

CHƯƠNG 1

TỔNG QUAN VỀ HỆ THỐNG PHÁT HIỆN VÀ NGĂN CHẶN XÂM NHẬP

Hệ thống phát hiện xâm nhập ra đời cách đây khoảng 25 năm và nó đã trở nên rất hữu dụng cho việc bảo vệ các hệ thống mạng và hệ thống máy tính. Bằng cách đưa ra các cảnh báo khi có dấu hiệu của sự xâm nhập đến hệ thống. Nhưng hệ thống IDS vẫn có nhiều hạn chế khi đưa ra các cảnh báo sai và cần có người giám sát. Thế hệ tiếp theo của IDS là hệ thống IPS ra đời năm 2004, đang trở nên rất phổ biến và đang dần thay thế cho các hệ thống IDS. Hệ thống IPS bao gồm cơ chế phát hiện, đưa ra các cảnh báo và còn có thể ngăn chặn các hoạt động tấn công bằng cách kết hợp với firewall.

1.1. HỆ THỐNG PHÁT HIỆN XÂM NHẬP

1.1.1. Khái niệm

Hệ thống phát hiện xâm nhập IDS là thiết bị phần cứng, phần mềm hay có sự kết hợp của cả hai để thực hiện việc giám sát, theo dõi và thu thập thông tin từ nhiều nguồn khác nhau. Sau đó sẽ phân tích để tìm ra dấu hiệu của sự xâm nhập hay tấn công hệ thống và thông báo đến người quản trị hệ thống. Nói một cách tổng quát, IDS là hệ thống phát hiện các dấu hiệu làm hại đến tính bảo mật, tính toàn vẹn và tính sẵn dùng của hệ thống máy tính hoặc hệ thống mạng, làm cơ sở cho bảo đảm an ninh hệ thống.

1.1.2. Phát hiện xâm nhập

Phát hiện xâm nhập là tập hợp các kỹ thuật và phương pháp được sử dụng để phát hiện các hành vi đáng ngờ cả ở cấp độ mạng và máy chủ. Hệ thống phát hiện xâm nhập phân thành hai loại cơ bản:

- Hệ thống phát hiện dựa trên dấu hiệu xâm nhập.
- Hệ thống phát hiện các dấu hiệu bất thường.

Kẻ tấn công có những dấu hiệu, giống như là virus, có thể được phát hiện bằng cách sử dụng phần mềm. Bằng cách tìm ra dữ liệu của gói tin mà có chứa bất kì dấu hiệu xâm nhập hoặc dị thường được biết đến. Dựa trên một tập hợp các dấu

hiệu (signatures) hoặc các qui tắc (rules). Hệ thống phát hiện có thể dò tìm, ghi lại các hoạt động đáng ngờ này và đưa ra các cảnh báo. Anomaly-based IDS thường dựa vào phần header giao thức của gói tin được cho là bất thường. Trong một số trường hợp các phương pháp có kết quả tốt hơn với Signature-based IDS. Thông thường IDS sẽ bắt lấy các gói tin trên mạng và đối chiếu với các rule để tìm ra các dấu hiệu bất thường của gói tin.

1.1.3. Chính sách của IDS

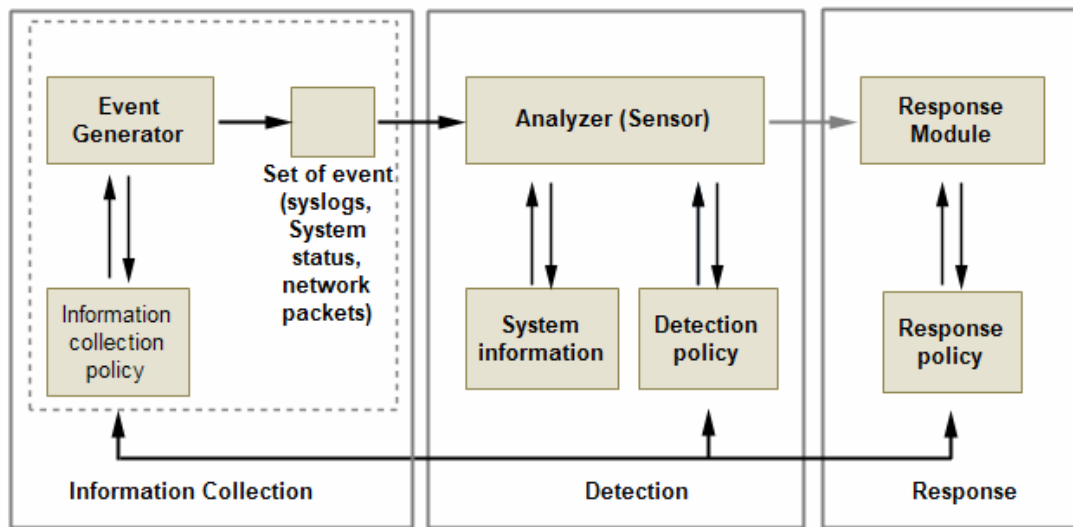
Trước khi cài đặt một hệ thống IDS lên hệ thống thì cần phải có một chính sách để phát hiện kẻ tấn công và cách xử lý khi phát hiện ra các hoạt động tấn công. Bằng cách nào đó chúng phải được áp dụng. Các chính sách cần chứa các phần sau (có thể thêm tùy theo yêu cầu của từng hệ thống):

- Ai sẽ giám sát hệ thống IDS? Tùy thuộc vào IDS, có thể có cơ chế cảnh báo để cung cấp thông tin về các hành động tấn công. Các cảnh báo này có thể ở hình thức văn bản đơn giản (simple text) hoặc chúng có thể ở dạng phức tạp hơn. Có thể được tích hợp vào các hệ thống quản lý mạng tập trung như HP Openview hoặc MySQL database. Cần phải có người quản trị để giám sát các hoạt động xâm nhập và các chính sách cần có người chịu trách nhiệm. Các hoạt động xâm nhập có thể được theo dõi và thông báo theo thời gian thực bằng cách sử dụng cửa sổ pop-up hoặc trên giao diện web. Các nhà quản trị phải có kiến thức về cảnh báo và mức độ an toàn của hệ thống.
- Ai sẽ điều hành IDS? Như với tất cả các hệ thống, IDS cần được được bảo trì thường xuyên.
- Ai sẽ xử lý các sự cố và như thế nào? Nếu các sự cố không được xử lý thì IDS xem như vô tác dụng.
- Các báo cáo có thể được tạo và hiển thị vào cuối ngày hoặc cuối tuần hoặc cuối tháng.
- Cập nhật các dấu hiệu. Các hacker thì luôn tạo ra các kỹ thuật mới để tấn công hệ thống. Các cuộc tấn công này được phát hiện bởi hệ thống IDS dựa trên các dấu hiệu tấn công.
- Các tài liệu thì rất cần thiết cho các dự án. Các chính sách IDS nên được mô tả dưới dạng tài liệu khi các cuộc tấn công được phát hiện. Các tài liệu có thể

bao gồm các log đơn giản hoặc các văn bản. Cần phải xây dựng một số hình thức để ghi và lưu trữ tài liệu. Các báo cáo cũng là các tài liệu.

1.1.4. Kiến trúc của hệ thống phát hiện xâm nhập

Kiến trúc của một hệ thống IDS bao gồm các thành phần chính sau: Thành phần thu thập gói tin (information collection), thành phần phân tích gói tin (detection) và thành phần phản hồi (respotion). Trong ba thành phần này, thành phần phân tích gói tin là quan trọng nhất và bộ cảm biến (sensor) đóng vai trò quan quyết định nên cần được phân tích để hiểu rõ hơn về kiến trúc của một hệ thống phát hiện xâm nhập



Hình 1-1. Kiến trúc của một hệ thống phát hiện xâm nhập

Bộ cảm biến được tích hợp với thành phần sưu tập dữ liệu. Bộ tạo sự kiện. Cách sưu tập này được xác định bởi chính sách tạo sự kiện để định nghĩa chế độ lọc thông tin sự kiện. Bộ tạo sự kiện (hệ điều hành, mạng, ứng dụng) cung cấp một số chính sách thích hợp cho các sự kiện, có thể là một bản ghi các sự kiện của hệ thống hoặc các gói mạng. Sổ chính sách này cùng với thông tin chính sách có thể được lưu trong hệ thống được bảo vệ hoặc bên ngoài.

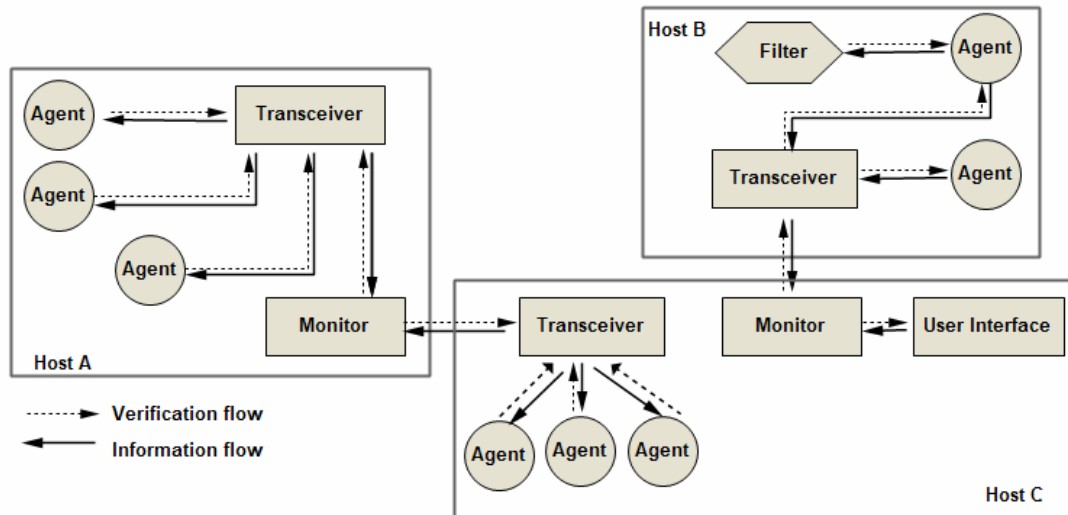
Vai trò của bộ cảm biến là dùng để lọc thông tin và loại bỏ dữ liệu không tương thích đạt được từ các sự kiện liên quan với hệ thống bảo vệ, vì vậy có thể phát hiện được các hành động nghi ngờ. Bộ phân tích sử dụng cơ sở dữ liệu chính sách phát hiện cho mục này. Ngoài ra còn có các thành phần: dấu hiệu tấn công, profile hành vi thông thường, các tham số cần thiết (ví dụ: các ngưỡng). Thêm vào

đó, cơ sở dữ liệu giữ các tham số cấu hình, gồm có các chế độ truyền thông với module đáp trả. Bộ cảm biến cũng có cơ sở dữ liệu của riêng nó, gồm dữ liệu lưu về các xâm phạm phức tạp tiềm ẩn (tạo ra từ nhiều hành động khác nhau).

IDS có thể được sắp đặt tập trung (ví dụ như được tích hợp vào trong tường lửa) hoặc phân tán. Một IDS phân tán gồm nhiều IDS khác nhau trên một mạng lớn, tất cả chúng truyền thông với nhau. Nhiều hệ thống tinh vi đi theo nguyên lý cấu trúc một tác nhân, nơi các module nhỏ được tổ chức trên một host trong mạng được bảo vệ.

Vai trò của tác nhân là để kiểm tra và lọc tất cả các hành động bên trong vùng được bảo vệ và phụ thuộc vào phương pháp được đưa ra. Tào phân tích bước đầu và thậm chí đảm trách cả hành động đáp trả. Mạng các tác nhân hợp tác báo cáo đến máy chủ phân tích trung tâm là một trong những thành phần quan trọng của IDS. DIDS có thể sử dụng nhiều công cụ phân tích tinh vi hơn, đặc biệt được trang bị sự phát hiện các tấn công phân tán. Các vai trò khác của tác nhân liên quan đến khả năng lưu động và tính roaming của nó trong các vị trí vật lý. Thêm vào đó, các tác nhân có thể đặc biệt dành cho việc phát hiện dấu hiệu tấn công đã biết nào đó. Đây là một hệ số quyết định khi nói đến nghĩa vụ bảo vệ liên quan đến các kiểu tấn công mới.

Giải pháp kiến trúc đa tác nhân được đưa ra năm 1994 là AAFID (các tác nhân tự trị cho việc phát hiện xâm phạm). Nó sử dụng các tác nhân để kiểm tra một khía cạnh nào đó về các hành vi hệ thống ở một thời điểm nào đó. Ví dụ: một tác nhân có thể cho biết một số không bình thường các telnet session bên trong hệ thống nó kiểm tra. Tác nhân có khả năng đưa ra một cảnh báo khi phát hiện một sự kiện khả nghi. Các tác nhân có thể được nhái và thay đổi bên trong các hệ thống khác (tính năng tự trị). Một phần trong các tác nhân, hệ thống có thể có các bộ phận thu phát để kiểm tra tất cả các hành động được kiểm soát bởi các tác nhân ở một host cụ thể nào đó. Các bộ thu nhận luôn luôn gửi các kết quả hoạt động của chúng đến bộ kiểm tra duy nhất. Các bộ kiểm tra nhận thông tin từ các mạng (không chủ từ một host), điều đó có nghĩa là chúng có thể tương quan với thông tin phân tán. Thêm vào đó một số bộ lọc có thể được đưa ra để chọn lọc và thu thập dữ liệu.



Hình 1-2. Giải pháp kiến trúc đa tác nhân

1.1.5. Phân loại hệ thống phát hiện xâm nhập

Có hai loại cơ bản là: *Network-based IDS* và *Host-based IDS*.

1.1.5.1. Network-based IDS (NIDS)

NIDS là một hệ thống phát hiện xâm nhập bằng cách thu thập dữ liệu của các gói tin lưu thông trên các phương tiện truyền dẫn như (cables, wireless) bằng cách sử dụng các card giao tiếp. Khi một gói dữ liệu phù hợp với qui tắc của hệ thống, một cảnh báo được tạo ra để thông báo đến nhà quản trị và các file log được lưu vào cơ sở dữ liệu.

a. Lợi thế của NIDS

- Quản lý được một phân đoạn mạng (network segment).
- Trong suốt với người sử dụng và kẻ tấn công.
- Cài đặt và bảo trì đơn giản, không làm ảnh hưởng đến mạng.
- Tránh được việc bị tấn công dịch vụ đến một host cụ thể.
- Có khả năng xác định được lỗi ở tầng network.
- Độc lập với hệ điều hành.

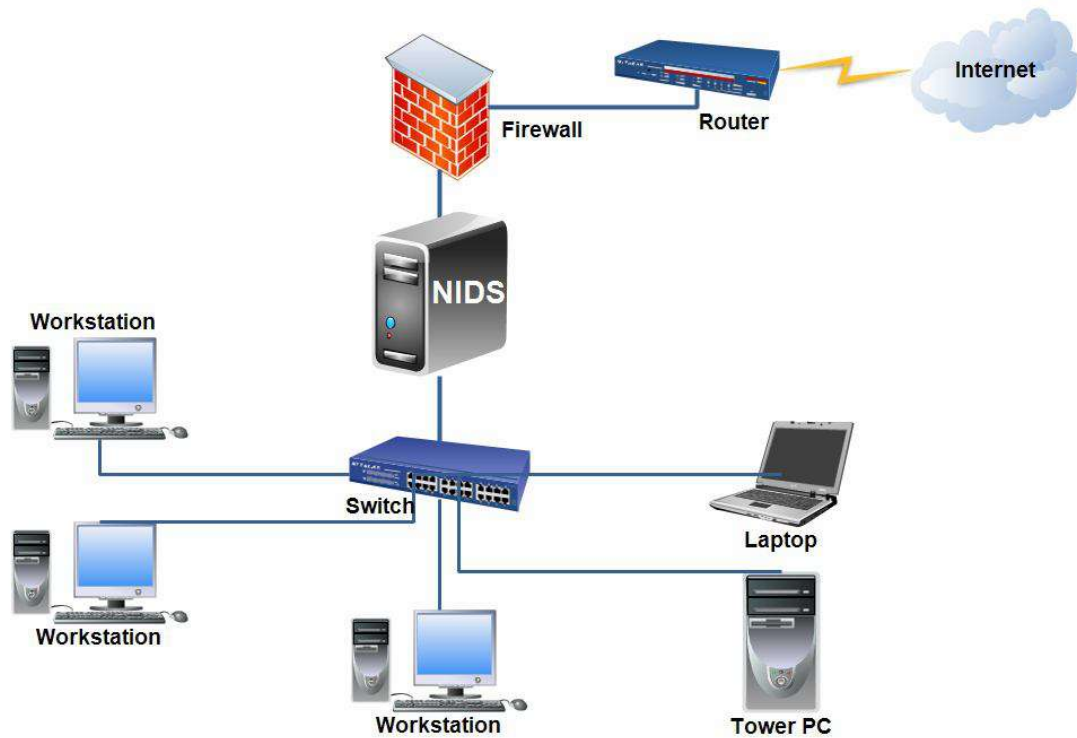
b. Hạn chế của NIDS

- Có thể xảy ra trường hợp báo động giả, tức là không có dấu hiệu bất thường mà IDS vẫn báo.
- Không thể phân tích được các lưu lượng đã được mã hóa như SSH, IPSec, SSL...

- NIDS đòi hỏi phải luôn được cập nhật các dấu hiệu tấn công mới nhất để thực sự hoạt động hiệu quả.
- Không thể cho biết việc mạng bị tấn công có thành công hay không, để người quản trị tiến hành bảo trì hệ thống.
- Một trong những hạn chế là giới hạn băng thông. Những bộ thu thập dữ liệu phải thu thập tất cả lưu lượng mạng, sắp xếp lại và phân tích chúng. Khi tốc độ mạng tăng lên thì khả năng của bộ thu thập thông tin cũng vậy. Một giải pháp là phải đảm bảo cho mạng được thiết kế chính xác.

Một cách mà hacker cố gắng che đậy cho hoạt động của họ khi gặp các hệ thống IDS là phân mảnh dữ liệu gói tin. Mỗi giao thức có một kích cỡ gói dữ liệu có hạn, nếu dữ liệu truyền qua mạng truyền qua mạng lớn hơn kích cỡ này thì dữ liệu bị phân mảnh. Phân mảnh đơn giản là quá trình chia nhỏ dữ liệu. Thứ tự sắp xếp không thành vấn đề miễn là không bị chồng chéo dữ liệu, bộ cảm biến phải tái hợp lại chúng.

Hacker cố gắng ngăn chặn phát hiện bằng cách gửi nhiều gói dữ liệu phân mảnh chồng chéo. Một bộ cảm biến không phát hiện được các hoạt động xâm nhập nếu không sắp xếp gói tin lại một cách chính xác.



