BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

-------o0o-------

ISO 9001:2015

ĐỒ ÁN TỐT NGHIỆP

NGÀNH CÔNG NGHỆ THÔNG TIN

HẢI PHÒNG 2019

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG
-------o0o-------

TÌM HIỂU HỆ THỐNG PHÁT HIỆN CẢNH BÁO NGUY CƠ TẤN CÔNG MẠNG

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY
Ngành: Công nghệ Thông tin

Sinh viên thực hiện: Phạm Quang Tuyến
Giáo viên hướng dẫn: TS Ngô Trường Giang
Mã số sinh viên: 1412101129

HẢI PHÒNG - 2019

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

-------o0o-------

NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP

Sinh viên: Phạm Quang Tuyến Mã số: 1412101129

Lớp: CT1802 Ngành: Công nghệ Thông tin

Tên đề tài: Tìm hiểu hệ thống phát hiện cảnh báo nguy cơ tấn công mạng

LỜI CẢM ƠN

Trong quá trình làm đồ án vừa qua, được sự giúp đỡ và chỉ bảo nhiệt tình của TS. Ngô Trường Giang – Trường Đại học Dân Lập Hải Phòng, đồ án của em đã được hoàn thành. Mặc dù đã cố gắng với sư tận tâm của thầy hướng dẫn song do thời gian và khả năng còn nhiều hạn chế nên đồ án không tránh khỏi những thiếu sót.

Em xin bày tỏ lòng biết ơn sâu sắc tới thầy Ngô Trường Giang đã tận tình hướng dẫn, chỉ bảo và dành rất nhiều thời gian quý báu của thầy cho em trong thời gian qua, đã giúp em hoàn thành đồ án đúng thời hạn.

Em xin cảm ơn các thầy cô giáo bộ môn khoa Công nghệ thông tin đã giảng dạy, trang bị cho em những kiến thức chuyên ngành, chuyên môn, chuyên sâu trong suốt 4 năm qua.

Xin cám ơn gia đình và bạn bè đã cổ vũ và động viên cho em trong suốt quá trình học tập cũng như thời gian làm đồ án, đã giúp em hoàn thành khóa học, đồ án theo quy định.

Em xin chân thành cảm ơn!

MỤC LỤC

DANH MỤC HÌNH VẼ

MỞ ĐẦU

Thế giới đang bắt đầu bước vào cuộc cách mạng công nghiệp lần thứ tư, một cuộc cách mạng sản xuất mới gắn liền với những đột phá chưa từng có về công nghệ, liên quan đến kết nối Internet, điện toán đám mây, in 3D, công nghệ cảm biến, thực tế ảo... Cuộc cách mạng sản xuất mới này được dự đoán sẽ tác động mạnh mẽ đến mọi quốc gia, chính phủ, doanh nghiệp và người dân khắp toàn cầu, cũng như làm thay đổi căn bản cách chúng ta sống, làm việc và sản xuất. Bên cạnh sự phát triển đó cũng tiềm ẩn những nguy cơ đe dọa đến mọi mặt của đời sống xã hội như việc đánh cắp thông tin, truy cập hệ thống trái phép, tấn công từ chối dịch vụ... Là nguy cơ mà người dùng Internet phải đương đầu.

Rất nhiều các giải pháp an ninh mạng đã được đưa ra và cũng đã có những đóng góp to lớn trong việc đảm bảo an toàn thông tin, ví dụ như: Firewall ngăn chặn những kết nối không đáng tin cậy, mã hóa làm tăng độ an toàn cho việc truyền dữ liệu, các chương trình diệt virus với cơ sở dữ liệu được cập nhật thường xuyên…

Tuy nhiên thực tế cho thấy chúng ta vẫn luôn thụ động trước các cuộc tấn công đặc biệt là các tấn công kiểu mới vì vậy yêu cầu đặt ra là cần có một hệ thống phát hiện và cảnh báo sớm trước các cuộc tấn công. Hệ thống phát hiện xâm nhập được xem như là một lựa chọn tối ưu.

Đồ án này trình bày về Hệ thống phát hiện cảnh báo nguy cơ tấn công mạng và tìm hiểu công cụ phát hiện cảnh báo nguy cơ tấn công mạng mã nguồn mở SNORT. Nội dung của đồ án bao gồm:

- Chương1: Tìm hiểu tổng quan giám sát an ninh mạng.
- Chương2: Tìm hiểu hệ thống phát hiện và chống xâm nhập mạng.
- Chương3: Ứng dụng phần mềm mã nguồn mở SNORT trong phát hiện xâm nhập mạng.

CHƯƠNG 1: TỔNG QUAN VỀ GIÁM SÁT AN NINH MẠNG.

1.1 Giám sát An ninh mạng.

Giám sát An ninh mạng là hệ thống được xây dựng nhằm mục đích thu thập, theo dõi, phân tích các sự kiện, dữ liệu ra vào mạng từ đó phát hiện các tấn công mạng và đưa ra cảnh báo cho hệ thống mạng được giám sát. Về bản chất đây là hệ thống phân tích sự kiện, luồng dữ liệu mà không tích hợp các giải pháp ngăn chặn vào trong đó. Hệ thống này hoạt động độc lập và chỉ thu thập nhật ký hệ thống của các thiết bị, ứng dụng hay các luồng dữ liệu chứ không ảnh hưởng đến chúng.

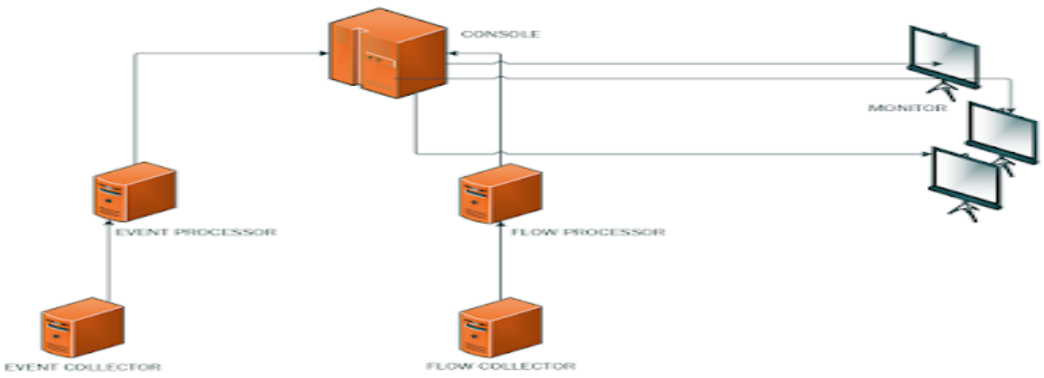
Trong các hệ thống thông tin, việc khắc phục các sự cố thường tốn một chi phí rất lớn. vì vậy, giải pháp giám sát mạng để phát hiện sớm các sự cố là một sự lựa chọn được nhiều người ưa thích nhằm mang lại hiệu quả cao với chi phí vừa phải.

1.2 Mô hình hệ thống và chức năng chính.

Về cơ bản hệ thống Giám sát an ninh mạng (GSANM) tuân thủ theo mô hình SIEM (Security Information and Event Management). Đây là mô hình chung cho hệ thống GSANM được sử dụng rất nhiều trên thế giới và các nhà sản xuất các thiết bị GSANM cũng dựa trên mô hình chuẩn này.

1.2.1 Các thành phần chính.

Hệ thống Giám sát an ninh mạng bao gồm các thành phần chính sau:



Hình 1-1: Thành phần của GSANM

CONSOLE:

Là nơi xử lý, lưu trữ các sự kiện an ninh được cảnh báo, các sự kiện này được gửi lên từ Event Processor và Flow Processor. Ngoài ra tại đây còn chứa các tập luật xử lý các dữ liệu, CONSOLE có khả năng hoạt động độc lập.

CONSOLE có hai giao diện, giao diện command line giúp người quản trị cấu hình, xử lý các lỗi hệ thống,... và giao diện web là nơi hiển thị các cảnh báo cũng như các sự kiện thu thập được. Các cảnh báo sẽ được lưu trữ tùy vào cấu hình quản trị trong bao lâu, thường là một năm cho mỗi hệ thống.

Năng lực hoạt động của CONSOLE tùy thuộc vào nhiều yếu tố như: Đường truyền mạng, cấu hình phần cứng, … thông thường hệ thống hoạt động với công suất 1000EPS và 100000FPM. Khi hệ thống GSANM được thiết lập và cấu hình thì CONSOLE sẽ tự động cấu hình tương ứng cho các thiết bị khác một cách chủ động sau khi kết nối vào các thiết bị thông qua cổng 22. Từ đó các việc cấu hình các thiết bị trong hệ thống GSANM có thể được thực hiện thông qua CONSOLE bằng hai cách đó là qua giao diện Web với cổng 443 hoặc qua giao diện command line.

EVENT PROCESSOR (EP):

Đây là nơi xử lý các sự kiện được gửi về từ Event Collector. Các sự kiện này sẽ được xử lý thông qua các tập luật tại đây. Nếu là cảnh báo hoặc các sự kiện từ các thiết bị an ninh đưa ra cảnh báo thì nó sẽ được gửi thẳng trực tiếp lên CONSOLE để xử lý. Nếu là các sự kiện không đưa ra cảnh báo sẽ được lưu trữ tại đây mà không chuyển lên CONSOLE.

Các sự kiện được lưu trữ tùy theo cấu hình của quản trị, thường là ba tháng cho các sự kiện không đưa ra cảnh báo. Các nhật ký hệ thống không đưa ra cảnh báo nó sẽ được quản lý qua giao diện web của CONSOLE.

FLOW PROCESSOR (FP):

Đây là nơi xử lý luồng dữ liệu, FP nhận dữ liệu từ Flow Collector và xử lý dựa trên các tập luật của FP. Sau đó, các cảnh báo sẽ được nó gửi lên CONSOLE còn các sự kiện không đưa ra cảnh báo sẽ được lưu trữ tại FP và được quản lý dựa trên giao diện web của CONSOLE. Thời gian lưu trữ các sự kiện này tùy thuộc vào cấu hình thường là ba tháng.

EVENT COLLECTOR (EC):

Là nơi thu thập nhật ký hệ thống, tiếp nhận các nhật ký hệ thống từ các thiết bị, hoặc các ứng dụng gửi về. Tại đây nhật ký hệ thống sẽ được mã hóa, nén và gửi về EP qua cổng 22. Sau đó nó sẽ được EP phân tích và xử lý.

Đối với EC có rất nhiều phương pháp lấy nhật ký hệ thống khác nhau VD: Cài đặt agent lên các máy tính cần thu thập và gửi nhật ký hệ thống đã được chỉ định về cho EC. Tại CONSOLE người quản trị sẽ cấu hình cho EC thu nhận các nhật ký hệ thống từ các agent này. Sau đó các nhật ký hệ thống này sẽ được quản lý dựa trên giao diện web của CONSOLE. EC chỉ có khả năng thu thập các sự kiện mà không có khả năng thu thập các luồng dữ liệu. Với một thiết bị, dịch vụ như IIS, thường có khoảng 20 sự kiện trên giây (20 EPS).

FLOW COLLECTOR (FC):

Đây là nơi thu thập các luồng dữ liệu từ mạng được giám sát. FC thường thu nhận luồng dữ liệu từ các switch có chức năng span port của Cisco. Sau đó dữ liệu cũng được nén, mã hóa và chuyển về FP xử lý thông qua cổng 22. CONSOLE sẽ cấu hình cho FC lắng nghe ở cổng Ethernet được kết nối với span port để thu thập dữ liệu. Khả năng xử lý hiện tại trên hệ thống GSANM đối với FC là 220000FPM.

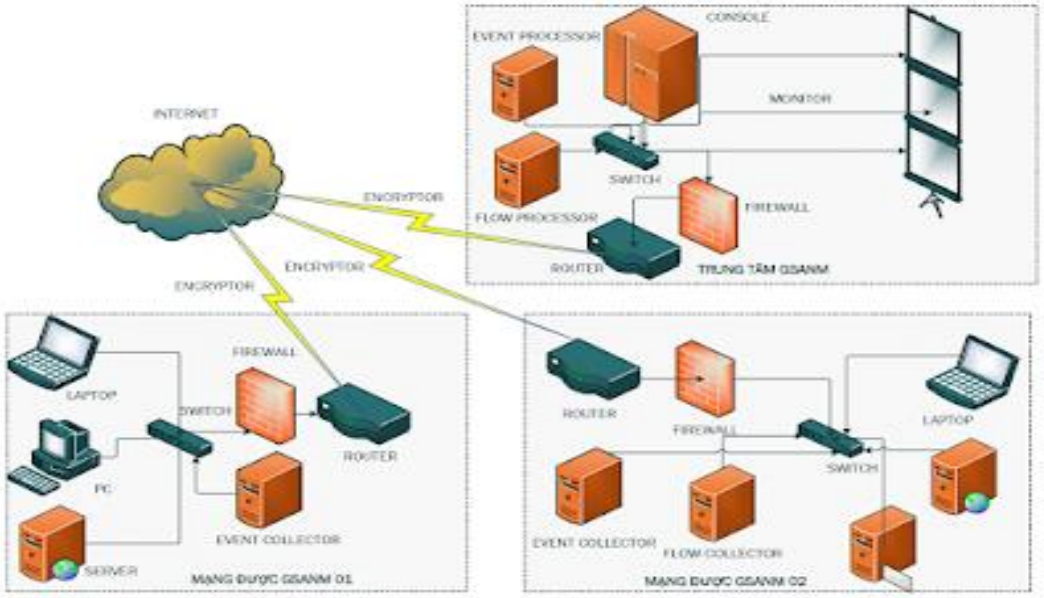
Các thiết bị phần cứng EC và FC của hệ thống GSANM có chức năng thu thập nhật ký hệ thống ở dạng "thô" là dạng chưa được phân tích. Đối với mỗi thiết bị này người quản trị hệ thống cần cung cấp địa chỉ IP tĩnh public,

sau đó việc trao đổi dữ liệu qua hệ thống sẽ được mã hóa, nén lại và gửi tới EP, FP để phân tích và xử lý thông qua cổng 22. CONSOLE sẽ hiển thị dữ liệu lên giao diện web để người quản trị có thể sẽ xem các cảnh báo này thông qua cổng 443.

