lượng trao đổi đối với từng đầu cuối, chọn Statistics -> Endpoints từ menu chính (Hình 4.26).



Hình 4.26 Cửa sổ về các đầu cuối trong Wireshark

Còn nếu muốn xem thống kê lưu lượng hội thoại, chọn Statistics > Conversations từ menu chính (Hình 4.27).



**Hình 4.27 Cửa sổ xem thông kê về lưu lượng hội thoại trong Wireshark**

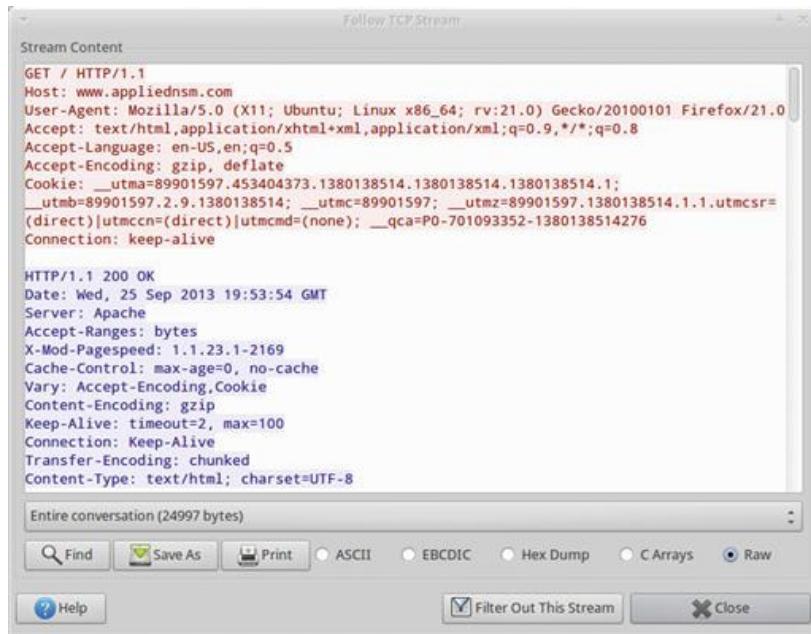
Cả hai cửa sổ đều có cách trình bày giống nhau. Chú ý rằng trên từng cửa sổ đều có nhiều tab cho phép xem về các giao thức khác nhau. Wireshark phân rã đầu cuối và các hội thoại theo các giao thức này và theo địa chỉ dựa trên các tầng. Do vậy, một đầu cuối Ethernet đơn lẻ có thể gắn với nhiều đầu cuối IPv4. Tương tự, một hội thoại giữa vài địa chỉ IP có thể bị giới hạn trong hai thiết bị vật lý, mỗi thiết bị có một địa chỉ MAC Ethernet.

Các cửa sổ này có ý nghĩa khi xác định thành phần nào có vai trò quan trọng trong tệp tin dữ liệu gói tin. Có thể xác định máy tính nào có lưu lượng truyền lớn nhất và từ đó giúp người dùng tập trung được vào mục tiêu. Cũng giống như cửa sổ cây giao thức, cửa sổ này cho phép tạo bộ lọc trực tiếp từ màn hình bằng cách bấm chuột phải vào một đầu cuối hay một hội thoại.

#### d. Hiển thị luồng dữ liệu

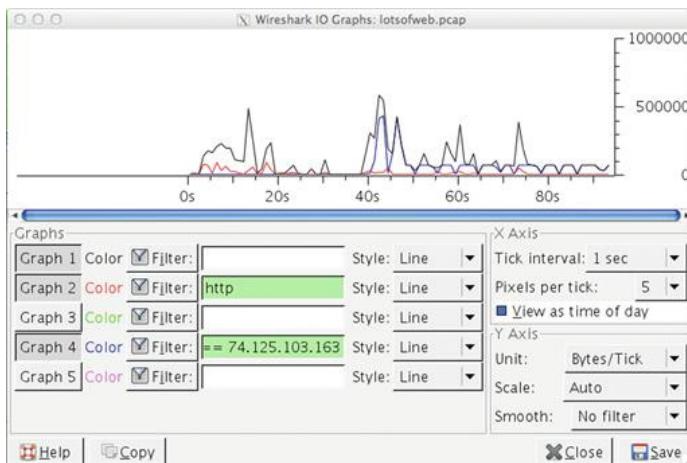
Ngoài khả năng hiển thị lưu lượng trong một hội thoại thông qua bộ lọc, Wireshark hỗ trợ tính năng hiển thị luồng dữ liệu trên tầng ứng dụng, bằng cách nhấn chuột phải vào một gói TCP, chọn “Follow TCP Stream”.

Hình 4.28 hiển thị luồng TCP của một kết nối HTTP. Wireshark lấy dữ liệu tầng ứng dụng chừa trong các gói tin của hội thoại, hợp nhất chúng đồng thời loại bỏ các thông tin tầng dưới. Tính năng này cho phép người dùng nhanh chóng xem được những gì đang diễn ra trong trao đổi HTTP hiện tại. Ngoài ra, người dùng có thể chỉ hiển thị thông tin trao đổi từ một hướng. Tương tự, tính năng này cũng có thể sử dụng trên luồng dữ liệu UDP và SSL.



**Hình 4.28 Chức năng “Following a TCP Stream”**

### e. Đồ thị IO



**Hình 4.29 Xem lưu lượng gói tin với đồ thị IO**

Nếu muốn xác định lưu lượng của gói tin trong một thời gian nào đó, Wireshark có tính năng tạo đồ thị IO như Hình 4.29. Hình vẽ hiển thị đồ thị cơ bản về lưu lượng cho một gói tin đơn lẻ. Trong trường hợp này, có một đường mô tả lưu lượng tất cả gói tin có trong tệp tin dữ liệu (đồ thị 1), và hai đường khác mô tả lưu lượng các gói tin phù hợp với bộ lọc hiển thị. Một trong số đó hiển thị tất cả lưu lượng HTTP (đồ thị 3) và cái còn lại hiển thị lưu lượng tạo ra từ máy tính với IP là 74.125.103.164 (đồ thị 4).

### e. Trích xuất đối tượng

Wireshark có khả năng phát hiện các tệp tin truyền trong lưu lượng đối với một giao thức nhất định. Nhờ đó, người dùng có thể trích xuất các tập tin này từ các gói tin đã bắt được. Wireshark hỗ trợ trích xuất tệp tin từ luồng dữ liệu HTTP, SMB và DICOM.

Các bước thực hiện như sau:

1. Mở tiến trình bắt gói tin trên một giao diện mạng
2. Mở trình duyệt và vào một số trang web
3. Dừng bắt gói tin
4. Từ menu chính, chọn File -> Export -> Objects -> HTTP
5. Wireshark hiển thị danh sách các file có trong lưu lượng đã bắt được (Hình 4.30). Người dùng chọn đối tượng cần trích xuất và nhấn Save As, sau đó chọn nơi lưu trữ tệp tin cùng tên tệp tin để lưu.

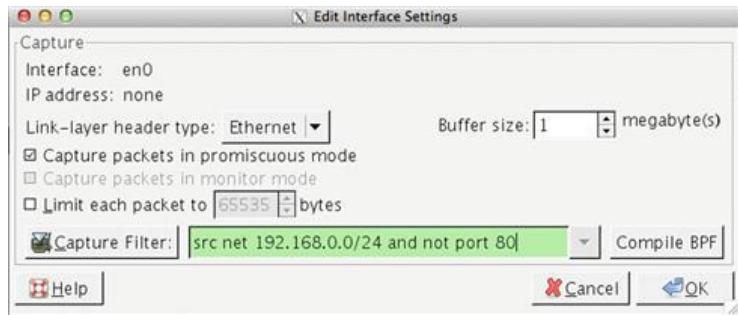
Packet num	Hostname	Content Type	Bytes	Filename
1969	online.wsj.com	text/xml	1773	lexus-portal.xml
1979	admedia.wsod.com	image/gif	3008	7_texture_170x40.gif
2011	online.wsj.com	image/jpeg	4042	lexus-portal-1.jpg
2024	online.wsj.com	image/jpeg	3581	lexus-portal-2.jpg
2037	wsj.vo.llnwd.net	image/jpeg	4728	112009hubpm_167x94.jpg
2055	online.wsj.com	text/html	4929	newentries.sync?r=856
2172	www.youtube.com	text/html	159752	watch?v=CMNry4PE93Y
2181	s.ytimg.com	application/x-javascript	1656	www-csi-vfl133369.js
2189	static.2mdn.net	application/x-shockwave-flash	60560	PID_1158125_child.swf
2206	i2.ytimg.com	image/jpeg	3609	default.jpg
2261	i2.ytimg.com	image/jpeg	3010	default.jpg
2266	i4.ytimg.com	image/jpeg	2741	default.jpg
2274	i3.ytimg.com	image/jpeg	3178	default.jpg
2282	i1.ytimg.com	image/jpeg	2833	default.jpg
2287	i1.ytimg.com	image/jpeg	1959	default.jpg
...				

Hình 4.30 Chọn một đối tượng HTTP để trích xuất

Một số các công cụ khác như khung làm việc phân tích tệp tin của Bro cũng có tính năng này nhưng sử dụng Wireshark tiện hơn rất nhiều.

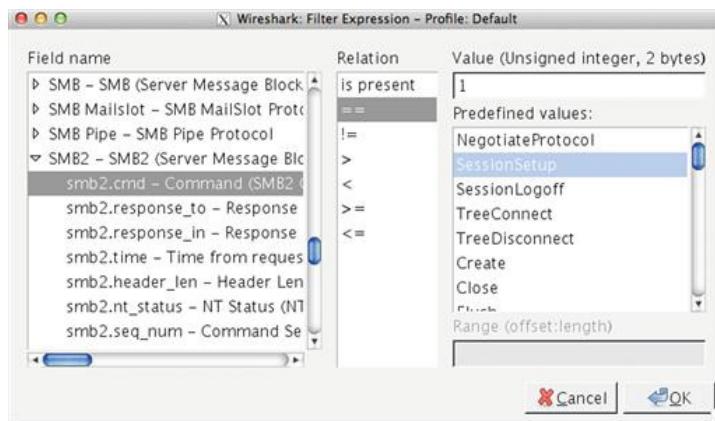
### e. Bộ lọc hiển thị và bắt gói tin

Wireshark cho phép sử dụng các bộ lọc bắt gói tin theo định dạng BPF cũng như các bộ lọc hiển thị để tương tác với các trường được tạo ra bởi các bộ phân tích giao thức. Bộ lọc bắt gói tin theo định dạng BPF trong Wireshark chỉ dùng để bắt dữ liệu. Để dùng một bộ lọc này, chọn Capture -> Options trong menu chính rồi nhấn kép vào giao diện mạng sẽ bắt gói tin, và cuối cùng đặt bộ lọc bắt gói tin và vùng hội thoại Capture Filter (Hình 4.31) và nhấn OK. Nay giờ, khi nhấn Start trên màn hình trước đó, bộ lọc bắt gói tin sẽ chạy và các gói tin không thỏa mãn điều kiện sẽ bị loại bỏ. Trong ví dụ này, thử dùng một bộ lọc để bắt gói tin từ mạng nguồn 192.168.0.0/24 mà không sử dụng cổng 80.



**Hình 4.31 Chọn một bộ lọc bắt gói tin**

Bộ lọc hiển thị có thể sử dụng bằng cách gõ trực tiếp vào hộp thoại lọc trên phần cửa sổ mô tả các gói. Khi đó Wireshark chỉ hiển thị các gói phù hợp với điều kiện. Khi muốn bỏ bộ lọc thì chỉ cần nhấn Clear. Cũng có thể tạo một bộ lọc nâng cao bằng cách nhấn nút Expression gần với hộp thoại lọc hiển thị (Hình 4.32).



**Hình 4.32 Tạo bộ lọc hiển thị với bộ tạo Expression Builder**

Hình 4.32 hiển thị biểu thức lọc cho các yêu cầu SessionSetup của giao thức SMB2.

#### 4.1.6 Lọc gói tin

Các bộ lọc hiển thị và bắt gói tin cho phép người dùng chỉ ra các gói tin muốn xem hoặc không muốn xem khi làm việc với một file dữ liệu. Khi phân tích các gói tin, phần lớn thời gian của người phân tích dành cho việc lọc dữ liệu thành các đoạn dữ liệu có giá trị xem xét. Do đó hiểu được về lọc gói tin và cách sử dụng trong các trường hợp khác nhau là rất quan trọng. Phần này sẽ xem xét hai loại cú pháp lọc gói tin là BPF (lọc bắt gói tin) và lọc hiển thị của Wireshark/tshark.

##### a. BPF (Berkeley Packet Filter)

Cú pháp BPF là cú pháp lọc gói tin phổ biến nhất, được sử dụng trong nhiều ứng dụng xử lý gói tin như tcpdump, Wireshark, tshark. BPF được dùng trong khi thu thập dữ liệu nhằm loại bỏ các dữ liệu không mong muốn, những dữ liệu không có ích trong việc phát hiện và phân tích, hay có thể được dùng trong khi đang phân tích lưu lượng đã được thu thập từ cảm biến.

Một bộ lọc sử dụng cú pháp BPF gọi là một biểu thức. Các biểu diễn này có cấu trúc và bộ khung cụ thể, gồm một hoặc nhiều đơn vị kết hợp với nhau bằng các phép toán. Ví dụ về định dạng biểu thức được hiển thị trong Hình 4.33.



Hình 4.33 Ví dụ về biểu thức BPF

Nhìn hình sẽ thấy có 2 đơn vị biểu thức udp port 53 và dst host 192.0.2.2. Toán tử ở giữa là toán tử liên kết (&&).

### b. Bộ lọc hiển thị Wireshark

Wireshark và tshark đều cung cấp tính năng sử dụng bộ lọc hiển thị. Tính năng này khác với bộ lọc bắt dữ liệu bởi vì nó sử dụng bộ phân tích giao thức để bắt thông tin về các trường giao thức riêng lẻ. Wireshark hỗ trợ khoảng 1000 giao thức và 141.000 trường giao thức. Không giống bộ lọc bắt gói tin, bộ lọc hiển thị chỉ sử dụng trên dữ liệu đã bắt.

