**ĐẠI HỌC ĐÀ NẴNG**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA CÔNG NGHỆ THÔNG TIN**



**PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH**

**ĐỀ TÀI: TÌM HIỂU HỆ ĐIỀU HÀNH LINUX VÀ XÂY DỰNG ỨNG DỤNG PHÁT HIỆN XÂM NHẬP**

**GIẢNG VIÊN HƯỚNG DẪN:**

**T.S Nguyễn Thế Xuân Ly**

**SINH VIÊN THỰC HIỆN:**

**1. Võ Hoàng Bảo Nhóm: 20.15B MSSV: 102200246**

**2. Bùi Hải Nam Nhóm: 20.15B MSSV: 102200273**

**3. Nguyễn Hoàng Quân Nhóm: 20.15B MSSV: 102200281**

**Đà Nẵng, 12/2022**

MỤC LỤC

[MỤC LỤC 2](#_Toc90369523)

[DANH SÁCH HÌNH VẼ 3](#_Toc90369524)

[DANH SÁCH BẢNG BIỂU 4](#_Toc90369525)

[DANH SÁCH CÁC TỪ VIẾT TẮT 5](#_Toc90369526)

[MỞ ĐẦU (GIỚI THIỆU ĐỀ TÀI) 6](#_Toc90369527)

[CHƯƠNG 1. CƠ SỞ LÝ THUYẾT 7](#_Toc90369528)

[1.1. …. 7](#_Toc90369529)

[1.2. …. 7](#_Toc90369530)

[CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG 7](#_Toc90369531)

[2.1. …. 7](#_Toc90369532)

[2.2. …. 7](#_Toc90369533)

[CHƯƠNG 3. TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ 7](#_Toc90369534)

[3.1. …. 7](#_Toc90369535)

[3.2. …. 7](#_Toc90369536)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 7](#_Toc90369537)

[TÀI LIỆU THAM KHẢO 8](#_Toc90369538)

[[1] Tên tác giả, Tên tài liệu, Tên nhà xuất bản, năm xuất bản 8](#_Toc90369539)

[[2] Tên chủ sở hữu, Tên bài viết, url, ngày truy cập 8](#_Toc90369540)

[PHỤ LỤC 9](#_Toc90369541)

DANH SÁCH HÌNH VẼ

No table of figures entries found.

DANH SÁCH BẢNG BIỂU

No table of table entries found.

DANH SÁCH CÁC TỪ VIẾT TẮT

No table of abbreviation entries found.

MỞ ĐẦU (GIỚI THIỆU ĐỀ TÀI)

Xã hội ngày càng phát triển, Internet trở thành một phần không thể thiếu đối với từng cá nhân, doanh nghiệp, các tổ chức, trường học. Internet đã trở thành công cụ, phương thức giúp cho các doanh nghiệp tiếp cận với khách hàng, cung cấp dịch vụ, quản lý dữ liệu của tổ chức một cách hiệu quả và nhanh chóng. Cùng với sự phát triển theo chiều hướng tốt, các cuộc tấn công và xâm nhập mạng của những kẻ xấu cũng phát triển theo. Không chỉ trên thế giới mà ở Việt Nam vấn đề “An toàn thông tin” đã và đang trở thành vấn đề nóng bỏng. Sự đa dạng và phức tạp trong các loại hình tấn công đã gây ra nhiều khó khăn cho việc ngăn chặn và phòng chống. Một hệ thống phòng chống và phát hiện xâm nhập sẽ giúp người quản trị có thể luôn luôn theo dõi và thu nhập nhiều thông tin đánh giá cho quá trình chống lại các hình thức tấn công và xâm nhập đó.

Hiện nay, các chương trình bảo mật, phòng chống virus, giám sát bảo vệ hệ thống đều có giá thành cao và được phát triển ở nước ngoài. Ngoài ra, các chương trình giám sát hầu hết được tích hợp trên các thiết bị phần cứng nên việc khai thác chức năng, hoặc người dùng tự phát triển mở rộng thêm chức năng của các chương trình này nhằm phục vụ cho công việc quản trị mạng bị hạn chế. Vì thế, nhu cầu có được một hệ thống hỗ trợ giám sát và bảo vệ hệ thống mạng trực quan nhằm giúp cho công việc quản trị mạng được tập trung và đạt hiệu quả cao là rất cần thiết. Đó là lý do mà chúng em chọn nghiên cứu và thực hiện đề tài: “Tìm hiểu hệ điều hành Linux và xây dựng hệ thống phát hiện xâm nhập”.

Vì thời gian thực hiện không nhiều, trong quá trình thực hiện mặc dù đã có nhiều cố gắng nên chắc chắn không tránh khỏi thiếu sót, mong nhận được sự góp ý từ quý thầy cô để chúng em hoàn thiện hơn.

Đặc biệt chúng em xin chân thành cảm ơn giảng viên ThS. Nguyễn Thế Xuân Ly đã tận tình giúp đỡ chúng em trong quá trình thực hiện đề tài này.

# CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

## Giới thiệu về hệ điều hành Linux

1.1.1 Giới thiệu chung

Linux là tên gọi của một hệ điều hành máy tính và cũng là tên hạt nhân của hệ điều hành. Nó có lẽ là một ví dụ nổi tiếng nhất của phần mềm tự do và của việc phát triển mã nguồn mở.

Phiên bản Linux đầu tiên do Linus Torvalds viết vào năm 1991, lúc ông còn là một sinh viên của Đại học Helsinki tại Phần Lan. Ông làm việc một cách hăng say trong vòng 3 năm liên tục và cho ra đời phiên bản đầu tiên Linux 1.0 vào năm 1994. Bộ phận chủ yếu này được phát triển và tung ra trên thị trường dưới bản quyền GNU General Public License. Do đó mà bất cứ ai cũng có thể tải và xem mã nguồn của Linux.

Một cách chính xác, thuật ngữ “Linux” được sử dụng để chỉ Nhân Linux, nhưng tên này được sử dụng một cách rộng rãi để miêu tả tổng thể một hệ điều hành giống Unix (còn được biết đến dưới tên GNU/Linux) được tạo ra bởi việc đóng gói nhân Linux cùng với các thư viện và công cụ GNU, cũng như là các bản phân phối Linux. Thực tế thì đó là tập hợp một số lượng lớn các phần mềm như máy chủ web, các ngôn ngữ lập trình, các hệ quản trị cơ sở dữ liệu, các môi trường làm việc desktop như GNOME và KDE, và các ứng thích hợp cho công việc văn phòng như OpenOffice.

Khởi đầu, Linux được phát triển cho dòng vi xử lý 386, hiện tại hệ điều hành này hỗ trợ một số lượng lớn các kiến trúc vi xử lý, và được sử dụng trong nhiều ứng dụng khác nhau từ máy tính cá nhân cho tới các siêu máy tính và các thiết bị nhúng như là các máy điện thoại di động.

Ban đầu, Linux được phát triển và sử dụng bởi những người say mê. Tuy nhiên, hiện nay Linux đã có được sự hỗ trợ bởi các công ty lớn như IBM và Hewlett-Packard, đồng thời nó cũng bắt kịp được các phiên bản Unix độc quyền và thậm chí là một thách thức đối với sự thống trị của Microsoft Windows trong một số lĩnh vực. Sở dĩ Linux đạt được những thành công một cách nhanh chóng là nhờ vào các đặc tính nổi bật so với các hệ thống khác: chi phí phần cứng thấp, tốc độ cao (khi so sánh với các phiên bản Unix độc quyền) và khả năng bảo mật tốt, độ tin cậy cao (khi so sánh với Windows) cũng như là các đặc điểm về giá thành rẻ, không bị phụ thuộc vào nhà cung cấp. Một đặc tính nổi trội của nó là được phát triển bởi một mô hình phát triển phần mềm nguồn mở hiệu quả.

Tuy nhiên, hiện tại số lượng phần cứng được hỗ trợ bởi Linux vẫn còn rất khiêm tốn so với Windows vì các trình điều khiển thiết bị tương thích với Windows nhiều hơn là Linux. Nhưng trong tương lai số lượng phần cứng được hỗ trợ cho Linux sẽ tăng lên.

1.1.2 Ưu điểm của hệ điều hành Linux

*Bản quyền và chi phí hợp lý*

Với bản chất là một mã nguồn mở, Linux được phát triển miễn phí cho người sử dụng. Người dùng có thể nhìn thấy hầu như mọi dòng code trong Linux. Trong đó Windows bản quyền và bộ Office bản quyền sẽ phải mất khoảng vài triệu đồng để sở hữu tương tự.

Sử dụng Linux, bạn sẽ được sử dụng miễn phí tất cả các tính năng kèm bộ ứng dụng cho người dùng văn phòng miễn phí. Học tập, xem phim, nghe nhạc, chat,… mọi hoạt động của bạn không phải lo lắng gì về vấn đề bản quyền nữa.

*Hỗ trợ tốt cho lập trinh viên – quản trị mạng*

Hệ thống Linux hoạt động ổn định, hiệu quả cao. Do đó, Linux được hỗ trợ tốt cho nhiều công việc, đặc biệt là đối với những công việc đòi hỏi tính ổn định hệ thống cao như quản trị mạng và lập trình viên.

*Sự hỗ trợ kỹ thuật tích cực*

Linux cung cấp dịch vụ hỗ trợ kỹ thuật 24/7 trong cả năm đối với mọi dịch vụ. Có thể kể đến như Red Hat, Novell và Canonical.

*Linh hoạt*

Nếu hiểu sâu về Linux, bạn hoàn toàn có thể dễ dàng chỉnh sửa theo ý mình. Linux đúng là một môi trường lý tưởng cho các lập trình viên cũng như các nhà phát triển vì Linux mang lại sự tương thích với rất nhiều môi trường khác nhau.