1.2.2 Phân loại

Mô hình GSNAM được triển khai có hai dạng chính sau:

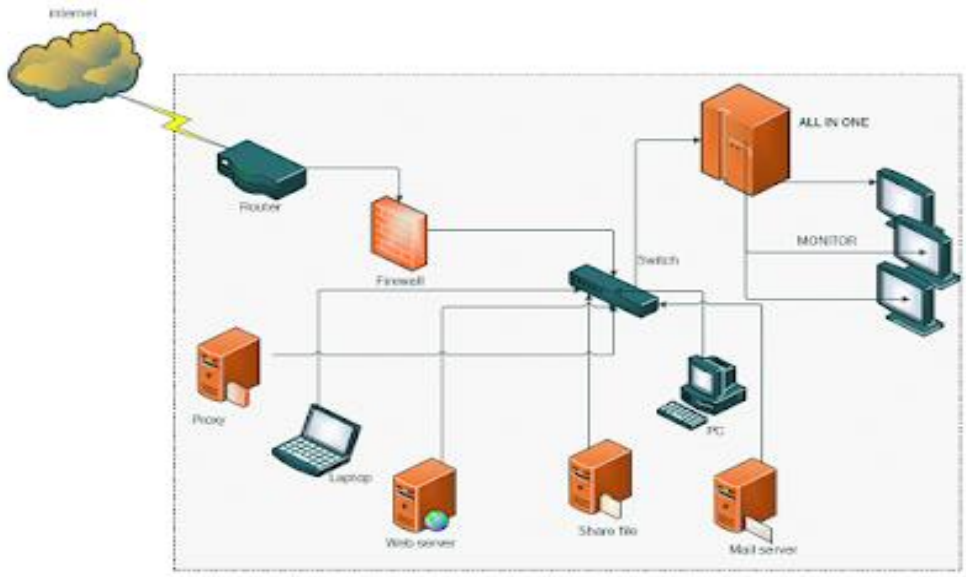
- Dạng phân tán (Distributed):



Hình 1-2: Mô hình GSANM phân tán

Là mô hình mà trong đó có hệ thống xử lý được đặt ở trung tâm GSANM và mọi hoạt động của hệ thống như: Các sự kiện, luồng dữ liệu, …sẽ được xử lý tại trung tâm sau đó được hiển thị lên giao diện Web site. Đối với mô hình này thường đòi hỏi một sự đầu tư quy mô và lực lượng con người phải nhiều mới đủ khả năng để vận hành hệ thống này.

- Dạng hoạt động độc lập (All in one):



Hình 1-3: Mô hình GSANM tập trung.

Đây là mô hình mà hệ thống được xây dựng riêng lẻ cho các đơn vị, và không liên quan tới nhau, có nghĩa là hệ thống hoạt động độc lập. Các nhật ký hệ thống và luồng dữ liệu được trực tiếp thu thập tại mạng con, sau đó đẩy về thiết bị GSANM và tại đây luồng dữ liệu sẽ được xử lý. Tuy nhiên, mô hình này phù hợp cho các ngân hàng và đơn vị nhỏ và yêu cầu về đầu tư và lực lượng con người không cao.

1.2.3 Chức năng

Đối với hệ thống GSANM chức năng chính của nó là sẽ thu thập các thành phần sau:

- Các sự kiện an ninh (Securtity Event): Được sinh ra từ các ứng dụng hoặc thiết bị như: Nhật ký hệ thống IIS, Firewall, VPN (Virtual Private Network), IDS (Intrusion Detection System), IPS (Intrusion Prevention System), …
- Bối cảnh hoạt động mạng (Network activity context): Tầng 7 bối cảnh ứng dụng từ lưu lượng mạng và lưu lượng các ứng dụng.
- Thông tin hệ điều hành: Tên nhà sản xuất và chi tiết về số phiên bản.

- Các nhật ký hệ thống ứng dụng: Kế hoạch nguồn lực doanh nghiệp (Enterprise Resource Planning – ERP), quy trình làm việc, cơ sở dữ liệu ứng dụng, nền tảng quản lý,...

Với mỗi dòng nhật ký hệ thống sinh được tính là một sự kiện, các sự kiên được tính trên giây (EPS), và xử lý các luồng dữ liệu này được tính trên phút (FPM) sau đó hệ thống sẽ tiến hành phân tích bằng các bộ luật và đưa ra các cảnh báo cần thiết tới nhà quản trị hệ thống.

1.3 Phát hiện và chống xâm nhập mạng

1.3.1 Hệ thống phát hiện xâm nhập (IDS).

IDS (Intrusion Detection Systems) là một hệ thống phòng chống nhằm phát hiện các hành động tấn công vào một mạng mục đích của nó là phát hiện và ngăn ngừa các hành động phá hoại đối với vấn đề bảo mật hệ thống hoặc những hành động trong tiến trình tấn công như sưu tập, quét các cổng một tính năng chính của hệ thống này là cung cấp thông tin nhận biết về những hành động không bình thường và đưa ra các báo cảnh thông báo cho quản trị viên mạng khóa các kết nối đang tấn công này thêm vào đó công cụ IDS cũng có thể phân biệt giữa những tấn công bên trong từ bên trong tổ chức (từ chính nhân viên hoặc khách hàng) và tấn công bên ngoài (tấn công từ hacker).

1.3.2 Hệ thống chống xâm nhập (IPS).

IPS (Intrusion Prevention Systems) là hệ thống theo dõi, ngăn ngừa kịp thời các hoạt động xâm nhập không mong muốn.

Chức năng chính của IPS là xác định các hoạt động nguy hại, lưu giữ các thông tin này. Sau đó kết hợp với firewall để dừng ngay các hoạt động này, và cuối cùng đưa ra các báo cáo chi tiết về các hoạt động xâm nhập trái phép trên.

Hệ thống IPS được xem là trường hợp mở rộng của hệ thống IDS, cách thức hoạt động cũng như đặc điểm của 2 hệ thống này tương tự nhau. Điểm khác nhau duy nhất là hệ thống IPS ngoài khả năng theo dõi, giám sát thì còn

có chức năng ngăn chặn kịp thời các hoạt động nguy hại đối với hệ thống. Hệ thống IPS sử dụng tập luật tương tự như hệ thống IDS.

1.3.3 Nguyên lý hoạt động hệ thống

Nguyên lý hoạt động của một hệ thống phát hiện và chống xâm nhập được chia làm 5 giai đoạn chính: **Giám sát mạng, phân tích lưu thông, Liên lạc giữa các thành phần**, **Cảnh báo** về các hành vi xâm nhập và cuối cùng có thể tiến hành **phản ứng** lại tùy theo chức năng của từng IDS.

1.3.3.1 Giám sát mạng (monotoring)

Giám sát mạng là quá trình thu thập thông tin về lưu thông trên mạng. Việc này thông thường được thực hiện bằng các Sensor. Yêu cầu đòi hỏi đối với giai đoạn này là có được thông tin đầy đủ và toàn vẹn về tình hình mạng. Đây cũng là một vấn đề khó khăn, bởi vì nếu theo dõi toàn bộ thông tin thì sẽ tốn khá nhiều tài nguyên, đồng thời gây ra nguy cơ tắc nghẽn mạng. Nên cần thiết phải cân nhắc để không làm ảnh hưởng đến toàn bộ hệ thống. Có thể sử dụng phương án là thu thập liên tục trong khoảng thời gian dài hoặc thu thập theo từng chu kì. Tuy nhiên khi đó những hành vi bắt được chỉ là những hành vi trong khoảng thời gian giám sát. Hoặc có thể theo vết những lưu thông TCP theo gói hoặc theo liên kết. Bằng cách này sẽ thấy được những dòng dữ liệu vào ra được phép. Nhưng nếu chỉ theo dõi những liên kết thành công sẽ có thể bỏ qua những thông tin có giá trị về những liên kết không thành công mà đây lại thường là những phần quan tâm trong một hệ thống IDS, ví dụ như hành động quét cổng.

1.3.3.2 Phân tích lưu thông (Analyzing)

Khi đã thu thập được những thông tin cần thiết từ những điểm trên mạng. IDS tiến hành phân tích những dữ liệu thu thập được. Mỗi hệ thống cần có một sự phân tích khác nhau vì không phải môi trường nào cũng giống nhau. Thông thường ở giai đoạn này, hệ thống IDS sẽ dò tìm trong dòng

traffic mang những dấu hiệu đáng nghi ngờ dựa trên kỹ thuật đối sánh mẫu hoặc phân tích hành vi bất thường.

1.3.3.3 Liên lạc

Giai đoạn này giữ một vai trò quan trọng trong hệ thống IDS. Việc liên lạc diễn ra khi Sensor phát hiện ra dấu hiệu tấn công hoặc Bộ xử lý thực hiện thay đổi cấu hình, điều khiển Sensor. Thông thường các hệ thống IDS sử dụng các bộ giao thức đặc biệt để trao đổi thông tin giữa các thành phần. Các giao thức này phải đảm bảo tính tin cậy, bí mật và chịu lỗi tốt, ví dụ: SSH, HTTPS, SNMPv3…Chẳng hạn hệ thống IDS của hãng Cisco thường sử dụng giao thức PostOffice định nghĩa một tập các thông điệp để giao tiếp giữa các thành phần.

1.3.3.4 Cảnh báo (Alert)

Sau khi đã phân tích xong dữ liệu, hệ thống IDS cần phải đưa ra được những cảnh báo. Ví dụ như:

- Cảnh báo địa chỉ không hợp lệ.
- Cảnh báo khi máy cố gắng kết nối đến những máy nằm trong danh sách cần theo dõi ở trong hay ngoài mạng.

1.3.3.5 Phản ứng (Response)

Trong một số hệ thống IDS tiên tiến hiện nay, sau khi các giai đoạn trên phát hiện được dấu hiệu tấn công, hệ thống không những cảnh báo cho người quản trị mà còn đưa ra các hành vi phòng vệ ngăn chặn hành vi tấn công đó. Điều này giúp tăng cường khả năng tự vệ của Mạng, vì nếu chỉ cần cảnh báo cho người quản trị thì đôi khi cuộc tấn công sẽ tiếp tục xảy ra gây ra các tác hại xấu. Một hệ thống IDS có thể phản ứng lại trước những tấn công phải được cấu hình để có quyền can thiệp vào hoạt động của Firewall, Switch và Router. Các hành động mà IDS có thể đưa ra như:

- Ngắt dịch vụ.

- Gián đoạn phiên.
- Cấm địa chỉ IP tấn công.
- Tạo log.

CHƯƠNG 2: HỆ THỐNG PHÁT HIỆN XÂM NHẬP MẠNG

Nếu như hiểu Firewall là một hệ thống "khóa" chốt chặn ở cửa ngõ mạng, thì hệ thống IDS có thể được coi như các "cảm ứng giám sát" được đặt khắp nơi trong mạng để cảnh báo về các cuộc tấn công đã "qua mặt" được Firewall hoặc xuất phát từ bên trong mạng. Một IDS có nhiệm vụ phân tích các gói tin mà Firewall cho phép đi qua, tìm kiếm các dấu hiệu tấn công từ các dấu hiệu đã biết hoặc thông qua việc phân tích các sự kiện bất thường, từ đó ngăn chặn các cuộc tấn công trước khi nó có thể gây ra những hậu quả xấu với tổ chức.

Cách đây khoảng 25 năm, khái niệm phát hiện xâm nhập xuất hiện qua một bài báo của James Anderson khi đó người ta cần IDS với mục đích là dò tìm và nghiên cứu các hành vi bất thường và thái độ của người sử dụng trong mạng, phát hiện ra các việc làm dụng đặc quyền để giám sát tài sản hệ thống mạng. Các nghiên cứu về hệ thống phát hiện xâm nhập được nghiên cứu chính thức từ năm 1983 đến năm 1988 trước khi được sử dụng tại mạng máy tính của không lực Hoa Kỳ. Cho đến tận năm 1996, các khái niệm IDS vẫn chưa được phổ biến, một số hệ thống IDS chỉ được xuất hiện trong các phòng thí nghiệm và viện nghiên cứu. Tuy nhiên trong thời gian này một số công nghệ IDS bắt đầu phát triển dựa trên sự bùng nổ của công nghệ thông tin đến năm 1997 IDS mới được biết đến rộng rãi và thực sự đem lại lợi nhuận với sự đi đầu của công ty ISS, một năm sau đó, Cisco nhận ra tầm quan trọng của IDS và đã mua lại một công ty cung cấp giải pháp IDS tên là Wheel. Hiện tại, các thống kê cho thấy IDS/IPS đang là một trong các công nghệ an ninh được sử dụng nhiều nhất và vẫn còn phát triển.

2.1 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.

2.1.1 Chính sách của IDS.

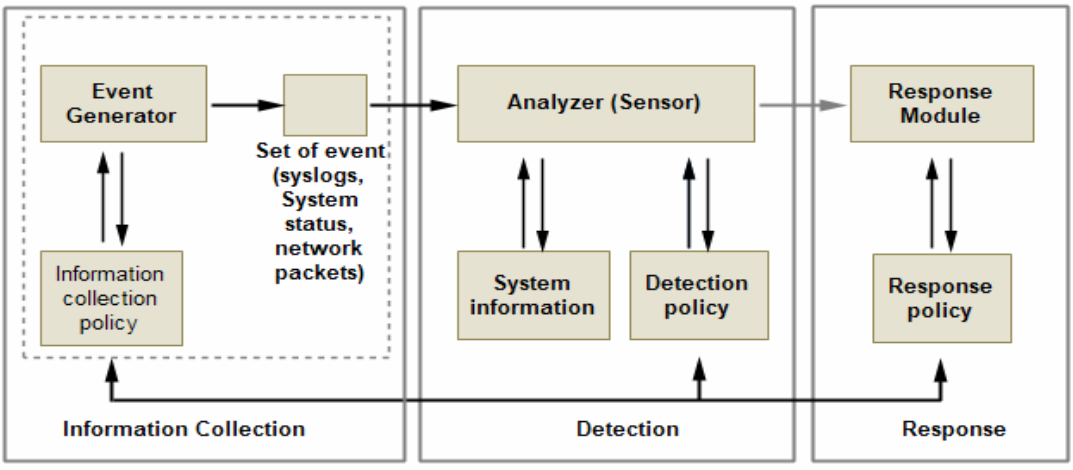
Trước khi cài đặt một hệ thống IDS lên hệ thống thì cần phải có một chính sách để phát hiện kẻ tấn công và cách xử lý khi phát hiện ra các hoạt động tấn công bằng cách nào đó chúng phải được áp dụng các chính sách cần chứa các phần sau (có thể thêm tùy theo yêu cầu của từng hệ thống):

- 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 Open View hoặc My SQL 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.

2.1.2 Kiến trúc 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 (response). 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 2-1: 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 số chính sách này cùng với thông tin chính sách có thể được lưu trong hệ thống được bảo vệ hoặc bên ngoài.