Hình 1-3. Network-based IDS

1.1.5.2. Host-based IDS (HIDS)

HIDS là hệ thống phát hiện xâm nhập được cài đặt trên các máy tính (host). HIDS cài đặt trên nhiều kiểu máy chủ khác nhau, trên máy trạm làm việc hoặc máy notebook. HIDS cho phép thực hiện một cách linh hoạt trên các phân đoạn mạng mà NIDS không thực hiện được. Lưu lượng đã gửi đến host được phân tích và chuyển qua host nếu chúng không tiềm ẩn các mã nguy hiểm. HIDS cụ thể hơn với các nền ứng dụng và phục vụ mạnh mẽ cho hệ điều hành. Nhiệm vụ chính của HIDS là giám sát sự thay đổi trên hệ thống. HIDS bao gồm các thành phần chính:

- Các tiến trình.
- Các entry của registry.
- Mức độ sử dụng CPU.
- Kiểm tra tính toàn vẹn và truy cập trên file hệ thống.
- Một vài thông số khác.

Các thông số này vượt qua một ngưỡng định trước hoặc thay đổi khả nghi trên hệ thống sẽ gây ra cảnh báo.

a. Ưu điểm của HIDS

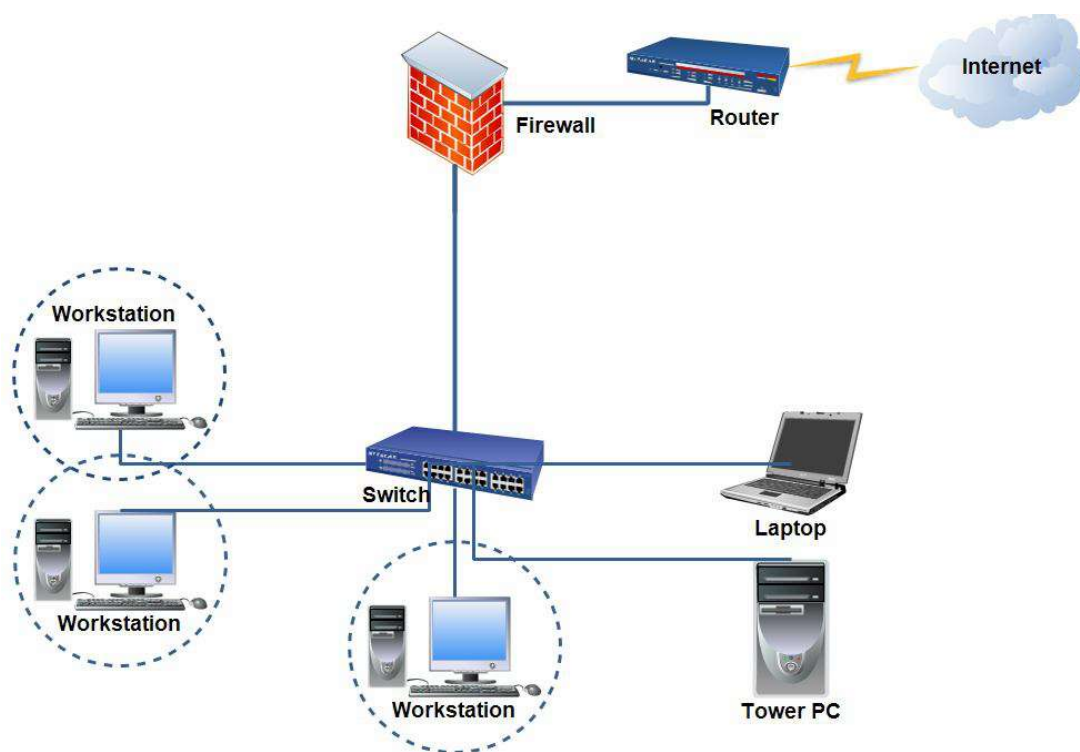
- Có khả năng xác định các user trong hệ thống liên quan đến sự kiện.
- HIDS có khả năng phát hiện các cuộc tấn công diễn ra trên một máy, NIDS không có khả năng này.
- Có khả năng phân tích các dữ liệu đã được mã hóa.
- Cung cấp các thông tin về host trong lúc cuộc tấn công đang diễn ra trên host.

b. Hạn chế của HIDS

- Thông tin từ HIDS sẽ không còn đáng tin cậy ngay sau khi cuộc tấn công vào host này thành công.
- Khi hệ điều hành bị thỏa hiệp tức là HIDS cũng mất tác dụng.
- HIDS phải được thiết lập trên từng host cần giám sát.
- HIDS không có khả năng phát hiện việc thăm dò mạng (Nmap, Netcat...).
- HIDS cần tài nguyên trên host để hoạt động.
- HIDS có thể không phát huy được hiệu quả khi bị tấn công từ chối dịch vụ DoS.
- Đa số được phát triển trên hệ điều hành Window. Tuy nhiên cũng có một số chạy trên Linux hoặc Unix.

Vì HIDS cần được cài đặt trên các máy chủ nên sẽ gây khó khăn cho nhà quản trị khi phải nâng cấp phiên bản, bảo trì phần mềm và cấu hình. Gây mất nhiều thời gian và phút tập. Thường hệ thống chỉ phân tích được những lưu lượng trên máy chủ nhận được, còn các lưu lượng chống lại một nhóm máy chủ, hoặc các hành động thăm dò như quét cổng thì chúng không phát huy được tác dụng. Nếu máy chủ bị thỏa hiệp hacker có thể tắt được HIDS trên máy đó. Khi đó HIDS sẽ bị vô hiệu hóa.

Do đó HIDS phải cung cấp đầy đủ khả năng cảnh báo. Trong môi trường hỗn tạp điều này có thể trở thành vấn đề nếu HIDS phải tương thích với nhiều hệ điều hành. Do đó, lựa chọn HIDS cũng là vấn đề quan trọng



Hình 1-4. Host-based IDS

1.1.5.3. So sánh giữa NIDS và HIDS

Bảng 1-1. So sánh, đánh giá giữa NIDS và HIDS

Chức năng	HIDS	NIDS	Các đánh giá
Bảo vệ trong mạng LAN	****	****	Cả hai đều bảo vệ khi user hoạt động khi trong mạng LAN
Bảo vệ ngoài mạng LAN	****	-	Chỉ có HIDS
Dễ dàng cho việc quản trị	****	****	Tương đương nhau xét về bối cảnh quản trị chung
Tính linh hoạt	****	**	HIDS là hệ thống linh hoạt hơn
Giá thành	***	*	HIDS là hệ thống ưu tiết kiệm hơn nếu chọn đúng sản phẩm
Dễ dàng trong việc bổ sung	****	****	Cả hai tương đương nhau

Đào tạo ngắn hạn cần thiết	****	**	HIDS yêu cầu việc đào tạo ít hơn NIDS
Tổng giá thành	***	**	HIDS tiêu tốn ít hơn
Băng tần cần yêu cầu trong LAN	0	2	NIDS sử dụng băng tần LAN rộng, còn HIDS thì không
Network overhead	1	2	NIDS cần 2 yêu cầu băng tần mạng đối với bất kỳ mạng LAN nào
Băng tần cần yêu cầu (Internet)	**	**	Cả hai đều cần băng tần Internet để cập nhật kịp thời các file mẫu
Các yêu cầu về cổng mở rộng	-	****	NIDS yêu cầu phải kích hoạt mở rộng cổng để đảm bảo lưu lượng LAN của bạn được quét
Chu kỳ nâng cấp cho các client	****	-	HIDS nâng cấp tất cả các client với một file mẫu trung tâm
Khả năng thích nghi trong các nền ứng dụng	**	****	NIDS có khả năng thích nghi trong các nền ứng dụng hơn
Chế độ quét thanh ghi cục bộ	****	-	Chỉ HIDS mới có thể thực hiện các kiểu quét này
Bản ghi	***	***	Cả hai hệ thống đều có chức năng bản ghi
Chức năng cảnh báo	***	***	Cả hai hệ thống đều có chức năng cảnh báo cho từng cá nhân và quản trị viên
Quét PAN	****	-	Chỉ có HIDS quét các vùng mạng cá nhân của bạn
Loại bỏ gói tin	-	****	Chỉ các tính năng NIDS mới có

			phương thức này
Kiến thức chuyên môn	***	****	Cần nhiều kiến thức chuyên môn khi cài đặt và sử dụng NIDS đối với toàn bộ vấn đề bảo mật mạng của bạn
Quản lý tập trung	**	***	NIDS có chiếm ưu thế hơn
Khả năng vô hiệu hóa các hệ số rủi ro	*	****	NIDS có hệ số rủi ro nhiều hơn so với HIDS
Khả năng cập nhật	***	***	Rõ ràng khả năng nâng cấp phần mềm là dễ hơn phần cứng. HIDS có thể được nâng cấp thông qua script được tập trung
Các nút phát hiện nhiều đoạn mạng LAN	****	**	HIDS có khả năng phát hiện theo nhiều đoạn mạng toàn diện hơn

1.2. HỆ THỐNG NGĂN CHẶN XÂM NHẬP

1.2.1. Khái niệm

Hệ thống ngăn chặn xâm nhập IPS là một kỹ thuật an ninh mới, kết hợp các ưu điểm của kỹ thuật firewall và hệ thống phát hiện xâm nhập IDS. Có khả năng phát hiện các cuộc tấn công và tự động ngăn chặn các cuộc tấn công đó.

IPS không đơn giản là dò các cuộc tấn công, chúng có khả năng ngăn chặn hoặc cản trở các cuộc tấn công đó. Chúng cho phép tổ chức ưu tiên, thực hiện các bước để ngăn chặn tấn công. Phần lớn các hệ thống IPS được đặt ở vành đai mạng, đủ khả năng bảo vệ tất cả các thiết bị trong mạng.

1.2.2. Kiến trúc của hệ thống ngăn chặn xâm nhập

Một hệ thống IPS gồm có 3 module chính:

- Module phân tích gói tin.
- Module phát hiện tấn công.
- Module phản ứng.

1.2.2.1 Module phân tích gói tin

Module này có nhiệm vụ phân tích cấu trúc thông tin của gói tin. NIC Card của máy tính được giám sát được đặt ở chế độ *promiscuous mode*, tất cả các gói tin qua chúng đều được sao chép lại và chuyển lên lớp trên. Bộ phân tích gói tin đọc thông tin từng trường trong gói tin, xác định chúng thuộc kiểu gói tin gì, dịch vụ gì, sử dụng loại giao thức nào... Các thông tin này được chuyển lên module phát hiện tấn công.

1.2.2.2 Module phát hiện tấn công

Đây là module quan trọng nhất của hệ thống phát hiện xâm nhập, có khả năng phát hiện ra các cuộc tấn công. Có một số phương pháp để phát hiện ra các dấu hiệu xâm nhập hoặc các kiểu tấn công (signature-based IPS, anomaly-based IPS,...).

a. Phương pháp dò sự lạm dụng:

Phương pháp này phân tích các hoạt động của hệ thống, tìm kiếm các sự kiện giống với các mẫu tấn công đã biết trước. Các mẫu tấn công này được gọi là dấu hiệu tấn công. Do vậy phương pháp này còn gọi là phương pháp dò dấu hiệu.

Phương pháp này có ưu điểm là phát hiện các cuộc tấn công nhanh và chính xác, không đưa ra các cảnh báo sai dẫn đến làm giảm khả năng hoạt động của mạng và giúp cho người quản trị xác định các lỗ hổng bảo mật trong hệ thống của mình. Tuy nhiên, phương pháp này có nhược điểm là không phát hiện được các cuộc tấn công không có trong cơ sở dữ liệu, các kiểu tấn công mới, do vậy hệ thống phải luôn luôn cập nhật các kiểu tấn công mới.

b. Phương pháp dò sự không bình thường:

Đây là kỹ thuật dò thông minh, nhận dạng ra các hành động không bình thường của mạng. Quan niệm của phương pháp này về các cuộc tấn công là khác với các hoạt động bình thường.

Ban đầu chúng sẽ lưu trữ các mô tả sơ lược về các hoạt động bình thường của hệ thống. Các cuộc tấn công sẽ có những hành động khác so với bình thường và phương pháp này có thể nhận dạng ra. Có một số kỹ thuật dò sự không bình thường của các cuộc tấn công.

- **Phát hiện mức ngưỡng:**

Kỹ thuật này nhấn mạnh việc đo đếm các hoạt động bình thường trên mạng. Các mức ngưỡng về các hoạt động bình thường được đặt ra. Nếu có sự bất thường nào đó, ví dụ như đăng nhập vào hệ thống quá số lần qui định, số lượng các tiến trình hoạt động trên CPU, số lượng một loại gói tin được gửi quá mức...Thì hệ thống cho rằng có dấu hiệu của sự tấn công.

- **Phát hiện nhờ quá trình tự học:**

Kỹ thuật này bao gồm 2 bước, khi bắt đầu thiết lập hệ thống phát hiện tấn công sẽ chạy ở chế độ tự học và tạo hồ sơ về cách cư xử của mạng với các hoạt động bình thường. Sau thời gian khởi tạo, hệ thống sẽ chạy ở chế độ làm việc, tiến hành theo dõi, phát hiện các hoạt động bất thường của mạng bằng cách so sánh với hồ sơ đã được tạo.

Chế độ tự học có thể chạy song song với chế độ làm việc để cập nhật hồ sơ của mình nhưng nếu dò ra các dấu hiệu tấn công thì chế độ tự học phải ngừng lại cho đến khi cuộc tấn công kết thúc

- **Phát hiện sự không bình thường của giao thức:**

Kỹ thuật này căn cứ vào hoạt động của các giao thức, các dịch vụ của hệ thống để tìm ra các gói tin không hợp lệ, các hoạt động bất thường vốn là dấu hiệu của sự xâm nhập. Kỹ thuật này rất hiệu quả trong việc ngăn chặn các hình thức quét mạng, quét cổng để thu thập thông tin hệ thống của hacker.

Phương pháp dò sự không bình thường của hệ thống rất hữu hiệu trong việc phát hiện các kiểu tấn công từ chối dịch vụ DoS. Ưu điểm của phương pháp này là có thể phát hiện các kiểu tấn công mới, cung cấp thông tin hữu ích bổ sung cho phương pháp dò sự lạm dụng. Tuy nhiên, chúng có nhược điểm là thường gây ra các cảnh báo sai lầm giảm hiệu suất hoạt động của mạng.

1.2.2.3 Module phản ứng

Khi có dấu hiệu của sự tấn công hoặc xâm nhập, module phát hiện tấn công sẽ gửi tín hiệu báo hiệu có sự tấn công hoặc xâm nhập đến module phản ứng. Lúc đó module phản ứng sẽ kích hoạt firewall thực hiện chức năng ngăn chặn cuộc tấn công. Tại module này, nếu chỉ đưa ra các cảnh báo tới các người quản trị và dừng lại ở đó thì hệ thống này được gọi là hệ thống phòng thủ bị động. Module phản ứng

này tùy theo hệ thống mà có các chức năng khác nhau. Dưới đây là một số kỹ thuật ngăn chặn:

- **Terminate session:**

Cơ chế của kỹ thuật này là hệ thống IPS gửi gói tin reset, thiết lập lại cuộc giao tiếp tới cả client và server. Kết quả cuộc giao tiếp sẽ được bắt đầu lại, các mục đích của hacker không đạt được, cuộc tấn công bị ngừng lại. Tuy nhiên phương pháp này có một số nhược điểm như thời gian gửi gói tin reset đến đích là quá lâu so với thời gian gói tin của hacker đến được Victim, dẫn đến reset quá chậm so với cuộc tấn công, phương pháp này không hiệu ứng với các giao thức hoạt động trên UDP như DNS, ngoài ra gói Reset phải có trường sequence number đúng (so với gói tin trước đó từ client) thì server mới chấp nhận, do vậy nếu hacker gửi các gói tin với tốc độ nhanh và trường sequence number thay đổi thì rất khó thực hiện được phương pháp này.

- **Drop attack:**

Kỹ thuật này dùng firewall để hủy bỏ gói tin hoặc chặn đường một gói tin đơn, một phiên làm việc hoặc một luồng thông tin giữa hacker và victim. Kiểu phản ứng này là an toàn nhất nhưng lại có nhược điểm là dễ nhầm với các gói tin hợp lệ.

- **Modify firewall polices:**

Kỹ thuật này cho phép người quản trị cấu hình lại chính sách bảo mật khi cuộc tấn công xảy ra. Sự cấu hình lại là tạm thời thay đổi các chính sách điều khiển truy cập bởi người dùng đặc biệt trong khi cảnh báo tới người quản trị.

- **Real-time Alerting:**

Gửi các cảnh báo thời gian thực đến người quản trị để họ nắm được chi tiết các cuộc tấn công, các đặc điểm và thông tin về chúng.

- **Log packet:**

Các dữ liệu của các gói tin sẽ được lưu trữ trong hệ thống các file log. Mục đích để các người quản trị có thể theo dõi các luồng thông tin và là nguồn thông tin giúp cho module phát hiện tấn công hoạt động.

Ba module trên hoạt động theo tuần tự tạo nên hệ thống IPS hoàn chỉnh. Một hệ thống IPS được xem là thành công nếu chúng hội tụ được các yếu tố: thực hiện

nhANH, chính xác, đưa ra các thông báo hợp lý, phân tích được toàn bộ thông lượng, cảm biến tối đa, ngăn chặn thành công và chính sách quản lý mềm dẻo.

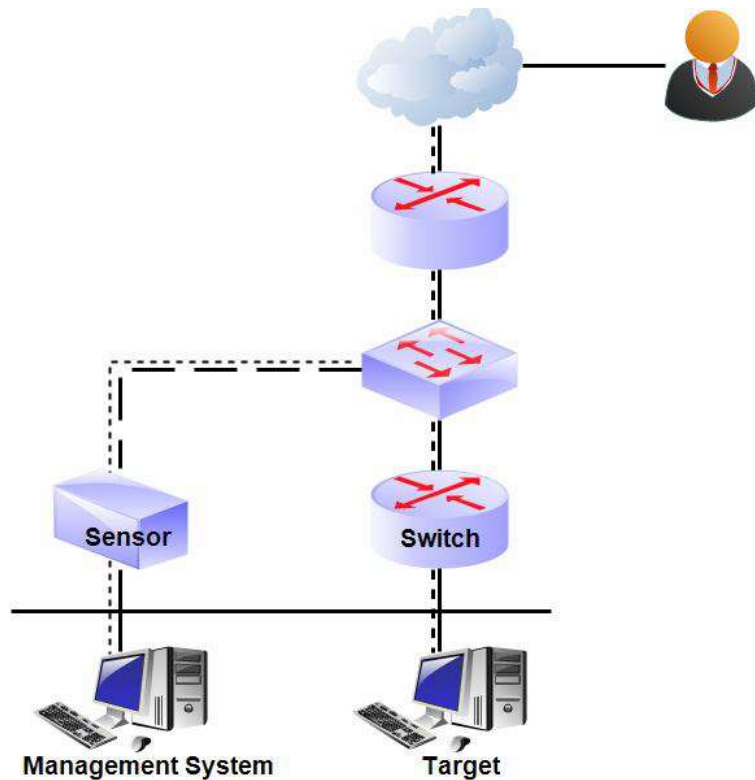
Các kiểu tấn công mới ngày càng phát triển đe dọa đến sự an toàn của các hệ thống mạng. Với các ưu điểm của mình, hệ thống IPS dần trở thành không thể thiếu trong các hệ thống bảo mật.