*Giao diện phong phú*

Với hệ điều hành Linux, giao diện hoàn toàn tách rời với hệ thống. Do đó, bạn có thể đổi môi trường giao diện mà không cần lo lắng xem phải cài lại chương trình hay không. Bạn có thể tha hồ chọn giao diện như GNOME, KDE hay gần đây hơn là Unity.

*Độ an toàn và khả năng bảo mật, hạn chế sự tấn công của các mã độc và virut cao*

Nếu như trên Windows, những con virut, mã độc,… luôn là thứ khiến bạn phải chật vật đối mặt với Linux sẽ mang lại cho bạn cảm giác an toàn. Điều tuyệt vời là tất cả bọn chúng đều không thể hoạt động được trên nền tảng này. Công việc của bạn chỉ là xóa khi thấy bọn chúng trong USB hay ổ cứng di động (ổ cứng SSD).

*Hoạt động tốt ngay cả trên những máy có cầu hình yếu*

Thông thường, một phiên bản mới được nâng cấp sẽ kèm theo yêu cầu phần cứng cũng phải nâng lên theo. Vậy trong trường hợp cấu hình máy tính của bạn không đủ đáp ứng yêu cầu của phiên bản mới thì sao?

Linux sẽ giúp bạn tránh khỏi những lo lắng đó. Với sự nâng cấp, hỗ trợ thường xuyên từ cộng đồng lập trình, Linux tự tin hoạt động mượt mà cực kỳ ổn định ngay ca trên các máy tính có cấu hình thấp.

1.1.3 Nhược điểm của hệ điều hành Linux

Bên cạnh những ưu điểm trên, hệ điều hành Linux vẫn còn một số điểm hạn chế:

* Số lượng ứng dụng hỗ trợ trên Linux chưa phong phú, còn khá ít sự lựa chọn cho người dùng.
* Một số nhà sản xuất không phát triển Driver hỗ trợ nền tảng Linux.
* Bạn sẽ mất một khoảng thời gian để “thích nghi” nếu chuyển từ hệ điều hành Window sang Linux vì thực sự khá khó để làm quen với Linux.
* Tùy vào tính chất, nhu cầu sử dụng mà bạn cần tìm một hệ điều hành thích hợp với mình.

## Tổng quan về IDS

1.2.1 Giới thiệu

Hệ thống phát hiện xâm nhập (Intrusion Detection System - IDS) là một hệ thống giám sát lưu thông mạng, các hoạt động khả nghi và cảnh báo cho hệ thống, nhà quản trị.

Ngoài ra IDS cũng đảm nhận việc phản ứng lại với các lưu thông bất thường hay có hại bằng cách thực hiện các hành động đã được thiết lập trước như khóa người dùng hay địa chỉ IP nguồn đó không cho truy cập hệ thống mạng, …

IDS cũng có thể phân biệt giữa những tấn công từ bên trong hay tấn công từ bên ngoài. IDS phát hiện tấn công dựa trên các dấu hiệu đặc biệt về các nguy cơ đã biết (giống như cách các phần mềm diệt virus dựa vào các dấu hiệu đặc biệt để phát hiện và diệt virus) hay dựa trên so sánh lưu thông mạng hiện tại với baseline (thông số đo đạc chuẩn của hệ thống) để tim ra các dấu hiệu khác thường.

1.2.2 Hệ thống phát hiện xâm nhập

Intrusion Detection Systems (IDS) có thể là một thiết bị phần cứng (các thiết bị phát hiện xâm nhập của Cisco (Cisco IDSM-2 hoặc Cisco IPS 4200 Series Sensors)) hoặc cũng có thể là một ứng dụng phần mềm giúp giám sát máy tính, hệ thống mạng trước các hành động đe dọa đến hệ thống hoặc vi phạm chính sách an ninh và báo cáo lại cho người quản trị hệ thống.

1.2.3 Cấu trúc chung

Cấu trúc hệ thống IDS phụ thuộc vào kiểu phương pháp được sử dụng để phát hiện xâm nhập, các cơ chế xử lý khác nhau được sử dụng đối với một IDS. Mô hình cấu trúc chung cho các hệ IDS là:

Diagram

Description automatically generated

Hình 1: Mô hình chung hệ thống IDS

Nhiệm vụ chính của các hệ thống phát hiện xâm phạm là phòng chống cho một hệ thống máy tính bằng cách phát hiện các dấu hiệu tấn công và có thể đẩy lùi nó. Việc phát hiện các tấn công phụ thuộc vào số lượng và kiểu hành động thích hợp. Để ngăn chặn xâm phạm tốt cần phải kết hợp tốt giữa “bả và bẫy” được sử dụng để xác định các mối đe dọa. Việc làm lệnh hướng sự tập trung của kẻ xâm nhập vào tài nguyên được bảo vệ cũng là một nhiệm vụ quan trọng. Cả hệ thống thực và hệ thống bẫy cần phải được kiểm tra một cách liên tục. Dữ liệu được tạo ra bằng các hệ thống phát hiện xâm nhập được kiểm tra một cách cẩn thận (đây là nhiệm vụ chính cho mỗi IDS) để phát hiện các dấu hiệu tấn công.

