ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

Lê Duy Hưng

PHÁT TRIỂN CÔNG CỤ PHÁT HIỆN VÀ XỬ LÝ CÁC GÓI TIN BẤT THƯỜNG DỰA TRÊN TẬP LUẬT TÙY CHỈNH

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY
Ngành: Truyền thông và Mạng máy tính

HÀ NỘI - 2020

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

Lê Duy Hưng

PHÁT TRIỂN CÔNG CỤ PHÁT HIỆN VÀ XỬ LÝ CÁC GÓI TIN BẤT THƯỜNG DỰA TRÊN TẬP LUẬT TÙY CHỈNH

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY
Ngành: Truyền thông và Mạng máy tính

Cán bộ hướng dẫn: TS. Phạm Mạnh Linh

Cán bộ đồng hướng dẫn: ThS. Đặng Văn Đô

HÀ NỘI - 2020

TÓM TẮT

Hiện nay, đi cùng với sự phát triển của Internet luôn là những vấn đề về an ninh như truy cập trái phép, đánh cắp dữ liệu, tấn công từ chối dịch vụ, …

Các giải pháp truyền thống cho vấn đề này thường là sử dụng Firewall, chương trình diệt virus. Tuy nhiên, trong điều kiện hiện nay, khi mà những cuộc tấn công mạng ngày càng tinh vi hơn, cần có một giải pháp để phát hiện, cảnh báo và xử lý xâm nhập. Hệ thống phát hiện và xử lý xâm nhập được xem là một lựa chọn.

Các hệ thống phát hiện và xử lý xâm nhập hiện nay thường dựa vào tập luật để quyết định sẽ làm gì với mỗi gói tin. Có thể kể đến những công cụ như Suricata, Snort. Các tùy chọn của những công cụ này rất đa dạng, tuy nhiên, nó lại dựa hoàn toàn vào gói tin, chứ không dựa vào những thông tin khác, ví dụ như trạng thái của máy tính tại thời điểm đó.

Vì vậy, em chọn đề tài "Phát triển công cụ phát hiện và xử lý các gói tin bất thường dựa trên tập luật tùy chỉnh". Với mục tiêu nghiên cứu, tìm hiểu và xây dựng một giải pháp để bảo vệ cho hệ thống mạng, đồng thời có thêm khả năng theo dõi mức độ sử dụng tài nguyên của máy chủ và tuỳ chỉnh tập luật để xử lý các gói tin dựa trên từng mức độ sử dụng tài nguyên đó. Đây là một hướng đi mới và có thể tiếp tục phát triển thêm ngoài phạm vi đồ án tốt nghiệp.

LỜI CẢM ƠN

Đầu tiên, em xin gửi lời cảm ơn chân thành tới TS. Phạm Mạnh Linh và ThS. Đặng Văn Đô đã cho em cơ hội được học tập và nghiên cứu, đã nhiệt tình giúp đỡ em trực tiếp giải quyết các vướng mắc kỹ thuật khi thực hiện khóa luận.

Sự hướng dẫn, hỗ trợ và động viên của các Thầy đã giúp em trong việc nghiên cứu và hoàn thành đồ án này. Em xin cảm ơn các thầy, cô trong bộ môn Mạng và Truyền thông máy tính, các thầy cô giảng dạy tại trường Đại học Công nghệ đã giúp đỡ em trong suốt quá trình học tập và nghiên cứu.

Bên cạnh đó, em xin cảm ơn gia đình và bạn bè tại trường Đại học Công nghệ đã đồng hành cùng em suốt hơn bốn năm qua.

Em xin chân thành cảm ơn!

Hà Nội, ngày tháng năm

Sinh viên

Lê Duy Hưng

LỜI CAM ĐOAN

Tôi xin cam đoan rằng mọi kết quả trình bày trong khóa luận đều do tôi thực hiện dưới sự hướng dẫn của TS. Phạm Mạnh Linh và đồng hướng dẫn là ThS. Đặng Văn Đô. Tất cả các tham khảo nghiên cứu liên quan đều được nêu rõ nguồn gốc một cách rõ ràng từ danh mục tài liệu tham khảo trong khóa luận. Khoá luận không sao chép lại từ tổ chức hoặc cá nhân nào khác mà không chỉ rõ về mặt tài liệu tham khảo.

Các thống kê, các kết quả trình bày trong khóa luận đều được lấy từ thực nghiệm khi chạy chương trình. Nếu sai tôi xin hoàn toàn chịu trách nhiệm theo quy định của trường Đại học Công Nghệ - Đại học Quốc gia Hà Nội.

Hà Nội, ngày tháng năm

Sinh viên

Lê Duy Hưng

MỤC LỤC

Bảng các ký hiệu, chữ viết tắt

HIDS	Host-based Intrusion Detection System
HIPS	Host-based Intrusion Prevention System
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection Systems
IP	Internet Protocol
IPS	Intrusion Prevention Systems
NIDS	Network-based Intrusion Detection System
NIPS	Network-based Intrusion Prevention System
TCP	Transmission Control Protocol
UDP	User Datagram Protocol

CHƯƠNG 1. GIỚI THIỆU BÀI TOÁN

Kể từ khi mạng Internet ra đời đến nay, nó đã mang đến sự thay đổi vô cùng to lớn đối với đời sống con người. Thế giới đã có nhiều sự biến đổi và ngày càng phụ thuộc vào công nghệ thông tin nói chung cũng như công nghệ Internet nói riêng. Điều đó cũng dẫn đến một mặt trái, đó là càng ngày càng nhiều các thông tin quan trọng của các cơ quan, tổ chức hay cá nhân được lưu trữ trên các mạng máy tính, trong khi đa số mạng máy tính lại không đảm bảo được độ an toàn, bảo mật thông tin tuyệt đối.

Theo thống kê của Bộ Công an, mỗi năm có hàng nghìn trang mạng của Việt Nam bị tin tặc tấn công nhằm đánh cắp thông tin, chiếm quyền điều khiển, thay đổi, chèn nội dung, cài cắm mã độc… Trong 6 tháng đầu năm 2019, Bộ Công an đã phát hiện trên 2.500 trang tin, cổng thông tin điện tử tên miền quốc gia Việt Nam bị tấn công; hàng trăm ngàn máy tính bị nhiễm mã độc. Đáng lưu ý, Việt Nam xếp thứ 4 trong Top 10 quốc gia bị kiểm soát bởi mạng máy tính ma botet. Điều này chỉ ra rằng chính sách về bảo mật về công nghệ thông tin ở Việt Nam đang chưa được quan tâm và đầu tư đủ mức cần thiết.

Khi một hệ thống thông tin bị tin tặc kiểm soát thì hậu quả sẽ là không thể lường trước được. Đặc biệt, nếu hệ thống đó là một trong những hệ thống trọng yếu của đất nước như hệ thống chính phủ điện tử, hệ thống ngân hàng, hệ thống viễn thông thì những thiệt hại về uy tín, kinh tế là rất lớn.

Để ngăn chặn những truy cập trái phép vào hệ thống, các giải pháp bảo mật truyền thống thường được sử dụng, như Firewall để ngăn chặn kết nối không đáng tin cậy, các thuật toán mã hóa làm tăng độ an toàn cho việc truyền dữ liệu, các chương trình diệt virus. Tuy nhiên các vụ vi phạm bảo mật ngày càng tinh vi, khó lường hơn, luôn cải tiến làm cho các phương pháp bảo mật truyền thống không hiệu quả. Những điều đó dẫn đến yêu cầu phải có một phương pháp bảo mật mới bổ trợ cho những phương pháp bảo mật truyền thống đang ngày càng kém hiệu quả.

Trong bối cảnh đó, việc sử dụng các IDS ngày càng trở nên phổ biến và đóng vai trò không thể thay thế trong chính sách bảo mật và an toàn thông tin của bất kỳ hệ thống thông tin nào.

Nhiệm vụ của IDS là thu thập dữ liệu về lưu lượng trong mạng, tiến hành phân tích, đánh giá, từ đó xác định xem có dấu hiệu của một cuộc tấn công hay không. IDS sẽ cảnh báo cho quản trị viên khi phát hiện dấu hiệu của hành vi đánh cắp hay phá hoại thông tin, và do đó sẽ tăng cường an ninh cho hệ thống.

Hệ thống phát hiện xâm nhập có hai hướng tiếp cận chính là dựa trên phát hiện bất thường và dựa trên dấu hiệu.

Với các tiếp cận dựa trên dấu hiệu, thì hệ thống sẽ xác định dữ liệu đang xét có phải là bất thường hay không bằng cách so sánh với các mẫu tấn công đã có từ trước. Cách tiếp cận này hiện đang được sử dụng rộng rãi tuy nhiên nó có điểm yếu khá rõ ràng là chỉ có thể phát hiện được các cuộc tấn công có dấu hiệu đã biết trước.

Cách tiếp cận dựa trên phát hiện bất thường khắc phục được nhược điểm này, bằng cách tạo ra sơ đồ mô tả "trạng thái bình thường". Một hành vi được hệ thống được coi là "bất thường" nếu có các thông số khác biệt đáng kể với mức "bình thường", dựa vào đó có thể coi rằng các "bất thường" này là dấu hiệu của hành vi tấn công. Có thể thấy hướng tiếp cận dựa trên hành vi bất thường có tính linh hoạt cao hơn và hoàn toàn có thể phát hiện các cuộc tấn công mới mà chưa có dấu hiệu cụ thể.

Hệ thống phát hiện xâm nhập cơ bản được chia làm hai hoại: Hệ thống phát hiện xâm nhập hoạt động trên một máy (HIDS) và hệ thống phát hiện xâm nhập hoạt động trên mạng (NIDS). Mục đích của HIDS là bảo đảm tính toàn vẹn cho máy tính, còn NIDS dùng để phát hiện những cuộc tấn công vào mạng. Ngoài phát hiện, một số IDS còn có khả năng ngăn chặn tấn công, những hệ thống như vậy được gọi là hệ thống xử lý xâm nhập (IPS). Những hệ thống này có khả năng hoạt động trong thời gian thực và có ý nghĩa rất lớn khi đặt trong mạng nội bộ được kết nối internet.

Các giải pháp IDS/IPS phổ biến hiện nay đang đi theo hướng dựa vào các thông tin của gói tin để đưa ra hành động cụ thể, vì vậy em sẽ đi theo hướng phát triển một công cụ có khả năng tương tự như IDS/IPS nhưng sẽ tập trung vào tình trạng của máy tính tại thời điểm gói tin đến để đưa ra hành động (cụ thể là mức độ sử dụng của CPU).