Bảng 4.1 Các toán tử so sánh của bộ lọc hiển thị

Toán tử (Tiếng Anh)	Toán tử Giống trong C	Mô tả	Ví dụ
eq	==	So sánh các giá trị bằng với một giá trị cụ thể	ip.addr == 192.168.1.155
ne	!=	So sánh các giá trị khác với một giá trị cụ thể	ip.addr != 192.168.1.155
gt	>	So sánh các giá trị lớn hơn một giá trị cụ thể	tcp.port gt 1023
lt	<	So sánh các giá trị nhỏ hơn một giá trị cụ thể	tcp.port < 1024
ge	>=	So sánh các giá trị lớn hơn hoặc bằng một giá trị cụ thể	udp.length >= 75
le	<=	So sánh các giá trị nhỏ hơn hoặc bằng một giá trị cụ thể	udp.length le 75
contains		So sánh các giá trị mà trong đó một giá trị cụ thể được chứa trong một trường	smtp.req.parameter contains "FROM"

Bộ lọc hiển thị cũng được mô tả bởi các biểu thức. Bao gồm tên trường, toán tử so sánh và một giá trị. Tên trường có thể là giao thức hay một trường thuộc giao thức. Tiếp theo là toán tử so sánh (hay gọi là toán tử quan hệ), được dùng để mô tả cách so sánh hai giá trị xác định có liên quan. Và phần tử cuối cùng là giá trị để dùng cho việc so sánh.

## 4.2. TRI THỨC VỀ NGUY CƠ BẢO MẬT VÀ TÀI NGUYÊN CẦN BẢO VỆ

Tri thức về nguy cơ bảo mật và tài nguyên cần bảo vệ (Friendly and threat intelligence – TI) được hiểu đơn giản là những tri thức giúp xác định các mối đe dọa bảo mật và đưa ra quyết định đúng đắn. TI có thể giúp giải quyết các vấn đề sau:

- Làm thế nào để cập nhật khôi lượng thông tin không lồ về các nguy cơ an ninh như các nhân tố xấu, phương thức tấn công, lỗ hổng, đối tượng,...?
- Làm thế nào để có được nhiều hơn tri thức về tương lai các nguy cơ bảo mật?
- Làm thế nào để thông báo đến người quản lý về các nguy hiểm và hậu quả của một nguy cơ cụ thể?

TI nhận được rất nhiều quan tâm gần đây. Có nhiều định nghĩa khác nhau về TI và dưới đây là một vài trích dẫn thường gặp:

- “TI là tri thức dựa trên bằng chứng, bao gồm ngữ cảnh, cơ chế, IOC, các tin tức liên quan về một mối đe dọa hoặc nguy cơ đang hiện diện đối với tài sản – có thể được sử dụng để đưa ra quyết định phản ứng xử lý đối với mối đe dọa đó” – Gartner
- “Tập hợp dữ liệu thu thập được, định mức và áp dụng đối với mỗi đe dọa bảo mật, nhân tố đe dọa, khai thác, mã độc, lỗ hổng và các chỉ số xâm hại” – SANS Institute

Theo báo cáo của Verizon, trong năm 2015 các công ty thiệt hại khoảng 400 triệu USD do rò rỉ thông tin (từ 79,790 sự cố bảo mật). Các nguy cơ (mối đe dọa) và rò rỉ luôn xảy ra khiến mọi tổ chức phải tìm cách bảo vệ chính mình. Tuy nhiên các nguy cơ luôn thay đổi và rủi ro càng tăng cao do các tổ chức phải phụ thuộc vào hệ thống IT của mình.

**Bảng 4.2 Một số IOC thông thường**

	<b>IOC</b>	<b>Ví dụ</b>
Mạng	<ul style="list-style-type: none"><li>• Địa chỉ IP</li><li>• URL</li><li>• Tên miền</li></ul>	Mã độc lây nhiễm vào các host nội bộ liên quan đến các nhân tố độc hại đã biết.
Thu điện tử	<ul style="list-style-type: none"><li>• Địa chỉ người gửi thư, tên thư.</li><li>• Tệp tin đính kèm</li><li>• Đường dẫn</li></ul>	Các nỗ lực lừa đảo máy chủ nội bộ nhấn vào một thư đáng ngờ và gửi đến một máy chủ điều khiển độc hại
Dựa trên máy chủ	<ul style="list-style-type: none"><li>• Tên tệp tin và hàm băm của tệp tin (như MD5)</li><li>• Khóa đăng ký</li><li>• Thư viện đường dẫn động (DLL)</li><li>• Tên Mutex</li></ul>	Các vụ tấn công từ bên ngoài bắt đầu từ các máy chủ hoặc các hành vi độc hại đã được biết đến.

Nguy cơ đến từ cả nguồn nội bộ lẫn từ bên ngoài khiến các tổ chức rất căng thẳng khi đối mặt với chúng. Mặc dù thông tin dưới dạng dữ liệu gốc có sẵn rất đa dạng, nhưng rất khó và tốn thời gian để lấy ra thông tin có ích rồi đưa ra giải pháp cần thiết.

Các yếu tố đó thúc đẩy càng nhiều người sử dụng TI, giúp phân loại nguy cơ với lượng dữ liệu, cảnh báo không lò, phân loại tấn công và cung cấp thông tin hữu ích.

Bảng 4.2 thể hiện một vài IOC thông thường của các vụ tấn công mà TI có thể xác định:

#### 4.2.1 Chu trình thu thập tri thức về nguy cơ bảo mật cho NSM

Việc thu thập tri thức được thực hiện theo một khung làm việc được gọi là Chu trình thu thập tri thức (Intelligence Cycle). Phụ thuộc vào nguồn tham khảo đến, chu trình thu thập có thể được chia thành nhiều bước. Trong tài liệu này, chúng ta mô tả chu trình gồm 6 bước, bao gồm: xác định yêu cầu, lập kế hoạch, thu thập, xử lý và truyền đi. Các bước trong chu trình có thể tiếp tục tự cung cấp thông tin cho chính nó để có thể tự hoàn thiện sau mỗi chu trình (Hình 4.34)



**Hình 4.34 Chu trình thu thập tri thức truyền thống**

Phần sau sẽ xem xét cụ thể từng bước khi áp dụng vào việc xây dựng thu thập tri thức về các nguy cơ bảo mật cũng như các tài nguyên cần bảo vệ cho NSM.

##### Bước 1: Xác định yêu cầu

Một sản phẩm TI được tạo ra dựa trên một yêu cầu xác định. Yêu cầu này cũng là nguồn để tạo ra các pha khác trong chu trình thu thập kiến thức. Trong lĩnh vực an toàn thông tin và NSM, yêu cầu trên cũng được tập trung vào nhu cầu về thông tin liên quan tới các tài nguyên cần bảo vệ (tri thức về tài nguyên cần bảo vệ), hoặc vào thông tin liên quan tới các máy tính có khả năng gây ra các mối đe dọa bảo mật tới các tài nguyên cần bảo vệ (tri thức về các mối đe dọa bảo mật).

Các yêu cầu này là các bản tin yêu cầu về thông tin và ngữ cảnh cho phép các chuyên gia NSM có thể đưa ra quyết định phù hợp với mục tiêu điều tra của họ. Trong pha này, tất cả đều là yêu cầu về các câu hỏi phù hợp. Các câu hỏi này phụ thuộc vào việc liệu các yêu cầu thu thập tri thức là liên tục hay theo tình huống. Ví dụ, việc phát triển một sản phẩm TI về tài nguyên cần bảo

về là một quá trình liên tục, có nghĩa là các câu hỏi nên được đặt ra một cách tổng quát và có thể lặp lại.

Một số câu hỏi thiết kế để tạo ra tri thức cơ sở cho các mẫu truyền tin bình thường có thể được viết như sau:

- Các mẫu về giao tiếp bình thường giữa các máy tính là như nào?
- Các mẫu về giao tiếp bình thường giữa các máy tính cần chú ý bảo vệ với các thực thể ngoài không rõ là như nào?
- Các dịch vụ nào thường được cung cấp bởi các máy tính bình thường?
- Tỷ lệ giao tiếp từ trong ra ngoài của các máy tính bình thường là như thế nào?

Việc xây dựng một sản phẩm TI về các nguy cơ bảo mật là một quá trình theo tình huống, nghĩa là các câu hỏi thường cụ thể và được thiết kế để tạo ra sản phẩm TI riêng lẻ cho một điều tra hiện tại. Các câu hỏi này có thể là:

- Liệu có máy tính có thể gây nguy hại nào từng liên lạc với các máy tính cần bảo vệ trước đó hay không, nếu có thì tới mức nào?
- Liệu có máy tính có thể gây nguy hại nào đăng ký với một ISP đã từng xuất hiện những hoạt động gây nguy hại?
- Nội dung lưu lượng tạo ra từ máy tính gây nguy hại so với hoạt động gắn với các thực thể gây nguy hại đã biết hiện nay như thế nào?

Chúng ta sẽ nghiên cứu kỹ hơn về bản chất của các yêu cầu thu thập tri thức về các tài nguyên cần bảo vệ cùng các nguy cơ bảo mật ở các phần tiếp theo.

## Bước 2: Lập kế hoạch

Cùng với việc xác định yêu cầu, việc lập kế hoạch hợp lý giúp đảm bảo hoàn thành các bước còn lại trong chu trình. Do đó, cần lập kế hoạch và gán các tài nguyên cho từng bước. Trong NSM, điều này có nghĩa là các thành phần khác nhau được dùng cho các bước khác nhau. Ví dụ, trong pha thu thập cần gán chuyên gia bậc ba và quản trị hệ thống (xem chương 1 về phân loại chuyên gia NSM) cho những công việc xử lý các cảm biến và sử dụng công cụ thu thập. Trong pha xử lý và phân tích cần gán chuyên gia cấp một và hai cho các tiến trình này đồng thời phân ra một phần thời gian của họ để làm công tác này.

Tất nhiên về mặt tài nguyên con người và kỹ thuật được gán cho các công việc sẽ thay đổi phụ thuộc vào môi trường và cách xây dựng đội ngũ kỹ thuật. Trong các tổ chức lớn hơn, có thể có một đội riêng để tạo ra các sản phẩm TI. Còn trong các tổ chức nhỏ, có thể chỉ có một người chịu trách nhiệm xây dựng toàn bộ sản phẩm.

### **Bước 3: Thu thập**

Pha thu thập thực hiện việc thu thập tri thức theo các yêu cầu đề ra. Các dữ liệu này cuối cùng sẽ được xử lý, phân tích và truyền đi.

Có thể thấy rằng nhu cầu thu thập cho các mục đích thu thập tri thức sẽ làm thay đổi kế hoạch thu thập thông tin tổng thể. Khi mục tiêu là thu thập tri thức về các tài nguyên cần bảo vệ một cách liên tục, có thể bao gồm tập hợp các thông kê có ích hay tập hợp các dữ liệu về tài sản thời gian thực thụ động như các dữ liệu được tạo ra bởi công cụ PRADS (sẽ đề cập sau). Khi đề cập tới thu thập tri thức về các nguy cơ bảo mật theo tình huống, dữ liệu sẽ thường được thu thập từ các nguồn dữ liệu NSM sẵn có như FPC hay dữ liệu phiên. Những dữ liệu này thường sẽ tập trung vào các tương tác của các thực thể có thể gây hại với các tài nguyên mạng tin cậy. Ngoài ra, các tiến trình thu thập tri thức mã nguồn mở được sử dụng để xác định các thông tin công khai liên quan tới các thực thể có thể gây hại. Bao gồm các thông tin về người đăng ký các địa chỉ IP, hay các thông tin đã biết về một tệp tin nghi ngờ nào đó.

Để thu thập tri thức hiệu quả, tiến trình thu thập đối với một số loại dữ liệu (FPC, PSTR, phiên,...) phải có đủ tài liệu và dễ dàng truy cập.

### **Bước 4: Xử lý**

Sau khi dữ liệu đã được thu thập, một số loại dữ liệu phải được tiếp tục xử lý để trở nên hữu ích cho việc phân tích. Điều này có thể có nghĩa là rất nhiều thứ khác nhau cho nhiều loại dữ liệu khác nhau.

Ở mức độ cao, xử lý là chuyên dữ liệu thu thập được thành một dạng hữu ích hơn. Điều này có thể là áp dụng bộ lọc vào một tệp tin PCAP để thu nhỏ lượng dữ liệu làm việc, hoặc lựa chọn các tệp tin log của một loại dữ liệu nhất định từ một bộ sưu tập các tệp tin log lớn hơn.

Ở mức độ chi tiết hơn, điều này có nghĩa là lấy dữ liệu từ một bên thứ ba hoặc công cụ tùy chỉnh và sử dụng một số lệnh BASH để định dạng đầu ra của những công cụ thành dạng dữ liệu dễ đọc hơn. Trong trường hợp một tổ chức sử dụng một công cụ tùy chỉnh hoặc cơ sở dữ liệu để thu thập thông tin, đó có thể có nghĩa là viết các truy vấn để chèn dữ liệu vào định dạng này, hoặc kéo nó ra khỏi định dạng đó thành dữ liệu dễ đọc hơn.

Cuối cùng, xử lý đôi khi có thể được xem như là một phần mở rộng của tập dữ liệu thu được khi dữ liệu được thu gọn, tinh chỉnh thành một hình thức lý tưởng cho các chuyên gia phân tích.

### **Bước 5: Phân tích**

Phân tích là giai đoạn kiểm tra, xem xét mối liên hệ và đưa vào các ngữ cảnh cần thiết cho các dữ liệu đã thu thập và xử lý, để làm cho chúng có ích. Giai đoạn này giúp cho thông tin thu thập được từ lúc chỉ là những mẫu dữ liệu rời rạc trở thành một sản phẩm hoàn thiện, có ích cho việc ra quyết định.

Trong khi phân tích và tạo ra các sản phẩm TI về nguy cơ bảo mật và các tài nguyên cần bảo vệ, chuyên gia phân tích sẽ lấy dữ liệu đầu ra từ một số công cụ và nguồn dữ liệu, kết hợp các dữ

liệu trên từng máy tính để chỉ ra bức tranh của riêng từng máy tính. Một lượng lớn hơn các tri thức thu thập cho các máy tính cục bộ sẽ làm bức tranh này thêm chi tiết về các xu hướng cũng như các đối tác giao tiếp thông thường của các máy tính. Việc phân tích về các máy tính có thể gây hại sẽ được tạo nên từ các tập dữ liệu nhỏ hơn, yêu cầu việc kết hợp các tri thức nguồn mở vào tiến trình phân tích.

Các kết quả cuối cùng có được từ quá trình này là sản phẩm TI sẵn sàng cho các chuyên gia phân tích.