1.2.3. Các kiểu IPS được triển khai trên thực tế

Trên thực tế có 2 kiểu IPS được triển khai là: *Promiscuous mode IPS* và *In-line IPS*.

1.2.3.1 Promiscuous mode IPS

Một IPS đứng trên firewall. Như vậy luồng dữ liệu vào hệ thống mạng sẽ cùng đi qua firewall và IPS. IPS có thể kiểm soát luồng dữ liệu vào, phân tích và phát hiện các dấu hiệu xâm nhập, tấn công. Với vị trí này, *promiscuous mode IPS* có thể quản lý firewall, chỉ dẫn firewall ngăn chặn các hành động đáng ngờ.

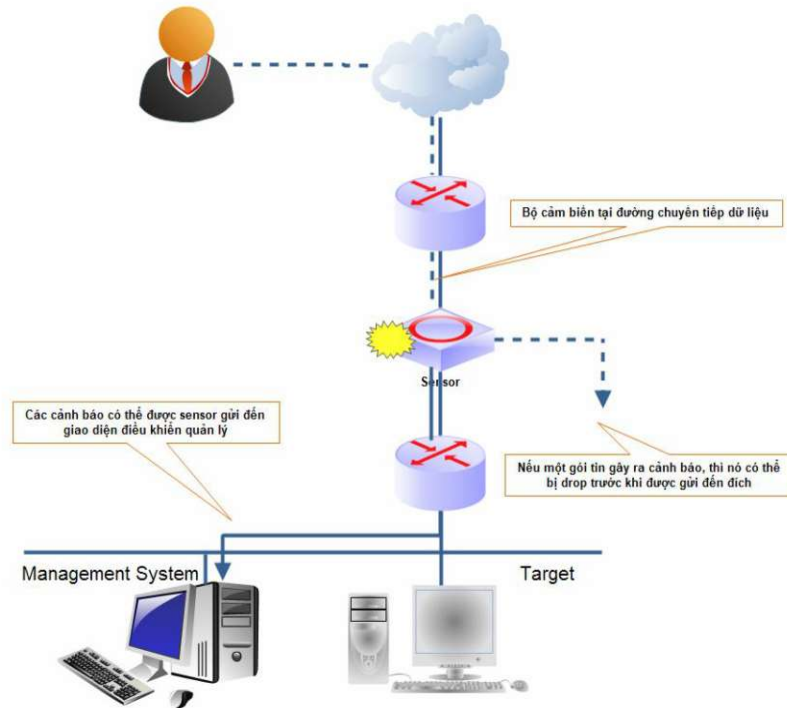


Hình 1-5. Promiscuous mode IPS

1.2.3.2. In-line mode IPS

Vị trí IPS đặt trước firewall, luồng dữ liệu phải đi qua chúng trước khi đến được firewall. Điểm khác chính so với *promiscuous mode IPS* là có thêm chức năng *traffic-blocking*. Điều này làm cho IPS có thể ngăn chặn luồng giao thông nguy hiểm nhanh hơn *promiscuous mode IPS* nhanh hơn. Tuy nhiên khi đặt ở vị trí này làm cho tốc độ luồng thông tin ra vào mạng chậm hơn.

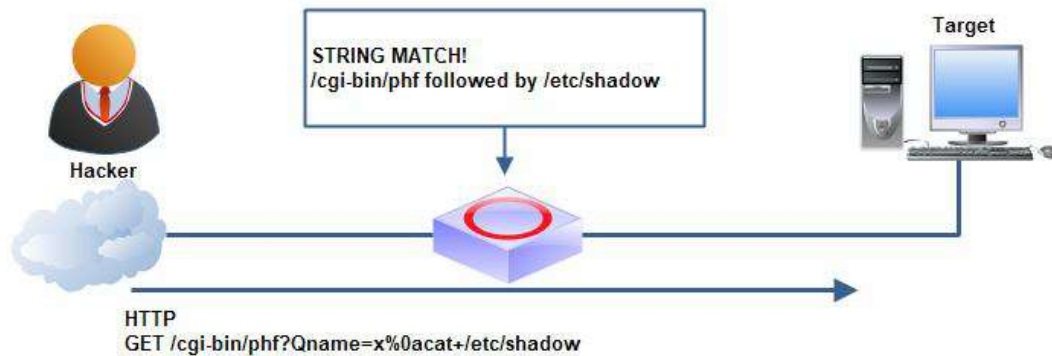
Với mục tiêu ngăn chặn các cuộc tấn công, hệ thống IPS phải hoạt động theo thời gian thực. Tốc độ hoạt động của hệ thống là một yếu tố vô cùng quan trọng. Quá trình phát hiện xâm nhập phải đủ nhanh để có thể ngăn chặn các cuộc tấn công ngay tức thì. Nếu không đáp ứng được điều này thì các cuộc tấn công đã thực hiện xong. Hệ thống IPS trở nên vô tác dụng.



Hình 1-6. Inline mode IPS

1.2.4. Công nghệ ngăn chặn xâm nhập của IPS

1.2.4.1. Signature-based IPS



Hình 1-7. Signature-based IPS

Là tạo ra các rule gắn liền với những hoạt động xâm nhập tiêu biểu. Việc tạo ra các signature-based yêu cầu người quản trị phải thật rõ các kỹ thuật tấn công, những mối nguy hại và cần phải biết phát triển những signature để có thể dò tìm những cuộc tấn công và các mối nguy hại cho hệ thống của mình.

Signature-based IPS giám sát tất cả các traffic và so sánh với dữ liệu hiện có. Nếu không có sẽ đưa ra những cảnh báo cho người quản trị biết về cuộc tấn công đó. Để xác định được một dấu hiệu tấn công thì cần phải biết cấu trúc của kiểu tấn công, signature-based IPS sẽ xem header của gói tin hoặc phần payload của dữ liệu.

Một signature-based là một tập những nguyên tắc sử dụng để xác định những hoạt động xâm nhập thông thường. Những nghiên cứu về những kỹ thuật nhằm tìm ra dấu hiệu tấn công, những mẫu và phương pháp để viết ra các dấu hiệu tấn công. Khi càng nhiều phương pháp tấn công và phương pháp khai thác được khám phá, những nhà sản xuất cung cấp bản cập nhật file dấu hiệu. Khi đã cập nhật file dấu hiệu thì hệ thống IPS có thể phân tích tất cả lưu lượng trên mạng. Nếu có dấu hiệu nào trùng với file dấu hiệu thì các cảnh báo được khởi tạo

a. Lợi ích của việc dùng Signature-Based IPS:

Những file dấu hiệu được tạo nên từ những hoạt động và phương pháp tấn công đã được biết, do đó nếu có sự trùng lặp thì xác suất xảy ra một cuộc tấn công là rất cao. Phát hiện sử dụng sai sẽ có ít cảnh báo nhầm (false positive report) hơn kiểu phát hiện sự bất thường. Phát hiện dựa trên dấu hiệu không theo dõi những mẫu lưu lượng hay tìm kiếm những sự bất thường. Thay vào đó nó theo dõi những

hoạt động đơn giản để tìm sự tương xứng đối với bất kỳ dấu hiệu nào đã được định dạng.

Bởi vì phương pháp phát hiện sử dụng sai dựa trên những dấu hiệu, không phải những mẫu lưu lượng. Hệ thống IPS có thể được định dạng và có thể bắt đầu bảo vệ mạng ngay lập tức. Những dấu hiệu trong cơ sở dữ liệu chứa những hoạt động xâm nhập đã biết và bản mô tả của những dấu hiệu này. Mỗi dấu hiệu trong cơ sở dữ liệu có thể được thấy cho phép, không cho phép những mức độ cảnh báo khác nhau cũng như những hành động ngăn cản khác nhau, có thể được định dạng cho những dấu hiệu riêng biệt. Phát hiện sử dụng sai dễ hiểu cũng như dễ định dạng hơn những hệ thống phát hiện sự bất thường .

File dấu hiệu có thể dễ dàng được người quản trị thấy và hiểu hành động nào phải được tương xứng cho một tín hiệu cảnh báo. Người quản trị bảo mật có thể có thể bật những dấu hiệu lên, sau đó họ thực hiện cuộc kiểm tra trên toàn mạng và xem xem có cảnh báo nào không.

Chính vì phát hiện sử dụng sai dễ hiểu ,bổ sung, kiểm tra, do đó nhà quản trị có những khả năng to lớn trong việc điều khiển cũng như tự tin vào hệ thống IPS của họ.

b. Những hạn chế của Signature-Based IPS:

Bên cạnh những lợi điểm của cơ chế phát hiện sử dụng sai thì nó cũng tồn tại nhiều hạn chế. Phát hiện sử dụng sai dễ dàng hơn trong định dạng và hiểu, nhưng chính sự giản đơn này trở thành cái giá phải trả cho sự mất mát những chức năng và overhead. Đây là những hạn chế:

- Không có khả năng phát hiện những cuộc tấn công mới hay chưa được biết : Hệ thống IPS sử dụng phát hiện sử dụng sai phải biết trước những hoạt động tấn công để nó có thể nhận ra đợt tấn công đó. Những dạng tấn công mới mà chưa từng được biết hay khám phá trước đây thường sẽ không bị phát hiện.
- Không có khả năng phát hiện những sự thay đổi của những cuộc tấn công đã biết : Những file dấu hiệu là những file tĩnh tức là chúng không thích nghi với một vài hệ thống dựa trên sự bất thường. Bằng cách thay đổi cách tấn công, một kẻ xâm nhập có thể thực hiện cuộc xâm nhập mà không bị phát hiện(false negative).

Khả năng quản trị cơ sở dữ liệu những dấu hiệu : Trách nhiệm của nhà quản trị bảo mật là bảo đảm file cơ sở dữ liệu luôn cập nhật và hiện hành. Đây là công việc mất nhiều thời gian cũng như khó khăn.

Những bộ cảm biến phải duy trì tình trạng thông tin : Giống như firewall, bộ cảm biến phải duy trì trạng thái dữ liệu. Hầu hết những bộ cảm biến giữ trạng thái thông tin trong bộ nhớ để tìm lại nhanh hơn, nhưng mà khoảng trống thì giới hạn.

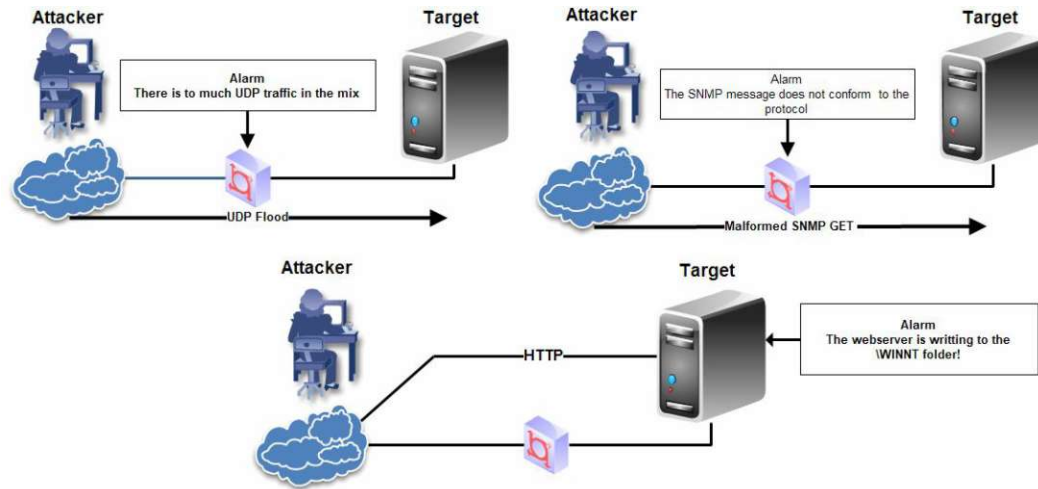
1.2.4.2. Anomaly-based IPS

Phát hiện dựa trên sự bất thường hay mô tả sơ lược phân tích những hoạt động của mạng máy tính và lưu lượng mạng nhằm tìm kiếm sự bất thường. Khi tìm thấy sự bất thường, một tín hiệu cảnh báo sẽ được khởi phát. Sự bất thường là bất cứ sự chệch hướng hay đi khỏi những thứ tự, dạng, nguyên tắc thông thường. Chính vì dạng phát hiện này tìm kiếm những bất thường nên nhà quản trị bảo mật phải định nghĩa đâu là những hoạt động, lưu lượng bất thường. Nhà quản trị bảo mật có thể định nghĩa những hoạt động bình thường bằng cách tạo ra những bản mô tả sơ lược nhóm người dùng (user group profiles).

Bản mô tả sơ lược nhóm người dùng thể hiện ranh giới giữa những hoạt động cũng như những lưu lượng mạng trên một nhóm người dùng cho trước. Những nhóm người dùng được định nghĩa bởi kỹ sư bảo mật và được dùng để thể hiện những chức năng công việc chung. Một cách điển hình, những nhóm sử dụng nên được chia theo những hoạt động cũng như những nguồn tài nguyên mà nhóm đó sử dụng.

Một web server phải có bản mô tả sơ lược của nó dựa trên lưu lượng web, tương tự như vậy đối với mail server. Bạn chắc chắn không mong đợi lưu lượng telnet với web server của mình cũng như không muốn lưu lượng SSH đến với mail server. Chính vì lý do này mà bạn nên có nhiều bản mô tả sơ lược khác nhau cho mỗi dạng dịch vụ có trên mạng của bạn. Đa dạng những kỹ thuật được sử dụng để xây dựng những bản mô tả sơ lược người dùng và nhiều hệ thống IPS có thể được định dạng để xây dựng những profile của chúng. Những phương pháp điển hình nhằm xây dựng bản mô tả sơ lược nhóm người dùng là lấy mẫu thống kê (statistical sampling), dựa trên những nguyên tắc và những mạng neural.

Mỗi profile được sử dụng như là định nghĩa cho người sử dụng thông thường và hoạt động mạng. Nếu một người sử dụng làm chệch quá xa những gì họ đã định nghĩa trong profile, hệ thống IPS sẽ phát sinh cảnh báo.



Hình 1-8. Anomaly-Based IPS

a. Lợi ích của việc dùng Anomaly-Based IPS

Với phương pháp này, kẻ xâm nhập không bao giờ biết lúc nào có, lúc nào không phát sinh cảnh báo bởi vì họ không có quyền truy cập vào những profile sử dụng để phát hiện những cuộc tấn công.

Những profile nhóm người dùng rất giống cơ sở dữ liệu dấu hiệu động luôn thay đổi khi mạng của bạn thay đổi. Với phương pháp dựa trên những dấu hiệu, kẻ xâm nhập có thể kiểm tra trên hệ thống IPS của họ cái gì làm phát sinh tín hiệu cảnh báo.

File dấu hiệu được cung cấp kèm theo với hệ thống IPS, vì thế kẻ xâm nhập có thể sử dụng hệ thống IPS đó để thực hiện kiểm tra. Một khi kẻ xâm nhập hiểu cái gì tạo ra cảnh báo thì họ có thể thay đổi phương pháp tấn công cũng như công cụ tấn công để đánh bại hệ IPS.

Chính vì phát hiện bất thường không sử dụng những cơ sở dữ liệu dấu hiệu định dạng trước nên kẻ xâm nhập không thể biết chính xác cái gì gây ra cảnh báo. Phát hiện bất thường có thể nhanh chóng phát hiện một cuộc tấn công từ bên trong sử dụng tài khoản người dùng bị thỏa hiệp (compromised user account).

Nếu tài khoản người dùng là sở hữu của một phụ tá quản trị đang được sử dụng để thi hành quản trị hệ thống, hệ IPS sử dụng phát hiện bất thường sẽ gây ra

một cảnh báo miễn là tài khoản đó không được sử dụng để quản trị hệ thống một cách bình thường.

Ưu điểm lớn nhất của phát hiện dựa trên profile hay sự bất thường là nó không dựa trên một tập những dấu hiệu đã được định dạng hay những đợt tấn công đã được biết profile có thể là động và có thể sử dụng trí tuệ nhân tạo để xác định những hoạt động bình thường.

Bởi vì phát hiện dựa trên profile không dựa trên những dấu hiệu đã biết, nó thực sự phù hợp cho việc phát hiện những cuộc tấn công chưa hề được biết trước đây miễn là nó chệch khỏi profile bình thường. Phát hiện dựa trên profile được sử dụng để phát hiện những phương pháp tấn công mới mà phát hiện bằng dấu hiệu không phát hiện được.

b. Hạn chế của việc dùng Anomaly-Based IPS:

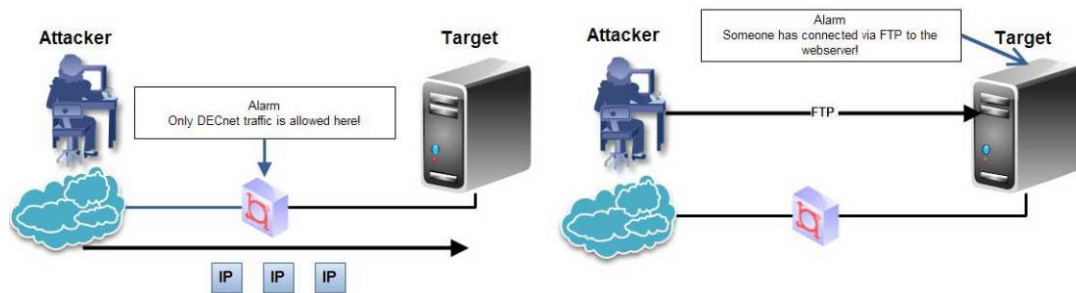
Nhiều hạn chế của phương pháp phát hiện bất thường phải làm với việc sáng tạo những profile nhóm người dùng, cũng như chất lượng của những profile này.

- Thời gian chuẩn bị ban đầu cao.
- Không có sự bảo vệ trong suốt thời gian khởi tạo ban đầu.
- Thường xuyên cập nhật profile khi thói quen người dùng thay đổi.

Khó khăn trong việc định nghĩa cách hành động thông thường : Hệ thống IPS chỉ thật sự tốt khi nó định nghĩa những hành động nào là bình thường. Định nghĩa những hoạt động bình thường thậm chí còn là thử thách khi mà môi trường nơi mà công việc của người dùng hay những trách nhiệm thay đổi thường xuyên.