Lợi ích của việc quyết định hành động dựa trên tình trạng hiện tại của máy tính có thể giúp máy tính không bị rơi vào trường hợp thiếu tài nguyên, việc từ chối kết nối đến nếu phần cứng đang hoạt động ở một ngưỡng nào đó sẽ khiến cho phần cứng không bao giờ rơi vào tình trạng phải hoạt động hết công suất, tăng tuổi thọ cho thiết bị, và làm cho máy tính hoạt động ổn định. Cụ thể với trường hợp CPU, mọi kết nối đi đến máy tính đều tăng tải cho CPU, vì vậy một trong những vấn đề của đồ án này là làm sao để CPU không phải hoạt động liên tục ở mức cao trong thời gian dài.

Với đề tài "PHÁT TRIỂN CÔNG CỤ PHÁT HIỆN VÀ XỬ LÝ CÁC GÓI TIN BẤT THƯỜNG DỰA TRÊN TẬP LUẬT TÙY CHỈNH", mục tiêu của đồ án là nghiên cứu về các giao thức cơ bản trong mạng máy tính, an toàn mạng, các mô hình hệ thống phát hiện và xử lý xâm nhập, Netfilter trong Linux, sau đó là tìm hiểu các kỹ thuật để xây dựng một công cụ phát hiện và xử lý gói tin bất thường trên Linux dựa trên bộ giao

thức TCP/IP. Trên cơ sở đó tiến hành xây dựng và cài đặt một công cụ IDS/IPS có thể hoạt động ở cả hai vị trí host hoặc trong mạng.

CHƯƠNG 2. LÝ THUYẾT

2.1. Các giao thức phổ biến ở tầng mạng và tầng giao vận

2.1.1. Internet Protocol (Giao thức Internet)

Internet Protocol là giao thức hướng dữ liệu được sử dụng bởi các máy nguồn và đích để truyền dữ liệu giữa các mạng chuyển mạch gói [4].

Dữ liệu được IP gửi đi theo các gói (packet). Cụ thể, khi gửi một gói tin từ máy đến một thiết bị khác mà máy đó chưa từng liên lạc trước đó, IP sẽ không cần phải thiết lập trước tuyến đường truyền tin, việc định tuyến sẽ diễn ra khi gói tin được gửi đi.

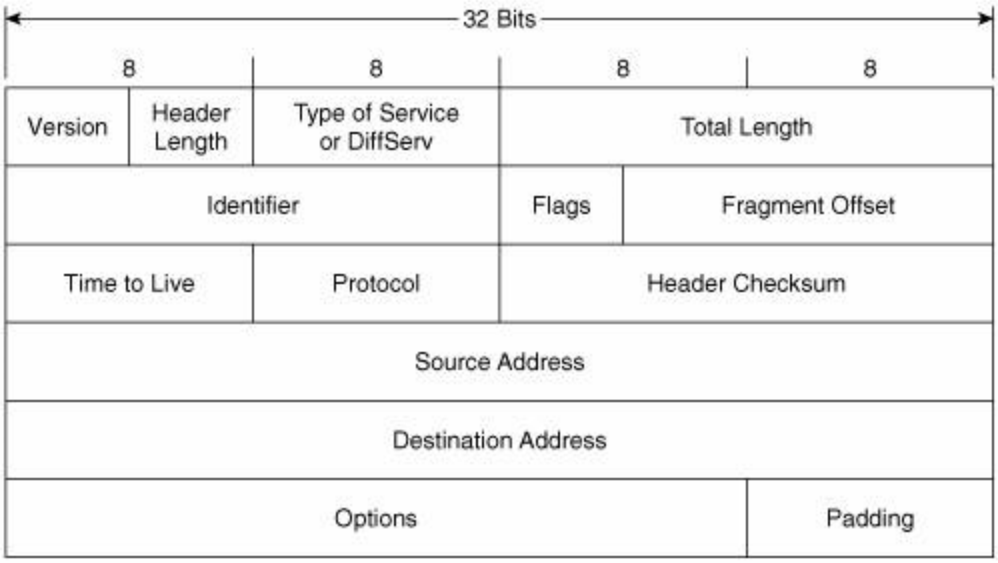
IP không đảm bảo tính toàn vẹn gói dữ liệu được gửi đi. Gói dữ liệu được gửi có thể đến đích mà không còn nguyên vẹn, hoặc không theo thứ tự được gửi ở máy nguồn, nó có thể bị trùng lặp hoặc bị mất hoàn toàn. Việc đảm bảo cho gói tin được gửi đi có thể được cung cấp từ các tầng phía trên IP.

Các thiết bị định tuyến chuyển tiếp các gói tin IP qua các mạng có tầng liên kết dữ liệu được kết nối với nhau. Việc không đảm bảo về gửi dữ liệu đồng nghĩa với việc chuyển mạch gói có thiết kế đơn giản hơn.

Ngày nay, IP rất phổ biến trong mạng Internet. Giao thức tầng mạng thông dụng nhất là IPv4 (giao thức IP phiên bản 4). Do Internet đang hết dần địa chỉ IPv4 nên IPv6 (giao thức IP phiên bản 6) được đề nghị sẽ thay thế IPv4.

IPv4 sử dụng 32 bit để đánh địa chỉ (tạo được khoảng 4 tỷ địa chỉ), IPv6 sử dụng 128 bit để đánh địa chỉ (tạo được khoảng 3.4×1038 địa chỉ).

Địa chỉ IPv4 được chia thành 4 số có giá trị nằm trong khoảng 0 - 255. Mỗi số được lưu bởi 1 byte, IPv4 có kích thước là 4 byte, được chia thành các lớp địa chỉ. Có 5 lớp là A, B, C, D, E.



Hình 2.1. Header của gói tin IPv4

Header của gói tin IPv4 bao gồm 13 trường, với 12 trường trong đó là bắt buộc. Trường thứ 13 là tùy chọn (options). Các trường này được lưu trữ theo kiểu byte có ý nghĩa cao ở địa chỉ thấp, nghĩa là bit có ý nghĩa cao luôn ở địa chỉ thấp.

Phiên bản (Version): Là trường đầu tiên trong header của gói tin IP, version có độ dài 4 bit.

Độ lớn của header (Internet Header Length) (IHL): Là trường thứ hai, có độ dài 4 bit, cho biết số lượng các từ 32-bit trong header. Vì một header của gói tin IPv4 có thể chứa rất nhiều option, IHL cho biết kích thước của header (tương ứng với offset của data). Giá trị nhỏ nhất cho IHL là 5 (RFC 791), do đó gói tin có độ dài nhỏ nhất là 5×32 = 160 bit và độ dài lớn nhất là 15×32 bit = 480 bit.

Differentiated Services (DS): Ban đầu được định nghĩa là trường TOS, hiện tại trường này được định nghĩa trong RFC 2474 là Differentiated services (DiffServ) và trong RFC 3168 là Explicit Congestion Notification (ECN), để phù hợp với IPv6. Chỉ định dịch vụ khi truyền các gói tin IP qua các router. Có độ dài 8 bit, xác định độ trễ, thông lượng, độ trễ, quyền ưu tiên và các đặc tính khác chỉ định độ tin cậy. Trường này gồm TOS (Type of Service) và Precedence. TOS xác định loại dịch vụ, bao gồm: giá trị, độ tin cậy, thông lượng, độ trễ hoặc bảo mật. Precedence xác định mức ưu tiên, có 8 mức với giá trị nằm trong khoảng 0-7. Các công nghệ yêu cầu luồng dữ liệu thời gian thực (real-time data streaming) sẽ sử dụng trường DS.

Total Length: Chỉ định tổng chiều dài gói tin IPv4 (cả phần header và phần data). Có độ dài 16 bit, chỉ định rằng gói tin IPv4 có kích thước nhỏ nhất là 20 byte (chỉ có header không có data) và có lớn nhất là 65.535 byte.

Identification: Định danh gói tin, có độ dài 16 bit. Định danh cho gói tin được cấp bởi nơi gửi gói tin. Nếu gói tin IPv4 bị phân mảnh trong quá trình truyền tin thì tất cả các phân mảnh của gói tin đều sẽ có trường định danh gói tin này, việc này giúp nút đích có thể phục hồi gói tin.

2.1.2. Transmission Control Protocol (Giao thức điều kiển truyền vận)

Transmission Control Protocol là một trong các giao thức của bộ giao thức TCP/IP. Các ứng dụng trên các máy được kết nối mạng sẽ sử TCP để tạo các "kết nối" với nhau, thông qua kết nối đó chúng có thể trao đổi dữ liệu. Đây là một giao thức đảm bảo dữ liệu được truyền đi một cách chính xác và theo đúng thứ tự các gói tin được gửi (truyền tin tin cậy). TCP còn có thể phân biệt giữa dữ liệu của các ứng dụng khác nhau cùng chạy trên cùng một máy tính.

TCP hỗ trợ nhiều giao thức ở tầng ứng dụng, trong đó có WWW, Email và Secure Shell.

Trong bộ giao thức TCP/IP, TCP nằm ở tầng Giao vận (Transport), nằm giữa tầng mạng ở bên dưới và tầng ứng dụng bên trên. Các tiến trình ở tầng ứng dụng thường yêu cầu các kết nối tin cậy để liên lạc với nhau, trong khi đó giao thức IP chỉ cung cấp dịch vụ chuyển gói tin không đáng tin cậy.

Các ứng dụng sẽ gửi dữ liệu cần truyền thành các dòng gồm các byte 8-bit tới TCP. TCP phân chia dòng byte này thành các đoạn ngắn gọi là segment có kích thước thích hợp cho việc chuyền tin, thông thường kích thước segment sẽ được quyết định dựa theo kích thước của đơn vị truyền dẫn tối đa (MTU) ở tầng liên kết dữ liệu của mạng mà máy tính đang kết nối. Sau đó, TCP sẽ chuyển gói tin thu được sau quá trình xử lý cho giao thức IP ở bên dưới để thực hiện việc truyền gói tin qua mạng tới được module TCP của máy đích. TCP gán cho mỗi gói tin "số thứ tự" (sequence number) để có thể thực hiện việc kiểm tra đảm bảo không có gói tin nào bị thất lạc dẫn tới mất mát thông tin. Cũng thông qua số thứ tự này, TCP có thể đảm bảo các gói tin được nhận bởi ứng dụng đích theo đúng thứ tự được gửi. Module TCP tại bên nhận có nhiệm vụ gửi lại "tin báo nhận" (acknowledgement) cho các gói tin đã nhận được thành công, tại bên gửi có một đồng hồ sẽ báo time-out nếu như không nhận được tin xác nhận trong khoản thời gian nhất định gọi là round-trip-time (RTT) nó sẽ coi là truyền tin thất bại và sẽ thực hiện gửi lại. TCP có thể kiểm tra xem có byte nào bị hỏng trong quá trình truyền hay không bằng cách sử dụng trường giá trị kiểm tra (checksum), giá trị của trường checksum được tính toán cho các khối dữ liệu tại nơi gửi sau đó bên nhận sẽ có nhiệm vụ kiểm tra lại.