### Bước 6: Truyền đi

Trong các trường hợp thực tế, một tổ chức sẽ không có đội ngũ chuyên dụng để thu thập tri thức. Điều đó có nghĩa là các chuyên gia phân tích NSM sẽ tạo ra các sản phẩm TI để riêng họ sử dụng. Đây là một lợi thế độc nhất vì người sử dụng các tri thức này thường sẽ là người tạo ra nó, hoặc ít nhất là sẽ ở chung nhóm hay trong cùng tổ chức. Trong pha cuối cùng của chu trình thu thập kiến thức, các sản phẩm tri thức được truyền tới các cá nhân hoặc nhóm đã đưa ra các yêu cầu thu thập kiến thức.

Trong hầu hết các trường hợp, các sản phẩm TI liên tục được đánh giá và cải tiến. Các khía cạnh tích cực và tiêu cực của sản phẩm cuối cùng được đánh giá cẩn thận và các đánh giá này sẽ được dùng tiếp trong việc xác định các yêu cầu thu thập tri thức và lập kế hoạch cho việc tạo sản phẩm. Do đó các tiến trình thu thập tri thức được gọi là chu trình, thay vì là một chuỗi các bước.

#### 4.2.2 Tạo tri thức về tài nguyên cần bảo vệ

Không thể bảo vệ hệ thống mạng một cách hiệu quả nếu không biết có những gì trên nó và các giao tiếp được thực hiện thế nào. Cụ thể hơn, dù cuộc tấn công phức tạp hay thậm chí đơn giản, nếu không biết vai trò của các thiết bị trong mạng, đặc biệt là nơi chứa dữ liệu quan trọng. Sau đó, cũng không thể xác định mức độ nghiêm trọng của các sự cố đã xảy ra khắc phục sự cố hay tiêu diệt những tấn công mạng. Đó là lý do tại sao cần xây dựng các tri thức về các tài nguyên cần bảo vệ.

Trong tài liệu này, tri thức về các tài nguyên cần bảo vệ là một sản phẩm phát triển liên tục, nơi các chuyên gia phân tích có thể tham chiếu để lấy thông tin về các máy tính cần bảo vệ. Thông tin này bao gồm tất cả những thứ mà các chuyên gia phân tích cần cho việc hỗ trợ trong trường hợp thực hiện kiểm tra, và dùng để tham khảo bất cứ khi nào cần. Nói chung, một chuyên gia phân tích sẽ cần tham khảo kho tri thức tài nguyên cần bảo vệ về một máy tính riêng lẻ bất kỳ khi nào họ đang điều tra dữ liệu cảnh báo gắn với máy tính đó. Thông thường một máy tính trong danh sách tài nguyên bảo vệ sẽ là một mục tiêu của các cuộc tấn công. Do đó, hiếm khi chuyên gia phân tích tham chiếu dữ liệu này hàng chục lần mỗi ca làm việc cho hàng loạt các máy tính. Dưới đây là một số cách tạo ra tri thức về tài nguyên cần bảo vệ từ dữ liệu mạng.

##### 4.2.2.1 Lịch sử của tài nguyên mạng và thực trạng

Khi bác sĩ đánh giá một bệnh nhân mới, điều đầu tiên họ thực hiện là đánh giá y sử và tình trạng thể chất của bệnh nhân. Các đánh giá y sử cho bệnh nhân bao gồm các điều kiện y tế hiện

tại và trước đó có thể ảnh hưởng đến sức khỏe hiện tại hay tương lai của bệnh nhân. Nó cũng thường gồm y sử về điều kiện sức khỏe gia đình của bệnh nhân, từ đó có thể xác định và giảm bớt các yếu tố nguy cơ đối với bệnh nhân.

Chuyển khái niệm này vào một tài nguyên mạng, có thể coi y sử của một tài sản mạng là lịch sử kết nối của nó. Điều này liên quan đến việc đánh giá các giao tiếp truyền thông trước đó giữa máy tính trong danh sách trắng và các máy tính khác trên mạng cũng như ngoài mạng. Thậm chí, bên cạnh các máy tính trong các kết nối mạng, cần xem xét cả các dịch vụ tham gia vào kết nối, kể cả client và server. Nếu có thể đánh giá lịch sử kết nối này, có thể đoán được tính hợp lệ của các kết nối mới sinh từ một máy tính trong danh sách trắng trong khi thực hiện điều tra.

Tương tự khi áp dụng khái niệm kiểm tra thể trạng y tế của bệnh nhân vào tài nguyên mạng, có thể tìm được các tiêu chí giúp xác định trạng thái của các tài nguyên mạng. Những tiêu chí này bao gồm địa chỉ IP, tên miền, thuộc VLAN nào, vai trò của thiết bị (máy trạm, máy chủ Web,...), hệ điều hành của thiết bị hoặc vị trí mạng vật lý của nó. Kết quả của đánh giá này là trạng thái hoạt động trên mạng của tài nguyên mạng, có thể được dùng để xác định hoạt động của máy tính trong khi thực hiện một điều tra.

Về phương pháp sử dụng để tạo ra lịch sử hoạt động và thực trạng của tài nguyên mạng, chúng ta sẽ mô tả các công cụ như kiểu Nmap để xác định cách kiểm tra thực trạng thông qua một mô hình tài sản, cũng như sử dụng PRADS để làm việc với phần lịch sử hoạt động thông qua thu thập dữ liệu tài nguyên thụ động theo thời gian thực.

#### 4.2.2.2 Xác định mô hình tài nguyên mạng

Mô hình tài sản mạng đơn giản là một danh sách các máy tính trong mạng và các thông tin quan trọng gắn với chúng, bao gồm các thông tin như địa chỉ IP, tên miền, vai trò (server, máy trạm, bộ định tuyến,...), dịch vụ cung cấp (web, SSH, proxy, ...) và hệ điều hành. Đây là hình thức cơ bản nhất của tri thức về tài nguyên cần bảo vệ.

Dễ thấy là có một số cách để xây dựng ra mô hình này. Hầu hết các tổ chức sẽ dùng một số hình thức kiểu phần mềm quản lý tài sản doanh nghiệp để lấy cung cấp dữ liệu này. Nếu không theo hình thức này, có thể tự tạo dữ liệu theo một cách thức nào đó.

Trong thực tế, dữ liệu tài sản hiếm khi chính xác 100%. Nhưng cần cố gắng hết sức để thu thập dữ liệu một cách chính xác nhất. Một cách tạo các dữ liệu loại này là thực hiện việc quét cổng nội bộ, với sự trợ giúp của các phần mềm như Nmap. Ví dụ, có thể chạy câu lệnh quét cổng cơ bản kiểu SYN scan:

```
nmap -sn 172.16.16.0/24
```

Câu lệnh này thực hiện quét ICMP đơn giản (ping) tới các máy tính trong mạng 172.16.16.0/24 và tạo ra kết quả như Hình 4.35.

```

Starting Nmap 6.25 ( http://nmap.org ) at 2013-09-30 15:25 EDT
Nmap scan report for 172.16.16.1
Host is up (0.0050s latency).
Nmap scan report for 172.16.16.2
Host is up (0.075s latency).
Nmap scan report for 172.16.16.3
Host is up (0.068s latency).
Nmap scan report for hercules (172.16.16.5)
Host is up (0.075s latency).
Nmap scan report for righthawk (172.16.16.10)
Host is up (0.036s latency).
Nmap scan report for 172.16.16.128
Host is up (0.00021s latency).
Nmap scan report for 172.16.16.132
Host is up (0.0037s latency).
Nmap scan report for 172.16.16.137
Host is up (0.012s latency).
Nmap scan report for 172.16.16.139
Host is up (0.0027s latency).
Nmap done: 256 IP addresses (9 hosts up) scanned in 3.77 seconds

```

**Hình 4.35 Kết quả quét với Nmap**

Dữ liệu đầu ra bao gồm danh sách địa chỉ IP của các máy tính (tất nhiên là giả thiết các máy tính đều đáp ứng với lệnh ping, nghĩa là không bị tường lửa chặn).

Với việc sử dụng các tính năng khác của Nmap, có thể lấy thêm được các dữ liệu khác về máy tính tương ứng trong mạng như các cổng và tương ứng là các dịch vụ đang chạy trên các máy tính, phiên bản và hệ điều hành chạy trên máy tính,...

Tuy nhiên, việc thu thập dữ liệu theo cách này cũng không hoàn toàn tin cậy do các máy tính có thể hoạt động vào thời điểm này mà không hoạt động vào thời điểm khác, hay việc chặn các thông tin do các chính sách của tường lửa,... Do đó, cần kết hợp với các nguồn dữ liệu khác để kiểm chứng các kết quả như các bản ghi giao dịch DNS, dữ liệu phiên sử dụng SiLK. Một lựa chọn khác là công cụ thụ động PRADS.

#### 4.2.2.3 PRADS

PRADS (Passive Real-time Asset Detection System) là một công cụ được xây dựng để nghe dữ liệu mạng, thu thập thông tin về các máy tính và dịch vụ, và có thể dùng để xây dựng bản đồ mạng. Công cụ này dựa trên hai công cụ rất mạnh khác là PADS (Passive Asset Detection System) và P0f (công cụ nhận biết hệ điều hành thụ động). PRADS kết hợp tính năng của cả hai công cụ đó nhằm xây dựng dịch vụ thu thập tri thức về các tài sản cần bảo vệ.

PRADS có trong Security Onion, do đó có thể kiểm tra dữ liệu này bằng cách truy vấn trong Sguil, một công cụ xem các cảnh báo từ các bộ phát hiện xâm nhập và dữ liệu từ công cụ NSM khác. Các mục PRADS có thể thấy trong cột Event Message (dù rằng có thể để tên là PADS). Hình 4.36 là một ví dụ về các mục bản ghi PRADS.

Có nhiều loại sự kiện trong dữ liệu PRADS được tạo ra. Để hiểu được các sự kiện này, xét một ví dụ bản ghi PRADS. Trong Security Onion, PRADS mặc định được kích hoạt với câu lệnh như sau:

```

prads -i eth1 -c /etc/nsm/< sensor-name >/prads.conf -u sguil -g sguil -L
/nsm/sensor_data/< sensor-name >/sancp/ -f /nsm/sensor_data/< sensor-name >/pads fifo -
b ip or (vlan and ip)

```

Realtime Events   Escalated Events										
ST	CNT	Sensor	Alert ID	Date/Time	Src IP	Sport	Dst IP	DPort	Pr	Event Message
RT	1	osprey-eth1	4.55	2013-10-01 13:12:21	172.16.16.132	63047	172.16.16.1	1780	6	PADS Changed Asset - http Microsoft (Windows/6.1 UPnP/1.0)
RT	1	osprey-eth1	4.56	2013-10-01 13:22:00	172.16.16.132	63071	184.106.31.93	80	6	PADS Changed Asset - http Microsoft Office/14.0 (Windows ...)
RT	1	osprey-eth1	4.14	2013-09-30 22:59:52	172.16.16.128	59433	172.16.16.1	1780	6	PADS Changed Asset - http Mozilla/4.0 (compatible; UPnP/1.0)
RT	1	osprey-eth1	4.47	2013-10-01 12:57:14	172.16.16.25	53991	108.168.255.243	80	6	PADS Changed Asset - http Ruby
RT	1	osprey-eth1	4.52	2013-10-01 13:03:43	172.16.16.132	63026	184.106.31.88	443	6	PADS Changed Asset - ssl TLS 1.0 Client Hello
RT	1	osprey-eth1	4.28	2013-10-01 02:26:57	172.16.16.132	53685	176.32.100.68	80	6	PADS New Asset - http Dalvik/1.4.0 (Linux; U; Android 2.3.4; ...)
RT	1	osprey-eth1	4.33	2013-10-01 06:49:53	172.16.16.21	55819	91.189.95.83	80	6	PADS New Asset - http Debian APT (HTTP/1.3 (0.8.16-exp12u...))
RT	1	osprey-eth1	4.34	2013-10-01 06:54:08	172.16.16.20	52587	91.189.91.15	80	6	PADS New Asset - http Debian APT (HTTP/1.3 (0.8.16-exp12u...))
RT	1	osprey-eth1	4.36	2013-10-01 07:53:50	172.16.16.10	60787	91.189.91.13	80	6	PADS New Asset - http Debian APT (HTTP/1.3 (0.8.16-exp12u...))
RT	1	osprey-eth1	4.37	2013-10-01 08:22:00	172.16.16.139	49838	67.148.153.243	80	6	PADS New Asset - http Hopper_NCMV/0.1

**Hình 4.36** *Dữ liệu PRADS trong Sguil*

Chi tiết về các tùy chọn như sau:

- -b < filter >: Nghe từ mạng dựa trên BPF.
  - -c < config file >: Tệp cấu hình PRADS.
  - -D: Chạy thường trú.
  - -f < file >: Ghi log vào tệp kiểu FIFO (first in, first out).
  - -g < group >: Nhóm mà PRADS sẽ chạy.
  - -i < interface >: Giao diện mạng sẽ nghe gói tin.
  - -L < directory >: Log ghi vào trong thư mục này.
  - -l < file >: Log ghi vào tệp tin văn bản này.
  - -r < file >: Đọc từ tệp tin PCAP.
  - -u < username >: tên người dùng hệ thống mà PRADS sẽ sử dụng khi chạy.
  - -v: Tăng mức chi tiết thông tin đầu ra của PRADS.

Dữ liệu PRADS được lưu theo kiểu định dạng cơ sở dữ liệu, dễ dàng tạo truy vấn cho các thông tin này nhưng lại không dễ xem với dữ liệu thô. Tuy nhiên, có thể lưu dữ liệu vào tệp tin văn bản (text) trong /var/log/prads-assets.log, ví dụ như trong Hình 4.37.

```
sander@osprey:/msa/sensor_data/osprey-eth1$ head -15 prods.log
asset,vlan,port,proto,service,[service-info],distance,discovered
172.16.16.128,0,63538,6,SWN,[65535:64:1:64:1460,N,W,N,N,T,S,E,E,P:unknown:unknown:[link:ether:net/socdes:uptime:2342hrs],0,1388640852
172.16.16.132,0,443,6,CLIENT,[unknown:https],0,1388640853
172.16.16.140,0,443,6,CLIENT,[unknown:https],0,1388640855
172.16.16.130,0,443,6,CLIENT,[unknown:https],0,1388640859
172.16.16.148,0,53,17,CLIENT,[unknown:ddosin],0,1388640859
172.16.16.140,0,58903,6,RST,[0:64:11:80:::ffff:fe80:4148]:[Linux? (dropped)],0,1388640859
172.16.16.129,0,63538,6,SWN,[65535:64:1:64:1460,S,E,E,P:MacOS:iPhone OS 3.1.3 (UC):[link:ether:net/socdes],0,1388640859
172.16.16.128,0,53,17,CLIENT,[unknown:ddosin],0,1388640859
172.16.16.28,0,53,17,CLIENT,[unknown:ddosin],0,1388640860
172.16.16.21,0,53,17,CLIENT,[unknown:ddosin],0,1388640860
172.16.16.128,0,443,6,CLIENT,[unknown:https],0,1388640866
172.16.16.130,0,58903,6,FIN,[0:92:64:11:52:N,N,T,ATN:unknown:unknown:uptime:955hrs],0,1388640866
172.16.16.138,0,58904,6,SWN,[65535:64:1:64:1468,N,W,N,N,T,S,E,E,P:unknown:unknown:[link:ether:net/socdes:uptime:955hrs],0,1388640866
172.16.16.128,0,443,6,CLIENT,[ssl:SSL 2.0 Client Hello],0,1388640866
```

**Hình 4.37 Tệp tin log của PRADS**

Dòng đầu tiên xác định định dạng cho các mục bản ghi, bao gồm: asset, vlan, port, proto, service, [service-info], distance, discovered.