Khi một sự xâm nhập được phát hiện, IDS đưa ra các cảnh báo đến các quản trị viên hệ thống về sự việc này. Bước tiếp theo được thực hiện bởi các quản trị viên hoặc có thể là bản thân IDS bằng cách lợi dụng các tham số đo bổ sung (các chức năng khóa để giới hạn các session, backup hệ thống, định tuyến các kết nối đến bẫy hệ thống, cơ sở hạ tầng hợp lệ,…) – theo các chính sách bảo mật của các tổ chức. Một IDS là một thành phần nằm trong chính sách bảo mật.

Giữa các nhiệm vụ IDS khác nhau, việc nhận ra kẻ xâm nhập là một trong những nhiệm vụ cơ bản. Nó cũng hữu dụng trong việc nghiên cứu mang tính pháp lý các tình tiết và việc cài đặt các bản vá thích hợp để cho phép phát hiện các tấn công trong tương lai nhằm vào các cá nhân cụ thể hoặc tài nguyên hệ thống.

Phát hiện xâm nhập đôi khi có thể đưa ra các báo cảnh sai, ví dụ những vấn đề xảy ra do trục trặc về giao diện mạng hoặc việc gửi phần mô tả các tấn công hoặc các chữ ký thông qua email.

Cấu trúc của một hệ thống phát hiện xâm phạm dạng tập trung:

Diagram

Description automatically generated

Hình 2: Cấu trúc tập trung

Bộ cảm biến được tích hợp với thành phần sưu tập dữ liệu – một 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. Trong trường hợp nào đó, ví dụ, khi luồng dữ liệu sự kiện được truyền tải trực tiếp đến bộ phân tích mà không có sự lưu dữ liệu nào được thực hiện. Điều này cũng liên quan một chút nào đó đến các gói mạng.

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.

Diagram

Description automatically generated

Hình 3: Cấu trúc đa tác nhân

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. IDS 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 ý nghĩa bảo vệ liên quan đến các kiểu tấn công mới. Các giải pháp dựa trên tác nhân IDS tạo cơ chế ít phức tạp hơn cho việc nâng cấp chính sách đáp trả.

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.

1.2.4 Những kĩ thuật phát hiện xâm nhập

1.2.4.1 Phát hiện sự bất thường (Anomaly Based ID):

Phát hiện giao thức bất thường (Protocol Anomaly Detection). Giao thức bất thường nghĩa là những trường hợp vi phạm các định dạng, các tiêu chuẩn các hành vi đã được quy định thành chuẩn Internet từ trước đó.

Vd: Kích thước gói tin icmp tối đa là 65,535 bytes, attacker cố tình gửi một gói tin có kích thước lớn hơn tiêu chuẩn đó để làm tràn bộ đệm (ping of death).

Phát hiện thâm nhập dựa trên sự thống kê bất thường (Statistical Anomaly Based ID). Kĩ thuật này nhấn mạnh việc đo đếm các hoạt động bình thường trên mạng. Ví dụ số tiến trình hoạt động quá mức trên CPU, số lượng gói tin gửi đi quá mức,…

1.2.4.2 Phát hiện sự lạm dụng/ dấu hiệu (Misuse/Signature Based ID)

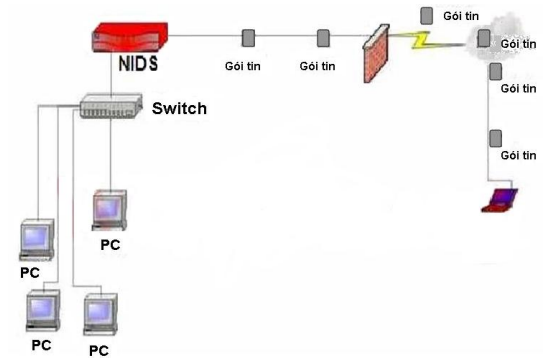
Phát hiện sự lạm dụng là phát hiện những kẻ xâm nhập đang cố gắng đột nhập vào hệ thống mà sử dụng một số kỹ thuật đã biết. Nó liên quan đến việc mô tả đặc điểm các cách thức xâm nhập vào hệ thống đã được biết đến, mỗi cách thức này được mô tả như một mẫu. Hệ thống phát hiện sự lạm dụng chỉ thực hiện kiểm soát đối với các mẫu đã rõ ràng. Mẫu có thể là một xâu bit cố định (ví dụ như một virus đặc tả việc chèn xâu),… dùng để mô tả một tập hay một chuỗi các hành động đáng nghi ngờ.

1.2.5 Phân biệt mô hình IDS

Có 2 mô hình IDS là Network Based IDS(NIDS) và Host Based IDS (HIDS).

NIDS

Được đặt giữa kết nối hệ thống mạng bên trong và mạng bên ngoài để giám sát toàn bộ lưu lượng vào ra. Có thể là một thiết bị phần cứng riêng biệt được thiết lập sẵn hay phần mềm cài đặt trên máy tính. Chủ yếu dùng để đo lưu lượng mạng được sử dụng. Tuy nhiên có thể xảy ra hiện tượng nghẽn cổ chai khi lưu lượng mạng hoạt động ở mức cao.