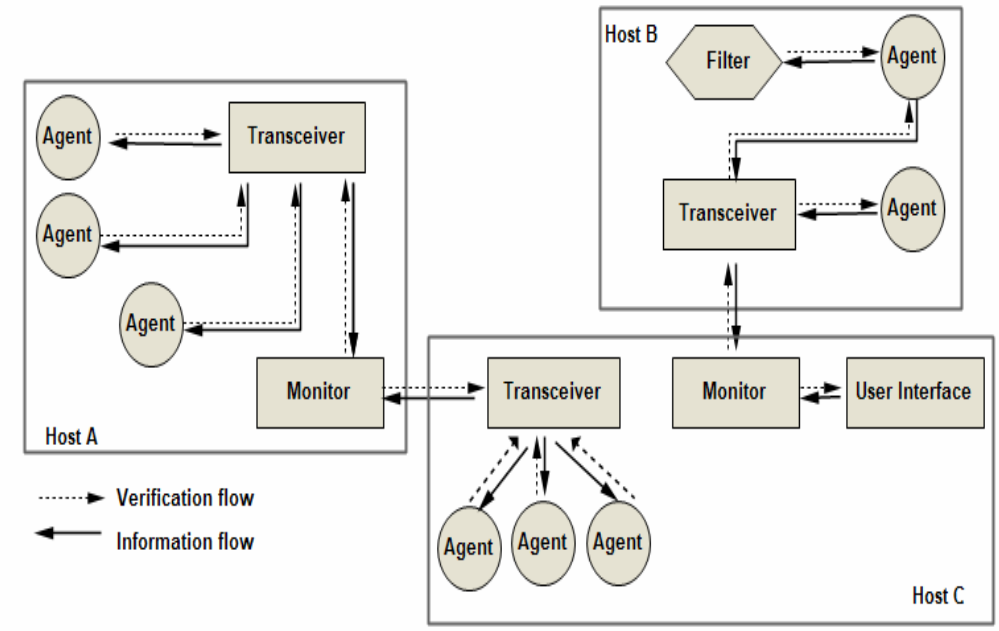
Vai trò của bộ cảm biến là dùng để lọc thông tin và loại bỏ dữ liệu không tương thích đạt được từ các sự kiện liên quan với hệ thống bảo vệ, vì vậy có thể phát hiện được các hành động nghi ngờ bộ phân tích sử dụng cơ sở dữ liệu chính sách phát hiện cho mục này. Ngoài ra còn có các thành phần: dấu hiệu 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 tạo 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 2-2: Giải pháp kiến trúc đa tác nhân

2.1.3 Phân loại hệ thống phát hiện xâm nhập.

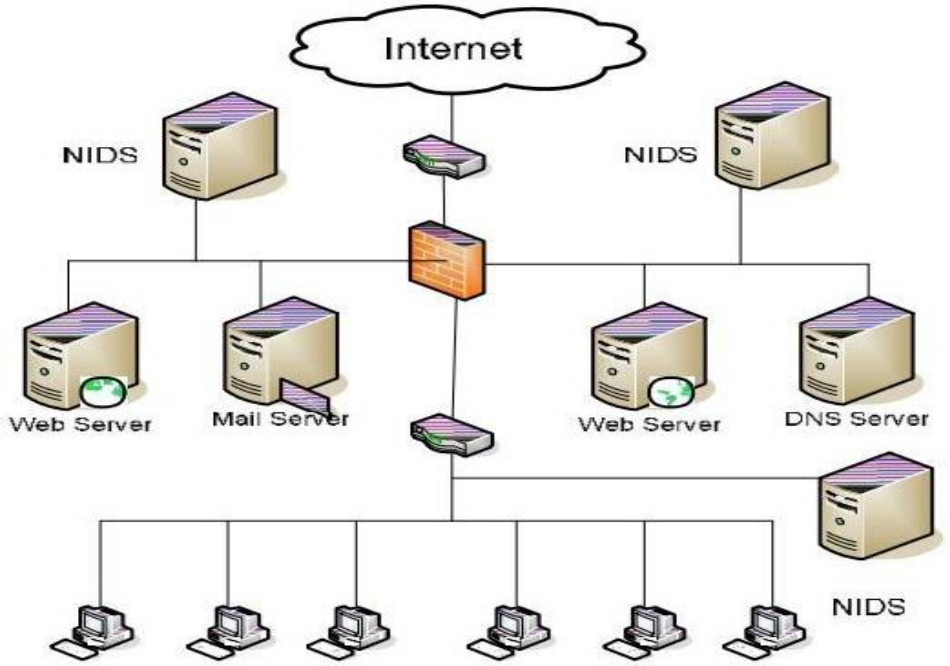
Cách thông thường nhất để phân loại các hệ thống IDS là dựa vào đặc điểm của nguồn dữ liệu thu thập được. Trong trường hợp này, các hệ thống IDS được chia thành các loại sau:

- Network-based IDS (NIDS): Sử dụng dữ liệu trên toàn bộ lưu thông mạng, cùng với dữ liệu kiểm tra từ một hoặc một vài máy trạm để phát hiện xâm nhập.
- Host-based IDS (HIDS): Sử dụng dữ liệu kiểm tra từ một máy trạm đơn để phát hiện xâm nhập.

2.1.3.1 Giám sát toàn bộ mạng NIDS (Network based IDS)

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.

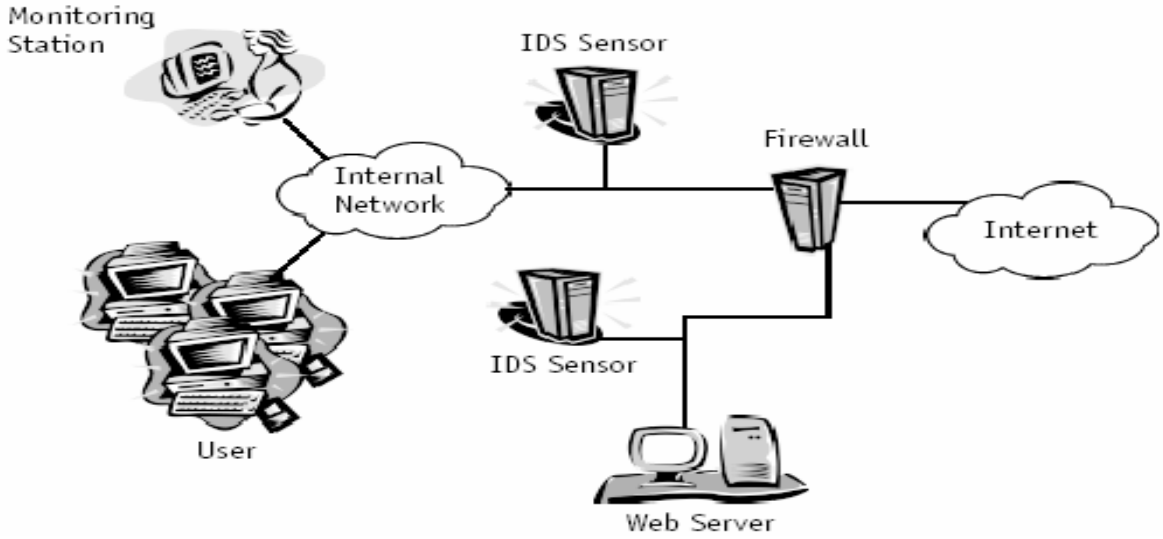
Trong hình thức này NIDS xác định các truy cập trái phép bằng việc giám sát các hoạt động mạng được tiến hành trên toàn bộ các phân mạng của hệ thống, NIDS sử dụng bộ dò và bộ cảm biến cài đặt trên toàn mạng. Những bộ dò này theo dõi trên mạng nhằm tìm kiếm những lưu lượng trùng với những mô tả sơ lược được định nghĩa hay là những dấu hiệu. Khi ghi nhận được một mẫu lưu lượng hay dấu hiệu, bộ cảm biến gửi tín hiệu cảnh báo đến trung tâm điều khiển và có thể được cấu hình nhằm tìm ra biện pháp ngăn chặn những xâm nhập xa hơn. NIDS là tập nhiều sensor được đặt ở toàn mạng để theo dõi những gói tin trong mạng, so sánh với các mẫu đã được định nghĩa để phát hiện đó là tấn công hay không.



Hình 2-3: Mô hình triển khai hệ thống NIDS

NIDS thường bao gồm có hai thành phần logic:

- Bộ cảm biến – Sensor: đặt tại một đoạn mạng, kiểm soát các cuộc lưu thông nghi ngờ trên đoạn mạng đó.
- Trạm quản lý: nhận các tín hiệu cảnh báo từ bộ cảm biến và thông báo cho một điều hành viên.



Hình 2-4: Mô hình NIDS

Một NIDS truyền thống với hai bộ cảm biến trên các đoạn mạng khác nhau cùng giao tiếp với một trạm kiểm soát.

Ưu điểm

- Chi phí thấp: Do chỉ cần cài đặt NIDS ở những vị trí trọng yếu là có thể giám sát lưu lượng toàn mạng nên hệ thống không cần phải nạp các phần mềm và quản lý trên các máy toàn mạng.
- Phát hiện được các cuộc tấn công mà HIDS bỏ qua: Khác với HIDS, NIDS kiểm tra header của tất cả các gói tin vì thế nó không bỏ sót các dấu hiệu xuất phát từ đây. Ví dụ: nhiều cuộc tấn công DoS, TearDrop (phân nhỏ) chỉ bị phát hiện khi xem header của các gói tin lưu chuyển trên mạng.
- Khó xoá bỏ dấu vết (evidence): Các thông tin lưu trong log file có thể bị kẻ đột nhập sửa đổi để che dấu các hoạt động xâm nhập, trong tình huống này HIDS khó có đủ thông tin để hoạt động. NIDS sử dụng lưu thông hiện hành trên mạng để phát hiện xâm nhập. Vì thế, kẻ đột nhập không thể xoá bỏ được các dấu vết tấn công. Các thông tin bắt được không chỉ chứa cách thức tấn công mà cả thông tin hỗ trợ cho việc xác minh và buộc tội kẻ đột nhập.
- Phát hiện và đối phó kịp thời: NIDS phát hiện các cuộc tấn công ngay khi xảy ra, vì thế việc cảnh báo và đối phó có thể thực hiện được nhanh hơn. VD: Một hacker thực hiện tấn công DoS dựa trên TCP có thể bị NIDS phát hiện và ngăn chặn ngay bằng việc gửi yêu cầu TCP reset nhằm chấm dứt cuộc tấn công trước khi nó xâm nhập và phá vỡ máy bị hại.
- Có tính độc lập cao: Lỗi hệ thống không có ảnh hưởng đáng kể nào đối với công việc của các máy trên mạng. Chúng chạy trên một hệ thống chuyên dụng dễ dàng cài đặt; đơn thuần chỉ mở thiết bị ra, thực hiện

một vài sự thay đổi cấu hình và cắm chúng vào trong mạng tại một vị trí cho phép nó kiểm soát các cuộc lưu thông nhạy cảm.

Nhược điểm

- Bị hạn chế với Switch: Nhiều lợi điểm của NIDS không phát huy được trong các mạng chuyển mạch hiện đại. Thiết bị switch chia mạng thành nhiều phần độc lập vì thế NIDS khó thu thập được thông tin trong toàn mạng. Do chỉ kiểm tra mạng trên đoạn mà nó trực tiếp kết nối tới, nó không thể phát hiện một cuộc tấn công xảy ra trên các đoạn mạng khác. Vấn đề này dẫn tới yêu cầu tổ chức cần phải mua một lượng lớn các bộ cảm biến để có thể bao phủ hết toàn mạng gây tốn kém về chi phí cài đặt.
- Hạn chế về hiệu năng: NIDS sẽ gặp khó khăn khi phải xử lý tất cả các gói tin trên mạng rộng hoặc có mật độ lưu thông cao, dẫn đến không thể phát hiện các cuộc tấn công thực hiện vào lúc "cao điểm". Một số nhà sản xuất đã khắc phục bằng cách cứng hoá hoàn toàn IDS nhằm tăng cường tốc độ cho nó. Tuy nhiên, do phải đảm bảo về mặt tốc độ nên một số gói tin được bỏ qua có thể gây lỗ hổng cho tấn công xâm nhập.
- Tăng thông lượng mạng: Một hệ thống phát hiện xâm nhập có thể cần truyền một dung lượng dữ liệu lớn trở về hệ thống phân tích trung tâm, có nghĩa là một gói tin được kiểm soát sẽ sinh ra một lượng lớn tải phân tích. Để khắc phục người ta thường sử dụng các tiến trình giảm dữ liệu linh hoạt để giảm bớt số lượng các lưu thông được truyền tải. Họ cũng thường thêm các chu trình tự ra các quyết định vào các bộ cảm biến và sử dụng các trạm trung tâm như một thiết bị hiển thị trạng thái hoặc trung tâm truyền thông hơn là thực hiện các phân tích thực tế. Điểm bất lợi là nó sẽ cung cấp rất ít thông tin liên quan cho các bộ cảm biến; bất kỳ bộ cảm biến nào sẽ không biết được việc một bộ cảm biến khác dò

được một cuộc tấn công. Một hệ thống như vậy sẽ không thể dò được các cuộc tấn công hiệp đồng hoặc phức tạp.

- Một hệ thống NIDS thường gặp khó khăn trong việc xử lý các cuộc tấn công trong một phiên được mã hoá. Lỗi này càng trở nên trầm trọng khi nhiều công ty và tổ chức đang áp dụng mạng riêng ảo VPN.
- Một số hệ thống NIDS cũng gặp khó khăn khi phát hiện các cuộc tấn công mạng từ các gói tin phân mảnh. Các gói tin định dạng sai này có thể làm cho NIDS hoạt động sai và đổ vỡ.

2.1.3.2 Giám sát máy tính đơn lẻ HIDS (Host based IDS)

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.

Host-based IDS tìm kiếm dấu hiệu của xâm nhập vào một host cục bộ; thường sử dụng các cơ chế kiểm tra và phân tích các thông tin được logging. Nó tìm kiếm các hoạt động bất thường như login, truy nhập file không thích hợp, bước leo thang các đặc quyền không được chấp nhận.

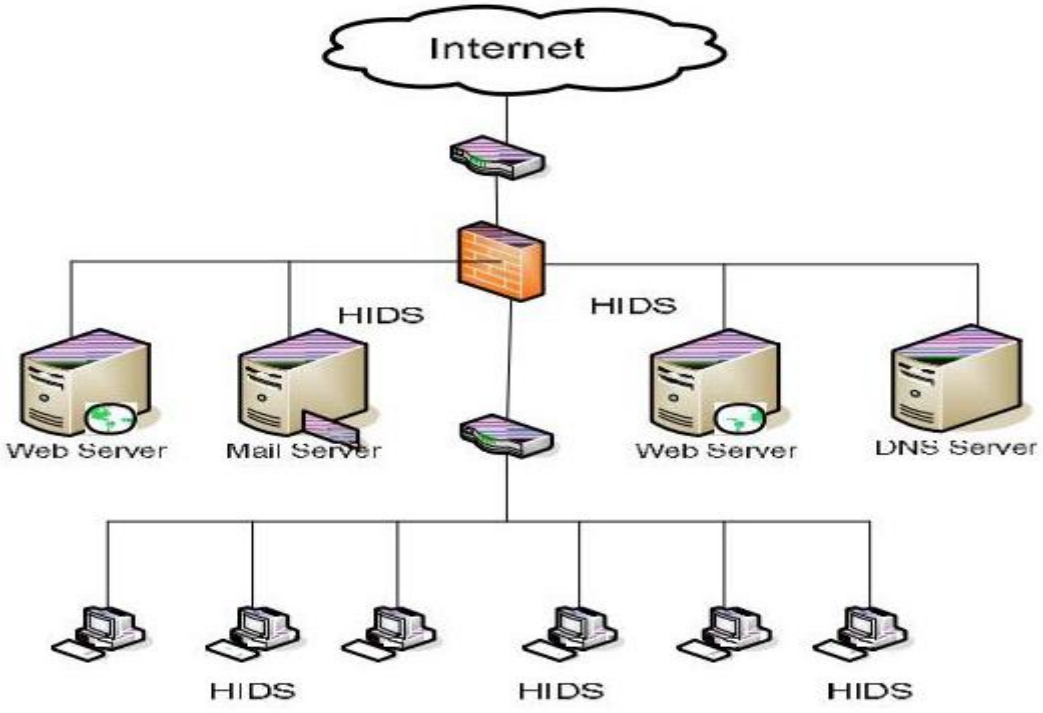
Kiến trúc IDS này thường dựa trên các luật (rule-based) để phân tích các hoạt động. Ví dụ đặc quyền của người sử dụng cấp cao chỉ có thể đạt được thông qua lệnh su-select user, như vậy những cố gắng liên tục để login vào account root có thể được coi là một cuộc tấn công.