Chi tiết về các trường như sau:

- Asset: Các địa chỉ IP của các tài sản trong biến home\_nets được phát hiện
- VLAN: Các tag VLAN của tài sản
- Port: Số hiệu cổng của các dịch vụ được phát hiện
- Proto: Các số hiệu giao thức của dịch vụ được phát hiện
- Service: Các dịch vụ PRADS đã xác định là đang sử dụng.
- [Service-info]: Các dấu vết phù hợp với các dịch vụ xác định, cùng với đầu ra của nó.
- Distance: Khoảng cách đến các tài sản dựa trên một giá trị thời gian sống ban đầu đã dự đoán.
- Discovered: Các dấu thời gian Unix (Unix timestamp) khi các dữ liệu được thu thập.

Dựa trên dữ liệu này, có thể thấy rằng PRADS tự nó không thực sự tạo ra các thông tin đã thấy trong Sguil như đây có phải là một tài sản mới hoặc tài sản đã bị thay đổi. PRADS chỉ đơn giản là ghi lại những dữ liệu mà nó quan sát chứ không có bất kỳ xử lý gì thêm. Điều này có nghĩa là các cảnh báo tài sản mới hay tài sản bị thay đổi như đã thấy trong Sguil thực sự là được tạo ra bởi chính Sguil, dựa trên dữ liệu PRADS chứ không phải được tạo ra bởi PRADS.

Có một vài cách sử dụng PRADS để có được tri thức về các tài nguyên cần bảo vệ. Cách thứ nhất là sử dụng Sguil và các thông báo về tài sản mới và tài sản đã bị thay đổi, ví dụ Hình 4.38.

The screenshot shows the Sguil interface with the title "SGUIL-0.8.0 - Connected To localhost". The top bar includes "File", "Query", "Reports", "Sound", "Off", "ServerName: localhost", "UserName: cander", "UserID: 2", and the date "2013-10-01 19:39:06 GMT". Below the title, there are tabs for "Realtime Events", "Escalated Events", and "Event Query 1". The "Event Query 1" tab is active, displaying the following SQL query in its content area:

```
(SELECT event.status, event.priority, sensor.hostname, event.timestamp AS datetime, event.sid, event.cid, event.signature,INET_NTOA(event.src_ip),
INET_NTODcevent.dst_ip), event.ip_proto, event.src_port, event.dst_port, event.signature_gen, event.signature_id, event.signature_rev FROM event JOIN INDEX
(event_p_key, sid_time) INNER JOIN sensor ON event.sid=sensor.sid WHERE event.timestamp > '2013-09-24' AND event.src_ip = INET_ATON('172.16.16.145')) UNION (
```

The main table displays a list of events with the following columns: ST, CNT, Sensor, Alert ID, Date/Time, Src IP, SPort, Dst IP, DPort, Pr, and Event Message. The data shows various alerts from the "osprey-et" sensor, mostly related to traffic from the IP 172.16.16.145, including HTTP and HTTPS connections to various destinations like apple.com, Google Talk, and Mozilla Firefox.

ST	CNT	Sensor	Alert ID	Date/Time	Src IP	SPort	Dst IP	DPort	Pr	Event Message
NA	1	osprey-et...	6.8166	2013-10-01 19:18:02	172.16.16.145	49167	23.62.111.152	80	6	URL init-p01md.apple.com
RT	1	osprey-et...	4.66	2013-10-01 19:18:02	172.16.16.145	49167	23.62.111.152	80	6	PADS New Asset - http server (bag [Mac OS X,10.8.5,12F...)
RT	1	osprey-et...	4.67	2013-10-01 19:18:03	172.16.16.145	49168	17.149.32.33	443	6	PADS New Asset - ssl TLS 1.0 Client Hello
NA	1	osprey-et...	6.8167	2013-10-01 19:18:03	172.16.16.145	49170	23.62.111.99	80	6	URL init-est.apple.com
RT	1	osprey-et...	4.68	2013-10-01 19:18:04	172.16.16.145	49171	17.149.34.62	5223	6	PADS New Asset - ssl TLS 1.0 Client Hello
NA	1	osprey-et...	6.8168	2013-10-01 19:18:04	172.16.16.145	49172	23.47.21.163	80	6	URL SVRSecure-G3-ala.verisign.com
RT	1	osprey-et...	4.69	2013-10-01 19:18:06	172.16.16.145	123	17.151.16.38	123	17	PADS New Asset - unknown @ntp
RT	1	osprey-et...	3.997	2013-10-01 19:23:21	172.16.16.145	49328	74.125.139.125	5222	6	ET CHAT Google IM traffic jabber client sign-on
RT	1	osprey-et...	3.998	2013-10-01 19:23:21	172.16.16.145	49328	74.125.139.125	5222	6	GPL CHAT MISC jabber/Google Talk Outgoing Traffic
RT	1	osprey-et...	3.996	2013-10-01 19:23:21	172.16.16.145	49328	74.125.139.125	5222	6	GPL CHAT Google Talk Legion
NA	1	osprey-et...	6.8175	2013-10-01 19:23:32	172.16.16.145	49340	172.16.16.1	1780	6	URL 172.16.16.1
RT	1	osprey-et...	4.71	2013-10-01 19:23:32	172.16.16.145	49340	172.16.16.1	1780	6	PADS Changed Asset - http Mozilla/4.0 (compatible; UP...
NA	1	osprey-et...	6.8176	2013-10-01 19:23:32	172.16.16.145	49341	184.28.140.224	80	6	URL configuration.apple.com
NA	1	osprey-et...	6.8177	2013-10-01 19:23:33	172.16.16.145	49343	184.28.140.224	80	6	URL configuration.apple.com
NA	1	osprey-et...	6.R178	2013-10-01 19:23:33	172.16.16.145	49344	172.16.16.1	1780	6	URL 172.16.16.1

Hình 4.38 Truy vấn Sguil về một máy tính

Trong Hình 4.38, câu truy vấn được tạo cho tất cả các sự kiện liên quan tới một cảnh báo. Ở đây, lưu ý một sự kiện gắn với máy tính có địa chỉ IP 172.16.16.145, bao gồm cảnh báo Snort, URL đã thăm và các cảnh báo PRADS. Trong số các cảnh báo PRADS đã thấy, có 4 cảnh báo tài sản mới, chỉ ra là máy tính lần đầu tiên kết nối tới mỗi địa chỉ IP có trong cảnh báo:

- Alert ID 4.66: HTTP Connection to 23.62.111.152
- Alert ID 4.67: HTTPS Connection to 17.149.32.33
- Alert ID 4.68: HTTPS Connection to 17.149.34.62

- Alert ID 4.69: NTP Connection to 17.151.16.38

Khi xem xét sự kiện này, chúng ta thấy được rằng, ngũ cảnh có ích sẽ giúp xác định được ngay là liệu một thiết bị tin cậy đã từng kết nối tới một thiết bị từ xa cụ thể nào chưa. Trong trường hợp thấy các lưu lượng đáng nghi đi tới một địa chỉ chưa biết, sự thực là thiết bị tin cậy của tổ chức chưa bao giờ liên lạc với địa chỉ này trước đó, thì có thể là một cảnh báo về một sự kiện đáng nghi ngờ đang diễn ra, do đó cần xem xét kỹ lưỡng.

Hình 4.38 cũng hiển thị một cảnh báo thay đổi tài sản, chỉ ra rằng việc sử dụng một HTTP client user agent string mới.

#### Alert ID 4.71: Mozilla/4.0 (compatible; UPnP/1.0; Windows NT/5.1)

Ngũ cảnh này cho thấy máy tính tin cậy đang thực hiện một điều gì đó chưa bao giờ làm trước đó. Dù điều này thể hiện người dùng đã tải một trình duyệt mới, nó cũng là một cảnh báo về hành vi có hại. Nên chú ý thêm về các thiết bị bắt đầu đưa ra các dịch vụ mới, đặc biệt là khi các thiết bị đó là máy trạm của người dùng và không được hoạt động như kiểu một server.

```

6 -----
IP: 172.16.16.133
OS: unknown unknown (8%) 1

Port Service    TCP-Application
80  CLIENT      server (bag [iPhone OS,6.1.4,188350,iPhone5,1])
443 CLIENT     @https
443 CLIENT     TLS 1.0 Client Hello
443 CLIENT     @https
993 CLIENT     TLS 1.0 Client Hello

Port Service    UDP-Application
53  CLIENT      @domain

7 -----
IP: 172.16.16.135
OS: Linux 2.6 (newer, 5) (80%) 1

Port Service    TCP-Application
80  CLIENT      Dalvik/1.4.0 (Linux; U; Android 2.3.4; Kindle Fire Build/GINGERBREAD)
443 CLIENT     TLS 1.0 Client Hello

Port Service    UDP-Application
53  CLIENT      @domain

8 -----
IP: 172.16.16.137
OS: unknown unknown (8%) 2

Port Service    TCP-Application
443 CLIENT     @https

Port Service    UDP-Application
53  CLIENT      @domain

```

**Hình 4.39 Dữ liệu báo cáo tài sản PRADS**

Một cách khác để tận dụng dữ liệu PRADS là sử dụng nó để xác định mô hình tài nguyên cơ sở. Do PRADS lưu tất cả các thông tin tài sản đã thu thập được cho các tài sản xác định trong biến home\_nets, dữ liệu này có thể được phân tích để chỉ ra các thông tin đã thu thập trên từng máy tính. Điều này được hoàn thành bằng cách sử dụng một Perl script có trong PRADS gọi là prads-asset-report. Script này sẽ lấy kết quả đầu ra từ một file log tài sản PRADS, và xuất ra một danh sách các thông tin biết về mỗi địa chỉ IP. Nếu đang sử dụng PRADS để log dữ liệu vào /var/log/prads-asset.log, thì có thể chạy dòng lệnh prads-asset-report để tạo dữ liệu này. Ngược

lại, có thể chỉ ra vị trí dữ liệu tài sản PRADS sử dụng tham số `-r <file>`. Ví dụ về dữ liệu này được thấy trong Hình 4.39.

Trong dữ liệu đầu ra có thể thấy được thông tin về hệ điều hành của từng thiết bị. Thông tin này sẽ càng chính xác nếu PRADS có được dữ liệu trao đổi của các thiết bị trên mạng nhiều hơn nữa.

Như vậy có thể thấy PRADS là một công cụ mạnh và đơn giản để tạo ra được mô hình tài sản từ PRADS. Có thể tham khảo thêm về công cụ này tại địa chỉ <http://gamelinux.github.io/prads/>.

#### 4.2.3 Tạo tri thức về nguy cơ bảo mật

Tri thức về các nguy cơ bảo mật (TI) là một phần của tập các tri thức như đã xác định trước đó trong phần đầu của chương. TI chỉ tập trung vào các bộ phận có thể gây hại, và tìm cách thu thập dữ liệu để hỗ trợ việc tạo ra một sản phẩm có thể được sử dụng để đưa ra quyết định về bản chất của các nguy cơ. Loại tri thức này có thể được chia thành ba loại con: TI chiến lược, TI khai thác và TI chiến thuật (Hình 4.40).



**Hình 4.40 Các loại tri thức về các nguy cơ bảo mật**

**TI chiến lược** là thông tin liên quan đến các chiến lược, chính sách, kế hoạch của kẻ tấn công ở mức cao. Thông thường, việc thu thập và phân tích thông tin ở cấp độ này chỉ xảy ra bởi chính phủ hoặc các tổ chức quân sự để đáp ứng với các nguy cơ từ các chính phủ hoặc quân đội khác. Các tổ chức lớn đang phát triển những tính năng này, và một số các tổ chức này hiện tại có bán dịch vụ về TI chiến lược. Các dịch vụ này tập trung vào những mục tiêu dài hạn của các lực lượng hỗ trợ tấn công hoặc cá nhân. Sản phẩm của loại TI này có thể bao gồm các tài liệu chính sách, thuyết chiến tranh, báo cáo vị thế, chính phủ, quân đội, hoặc mục tiêu nhóm.

**TI khai thác** là thông tin liên quan đến cách một kẻ tấn công hoặc nhóm những kẻ tấn công lập kế hoạch và hỗ trợ các hoạt động nhằm hỗ trợ cho các mục tiêu chiến lược. Điều này khác với TI chiến lược vì nó tập trung vào mục tiêu hẹp hơn, thường giới hạn cho các mục tiêu ngắn hạn chỉ là một phần của bức tranh lớn. TI khai thác thường dùng nhiều trong phạm vi chính phủ hoặc các tổ chức quân sự. Tuy nhiên các tổ chức riêng rẽ cũng thường trở thành nạn nhân của kẻ tấn công khi chúng thực hiện các hành vi nhằm thỏa mãn mục tiêu khai thác. Do vậy, một số tổ chức công cộng sẽ để ý tới các cuộc tấn công này, với khả năng tạo ra TI khai thác.

**TI chiến thuật** đề cập tới các thông tin liên quan đến các hành động cụ thể thực hiện trong khi tiến hành các hoạt động ở cấp độ nhiệm vụ. Đây là nơi chúng ta đi sâu vào các công cụ, chiến thuật và thủ tục được sử dụng bởi kẻ tấn công, cũng là nơi các doanh nghiệp thực hiện NSM sẽ tập trung nỗ lực của họ vào. Nó thường gồm các chỉ báo tấn công (địa chỉ IP, tên file, chuỗi văn bản) hay danh sách các công cụ tấn công cụ thể. Loại thông tin này thường là tạm thời và nhanh chóng bị lỗi thời.