- Cảnh báo nhầm: Những hệ thống dựa trên sự bất thường có xu hướng có nhiều false positive bởi vì chúng thường tìm những điều khác thường.
- Khó hiểu : Hạn chế cuối cùng của phương pháp phát hiện dựa trên sự bất thường là sự phức tạp. Lấy mẫu thống kê, dựa trên nguyên tắc, và mạng neural là những phương cách nhằm tạo profile mà thật khó hiểu và giải thích.

1.2.4.3. Policy-Based IPS



Hình 1-9 Policy-Based IPS

Một Policy-Based IPS nó sẽ phản ứng hoặc có những hành động nếu có sự vi phạm của một cấu hình policy xảy ra. Bởi vậy, một Policy-Based IPS cung cấp một hoặc nhiều phương thức được ưu chuộng để ngăn chặn.

a. Lợi ích của việc dùng Policy-Based IPS.

- Ta có thể policy cho từng thiết bị một trong hệ thống mạng.
- Một trong những tính năng quan trọng của *Policy-Based IPS* là xác thực và phản ứng nhanh, rất ít có những cảnh báo sai. Đây là những lợi ích có thể chấp nhận được bởi vì người quản trị hệ thống đưa các security policy tới IPS một cách chính xác nó là gì và nó có được cho phép hay không?

b. Hạn chế của việc dùng Policy-Based IPS.

- Khi đó công việc của người quản trị cực kỳ vất vả.
- Khi một thiết bị mới được thêm vào trong mạng thì lại phải cấu hình.
- Khó khăn khi quản trị từ xa.

1.2.4.4. Protocol Analysis-Based IPS

Giải pháp phân tích giao thức(Protocol Analysis-Based IPS) về việc chống xâm nhập thì cũng tương tự như Signature-Based IPS, nhưng nó sẽ đi sâu hơn về việc phân tích các giao thức trong gói tin (packet). Ví dụ: Một hacker bắt đầu chạy một chương trình tấn công tới một Server. Trước tiên hacker phải gửi một gói tin IP cùng với kiểu giao thức, theo một RFC, có thể không chứa dữ liệu trong payload. Một Protocol Analysis-Based sẽ phát hiện ra kiểu tấn công cơ bản trên một số giao thức.

- Kiểm tra khả năng của giao thức để xác định gói tin đó có hợp pháp hay không?
- Kiểm tra nội dung trong Payload (pattern matching).
- Thực hiện những cảnh báo không bình thường.

1.3. SO SÁNH GIỮA HỆ THỐNG IDS VÀ IPS

Ở mức cơ bản nhất, IDS khá thụ động, theo dõi dữ liệu của packet đi qua mạng từ một port giám sát, so sánh các traffic này đến các rules được thiết lập và đưa ra các cảnh báo nếu phát hiện bất kỳ dấu hiệu bất thường nào. Một hệ thống IDS có thể phát hiện hầu hết các loại traffic độc hại đã bị tường lửa để trượt, bao gồm các cuộc tấn công từ chối dịch vụ, tấn công dữ liệu trên các ứng dụng, đăng nhập trái phép máy chủ, và các phần mềm độc hại như virus, Trojan, và worms.

Hầu hết các hệ thống IDS sử dụng một số phương pháp để phát hiện ra các mối đe dọa, thường dựa trên dấu hiệu xâm nhập và phân tích trạng thái của giao thức.

IDS lưu các file log vào CSDL và tạo ra các cảnh báo đến người quản trị. IDS cho tầm nhìn sâu với các hoạt động mạng, nên nó giúp xác định các vấn đề với chính sách an ninh của một tổ chức.

Vấn đề chính của IDS là thường đưa ra các báo động giả. Cần phải tối đa hóa tính chính xác trong việc phát hiện ra các dấu hiệu bất thường .

1.3.1. Lợi thế của IPS

Ở mức cơ bản nhất, IPS có tất cả tính năng của hệ thống IDS. Ngoài ra nó còn ngăn chặn các luồng lưu lượng gây nguy hại đến hệ thống. Nó có thể chấm dứt sự kết nối mạng của kẻ đang cố gắng tấn công vào hệ thống, bằng cách chặn tài khoản người dùng, địa chỉ IP, hoặc các thuộc tính liên kết đến kẻ tấn công. Hoặc chặn tất cả các truy cập vào máy chủ, dịch vụ, ứng dụng.

Ngoài ra, một IPS có thể phản ứng với các mối đe dọa theo hai cách. Nó có thể cấu hình lại các điều khiển bảo mật khác như router hoặc firewall, để chặn đứng các cuộc tấn công. Một số IPS thậm chí còn áp dụng các bản vá lỗi nếu máy chủ có lỗ hổng. Ngoài ra, một số IPS có thể loại bỏ các nội dung độc hại từ cuộc tấn công, như xóa các tệp tin đính kèm với mail của user mà chứa nội dung nguy hiểm đến hệ thống.

1.3.2. Bảo vệ hai lần

Bởi vì IDS và IPS được đặt ở các vị trí khác nhau trên mạng. Chúng nên được sử dụng đồng thời. Một hệ thống IPS đặt bên ngoài mạng sẽ ngăn chặn được các cuộc tấn công zero day, như là virus hoặc worm. Ngay cả các mối đe dọa mới nhất cũng có thể được ngăn chặn. Một IDS đặt bên trong mạng sẽ giám sát được các hoạt động nội bộ.

CHƯƠNG 2

SNORT VÀ IPTABLES TRÊN HỆ ĐIỀU HÀNH LINUX

Có hai cách phổ biến để bảo vệ hệ thống mạng là firewall và hệ thống phát hiện xâm nhập IDS. Tuy nhiên chúng mang lại hiệu quả không cao khi hoạt động độc lập. Sự kết hợp giữa hệ thống phát hiện xâm nhập Snort (Snort_inline) và iptables firewall của hệ điều hành Linux thực sự mang lại hiệu quả cao trong việc phát hiện và ngăn chặn các cuộc tấn công trái phép đến hệ thống mạng. Chương này sẽ giới thiệu về hệ thống phát hiện xâm nhập Snort và iptables firewall của Linux và sự kết hợp của chúng để xây dựng nên một hệ thống IPS hoàn chỉnh.

2.1. TỔNG QUAN VỀ SNORT

2.1.1. Giới thiệu về Snort

Snort là một hệ thống ngăn chặn xâm nhập và phát hiện xâm nhập mã nguồn mở được phát triển bởi sourcefire. Kết hợp những lợi ích của dấu hiệu, giao thức và dấu hiệu bất thường, Snort là công nghệ IDS/IPS được triển khai rộng rãi trên toàn thế giới.

Snort là một ứng dụng bảo mật hiện đại có ba chức năng chính: nó có thể phục vụ như một bộ phận lắng nghe gói tin, lưu lại thông tin gói tin hoặc một hệ thống phát hiện xâm nhập mạng (NIDS). Bên cạnh đó có rất nhiều add-on cho Snort để quản lý (ghi log, quản lý, tạo rules...). Tuy không phải là phần lõi của Snort nhưng các thành phần này đóng vai trò quan trọng trong việc sử dụng cũng như khai thác các tính năng của Snort.

Thông thường, Snort chỉ nói chuyện với TCP/IP. Mặc dù, với các phần tùy chỉnh mở rộng, Snort có thể thực hiện để hỗ trợ các giao thức mạng khác, chẳng hạn như Novell's IPX. TCP/IP là một giao thức phổ biến của Internet. Do đó, Snort chủ yếu phân tích và cảnh báo trên giao thức TCP/IP.

2.1.2. Các yêu cầu với hệ thống Snort

2.1.2.1. Qui mô của hệ thống mạng cần bảo vệ

Nói một cách tổng quát, qui mô mạng càng lớn, các máy móc cần phải tốt hơn ví dụ như các Snort sensor. Snort cần có thể theo kịp với qui mô của mạng, cần

có đủ không gian để chứa các cảnh báo, các bộ xử lý đủ nhanh và đủ bộ nhớ để xử lý những luồng lưu lượng mạng.

2.1.2.2. Phần cứng máy tính

Yêu cầu phần cứng đóng một vai trò thiết yếu trong việc thiết kế một hệ thống an ninh tốt.

2.1.2.3. Hệ điều hành

Snort chạy trên nhiều hệ điều hành khác nhau như: Linux, FreeBSD, NetBSD, OpenBSD, và Window. Các hệ thống khác được hỗ trợ bao gồm kiến trúc Sparc-Solaris, MacOS X và MkLinux, và PA-RISC HP UX.

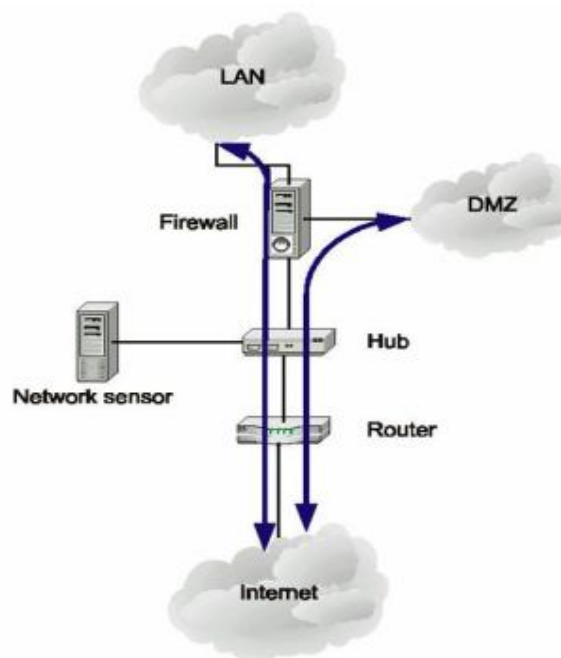
2.1.2.4. Các phần mềm hỗ trợ khác

Ngoài các hệ điều hành cơ bản, một số công cụ cơ bản giúp biên dịch Snort như: *autoconf and automake, gcc, lex and yacc, or the GNU equivalents flex and bison, libpcap*.

Một số công cụ giúp quản lý Snort như công cụ phân tích console phổ biến cho hệ thống phát hiện (ACID) có giao diện web. Một số công cụ phổ biến như: *ACID, Oinkmaster, SnortSnart, SnortResport*.

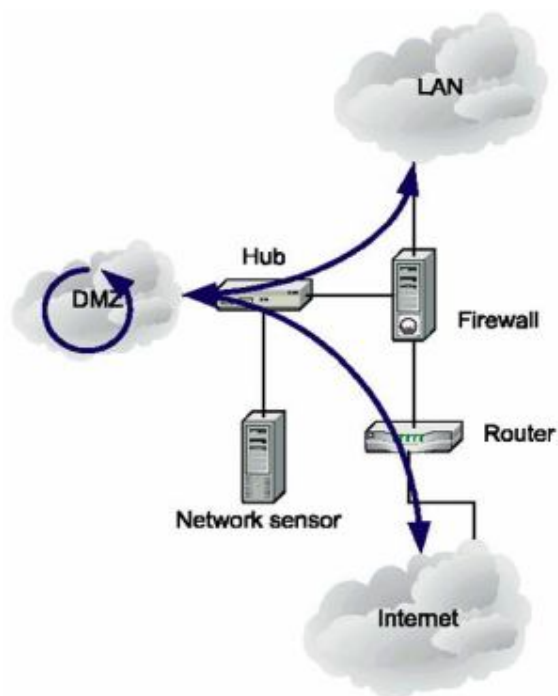
2.1.3. Vị trí của Snort trong hệ thống mạng

2.1.3.1. Giữa Router và firewall



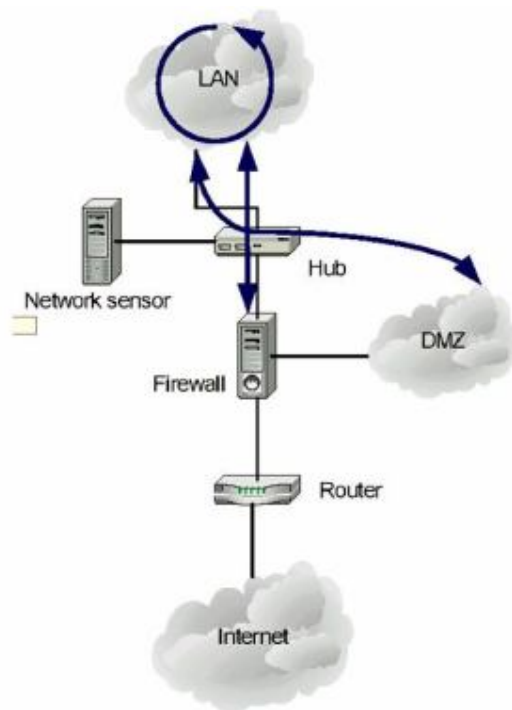
Hình 2-1. Snort-sensor đặt giữa Router và firewall

2.1.3.2. Trong vùng DMZ



Hình 2-2. Snort-sensor đặt trong vùng DMZ

2.1.3.3. Sau firewall



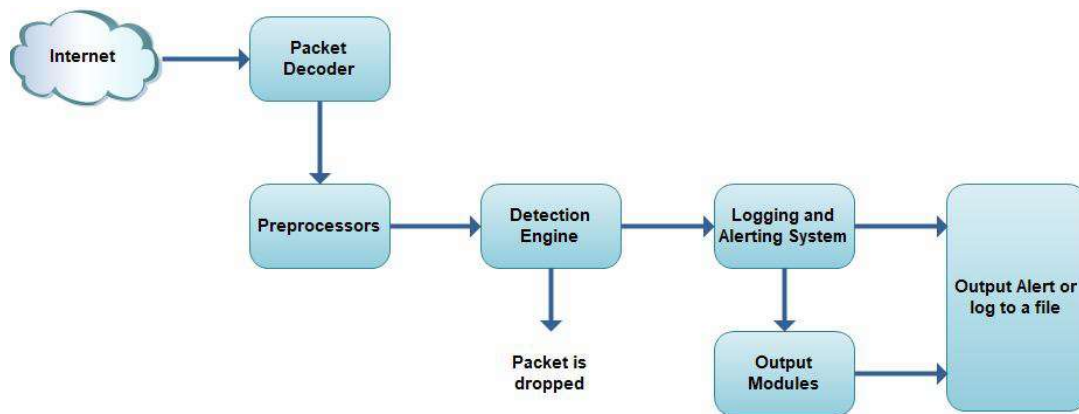
Hình 2-3. Snort-sensor đặt sau firewall

2.1.4. Các thành phần của Snort

Snort được chia thành nhiều thành phần một cách logic. Những thành phần này làm việc cùng nhau để phát hiện các cuộc tấn công cụ thể và để tạo ra các định dạng cần thiết từ hệ thống phát hiện. Snort bao gồm các thành phần chính sau đây:

- Packet Decoder
- Preprocessors
- Detection Engine
- Logging and alerting system
- Output Modules

Ở hình 2.4 cho thấy các thành phần này được sắp xếp. Bất kỳ dữ liệu nào đến từ internet đều đi vào packet decoder. Trên đường đi của nó với các module đầu ra, nó hoặc bị loại bỏ, ghi nhận hoặc một cảnh báo được tạo ra.



Hình 2-4. Các thành phần của Snort

2.1.4.1. Packet Decoder (bộ phận giải mã gói tin)

Các gói dữ liệu đi vào qua các cổng giao tiếp mạng, các cổng giao tiếp này có thể là: Ethernet, SLIP, PPP... Và được giải mã bởi packet decoder, trong đó xác định giao thức được sử dụng cho gói tin và dữ liệu phù hợp với hành vi được cho phép của phần giao thức của chúng. Packet Decoder có thể tạo ra các cảnh báo riêng của mình dựa trên các tiêu đề của giao thức, các gói tin quá dài, bất thường hoặc không chính xác các tùy chọn TCP được thiết lập trong các tiêu đề, và các hành vi khác. Có thể kích hoạt hoặc vô hiệu hóa các cảnh báo dài dòng cho tất cả các trường trong tập tin snort.conf.

Sau khi dữ liệu được giải mã đúng, chúng sẽ được gửi đến bộ phận tiền xử lý (preprocessor).

2.1.4.2. Preprocessor (bộ phận tiền xử lý)

Các Preprocessor là những thành phần hoặc plug-in có thể sử dụng cho Snort để sắp xếp, chỉnh sửa các gói dữ liệu trước khi bộ phận Detection Engine làm việc với chúng. Một số Preprocessor cũng thực hiện phát hiện dấu hiệu dị thường bằng cách tìm trong phần tiêu đề của gói tin và tạo ra các cảnh báo.

Preprocessor rất quan trọng với bất kỳ hệ thống IDS nào để chuẩn bị dữ liệu cần thiết về gói tin để bộ phận Detection Engine làm việc.

Preprocessor còn dùng để tái hợp gói tin cho các gói tin có kích thước lớn. Ngoài ra nó còn giải mã các gói tin đã được mã hóa trước khi chuyển đến bộ phận Detection Engine.

2.1.4.3. Detection Engine (bộ phận kiểm tra)

Detection Engine là bộ phận quan trọng nhất của Snort. Trách nhiệm của nó là phát hiện bất kỳ dấu hiệu tấn công nào tồn tại trong gói tin bằng cách sử dụng các rule để đối chiếu với thông tin trong gói tin. Nếu gói tin là phù hợp với rule, hành động thích hợp được thực hiện

Hiệu suất hoạt động của bộ phận này phụ thuộc các yếu tố như: Số lượng rule, cấu hình máy mà Snort đang chạy, tốc độ bus sử dụng cho máy Snort, lưu lượng mạng.

Detection Engine có thể phân chia gói tin và áp dụng rule cho các phần khác nhau của gói tin. Các phần đó có thể là:

- Phần IP header của gói tin
- Phần header của tầng transport: Đây là phần tiêu đề bao gồm TCP, UDP hoặc các header của tầng transport khác. Nó cũng có thể làm việc với header của ICMP.
- Phần header của các lớp ứng dụng: Bao gồm header của lớp ứng dụng, nhưng không giới hạn, DNS header, FTP header, SNMP header, và SMTP header.
- Packet payload: Có nghĩa là có thể tạo ra rule được sử dụng bởi detection engine để tìm kiếm một chuỗi bên trong dữ liệu của gói tin.

Bộ phận này hoạt động theo hai cách khác nhau theo hai phiên bản của Snort.

- **Phiên bản 1.x:** Việc xử lý gói tin còn hạn chế trong trường hợp các dấu hiệu trong gói tin đó phù hợp với dấu hiệu trong nhiều rule. Khi đó nếu có rule nào được áp dụng trước thì các rule còn lại sẽ bị bỏ qua mặc dù các rule có độ ưu tiên khác nhau. Như vậy sẽ nảy sinh trường hợp các rule có độ ưu tiên cao hơn bị bỏ qua.
- **Phiên bản 2.x:** Nhược điểm trên của phiên bản 1.x được khắc phục hoàn toàn nhờ vào cơ chế kiểm tra trên toàn bộ rule. Sau đó lấy ra rule có độ ưu tiên cao nhất để tạo thông báo.