Hình …: Mô hình NIDS

Một số sản phẩm NIDS:

- Cisco IDS

- Dragon IDS/IPS

HIDS

Được cài đặt cục bộ trên một máy tính làm cho nó trở nên linh hoạt hơn nhiều so với NIDS. Kiểm soát lưu lượng vào ra trên một máy tính, có thể được triển khai trên nhiều máy tính trong hệ thống mạng. HIDS có thể được cài đặt trên nhiều dạng máy tính khác nhau cụ thể như các máy chủ, máy trạm, máy tính xách tay. HIDS cho phép bạn thực hiện một cách linh hoạt trong các đoạn mạng mà NIDS không thể thực hiện được. Lưu lượng đã gửi tới máy tính HIDS được phân tích và chuyển qua nếu chúng không chứa mã nguy hiểm. HIDS được thiết kế hoạt động chủ yếu trên hệ điều hành Windows, mặc dù vậy vẫn có các sản phẩm hoạt động trong nền ứng dụng UNIX và nhiều hệ điều hành khác.

Diagram

Description automatically generated

Hình…: Mô hình HIDS

## Hệ thống IDS Snort

1.3.1 Giới thiệu

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.

1.3.2 Snort là gì?

Snort là một số hệ thống phát hiện xâm nhập hàng đầu bằng phần mềm. Snort là một sản phẩm IDS Open source của công ty SourceFire, họat động dựa trên một tập luật linh họat, thông qua phân tích các protocol, tìm kiếm nội dung và các bộ tiền xử lý (preprocessor) để phát hiện ra hàng ngàn lọai sâu (worm), các kiểu tấn công, quét cổng và những hành động đáng ngờ khác trên mạng.

1.3.3 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:

Bộ phận giải mã gói tin (Packet Decoder).

Bộ phận tiền xử lý (Preprocessors).

Bộ phận phát hiện (Detection Engine).

Bộ phận phát hiện (Detection Engine).

Bộ phận ghi nhận và thông báo (Loging and alerting system).

Bộ phận đầu ra (Output Modules).

Bất kì dữ liệu nào đến từ internet đều đi vào packet decoder. Trên đường đi của nó với các module đầu ra, nó có thể bị loại bỏ, ghi nhận hoặc có một cảnh báo được tạo ra.

Diagram

Description automatically generated

Hình a: Các thành phần của Snort

1.3.3.1 Bộ phận giải mã gói tin (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, 802.11, FDDI, 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 tiêu đề của giao thức, các gói tin quá dài, bất thường hoặc không chính xác tùy chọn TCP, UDP, ICMP và IP 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 preprocessor.

1.3.3.2 Bộ phận tiền xử lý (reprocessors)

Bộ phận tiền xử lý là một thành phần rất quan trọng đối với bất kỳ một hệ thống IDS nào để có thể chuẩn bị gói dữ liệu đưa vào cho module phát hiện phân tích. Ba nhiệm vụ chính của các module 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 phân thành các luồng riêng giúp Snort có thể hiểu được các phiên làm việc khác nhau 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. 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 module 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 module 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 lí 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 module 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 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. Backoffice dùng để đưa ra cảnh báo khi hệ thống đã bị nhiễm troạn backoffice và kẻ tấn công từ xa kết nối tới backoffice thực hiện các lệnh từ xa.

1.3.3.3 Bộ phận phát hiện (Detection Engine)

Bộ phận phát hiện: Đây là module quan trọng nhất của Snort. Nó chịu trách nhiệm phát hiện các dấu hiệu xâm nhập. Module 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 module 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 hôi được cùng lúc. Khả năng xử lý của module 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 module 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,…

Packet payload (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 module 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.

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 phúc 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.

1.3.3.4 Bộ phận ghi nhận và thông báo (Loging 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.

1.3.3.5 Bộ phận đầu ra (Output Modules)

Bộ phận đầu ra: module 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 các cảnh báo vào cơ sở dữ liệu.

Tạo file log dưới dạng xml: ghi log file dạng xml rất thuận tiện cho việc trao đổi và chia sẻ dữ liệu.

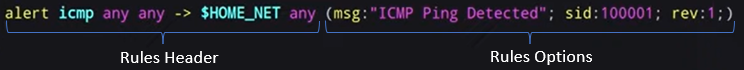
Cấu hình lại Router, Firewall.

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 về hệ thống IDS một cách tập trung và thuận tiện hơn.

Gửi các thông điêp 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 module thể hiện thông tin riêng tùy theo mục đích sử dụng.

1.3.4 Cấu trúc luật của Snort

Ví dụ:



Hình…: Ví dụ một luật đơn giản của Snort

Một luật trong Snort được chia thành hai phần header và options. Phần header bao gồm: rule action, protocol, địa chỉ IP nguồn, địa chỉ IP đích, subnetmask, port nguồn, port đích. Phần options bao gồm các thông điệp cảnh báo, thông tin các phần của gói tin sẽ được kiểm tra để xác định xem hành động nào sẽ được áp dụng.