#### 4.2.3.1 Nghiên cứu về các máy trạm không tin cậy

Khi một cảnh báo được tạo ra đối với một kết nối đáng ngờ giữa một máy tính tin cậy và một máy tính không tin cậy, việc cần làm đối với một chuyên gia phân tích là tạo ra TI chiến thuật liên quan tới máy tính không tin cậy. Sau đó, hầu hết các cảnh báo IDS sẽ cho biết địa chỉ IP của máy tính và mẫu trao đổi thông tin gây ra cảnh báo. Trong phần này sẽ xem xét các thông tin có thể có được khi chỉ có duy nhất địa chỉ IP của máy tính hoặc tên miền.

##### a. Các nguồn dữ liệu nội bộ

Cách nhanh nhất để có được thông tin về các máy tính không tin cậy và ở bên ngoài là kiểm tra dữ các nguồn dữ liệu nội bộ hiện có. Có thể thu thập được các thông tin này bằng các câu hỏi:

1. Máy tính không tin cậy có bao giờ liên lạc với máy tính tin cậy trước đó?
2. Bản chất kết nối của máy tính này với các máy tính tin cậy là gì?
3. Máy tính không tin cậy có bao giờ liên lạc với máy tính tin cậy khác trên mạng?

Các câu trả lời cho những câu hỏi này có thể nằm trong phạm vi của các nguồn dữ liệu khác nhau.

Câu hỏi 1 dễ dàng trả lời được khi có tri thức về các máy tính cần bảo vệ phù hợp, ví dụ như dữ liệu PRADS.

Câu hỏi 2 chỉ có thể trả lời được khi có nguồn dữ liệu với mức chi tiết cao. Dữ liệu phiên chỉ có thể chỉ ra một vài thông số kết nối cơ bản và các cổng đang sử dụng mà không có thông tin chi tiết hơn về những gì chính xác đang diễn ra. Trong một số trường hợp, các công cụ phát hiện xâm nhập có thể có các thông tin này. Snort và Suricata thông thường sẽ cung cấp gói tin tấn công với chữ ký của chúng, và công cụ như Bro cung cấp các dữ liệu thêm khi được cấu hình. Trong trường hợp khác, có thể phân tích dữ liệu FPC hay PSTR để tìm câu trả lời.

Để trả lời câu hỏi 3, cần bắt đầu với dữ liệu phiên để lấy thông tin về các bản ghi về kết nối giữa các máy tính. FPC và PSTR có thể chứa các dữ liệu này. Nếu không có thì có thể xem xét trong dữ liệu PRADS.

Kết hợp các trả lời của 3 câu hỏi trên sẽ giúp xây dựng được TI xung quanh các hành vi của máy tính không tin cậy trên hệ thống mạng.

### b. Tri thức mã nguồn mở

Đây là loại tri thức thu thập từ các nguồn công khai. Trong NSM, thông thường loại thông tin này lấy từ các trang web mở. Nó giúp lấy được thông tin mà không phải gửi gói tin trực tiếp tới các máy tính không tin cậy. Các trang web có thể được dùng để thực hiện thu thập tri thức mà nguồn mở (OSINT) liên quan tới địa chỉ IP, tên miền và các tệp tin nguy hại. Danh sách các trang web này có thể tham khảo tại <http://www.appliednsm.com/osint-resources>.

Mặc dù có nhiều trang web hỗ trợ tìm kiếm thông tin về địa chỉ IP và thông tin đăng ký tên miền, chúng ta có thể sử dụng trang web tổng hợp như Robtex (<http://www.robtex.com>) với rất nhiều thông tin hữu ích. Hình 4.41 minh họa một tìm kiếm cho [espn.com](http://espn.com) và hiển thị với các Tab bản ghi của Robtex.

Result Summary Records Graph Shared Whois Blacklists Analysis Contact						
espn.com		Lucky	Search	Google Custom Search		
Host name	<input checked="" type="checkbox"/>					
AS name	<input checked="" type="checkbox"/>					
IP location	<input checked="" type="checkbox"/>					
Information age	<input checked="" type="checkbox"/>					
Base	Record Type	Name	IP-number	Reverse	Route	Autonomous System
<a href="#">espn.com</a>	A	<a href="#">192.161.132.252</a> DISNEY.COM, Seattle, WA, United States	192.161.132.252	T	192.161.132.0/24	AS65337 DISNEY AS65337 Disney Online
	AAAA	<a href="#">192.234.2.111</a> ESPN, Bristol, CT, United States	192.234.2.111		192.234.2.0/24	AS65338 ESPN
	AAAA	<a href="#">192.234.2.112</a> ESPN, Bristol, CT, United States	192.234.2.112		192.234.2.0/24	AS65338 ESPN Inc.
	AAAA	<a href="#">192.234.2.113</a> 12 days old	192.234.2.113		192.234.2.0/24	AS65332 SIMPSON (Switzerland) Ltd
	AAAA	<a href="#">192.234.2.114</a> ESPN, Los Angeles, CA, United States	192.234.2.114		192.234.2.0/24	AS65332 Route update via web
<a href="#">espn.com</a>	AAAA	<a href="#">208.61.68.230</a> ESPN-NSA-NJ, Newark, NJ, United States	208.61.68.230		208.61.68.0/24	AS65338 Bat Blue IPv4 Network 208.61.68.0/24
	AAAA	<a href="#">74.123.202.183</a> 37 days old	74.123.202.183		74.123.202.0/24	AS65338 Bat Blue IPv4 Network Block
	AAAA	<a href="#">208.61.68.231</a> 11 days old	208.61.68.231		208.61.68.0/24	AS65338 Bat Blue IPv4 Network 208.61.68.0/24
	AAAA	<a href="#">74.123.202.26</a> 25 days old	74.123.202.26		74.123.202.0/24	AS65338 Bat Blue IPv4 Network Block
	AAAA	<a href="#">74.123.202.37</a> 25 days old	74.123.202.37		74.123.202.0/24	AS65338 Bat Blue IPv4 Network Block
<a href="#">espn.market1.batblue.net</a>	AAAA	<a href="#">208.61.68.14</a> 18 days old	208.61.68.14		208.61.68.0/24	AS65338 Bat Blue IPv4 Network 208.61.68.0/24
	AAAA	<a href="#">208.61.68.15</a> 18 days old	208.61.68.15		208.61.68.0/24	AS65338 Bat Blue IPv4 Network 208.61.68.0/24
	AAAA	<a href="#">74.123.202.12</a> 10 days old	74.123.202.12		74.123.202.0/24	AS65338 Bat Blue IPv4 Network Block
	AAAA	<a href="#">74.123.202.13</a> 10 days old	74.123.202.13		74.123.202.0/24	AS65338 Bat Blue IPv4 Network Block
	AAAA	<a href="#">74.123.202.14</a> 10 days old	74.123.202.14		74.123.202.0/24	AS65338 Bat Blue IPv4 Network Block

**Hình 4.41 Tab bản ghi trong Robtex**

Thông tin khác nữa có thể thu thập là danh tiếng của các địa chỉ IP và tên miền. Lý do là nếu trong quá khứ một địa chỉ IP hay tên miền có gắn với những hoạt động gây hại thì rất có thể hiện tại vẫn như vậy. Công ty NoVirusThanks có 2 dịch vụ miễn phí là trang web IPVoid (<http://www.ipvoid.com/>) và URLVoid (<http://www.urlvoid.com/>) kết hợp nhiều danh sách danh tiếng khác nhau để cho kết quả về một địa chỉ IP hay tên miền đang xem xét có trong danh sách đó không. Hình 4.42 và 4.43 là hai ví dụ với hai dịch vụ này.

#### Domain Information

Analyzed On	2013-10-03 15:29 GMT
Website Address	.eu
Blacklist Status	BLACKLISTED
Detection Ratio	3 / 28 (11 %)
Domain 1st Registered	Unknown
Google Page Rank	PAGE RANK 4.3
Alexa Rank	Unknown

#### Website Blacklist Report

Engine	Status	Info
SCUMWARE	DETECTED	
BitDefender	DETECTED	
GoogleSafeBrowsing	DETECTED	
SpamhausDBL	NOT FOUND	
MyWOT	NOT FOUND	
MalwareDomainList	NOT FOUND	

**Hình 4.42 URLVoid**

#### IP Blacklist Report

Engine	Status	Info
TornevallNET	DETECTED	
Spamhaus	DETECTED	
DNSBL_AbuseCH	DETECTED	
SpamCop	DETECTED	
PSBL	DETECTED	
WPBL	DETECTED	
ProjectHoneypot	DETECTED	
CBL_AbuseAt	DETECTED	
SORBS	DETECTED	
NIX_Spam	DETECTED	
Swinog_DNSRBL	DETECTED	
BlockList_de	NOT FOUND	
MyWOT	NOT FOUND	

**Hình 4.43 IPVoid**

#### 4.2.3.2 Nghiên cứu về các tệp tin không tin cậy

Các tri thức cần thu thập về tệp tin có thể sử dụng để xây dựng TI chiến thuật về các nguy cơ đang điều tra.

Có một số biện pháp để trích xuất các tệp tin đáng nghi trong thời gian thực như sử dụng Bro, còn nếu đang truy cập tới toàn bộ dữ liệu đang xem xét thì có thể sử dụng Wireshark.

Giống như thu thập thông tin về máy tính, có những trang web và dịch vụ giúp phân tích các tập tin này, ví dụ như các trang web phân tích mã độc trực tuyến. Chẳng hạn Virustotal với 49 loại virus engine. Nếu muốn có một môi trường để có thể tự mình kiểm soát thì có thể sử dụng Cuckoo sandbox hay Malwr sandbox (<http://www.malwr.com>). Trang web này sử dụng Cuckoo để thực hiện phân tích mã độc. Hình 4.44 và 4.45 thể hiện các báo cáo kết quả phân tích.

The screenshot shows the 'Signatures' section of a VirusTotal report. It lists several malicious behaviors:

- Starts servers listening on 0.0.0.0:0
- File has been identified by at least one AntiVirus on VirusTotal as malicious.
- Performs some HTTP requests
- Retrieves Windows ProductID, probably to fingerprint the sandbox
- Collects information to fingerprint the system (MachineGuid, DigitalProductId, SystemBiosDate)



**Hình 4.44 Báo cáo hiển thị các chữ ký tìm thấy và Screenshot**

• mypcbackup0529.exe 1088	BackupSetup.exe 376	dotnetfx.exe 180	setup.exe 1928	clwireg.exe 1244	
<p>mpcbackup0529.exe    BackupSetup.exe    dotnetfx.exe    setup.exe    clwireg.exe</p> <p>mpcbackup0529.exe, PID: 1088, Parent PID: 1824</p> <p style="text-align: center;">1 2 3 ... 80</p>					
<p style="text-align: center;"><a href="#">network</a> <a href="#">filesystem</a> <a href="#">registry</a> <a href="#">process</a> <a href="#">services</a> <a href="#">synchronization</a></p>					
TIME	API	ARGUMENTS	STATUS	RETURN	REPEATED
2013-10-03 04:54:58,143	LdrGetDllHandle	FileName: C:\WINDOWS\system 32\rpcess.dll ModuleHandle: 0x00000000	failed	0xc0000135	1 time
2013-10-03 04:54:58,143	DeviceIoControl	DeviceHandle: 0x00000048 IoControlCode: 3735560 InBuffer: E\x10p\xeb- -X2\xcb\xaa\x971\xbe\x16g\xfd,YA6L	success	0x00000001	

**Hình 4.45 Báo cáo mô tả các kết quả phân tích hành vi**

Cuckoo sandbox tuy khó thiết lập và làm quen nhưng rất hữu ích và thuận tiện so với các dịch vụ trực tuyến. Khi đã quen với Cuckoo, có thể thấy đây là một sandbox phân tích mã độc giàu tính năng và thuận tiện trong nhiều tình huống.

### 4.3. QUY TRÌNH PHÂN TÍCH

Trong NSM, nói chung mỗi quy trình phân tích gồm 3 phần: đầu vào, điều tra và đầu ra. Phân tích là một cách tiếp cận có hệ thống để xác định khi một sự cố xảy ra. Các đầu vào thường là một số loại cảnh báo IDS hoặc một sự cố bất thường, và đầu ra là các quyết định về một sự cố đã xảy ra. Sau đây sẽ xem xét một số phương pháp phân tích.

#### 4.3.1 Các phương pháp phân tích

##### 4.3.1.1 Điều tra quan hệ

Thuật ngữ “điều tra” ở đây có thể liên tưởng tới một cuộc điều tra của cảnh sát. Đó là vì quá trình điều tra một vi phạm an ninh thông tin và điều tra tội phạm khá giống nhau. Trong thực tế, cách tiếp cận mà các nhà điều tra bên cảnh sát thường sử dụng để tìm hiểu được cẩn kẽ về tội phạm có thể được dùng để làm khung cho một phương pháp phân tích. Điều này được gọi là điều tra quan hệ.

Các phương pháp quan hệ được dựa trên việc xác định các mối quan hệ tuyến tính giữa các thực thể. Đây là loại điều tra dựa trên các mối quan hệ tồn tại giữa các đầu mối và các cá nhân liên quan đến tội phạm. Một mạng lưới các máy tính có thể liên tưởng tới một mạng lưới của con người. Tất cả mọi thứ được kết nối, và mỗi hành động được thực hiện có thể dẫn đến hành động khác xảy ra.

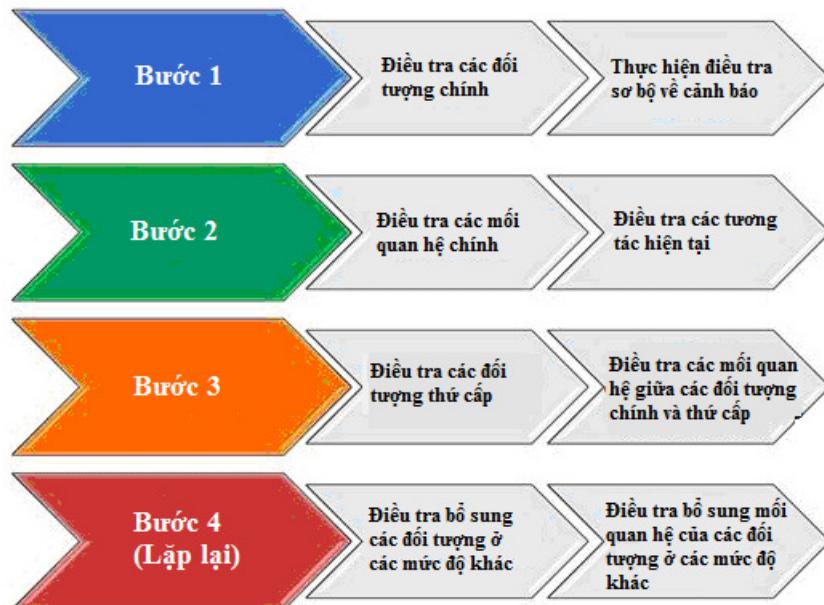
Điều này có nghĩa rằng nếu các chuyên gia phân tích có thể xác định đủ các mối quan hệ giữa các thực thể, họ có thể tạo ra một trang web cho phép xem hình ảnh đầy đủ về những gì đang xảy ra trong quá trình điều tra một vụ việc.