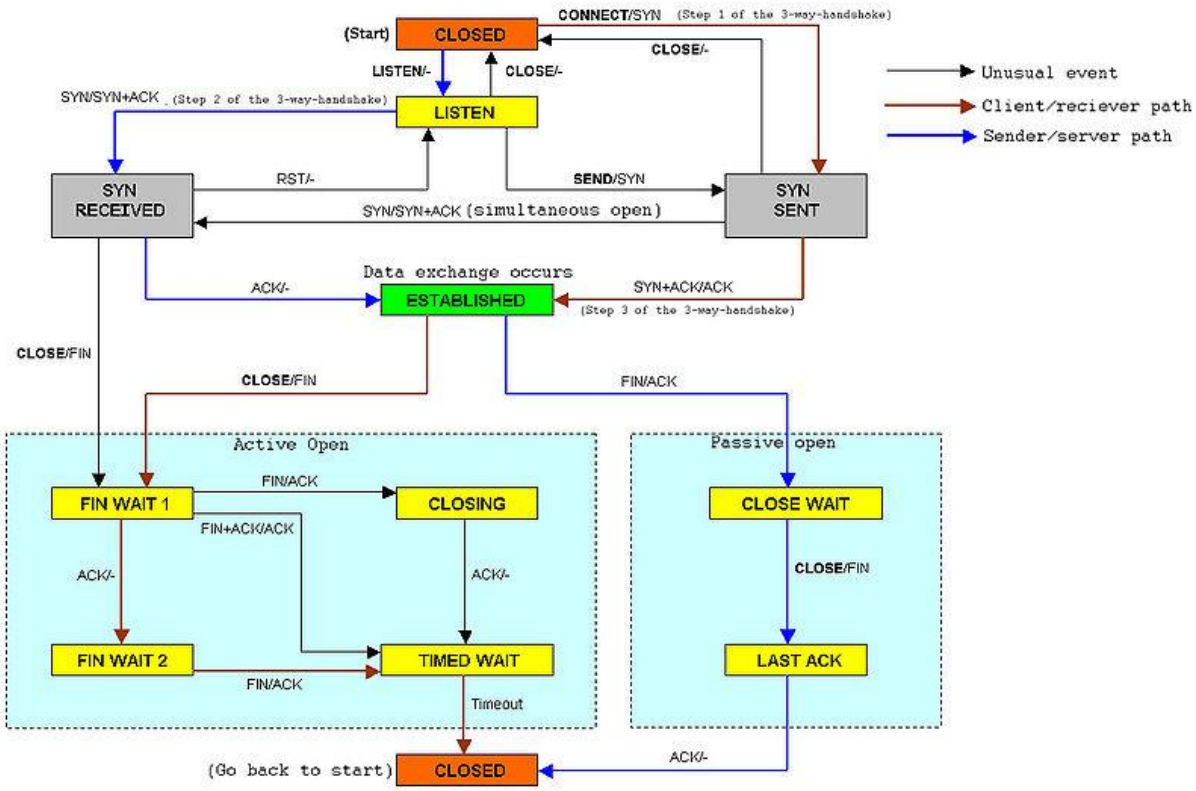
Cách hoạt động của TCP

TCP cần phải thực hiện tạo một kết nối giữa bên gửi và bên nhận trước khi bắt đầu truyền tin, sau khi việc truyền dữ liệu được hoàn thành kết nối sẽ được đóng. Cụ thể, các kết nối TCP sẽ gồm 3 bước:

- Thiết lập kết nối
- Truyền dữ liệu
- Kết thúc kết nối

Các trạng thái khác nhau của một socket:

- LISTEN
- SYN-SENT
- SYN-RECEIVED
- ESTABLISHED
- FIN-WAIT
- CLOSE-WAIT
- CLOSING
- LAST-ACK
- TIME-WAIT
- CLOSER



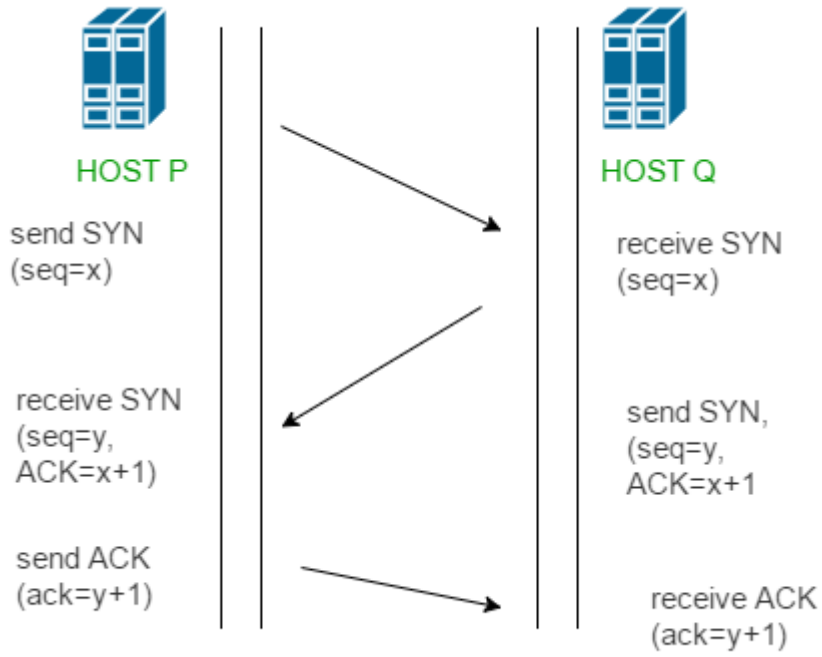
Hình 2.2. Sơ đồ trạng thái của TCP

Thiết lập kết nối

Để có thể tạo một kết nối, TCP sử dụng phương pháp bắt tay 3 bước (3-way handshake). Để có thể chấp nhận kết nối từ client, server phải đăng ký một cổng và mở cổng đó cho các kết nối (mở bị động). Sau khi server đã hoàn tất mở bị động thì client có thể bắt đầu mở chủ động. Quy trình bắt tay 3 bước để thiết lập kết nối TCP xảy ra như sau:

- Client sẽ gửi một gói tin TCP gọi là SYN tới server để yêu cầu mở cổng dịch vụ, trong gói tin này sẽ có một tham số là sequence number, tham số này được gán một giá trị ngẫu nhiên X.
- Server sẽ gửi một gói tin hồi đáp SYN-ACK cho client, gói tin có hai tham số quan trọng là acknowledgment number được gán giá trị bằng X+1 và sequence number được gán một giá trị ngẫu nhiên Y.
- Để quá trình bắt tay ba bước hoàn tất, client sẽ tiếp tục gửi bản tin ACK tới server, tham số trong gói tin này bao gồm sequence number được gán cho giá trị bằng X + 1 và acknowledgment number được gán giá trị bằng Y + 1.

Lúc này, cả client và server đều được xác nhận rằng, một kết nối TCP đã được thiết lập giữa hai bên, và việc truyền dữ liệu đã có thể bắt đầu.



Hình 2.3. Các bước thiết lập kết nối TCP

Truyền dữ liệu

Một số đặc điểm để phân biệt TCP với UDP (khả năng truyền dữ liệu):

- TCP có khả năng truyền dữ liệu không lỗi (nhờ cơ chế sửa lỗi/truyền lại)
- TCP có khả năng truyền các gói dữ liệu theo đúng thứ tự
- TCP có khả năng Truyền lại các gói dữ liệu mất trên đường truyền
- TCP có khả năng loại bỏ các gói dữ liệu bị trùng lặp
- TCP có cơ chế hạn chế tắc nghẽn đường truyền

Ở hai bước đầu tiên trong ba bước bắt tay, server và client trao đổi với nhau một số thứ tự gói ban đầu (Initial Sequence Number - ISN). Số này được chọn một cách hoàn toàn ngẫu nhiên. Số thứ tự này sẽ được mỗi máy tính dùng để đánh dấu các khối dữ liệu gửi đi. Sau mỗi byte được truyền đi, số này lại được tăng lên. Nhờ vậy máy nhận có thể sắp xếp lại các gói tin bất kể chúng tới theo thứ tự nào.

Theo lý thuyết, mỗi byte được gửi đi đều có một số thứ tự và bên nhận sẽ gửi lại tin báo nhận (ACK) mỗi khi nhận được byte đó. Thực tế thì chỉ có byte dữ liệu đầu tiên được gán số thứ tự và bên nhận sẽ gửi tin báo nhận bằng cách gửi số thứ tự của byte đang chờ.

Tin báo nhận là tín hiệu về tình trạng đường truyền giữa hai máy tính. Từ đó, hai bên có thể thay đổi tốc độ truyền nhận dữ liệu để phù hợp với điều kiện đường truyền.

Vấn đề này được gọi là là điều khiển lưu lượng, kiểm soát tắc nghẽn. TCP sử dụng một số cơ chế để ngăn ngừa khả năng nghẽn mạng và đạt được hiệu suất truyền tin cao. Các cơ chế này có thể kể đến: cửa sổ trượt (sliding window), thuật toán tránh nghẽn mạng (congestion avoidance).

Kích thước cửa sổ của giao thức TCP

Kích thước của cửa sổ là kích thước của khối dữ liệu có thể lưu trong bộ nhớ đệm của bên nhận. Đây cũng là lượng thông tin tối đa mà bên gửi có thể gửi đi trước khi nhận được tin báo nhận từ bên nhận.

Dãn kích thước cửa sổ trong TCP

Để tận dụng tối đa khả năng truyền dẫn của mạng thì kích thước của cửa sổ cần được tăng lên. Trường điều khiển kích thước cửa sổ của TCP có độ dài là 2 byte do đó kích thước tối đa của cửa sổ này là 65535 byte.

Do trường không thể thay đổi giá trị của trường điều khiển nên cần sử dụng một hệ số dãn. Hệ số này có thể sử dụng để tăng kích thước tối đa của cửa sổ từ 65535 byte lên tới 1 gigabyte.

Việc tăng kích thước cửa sổ lên mức lớn hơn nữa là cần thiết đối với TCP Tuning.

Việc tăng kích thước cửa sổ chỉ được dùng trong quá trình bắt tay 3 bước. Giá trị của trường co giãn cửa sổ tương ứng với số bit cần được dịch trái đối với giá trị của trường kích thước cửa sổ. Hệ số dãn có thể nằm trong khoảng 0-14

Kết thúc kết nối

Để kết thúc kết nối TCP, hai bên tiến hành quá trình bắt tay 4 bước và chiều của kết nối kết thúc độc lập với nhau. Khi một bên muốn kết thúc kết nối, nó gửi đi một gói tin FIN và bên kia gửi lại gói tin ACK. Vì vậy, trong quá trình kết thúc kết nối sẽ có 2 cặp gói tin trao đổi.