Bằng cách cài đặt một phần mềm trên tất cả các máy tính chủ, IDS dựa trên máy chủ quan sát tất cả những hoạt động hệ thống, như các file log và những lưu lượng mạng thu thập được.

Hệ thống dựa trên máy chủ cũng theo dõi OS, những cuộc gọi hệ thống, lịch sử sổ sách (audit log) và những thông điệp báo lỗi trên hệ thống máy chủ. Trong khi những đầu dò của mạng có thể phát hiện một cuộc tấn công, thì chỉ có hệ thống dựa trên máy chủ mới có thể xác định xem cuộc tấn công có thành công hay không.

HIDS thường được cài đặt trên một máy tính nhất định. Thay vì giám sát hoạt động của một Network segment, HIDS chỉ giám sát các hoạt động trên một máy tính. Nó thường được đặt trên các Host xung yếu của tổ chức, và các server trong vùng DMZ. thường là mục tiêu tấn công đầu tiên. Nhiệm vụ chính của HIDS là giám sát các thay đổi trên hệ thống, bao gồm:

- Các tiến trình.
- Các entry của Registry.
- Mức độ sử dụng CPU.
- Các thông số này khi vượt qua một ngưỡng định trước hoặc những thay đổi khả nghi trên hệ thống file sẽ gây ra báo động.



Hình 2-5: Mô hình hệ thống HIDS

Ưu điểm

- Xác định được kết quả của cuộc tấn công: Do HIDS sử dụng dữ liệu log lưu các sự kiện xảy ra, nó có thể biết được cuộc tấn công là thành công hay thất bại với độ chính xác cao hơn NIDS. Vì thế, HIDS có thể bổ sung thông tin tiếp theo khi cuộc tấn công được sớm phát hiện với NIDS.
- Giám sát được các hoạt động cụ thể của hệ thống: HIDS có thể giám sát các hoạt động mà NIDS không thể như: truy nhập file, thay đổi quyền, các hành động thực thi, truy nhập dịch vụ được phân quyền. Đồng thời nó cũng giám sát các hoạt động chỉ được thực hiện bởi người quản trị. Vì thế, hệ thống host-based IDS có thể là một công cụ cực mạnh để phân tích các cuộc tấn công có thể xảy ra do nó thường cung cấp nhiều thông tin chi tiết và chính xác hơn một hệ network-based IDS.
- Phát hiện các xâm nhập mà NIDS bỏ qua: chẳng hạn kẻ đột nhập sử dụng bàn phím xâm nhập vào một server sẽ không bị NIDS phát hiện.
- Thích nghi tốt với môi trường chuyển mạch, mã hoá: Việc chuyển mạch và mã hoá thực hiện trên mạng và do HIDS cài đặt trên máy nên nó không bị ảnh hưởng bởi hai kỹ thuật trên.
- Không yêu cầu thêm phần cứng: Được cài đặt trực tiếp lên hạ tầng mạng có sẵn (FTP Server, WebServer) nên HIDS không yêu cầu phải cài đặt thêm các phần cứng khác.

Nhược điểm

- Khó quản trị: các hệ thống host-based yêu cầu phải được cài đặt trên tất cả các thiết bị đặc biệt mà bạn muốn bảo vệ. Đây là một khối lượng công việc lớn để cấu hình, quản lí, cập nhật.
- Thông tin nguồn không an toàn: một vấn đề khác kết hợp với các hệ thống host-based là nó hướng đến việc tin vào nhật ký mặc định và

năng lực kiểm soát của server. Các thông tin này có thể bị tấn công và đột nhập dẫn đến hệ thống hoạt đông sai, không phát hiện được xâm nhập.

- Hệ thống host-based tương đối đắt: nhiều tổ chức không có đủ nguồn tài chính để bảo vệ toàn bộ các đoạn mạng của mình sử dụng các hệ thống host-based. Những tổ chức đó phải rất thận trọng trong việc chọn các hệ thống nào để bảo vệ. Nó có thể để lại các lỗ hổng lớn trong mức độ bao phủ phát hiện xâm nhập. Ví dụ như một kẻ tấn công trên một hệ thống láng giềng không được bảo vệ có thể đánh hơi thấy các thông tin xác thực hoặc các tài liệu dễ bị xâm phạm khác trên mạng.
- Chiếm tài nguyên hệ thống: Do cài đặt trên các máy cần bảo vệ nên HIDS phải sử dụng các tài nguyên của hệ thống để hoạt động như: bộ vi xử lí, RAM, bộ nhớ ngoài.
- HIDS không có khả năng phát hiện các cuộc dò quét mạng (Nmap, Netcat…)

2.1.3.3 So sánh giữa NIDS và HIDS:

Bảng 2-1: 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 như 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 (Inter-net)	**	**	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 cli-ent	****	-	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 đề 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
	***	***	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

2.2 Tổng quan về snort.

2.2.1 Giới thiệu

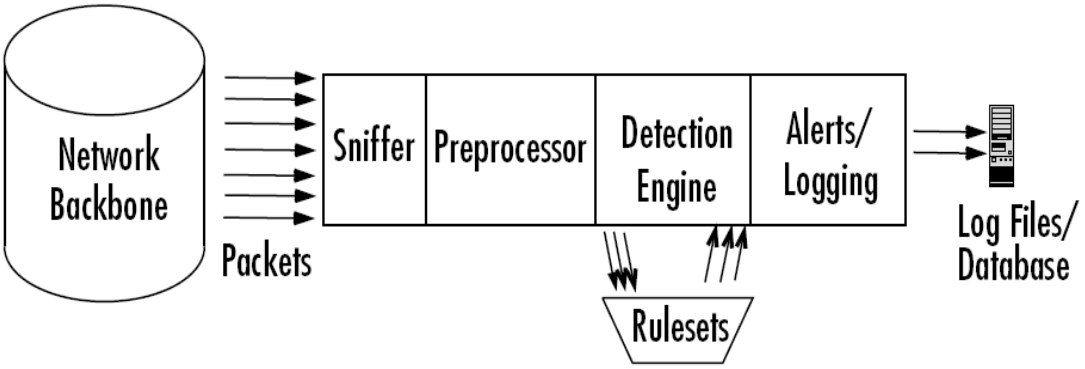
Snort là một NIDS được Martin Roesh phát triển dưới mô hình mã nguồn mở. Tuy Snort miễn phí nhưng nó lại có rất nhiều tính năng tuyệt vời mà không phải sản phẩm thương mại nào cũng có thể có được. Với kiến trúc thiết kế theo kiểu module, người dùng có thể tự tăng cường tính năng cho hệ thống Snort của mình bằng việc cài đặt hay viết thêm mới các module. Cơ sở dữ liệu luật của Snort đã lên tới 2930 luật và được cập nhật thường xuyên bởi một cộng đồng người sử dụng.

Bên cạnh việc có thể hoạt động như một ứng dụng thu bắt gói tin thông thường, Snort còn có thể được cấu hình để chạy như một NIDS. Snort hỗ trợ khả năng hoạt động trên các giao thức sau: Ethernet, 802.11,Token Ring, FDDI, Cisco HDLC, SLIP, PPP, và PF của OpenBSD.

2.2.2 Kiến trúc của snort

Snort bao gồm nhiều thành phần, với mỗi phần có một chức năng riêng. Các phần chính đó là:

- Môđun giải mã gói tin (Packet Decoder)
- Môđun tiền xử lý (Preprocessors)
- Môđun phát hiện (Detection Engine)
- Môđun log và cảnh báo (Logging and Alerting System)
- Môđun kết xuất thông tin (Output Module)
- Kiến trúc của Snort được mô tả trong hình sau:

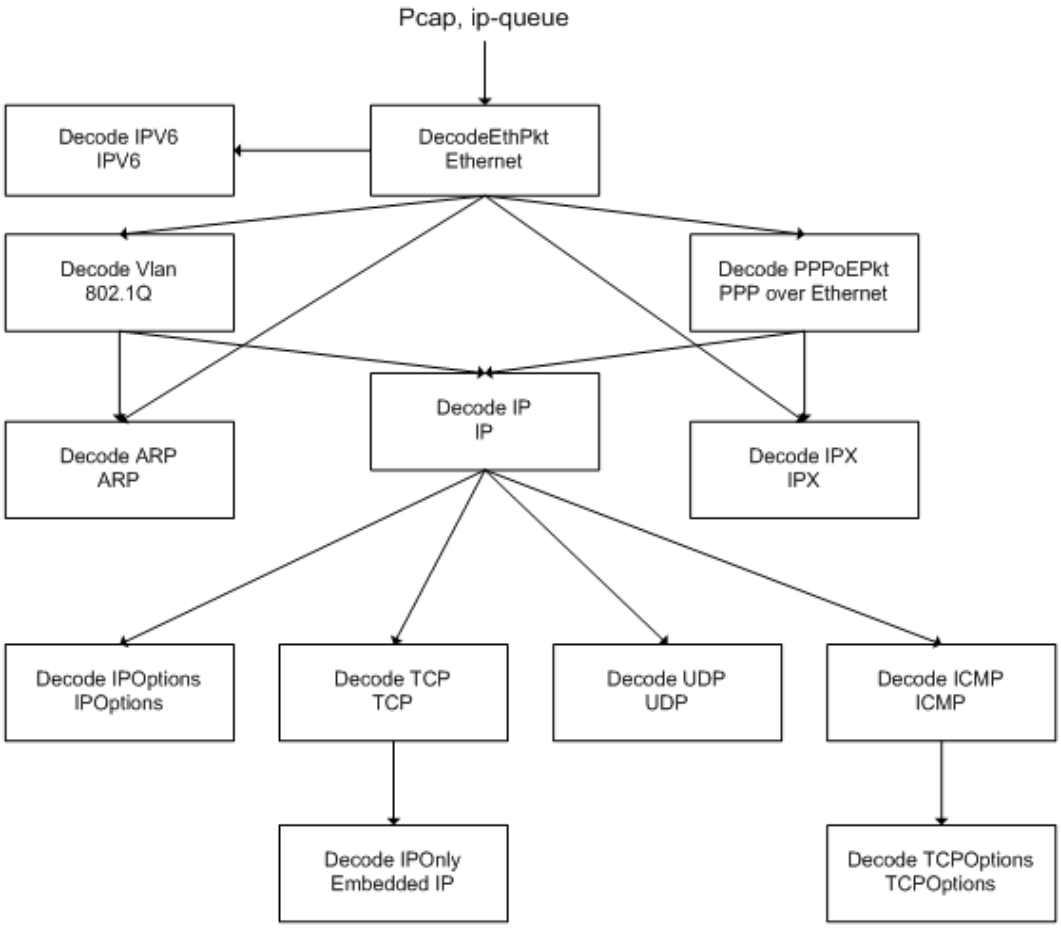


Hình 2-6: Mô hình kiến trúc hệ thống Snort

Khi Snort hoạt động nó sẽ thực hiện việc lắng nghe và thu bắt tất cả các gói tin nào di chuyển qua nó. Các gói tin sau khi bị bắt được đưa vào Môđun Giải mã gói tin. Tiếp theo gói tin sẽ được đưa vào môđun Tiền xử lý, rồi môđun Phát hiện. Tại đây tùy theo việc có phát hiện được xâm nhập hay không mà gói tin có thể được bỏ qua để lưu thông tiếp hoặc được đưa vào môđun Log và cảnh báo để xử lý. Khi các cảnh báo được xác định môđun Kết xuất thông tin sẽ thực hiện việc đưa cảnh báo ra theo đúng định dạng mong muốn. Sau đây ta sẽ đi sâu vào chi tiết hơn về cơ chế hoạt động và chức năng của từng thành phần.

2.2.2.1 Modun giải mã gói tin.

Snort sử dụng thư viện pcap để bắt mọi gói tin trên mạng lưu thông qua hệ thống. Hình sau mô tả việc một gói tin Ethernet sẽ được giải mã thế nào:



Hình 2-7: Xử lý một gói tin Ethernet

Một gói tin sau khi được giải mã sẽ được đưa tiếp vào môđun tiền xử lý.

2.2.2.2 Modun tiền xử lý

Môđun tiền xử lý là một môđun rất quan trọng đối với bất kỳ một hệ thống IDS nào để có thể chuẩn bị gói dữ liệu đưa và cho môđun Phát hiện phân tích. Ba nhiệm vụ chính của các môđun loại này là:

Kết hợp lại các gói tin: Khi một lượng dữ liệu lớn được gửi đi, thông tin sẽ không đóng gói toàn bộ vào một gói tin mà phải thực hiện việc phân mảnh, chia gói tin ban đầu thành nhiều gói tin rồi mới gửi đi. Khi Snort nhận được các gói tin này nó phải thực hiện việc ghép nối lại để có được dữ liệu nguyên dạng ban đầu, từ đó mới thực hiện được các công việc xử lý tiếp. Như ta đã biết khi một phiên làm việc của hệ thống diễn ra, sẽ có rất nhiều gói tin được trao đổi trong phiên đó. Một gói tin riêng lẻ sẽ không có trạng thái và

nếu công việc phát hiện xâm nhập chỉ dựa hoàn toàn vào gói tin đó sẽ không đem lại hiệu quả cao. Module tiền xử lý stream giúp Snort có thể hiểu được các phiên làm việc khác nhau (nói cách khác đem lại tính có trạng thái cho các gói tin) từ đó giúp đạt được hiệu quả cao hơn trong việc phát hiện xâm nhập.

Giải mã và chuẩn hóa giao thức (decode/normalize): công việc phát hiện xâm nhập dựa trên dấu hiệu nhận dạng nhiều khi bị thất bại khi kiểm tra các giao thức có dữ liệu có thể được thể hiện dưới nhiều dạng khác nhau. Ví dụ: một web server có thể chấp nhận nhiều dạng URL như URL được viết dưới dạng mã hexa/Unicode, URL chấp nhận cả dấu \ hay / hoặc nhiều ký tự này liên tiếp cùng lúc. Chẳng hạn ta có dấu hiệu nhận dạng "scripts/iisadmin", kẻ tấn công có thể vượt qua được bằng cách tùy biến các yêu cầu gửi đến web server như sau:

"scripts/./iisadmin"

"scripts/examples/../iisadmin"

"scripts\iisadmin"

"scripts/.\iisadmin"

Hoặc thực hiện việc mã hóa các chuỗi này dưới dạng khác. Nếu Snort chỉ thực hiện đơn thuần việc so sánh dữ liệu với dấu hiệu nhận dạng sẽ xảy ra tình trạng bỏ sót các hành vi xâm nhập. Do vậy, một số môđun tiền xử lý của Snort phải có nhiệm vụ giải mã và chỉnh sửa, sắp xếp lại các thông tin đầu vào này để thông tin khi đưa đến môđun phát hiện có thể phát hiện được mà không bỏ sót. Hiện nay Snort đã hỗ trợ việc giải mã và chuẩn hóa cho các giao thức: telnet, http, rpc, arp.