Quá trình điều tra quan hệ gồm bốn bước (Hình 4.46).

##### Bước 1: Điều tra các đối tượng chính và thực hiện điều tra sơ bộ về các cảnh báo

Trong một cuộc điều tra của cảnh sát, cơ quan pháp luật thường được thông báo về một sự kiện theo một đơn khiếu nại, thường được gửi đi từ đồn cảnh sát. Đơn khiếu nại này đưa ra thông tin về các đối tượng liên quan đến việc khiếu nại và bản chất của việc khiếu nại riêng của mình.

Khi đến hiện trường, điều đầu tiên một điều tra viên làm là xác định các đối tượng (chính) tham gia và xác định liệu đơn khiếu nại có giá trị điều tra thêm hay không. Xác định này được thực hiện dựa trên luật pháp. Và phán đoán ban đầu của điều tra viên là liệu có khả năng vụ việc tồn tại yếu tố vi phạm pháp luật hay không. Nếu cho rằng khả năng này tồn tại, điều tra viên sẽ bắt đầu thu thập thông tin từ mỗi đối tượng liên quan, bao gồm các xác minh rằng họ có dấu hiệu hợp pháp, xem hồ sơ hình sự trước đó, và thực hiện kiểm tra sơ bộ để xác định xem họ đang sở hữu vũ khí hoặc vật bất hợp pháp.



**Hình 4.46 Phương pháp phân tích điều tra quan hệ**

Trong một cuộc điều tra NSM, các chuyên gia phân tích thường được thông báo về một sự kiện từ dữ liệu cảnh báo, bao gồm cả các thông báo được tạo ra bởi một IDS. Cảnh báo này thường bao gồm các máy tính có liên quan với các sự kiện và tính chất của các cảnh báo. Trong trường hợp này, các cảnh báo tương tự như khiếu nại, và các máy chủ cũng tương tự như các đối tượng. Trong một chuỗi các sự kiện tương tự, các chuyên gia phân tích NSM phải thực hiện một xác định sơ bộ là liệu cảnh báo có giá trị điều tra thêm hay không. Thông thường, điều này có nghĩa là kiểm tra chi tiết các luật hoặc cơ chế phát hiện gây ra các cảnh báo, và xác định liệu những lưu lượng gắn với nó có thực sự phù hợp với cảnh báo không. Về cơ bản, đây là một nỗ lực để nhanh chóng xác định có dương tính giả xảy ra hay không. Nếu những cảnh báo không phải là dương tính giả, bước tiếp theo của việc phân tích nên bắt đầu với việc thu thập thông tin về các đối tượng chính gắn với các cảnh báo: các địa chỉ IP của các tài nguyên mạng tin cậy và nguy hiểm. Điều này bao gồm việc thu thập TI chiến thuật và các tài nguyên cần bảo vệ như đã thảo luận trong phần trước.

#### **Bước 2: Điều tra mối quan hệ chính và tương tác hiện tại**

Khi điều tra viên đã điều tra cả hai đối tượng, anh ta sẽ điều tra các mối quan hệ giữa các đối tượng này, bao gồm các mối quan hệ trước đó cũng như các tương tác hiện tại. Ví dụ, khi xem xét một đơn khiếu nại, điều tra viên sẽ cố gắng xác định xem hai đối tượng đã ở trong một mối quan hệ như thế nào, thời gian của mối quan hệ đó, các đối tượng sống chung với nhau hay không,... Sau đó, điều tra viên sẽ xác định những hành động xảy ra dẫn đến việc khiếu nại, khi nào sự kiện dẫn tới căng thẳng như hiện tại, và những gì xảy ra sau đó.

Các chuyên gia phân tích NSM sẽ làm điều tương tự để điều tra mối quan hệ cơ bản giữa các máy tính cần bảo vệ và máy tính nguy hiểm. Họ bắt đầu bằng việc xác định bản chất các kết nối trước đó giữa các máy. Các câu hỏi sau đây có thể đưa ra:

- Hai máy tính này đã từng liên lạc với nhau trước đó?
- Nếu có thì cổng, giao thức, và các dịch vụ nào có liên quan?

Tiếp theo, các chuyên gia phân tích sẽ điều tra kỹ lưỡng các kết nối gắn với các cảnh báo ban đầu. Đây là nơi mà dữ liệu từ nhiều nguồn được lấy và phân tích để tìm kiếm các kết nối, bao gồm các hoạt động như sau:

- Thu thập dữ liệu PCAP
- Thực hiện phân tích gói
- Thu thập dữ liệu PSTR
- Trích xuất các tệp tin và thực hiện phân tích phần mềm độc hại
- Tạo thống kê từ dữ liệu phiên

Trong một số trường hợp, lúc này các chuyên gia phân tích có thể xác định liệu một sự cố đã xảy ra hay không. Khi đó, các điều tra có thể kết thúc ở đây. Nếu vụ việc không được định nghĩa rõ ràng vào thời điểm này hay không có quyết định cụ thể nào, các bước tiếp theo sẽ được thực hiện.

### ***Bước 3: Điều tra các đối tượng thứ cấp và mối quan hệ***

Khi một điều tra viên đang điều tra các đối tượng chính và mối quan hệ giữa chúng, thông thường họ sẽ xác định các đối tượng thứ cấp. Đây là những cá nhân có liên quan đến việc khiếu nại theo một cách nào đó, và có thể bao gồm các cộng sự của đối tượng làm đơn khiếu nại, hoặc các nhân chứng khác. Khi các đối tượng này được xác định, các điều tra thường được hỗ trợ thông qua thực hiện các bước điều tra tương tự được nêu trong hai bước đầu tiên, bao gồm một cuộc điều tra của các đối tượng này, cũng như các mối quan hệ giữa họ và các đối tượng chính.

Trong một cuộc điều tra NSM, điều này xảy ra thường xuyên. Ví dụ, khi đang điều tra mối quan hệ giữa hai máy tính, một chuyên gia phân tích có thể thấy rằng các máy tính cần được bảo vệ đã giao tiếp với các máy tính nguy hại khác hoặc ngược lại. Hơn nữa, phân tích các tệp tin độc hại có thể mang lại các địa chỉ IP để lộ nguồn các liên lạc gây nghi vấn khác. Những máy tính này đều được coi là đối tượng thứ cấp.

Khi đối tượng thứ cấp được xác định, chúng cần được điều tra theo cách tương tự như các đối tượng chính. Tiếp đó, các mối quan hệ giữa các đối tượng thứ cấp và các đối tượng chính cần được kiểm tra.

#### **Bước 4: Điều tra bổ sung về quan hệ của các đối tượng**

Tại thời điểm này, việc điều tra các đối tượng và các mối quan hệ nên lặp lại nhiều lần khi cần thiết, và có thể đòi hỏi các đối tượng mức ba hoặc mức bốn. Khi thực hiện, nên đánh giá các đối tượng và các mối quan hệ một cách đầy đủ trên cơ sở của mỗi cấp độ, điều tra đầy đủ mỗi mức trước khi chuyển sang mức kế tiếp, nếu không sẽ rất dễ dàng mất dấu và bỏ quên các kết nối quan trọng khi xem xét các máy tính khác. Khi kết thúc, có thể mô tả mối quan hệ giữa các đối tượng và cách các hoạt động độc hại đã xảy ra.

**Để minh họa quy trình điều tra quan hệ, xét một kịch bản như sau:**

#### **Bước 1: Điều tra các đối tượng chính và thực hiện điều tra sơ bộ về các khiếu nại**

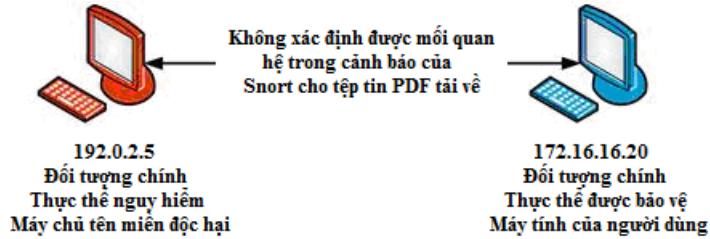
Các chuyên gia phân tích được thông báo rằng một sự kiện bất thường đã được phát hiện với các cảnh báo Snort sau đây:

ET WEB\_CLIENT PDF With Embedded File

Trong cảnh báo này, IP nguồn là 192.0.2.5 (máy tính A nguy hiểm) và địa chỉ IP đích là 172.16.16.20 (Host B cần bảo vệ). Đây là những đối tượng chính. Việc kiểm tra sơ bộ về liên lạc gắn với hoạt động này chỉ ra rằng có một tệp tin PDF được tải về. Các dữ liệu PCAP cho chuỗi liên lạc thu được, và các tệp tin PDF được chiết xuất từ các tệp tin bằng cách sử dụng Wireshark. Các chuỗi băm MD5 của tệp tin PDF được gửi đến Team Cymru Malware Hash Registry, và nó chỉ ra rằng 23% các engine phát hiện vi-rút cho rằng tập tin này là độc hại. Dựa trên điều này, nên thực hiện các quyết định tiếp tục điều tra sau đó.

Bước tiếp theo là thu thập dữ liệu một cách thân thiện và có chiến thuật đe dọa tình báo liên quan đến cả hai máy chủ. Quá trình này sẽ xác định các thông tin sau đây:

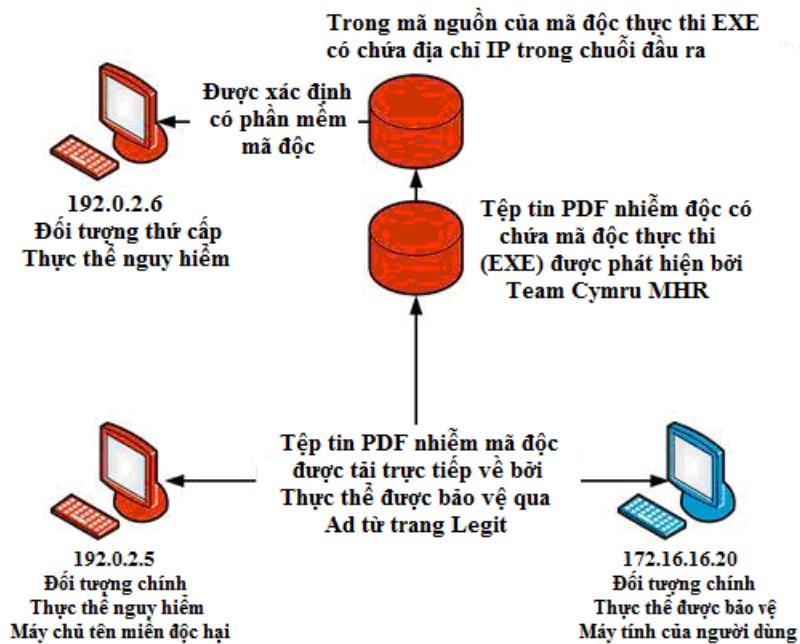
- Thông tin về các tài nguyên mạng cần bảo vệ cho 172.16.16.20:
  - Hệ thống này là một máy trạm của người dùng đang chạy Windows 7
  - Hệ thống không có các dịch vụ nghe hoặc mở cổng
  - Người dùng hệ thống này lướt web thường xuyên, và nhiều thông báo tài sản mới tồn tại trong dữ liệu PRADS
- TI đối với 192.0.2.5:
  - IPVoid trả về 0 kết quả trên các danh sách đen đối với địa chỉ IP này
  - URLVoid tìm thấy 5 kết quả trên các danh sách đen cho tên miền nơi các tệp tin PDF đã được tải về
  - Dữ liệu NetFlow chỉ ra rằng địa chỉ IP này đã không liên lạc với bất kỳ thiết bị nào khác trên mạng được bảo vệ



**Hình 4.47 Các đối tượng chính ban đầu**

#### Bước 2: Điều tra mối quan hệ chính và tương tác hiện tại

Để điều tra các mối quan hệ giữa 172.16.16.20 và 192.0.2.5, hành động đầu tiên được thực hiện là một phân tích các dữ liệu gói tin cho liên lạc đã xảy ra vào khoảng thời gian có cảnh báo. Gói dữ liệu sẽ được tải về cho liên lạc giữa hai máy này với khoảng thời gian thiết lập để lấy dữ liệu từ 10 phút trước khi cảnh báo xảy ra tới 10 phút sau khi cảnh báo xảy ra. Sau khi thực hiện phân tích gói tin trên dữ liệu này, xác định được rằng máy tính được bảo vệ đã được chuyển hướng đến máy tính độc hại từ một quảng cáo của bên thứ ba trên một trang web hợp pháp. Các máy tính được bảo vệ tải tệp tin về, và giao tiếp với các máy tính không còn nguy hiểm.



**Hình 4.48 Quan hệ của các đối tượng chính**

Các bước tiếp theo thực hiện để điều tra mối quan hệ giữa 172.16.16.20 và 192.0.2.5 là kiểm tra các tệp tin PDF đã được tải về. Tệp tin PDF này được gửi lên một Cuckoo sandbox để thực hiện phân tích mã độc tự động. Các phân tích hành vi của tập tin này chỉ ra rằng PDF này có chứa một tệp tin thực thi. Các tệp tin thực thi có chứa địa chỉ IP 192.0.2.6 trong cấu hình của nó. Không có thông tin khác được xác định từ các phân tích phần mềm độc hại về tập tin này.