Một kết nối TCP có thể tồn tại ở dạng "nửa mở": một bên chỉ nhận thông tin (do đã kết thúc gửi), bên kia vẫn tiếp tục gửi.

Cấu trúc gói tin

Một gói tin TCP luôn bao gồm hai phần

- Header (độ dài cố định 20 byte)
- Dữ liệu

Phần header có 11 trường, 10 trường trong đó là bắt buộc. Trường thứ 11 là tùy chọn:

- Source port: Số hiệu cổng của máy tính gửi tin.
- Destination port: Số hiệu cổng của máy tính nhận tin.

- Sequence number: Nếu cờ SYN bật thì trường này là số thứ tự gói ban đầu và byte đầu tiên được gửi có số thứ tự này cộng thêm 1. Nếu không có cờ SYN thì trường này là số thứ tự của byte đầu tiên.
- Acknowledgement number: Nếu cờ ACK bật thì giá trị của trường này là số thứ tự của gói tin tiếp theo mà bên nhận cần.
- Data offset: Trường này quy định độ dài của phần header. Phần header có độ dài tối thiểu là 160 bit và tối đa là 480 bit.
- Reserved: Trường này dành cho tương lai và có giá trị là 0.
- Flags (Control bits): Bao gồm 6 cờ:
  - URG: Cờ cho trường Urgent pointer
  - ACK: Cờ cho trường Acknowledgement
  - PSH: Hàm Push
  - RST: Thiết lập lại đường truyền
  - SYN: Đồng bộ lại số thứ tự
  - FIN: Không gửi thêm số liệu
- Window: Số byte có thể nhận bắt đầu từ giá trị của ACK
- Checksum: Kiểm tra tính toàn vẹn cho header và data
- Urgent pointer: Nếu cờ URG bật thì giá trị trường này là số từ 16 bit mà sequence number cần dịch trái.
- Options: Trường này là tùy chọn (Độ dài chia hết cho 32 bit).

Trường cuối cùng không thuộc về header của TCP. Giá trị của trường này là thông tin dành cho các tầng trên (trong mô hình OSI/tầng ứng dụng trong mô hình TCP/IP). Thông tin về giao thức của tầng trên phụ thuộc vào cổng được chọn chứ không được chỉ rõ trong phần header.

+	Bít 0 - 3	4 - 9	10 - 15	16 - 31
0	Source Port			Destination Port
32	Sequence Number			
64	Acknowledgement Number			
96	Data Offset	Reserved	Flags	Window
128	Checksum			Urgent Pointer
160	Options (optional)			
160/192+	Data			

Hình 2.4. Cấu trúc của gói tin TCP (Bao gồm header và data)

2.1.3. User Datagram Protocol

User Datagram Protocol là một trong những giao thức của bộ giao thức TCP/IP. Với UDP, các các máy tính có thể gửi đi những khối dữ liệu ngắn (datagram) tới máy khác. UDP không cung cấp khả năng truyền tin tin cậy như TCP; các gói dữ liệu được chuyển bằng UDP có thể đến đích không đúng thứ tự truyền đi hoặc bị mất mà không có thông báo. Đổi lại, UDP nhanh và hiệu quả hơn TCP khi dữ liệu cần truyền đi có kích thước nhỏ và có yêu cầu khắt khe về thời gian. Do vậy UDP hiệu quả trong việc trả lời các truy vấn nhỏ với số lượng lớn yêu cầu.

Những ứng dụng phổ biến sử dụng UDP như Voice over IP, ứng dụng streaming media, DNS (Domain Name System), game trực tuyến và Trivial File Transfer Protocol (TFTP).

UDP là giao thức hướng thông điệp nhỏ nhất của tầng giao vận hiện được mô tả trong RFC 768 của IETF.

Trong bộ giao thức TCP/IP, UDP nằm ở tầng giao vận, cung cấp dịch vụ cho tầng mạng bên dưới và tầng ứng dụng phía trên.

UDP không đảm bảo cho các tầng phía trên rằng dữ liệu đã được gửi đi và phía gửi tin cũng không có trạng thái gì về dữ liệu đã gửi.

UDP không cung cấp các thông tin cho truyền tin tin cậy. Các loại thông tin dùng cho việc truyền tin cậy nếu cần phải được xây dựng ở các tầng mạng cao hơn.

+	Bits 0 - 15	16 - 31
0	Source Port	Destination Port
32	Length	Checksum
64	Data	

Hình 2.5. Cấu trúc của gói tin UDP (bao gồm header và data)

Header của gói tin UDP bao gồm:

- Source port: Trường này xác định cổng của phía gửi tin dùng để nhận thông tin phản hồi. Nếu không được dùng đến sẽ có giá trị bằng 0.
- Destination port: Trường xác định cổng của phía nhận thông tin, trường này là bắt buộc phải có.
- Length: Trường này có độ dài 16 bit, có nhiệm vụ xác định chiều dài của toàn bộ datagram bao gồm: phần header và data. Chiều dài tối thiểu là 8 byte khi gói tin chỉ có header, không có data.
- Checksum: Trường này có độ dài 16 bit, được dùng để kiểm tra lỗi của phần header và data (kiểm tra gói tin có nguyên vẹn không). Phương pháp tính checksum được định nghĩa trong RFC 768.

Do không có tính tin cậy, các ứng dụng sử dụng UDP phải chấp nhận mất mát, lỗi hoặc trùng dữ liệu. Các ứng dụng có nhu cầu truyền tin tin cậy phải thêm những kỹ thuật làm tin cậy cơ bản vào tầng ứng dụng. Thực tế thì hầu hết các ứng dụng UDP không cần thêm những kỹ thuật làm tin cậy này. Nếu một ứng dụng yêu cầu cao về tính tin cậy, có thể sử dụng giao thức TCP để thay thế.

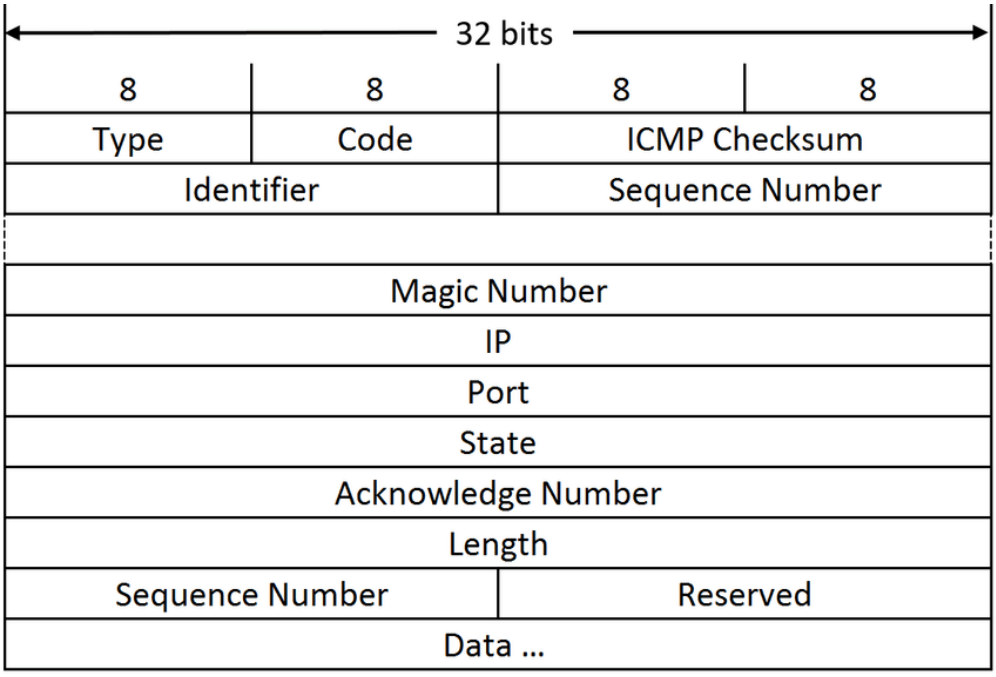
Vì UDP không có cơ chế phát hiện tắc nghẽn, các thiết bị trên mạng như router sẽ dùng hàng đợi gói (packet queueing) hoặc kỹ thuật bỏ gói làm những giải pháp để giảm tải cho UDP. Giao thức Datagram Congestion Control Protocol (DCCP) được thiết kế cho vấn đề kiểm soát tắc nghẽn này bằng cách thêm hành vi kiểm soát tắc nghẽn cho các dòng dữ liệu UDP như streaming media.

2.1.4. Internet Control Message Protocol

Internet Control Message Protocol là một giao thức được các thiết bị mạng như router dùng để gửi đi các thông báo nhằm xác định xem một địa chỉ có tồn tại hay không. ICMP cũng có khả năng chuyển tiếp các thông điệp truy vấn. ICMP khác với các giao thức tầng giao vận như TCP và UDP ở chỗ nó không thường được sử dụng để trao đổi dữ liệu giữa các máy, ICMP thường được sử dụng bởi các công cụ chẩn đoán mạng như ping và traceroute.

ICMP tuân theo các nguyên tắc:

- ICMP sử dụng IP để làm cơ sở, gói tin ICMP được đóng gói bên trong gói tin IP.
- ICMP có thể nhận biết được một số tình trạng lỗi, nhưng điều này không làm cho IP trở thành một giao thức đáng tin cậy.
- ICMP có thể phân tích sai sót trong mỗi gói tin IP, trừ các gói tin mang thông điệp ICMP.
- ICMP không trả lời các các gói tin có điểm đến là địa chỉ multicast hoặc broadcast.
- ICMP chỉ trả lời các gói tin có điểm đến là địa chỉ unicast.



Hình 2.6. Header của gói tin ICMP

2.2. Sơ lược về an ninh mạng

An ninh mạng (cyber security), an ninh máy tính (computer security), bảo mật công nghệ thông tin (IT security) là công việc bảo vệ hệ thống mạng máy tính khỏi các hành vi làm tổn hại đến phần cứng, phần mềm và dữ liệu, đảm bảo sự sẵn sàng của các dịch vụ trong hệ thống mạng [1].

An ninh mạng là bảo vệ các hệ thống mạng, máy tính, chương trình và dữ liệu khỏi những cuộc tấn công mạng.

An ninh mạng bao gồm việc kiểm soát, ngăn chặn, chống lại các tác hại có thể xảy ra thông qua qua truy cập đến hệ thống phần cứng, phần mềm và cơ sở dữ liệu.