Phát hiện các xâm nhập bất thường (nonrule /anormal): các plugin tiền xử lý dạng này thường dùng để đối phó với các xâm nhập không thể hoặc rất khó phát hiện được bằng các luật thông thường hoặc các dấu hiệu bất thường

trong giao thức. Các môđun tiền xử lý dạng này có thể thực hiện việc phát hiện xâm nhập theo bất cứ cách nào mà ta nghĩ ra từ đó tăng cường thêm tính năng cho Snort. Ví dụ, một plugin tiền xử lý có nhiệm vụ thống kê thông lượng mạng tại thời điểm bình thường để rồi khi có thông lượng mạng bất thường xảy ra nó có thể tính toán, phát hiện và đưa ra cảnh báo (phát hiện xâm nhập theo mô hình thống kê). Phiên bản hiện tại của Snort có đi kèm hai plugin giúp phát hiện các xâm nhập bất thường đó là portscan và bo (backoffice). Portcan dùng để đưa ra cảnh báo khi kẻ tấn công thực hiện việc quét các cổng của hệ thống để tìm lỗ hổng. Bo dùng để đưa ra cảnh báo khi hệ thống đã bị nhiễm trojan backoffice và kẻ tấn công từ xa kết nối tới backoffice thực hiện các lệnh từ xa.

2.2.2.3 Modun phát hiện

Đây là môđun quan trọng nhất của Snort. Nó chịu trách nhiệm phát hiện các dấu hiệu xâm nhập. Môđun phát hiện sử dụng các luật được định nghĩa trước để so sánh với dữ liệu thu thập được từ đó xác định xem có xâm nhập xảy ra hay không. Rồi tiếp theo mới có thể thực hiện một số công việc như ghi log, tạo thông báo và kết xuất thông tin.

Một vấn đề rất quan trọng trong môđun phát hiện là vấn đề thời gian xử lý các gói tin: một IDS thường nhận được rất nhiều gói tin và bản thân nó cũng có rất nhiều các luật xử lý. Có thể mất những khoảng thời gian khác nhau cho việc xử lý các gói tin khác nhau. Và khi thông lượng mạng quá lớn có thể xảy ra việc bỏ sót hoặc không phản hồi được đúng lúc. Khả năng xử lý của môđun phát hiện dựa trên một số yếu tố như: số lượng các luật, tốc độ của hệ thống đang chạy Snort, tải trên mạng. Một số thử nghiệm cho biết, phiên bản hiện tại của Snort khi được tối ưu hóa chạy trên hệ thống có nhiều bộ vi xử lý và cấu hình máy tính tương đối mạnh thì có thể hoạt động tốt trên cả các mạng cỡ Giga.

Một môđun phát hiện cũng có khả năng tách các phần của gói tin ra và áp dụng các luật lên từng phần nào của gói tin đó. Các phần đó có thể là:

- IP header
- Header ở tầng giao vận: TCP, UDP
- Header ở tầng ứng dụng: DNS header, HTTP header, FTP header, …
- Phần tải của gói tin (bạn cũng có thể áp dụng các luật lên các phần dữ liệu được truyền đi của gói tin)

Một vấn đề nữa trong Môđun phát hiện đó là việc xử lý thế nào khi một gói tin bị phát hiện bởi nhiều luật. Do các luật trong Snort cũng được đánh thứ tự ưu tiên, nên một gói tin khi bị phát hiện bởi nhiều luật khác nhau, cảnh báo được đưa ra sẽ là cảnh báo ứng với luật có mức ưu tiên lớn nhất.

2.2.2.4 Môđun log và cảnh báo.

Tùy thuộc vào việc môđun Phát hiện có nhận dạng được xâm nhập hay không mà gói tin có thể bị ghi log hoặc đưa ra cảnh báo. Các file log là các file text dữ liệu trong đó có thể được ghi dưới nhiều định dạng khác nhau chẳng hạn tcpdump.

2.2.2.5 Mô đun kết xuất thông tin.

Môđun này có thể thực hiện các thao tác khác nhau tùy theo việc bạn muốn lưu kết quả xuất ra như thế nào. Tùy theo việc cấu hình hệ thống mà nó có thể thực hiện các công việc như là:

- Ghi log file
- Ghi syslog: syslog và một chuẩn lưu trữ các file log được sử dụng rất nhiều trên các hệ thống Unix, Linux.
- Ghi cảnh báo vào cơ sở dữ liệu.
- Tạo file log dạng xml: việc ghi log file dạng xml rất thuận tiện cho việc trao đổi và chia sẻ dữ liệu.

- Cấu hình lại Router, firewall.
- Gửi các cảnh báo được gói trong gói tin sử dụng giao thức SNMP. Các gói tin dạng SNMP này sẽ được gửi tới một SNMP server từ đó giúp cho việc quản lý các cảnh báo và hệ thống IDS một cách tập trung và thuận tiện hơn.
- Gửi các thông điệp SMB (Server Message Block) tới các máy tính Windows.

Nếu không hài lòng với các cách xuất thông tin như trên, ta có thể viết các môđun kết xuất thông tin riêng tuỳ theo mục đích sử dụng.

2.2.3 Bộ luật của snort.

Cũng giống như virus, hầu hết các hoạt động tấn công hay xâm nhập đều có các dấu hiệu riêng. Các thông tin về các dấu hiệu này sẽ được sử dụng để tạo nên các luật cho Snort. Thông thường, các bẫy (honey pots) được tạo ra để tìm hiểu xem các kẻ tấn công làm gì cũng như các thông tin về công cụ và công nghệ chúng sử dụng. Và ngược lại, cũng có các cơ sở dữ liệu về các lỗ hổng bảo mật mà những kẻ tấn công muốn khai thác. Các dạng tấn công đã biết này được dùng như các dấu hiệu để phát hiện tấn công xâm nhập. Các dấu hiệu đó có thể xuất hiện trong phần header của các gói tin hoặc nằm trong phần nội dung của chúng. Hệ thống phát hiện của Snort hoạt động dựa trên các luật (rules) và các luật này lại được dựa trên các dấu hiệu nhận dạng tấn công. Các luật có thể được áp dụng cho tất cả các phần khác nhau của một gói tin dữ liệu.

Một luật có thể được sử dụng để tạo nên một thông điệp cảnh báo, log một thông điệp hay có thể bỏ qua một gói tin.

Cấu trúc luật của Snort.

Hãy xem xét một ví dụ đơn giản:

alert tcp 192.168.2.0/24 23 -> any any (content:"confidential"; msg: "Detected confidential")

Ta thấy cấu trúc của một luật có dạng như sau:

Hình 2-8: Cấu trúc luật của Snort

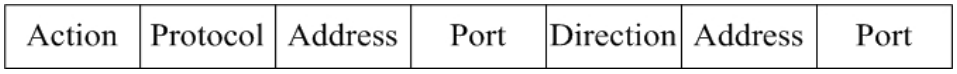
Diễn giải:

Tất cả các Luật của Snort về logic đều gồm 2 phần: Phần header và phần Option.

- Phần Header chứa thông tin về hành động mà luật đó sẽ thực hiện khi phát hiện ra có xâm nhập nằm trong gói tin và nó cũng chứa các tiêu chuẩn để áp dụng luật với gói tin đó.
- Phần Option chứa một thông điệp cảnh báo và các thông tin về các phần của gói tin dùng để tạo nên cảnh báo. Phần Option chứa các tiêu chuẩn phụ thêm để đối sánh luật với gói tin. Một luật có thể phát hiện được một hay nhiều hoạt động thăm dò hay tấn công. Các luật thông minh có khả năng áp dụng cho nhiều dấu hiệu xâm nhập.

2.2.3.1 Phần tiêu đề (Header).

Dưới đây là cấu trúc chung của phần Header của một luật Snort:



Hình 2-9: Header luật của Snort

Như phần trên đã trình bày, Header của luật bao gồm nhiều phần. Sau đây, là chi tiết cụ thể của từng phần một.

Action

Là phần qui định loại hành động nào được thực thi khi các dấu hiệu của gói tin được nhận dạng chính xác bằng luật đó. Thông thường, các hành động tạo ra một cảnh báo hoặc log thông điệp hoặc kích hoạt một luật khác. Action

chỉ ra hành động nào được thực hiện khi mà các điều kiện của luật được thoã mãn. Một hành động được thực hiện khi và chỉ khi tất cả các điều kiện đều phù hợp. Có 5 hành động đã được định nghĩa nhưng ta có thể tạo ra các hành động riêng tuỳ thuộc vào yêu cầu của mình. Đối với các phiên bản trước của Snort thì khi nhiều luật là phù hợp với một gói tin nào đó thì chỉ một luật được áp dụng. Sau khi áp dụng luật đầu tiên thì các luật tiếp theo sẽ không áp dụng cho gói tin ấy nữa. Nhưng đối với các phiên bản sau của Snort thì tất cả các luật sẽ được áp dụng gói tin đó.

- Pass: Hành động này hướng dẫn Snort bỏ qua gói tin này. Hành động này đóng vai trò quan trọng trong việc tăng cường tốc độ hoạt động của Snort khi mà ta không muốn áp dụng các kiểm tra trên các gói tin nhất định. Ví dụ ta sử dụng các bẫy (đặt trên một máy nào đó) để nhử các hacker tấn công vào thì ta phải cho tất cả các gói tin đi đến được máy đó. Hoặc là dùng một máy quét để kiểm tra độ an toàn mạng của mình thì ta phải bỏ qua tất cả các gói tin đến từ máy kiểm tra đó.
- Log: Hành động này dùng để log gói tin. Có thể log vào file hay vào cơ sở dữ liệu tuỳ thuộc vào nhu cầu của mình.
- Alert: Gửi một thông điệp cảnh báo khi dấu hiệu xâm nhập được phát hiện. Có nhiều cách để gửi thông điệp như gửi ra file hoặc ra một Console. Tất nhiên là sau khi gửi thông điệp cảnh báo thì gói tin sẽ được log lại.
- Activate: sử dụng để tạo ra một cảnh báo và kích hoạt một luật khác kiểm tra thêm các điều kiện của gói tin.
- Dynamic: chỉ ra đây là luật được gọi bởi các luật khác có hành động là Activate.

Các hành động do người dùng định nghĩa: một hành động mới được định nghĩa theo cấu trúc sau:

ruletype action_name

{

action definition

}

ruletype là từ khoá.

Hành động được định nghĩa chính xác trong dấu ngoặc nhọn: có thể là một hàm viết bằng ngôn ngữ C chẳng hạn.

Ví dụ như:

ruletype smb_db_alert

{

type alert

output alert_smb: workstation.list

output database: log, mysql, user=test password=test

dbname=snort host = localhost

}

Đây là hành động có tên là *smb_db_alert* dùng để gửi thông điệp cảnh báo dưới dạng cửa sổ pop-up SMB tới các máy có tên trong danh sách liệt kê trong file workstation.list và tới cơ sở dữ liệu MySQL tên là snort.

Protocols

Là phần thứ hai của một luật có chức năng chỉ ra loại gói tin mà luật sẽ được áp dụng. Protocols qui định việc áp dụng luật cho các packet chỉ thuộc một giao thức cụ thể nào đó. Hiện tại Snort hiểu được các protocol sau:

- IP
- ICMP

- TCP
- UDP

Nếu là IP thì Snort sẽ kiểm tra header của lớp liên kết để xác định loại gói tin. Nếu bất kì giao thức nào khác được sử dụng thì Snort sử dụng header IP để xác định loại protocol. Protocol chỉ đóng vai trò trong việc chỉ rõ tiêu chuẩn trong phần header của luật. Phần option của luật có thể có các điều kiện không liên quan gì đến protocol.

Address

Là phần địa chỉ nguồn và địa chỉ đích. Các địa chỉ có thể là một máy đơn, nhiều máy hoặc của một mạng nào đó. Trong hai phần địa chỉ trên thì một sẽ là địa chỉ nguồn, một sẽ là địa chỉ đích và địa chỉ nào thuộc loại nào sẽ do phần Direction "->" qui định. Có hai phần địa chỉ trong một luật của Snort. Các địa chỉ này được dùng để kiểm tra nguồn sinh ra và đích đến của gói tin. Địa chỉ có thể là địa chỉ của một IP đơn hoặc là địa chỉ của một mạng. Ta có thể dùng từ any để áp dụng luật cho tất cả các địa chỉ.

Địa chỉ được viết ngay theo sau một dấu gạch chéo và số bít trong subnet mask. Ví dụ như địa chỉ 192.168.2.0/24 thể hiện mạng lớp C 192.168.2.0 với 24 bít của subnet mask. Subnet mask 24 bít chính là 255.255.255.0. Ta biết rằng:

- Nếu subnet mask là 24 bít thì đó là mạng lớp C
- Nếu subnet mask là 16 bít thì đó là mạng lớp B
- Nếu subnet mask là 8 bít thì đó là mạng lớp A
- Nếu subnet mask là 32 bít thì đó là địa chỉ IP đơn.

Trong hai địa chỉ của một luật Snort thì có một địa chỉ là địa chỉ nguồn và địa chỉ còn lại là địa chỉ đích. Việc xác định đâu là địa chỉ nguồn, đâu là địa chỉ đích thì phụ thuộc vào phần hướng (direction).

Ví dụ như luật:

alert tcp any any -> 192.168.1.10/32 80 (msg: “TTL=100”; ttl: 100;)

Luật trên sẽ tạo ra một cảnh báo đối với tất cả các gói tin từ bất kì nguồn nào có TTL = 100 đi đến web server 192.168.1.10 tại cổng 80.

Ngăn chặn địa chỉ hay loại trừ địa chỉ

Snort cung cấp cho ta kĩ thuật để loại trừ địa chỉ bằng cách sử dụng dấu phủ định (dấu !). Dấu phủ định này đứng trước địa chỉ sẽ chỉ cho Snort không kiểm tra các gói tin đến từ hay đi tới địa chỉ đó. Ví dụ, luật sau sẽ áp dụng cho tất cả các gói tin ngoại trừ các gói có nguồn xuất phát từ mạng lớp C 192.168.2.0.

alert icmp![192.168.2.0/24] any

-> any any (msg: “Ping with TTL=100”; ttl: 100;)

Danh sách địa chỉ

Ta có thể định rõ ra danh sách các địa chỉ trong một luật của Snort. Ví dụ nếu bạn muốn áp dụng luật cho tất cả các gói tin trừ các gói xuất phát từ hai mạng lớp C 192.168.2.0 và 192.168.8.0 thì luật được viết như sau:

alert icmp![192.168.2.0/24, 192.168.8.0/24] any

-> any any (msg: “Ping with TTL=100”; ttl: 100;)

Hai dấu [] chỉ cần dùng khi có dấu ! đứng trước.

Cổng (Port Number)

Xác định các cổng nguồn và đích của một gói tin mà trên đó luật được áp dụng. Số hiệu cổng dùng để áp dụng luật cho các gói tin đến từ hoặc đi đến một cổng hay một phạm vi cổng cụ thể nào đó. Ví dụ ta có thể sử dụng số cổng nguồn là 23 để áp dụng luật cho tất cả các gói tin đến từ một server Telnet. Từ any cũng được dùng để đại diện cho tất cả các cổng. Chú ý là số

hiệu cổng chỉ có ý nghĩa trong các giao thức TCP và UDP thôi. Nếu protocol của luật là IP hay ICMP thì số hiệu cổng không đóng vai trò gì cả.

Ví dụ:

alert tcp 192.168.2.0/24 23 -> any any (content: “confidential”; msg: ”Detected confidential”;)

Số hiệu cổng chỉ hữu dụng khi ta muốn áp dụng một luật chỉ cho một loại gói tin dữ liệu cụ thể nào đó. Ví dụ như là một luật để chống hack cho web thì ta chỉ cần sử dụng cổng 80 để phát hiện tấn công.

Dãy cổng hay phạm vi cổng:

Ta có thể áp dụng luật cho dãy các cổng thay vì chỉ cho một cổng nào đó. Cổng bắt đầu và cổng kết thúc phân cách nhau bởi dấu hai chấm “:”.