Rules được chia thành 2 thành phần chính là rules header và rules options.

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.

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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Action | Protocol | Address | Port | Direction | Address | Port |

Hình a: Header luật của Snort

- Action: là phần quy 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.

- Protocol: 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 “->” quy định.

- Port: xác định các cổng nguồn và đích của một gói tin mà trên đó luật được áp dụng.

- Direction: phần này sẽ chỉ ra đâu là địa chỉ nguồn, đâu là địa chỉ đích.

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 “ICMP Ping Detected” sẽ được tạo khi tìm thấy điều kiện là phát hiện ping.

1.3.5 Mô hình cho hệ thống IDS-SNORT

Diagram

Description automatically generated

Hình 4: Mô hình IDS-SNORT

Trên đây là mô hình cho hệ thống giám sát phát hiện xâm nhập hệ thống SNORT. Với máy chủ IDS-SNORT được cài đặt và cấu hình SNORT để phát hiện kiểm tra các gói tin.

1.3.6 Các chế độ thực thi của Snort

1.3.6.1 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.

1.3.6.2 Packet 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 Is-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 port 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 tốc độ 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 cỡ -r và phần hiển thị giống nhưu ở mode sniffer:

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

Trong phần này Snort không giới hạn để dọc 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.

1.3.7 Cài đặt Snort

*Cập nhật hệ thống*

Để đảm bảo rằng tất cả các phụ thuộc của Snort đều được cập nhật, hãy thực thi lệnh bên dưới:

# apt-get update && apt-get upgrade

Trong quá trình cập nhật hệ thống chúng ta chờ từ 5-15 phút.

*Cài đặt thư viện cần thiết*

Trước khi cài dặt Snort, chúng ta cần cài đặt các thư viện cần thiết trên hệ thống.

# apt install build-essential libpcap-dev libpcre3-dev libnet1-dev zlib1g-dev luajit hwloc libdnet-dev libdumbnet-dev bison flex liblzma-dev openssl libssl-dev pkg-config libhwloc-dev cmake cpputest libsqlite3-dev uuid-dev libcmocka-dev libnetfilter-queue-dev libmnl-dev autotools-dev libluajit-5.1-dev libunwind-dev -y

Sau khi các gói phụ thuộc được cài đặt, chúng ta sẽ tiến hành tạo một thư mục nơi biên dịch và lưu giữ các file cho Snort bằng lệnh sau:

# mkdir snort-source-files

Sau đó, tải và cài đặt phiên bản mới nhất của thư viện Snort (LibDAQ). Để cài đặt LibDAQ, chúng ta sẽ cần xây dựng và cài đặt bằng lệnh sau.

# git clone <https://github.com/snort3/libdaq.git>

cd libdaq

./bootstrap

./configure

Make

Make install

Cài đặt Snort 3 trên Ubuntu 20.04

Sau khi các gói phụ thuộc được thiết lập, chúng ta sẽ tải xuống và cài đặt Snort 3 trên Ubuntu 20.04.

# cd ../

Git clone <https://github.com/snortadmin/snort3.git>

Di chuyển thư mục snort 3 và thực thi câu lệnh sau:

# cd snort3/

./configure\_cmake.sh –prefix=/usr/local –enable-tcmalloc

Tiếp theo di chuyển đến thư mục build để biên dịch và cài đặt Snort 3:

# cd build/

make

make install

Tiến hành cập nhật thư viện

# sudo ldconfig

Sau đó tiến hành tạo symlink thư mục /usr/sbin.snort:

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

Sau khi cài đặt thành công chúng ta tiến hành kiểm tra version của snort:

# snort -V

Trong Snort, Rulesets là lợi thế chính cho công cụ phát hiện xâm nhập.

Có ba loại quy tắc Snort: Community Rules, Registered Rules, Subscriber Rules/

Chúng ta sẽ cài đặt Community Rules.

Tạo một thư mục cho Community Rules trong /usr/local/etc/

# mkdir /usr/local/etc/rules

## Giới thiệu về IPtables

Iptables là một tường lửa ứng dụng lọc gói dữ liệu rất mạnh, miễn phí và có sẵn trên Linux. Netfilter/Iptables gồm 2 phần là Netfilter ở trong nhân Linux và Iptables nằm ngoài nhân. Iptables chịu trách nhiệm giao tiếp giữa người dùng và Netfilter để đẩy các luật của người dùng vào cho Netfiler xử lí. 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.

Diagram

Description automatically generated

Hình 2: Sơ đồ Netfilter/Iptables

**1.4.1 Cơ chế xử lý package trong IPtables**

Tất cả gói đi qua một chuỗi được xây dựng trong các bảng (hàng đợi) để xử lý.

Iptables được chia làm 3 bảng:

*1.4.1.1 NAT table*

NAT table: thực thi chức năng NAT (Network Address Translation), bao gồm ba loại built-in chains sau đây:

+ Pre-routing chain: 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-routing chain: 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.