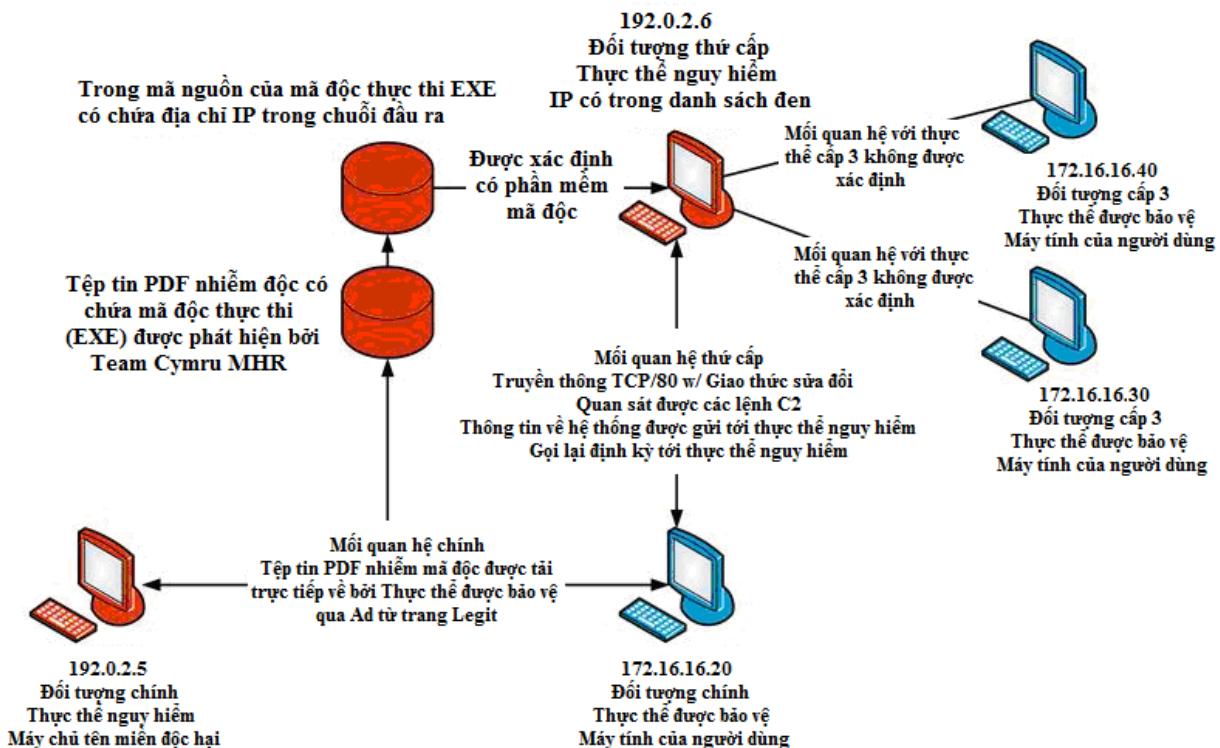
Tại thời điểm này, đã hoàn thành việc điều tra của các đối tượng chính và mối quan hệ giữa chúng. Trong khi tất cả mọi thứ đều cho thấy rằng đây là một sự cố, chúng ta vẫn chưa thể hoàn toàn chắc chắn điều này. Tuy nhiên, đã xác định được một đối tượng thứ cấp, vì vậy chúng ta cần sẽ chuyển sang bước tiếp theo của cuộc điều tra với dữ liệu đó.

### Bước 3: Điều tra các đối tượng thứ cấp và mối quan hệ

Đã xác định được đối tượng thứ cấp 192.0.2.6 mã hóa trong tệp tin thực thi đã được gắn trong tệp tin PDF được tải xuống bởi đối tượng chính. Bây giờ, cần phải điều tra đối tượng bằng cách thu thập thông tin TI đối với địa chỉ IP này:

TI đối với 192.0.2.6:

- IPVoid trả về 2 kết quả phù hợp trên danh sách đen cho địa chỉ IP này.
- Dữ liệu NetFlow cho thấy đối tượng chính 172.16.16.20 đã liên lạc với máy tính này. Giao tiếp xảy ra khoảng ba mươi phút sau cảnh báo ban đầu.
- Dữ liệu NetFlow chỉ ra rằng hai máy tính được bảo vệ khác trên mạng của tổ chức đã giao tiếp với địa chỉ IP này theo định kỳ với lưu lượng thấp trong vài ngày qua. Địa chỉ của chúng là 172.16.16.30 và 172.16.16.40.



Hình 4.49 Quan hệ của các đối tượng chính và thứ cấp

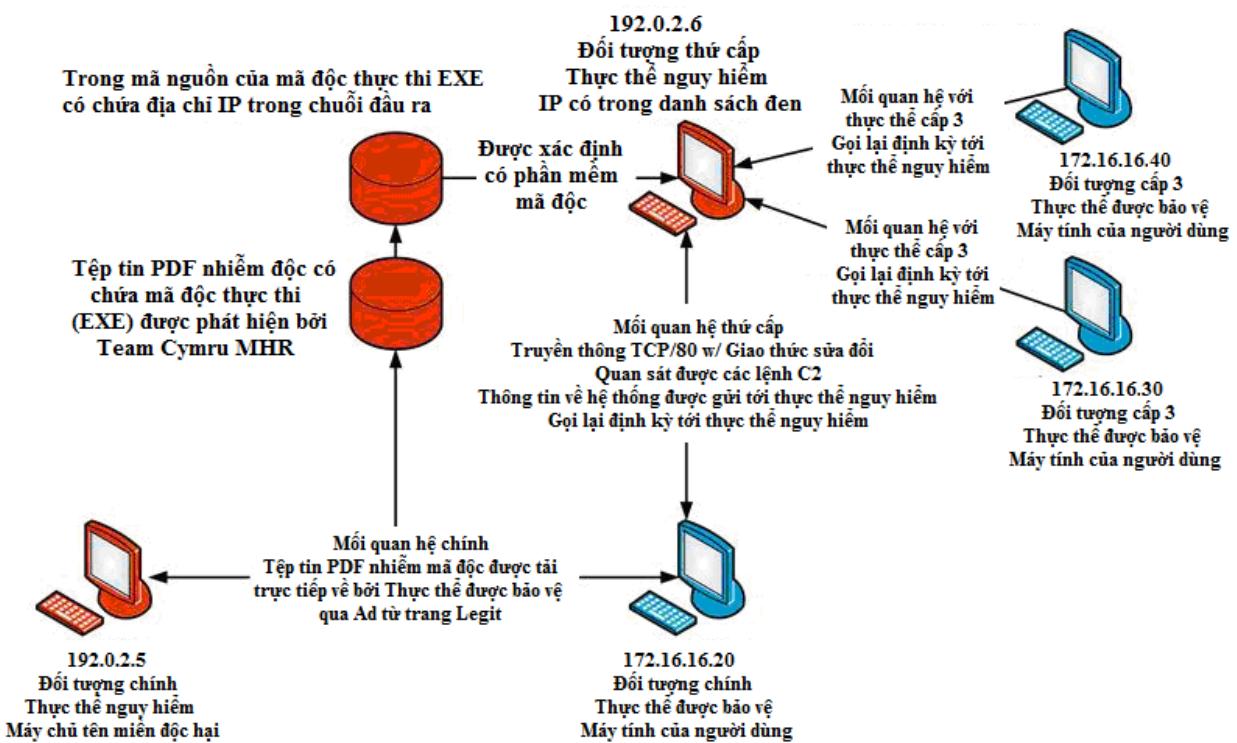
Dựa vào các thông tin trên, nhận thấy rằng vấn đề này có thể nguy hiểm hơn chúng ta nghĩ lúc đầu. Tiếp theo, cần phải xác định mối quan hệ giữa đối tượng thứ cấp 192.0.2.6 và đối tượng

chính 172.16.16.20. Dựa trên thông tin TI biết rằng giao tiếp xảy ra giữa hai thiết bị. Bước tiếp theo là thu thập dữ liệu PCAP về liên lạc xảy ra giữa các máy tính. Khi dữ liệu này được thu thập, phân tích cho thấy mặc dù các thiết bị này giao tiếp trên cổng 80, nhưng chúng không sử dụng các giao thức HTTP, thay vào đó, chúng đang sử dụng một giao thức sửa đổi, và có thể thấy rằng các lệnh đã được thực thi cho hệ thống này. Những lệnh này dẫn đến các hệ thống được bảo vệ đã truyền đi các thông tin hệ thống cho máy tính nguy hiểm. Tại thời điểm này cũng nhận thấy có một cuộc gọi lại (call back) định kỳ được truyền đến máy tính nguy hiểm.

Lúc này, chúng ta đã có đủ thông tin để quyết định rằng nên khai báo một sự cố, và xác định là 172.16.16.20 đã bị tổn hại (Hình 4.49). Trong một số trường hợp, việc điều tra có thể kết thúc ở đây. Tuy nhiên, hãy nhớ là đã xác định hai máy tính bổ sung (bây giờ được xác định là các đối tượng mức ba) đã giao tiếp với IP 192.0.2.6 nguy hiểm. Điều này có nghĩa là có thể các máy tính đó cũng có thể bị tổn hại.

#### **Bước 4: Điều tra bổ sung về quan hệ của các đối tượng**

Một kiểm tra về các gói dữ liệu truyền giữa các máy tính mức ba và 172.16.16.20 tiết lộ rằng nó cũng đang tham gia vào hành vi callback như đã được xác định trong các máy tính được bảo vệ (Hình 4.50). Do vậy, có thể xác định rằng các máy tính được bảo vệ ở mức ba cũng đang bị tổn hại.



**Hình 4.50 Quan hệ của tất cả các đối tượng**

## **Tổng kết về sự có:**

Kịch bản này được dựa trên một sự có thực sự xảy ra trong một doanh nghiệp. Việc sử dụng quy trình phân tích hệ thống để xác định các máy tính và xây dựng các mối quan hệ giữa chúng không chỉ cho phép chúng ta xác định liệu một tấn công xảy ra hay không, mà còn cho chúng ta tìm các máy tính khác cũng đã bị tổn hại nhưng không được xác định trong các cảnh báo ban đầu. Đây là một ví dụ điển hình về một quá trình có cấu trúc có thể giúp một chuyên gia phân tích thực hiện công việc điều tra từ A đến Z mà không bị sai sót hoặc quá tải thông tin. Rất dễ bị lạc lối trong một kịch bản như thế này, do vậy, điều quan trọng là tiếp cận từng bước như dự định và không đi quá xa khỏi con đường đang đi.

### **4.3.1.2 Chẩn đoán khác biệt**

Mục tiêu của một chuyên gia phân tích NSM là phân loại các cảnh báo được tạo ra bởi cơ chế phát hiện và điều tra các nguồn dữ liệu để thực hiện các kiểm tra có liên quan, nghiên cứu để xem liệu có vi phạm an ninh mạng nào đã xảy ra hay không. Mục tiêu này rất tương tự với mục tiêu của một bác sĩ, là phân loại các dấu hiệu biểu hiện của một bệnh nhân và điều tra nhiều nguồn dữ liệu, thực hiện các xét nghiệm có liên quan rồi nghiên cứu xem liệu các phát hiện của họ có cho thấy một lỗ hổng nào đó trong hệ thống miễn dịch của người hay không.

Một trong những phương pháp chẩn đoán phổ biến nhất được sử dụng trong y học lâm sàng được gọi là chẩn đoán phân biệt. Trong đó, nhóm các bác sĩ sẽ trình bày một tập các dấu hiệu và họ sẽ tạo ra một danh sách các chẩn đoán tiềm năng trên một bảng trắng. Phần còn lại sẽ dành cho nghiên cứu và thực hiện các xét nghiệm khác nhau để loại bỏ từng kết luận có khả năng cho đến khi chỉ có một kết luận còn lại.

Phương pháp chuẩn đoán phân biệt dựa trên một quá trình loại trừ, bao gồm năm bước riêng biệt, mặc dù trong một số trường hợp chỉ cần có hai bước. Quy trình chuẩn đoán phân biệt như sau:

#### **Bước 1: Xác định và liệt kê các dấu hiệu**

Trong y học, các dấu hiệu thường được truyền đạt bằng lời nói của những người đang bị bệnh. Trong NSM, một dấu hiệu phổ biến nhất là một cảnh báo được tạo ra bởi một số hình thức của hệ thống phát hiện xâm nhập hoặc các phần mềm phát hiện khác. Mặc dù bước này tập trung chủ yếu vào các dấu hiệu ban đầu, các dấu hiệu khác có thể được bổ sung vào danh sách này khi tiến hành kiểm tra hoặc điều tra bổ sung.

#### **Bước 2: Xem xét và đánh giá chẩn đoán phổ biến nhất đầu tiên**

Chẩn đoán phổ biến nhất có khả năng là chính xác, vậy nên chẩn đoán này nên được đánh giá đầu tiên. Chuyên gia phân tích cần tập trung vào điều tra cần thiết để nhanh chóng xác nhận chẩn đoán này. Nếu chẩn đoán phổ biến không thể khẳng định trong bước đầu tiên thì chuyên gia phân tích cần tiến hành bước tiếp theo.

### **Bước 3: Liệt kê tất cả chẩn đoán có thể cho các dấu hiệu đã biết**

Bước tiếp theo trong quy trình là liệt kê tất cả các chẩn đoán có thể dựa trên các thông tin hiện có với các dấu hiệu được đánh giá ban đầu. Bước này đòi hỏi một số suy nghĩ sáng tạo và thường là thành công nhất khi có nhiều chuyên gia phân tích tham gia trong việc tạo ra những ý tưởng. Mỗi chẩn đoán có khả năng trong danh sách này sẽ được coi như một ứng cử viên.

### **Bước 4: Đánh giá mức ưu tiên trong danh sách ứng viên theo mức độ nghiêm trọng**

Khi một danh sách các ứng viên được tạo ra, bác sĩ sẽ đánh mức ưu tiên các ứng viên bằng cách cho mức ưu tiên cao nhất nếu đó là mối đe dọa lớn nhất tới cuộc sống con người. Trong trường hợp của một chuyên gia phân tích NSM, cũng cần đánh mức ưu tiên, nhưng cần theo mức độ đe dọa đối với an ninh mạng của tổ chức. Điều này sẽ phụ thuộc vào bản chất của tổ chức. Ví dụ, nếu "MySQL Database Root Compromise" là một điều kiện cần xem xét thì một công ty có cơ sở dữ liệu chứa số liệu an sinh xã hội sẽ được ưu tiên đánh giá cao điều kiện này nhiều hơn so với một công ty sử dụng một cơ sở dữ liệu đơn giản để lưu trữ một danh sách bán hàng của nhân viên của mình.