Ví dụ:

alert udp any 1024:2048 -> any any (msg: “UDP ports”;)

Ta cũng có thể dùn cổng theo kiểu cận trên và cận dưới, tức là chỉ sử dụng cổng bắt đầu hoặc cổng kết thúc mà thôi. Ví dụ như là “1024:” hoặc là “:2048”

Dấu phủ định cũng được áp dụng trong việc sử dụng cổng. Ví dụ sau sẽ log tất cả các gói tin ngoại trừ các gói tin xuất phát từ cổng 53.

log udp any !53 -> any any log udp

Sau đây là một số cổng thông dụng hay là các cổng của các dịch vụ thông dụng nhất:

- 20 FTP data
- 21 FTP
- 22 SSH
- 23 Telnet

- 24 SMTP
- 53 DNS Server
- 80 HTTP
- 110 POP3
- 161 SNMP
- 443 HTTPS
- 3360 MySQL

Hướng – Direction

Chỉ ra đâu là nguồn đâu là đích, có thể là -> hay <- hoặc <>. Trường hợp <> là khi ta muốn kiểm tra cả Client và Server.

Ví dụ:

"alert icmp any any -> any any (msg: "Ping with TTL=100";ttl: 100;)"

- Phần đứng trước dấu mở ngoặc là phần Header của luật còn phần còn lại là phần Option. Chi tiết của phần Header như sau:
- Hành động của luật ở đây là "alert": một cảnh báo sẽ được tạo ra nếu như các điều kiện của gói tin là phù hợp với luật(gói tin luôn được log lại mỗi khi cảnh báo được tạo ra).
- Protocol của luật ở đây là ICMP tức là luật chỉ áp dụng cho các gói tin thuộc loại ICMP. Bởi vậy, nếu như một gói tin không thuộc loại ICMP thì phần còn lại của luật sẽ không cần đối chiếu.
- Địa chỉ nguồn ở đây là "any": tức là luật sẽ áp dụng cho tất cả các gói tin đến từ mọi nguồn còn cổng thì cũng là "any" vì đối với loại gói tin ICMP thì cổng không có ý nghĩa. Số hiệu cổng chỉ có ý nghĩa với các gói tin thuộc loại TCP hoặc UDP thôi.

- Còn phần Option trong dấu đóng ngoặc chỉ ra một cảnh báo chứa dòng "Ping with TTL=100" sẽ được tạo khi tìm thấy điều kiện TTL=100. TTL là Time To Live là một trường trong Header IP.

2.2.3.2 Các tùy chọn (Option).

Phần Rule Option nằm ngay sau phần Rule Header và được bao bọc trong dấu ngoặc đơn. Nếu có nhiều option thì các option sẽ được phân cách với nhau bằng dấu chấm phẩy ";".Nếu nhiều option được sử dụng thì các option này phải đồng thời được thoã mãn tức là theo logic các option này liên kết với nhau bằng AND.

Mọi option được định nghĩa bằng các từ khoá. Một số các option còn chứa các tham số. Nói chung một option gồm 2 phần: một từ khoá và một tham số, hai phần này phân cách nhau bằng dấu hai chấm. Ví dụ đã dùng:

msg: "Detected confidented";

msg là từ khoá còn "Detected confidented" là tham số.

Sau đây là chi tiết một số các option của luật Snort.

Từ khoá ack

Trong header TCP có chứa trường Acknowledgement Number với độ dài 32 bit. Trường này có ý nghĩa là chỉ ra số thứ tự tiếp theo gói tin TCP của bên gửi đang được chờ để nhận. Trường này chỉ có ý nghĩa khi mà cờ ACK được thiết lập.

Các công cụ như Nmap sử dụng đặc điểm này ping một máy. Ví dụ, nó có thể gửi một gói tin TCP tới cổng 80 với cờ ACK được bật và số thứ tự là 0. Bởi vậy, bên nhận sẽ thấy gói tin không hợp lệ và sẽ gửi trở lại gói tin RST. Khi mà Nmap nhận được gói tin RST thì tức là địa chỉ đích đang "sống". Phương pháp này vẫn làm việc tốt đối với các máy không trả lời gói tin thuộc dạng ping ICMP ECHO REQUEST.

Vậy để kiểm tra loại ping TCP này thì ta có thể dùng luật như sau:

alert tcp any any -> 192.168.1.0/24 any (flags: A; ack: 0; msg: "TCP ping detected")

Từ khoá classtype

Các luật có thể được phân loại và gán cho một số chỉ độ ưu tiên nào đó để nhóm và phân biệt chúng với nhau. Để hiểu rõ hơn về từ khoá này ta đầu tiên phải hiểu được file *classification.config* (được bao gồm trong file *snort.conf* sử dụng từ khoá include). Mỗi dòng trong file classification.config có cú pháp như sau:

config classification: name, description, priority

trong đó:

- name: là tên dùng để phân loại, tên này sẽ được dùng với từ khoá classtype trong các luật Snort.
- description: mô tả về loại lớp này
- priority: là một số chỉ độ ưu tiên mặc định của lớp này. Độ ưu tiên này có thể được điều chỉnh trong từ khoá priority của phần option trong luật của Snort.

Ví dụ:

config classification: DoS, Denial of Service Attack, 2

và trong luật:

alert udp any any -> 192.168.1.0/24 6838 (msg:"DoS"; content: "server"; classtype: DoS;)

alert udp any any -> 192.168.1.0/24 6838 (msg:"DoS"; content: "server"; classtype: DoS; priority: 1;)

Trong câu lệnh thứ 2 thì ta đã ghi đè lên giá trị priority mặc định của lớp đã định nghĩa.

Từ khoá content

Một đặc tính quan trọng của Snort là nó có khả năng tìm một mẫu dữ liệu bên trong một gói tin. Mẫu này có thể dưới dạng chuỗi ASCII hoặc là một chuỗi nhị phân dưới dạng các kí tự hệ 16. Giống như virus, các tấn công cũng có các dấu hiệu nhận dạng và từ khoá content này dùng để tìm các dấu hiệu đó bên trong gói tin. Ví dụ:

alert tcp 192.168.1.0/24 any -> ![192.168.1.0/24] any (content: “GET”; msg: “GET match”;)

Luật trên tìm mẫu “GET” trong phần dữ liệu của tất cả các gói tin TCP có nguồn đi từ mạng 192.168.1.0/24 và đi đến các địa chỉ không thuộc mạng đó. Từ “GET” này rất hay được dùng trong các tấn công HTTP.

Một luật khác cũng thực hiện đúng nhiệm vụ giống như lệnh trên nhưng mẫu dữ liệu lại dưới dạng hệ 16 là:

alert tcp 192.168.1.0/24 any -> ![192.168.1.0/24] any (content: “|47 45 54|”; msg: “GET match”;)

Để ý rằng số 47 ở hệ 16 chính là bằng kí tự ASCII: G và tương tự 45 là E và 54 là T. Ta có thể dùng cả hai dạng trên trong cùng một luật nhưng nhớ là phải để dạng thập lục phân giữa cặp kí tự ||.

Tuy nhiên khi sử dụng từ khoá content ta cần nhớ rằng:

- Đối sánh nội dung sẽ phải xử lý tính toán rất lớn và ta phải hết sức cân nhắc khi sử dụng nhiều luật có đối sánh nội dung.
- Ta có thể sử dụng nhiều từ khoá content trong cùng một luật để tìm nhiều dấu hiệu trong cùng một gói tin.
- Đối sánh nội dung là công việc rất nhạy cảm.

Có 3 từ khoá khác hay được dùng cùng với từ khoá content dùng để bổ sung thêm các điều kiện để tìm kiếm là:

- offset: dùng để xác định vị trí bắt đầu tìm kiếm (chuỗi chứa trong từ khoá content) là offset tính từ đầu phần dữ liệu của gói tin. Ví dụ sau sẽ tìm chuỗi "HTTP" bắt đầu từ vị trí cách đầu đoạn dữ liệu của gói tin là 4 byte:
- alert tcp 192.168.1.0/24 any -> any any (content: "HTTP"; offset: 4; msg: "HTTP matched";)
- dept: dùng để xác định vị trí mà từ đó Snort sẽ dừng việc tìm kiếm.Từ khoá này cũng thường được dùng chung với từ khoá offset vừa nêu trên.

  Ví dụ:

  alert tcp 192.168.1.0/24 any -> any any (content: "HTTP"; offset: 4; dept: 40; msg: "HTTP matched";).

Từ khoá này sẽ giúp cho việc tiêu tốn thời gian tìm kiếm khi mà đoạn dữ liệu trong gói tin là khá lớn.

- content-list: được sử dụng cùng với một file. Tên file (được chỉ ra trong phần tham số của từ khoá này) là một file text chứa danh sách các chuỗi cần tìm trong phần dữ liệu của gói tin. Mỗi chuỗi nằm trên một dòng riêng biệt. Ví dụ như file test có dạng như sau:

  "test"

  "Snort"

  "NIDS"

  và ta có luật sau:

  alert tcp 192.168.1.0/24 any -> any any (content-list: "test";msg: "This is my Test";).

Ta cũng có thể dùng kí tự phủ định ! trước tên file để cảnh báo đối với các gói tin không tìm thấy một chuỗi nào trong file đó.

Từ khoá dsize

Dùng để đối sánh theo chiều dài của phần dữ liệu. Rất nhiều tấn công sử dụng lỗi tràn bộ đệm bằng cách gửi các gói tin có kích thước rất lớn. Sử dụng từ khoá này, ta có thể so sánh độ lớn của phần dữ liệu của gói tin với một số nào đó.

alert ip any any -> 192.168.1.0/24 any (dsize: > 6000; msg: “Goi tin co kich thuoc lon”;)

Từ khoá flags

Từ khoá này được dùng để phát hiện xem những bit cờ flag nào được bật (thiết lập) trong phần TCP header của gói tin. Mỗi cờ có thể được sử dụng như một tham số trong từ khoá flags. Sau đây là một số các cờ sử dụng trong từ khoá flags:

Bảng 3-1 Các cờ sử dụng với từ khoá flags

Flag	Kí tự tham số dùng trong luật của Snort
FIN (Finish Flag)	F
SYN – Sync Flag	S
RST – Reset Flag	R
PSH – Push Flag	P
ACK Acknowledge Flag	A
URG Urgent Flag	U
Reserved Bit 1	1
Reserved Bit 2	2
No Flag set	0

Ta có thể sử dụng các dấu +, * và ! để thực hiện các phép toán logic AND, OR và NOT trên các bit cờ muốn kiểm tra. Ví dụ luật sau đây sẽ phát hiện một hành động quét dùng gói tin TCP SYN-FIN:

alert tcp any any -> 192.168.1.0/24 any (flags: SF; msg: "SYNC-FIN packet detected";)

Từ khoá fragbits

Phần IP header của gói tin chứa 3 bit dùng để chống phân mảnh và tổng hợp các gói tin IP. Các bit đó là:

- Reserved Bit (RB) dùng để dành cho tương lai.
- Don't Fragment Bit (DF): nếu bit này được thiết lập thì tức là gói tin đó không bị phân mảnh.
- More Fragments Bit (MF): nếu được thiết lập thì tức là các phần khác (gói tin bị phân mảnh) của gói tin vẫn đang còn trên đường đi mà chưa tới đích. Nếu bit này không được thiết lập thì có nghĩa là đây là phần cuối cùng của gói tin (hoặc là gói duy nhất). Điều này xuất phát từ nguyên nhân: Nơi gửi đi phải chia gói tin IP thành nhiều đoạn nhỏ do phụ thuộc vào Đơn vị truyền dữ liệu lớn nhất cho phép (Maximum Transfer Units - MTU) trên đường truyền. Kích thước của gói tin không được phép vượt quá kích thước lớn nhất này. Do vậy, bit MF này giúp bên đích có thể tổng hợp lại các phần khác nhau thành một gói tin hoàn chỉnh.

Đôi khi các bit này bị các hacker sử dụng để tấn công và khai thác thông tin trên mạng của ta. Ví dụ, bit DF có thể được dùng để tìm MTU lớn nhất và nhỏ nhất trên đường đi từ nguồn xuất phát đến đích đến.

Sử dụng fragbits, ta có thể kiểm tra xem các bit trên có được thiết lập hay không. Ví dụ luật sau sẽ phát hiện xem bit DF trong gói tin ICMP có được bật hay không:

alert icmp any any -> 192.168.1.0/24 any (fragbits: D; msg: “Dont Fragment bit set”;)

Trong luật này, D dùng cho bit DF, R cho bit dự trữ và M cho bit MF. Ta cũng có thể dùng dấu phủ định ! trong luật này để kiểm tra khi bit không được bật:

alert icmp any any -> 192.168.1.0/24 any (fragbits: !D; msg: “Dont Fragment bit not set”;)

2.2.4 Chế độ ngăn chặn của Snort: Snort – Inline

2.2.4.1 Tích hợp khả năng ngăn chặn vào Snort

Snort-inline là một nhánh phát triển của Snort do **William Metcalf** khởi xướng và lãnh đạo. Đến phiên bản 2.3.0 RC1 của Snort, inline-mode đã được tích hợp vào bản chính thức do snort.org phát hành. Sự kiện này đã biến Snort từ một IDS thuần túy trở thành một hệ thống có các khả năng của một IPS, mặc dù chế độ này vẫn chỉ là tùy chọn chứ không phải mặc định.

Ý tưởng chính của inline-mode là kết hợp khả năng ngăn chặn của iptables vào bên trong snort. Điều này được thực hiện bằng cách thay đổi môđun phát hiện và môđun xử lý cho phép snort tương tác với iptables. Cụ thể, việc chặn bắt các gói tin trong Snort được thực hiện thông qua Netfilter và thư viện libpcap sẽ được thay thế bằng việc sử dụng ipqueue và thư viện libipq. Hành động ngăn chặn của snort-inline sẽ được thực hiện bằng devel-mode của iptables.

2.2.4.2 Những bổ sung cho cấu trúc luật của Snort hỗ trợ Inline mode

Để hỗ trợ tính năng ngăn chặn của Snort-inline, một số thay đổi và bổ sung đã được đưa vào bộ luật Snort. Đó là đưa thêm 3 hành động DROP, SDROP, INJECT và thay đổi trình tự ưu tiên của các luật trong Snort.

- DROP: Hành động DROP yêu cầu iptables loại bỏ gói tin và ghi lại thông tin như hành động LOG.

- SDROP: Hành động SDROP cũng tương tự như hành động DROP, điều khác biệt là ở chỗ Snort sẽ không ghi lại thông tin như hành động LOG.
- REJECT: Hành động REJECT yêu cầu iptables từ chối gói tin, có nghĩa là iptables sẽ loại bỏ và gửi lại một thông báo cho nguồn gửi gói tin đó. Hành động REJECT không ghi lại bất cử thông tin gì.