An ninh mạng được chia thành ba phần chính: bảo mật công nghệ thông tin, an ninh mạng và an ninh máy tính.

- Bảo mật công nghệ thông tin: Bảo vệ dữ liệu cả khi chúng được lưu trữ và khi di chuyển trên các mạng lưới thông tin. Trong khi an ninh mạng chỉ bảo vệ dữ liệu số, bảo mật công nghệ thông tin có nhiệm vụ bảo vệ cả dữ liệu số lẫn dữ liệu vật lý.
- An ninh mạng: Thực hiện nhiệm vụ đảm bảo an toàn cho dữ liệu số trên các mạng lưới, máy tính khỏi sự truy cập, tấn công và phá hủy bất hợp pháp. Là một tập hợp con của bảo mật công nghệ thông tin.
- An ninh máy tính: Là một tập con của an ninh mạng. Loại bảo mật này sử dụng cả phần cứng và phần mềm để bảo vệ tất cả dữ liệu được gửi đến hệ thống mạng lưới thông tin.

Lĩnh vực an ninh mạng đang trở nên ngày càng quan trọng do sự phát triển của các hệ thống máy tính và Internet trên khắp thế giới, cũng như sự phát triển của hệ thống mạng không dây và sự phát triển của các thiết bị "thông minh", bao gồm điện thoại thông minh, TV thông minh, và các thiết bị khác như camera, cảm biến kết nối vào hệ thống Internet of Things.

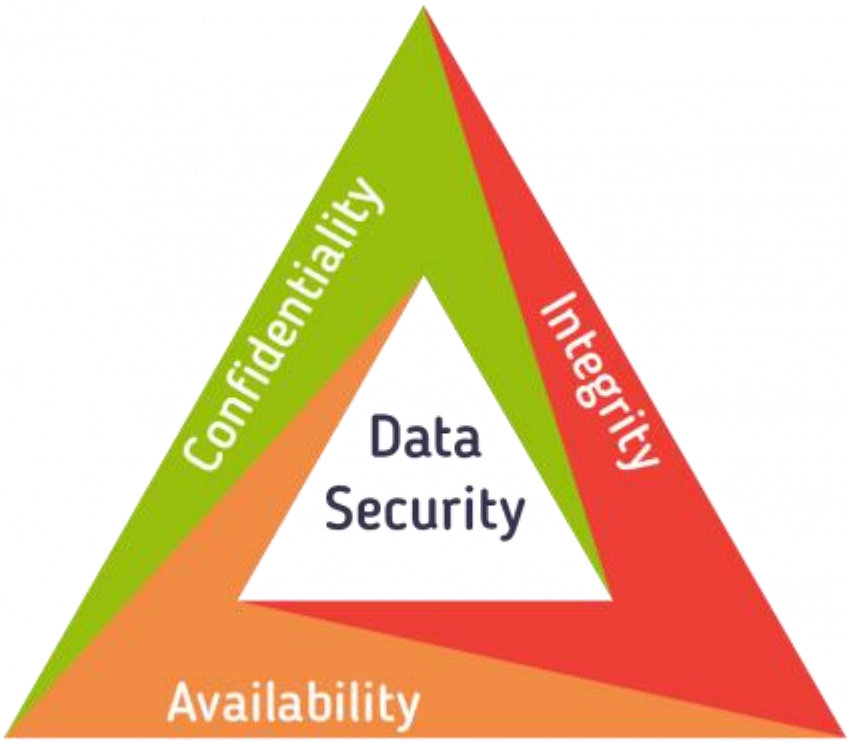
2.2.1. Các mục tiêu của an ninh mạng

Mục tiêu chính của an ninh mạng là bảo vệ thông tin khỏi bị xâm phạm, tấn công, đảm bảo cho thông tin nguyên vẹn và luôn sẵn sàng. Tương ứng với ba mục tiêu sau:

- Mục tiêu về tính bảo mật của dữ liệu.
- Mục tiêu về tính toàn vẹn của dữ liệu.
- Mục tiêu về tính sẵn có của dữ liệu.

Những mục tiêu này tạo thành bộ ba "Bảo mật – Toàn vẹn – Sẵn có" (Confidentiality – Integrity – Availability), đây là cơ sở cốt lõi của tất cả các chương trình bảo mật thông tin.

Các yếu tố của tam giác CIA được coi là những yếu tố quan trọng nhất của bảo mật thông tin.



Hình 2.7 Mô hình CIA của lĩnh vực an toàn thông tin

Tính bảo mật (Confidentiality)

Bảo mật là quyền riêng tư và việc tránh tiết lộ thông tin trái phép. Bảo mật cung cấp quyền truy cập cho những người được phép. Yếu tố này ngăn chặn thông tin tiếp cận sai người trong khi đảm bảo rằng người dùng có thể thu thập được thông tin cần thiết.

Các công cụ chính phục vụ cho tiêu chí "bảo mật":

- Mã hóa (Encryption): Mã hóa là một phương pháp chuyển đổi thông tin khiến dữ liệu trở nên không thể đọc được đối với người dùng trái phép bằng cách sử dụng thuật toán mã hóa.
- Kiểm soát quyền truy cập (Access Control): Đây là công cụ xác định các quy tắc và chính sách để giới hạn quyền truy cập vào hệ thống hoặc các tài nguyên, dữ liệu. Kiểm soát quyền truy cập bao gồm quá trình cấp cho người dùng quyền truy cập và một số đặc quyền nhất định đối với hệ thống, tài nguyên hoặc thông tin.
- Xác thực (Authentication): Xác thực là quá trình xác nhận danh tính hoặc vai trò của người dùng.

- Ủy quyền (Authorization): Đây là cơ chế bảo mật được sử dụng để xác định danh tính của một người hoặc hệ thống được phép truy cập vào dữ liệu, dựa trên chính sách kiểm soát quyền truy cập, bao gồm các chương trình máy tính, tệp tin, dịch vụ, dữ liệu và tính năng ứng dụng.
- Bảo mật vật lý (Physical Security): Đây là các biện pháp được thiết kế để ngăn chặn sự truy cập trái phép vào các tài sản công nghệ thông tin như cơ sở vật chất, thiết bị, nhân sự, tài nguyên và các loại tài sản khác nhằm tránh bị hư hại.

Tính toàn vẹn (Integrity)

Tính toàn vẹn đề cập đến các phương pháp nhằm đảm bảo nguồn dữ liệu là thật, chính xác và được bảo vệ khỏi sự sửa đổi trái phép của người dùng.

Các công cụ chính phục vụ cho tiêu chí "toàn vẹn":

- Sao lưu (Backups): Sao lưu là lưu trữ dữ liệu định kỳ. Đây là một quá trình tạo lập các bản sao của dữ liệu để sử dụng trong trường hợp khi dữ liệu gốc bị mất hoặc bị hủy.
- Tổng kiểm tra (Checksums): Checksums là một giá trị số được sử dụng để xác minh tính toàn vẹn của dữ liệu được truyền đi. Chúng thường được sử dụng để so sánh hai bộ dữ liệu, nhằm đảm bảo rằng chúng giống hệt nhau. Hàm tổng kiểm tra phụ thuộc vào toàn bộ nội dung của tệp, nó được thiết kế theo cách mà ngay cả một thay đổi nhỏ đối với tệp đầu vào (chẳng hạn như lệch một bit) có thể dẫn đến giá trị đầu ra khác nhau.
- Mã chỉnh dữ liệu (Data Correcting Codes): Đây là một phương pháp để lưu trữ dữ liệu theo cách mà những thay đổi nhỏ nhất cũng có thể dễ dàng được phát hiện và tự động điều chỉnh.

Tính sẵn có (Availability)

Mọi hệ thống thông tin đều đảm bảo được thông tin phải luôn sẵn sàng khi cần thiết. Hệ thống có tính sẵn sàng cao hướng đến sự sẵn có, khả dụng ở mọi thời điểm, tránh được rủi ro, đảm bảo thông tin có thể được truy cập và sửa đổi kịp thời bởi những người được ủy quyền.

Các công cụ chính phục vụ cho tiêu chí "sẵn có":

- Bảo vệ vật lý (Physical Protections): Có nghĩa là giữ thông tin có sẵn ngay cả trong trường hợp phải đối mặt với thách thức về vật chất. Đảm bảo các thông tin nhạy cảm và công nghệ thông tin quan trọng được lưu trữ trong các khu vực an toàn.

- Tính toán dự phòng (Computational Redundancies): Được áp dụng nhằm bảo vệ máy tính và các thiết bị được lưu trữ, đóng vai trò dự phòng trong trường hợp xảy ra hỏng hóc.

2.2.2. Tấn công mạng và mục tiêu

Tấn công mạng là tất cả các hình thức xâm nhập trái phép vào một hệ thống máy tính, website, cơ sở dữ liệu, hạ tầng mạng, thiết bị của một cá nhân hoặc tổ chức thông qua mạng Internet với những mục đích bất hợp pháp.

Các mục tiêu của tấn công mạng:

- Tấn công mạng không mục tiêu: Trong các cuộc tấn công không mục tiêu, đối tượng bị đến là số lượng thiết bị, dịch vụ hoặc người dùng bị ảnh hưởng càng tốt. Để thực hiện các cuộc tấn công này, cần sử dụng những loại kỹ thuật mà có thể tận dụng được sự công khai, rộng rãi của Internet.
- Tấn công mạng có mục tiêu: Đối với cuộc tấn công có mục tiêu, một cá nhân hoặc tổ chức sẽ dễ dàng rơi vào tình trạng bị tấn công. Tấn công có mục tiêu thường gây ra tổn hại nặng nề hơn so với một cuộc tấn công không nhắm mục tiêu, bởi vì nó được thiết kế riêng để tấn công vào các hệ thống, quy trình.

2.2.3. Lỗ hổng bảo mật và các loại tấn công phổ biến

Lỗ hổng bảo mật chỉ những điểm yếu của hệ thống mạng máy tính. Ngày nay, cơ sở dữ liệu Common Vulnerabilities and Exposures (CVE) chứa thông tin về phần lớn các lỗ hổng bảo mật đã được phát hiện. Lỗ hổng mà đã bị lợi dụng để thực hiện hoạt động tấn công ít nhất một lần hoặc đã bị khai thác được goi là lỗ hổng bị khai thác.

Các kỹ thuật tấn công mạng phổ biến thường được xếp vào một trong các mục dưới đây:

Kỹ thuật "Tấn công bằng phần mềm độc hại (Malware Attack)"

Malware attack là một trong những hình thức tấn công mạng phổ biến nhất. Malware được tạo ra với mục đích làm hư hỏng máy tính của người dùng. Thông thường, malware được đính kèm trong thư điện tử hoặc các tệp tin được ngụy trang.