+ OUTPUT: Trong loại này firewall thực hiện quá trình NAT.

Diagram

Description automatically generated

Hình 3: Quá trình xử lý gói tin trong bảng NAT

*1.4.1.2 Filter table*

Filter table: chịu trách nhiệm thiết lập bộ lọc packet (packet filtering), có ba loại built-in chains:

- 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.

Diagram

Description automatically generated

Hình 3: Quá trình xử lý gói tin trong bảng Filter

1.4.1.3 Mangle table

Mangle table: Chịu trách nhiệm biến đổi quality of service bits trong TCP header.

Diagram

Description automatically generated

Hình 4: Quá trình xử lý gói tin trong bảng Mangle

**1.4.2 Một số khái niệm trong Iptables.**

**- Chain:** là một chuỗi bao gồm nhiều Rule, mặc định các bảng có sẵn các chain như INPUT, OUTPUT …

- **Rule:** là các luật đưa ra các điều kiện so sánh, tập hợp nhiều rule tạo thành Chain.

- **Jump:** là cơ chế chuyển một packet đến một target nào đó để xử lý thêm một số thao tác khác.

- **Target:** là quá trình xử lý khi các gói tin đủ điều kiện. Các target được xây dựng sẵn trong iptables như:

+ ACCEPT: chấp nhận chuyển gói tin đi qua.

+ DROP: chặn những gói tin.

+ REJECT: chặn những gói tin và báo lại cho người gửi.

+ DNAT: thay đổi địa chỉ đích của gói tin.

+ SNAT: thay đổi địa chỉ nguồn của gói tin.

+ MASQUERADING: là một kỹ thuật đặc biệt của SNAT.

**1.4.3 Các kiểu chặn trong IPtables**

1.4.3.1 Chặn theo giao thức

Chặn theo giao thức: Dựa trên giao thức TCP, UDP và các port.

**Bảng 1: Tùy chọn chặn theo giao thức trong IPtables.**

|  |  |
| --- | --- |
| Sử dụng với tùy chọn  -p tcp hoặc – udp | Giải thích |
| --sprort <port> | Địa chỉ nguồn: có thể là một port hoặc một dải port.  Start-portnumber:end-portnumber |
| --dport<port> | Địa chỉ đích: có thể là một port hoặc một dải port.  Start-portnumber:end-portnumber |
| --syn(chỉ sử dụng cho -p tcp) | Sử dụng để xác định yêu cầu thiết lập connection mới. |

Ví dụ:

# iptables -A FORWARD -s 0/0 -i eth0 -d 192.168.1.58 -o eth1 -p TCP -sport 1024:65535 -dport 80 -j ACCEPT

Trong ví dụ này cho phép chuyển các gói tin đi vào từ interface eth0 và ra interface eth1 có địa chỉ nguồn bất kỳ và địa chỉ đích là 192.168.1.58 với giao thức TCP có port nguồn từ 1024 đến 65535 và port đích là 80 (http).

1.4.3.2 Chặn gói ICMP (Ping).

**Bảng 2: Tùy chọn chặn gói ICMP trong IPtables.**

|  |  |
| --- | --- |
| Sử dụng với tùy chọn  ---icmp-type | Giải thích |
| --icmp-type<type> | Sử dụng với các tùy chọn thông dụng như: echo-reply và echo-request |

Ví dụ:

#iptables -A OUTPUT -p icmp --icmp-type echo-request -j ACCEPT

#iptables -A INPUT -p icmp --icmp-type echo-reply -j ACCEPT

Trong ví dụ này Iptables cho phép các máy tính trong mạng nội bộ ping ra ngoài.

1.4.3.3 Chặn dựa trên trạng thái kết nối

**Bảng 3: Tùy chọn chặn dựa trên trạng thái kết nối trong IPtables.**

|  |  |
| --- | --- |
| Sử dụng với tùy chọn  -m multiport | Giải thích |
| --sport<port, port> | Sử dụng để chỉ ra nhiều port nguồn |
| --dport<port, port> | Sử dụng để chỉ ra nhiều port đích |
| --state<state> | Sử dụng với các tùy chọn trạng thái như: ESTABLISHED, NEW, RELATED |

Ví dụ:

#iptables -A FORWARD -s 0/0 -i eth0 -d 192.168.1.58 -o eth1 -p TCP --sport 1024:65535 -m multiport --dport 80,443 -j ACCEPT

Trong ví dụ trên Iptables cho phép chuyển các gói tin từ eth0 đến eth1 với điều kiện các gói tin có địa chỉ nguồn bất kỳ và địa chỉ đích là 192.168.1.58 , giao thức TCP có port nguồn từ 1024 đến 65535 và port đích là 80 (http) và 443 (SSL).

#iptables -A FORWARD -d 0/0 -o eth0 -s 192.168.1.58 -i eth1 -p TCP -m state --state ESTABLISHED -j ACCEPT

Trong ví dụ trên Iptables cho phép chuyển các gói tin từ eth0 đến eth1 với điều kiện các gói tin có địa chỉ nguồn bất kỳ và địa chỉ đích là 192.168.1.58 , giao thức TCP với trạng thái kết nối mới.