Trình tự ưu tiên của các luật:

Trong các phiên bản gốc, trình tự ưu tiên của các hành động trong Snort là:

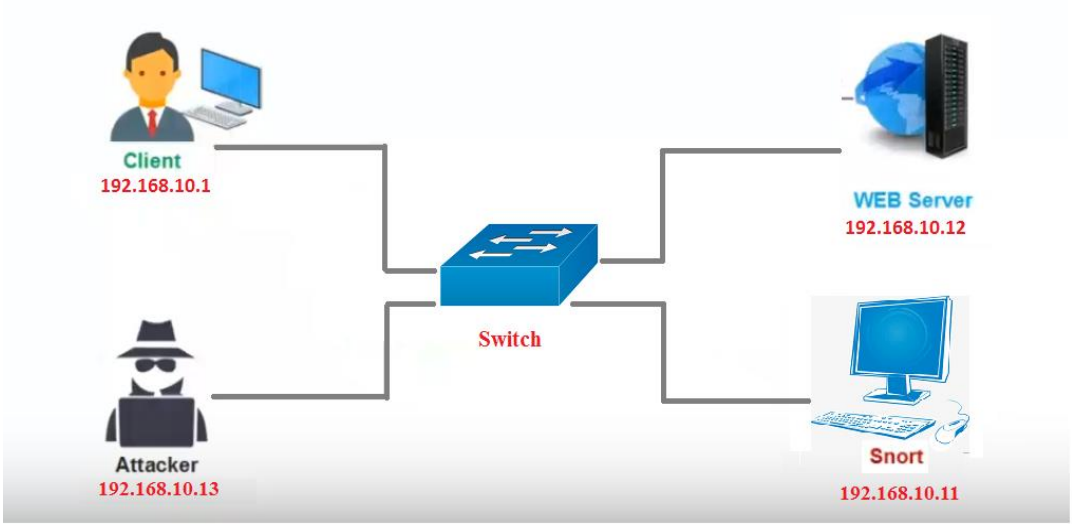
activation->dynamic-> alert->pass->log

Trong inline-mode, trình tự ưu tiên này được thay đổi như sau:

activation->dynamic->pass->drop->sdrop->reject->alert->log

CHƯƠNG 3: THỰC NGHIỆM PHÁP HIỆN XÂM NHẬP MẠNG VỚI SNORT

3.1 Mô hình thử nghiệm



Hình 3-1: Mô hình thử nghiệm

3.2 Thiết lập cấu hình, chuẩn bị môi trường cài đặt:

Môi trường giả lập: **Vmware Workstation Pro 14**

- Snort: CentOS 6.5 – **Vmnet: 192.168.10.11**
- Web Server: Windows Server 2012 – **Vmnet: 192.168.10.12**
- Attacker: Windows 7 – **Vmnet: 192.168.10.13**
- Client: Windown 7 – **192.168.10.1 (Cài đặt PuTTy và WinSCP)**

3.3 Cài đặt SNORT

Bước 1: Sử dung PuTTy kết nối đến máy chủ Linux CentOS 6.5

Bước 2: Tải về các gói sau và sử dụng WinSCP tải các gói cài đặt nên máy chủ Linux CentOS 6.5

- Libdnet-1.12.tgz
- libpcap-1.0.0.tar.gz
- daq-2.0.0.tar.gz

- snort-2.8.4.1.tar.gz
- adodb519.zip
- base-1.3.9.tar.gz
- Snortrules-snapshot--2953.tar.

Bước 3: Cài đặt

- **Cài đặt thêm các gói hỗ trợ cho Web**

yum install -y mysql-server mysql-bench mysql-devel httpd php php-mbstring php-devel php-mysql php-pear gcc pcre-devel php-gd gd glib2-devel gcc-c++ libpcap libpcap-devel flex bison

- **Giải nén và cài đặt Libdnet-1.12.tgz**

  # tar -zxvf libdnet-1.12.tgz

  # cd libdnet-1.12

  #./configure --prefix=/usr/local/snort --bindir=/usr/local/bin/

  # make && make install

- **Chuyển về thư mục /root. Giải nén và cài đặt gói libpcap-1.0.0.tar.gz**

  # cd

  # tar -zxvf libpcap-1.0.0.tar.gz

  # cd libpcap-1.0.0

  #./configure --prefix=/usr/local/snort/ --bindir=/usr/local/bin/

  # make && make install

- **Chuyển về thư mục /root, giải nén và cài đặt gói daq-2.0.0.tar.gz**

  # tar -zxvf daq-2.0.0.tar.gz

  # cd daq-2.0.0

#./configure --prefix=/usr/local/snort/ -bindir=/usr/local/bin/ --with-libpcap-includes=/usr/local/snort/include/ --with-libpcap-libraries=/usr/local/snort/lib/ --enable-static

make && make install

- **Tạo người dùng và nhóm người dùng cho Snort**

groupadd snort

useradd -g snort -d /etc/snort -M -s /sbin/nologin snort

- **Chuyển về thư mục /root và tiến hành giải nén, cài đặt snort-2.8.4.1.tar.gz**

cd

tar -zxvf snort-2.8.4.1.tar.gz

cd snort-2.8.4.1

#./configure --with-mysql-libraries=/usr/lib64/mysql --enable-sourcefire

make && make install

- **Khởi tạo link liên kết cho tệp tin snort tới thư mục sbin**

ln -s /usr/local/bin/snort /usr/sbin/

- **Copy kịch bản khởi động của snort tới thư mục /etc/init.d/**

cp rpm/snortd /etc/init.d/

cp rpm/snort.sysconfig /etc/sysconfig/snort

- **Cấp quyền với tệp tin khởi động của Snort**

chmod 755 /etc/init.d/snortd

- **Tạo các thư mục chứa các file cài đặt và thư mục chứa các file log**

mkdir /etc/snort

mkdir /var/log/snort

- **Thay đổi quyền sở hữu của thư mục /var/log/snort**

chown snort:snort /var/log/snort/

- **Chuyển về thư mục /root và copy tất cả các file cấu hình của Snort tới thư mục /etc/snort**

cp snort-2.8.4.1/etc/* /etc/snort/

rm -rf /etc/snort/Makefile* (tệp tin hỗ trợ cho việc biên dịch các chương trình viết bằng mã nguôn C)

mkdir -p /usr/local/lib64/snort_dynamicrules

- **Copy toàn bộ thư mục Snortrules-snapshot--2953.tar.gz vào thư mục /etc/snort**

cp snortrules-snapshot-2953.tar.gz /etc/snort/

- **Chuyển đến thư mục /etc/snort tiến hành giải nén và cài đặt Snortrule**

cd /etc/snort/

tar -zxvf snortrules-snapshot-2953.tar.gz

cp /etc/snort/so_rules/precompiled/Centos-5-4/x86-64/2.9.5.3/*.so /usr/local/lib64/snort_dynamicrules/

cat /etc/snort/so_rules/*.rules >> /etc/snort/rules/so-rules.rules

- **Chuyển đến thư mục /etc/snort và tiến hành cấu hình Snort**

#cd /etc/snort

vi snort.conf

- **Dòng 110 trỏ đường dẫn tới thư mục chứa Rules**

var RULE_PATH /etc/snort/rules

- **Dòng 111 trỏ đường dẫn tới thư mục chứa thư viện**

var PREPROC_RULE_PATH /etc/snort/preproc_rules

- **Dòng 688 tiến hành bỏ dấu # và điền thông tin về database**

output database: log, mysql, user=snort password=123456 dbname=snort host=localhost

- **Lưu file cấu hình**

vi /etc/sysconfig/snort

- **Dòng 69, 75, 81 thêm dấu # vào đầu dòng**

Bước 4: Tạo Database cho snort

- **Khởi động dịch vụ MySQL, và cho phép khởi động cùng hệ thống**

service mysqld start

chkconfig mysqld on

- **Tạo CSDL cho Snort với MySQL**

echo "set password for root@localhost=password('123456'); " | mysql -u root

echo "create database snort;" | mysql -u root -p

- **Gán toàn quyền cho User snort trên database snort**

echo "grant all privileges on snort.* to snort@localhost with grant option;" | mysql -u root -p

- **Set password cho User snort truy cập database**

echo "set password for snort@localhost=password('123456'); " | mysql -u root -p

- **Import CSDL**

cd snort-2.8.4.1/schemas/

mysql -u root -p < create_mysql snort

Bước 5: Cài đặt giao diện quản trị

- **Giải nén gói adodb519.zip**

  # unzip adodb519.zip

- **Di chuyển toàn bộ thư mục adodb vừa giải nén vào /var/www/adodb**

  # mv adodb5 /var/www/adodb

- **Giải nén gói base-1.3.9.tar.gz**

  # tar -zxvf base-1.3.9.tar.gz

- **Di chuyển toàn bộ thư mục base-1.3.9 vừa giải nén vào thư mục /var/www/html/base**

  # mv base-1.3.9 /var/www/html/base

- **Thay đổi quyền sở hữu của thư mục base**

  # chown apache:apache /var/www/html/base/

- **Sửa file php.ini**

  # vi /etc/php.ini

- **Bỏ tất cả dấu ; đầu dòng, từ dòng 112 - 115**

- **Sửa file httpd.conf**

  # vi /etc/httpd/conf/httpd.conf

- **Tại dòng 276 bỏ dấu # đầu dòng**

- **Khởi động dịch vụ httpd, và cho phép khởi động cùng hệ thống**

  # service httpd start

  # chkconfig httpd on

- **Cài đặt gói Epel**

rpm -ivh epel-release-6-8.noarch.rpm

- **Cài dặt thêm các gói sau**

yum -y install pcre pcre-devel php-pear php-pear-Number php-pear-Number-Words php-pear-Image-Color php-pear-Image-Canvas php-pear-Image-Graph

- **Sủa file snort.conf**

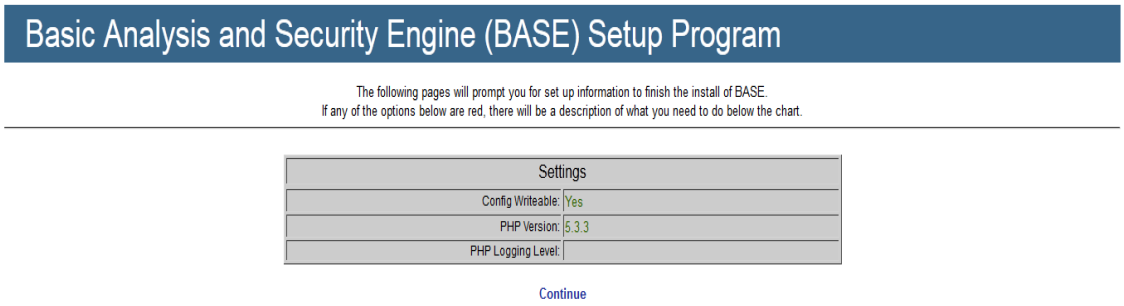
vi /etc/snort/snort.conf

- **Từ dòng 810 - 862 tiến hành thém dâu # vào đầu tất cả các dòng trỏ đường dẫn tới các Rules**
- **Khởi động dịch vụ Snort và cho phép khởi động cùng hệ thống**

service snortd start

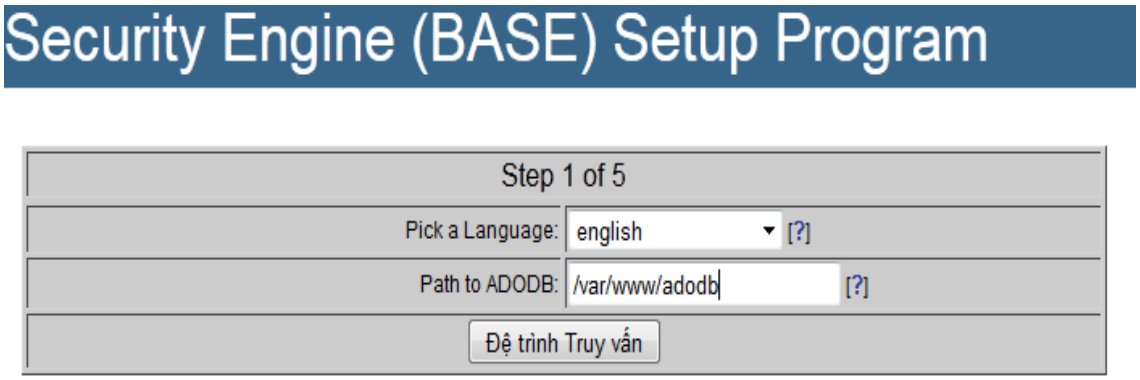
chkconfig snortd on

- **Sang máy Client đăng nhập vào Base để kiểm tra với địa chỉ http://192.168.10.12/base Chọn Continue để tiếp tục**



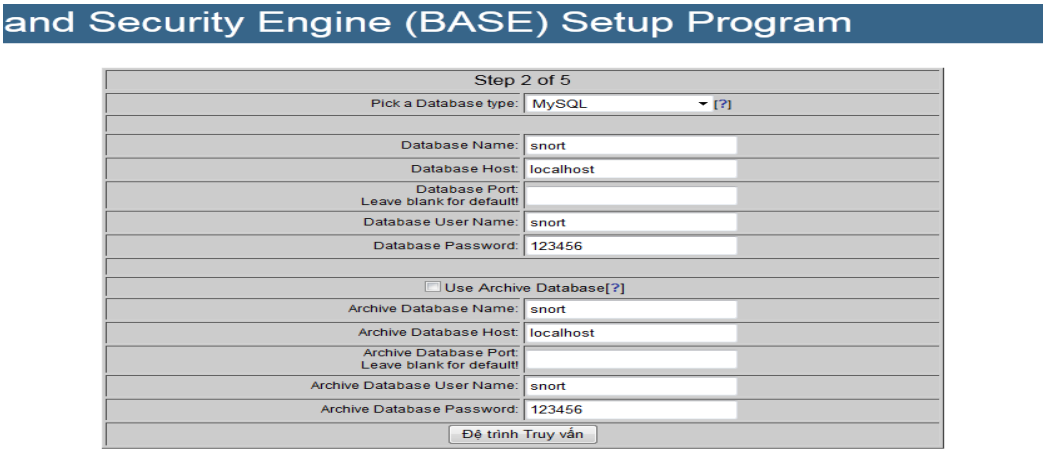
Hình 3-2: Hướng dẫn cài đặt SNORT - Thiết lập

- **Trỏ đường dẫn tới adodb**



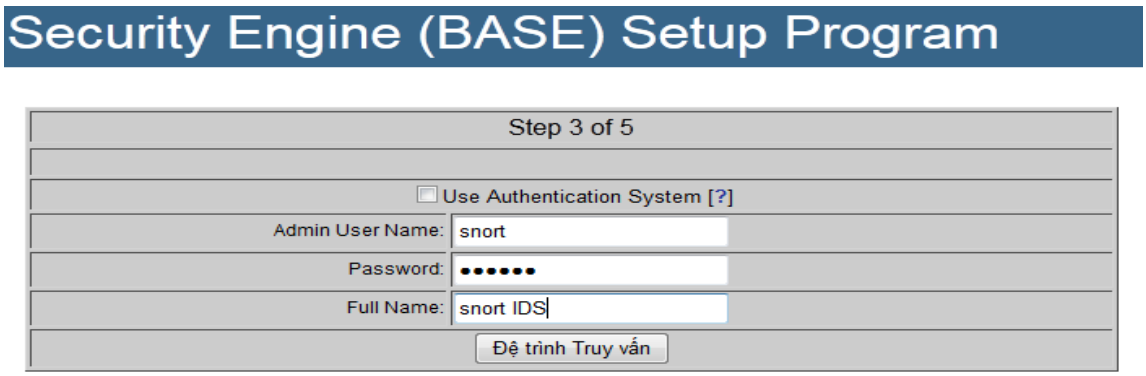
Hình 3-3: Hướng dẫn cài đặt SNORT - Bước 1