Có vô số loại phần mềm độc hại khác nhau, điển hình như:

- Virus: Là những đoạn mã chương trình tự sao chép, đính kèm vào các tệp tin sạch, được thiết kế để xâm nhập, lây lan khắp hệ thống máy tính nhằm thực thi một số tác vụ nào đó.
- Trojan Horse: Khác với virus, phần mềm này không có chức năng tự sao chép. Trojan Horse sẽ được ngụy trang thành các phần mềm hợp pháp, người

dùng sẽ bị lừa cài đặt Trojan Horse vào máy tính của họ, sau đó chúng có thể gây thiệt hại đến máy tính hoặc thu thập các dữ liệu cá nhân.

- Phần mềm gián điệp (Spyware): Đây là loại virus có khả năng thâm nhập trực tiếp vào hệ điều hành, bí mật lưu lại những gì người dùng làm.
- Phần mềm tống tiền (Ransomware): Ngăn cản người dùng truy cập vào dữ liệu quan trọng để đòi tiền chuộc.
- Phần mềm quảng cáo (Adware): Có thể được sử dụng để hiện quảng cáo, hoặc phát tán, cài đặt các phần mềm độc hại khác.
- Botnets: Mạng lưới các máy tính bị nhiễm phần mềm độc hại được sử dụng để thực hiện các hành động mà không có sự cho phép của người dùng.

Kỹ thuật "Tấn công giả mạo (Phishing Attack)"

Phishing là hình thức giả mạo thành một đơn vị/cá nhân uy tín để chiếm lấy lòng tin của người dùng, với mục tiêu đánh cắp dữ liệu cá nhân nhạy hoặc cài đặt các phần mềm độc hại vào máy tính nạn nhân. Phishing attack thường được thực hiện bằng cách sử dụng email hoặc tin nhắn.

Kỹ thuật "Tấn công trung gian (Man-in-the-middle Attack)"

Tấn công trung gian (MitM), xảy ra khi kẻ tấn công xâm nhập vào một giao dịch đang diễn ra giữa 2 đối tượng, một khi đã xen vào thành công, chúng có thể đánh cắp và chỉnh sửa dữ liệu. Một số biến thể của tấn công trung gian có thể kể đến như đánh cắp mật khẩu, chuyển tiếp các thông tin không xác thực.

Kỹ thuật "Tấn công từ chối dịch vụ (Denial of Service)"

Tấn công từ chối dịch vụ (DoS) có mục đích làm cho tài nguyên mạng, dịch vụ trở nên không sẵn sàng để phục vụ cho người dùng. Có thể từ chối dịch vụ cho từng đối tượng, ví dụ như nhập sai mật khẩu nhiều lần liên tục để khiến tài khoản nạn nhân bị khóa. Hoặc hoặc khiến cho mạng và máy tính quá tải để chặn tất cả người dùng cùng một lúc. Tường lửa có thể chặn cuộc tấn công từ một IP duy nhất, nhưng với hình thức tấn công từ chối dịch vụ phân tán (DDoS) thì không.

Kỹ thuật "Tấn công cơ sở dữ liệu (SQL Injection Attack)"

Kẻ tấn công chèn một đoạn mã độc hại vào server sử dụng ngôn ngữ truy vấn có cấu trúc (SQL), mục đích là để khiến server trả về những thông tin quan trọng mà lẽ ra không được tiết lộ. Các cuộc tấn công SQL Injection xuất phát từ lỗ hổng của website với mức bảo mật yếu.

Kỹ thuật "Tấn công "cửa hậu" (Backdoor Attack)"

Backdoor là phương pháp bí mật vượt qua thủ tục chứng thực người dùng thông thường, truy cập vào thiết bị mà không bị phát hiện bởi việc giám sát thông thường.

Backdoor có thể được thêm vào bởi người có thẩm quyền để cho phép một số truy cập hợp pháp, hoặc bởi những kẻ tấn công vì những mục đích phi pháp.

Kỹ thuật "Khai thác lỗ hổng (Zero-day Exploits)"

Lỗ hổng Zero-day là thuật ngữ để chỉ những lỗ hổng phần mềm hoặc phần cứng chưa được biết đến và chưa được khắc phục. Kẻ tấn công có thể tận dụng lỗ hổng này để tấn công xâm nhập vào hệ thống máy tính.

2.3. Intrusion Detection System và Intrusion Prevention System

2.3.1. Intrusion Detection System

Intrusion Detection System là hệ thống giám sát lưu lượng mạng nhằm phát hiện các bất thường, các hoạt động xâm nhập trái phép vào hệ thống mạng. IDS có thể nhận biết được những cuộc tấn công đến từ cả bên trong và từ bên ngoài mạng [2].

IDS phát hiện tấn công dựa trên các dấu hiệu khác thường (tương tự như các phần mềm diệt Virus), ngoài ra IDS còn phân tích lưu thông mạng hiện tại và so sánh nó với thông số đo đạt chuẩn của hệ thống để tìm ra các dấu hiệu khác thường.

IDS có các tính năng chính:

- Giám sát lưu lượng mạng và phát hiện các hoạt động khả nghi trong mạng.
- Cảnh báo về tình trạng hệ thống mạng cho quản trị viên.
- Kết hợp với các hệ thống bảo mật truyền thống như tường lửa, chương trình diệt virus, … tạo thành một hệ thống bảo mật hoàn chỉnh.

Phân loại IDS:

- Network-based Intrusion Detection System: Theo dõi hoạt động bất thường trên trong toàn mạng. NIDS thường được đặt sau router để có thể theo dõi và bảo vệ được toàn bộ hệ thống mạng.
- Host-based Intrusion Detection System: Theo dõi các hoạt động bất thường trên các host riêng biệt. HIDS thường được đặt trên các máy cần theo dõi.

Ngày nay, IDS là một trong những thành phần quan trọng nhất trong các giải pháp bảo vệ hệ thống. Khi triển khai IDS có thể giúp quản trị viên:

- Phát hiện, theo dõi các hoạt động bất thường xảy ra trong hệ thống.
- Xác định máy nào đang tác động đến hệ thống và máy đó làm như thế nào.
- Xác định vị trí các hoạt động xâm nhập xảy ra trong mạng.

Ưu điểm của IDS:

- Cung cấp một cái nhìn toàn diện về toàn bộ lưu lượng trong hệ thống mạng.
- Giúp phát hiện và kiểm tra các sự cố xảy ra trong hệ thống mạng.
- Sử dụng để thu thập thông tin cho việc ứng cứu sự cố và điều tra sau sự cố.

Hạn chế của IDS:

- Có thể gây ra tình trạng báo động sai nếu được cấu hình không hợp lý.
- Gần như không thể phân tích lưu lượng đã bị mã hóa.
- Chi phí triển khai và vận hành IDS lớn.

Hệ thống tập luật của IDS

Tập luật là một trong những thành phần quan trọng nhất của IDS. Đây là nơi sẽ định nghĩa dấu hiệu để IDS so sánh, đối chiếu với dữ liệu ở đầu vào. Thông thường, tập luật bao gồm rất nhiều luật, mỗi luật sẽ gồm hai thành phần cơ bản: Rule Header và Rule Options (có thể không có).

Rule header bao gồm các thông tin sau:

- Rule Action: Hành động sẽ được thực thi khi dữ liệu đầu vào "khớp" với luật (alert, log, pass, active, dynamic, drop…).
- Protocol: Giao của luật (TCP, UDP, ICMP, IP…)
- IP address: Địa chỉ IP của luật.
- Port number: Địa chỉ cổng của luật.
- Direction: Hướng đi của dữ liệu.

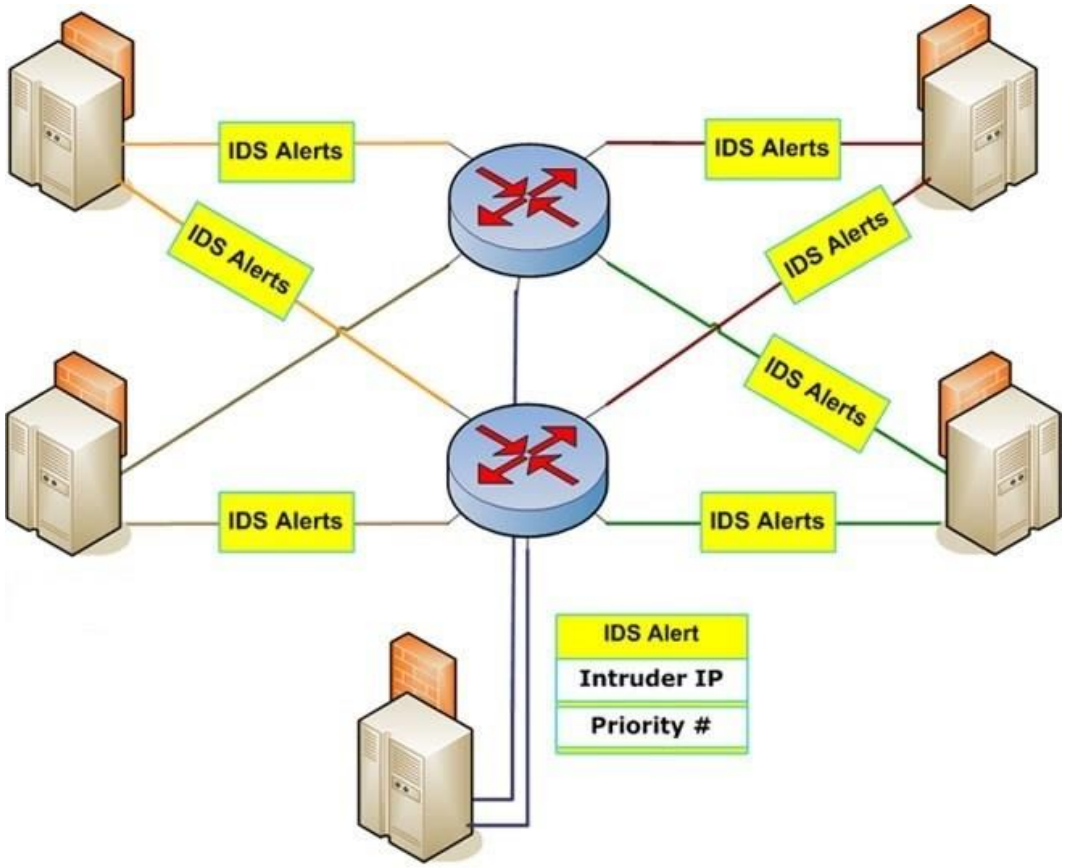
Rule options bao gồm 4 danh mục:

- General: Thông tin chung về luật (msg, reference, rev, classtype…).
- Payload: Tìm kiếm nội dung payload của gói tin (content, offset, depth, distance, within…).
- Non-payload: Tìm kiếm nội dung non-payload của gói tin (ttl, ack, tos, id, dsize…).
- Post-detection: Phương pháp thực thi kế tiếp (logto, session, tag…).