Tốc độ của phiên bản 2.x nhanh hơn rất nhiều so với phiên bản 1.x nhờ phiên bản 2.x được biên dịch lại.

2.1.4.4. Logging and Alerting System (Bộ phận ghi nhận và thông báo)

Khi bộ phận detection engine phát hiện ra các dấu hiệu tấn công thì nó sẽ thông báo cho bộ phận Logging and Alerting System. Các ghi nhận, thông báo có thể được lưu dưới dạng văn bản hoặc một số định dạng khác. Mặc định thì chúng được lưu tại thư mục `./var/log/snort`.

2.1.4.5. Output Modules (bộ phận đầu ra)

Bộ phận đầu ra của Snort phụ thuộc vào việc ta ghi các ghi nhận, thông báo theo cách thức nào. Có thể cấu hình bộ phận này để thực hiện các chức năng sau:

- Lưu các ghi nhận và thông báo theo định dạng các file văn bản hoặc vào cơ sở dữ liệu.
- Gửi thông tin SNMP.
- Gửi các thông điệp đến hệ thống ghi log.
- Lưu các ghi nhận và thông báo vào cơ sở dữ liệu (MySQL, Oracle...).
- Tạo đầu ra XML.
- Chính sửa cấu hình trên Router, Firewall.
- Gửi các thông điệp SMB.

2.1.5. Các chế độ thực thi của Snort

2.1.5.1. Sniff mode

Ở chế độ này, Snort hoạt động như một chương trình thu thập và phân tích gói tin thông thường. Không cần sử dụng file cấu hình, các thông tin Snort sẽ thu được khi hoạt động ở chế độ này:

- Date and time.
- Source IP address.
- Source port number.
- Destination IP address.
- Destination port.
- Transport layer protocol used in this packet.
- Time to live or TTL value in this packet.
- Type of service or TOS value.
- Packer ID.
- Length of IP header.
- IP payload.
- Don't fragment or DF bit is set in IP header.
- Two TCP flags A and P are on.
- TCP sequence number.
- Acknowledgement number in TCP header.
- TCP Window field.
- TCP header length.

2.1.5.2. Pakcet logger mode

Khi chạy ở chế độ này, Snort sẽ tập hợp tất cả các packet nó thấy được và đưa vào log theo cấu trúc phân tầng. Nói cách khác, một thư mục mới sẽ được tạo ra ứng với mỗi địa chỉ nó bắt được, và dữ liệu sẽ phụ thuộc vào địa chỉ mà nó lưu trong thư mục đó. Snort đặt các packet vào trong file ASCII, với tên liên quan đến giao thức và cổng. Sự sắp xếp này dễ dàng nhận ra ai đang kết nối vào mạng của

mình và giao thức, cổng nào đang sử dụng. Đơn giản sử dụng `ls-R` để hiện danh sách các thư mục.

Tuy nhiên sự phân cấp này sẽ tạo ra nhiều thư mục trong giờ cao điểm nên rất khó để xem hết tất cả thư mục và file này. Nếu ai đó sử dụng full scan với 65536 TCP Port và 65535 UDP ports và sẽ tạo ra 131000 hoặc từng ấy file .

Log với dạng nhị phân (binary) tất cả những gì có thể đọc được bởi Snort, nó làm tăng đáng kể khả năng bắt gói tin của Snort. Hầu hết các hệ thống có thể capture và log ở tốc độ 100Mbps mà không có vấn đề gì.

Để log packet ở chế độ nhị phân, sử dụng cờ `-b`:

```
#Snort -b -l /usr/local/log/Snort/temp.log
```

Khi đã capture, ta có thể đọc lại file mới vừa tạo ra ngay với cờ `-r` và phần hiển thị giống như ở mode sniffer:

```
#Snort -r /usr/local/log/Snort/temp.log
```

Trong phần này Snort không giới hạn để đọc các file binary trong chế độ sniffer. Ta có thể chạy Snort ở chế độ NIDS với việc set các rule hoặc filters để tìm những traffic nghi ngờ.

2.1.5.3. NIDS mode

Snort thường được sử dụng như một NIDS. Nó nhẹ, nhanh chóng, hiệu quả và sử dụng các rule để áp dụng lên gói tin. Khi phát hiện có dấu hiệu tấn công ở trong gói tin thì nó sẽ ghi lại và tạo thông báo. Khi dùng ở chế độ này phải khai báo file cấu hình cho Snort hoạt động. Thông tin về thông báo khi hoạt động ở chế độ này:

- **Fast mode:** Date and time, Alert message, Source and destination IP address, Source and destination ports, Type of packet.
- **Full mode:** Gồm các thông tin như chế độ fast mode và thêm một số thông tin sau: TTL value, TOS value, Length of packet header, length of packet, Type of packet, Code of packet, ID of packet, Sequence number.

2.1.5.4. Inline mode

Đây là phiên bản chỉnh sửa từ Snort cho phép phân tích các gói tin từ firewall iptables sử dụng các tập lệnh mới như: pass, drop, reject.

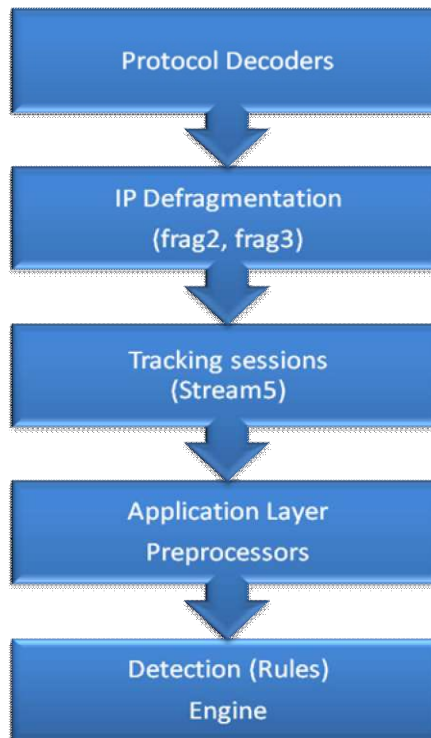
2.1.6. Preprocessor (Bộ tiền xử lý)

2.1.6.1. Giới thiệu

Bộ phận preprocessor là một trong những bộ phận quan trọng, cấu thành nên một hệ thống Snort hoàn thiện. Các tiền xử lý là các module với những đoạn mã phức tạp được biên dịch nhằm nâng cao khả năng thực thi cho Snort. Các preprocessor không chỉ thực thi các chức năng kiểm tra, rà soát các giao thức thông thường mà chúng còn có khả năng tạo ra các thông báo, giảm tải rất nhiều cho bộ phận Detection Engine.

Quá trình sử dụng và vận hành các preprocessor một cách thích hợp làm cho hệ thống IDS trở nên uyển chuyển linh hoạt hơn rất nhiều, làm tăng khả năng nhận dạng các packet nghi ngờ, tăng khả năng nhận diện attacker sử dụng các kỹ thuật để đánh lạc hướng IDS.

2.1.6.2 Mô hình



Hình 2-5. Mô hình xử lý của bộ phận tiền xử lý

2.1.6.3 Một số tiền xử lý thông dụng

a. Frag3

Các IDS hoạt động nhờ vào việc đối chiếu các rule với từng packet riêng biệt. Do đó, kẻ tấn công có thể chia nhỏ gói tin ra (thay đổi kích thước gói tin) để đánh lừa cơ chế này. Do đó frag3 thực hiện ghép nối gói tin lại với nhau thành một gói hoàn chỉnh rồi mới chuyển đến bộ phận Detection Engine để xử lý.

Frag3 được đưa ra nhằm thay thế cho Frag2 và có các đặc điểm sau:

- Thực thi nhanh hơn Frag2 trong việc xử lý các dữ liệu phứt tạp (khoảng 250%).
- Có hai cơ chế quản lý bộ nhớ để thực thi cho từng môi trường riêng biệt.
- Sử dụng công nghệ anti-evasion (chống khả năng đánh lừa của kẻ tấn công).

Frag2 sử dụng thuật toán splay trees trong việc quản lý dữ liệu cấu trúc gói tin đã phân mảnh. Đây là một thuật toán tiên tiến nhưng giải thuật này chỉ phù hợp với dữ liệu có ít sự thay đổi. Còn khi đặt thuật toán này trong môi trường mà dữ liệu có sự biến đổi cao thì bị hạn chế về khả năng thực thi (performance). Để giải quyết những hạn chế đó thì Frag3 ra đời.

Frag3 sử dụng cấu trúc dữ liệu sfxhash để quản lý dữ liệu trong môi trường phân mảnh cao. Target-based analysis là một khái niệm mới trong NIDS. Ý tưởng của hệ thống này là dựa vào hệ thống đích thực tế trong mạng thay vì chỉ dựa vào các giao thức và thông tin tấn công chứa bên trong nó. Nếu một kẻ tấn công có nhiều thông tin về hệ thống đích hơn IDS thì chúng có thể đánh lừa được các IDS.

b. Stream5

Stream5 là một module theo kiểu target-based, được thiết kế để giúp Snort chống lại các tấn tới các sensor bằng cách gửi nhiều các packet chứa dữ liệu giống nhau như trong rule nhằm cho IDS báo động sai.

Stream5 thay thế cho các tiền xử lý như Stream4 và flow. Nó có khả năng theo dõi cả phiên của TCP và UDP.

Stream4 và Stream5 không thể dùng đồng thời. Vì vậy khi dùng Stream5 thì phải xóa bỏ cấu hình Stream4 và Flow trong file cấu hình snort.config

Các đặc điểm của Stream5:

- **Transport protocols**

Các phiên TCP được định nghĩa thông qua kết nối TCP. Các phiên UDP được thiết lập là kết quả của hàng loạt gói UDP gửi đồng thời trên cùng một cổng.

- **Target-based**

Trong stream 5 cũng giới thiệu về các action target-based để điều khiển việc chồng chéo dữ liệu và các dấu hiệu bất thường trong gói TCP khác. Các phương thức điều khiển quá trình chồng chéo dữ liệu, giá trị TCP Timestamp, dữ liệu trong SYN, FIN,... và các chính sách đều được hỗ trợ trong stream 5 đã được nghiên cứu trên nhiều hệ điều hành khác nhau.

- **Stream API**

Stream5 hỗ trợ đầy đủ Stream API cho phép cấu hình động các giao thức hoặc các Preprocessor khi có yêu cầu của giao thức thuộc lớp ứng dụng, xác định các session nào bị bỏ qua, cập nhật thông tin về các sensor mới mà có thể được sử dụng cho sau này.

- **Rule Options**

Stream5 đã thêm vào lựa chọn stream-size. Lựa chọn này cho phép các rule đối chiếu lưu lượng theo các byte được xác định trước, được xác định bởi thông số TCP sequence number.

Định dạng:

`Stream_size:<direction>,<operation>,<size>`

+ Direction nhận các giá trị sau:

Client: chỉ cho dữ liệu phía client

Server : chỉ cho dữ liệu phía server

Both: cho dữ liệu ở cả hai phía

Either: cho dữ liệu một trong hai bên hoặc là client hoặc là server.

+ Operator: =, <, >, !=, <=, =>

Ngoài ra còn một số tiền xử lý khác như:

- sfPortscan.
- RPC Decode.
- Performance Monitor.
- HTTP inspect.

- SMTP Preprocessor.
- FTP/Telnet Preprocessor.
- SSH.
- DCE/ RPC.
- SSL/ TLS.
- ARP Spoof Preprocessor.
- DCE/ RPC 2 Preprocessor.

2.1.7. Cấu trúc của Rules

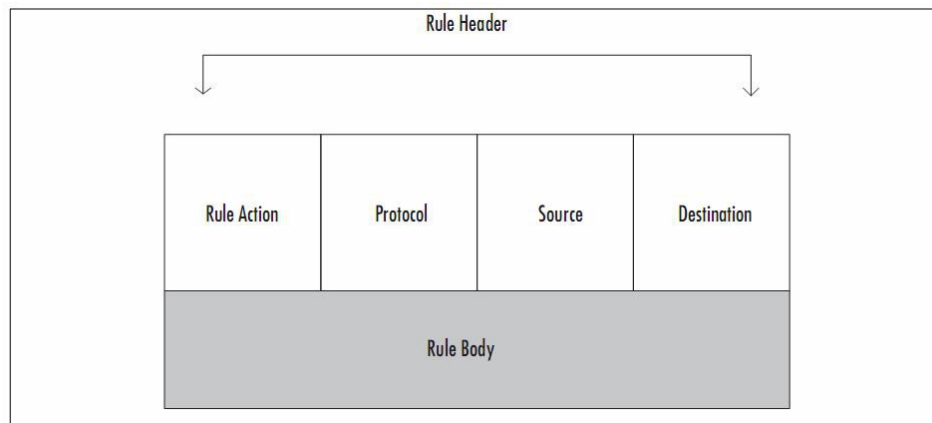
Một trong những chức năng được đánh giá cao nhất của Snort là cho phép người sử dụng tự viết các rule của riêng mình. Ngoài số lượng lớn các rule đi kèm với Snort, người quản trị có thể vận dụng khả năng của mình để phát triển ra các rule riêng thay vì phụ thuộc vào các cơ quan, tổ chức bên ngoài.

Vậy rule là gì? Rule là tập hợp các qui tắc để lựa chọn các traffic mạng phù hợp với một mô hình định trước.

Rule Snort được chia làm hai phần: *rule header* và *rule options*.

2.1.7.1. Rule header.

Rule header chứa thông tin để xác định một packet cũng như tất cả những gì cần thực hiện với tất cả các thuộc tính chỉ định trong rule. Rule header bao gồm các phần sau: *Rule actions*, *protocol*, *IP address*, *port number*, *Direction operator*.



Hình 2-6. Cấu trúc của rule header

a. Rule action

Cho Snort biết phải làm gì khi nó tìm thấy một gói tin phù hợp với rule, có năm hành động được mặc định sẵn trong Snort:

- alert: Cảnh báo và ghi lại packet.
- log: ghi lại packet.
- pass: bỏ qua packet.
- Active: Cảnh báo và thực hiện gọi một rule khác.
- Dynamic: Ở trạng thái idle cho đến khi một rule khác được kích hoạt.
- Ngoài ra khi chạy Snort ở chế độ inline, cần thêm các tùy chọn là drop, reject và sdrop.
- drop: cho phép iptables bỏ qua packet này và log lại packet vừa bỏ qua.
- reject: cho phép iptables bỏ qua packet này, log lại packet, đồng thời gửi thông báo từ chối đến máy nguồn.
- sdrop: cho phép iptables bỏ qua packet này nhưng không log lại packet, cũng không thông báo đến máy nguồn.

b. Protocols

Trường tiếp theo của rule là protocol. Hiện nay Snort chỉ hỗ trợ bốn giao thức sau: TCP, UDP, ICMP, IP. Trong tương lai có thể hỗ trợ thêm các giao thức khác như: ARP, IGRP, GRE, OSPF, RIP...

c. IP address

Các địa chỉ IP được hình thành bởi dạng thập phân: xxxx.xxxx.xxxx.xxxx và một CIDR. Snort không cung cấp cơ chế tra cứu tên host tương ứng với địa chỉ IP.

CIDR : cho biết địa chỉ lớp mạng.

Các định dạng:

- Any: bất kì địa chỉ IP nào.
- Static: một địa chỉ IP duy nhất.
- Class: một lớp các địa chỉ IP.
- Negation: Phủ định lại các địa chỉ trên.

d. Port number

Port number có thể được xác định gồm:

- Any ports: Có nghĩa là bất kỳ port nào.
- Static port: là chỉ định một port duy nhất, như: 80 (web), 21 (telnet), ...
- Ranger: phạm vi các port có thể được áp dụng.

e. Direction Operator

Chỉ ra hướng đi của rule, có hai loại đó là:

- \rightarrow : chỉ ra hướng của rule bắt nguồn từ địa chỉ IP và port bên trái .
- \leftrightarrow : Hướng của rule này là hai chiều, điều này sẽ thuận lợi cho việc phân tích cả hai mặt của một traffic, như là telnet hoặc POP3...

f. Active/ Dynamic rules

Active/ Dynamic rules cung cấp cho snort những tính năng mạnh mẽ. Có một rule khác khi hành động được thực hiện với một số gói tin. Điều này rất hữu ích cho snort để thực hiện ghi lại một số rule cụ thể.

2.1.7.2. Rule Options

Đây chính là trái tim chính của Snort, có 4 loại rule options chính: *general*, *Payload*, *Non-Payload*, *Post-detections*.

a. General options

Cung cấp thông tin về rule nhưng không gây ra bất kỳ ảnh hưởng nào đến quá trình phát hiện packet.

- **msg:**

Được sử dụng để thêm một chuỗi kí tự vào việc ghi log hoặc đưa ra cảnh báo. Thêm vào thông điệp sau dấu ngoặc kép.

Định dạng:

```
msg: "<message text>" ;
```

Ví dụ:

```
alert tcp 192.168.1.0/24 any → any any (msg: "<HTTP matched>" ;  
content: "HTTP", offset: 4)
```

- **reference:**

Là từ khóa cho phép tham chiếu đến các hệ thống phát hiện các kiểu tấn công ở bên ngoài. Nó không đóng một vai trò quan trọng nào trong cơ chế phát hiện. Có nhiều hệ thống tham khảo như CVE và Brugtraq những hệ thống này giữ thông tin về các kiểu tấn công đã được biết.

Định dạng:

reference: <id system>, <id>;

Ví dụ:

```
alert tcp any any -> any 21 (msg:"IDS287/ftp-wuftp260-venglin-  
linux"; flags:AP; content:"|31c031db 31c9b046 cd80 31c031db|";  
reference:arachnids, IDS287; reference:bugtraq,1387;  
reference:cve,CAN-2000-1574;)
```

- **gid:**

Là từ khóa dùng để xác định bộ phận nào của snort sẽ tạo ra sự kiện khi thực thi, nó giúp cho quá trình giải mã của preprocessor. Nếu không được định nghĩa trong rule nó sẽ lấy giá trị là 1. Để tránh xung đột với các rule mặc định của snort, khuyến cáo lấy giá trị lớn hơn 1.000.000. Từ khóa gid được sử dụng với từ khóa sid.

Định dạng:

gid: <generator id>;

Ví dụ:

```
alert tcp any any -> any 80 (content:"BOB"; gid:1000001; sid:1;  
rev:1;)
```

- **sid:**

Là từ khóa duy nhất để xác định snort rule, nó cho phép các thành phần output xác định các rule dễ dàng hơn. Option này nên dùng với từ khóa dev.