### **Bước 5: Loại bỏ các điều kiện ứng viên, và bắt đầu với cái nghiêm trọng nhất**

Bước cuối cùng là nơi mà phần lớn các hành động xảy ra. Dựa trên danh sách ưu tiên tạo ra trong bước trước, chuyên gia phân tích nên bắt đầu làm những gì là cần thiết để loại bỏ các điều kiện ứng cử viên, bắt đầu với những điều kiện đặt ra mối đe dọa lớn đối với an ninh mạng. Quá trình này loại bỏ yêu cầu xem xét từng điều kiện ứng cử viên và thực hiện các kiểm tra, tiến hành nghiên cứu, điều tra các nguồn dữ liệu khác nhằm loại trừ chúng ra khỏi danh sách các khả năng. Trong một số trường hợp, điều tra một điều kiện ứng viên có thể loại trừ nhiều điều kiện ứng viên, từ đó thúc đẩy nhanh quá trình này. Cuối cùng, mục tiêu của bước cuối cùng là đưa ra một chẩn đoán, vì vậy một sự cố có thể được khai báo hoặc cảnh báo có thể bị loại bỏ như một dương tính giả. Lưu ý rằng "Liên lạc bình thường" là một chẩn đoán hoàn toàn chấp nhận được, và sẽ là chẩn đoán phổ biến nhất mà một chuyên gia phân tích NSM thường có.

### **Ví dụ tình huống**

#### **Bước 1: Xác định và liệt kê các dấu hiệu**

Các dấu hiệu sau đây được quan sát qua cảnh báo IDS và điều tra các dữ liệu đã có:

1. Một máy chủ tin cậy bắt đầu gửi lưu lượng đến một địa chỉ IP ở Nga
2. Luồng dữ liệu diễn ra đều đặn sau mỗi 10 phút
3. Lưu lượng là HTTPS trên cổng 443, và như vậy sẽ được mã hóa và không thể đọc

#### **Bước 2: Xem xét và đánh giá chẩn đoán thường thấy nhất đầu tiên**

Dựa trên những dấu hiệu này, giả định hợp lý nhất là máy bị nhiễm một số dạng phần mềm độc hại. Dữ liệu sẽ đi đến một địa chỉ IP ở Nga thường xuyên 10 phút một lần. Mặc dù những điều đó là đáng chú ý, chúng ta không thể kết luận đó là phần mềm độc hại. Có rất nhiều cơ chế

giao tiếp định kỳ bình thường. Ví dụ như chat dựa trên web, RSS, e-mail trên web, các mã chứng khoán, quy trình cập nhật phần mềm, và nhiều hơn nữa. Với nguyên tắc là nếu không chứng minh được dữ liệu là xấu thì chúng sẽ là tốt. Do đó nên nghĩ rằng việc chẩn đoán phổ biến nhất ở đây là đây là bình thường.

Nếu vẫn có nghi ngờ, trong trường hợp này, có thể bắt đầu với một số bộ sưu tập TI với các IP ở Nga. Một kỹ thuật khác là để kiểm tra các bản ghi hệ thống hay IDS trên máy chủ xem liệu có bất kỳ hoạt động đáng ngờ nào đang xảy ra trên máy tính tại các khoảng thời gian tương tự như lưu lượng hiện tại. Cách khác nữa là kiểm tra TI cho các thiết bị cần bảo vệ. Ví dụ, người dùng từ Nga? Họ sử dụng một phần mềm chống vi-rút (như Kaspersky) với máy chủ cập nhật có thể ở Nga? Những điều này có thể giúp xác định xem lưu lượng có bình thường hay không.

Mục đích ở đây là giả định rằng chưa thể đưa ra quyết định cuối cùng về việc đây có phải là giao tiếp bình thường hay không.

### **Bước 3: Liệt kê tất cả phán đoán có thể cho các dấu hiệu đã biết**

Có thể có một số lựa chọn cho kịch bản hiện tại. Để ngắn gọn, ở đây chỉ liệt kê một số:

- Truyền thông bình thường
- Nhiễm mã độc
- Dữ liệu bị rò rỉ từ máy tính bị tổn hại
- Cấu hình sai: cũng có khả năng quản trị hệ thống gõ sai một địa chỉ IP và có một phần mềm đang phải giao tiếp định kỳ với một hệ thống có địa chỉ IP ở Nga. Trường hợp này cũng khá phổ biến trong thực tế.

### **Bước 4: Sắp xếp danh sách ưu tiên theo mức độ nghiêm trọng**

Mức ưu tiên sẽ khác nhau tùy thuộc vào mức độ rủi ro tới tổ chức. Để tổng quát, chúng ta lựa chọn các mức ưu tiên như sau, với mức ưu tiên 1 là cao nhất:

- Ưu tiên 1: Số liệu rò rỉ từ máy tính bị tổn hại
- Ưu tiên 2: Nhiễm mã độc
- Ưu tiên 3: Cấu hình sai
- Ưu tiên 4: Liên lạc bình thường

### **Bước 5: Loại bỏ dần các phán đoán, bắt đầu với những phán đoán nghiêm trọng nhất**

Lúc này có thể thu thập dữ liệu và thực hiện các kiểm tra để loại trừ dần từng phán đoán.

- *Ưu tiên 1: Dữ liệu rò rỉ từ máy tính bị tổn hại*

Phán đoán này có chút khó khăn để loại bỏ. Bắt toàn bộ nội dung gói tin sẽ không cung cấp nhiều trợ giúp do lưu lượng đã mã hóa. Nếu có dữ liệu phiên sẽ có thể xác định số lượng dữ liệu đi ra ngoài. Nếu chỉ có một vài byte đi ra ngoài 10 phút một lần thì có khả năng đây

không phải là rò rỉ dữ liệu, vì nếu rò rỉ thì sẽ có lượng lớn dữ liệu đi ra ngoài. Cũng hữu ích trong việc xác định liệu có máy chủ khác trên mạng của tổ chức đang liên lạc với địa chỉ IP này hoặc bất kỳ IP nào khác trong không gian địa chỉ. Cuối cùng, việc so sánh lưu lượng bình thường của máy chủ nội bộ với các lưu lượng có khả năng là độc hại sẽ giúp cung cấp một số thông tin hữu ích. Có thể được thực hiện điều này với dữ liệu TI cho các máy tính cần bảo vệ, như các dữ liệu được thu thập bởi PRADS.

- *Ưu tiên 2: Nghiêm mã độc*

Có thể kiểm tra bằng phần mềm chống vi-rút mạng hoặc bản ghi chi tiết HIDS.

- *Ưu tiên 3: Cấu hình sai*

Cách xử lý tốt nhất trong trường hợp này là so sánh lưu lượng truy cập của các máy tính tin cậy với lưu lượng của một hoặc nhiều máy tính với vai trò tương tự trên mạng. Nếu máy tính khác trên cùng mạng con (subnet) có các mẫu lưu lượng giống nhau, nhưng đi đến địa chỉ IP khác nhau, thì có khả năng địa chỉ IP sai được nhập vào một phần mềm ở đâu đó. Nếu có quyền truy cập thì nên kiểm tra các bản ghi trên máy tính, vì điều này sẽ hữu ích trong việc tìm ra sai sót do hồ sơ của các sai sót có thể nằm trong log hệ thống của Windows hoặc Unix.

- *Ưu tiên 4: Truyền thông bình thường*

### Tiến hành chẩn đoán

Tại thời điểm này, cần sử dụng kinh nghiệm cùng trực giác của chuyên gia phân tích để quyết định liệu có gì độc hại thực sự đã xảy ra. Nếu việc phân tích được hoàn thành một cách kỹ lưỡng trước đó thì có thể cho rằng hoạt động trên là tốt. Nếu vẫn còn linh cảm điều gì đó kỳ quặc đang xảy ra thì cần theo dõi các máy tính thêm, rồi đánh giá lại dữ liệu đã thu thập thêm một lần nữa.

#### 4.3.1.3 Thực hiện các phương pháp phân tích

Hai phương pháp phân tích đã mô tả ở đây rất khác nhau. Thực sự là không có công thức rõ ràng cho việc lựa chọn phương pháp đúng do mỗi phương pháp có những điểm mạnh và điểm yếu và năng lực của chuyên gia phân tích. Tuy nhiên, phương pháp điều tra quan hệ có thể tốt hơn trong các tình huống phức tạp và có nhiều máy tính tham gia. Đó là do phương pháp này có khả năng theo dõi một lượng lớn các thực thể và các mối quan hệ mà không sợ quá tải hoặc gây lỗi. Phương pháp chẩn đoán khác biệt có thể làm việc tốt trong các tình huống có ít máy tính liên quan và có thể gắn với một vài triệu chứng khác biệt.

#### 4.3.2 Các phương pháp quy chuẩn thực tiễn tốt nhất cho phân tích

##### a. Luôn đặt ra các giả định

Việc phân tích luôn phải dựa trên các giả định và dự đoán từ những thông tin được xác định. Ban đầu, những thông tin này có thể ít, nhưng về sau có thể được bổ sung thêm các thông tin khác có liên quan, dẫn đến việc dự đoán có thể thay đổi. Điều này là rất bình thường, nhưng lại có thể

mang lại những kết quả khả quan hơn. Vì vậy, luôn cần đặt ra các câu hỏi để làm tăng thêm giả định và những dự đoán.

#### **b. Cần phải lưu ý về dữ liệu**

Chuyên gia phân tích phụ thuộc vào dữ liệu để thực hiện công việc của họ. Những dữ liệu này có thể có nhiều dạng, như là một tệp tin PCAP, một bản ghi PSTR, hoặc một tệp tin IIS. Vì hầu hết thời gian là dành cho việc sử dụng các công cụ khác nhau để tương tác với dữ liệu, nên điều quan trọng là phải lưu ý về cách công cụ tương tác với dữ liệu. Tuy nhiên, các công cụ được tạo ra đôi khi có những "tính năng" có thể làm che dấu dữ liệu và ngăn cản việc phân tích phù hợp.

#### **c. Nên làm việc theo nhóm**

Mỗi người giỏi một lĩnh vực khác nhau, và không ai luôn hoạt động với 100% hiệu suất. Do vậy, bất cứ khi nào có thể, chúng ta nên cùng nhau xem xét những vấn đề đang gặp phải.

#### **d. Không bao giờ đánh động tin tức**

Vấn đề là khoảng 99% thời gian chúng ta không biết đang đối mặt với ai hoặc những gì. Mặc dù có thể chỉ nhìn thấy hoạt động quét, nhưng các máy chủ đó có thể được vận hành bởi một nhóm lớn những kẻ tấn công hoặc thậm chí là một bộ phận quân sự của một nước khác. Ngay cả một hoạt động đơn giản như ping cũng có thể lộ ra là bạn biết chúng đang tồn tại. Điều này sẽ thúc đẩy tin tức thay đổi chiến thuật. Chúng ta không thể biết những người mà chúng ta đang đối phó, động lực của họ là gì, hay khả năng của họ, vì vậy không bao giờ nên đánh động họ. Vấn đề đơn giản là chúng ta không biết liệu có khả năng xử lý được các hậu quả hay không.

#### **e. Gói tin vốn dĩ là tốt**

Các gói tin phải được coi là vô hại cho đến khi được chứng minh có hại.

#### **f. Wireshark chỉ là một công cụ phân tích**

Chuyên gia phân tích cần phải hiểu rằng các công cụ rất quan trọng cho công việc, nhưng chúng chỉ là một phần trong đó. Wireshark cũng chỉ là một công cụ trong kho "vũ khí" của một chuyên gia phân tích cho phép anh ta tìm ra thông tin về các gói tin.

Chuyên gia phân tích cần tiếp cận một cách khoa học, bổ sung các công cụ và quy trình, nhận thức được bức tranh toàn cảnh, chú ý đến từng chi tiết, và cuối cùng là kết hợp tất cả những điều này cùng với những kinh nghiệm có được qua thời gian sẽ giúp họ phát triển kỹ thuật phân tích của riêng mình.

#### **g. Cần thực hiện phân loại sự kiện rõ ràng**

Cần phải có một hệ thống phân loại để xác định các vụ việc cần ưu tiên điều tra và thông báo. Ví dụ như DoD Cyber Incident và Cyber Event Categorization System. Bảng 4.3 mô tả sơ bộ các loại, sắp xếp theo thứ tự ưu tiên mỗi loại nên dùng.

**Bảng 4.3 DoD Cyber Incident và Cyber Event Categorization System**

Ưu tiên	Phân loại	Tên	Sự cố/sự kiện Incident/Event
0	0	Training and Exercise	N/A
1	1	Root-Level Intrusion	Incident
2	2	User-Level Intrusion	Incident
3	4	Denial of Service	Incident
4	7	[Installed/Executed] Malicious Logic	Incident
5	3	Unsuccessful Activity Attempt	Event
6	5	Non-Compliance Activity	Event
7	6	Reconnaissance	Event
8	8	Investigating	Event
9	9	Explained Anomaly	Event

#### **h. Quy tắc của 10**

Các chuyên gia phân tích mới thường có thói quen lấy quá nhiều dữ liệu hoặc quá ít khi điều tra một sự kiện xảy ra tại một thời điểm cụ thể. Ví dụ, chuyên gia phân tích thấy một sự kiện xảy ra vào ngày 07 tháng 10, lúc 08:35 và sẽ cố gắng để lấy dữ liệu NSM gắn với máy tính đó cho cả ngày 7 tháng 10. Điều này sẽ tạo ra tình huống là có quá nhiều dữ liệu cần cho phân tích. Ngược lại, nếu chỉ lấy dữ liệu xảy ra vào ngày 07 tháng 10, lúc 08:35, chính xác theo phút, sẽ tạo ra tình huống là không đủ thông tin để xác định chính xác những gì đã xảy ra.

Để ngăn chặn việc này, cần theo quy tắc của 10. Quy tắc này nói rằng bất cứ lúc nào cần phân tích một sự kiện xảy ra tại một thời điểm, nên bắt đầu bằng cách lấy dữ liệu xảy ra 10 phút trước và 10 phút sau đó, khi đó chuyên gia phân tích sẽ có đủ dữ liệu để xác định những gì đã dẫn đến sự kiện và những gì xảy ra sau đó. Khi chuyên gia phân tích thực hiện phân tích dữ liệu này, họ có thể đưa ra quyết định về việc lấy dữ liệu nhiều hơn nếu cần. Tuy nhiên, quy tắc này không phù hợp với mọi tình huống, nhưng nó có hiệu quả tốt cho các chuyên gia phân tích mới trong 99% của các cuộc điều tra.

## TÀI LIỆU THAM KHẢO

- [1] Chris Sanders and Jason Smith, *Applied Network Security Monitoring*, Syngress, 2014
- [2] Richard Bejtlich, *The Tao of Network Security Monitoring: Beyond Intrusion Detection*, Addison-Wesley, 2004
- [3] Richard Bejtlich, *The Practice Of Network Security Monitoring*, No Starch Press, 2013
- [4] John R. Vacca, *Network and System Security*, Elsevier Inc., 2010
- [5] Chris Fry and Martin Nystrom, *Security Monitoring*, O'Reilly Media Inc., 2009