- **Nhập các thông tin về database, username, password của Username quản trị database**



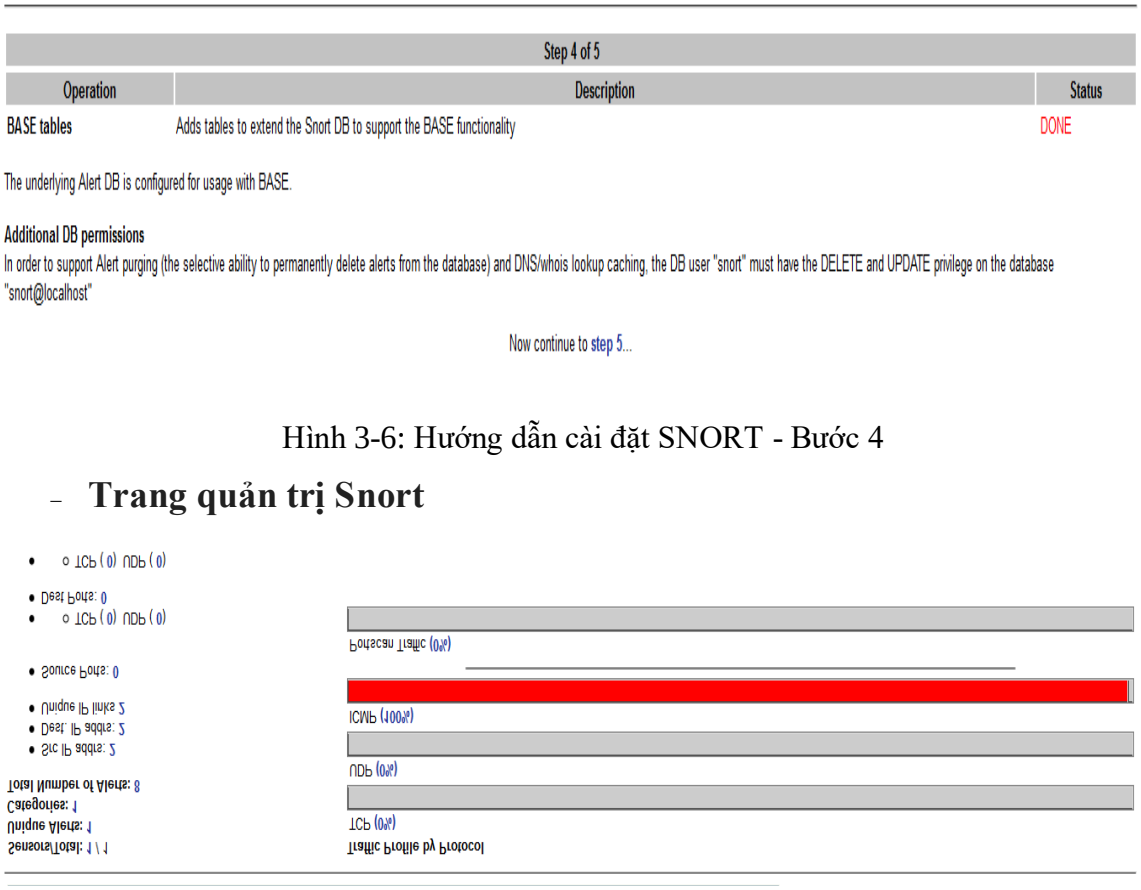
Hình 3-4: Hướng dẫn cài đặt SNORT - Bước 2

- **Nhập Username và password quản trị**



Hình 3-5: Hướng dẫn cài đặt SNORT - Bước 3

- **Chọn Continue để vào bước tiếp theo**



Step 4 of 5

Operation	Description	Status
BASE tables	Adds tables to extend the Snort DB to support the BASE functionality	DONE

The underlying Alert DB is configured for usage with BASE.

Additional DB permissions

In order to support Alert purging (the selective ability to permanently delete alerts from the database) and DNS/whois lookup caching, the DB user "snort" must have the DELETE and UPDATE privilege on the database "snort@localhost"

Now continue to step 5...

Hình 3-6: Hướng dẫn cài đặt SNORT - Bước 4

- **Trang quản trị Snort**

Hình 3-7: Trang quản trị Snort

3.4 Thiết lập một số luật cơ bản:

3.4.1 Tạo luật cảnh báo PING với kích thước lớn:

- Tạo Rules icmp.rules: # vi /etc/snort/rules/icmp.rules
- Tại đây tiến hành soạn 1 rules mới với nội dung như sau:

alert icmp any any -> 192.168.10.0/24 any (msg:"He thong dang bi tan cong bang PING goi dung luong cao"; dsize: >500; sid:1111;)

- Trỏ đường dẫn tới Rules vừa khởi tạo

vi /etc/snort/snort.conf

- Bỏ dấu # trước dường dẫn tới Rules icmp vừa khởi tạo

include $RULE_PATH/icmp.rules

- Khởi động lại dịch vụ Snort

service snortd restart

- Tạo lệnh ping ở máy Client như sau:

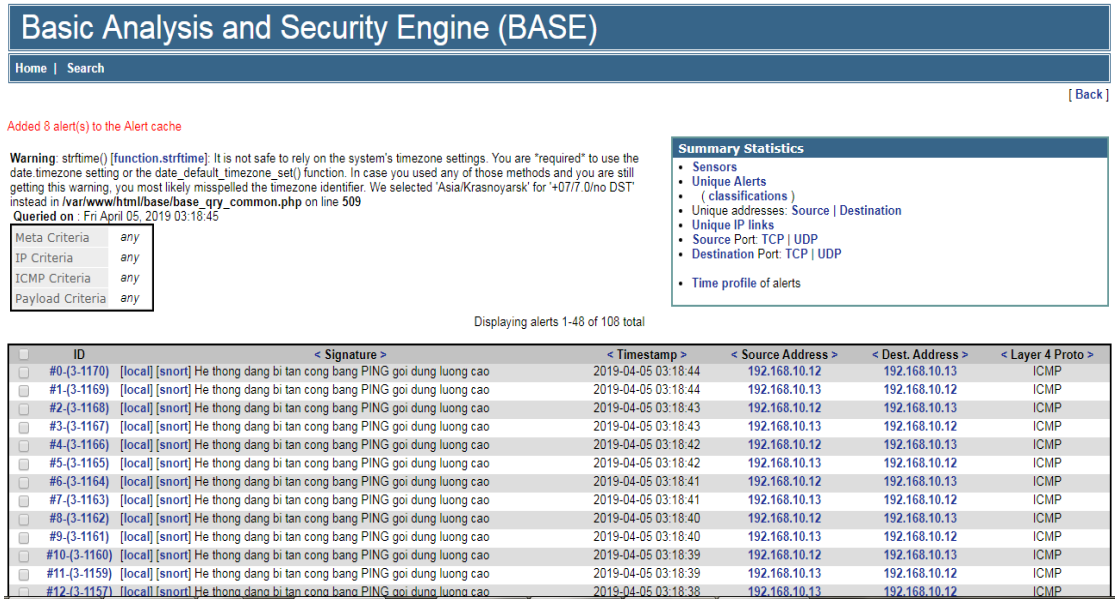
ping -l 1000 -f 192.168.10.13 –t

Mô tả tình huống:

Kiểu tấn công dùng giao thức ICMP. Có 2 phần quan trọng trong ICMP packet là ICMP ECHO_REQUEST và ICMP ECHO_RESPONSE datagrams và thông thường dùng PING command để thi hành các hoạt động của ICMP. Khi 1 máy tính gửi ICMP ECHO_REQUEST đến 1 máy nào đó, nếu máy đó đang hoạt động thì nó sẽ gửi trả lại ICMP ECHO_RESPONSE. Hacker dùng PING program để tạo nên kích thước lớn cho gói tin ICMP (gói gọn trong 1 IP packet), có nhiều cách để gửi ICMP datagrams mà packet mà chỉ bao gồm 8 bits ICMP header infomation, Hacker thuong dùng PING program để gửi những packet lớn hơn 65536 bytes (vượt qua sự cho phép của TCP/IP)

Thực nghiệm: Máy Attacker gửi gói ICMP Ping tới Webserver "ping -l 1000 -f 192.168.10.12 –t". Snort phát hiện và đưa ra cảnh báo tại trang quản trị Snort.

- **Kết quả tại Trang quản trị Snort như sau:**



Hình 3-8: Cảnh báo PING với kích thước lớn

3.4.2 Tạo luật cảnh báo truy cập Web:

- Tạo Rules icmp.rules: # vi /etc/snort/rules/tcp.rules
- Tại đây tiến hành soạn 1 rules mới với nội dung như sau:

alert tcp any any -> 192.168.10.0/24 any (content: ""; msg:"Phat hien xam nhap website trai phep"; sid:2222; rev: 1;)

- Trỏ đường dẫn tới Rules vừa khởi tạo

vi /etc/snort/snort.conf

- Bỏ dấu # trước dường dẫn tới Rules icmp vừa khởi tạo

include $RULE_PATH/icmp.rules

- Khởi động lại dịch vụ Snort

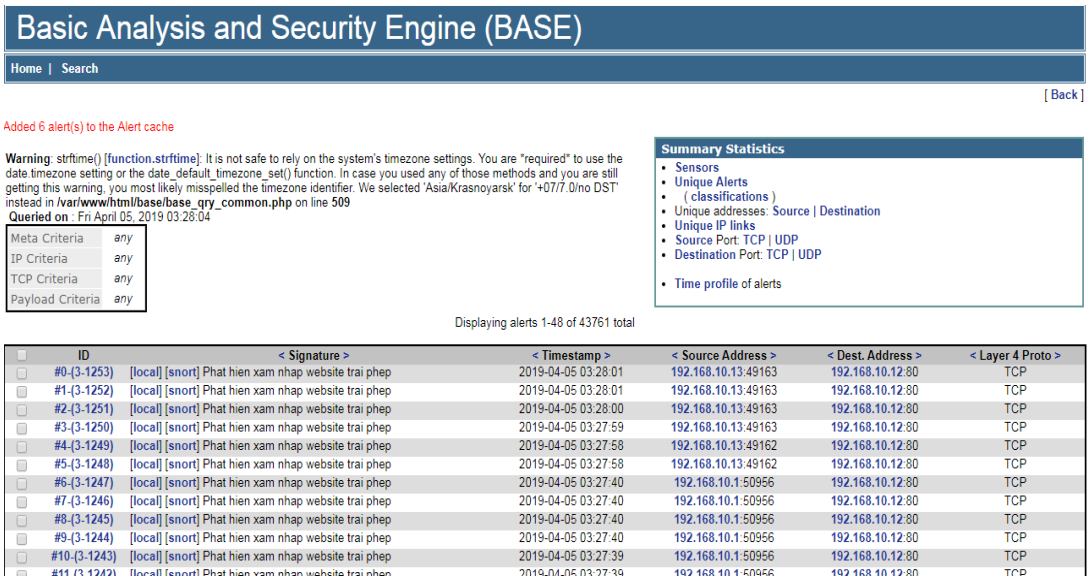
service snortd restart

Mô tả tình huống:

Thông thường, tấn công DoS xảy ra khi hacker thực hiện các hoạt động nhằm mục đích "flood" (làm lụt) network với một khối lượng thông tin khổng lồ. Khi người dùng nhập một URL của website nào đó vào trình duyệt, tức là đang gửi một request đến server máy tính của website đó. Về cơ bản, mỗi Server sẽ chỉ có khả năng xử lý một số reqquest nhất định được gửi đến trong cùng một lúc. Lợi dụng đặc điểm này, kẻ tấn công sẽ khuếch đại số request lên với khối lượng khổng lồ khiến cho sever mất khả năng xử lí lượng reqquest này. Đây chính là hình thức "từ chối dịch vụ", người dùng không thể truy cập trang web đó.

Thực nghiệm: Cùng một thời điểm máy Attacker và máy Client truy cập liên tục nhiều lần vào website đã được tạo https://192.168.10.12. Snort phát hiện và đưa ra cảnh báo tại trang quản trị Snort.

- **Kết quả tại Trang quản trị Snort như sau:**



Hình 3-9: Cảnh báo truy cập Web

KẾT LUẬN

Trong khuôn khổ đồ án, về mặt lý thuyết đồ án đã trình bày 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 giải pháp xây dựng một hệ thống phát hiện xâm nhập mạng (IDS). Các nội dung nghiên cứu mà đề tài đã đặt ra và giải quyết được các vấn đề sau:

- Tìm hiểu cơ bản về hệ thống giám sát an ninh mạng, các thành phần và chức năng chính của hệ thống giám sát an ninh mạng.
- Nghiên cứu, tìm hiểu hệ thống phát hiện xâm nhập mạng IDS và ngăn chặn xâm nhập mạng IPS.
- Cài đặt và thử nghiệm thành công ứng dụng phần mềm mã nguồn mở SNORT phát hiện xâm nhập mạng.
- Kết hợp xây dựng được giao diện quản trị ứng dụng SNORT trực quan với người sử dụng.

Song song với kết quả mà em đạt được thì em nhận thấy đồ án vẫn còn một số điểm hạn chế. Đồ án mới chỉ ở mức cài đặt và thử nghiệm, để có thể đưa vào thực nghiệm và sử dụng trong thực tế, cần phải tích hợp hoàn chỉnh, bổ sung thêm các chức năng và đó cũng là hướng phát triển nghiên cứu của đề tài, cụ thể là:

- Phát hiện thêm nhiều kiểu tấn công khác.
- Có thêm chức năng tự động phản ứng trước các cuộc tấn công.
- Tích hợp thêm nhiều phần mềm khác với nhiều tính năng hơn, giúp cho hệ thống giám sát an ninh mạng có thể giám sát được hệ thống chặt chẽ hơn.
- Hiện tại Snort với khả năng phát hiện xâm nhập dựa vào các mẫu sẵn có. Vì thế đối với các kiểu tấn công mới sẽ không thể phát hiện ra. Do vậy cần xây dựng thêm chức năng phân tích dựa vào sự kiện bất thường

và phân tích dựa trên giao thức để có thể phát hiện tấn công một cách chính xác nhất.

Hy vọng trong thời gian sắp tới, em có thể nghiên cứu sâu hơn nữa để hoàn thiện và phát triển đề tài thành sản phẩm được ứng dụng vào thực tiễn.

TÀI LIỆU THAM KHẢO

[1.] Andrew R. Bakeer & Joel Esler (2007), Snort IDS and IPS Toolkit. Syngress Publishing, Inc.

[2.] The Snort Team (2012), Snort® User Manual 2.9.3, The Snort Project.

[3.] David Gullett (2012), Snort 2.9.3 and Snort Report 1.3.3 on Ubuntu 12.04 LTS Install Guide, Symmetrix Technologies.

[4.] VNCERT (2007), "Nghiên cứu xây dựng mô hình hệ thống quản lý An toàn Internet theo cấu trúc phân bổ", Hà Nội

[5.] Học viện kỹ thuật mật mã (2008), "Bộ giao thức TCP/IP", Học viện kỹ thuật mật mã, Hà Nội

[6.] https://www.snort.org

[7.] https://seclists.org/snort/

[8.] https://fossies.org/linux/snort/configure.in

[9.] https://www.academia.edu/4302986/Cai_d%E1%BA%B7t_ Snort_Barnyard_BASE_tren_Cent_OS_5_2