Định dạng:

sid: <snort rules id>;

+ id <100: Dự trữ cho tương lai.

+ 100<id<1.000.000: Xác định rule đi kèm theo bảng phân phối.

+ id>1.000.000: Do người viết rule tự định nghĩa.

Ví dụ:

```
alert tcp any any -> any 80 (content:"BOB"; sid:1000983; rev:1;)
```

- **rev:**

Từ khóa để chỉ ra số revision của rule. Nếu rule được cập nhật, thì từ khóa này được sử dụng để phân biệt giữa các phiên bản. Các module output cũng có thể sử dụng từ khóa này để nhận dạng số revision. Option này nên dùng với từ khóa dev.

Định dạng :

rev: <revision integer>;

Ví dụ:

```
alert tcp any any -> any 80 (content:"BOB"; sid:1000983; rev:1;)
```

- ***Classtype***

Classtype là từ khóa sử dụng để phân loại rule phát hiện tấn công khác nhau.

Định dạng:

classtype: <class name>;

Ví dụ:

```
alert tcp any any -> any 80 (msg:"EXPLOIT ntpdx overflow"; dsize:>128; classtype:attempted-admin; priority:10 );
```

- ***priority***

Đây là từ khóa chỉ độ ưu tiên cho rule, từ khóa classtype chỉ ra độ ưu tiên mặc định. Tuy nhiên nếu ta thiết lập thêm giá trị này nó có thể ghi đè lên giá trị mặc định đó.

Định dạng:

priority: <priority integer>;

Ví dụ:

```
alert TCP any any -> any 80 (msg: "WEB-MISC phf attempt"; flags:A+; content: "/cgi-bin/phf"; priority:10;)
```

- ***metadata:***

Cho phép người dùng nhúng thêm thông tin về rule.

Định dạng:

Metadata : key1 value1

Metadata : key1 value1, key2value2

Ví dụ:

```
alert tcp any any -> any 80 (msg: "Shared Library Rule Example"; metadata:engine shared, soid 3|12345;)
```

b. Payload Detection Rule Options

Tìm kiếm thông tin trong phần payload của packet. Phần này gồm các từ khóa như: *content, nocase, rawbytes, depth, offset, distance, within, http client body, http cookie, http header, http method, http uri, fast pattern, uricontent, urilent, isdataat, pcre, byte test, byte jump, ftpbuonce, asnl, cvs.*

- **content**

Content là từ khóa điểm quan trọng trong Snort, nó cho phép người dùng thiết lập các rule nhằm tìm ra nội dung đặc biệt trong gói tin. Việc lựa chọn dữ liệu cho gói content tương đối phức tạp, nó có thể chứa dữ liệu ở dạng văn bản hoặc ở dạng nhị phân

Định dạng:

```
content : [!] " <content string>";
```

Ví dụ:

```
alert tcp any any -> any 139 (content:"|5c00|P|00|I|00|P|00|E|00|5c|" ;)
```

Hoặc phủ định:

```
alert tcp any any -> any 80 (content:! "GET" ;)
```

- **Nocase**

Là từ khóa được sử dụng kết hợp với từ khóa content. Nó không có đối số, mục đích của nó là thực hiện việc tìm kiếm mẫu cụ thể không phân biệt kí tự hoa hoặc thường.

Định dạng

```
No case;
```

Ví dụ:

```
alert tcp any any -> any 21 (msg:"FTP ROOT"; content:"USER root"; nocase;)
```

- **offset**

offset là từ khóa sử dụng kết hợp với từ khóa content. Sử dụng khóa này, có thể bắt đầu tìm kiếm từ một vị trí xác định so với vị trí bắt đầu của gói tin. Sử dụng một con số như là đối số của từ khóa này

Định dạng:

```
Offset: <number>;
```

- **depth**

depth là từ khóa được sử dụng kết hợp với từ khóa content để xác định giới hạn của việc so sánh mẫu. Sử dụng từ khóa này, có thể xác định một vị trí so với vị trí bắt đầu. Dữ liệu sau vị trí này sẽ không được tìm kiếm để so mẫu. Nếu dùng cả

từ khóa offset và depth thì có thể xác định một khoảng dữ liệu thực hiện việc so sánh mẫu.

Định dạng:

depth : <number>;

Ví dụ:

```
alert tcp any any -> any 80 (content: "cgi-bin/phf"; offset:4;
depth:20;)
```

- **distance**

Từ khóa distance cũng tương tự như offset, điểm khác biệt là offset cho biết vị trí tìm kiếm tính từ đầu payload, trong khi distance sẽ tính từ vị trí của mẫu trước đó. Từ khóa này được dùng kết hợp với từ khóa content.

Định dạng:

distance: <byte count>;

Ví dụ:

```
alert tcp any any -> any any (content:"ABC"; content: "DEF";
distance:1;)
```

c. Non-Payload Detection Rule Options

Tìm kiếm thông tin trong phần non-payload của packet, bao gồm các từ khóa: *frag*, *offset*, *ttl*, *tos*, *id*, *ipopts*, *fragbits*, *dsize*, *flags*, *flow*, *flowbits*, *seq*, *ack*, *window*, *itype*, *icode*, *icmp id*, *icmp seq*, *rpc*, *ip proto*, *sameip*, *stream size*.

- **ttl**

Là từ khóa được sử dụng để kiểm tra trường TTL (time to live) trong phần header ip của gói tin. Từ khóa này có thể sử dụng với tất cả các giao thức xây dựng trên IP như ICMP, UDP và TCP. Sử dụng từ khóa ttl để kiểm tra ai đó đang cố gắng traceroute hệ thống mạng.

- **tos**

Đây là từ khóa được sử dụng để phát hiện một giá trị cụ thể trong trường TOS (Type of service) của IP Header.

Định dạng:

tos: [!] <number>;

- ***id***

id là từ khóa được sử dụng để kiểm tra trường ID của header gói tin IP. Mục đích của nó là phát hiện các cách tấn công một số ID cố định.

Định dạng:

```
id: <number>;
```

- ***dsiz***

dsiz là từ khóa được sử dụng để tìm chiều dài một phần dữ liệu của gói tin. Nhiều cách tấn công sử dụng lỗ hổng tràn bộ đệm bằng cách gửi gói tin có kích thước lớn. Sử dụng từ khóa này để tìm thấy gói tin có chiều dài dữ liệu lớn hoặc nhỏ hơn một số xác định.

Định dạng:

```
dsiz : [<>] <number> [<><number>];
```

Ví dụ:

- ***flags***

flags là từ khóa được sử dụng để tìm ra bit flag nào được thiết lập trong header TCP của gói tin. Các bit sau có thể được kiểm tra:

F- FIN

S-SYN

R-RST

P-PSH

A-ACK

U-URG

1- Reserved bit 1

2- Reserved bit 2

0- No TCP flags set

Một số tùy chọn khác được sử dụng

+ Phù hợp với một hoặc nhiều bit được chỉ ra.

* Phù hợp với bất kỳ bit nào được thiết lập

! Phù hợp với các bit không được thiết lập.

Định dạng:

```
flags: [ ! | * | + ] <FSRPAU120> [ , <FSRPAU120> ] ;
```

Ví dụ:

```
alert tcp any any -> any any (flags:SF,12;)
```

d. Post-Detection Rule Options

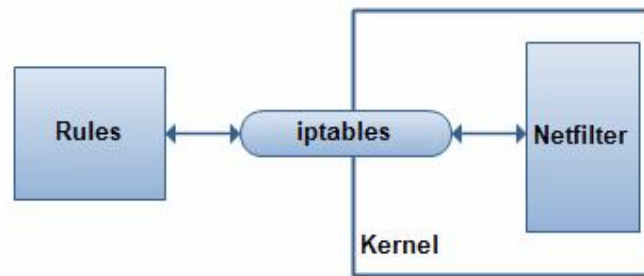
Xảy ra khi một rule được kích hoạt, gồm các từ khóa: *logto*, *session*, *resp*, *react*, *tag*, *activated by*, *count*.

2.2. FIREWALL IPTABLES TRONG HỆ ĐIỀU HÀNH LINUX

2.2.1. Giới thiệu về Iptables

Iptables là một ứng dụng tường lửa lọc gói dữ liệu rất mạnh, miễn phí và có sẵn trên hệ điều hành linux (kernel 2.4 trở đi).

Netfilter/iptables gồm có 2 phần chính. Netfilter ở trong nhân và iptables nằm ngoài nhân. Iptables chịu trách nhiệm giao tiếp với người sử dụng và netfilter để đẩy các luật của người dùng vào cho netfilter xử lý. Netfilter tiến hành lọc các gói dữ liệu ở mức IP. Netfilter làm việc trực tiếp trong nhân, nhanh và không làm giảm tốc độ của hệ thống.



Hình 2-7. Netfilter/iptables

Tiền thân của iptables là ipchain (kernel 2.2) và một trong những điểm cải tiến quan trọng của iptables là stateful packet filtering.

Iptables còn cung cấp các tính năng như NAT (Network Address Translation) và rate limit rất hữu hiệu khi chống DoS.

2.2.2. Cơ chế xử lý của iptables

2.2.2.1. Cấu trúc của iptables

Iptables được chia làm 4 bảng (tables):

- Bảng filter dùng để lọc gói dữ liệu.
- Bảng NAT dùng để thao tác với các gói dữ liệu được NAT nguồn hay NAT đích.

- Bảng Mangle dùng để thay đổi các thông số trong gói IP.
- Và bảng conntrack dùng để theo dõi các kết nối.

Mỗi tables có nhiều chuỗi (chains). Chain gồm nhiều luật (rule) để thao tác với gói dữ liệu. Rule có thể là:

- ACCEPT-Chấp nhận gói dữ liệu.
- DROP-Thả gói.
- REJECT-Loại bỏ gói.
- REFERENCE-Tham chiếu đến chain khác.

2.2.2.2. Các đổi địa chỉ IP động (dynamic IP)

NAT động là một trong những kỹ thuật chuyển đổi địa chỉ IP NAT (Network Address Translation). Các địa chỉ IP nội bộ được chuyển sang IP NAT như sau.

NAT Router đảm nhận việc chuyển dãy IP nội bộ 192.168.0.x sang dãy IP mới 203.162.2.x. Khi có gói dữ liệu với IP nguồn là 192.168.0.200 đến router, router sẽ đổi IP nguồn thành 203.162.2.200 sau đó mới gửi ra ngoài. Quá trình này gọi là SNAT (Source-NAT, NAT nguồn). Router lưu dữ liệu trong một bảng gọi là bảng NAT động. Ngược lại, khi có một gói từ liệu từ gửi từ ngoài vào với IP đích là 203.162.2.200, router sẽ căn cứ vào bảng NAT động hiện tại để đổi địa chỉ đích 203.162.2.200 thành địa chỉ đích mới là 192.168.0.200. Quá trình này gọi là DNAT (Destination-NAT, NAT đích). Liên lạc giữa 192.168.0.200 và 203.162.2.200 là hoàn toàn trong suốt (transparent) qua NAT router. NAT router tiến hành chuyển tiếp (forward) gói dữ liệu từ 192.168.0.200 đến 203.162.2.200 và ngược lại.

2.2.2.3 Cách đóng giả địa chỉ IP.

NAT Router chuyển dãy IP nội bộ 192.168.0.x sang một IP duy nhất là 203.162.2.4 bằng cách dùng các số hiệu cổng (port-number) khác nhau. Chẳng hạn khi có gói dữ liệu IP với nguồn 192.168.0.168:1204, đích 211.200.51.15:80 đến router, router sẽ đổi nguồn thành 203.162.2.4:26314 và lưu dữ liệu này vào một bảng gọi là bảng masquerade động. Khi có một gói dữ liệu từ ngoài vào với nguồn là 221.200.51.15:80, đích 203.162.2.4:26314 đến router, router sẽ căn cứ vào bảng masquerade động hiện tại để đổi đích từ 203.162.2.4:26314 thành 192.168.0.164:1204. Liên lạc giữa các máy trong mạng LAN với máy khác bên ngoài hoàn toàn trong suốt qua router.

2.2.3. Cơ chế xử lý gói tin.

Tất cả mọi gói dữ liệu đều được kiểm tra bởi iptables bằng cách dùng các bảng tuần tự xây dựng sẵn (queue). Có 3 loại bảng này gồm :

Mangle table: chịu trách nhiệm biến đổi quality of service bits trong TCP header. Thông thường loại table này được ứng dụng trong SOHO (Small Office/Home Office).

Filter queue: chịu trách nhiệm thiết lập bộ lọc packet (packet filtering), có ba loại built-in chains được mô tả để thực hiện các chính sách về firewall (firewall policy rules).

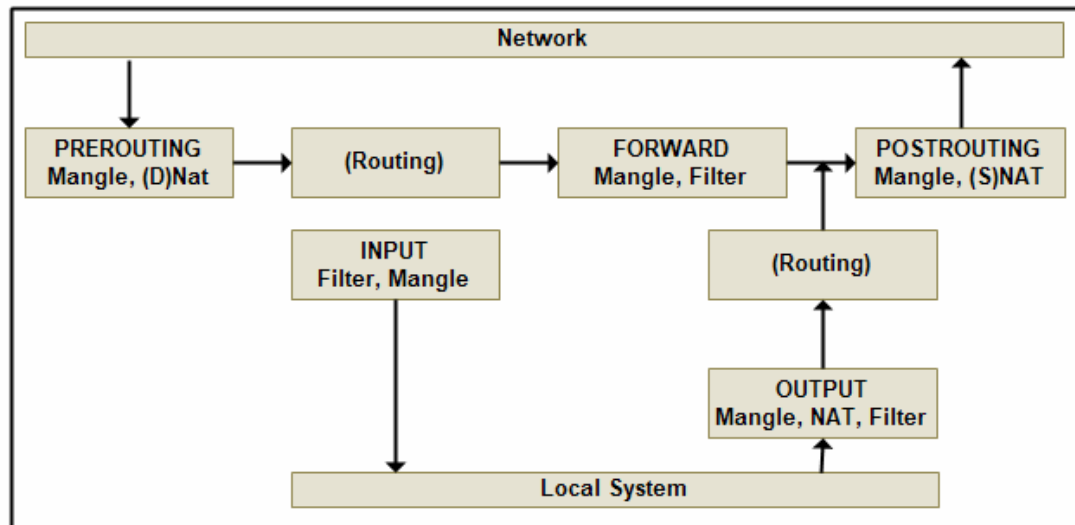
- Forward chain: Cho phép packet nguồn chuyển qua firewall.
- Input chain: Cho phép những gói tin đi vào từ firewall.
- Output chain: Cho phép những gói tin đi ra từ firewall.

NAT queue: thực thi chức năng NAT (Network Address Translation), cung cấp hai loại built-in chains sau đây:

- **Pre-routingchain:** NAT từ ngoài vào trong nội bộ. Quá trình NAT sẽ thực hiện trước khi thực thi cơ chế routing. Điều này thuận lợi cho việc đổi địa chỉ đích để địa chỉ tương thích với bảng định tuyến của firewall, khi cấu hình ta có thể dùng khóa DNAT để mô tả kỹ thuật này.
- **Post-routingchain:** NAT từ trong ra ngoài. Quá trình NAT sẽ thực hiện sau khi thực hiện cơ chế định tuyến. Quá trình này nhằm thay đổi địa chỉ nguồn của gói tin. Kỹ thuật này được gọi là NAT one-to-one hoặc many-to-one, được gọi là Source NAT hay SNAT.

Bảng 2-1. Các loại queues và chain cùng chức năng của nó

Loại queue	Chức năng queues	Quy tắc xử lý gói	Chức năng của chain
Filter	Lọc gói	FORWARD	Lọc gói dữ liệu đi đến các server khác kết nối trên các NIC khác của firewall.
		INPUT	Lọc gói đi đến firewall
		OUTPUT	Lọc gói đi ra khỏi firewall
NAT	Network Address Translation (Biên dịch địa chỉ mạng)	PREROUTING	Việc thay đổi địa chỉ diễn ra trước khi dẫn đường. Thay đổi địa chỉ đích sẽ giúp gói dữ liệu phù hợp với bảng chỉ đường của firewall. Sử dụng destination NAT or DNAT .
		POSTROUTING	Việc thay đổi địa chỉ diễn ra sau khi dẫn đường . Sử dụng source NAT , or SNAT .
		OUTPUT	NAT sử dụng cho các gói dữ liệu xuất phát từ firewall . Hiếm khi dùng trong môi trường SOHO (small office - home office) .
Mangle	Chỉnh sửa TCP header .	PREROUTING POSTROUTING OUTPUT INPUT FORWARD	Điều chỉnh các bit quy định chất lượng dịch vụ trước khi dẫn đường. Hiếm khi dùng trong môi trường SOHO (Small Office - Home Office) .