Thiết kế IDS trong mô hình mạng doanh nghiệp

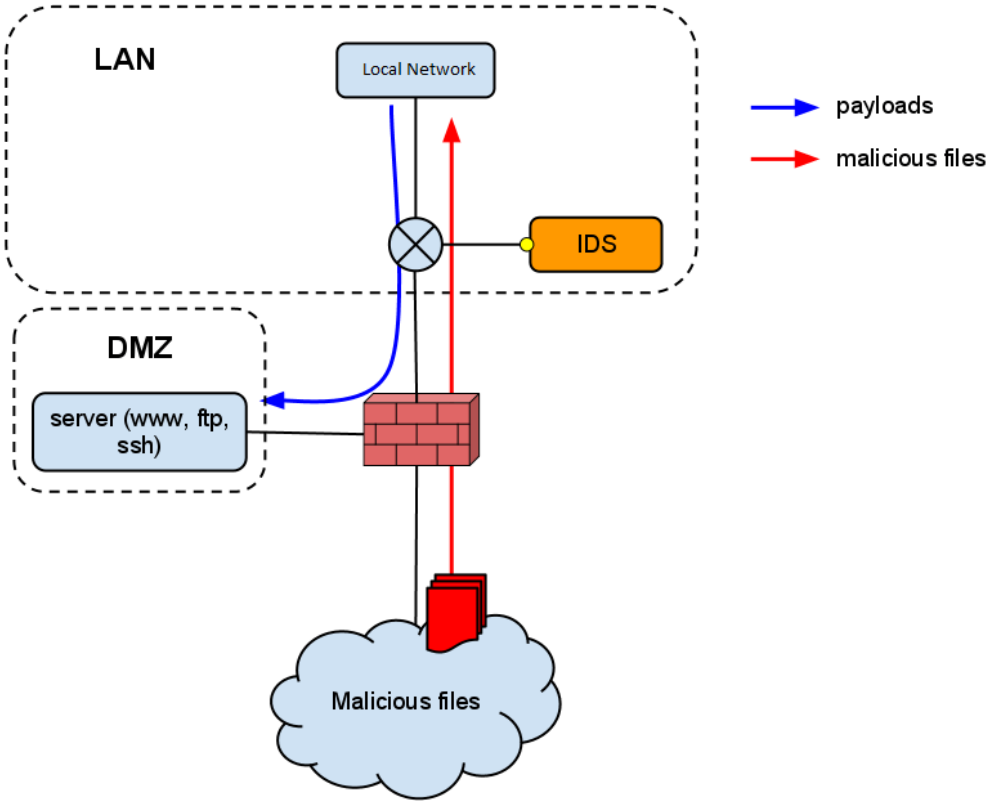
Tùy vào mục đích cũng như cấu trúc mạng, có thể đặt IDS tại các vị trí khác nhau để tận dụng tối đa khả năng của hệ thống này.

Các vị trí thường đặt IDS trong mô hình mạng:



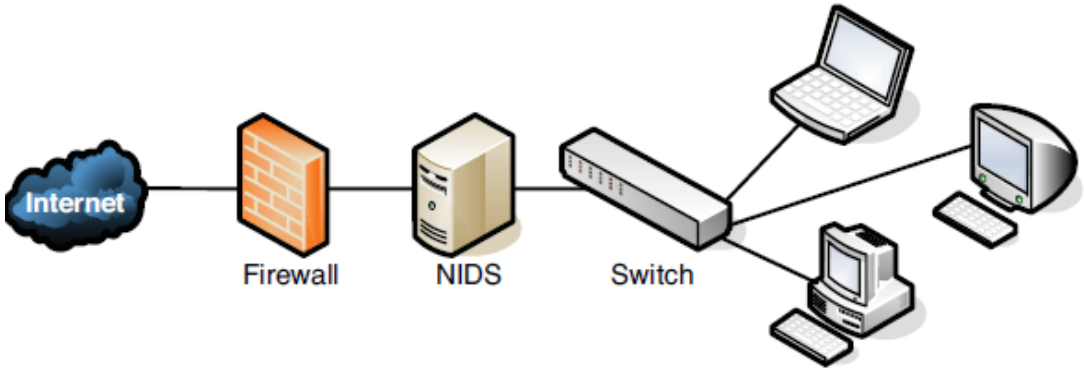
Hình 2.8. IDS được đặt giữa router và firewall

Khi IDS được đặt giữa router và firewall như trong Hình 2.8, IDS sẽ theo dõi tất cả các lưu lượng trên cả 2 chiều. Khi triển khai theo cấu trúc này thì IDS phải chịu áp lực rất lớn về lưu lượng, nhưng lại có khả năng giám sát toàn bộ lưu lượng của hệ thống mạng. Vì vậy, trong trường hợp này nên lựa chọn các thiết bị IDS có khả năng chịu tải cao để nâng cao hiệu năng.



Hình 2.9. IDS được đặt trong miền DMZ

Khi IDS được đặt trong miền DMZ như Hình 2.9, IDS sẽ theo dõi tất cả lưu lượng vào/ra trong miền DMZ.



Hình 2.10. IDS được đặt sau firewall

Khi IDS được đặt sau firewall như Hình 2.10, IDS sẽ theo dõi tất cả lưu lượng trao đổi phía sau firewall như:

- Tất cả dữ liệu trao đổi trong LAN.
- Tất cả dữ liệu từ LAN vào/ra DMZ và ngược lại.

2.3.2. Intrusion Prevention System

Intrusion Prevention Systems là hệ thống theo dõi, phát hiện và xử lý các hoạt động xâm nhập mạng. Chức năng chính của IPS là xác định các hành động nguy hại và ngăn chặn các hành động này, đưa ra các báo cáo chi tiết. IPS có thể xem là trường hợp mở rộng của IDS, hai hệ thống này có cách thức hoạt động và các đặc điểm tương tự nhau. Điểm khác nhau duy nhất là hệ thống IPS ngoài khả năng theo dõi, giám sát thì còn có khả năng ngăn chặn các hoạt động gây nguy hại đối với hệ thống. IDS và IPS có thể sử dụng cùng tập luật với nhau [3].

Phân loại IPS:

- Network-based Intrusion Prevention: Thường được đặt trước hoặc sau firewall. Khi triển khai IPS trước firewall có thể bảo vệ được toàn bộ hệ thống phía sau kể cả firewall, vùng DMZ. Có thể giảm thiểu nguy cơ bị tấn công từ chối dịch vụ. Khi triển khai IPS sau firewall có thể xử lý được một số kiểu tấn công thông qua khai thác điểm yếu trên các thiết bị di động sử dụng VPN để kết nối vào bên trong.
- Host-based Intrusion Prevention: Thường được triển khai với mục đích phát hiện và ngăn chặn kịp thời các hoạt động thâm nhập trên các host. Ngoài khả năng phát hiện ngăn ngừa các hoạt động thâm nhập, HIPS còn có khả năng phát hiện sự thay đổi các tập tin cấu hình.

Mỗi thành phần trong hệ thống mạng đều có chức năng, điểm mạnh, điểm yếu khác nhau. Khi được sử dụng đúng mục đích sẽ đem lại hiệu quả cao. IPS là một trong những thành phần quan trọng trong các giải pháp bảo vệ hệ thống. Khi triển khai có thể giúp quản trị viên:

- Theo dõi và xử lý các hoạt động bất thường với hệ thống.
- Xác định máy nào đang tác động đến hệ thống và máy đó làm như thế nào, xác định vị trí các hoạt động xâm nhập xảy ra trong hệ thống mạng.
- Kết hợp với firewall để ngăn chặn các hoạt động thâm nhập hệ thống.

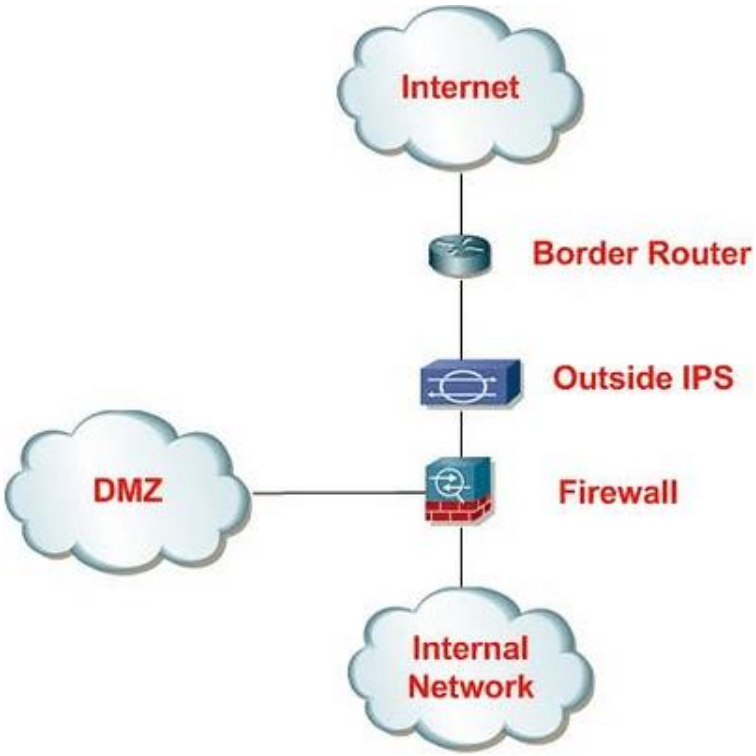
Ưu điểm của IPS:

- Cung cấp giải pháp bảo vệ toàn diện hơn cho hệ thống mạng.
- Kịp thời ngăn chặn các cuộc tấn công nhắm vào hệ thống mạng

Hạn chế của IPS:

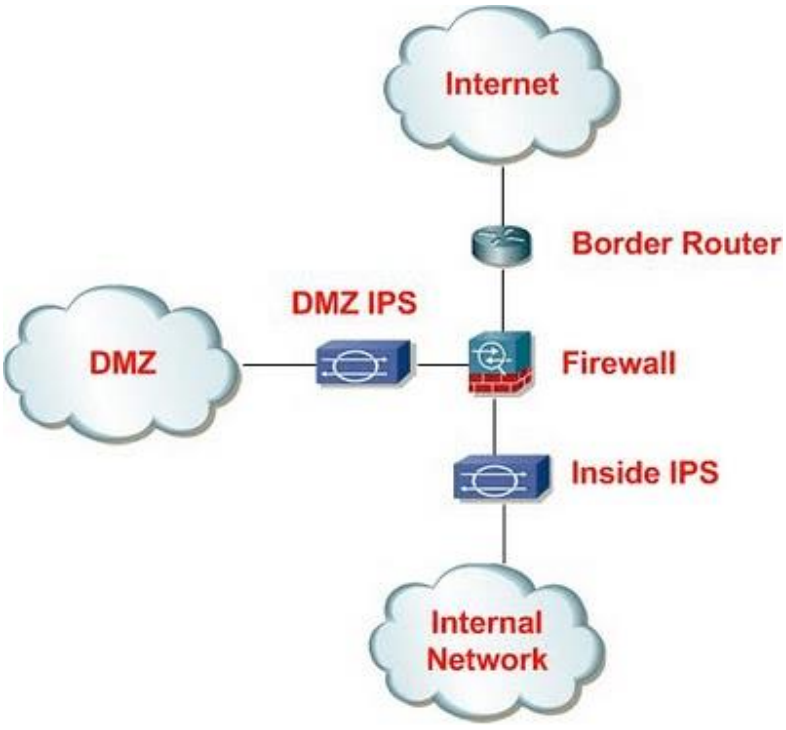
- Có khả năng gây ra tình trạng phát hiện nhầm, làm cho các truy cập hợp lệ không thể tới hệ thống.