**1.4.4 Cài đặt Iptables**

1.4.4.1 Các câu lệnh trợ giúp

#iptables -h

#iptables -m match -h

#iptables -j TARGET -h

#man iptables

1.4.4.2 Các tùy chọn

**Bảng 5: Các tùy chọn trong IPtables**

|  |  |
| --- | --- |
| Tùy chọn | Giải thích |
| -A (--append) | Ghi thêm một rule vào chain |
| -D (--delete) | Xóa rule |
| -E (--rename-chain) | Đổi tên chain |
| -F (--flush) | Xóa tất cả các rule trong một chain |
| -I (--insert) | Chèn thêm rule vào chain tại vị trí được xác định |
| -L (--list) | Xem các rule trong chain |
| -N (--new-chain) | Tạo một chain mới |
| -P (--policy) | Tạo rule mặc định cho chain |
| -R (--replace) | Thay thế một rule bằng một rule khác tại vị trí xác định |
| -V (--version) | Xem phiên bản của Iptables |
| -X (--delete-chain) | Xóa chain |
| -Z (--zero) | Xóa bộ đếm trong chain |
| -t | Chỉ định bảng cho iptables bao gồm: filter, nat, mangle tables |
| -j | Nhảy đến một target |
| -p | Mô tả các giao thức bao gồm: icmp, tcp, udp và all |
| -s | Chỉ định địa chỉ nguồn |
| -d | Chỉ định địa chỉ đích |
| -i | Chỉ định địa chỉ interface cho các gói tin đi ra |
| -m multiport | Chỉ định nhiều dãy cổng |
| -m state –state | Kiểm tra trạng thái:  ESTABLISHED: đã thiết lập connection  NEW: bắt đầu thiết lập connection  RELATED: thiết lập connection thứ 2 |

1.4.4.3 Các câu lệnh phục hồi

-iptables-restore: lệnh phục hồi iptables có hai tùy chọn:

**Bảng 6 Câu lệnh hồi phục và các tùy chọn**

|  |  |
| --- | --- |
| Tùy chọn | Giải thích |
| -c (--counters) | Phục hồi bộ đếm gói và byte cho các rule |
| -n (--noflush) | Tắt chức năng xóa (-flush) trong tables trước khi phục hồi |

-iptables-save: lệnh ghi lại Iptables có hai tùy chọn:

**Bảng 7: câu lệnh hồi phục và các tùy chọn**

|  |  |
| --- | --- |
| Tùy chọn | Giải thích |
| -c (--counters) | Hiện bộ đếm gói và byte cho các rule |
| -t (--table) | Chỉ hiện bảng được chỉ định |

# CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## ….

## ….

# CHƯƠNG 3. TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ

## ….

Demo ở đây

## ….

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

**1. Kết luận**

Thông qua nghiên cứu và triển khai hệ thống IDS SNORT, chúng em đã nhận thức được tình hình an ninh mạng hiện nay và những yêu cầu cần thiết để thiết lập là duy trì một hệ thống mạng an toàn. Bằng cách tìm hiểu các tài liệu liên quan, tham khảo ý kiến của giảng viên và thực hành, chúng em đã nắm vững được các kiến thức cơ sở và thử nghiệm được các tấn công thâm nhập một hệ thống.

Tuy nhiên, đối với vấn đề khá mới này, em cũng vướng phải những khó khăn:

- Thời gian thực hành với công cụ Snort còn ít, chưa vận dụng được tối đa sức mạnh của công cụ này.

- Tuy đã cố gắng nắm bắt được nền tảng của công nghệ IDS, nhưng nhóm chúng em vẫn dừng lại ở mức nghiên cứu lý thuyết, chưa đi sâu vào thực tế.

**2. Hướng phát triển**

Thông qua bài PBL lần này, chúng em nhận thấy việc đảm bảo an ninh mạng ngày nay là một vấn đề cần thiết song cũng khá khó khăn.

Sau khi hoàn thành PBL, chúng em mong muốn tiếp tục nghiên cứu chi tiết hơn về các tấn công mạng một cách có hệ thống (hoặc một cách toàn diện hơn), cũng như các biện pháp bảo đảm an ninh mạng có hiệu quả hơn. Ví dụ, tìm hiểu về cách thức phát hiện ra một lỗ hổng trong hệ thống, đồng thời nghiên cứu cách thức phòng tránh các lỗ hổng đó trước khi kẻ tấn công lợi dụng được nó.

Kết quả nghiên cứu của PBL này sẽ giúp định hướng các nghiên cứu sâu hơn về an ninh mạng trong các môi trường và hệ thống mạng khác sau này.

TÀI LIỆU THAM KHẢO

[1] Tên tác giả, Tên tài liệu, Tên nhà xuất bản, năm xuất bản

[2] Tên chủ sở hữu, Tên bài viết, url, ngày truy cập

PHỤ LỤC