Hình 2-8. Mô tả sự lọc và quản lý trong iptables

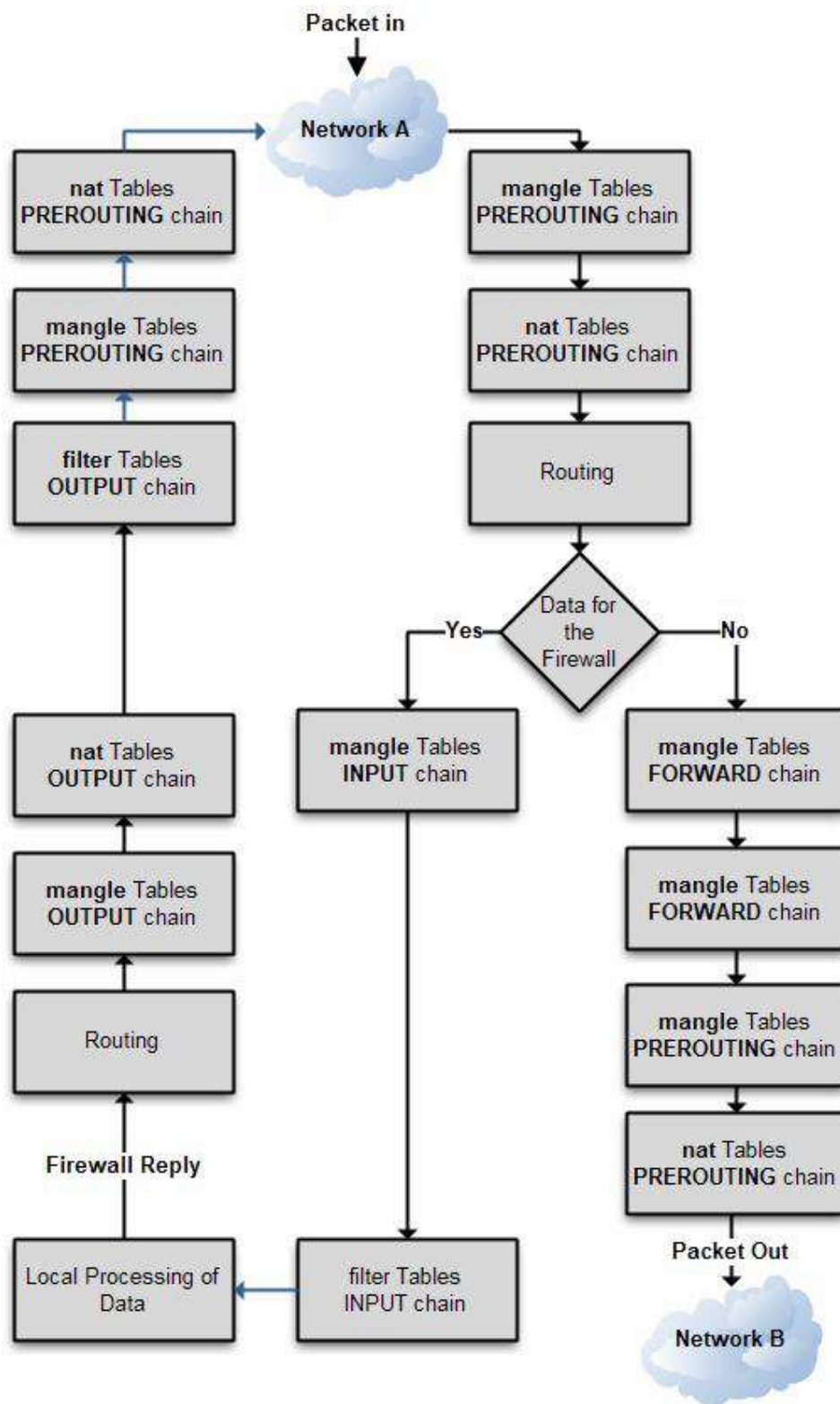
Ví dụ: Mô tả đường đi của gói dữ liệu

Đầu tiên, gói dữ liệu đến mạng A , tiếp đó nó được kiểm tra bởi mangle table PREROUTING chain (nếu cần). Tiếp theo là kiểm tra gói dữ liệu bởi nat table's PREROUTING chain để kiểm tra xem gói dữ liệu có cần DNAT hay không? DNAT sẽ thay đổi địa chỉ đích của gói dữ liệu . Rồi gói dữ liệu được dẫn đi .

Nếu gói dữ liệu đi vào một mạng được bảo vệ, thì nó sẽ được lọc bởi FORWARD chain của filter table, và nếu cần gói dữ liệu sẽ được SNAT trong POSTROUTING chain để thay đổi IP nguồn trước khi vào mạng B.

Nếu gói dữ liệu được định hướng đi vào trong bên trong firewall , nó sẽ được kiểm tra bởi INPUT chain trong mangle table, và nếu gói dữ liệu qua được các kiểm tra của INPUT chain trong filter table, nó sẽ vào trong các chương trình của server bên trong firewall .

Khi firewall cần gói dữ liệu ra ngoài . Gói dữ liệu sẽ được dẫn và đi qua sự kiểm tra của OUTPUT chain trong mangle table(nếu cần), tiếp đó là kiểm tra trong OUTPUT chain của nat table để xem DNAT (DNAT sẽ thay đổi địa chỉ đến) có cần hay không và OUTPUT chain của filter table sẽ kiểm tra gói dữ liệu nhằm phát hiện các gói dữ liệu không được phép gửi đi. Cuối cùng trước khi gói dữ liệu được đưa ra lại Internet, SNAT and QoS sẽ được kiểm tra trong POSTROUTING chain.



Hình 2-9. Đường đi của gói dữ liệu

2.2.4 Target

Target là hành động sẽ diễn ra khi một gói dữ liệu được kiểm tra và phù hợp với một yêu cầu nào đó. Khi một target đã được nhận dạng , gói dữ liệu cần nhảy (jump) để thực hiện các xử lý tiếp theo . Bảng sau liệt kê các targets mà iptables sử dụng.

Bảng 2-2. Miêu tả các target mà iptables hay sử dụng nhất

Tar	Ý nghĩa	Tùy Chọn
ACCEPT	iptables ngừng xử lý gói dữ liệu đó và chuyển tiếp nó vào một ứng dụng cuối hoặc hệ điều hành để xử lý .	
DROP	Iptables ngừng xử lý gói dữ liệu đó và gói dữ liệu bị chặn, loại bỏ.	
LOG	Thông tin của gói sẽ được đưa vào syslog để kiểm tra. Iptables tiếp tục xử lý gói với quy luật kế tiếp .	--log-prefix "string" Iptables sẽ thêm vào log message một chuỗi do người dùng định sẵn . Thông thường là để thông báo lý do vì sao gói bị bỏ .

REJECT	<p>Tương tự như DROP, nhưng sau: icmp-port- nó sẽ gửi trả lại cho phía người gửi một thông báo lỗi rằng gói đã bị chặn và loại bỏ .</p>	<p>--reject-with <i>qualifier</i></p> <p>Tham số <i>qualifier</i> sẽ cho biết loại thông báo gửi trả lại phía gửi. Qualifier gồm các loại unreachable(default) icmp-net-unreachable icmp-host-unreachable icmp-proto-nreachable icmp-net-prohibited icmp-host-prohibited tcp-reset echo-ply.</p>
DNAT	<p>Dùng để thực hiện Destination network address translation, địa chỉ đích của gói dữ liệu sẽ được viết lại.</p>	<p>--to-destination <i>ipaddress</i></p> <p>Iptables sẽ viết lại địa chỉ <i>ipaddress</i> vào địa chỉ đích của gói dữ liệu.</p>
SNAT	<p>Dùng để thực hiện Source Network address translation, viết lại địa chỉ nguồn của gói dữ liệu.</p>	<p>--to-source <<i>address</i>> [-<<i>address</i>>][:<<i>Port</i>> -<<i>port</i>>]</p> <p>Miêu tả IP và port sẽ được viết lại bởi iptables .</p>

MASQUERADE	Dùng để thực hiện Source Networkaddress Translation . Mặc định thì địa chỉ IP nguồn sẽ giống như IP nguồn của firewall.	<code>[--to-ports <port>[<port>]]</code> Ghi rõ tầm các port nguồn mà port nguồn gốc có thể ánh xạ được.
------------	--	---

2.2.5 Ưu điểm và nhược điểm của Iptables

2.2.5.1. Ưu điểm

Linux được nhiều người thừa nhận như là một nền tảng hệ điều hành an toàn, ít bị tấn công, không chỉ bởi kiến trúc của phần lõi bên dưới, mà còn nhờ những lớp giáp trụ bảo vệ bên trên. Một trong những lớp che chắn hiệu quả nhất ở lớp ngoài cùng là phần mềm tường lửa nguồn mở nổi tiếng iptables.

Ưu điểm của iptables là ở chỗ chúng là một phần của lõi Linux 2.4 (và sau này). Iptables là một công cụ quản lý cấu hình tường lửa. Với nó, có thể tạo ra một tập các đối tượng mô tả tường lửa của bạn, các máy chủ và các mạng con của mạng của bạn và sau đó kéo những đối tượng này vào trong các quy tắc cách xử sự để triển khai tường lửa của bạn. Điều đó dễ dàng hơn nhiều so với sửa chữa các tập tin cấu hình một cách thủ công và nó là nguồn mở.

Ngoài ra Iptables còn có:

- Là một statefull firewall.
- Filter packet dựa trên địa chỉ MAC và các cờ của TCP header.
- NAT tốt hơn.
- Hỗ trợ việc tích hợp một cách trong suốt với các chương trình như Web proxy: Squid.

Một ưu điểm khác của iptables nó là giới hạn được số lượng kết nối, giúp cho ta chống được các cơ chế tấn công như DoS (Denial of Service attack).

2.2.5.2. Nhược điểm

Nhược điểm lớn nhất của iptables là việc cài đặt và hiểu rõ cấu hình chúng không dễ dàng chút nào.

Sử dụng tường lửa cần phải xử lý một lượng lớn thông tin nên việc xử lý lọc thông tin có thể làm chậm quá trình kết nối của người kết nối.

Việc sử dụng tường lửa chỉ hữu hiệu đối với những người không thành thạo kỹ thuật vượt tường lửa, những người sử dụng khác có hiểu biết có thể dễ dàng vượt qua tường lửa bằng cách sử dụng các proxy không bị ngăn chặn.

2.3. KẾT HỢP GIỮA SNORT-INLINE VÀ IPTABLES

2.3.1. Snort-inline

Snort inline về cơ bản là một phiên bản sửa đổi của snort chấp nhận các gói tin từ iptables và IPFW qua libipq (linux) hoặc làm chệch hướng các socket (FreeBSD). Nó nhận được các gói tin được gửi từ netfilter firewall với sự trợ giúp của thư viện libipq, so sánh chúng với các dấu hiệu xâm nhập của snort và sẽ drop chúng nếu giống với rule. Sau cùng gửi chúng lại netfilter nơi mà snort-inline drop các gói tin.

2.3.2. Snort-inline và Iptables

Netfilter là một module của kernel linux có sẵn ở các phiên bản kernel 2.4 trở đi. Nó cung cấp 3 chức năng chính:

- Packet filtering: Accept hay drop các gói tin.
- NAT : Thay đổi địa chỉ nguồn/ đích của địa chỉ IP của các gói tin.
- Packet mangling : định dạng các gói tin.

IPtables là một công cụ cần thiết để cấu hình netfilter, nó cần phải được chạy bởi quyền root.

Sau đó, nếu một gói tin phù hợp với dấu hiệu tấn công của Snort_inline, nó được gắn thẻ libipq và gửi trả lại Netfilter nơi mà nó được drop.

Snort_inline có hai chế độ: *Drop mode* và *Replace mode*.

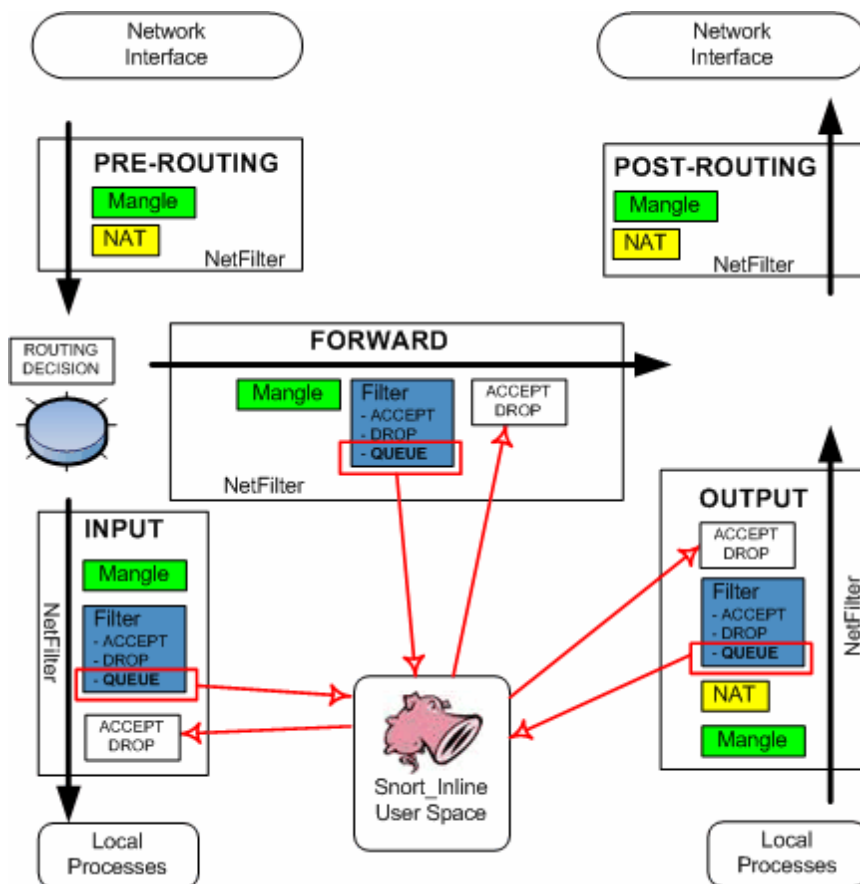
a. Drop mode:

Một packet được drop khi nó phù hợp với các dấu hiệu tấn công. Có 3 tùy chọn trong chế độ này:

- drop: Drop một gói tin, gửi một thiết lập đến máy chủ, ghi lại sự kiện.
- sdrops: Drop một gói tin mà không gửi thiết lập đến máy chủ.
- ignore: Drop một packet, gửi một thiết lập đến máy chủ, không ghi lại sự kiện.

b. Replace mode:

Packet bị sửa đổi nếu nó phù hợp với dấu hiệu tấn công.



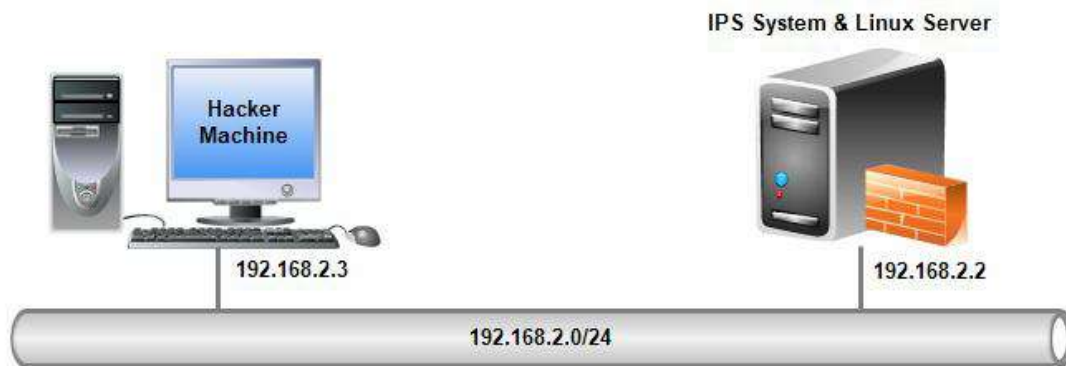
Hình 2-10. Snort-inline và netfilter

CHƯƠNG 3

TRIỂN KHAI HỆ THỐNG IPS VỚI SNORT-INLINE VÀ IPTABLES

Trong chương này chúng ta tiến hành triển khai một hệ thống IPS trên thực tế sử dụng `snort_inline` và `iptables` firewall của Linux để tiến hành ngăn chặn các hoạt động trái phép đến hệ thống mạng được IPS bảo vệ.

3.1. MÔ HÌNH TRIỂN KHAI



Hình 3-1. Mô hình triển khai IPS với snort-inline và iptables

3.1.1. Mô tả yêu cầu

3.1.1.1. Yêu cầu máy chủ:

- Cài đặt hệ điều hành linux, cụ thể là CentOS.
- Cài đặt snort-inline và các công cụ hỗ trợ, bật chức năng firewall iptables của hệ thống để xây dựng một hệ thống IPS.
- Máy chủ và IPS System cài chung trên host có Server có địa chỉ IP tĩnh là 192.168.2.2

3.1.1.2. Yêu cầu máy hacker:

- Máy tấn công vào hệ thống chạy hệ điều hành Linux-Backtrack4. Đây là một hệ điều hành với rất nhiều công cụ bảo mật được hỗ trợ.
- Cấu hình địa chỉ IP tĩnh là 192.168.2.3

3.2. CÀI ĐẶT SNORT

3.2.1. Cài đặt các gói hỗ trợ

Đầu tiên cần cài các gói phần mềm hỗ trợ sau:

httpd	httpd-devel	mysql
mysql-sever	mysql-devel	php
php-mysql	php-mbstring	php-mcrypt
iptables	iptables-devel	libnet
Pcre	pcre-devel	gcc

Trong cửa sổ dòng lệnh dùng lệnh sau để cài đặt:

```
root@localhost# yum install <tên gói>
```

3.2.2. Cấu hình mysql và cài phpmyadmin

3.2.2.1. Cấu hình mysql

```
[root@localhost]# chkconfig --levels 235 mysqld on
[root@localhost]# /etc/init.d/mysqld start
[root@localhost]# mysqladmin -u root password mysqlpassword
```

3.2.2.2. Cài đặt phpmyadmin

phpmyadmin dùng để quản lý mysql

```
[root@localhost]# wget http://packages.sw.be/rpmforge-release/rpmforge-release-0.3.6-1.el5.rf.i386.rpm
[root@localhost]# rpm -Uvh rpmforge-release-0.3.6-1.el5.rf.i386.rpm
[root@localhost]# yum install phpmyadmin
[root@localhost]# vi /etc/httpd/conf.d/phpmyadmin.conf
```

```
#
# Web application to manage MySQL
#
#<Directory "/usr/share/phpmyadmin">
# Order Deny,Allow
# Deny from all
# Allow from 127.0.0.1
#</Directory>
Alias /phpmyadmin /usr/share/phpmyadmin
Alias /phpMyAdmin /usr/share/phpmyadmin
Alias /mysqladmin /usr/share/phpmyadmin
```



```
[root@localhost]# vi /usr/share/phpmyadmin/config.inc.php
```

Thay \$cfg['Servers'][\$i]['auth_type'] = 'cookie';

Bằng \$cfg['Servers'][\$i]['auth_type'] = 'http';

3.2.3. Cài đặt Snort_inline

Download snort_inline tại địa chỉ:

```
[root@localhost#wget http://sourceforge.net/projects/snort-inline/files/snort_inline%20source%20%282.8.x%29/snort_inline-2.8.2.1-RC1/snort_inline-2.8.2.1-RC1.tar.gz/download
```

```
[root@localhost]# tar xvfz snort_inline-2.8.2.1-RC1.tar.gz
```

```
[root@localhost]# mkdir /etc/snort_inline
```

```
[root@localhost]# mkdir /etc/snort_inline/rules/
```

```
[root@localhost]# cp snort_inline-2.8.2.1-RC1/etc/*  
/etc/snort_inline/
```

```
[root@localhost]# cp /root/snort_inline02.8.2.1-RC1/etc/reference.config /etc/snort_inline/rules
```

```
[root@localhost]# cp /root/snort_inline02.8.2.1-RC1/etc/classification.config /etc/snort_inline/rules
```

```
[root@localhost]# vi /etc/snort_inline/snort_inline.conf
```

Tìm dòng

```
# var RULE_PATH /etc/snort_inline/drop-rules
```

Thay thế thành

```
# var RULE_PATH /etc/snort_inline/rules
```

```
output      database:  log,      mysql,      user=snort      password=12345  
dbname=snort host=localhost
```

```
[root@localhost]# cd snort_inline-2.8.2.1
```

```
./configure --with-mysql --enable-dynamicplugin
```

```
./make && make install
```

Như vậy, đã cài đặt xong. Copy rule vào thư mục /etc/snort_inline/rules

3.2.4. Cài đặt, cấu hình ACIDBase để quản lý Snort

Cần phải đảm bảo đã cài đặt các phần mềm sau:

- Snort_inline.
- Apache.
- PHP.
- MySQL.

- Adodb (download tại địa chỉ <http://sourceforge.net/projects/adodb/files/> sau đó giải nén copy vào thư mục `/var/www/html/`)

Bước 1: Tạo cơ sở dữ liệu trong mysql

Tạo cơ sở dữ liệu với tên snort, tạo 6 bảng sau: *acid_event*, *acid_ag*, *acid_ag_alert*, *acid_ip_cache*, *base_roles*, *base_users*. Các bảng này đi kèm theo bảng phân phối ACIDBase.

Bước 2: chỉnh sửa nội dung file base_conf.php

Đường dẫn đến thư mục cài đặt Base: `$BASE_urlpath = '/base';`

Đường dẫn đến thư mục adodb: `$DBlib_path = '/var/www/html/adodb';`

Cơ sở dữ liệu sử dụng: `$DBtype = 'mysql';`

Khai báo tên cơ sở dữ liệu, tài khoản đăng nhập, mật khẩu

```
$alert_dbname = 'snort';
$alert_host   = 'localhost';
$alert_port   = '';
$alert_user   = 'snort';
$alert_password = '12345';
```

3.2.5. Tạo file khởi động Snort_inline cùng với hệ điều hành

Tạo một file snortd trong thư mục `/etc/init.d/` với nội dung sau

```
#!/bin/bash
#
# snort_inline

start() {
# Start daemons.
echo "Starting ip_queue module:"
lsmod | grep ip_queue >/dev/null || /sbin/modprobe ip_queue;
#
echo "Starting iptables rules:"
# iptables traffic sent to the QUEUE:
# accept internal localhost connections
iptables -A INPUT -i lo -s 127.0.0.1 -d 127.0.0.1 -j ACCEPT
iptables -A OUTPUT -o lo -s 127.0.0.1 -d 127.0.0.1 -j ACCEPT
# send all the incoming, outgoing and forwarding traffic to the QUEUE
iptables -A INPUT -j QUEUE
iptables -A FORWARD -j QUEUE
iptables -A OUTPUT -j QUEUE
# Start Snort_inline
echo "Starting snort_inline: "
/usr/local/bin/snort_inline -c /etc/snort_inline/snort_inline.conf -Q -D -v \
-I /var/log/snort_inline
# -Q -> process the queued traffic
# -D -> run as a daemon
# -v -> verbose
# -I -> log path
# -c -> config path
}

stop() {
```

```
# Stop daemons.
# Stop Snort_Inline
# echo "Shutting down snort_inline: "
killall snort_inline
# Remove all the iptables rules and
# set the default Netfilter policies to accept
echo "Removing iptables rules:"
iptables -F
# -F -> flush iptables
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT
# -P -> default policy
}

restart(){
stop
start
}

case "$1" in

start)
start
;;

stop)
stop
;;

restart)
restart
;;
*)
echo "$Usage: $0 {start|stop|restart|}"
exit 1
esac
```

Sau đó copy file này vào thư mục `./root/sbin/`

3.2.6. Tạo rule cho Snort_inline

Tạo rule lưu tại `/root/etc/snort_inline/rules`

Ta tạo 2 rule như sau:

- **Rule 1:**

```
alert icmp any any → 192.168.2.2/24 80 (msg: "ping";
ttl:128;sid:1000001;)
```

Rule trên có nghĩa là hệ thống sẽ đưa ra cảnh báo khi có bất kì máy nào ping đến máy chủ có địa chỉ 192.168.2.2. Giá trị ttl=128 ở đây là giá trị mặc định của gói icmp.

Rule thứ 2:

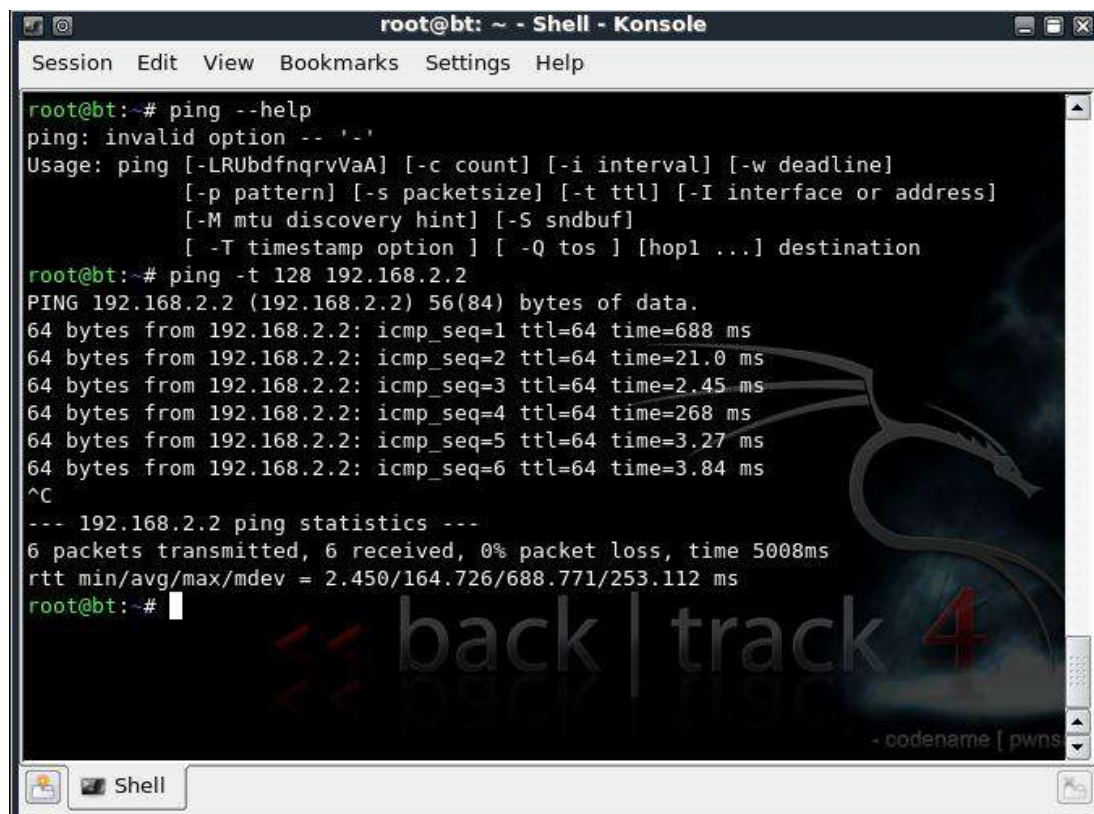
```
drop icmp any any → 192.168.1.9/24 80 (msg: "Drop Ping" ;
ttl:100;sid:1000002;)
```

Rule này có nghĩa là IPS sẽ ngắt kết nối đến server nếu có bất kì máy nào sử dụng lệnh ping với gói icmp có giá trị ttl=100.

3.3. DEMO KẾT QUẢ

Trước tiên ta chỉ chạy rule thứ 1, từ máy hacker ta tiến sử dụng lệnh ping đến địa chỉ sever. Kết quả thu được như sau:

Bước 1: Tại máy hacker



```
root@bt: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

root@bt:~# ping --help
ping: invalid option -- '-'
Usage: ping [-LRUbdfnqrVvA] [-c count] [-i interval] [-w deadline]
          [-p pattern] [-s packetsize] [-t ttl] [-I interface or address]
          [-M mtu discovery hint] [-S sndbuf]
          [-T timestamp option] [-Q tos] [hop1 ...] destination
root@bt:~# ping -t 128 192.168.2.2
PING 192.168.2.2 (192.168.2.2) 56(84) bytes of data.
64 bytes from 192.168.2.2: icmp_seq=1 ttl=64 time=688 ms
64 bytes from 192.168.2.2: icmp_seq=2 ttl=64 time=21.0 ms
64 bytes from 192.168.2.2: icmp_seq=3 ttl=64 time=2.45 ms
64 bytes from 192.168.2.2: icmp_seq=4 ttl=64 time=268 ms
64 bytes from 192.168.2.2: icmp_seq=5 ttl=64 time=3.27 ms
64 bytes from 192.168.2.2: icmp_seq=6 ttl=64 time=3.84 ms
^C
--- 192.168.2.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5008ms
rtt min/avg/max/mdev = 2.450/164.726/688.771/253.112 ms
root@bt:~#
```

Hình 3-2. Từ máy hacker ping với giá trị ttl=100 đến máy chủ

Kết quả: Khi đó chúng ta sẽ nhận lại được tín hiệu reply từ máy server.

Bước 2: Tại máy server

Ta truy cập vào ACIDBase để xem log được ghi lại:

Basic Analysis and Security Engine (BASE) : Query Results - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/base/base_qry_main.php?new=1&la:

Most Visited CentOS Support

Meta Criteria any
IP Criteria any
ICMP Criteria any
Payload Criteria any

- Sensors
- Unique Alerts
- (classifications)
- Unique addresses: Source | Destination
- Unique IP links
- Source Port: TCP | UDP
- Destination Port: TCP | UDP
- Time profile of alerts

Displaying alerts 1-6 of 6 total

ID	Signature	Timestamp	Source Address	Dest. Address	Layer 4 Proto
#0-(7-1072) [snort]	ping	2010-05-05 04:30:24	192.168.2.3	192.168.2.2	ICMP
#1-(7-1023) [snort]	ping	2010-05-05 04:30:23	192.168.2.3	192.168.2.2	ICMP
#2-(7-1022) [snort]	ping	2010-05-05 04:30:22	192.168.2.3	192.168.2.2	ICMP
#3-(7-1021) [snort]	ping	2010-05-05 04:30:21	192.168.2.3	192.168.2.2	ICMP
#4-(7-1020) [snort]	ping	2010-05-05 04:30:20	192.168.2.3	192.168.2.2	ICMP
#5-(7-1019) [snort]	ping	2010-05-05 04:30:19	192.168.2.3	192.168.2.2	ICMP

ACTION

{ action }

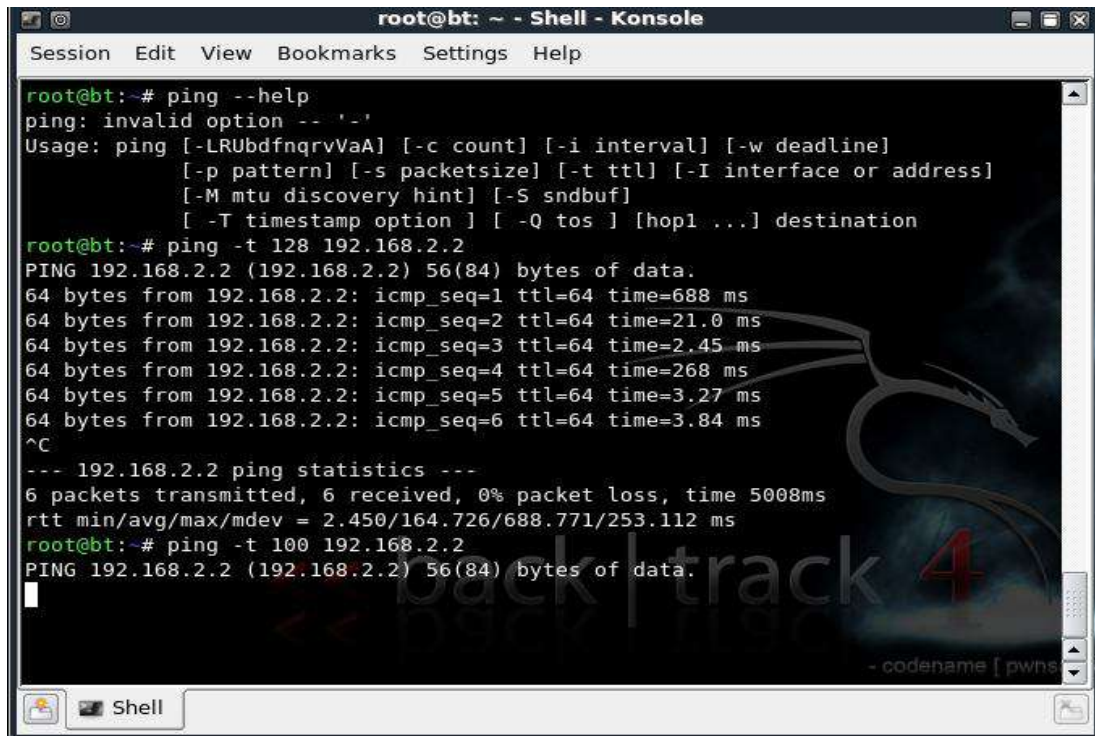
Selected ALL on Screen Entire Query

Done rules

root@lo... Basic An... Computer / etc rules

Hình 3-3. Các file log được ghi lại tại server

Bước 3: Ta tiến hành dùng lệnh ping với giá trị ttl=100.



```
root@bt:~# ping --help
ping: invalid option -- '-'
Usage: ping [-LRUbdfnqrVvA] [-c count] [-i interval] [-w deadline]
        [-p pattern] [-s packetsize] [-t ttl] [-I interface or address]
        [-M mtu discovery hint] [-S sndbuf]
        [-T timestamp option] [-Q tos] [hop1 ...] destination
root@bt:~# ping -t 128 192.168.2.2
PING 192.168.2.2 (192.168.2.2) 56(84) bytes of data:
64 bytes from 192.168.2.2: icmp_seq=1 ttl=64 time=688 ms
64 bytes from 192.168.2.2: icmp_seq=2 ttl=64 time=21.0 ms
64 bytes from 192.168.2.2: icmp_seq=3 ttl=64 time=2.45 ms
64 bytes from 192.168.2.2: icmp_seq=4 ttl=64 time=268 ms
64 bytes from 192.168.2.2: icmp_seq=5 ttl=64 time=3.27 ms
64 bytes from 192.168.2.2: icmp_seq=6 ttl=64 time=3.84 ms
^C
--- 192.168.2.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5008ms
rtt min/avg/max/mdev = 2.450/164.726/688.771/253.112 ms
root@bt:~# ping -t 100 192.168.2.2
PING 192.168.2.2 (192.168.2.2) 56(84) bytes of data:
^
```

Hình 3-4. Từ máy hacker tiến hành ping đến máy server

Kết quả: Server không reply lại, máy hacker không thể kết nối đến IPS Server.

Bước 4: Ta truy cập vào Acid base để xem log

Meta Criteria any
IP Criteria any
ICMP Criteria any
Payload Criteria any

- Sensors
- Unique Alerts
- (classifications)
- Unique addresses: Source | Destination
- Unique IP links
- Source Port: TCP | UDP
- Destination Port: TCP | UDP
- Time profile of alerts

Displaying alerts 1-48 of 56 total

ID	Signature	Timestamp	Source Address	Dest. Address	Layer 4 Proto
#0-(7-1196)	[snort] Drop Ping	2010-05-05 04:39:48	192.168.2.3	192.168.2.2	ICMP
#1-(7-1195)	[snort] Drop Ping	2010-05-05 04:39:47	192.168.2.3	192.168.2.2	ICMP
#2-(7-1194)	[snort] Drop Ping	2010-05-05 04:39:46	192.168.2.3	192.168.2.2	ICMP
#3-(7-1193)	[snort] Drop Ping	2010-05-05 04:39:45	192.168.2.3	192.168.2.2	ICMP
#4-(7-1192)	[snort] Drop Ping	2010-05-05 04:39:44	192.168.2.3	192.168.2.2	ICMP
#5-(7-1191)	[snort] Drop Ping	2010-05-05 04:39:43	192.168.2.3	192.168.2.2	ICMP
#6-(7-1190)	[snort] Drop Ping	2010-05-05 04:39:42	192.168.2.3	192.168.2.2	ICMP
#7-(7-1189)	[snort] Drop Ping	2010-05-05 04:39:41	192.168.2.3	192.168.2.2	ICMP
#8-(7-1188)	[snort] Drop Ping	2010-05-05 04:39:40	192.168.2.3	192.168.2.2	ICMP

Done

root@lo... Basic An... Computer / etc rules

Hình 3-5. Các file log được hệ thống IPS ghi lại

KẾT LUẬN VÀ HƯỚNG MỞ

KẾT LUẬN

Về mặt lý thuyết luận văn đã nêu được những vấn đề cơ bản nhất của một hệ thống phát hiện xâm nhập và hệ thống ngăn chặn xâm nhập. Bên cạnh đó đưa ra được giải pháp xây dựng một hệ thống IPS trên thực tế đã được triển khai rất hiệu quả và được đánh giá cao.

Đã xây dựng thành công một hệ thống IPS trên thực tế và hoạt động đúng với các yêu cầu đặt ra.

Hạn chế của đề tài là chỉ triển khai hệ thống trên một phân đoạn mạng nhỏ, nên chưa đánh giá được hết hiệu suất của hệ thống và các vấn đề hệ thống IPS sẽ gặp phải khi triển khai trên thực tế

HƯỚNG MỞ

Ứng dụng triển khai hệ thống IPS với Snort và iptables trên thực tế để đánh giá hết hiệu năng cũng như các vấn đề sẽ gặp phải. Từ đó có biện pháp để khắc phục, hoàn thiện hơn cho hệ thống.

Ứng dụng Snort để xây dựng các hệ thống IDS, IPS lớn có thể đặt tại các ISP để hạn chế các hoạt động tấn công mạng cho một mạng lớn. Xây dựng và phát triển hệ thống IPS phân tán.

TÀI LIỆU THAM KHẢO

1. Tiếng việt

- [1] **Trần Văn Khá** – *Firewall trong linux bằng iptables*. Đại Học Duy Tân, 2008.

2. Tiếng Anh

- [1] **Rafeeq Ur Rehman** – *Intrusion Detection Systems with Snort*. Prentice Hall PTR, 2003
- [2] **Jay Beale and Snort Development Team** – *Snort 2.1 Intrusion Detection Second edition*. Syngress Publishing, Inc, 2004
- [3] **The snort project** - *Snort® Users Manual*. Sourcefire Inc, 2009
- [4] **Red Hat Product Documentation Team** - *Red Hat Enterprise Linux 4: Security Guide*. Red Hat Inc, 2008

3. Trang web tham khảo

- [1] http://www.windowsecurity.com/articles/Intrusion_Detection_Systems_IDS_Part_I_network_intrusions_attack_symptoms_IDS_tasks_and_IDS_architecture.html
- [2] <http://www.windowsecurity.com/articles/IDS-Part2-Classification-methods-techniques.html>
- [3] http://www.windowsecurity.com/articles/Hids_vs_Nids_Part2.html
- [4] http://www.openmaniak.com/inline_final.php
- [5] <http://www.focus.com/fyi/it-security/ids-vs-ips/>
- [6] <http://linuxgazette.net/117/savage.html>
- [8] <http://snort.org>
- [9] <http://sourcefire.com>
- [10] <http://hvaonline.net>