Các vị trí đặt IPS trong mạng:



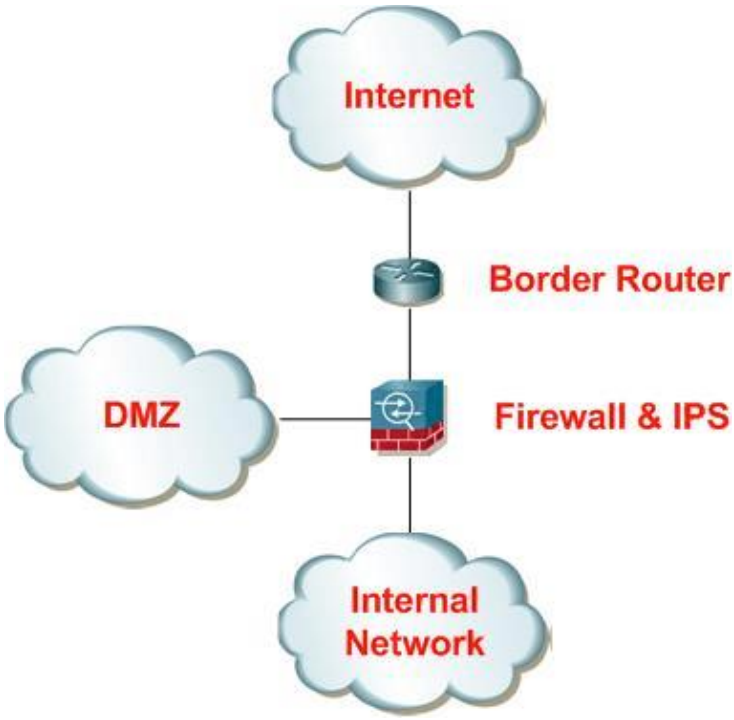
Hình 2.11. IPS được đặt trước firewall

Khi IPS được đặt trước firewall và sau router như hình 2.11, IPS sẽ theo dõi và bảo vệ toàn bộ lưu lượng mạng theo hai chiều. Điều này yêu cầu IPS phải có khả năng chịu tải cao vì áp lực sẽ rất lớn.



Hình 2.12. IPS được đặt giữa firewall và miền DMZ

Khi IPS được đặt giữa firewall và miền DMZ như hình 2.12, IPS sẽ theo dõi và bảo vệ toàn bộ lưu lượng vào/ra miền DMZ.



Hình 2.13. IPS là một module trong giải pháp UTM

Khi IPS là một module trong phải pháp UTM như hình 2.12, IPS sẽ được tích hợp vào firewall để cung cấp một giải pháp "all-in-one".

2.4. Iptables và Netfilter

Iptables là phần mềm tường lửa cơ bản được sử dụng trong Linux. Iptables hoạt động bằng cách tương tác với các hooks lọc gói trong Linux kernel networking stack. Những kernel hooks này được gọi là netfilter framework [5].

Mỗi gói tin vào hệ thống mạng (đến hoặc đi) sẽ kích hoạt các hooks này, cho phép các chương trình đăng ký với các hook này tương tác với gói tin.

2.4.1. Netfilter hooks

Có năm hooks netfilter mà các chương trình có thể đăng ký. Khi các gói tin đi đến, chúng sẽ kích hoạt các module đã đăng ký với các hooks này. Các hooks mà gói tin sẽ kích hoạt tùy thuộc vào việc gói đến hay đi, đích đến của gói và liệu gói bị drop hay bị từ chối trước đó.

Các hooks sau biểu thị các điểm được xác định rõ trong networking stack:

- NF_IP_PRE_ROUTING: Hook này sẽ được kích hoạt bởi bất kỳ lưu lượng truy cập đến nào ngay sau khi vào network stack. Hook này được xử lý trước khi đưa ra bất kỳ quyết định định tuyến nào.

- NF_IP_LOCAL_IN: Hook này được kích hoạt sau khi gói tin đến đã được định tuyến nếu đích đến của nó là hệ thống cục bộ.
- NF_IP_FORWARD: Hook này được kích hoạt sau khi gói đến đã được định tuyến nếu nó được chuyển tới máy khác.
- NF_IP_LOCAL_OUT: Hook này được kích hoạt bởi bất kỳ lưu lượng truy cập nào được tạo ra cục bộ ngay khi mà nó đến network stack.
- NF_IP_POST_ROUTING: Hook này được kích hoạt bởi bất kỳ lưu lượng đi hoặc chuyển tiếp nào sau khi đã việc định tuyến đã diễn ra và ngay trước khi lưu lượng được chuyển đi.

Modules muốn đăng ký tại các hook này phải cung cấp số ưu tiên để giúp xác định thứ tự chúng sẽ được gọi khi hook được kích hoạt. Điều này cho phép nhiều modules (hoặc nhiều phiên bản của cùng một module) được kết nối với từng hook theo thứ tự xác định. Mỗi module sẽ được gọi lần lượt và sẽ trả lại quyết định cho netfilter framework sau khi xử lý cho biết những gì sẽ được thực hiện với gói tin.

2.4.2. Các bảng và Chain của Iptables

Tường lửa Iptables sử dụng các bảng để tổ chức các rules của nó. Các bảng này phân loại các rules theo loại quyết định mà chúng được sử dụng để đưa ra.

Trong mỗi bảng Iptables, các rules được tổ chức thêm trong các chain riêng biệt. Trong khi các bảng được xác định bởi mục đích chung của các rules mà chúng nắm giữ, các chain đại diện cho các hooks sẽ kích hoạt chúng. Chains cơ bản xác định khi nào các rules sẽ được thực thi.

Bảng 2.1. Các chain và rule tương ứng

Chain	Rule
PREROUTING	Rule trong chain này được thực thi ngay khi gói tin vừa vào đến Network Interface. Chain này tồn tại ở các table: nat, mangle và raw.
INPUT	Rule trong chain này được thực thi ngay trước khi gói tin gặp tiến trình. Chain này chỉ tồn tại ở table mangle và nat.
OUTPUT	Rule trong chain này được thực thi ngay sau khi gói tin được tiến trình tạo ra. Chain này tồn tại ở các table: raw, mangle, nat và filter.
FORWARD	Rule này thực thi cho các gói tin được định tuyến qua host hiện tại. Chain này chỉ tồn tại ở table mangle và filter.

POSTROUTING	Rule này thực thi ngay khi gói tin rời Network Interface. Chain này chỉ tồn tại ở table mangle và nat.

Chain cho phép quản trị viên kiểm soát vị trí trong đường đi của gói tin mà ở đó, rule sẽ được thực thi. Vì mỗi bảng có nhiều chain, ảnh hưởng của một bảng có thể được tác động tại nhiều điểm trong quá trình xử lý. Vì các loại quyết định nhất định chỉ có ý nghĩa tại một số điểm nhất định trong network stack, các bảng sẽ không có chain được đăng ký với mỗi hook kernel.

Chỉ có năm hook kernel của netfilter, do đó, chains từ nhiều bảng được đăng ký tại mỗi hook.

2.4.3. Các loại bảng của Iptables

Bảng 2.2. Các loại bảng trong Iptables

Tên bảng	Miêu tả
Filter Table	Bảng Filter là một trong những bảng được sử dụng rộng rãi nhất trong Iptables. Bảng Filter được sử dụng để đưa ra quyết định về việc có nên để gói tin tiếp tục đến đích dự định hay từ chối yêu cầu của nó hay không. Theo cách nói tường lửa, đây được gọi là gói "lọc".
NAT Table	Bảng NAT được sử dụng để thực hiện các rules dịch địa chỉ mạng. Khi các gói vào network stack, các rule trong bảng này sẽ xác định xem và cách sửa đổi địa chỉ nguồn hoặc đích của gói để tác động đến cách gói và bất kỳ lưu lượng phản hồi nào được định tuyến. Bảng có vai trò định tuyến các gói đến các mạng khi không thể truy cập trực tiếp.
Mangle Table	Bảng Mangle được sử dụng để thay đổi header của các gói tin IP theo nhiều cách khác nhau. Chẳng hạn, có thể điều chỉnh giá trị TTL của một gói tin.
Raw Table	Tường lửa Iptables có trạng thái, nghĩa là các gói được đánh giá liên quan đến mối quan hệ của chúng với các gói trước đó. Các tính năng theo dõi kết nối được xây dựng trên đỉnh của bộ lọc mạng cho phép Iptables xem các gói như một phần của kết nối hoặc phiên liên tục thay vì như một luồng các gói rời rạc, không liên quan. Theo dõi kết nối một cách logic thường được áp dụng ngay sau khi gói truy

	cập vào network interface. Mục đích duy nhất của bảng raw là cung cấp một cơ chế đánh dấu các gói để từ chối theo dõi kết nối.
Security Table	Bảng Security được sử dụng để đặt các dấu hiệu bối cảnh bảo mật của Selinux bên trong trên các gói, điều này sẽ ảnh hưởng đến cách thức Selinux hoặc các hệ thống khác có thể diễn giải bối cảnh bảo mật của Selinux xử lý các gói. Các dấu này có thể được áp dụng cho mỗi gói hoặc mỗi kết nối.

2.4.4. Chain nào được thực hiện trong mỗi bảng?

Bảng 2.3. Các chain được thực hiện trong bảng

Tables/Chains	PRE-ROUTING	INPUT	FORWARD	OUTPUT	POST-ROUTING
(routing decision)				✓	
raw	✓			✓	
(connection tracking enabled)	✓			✓	
mangle	✓	✓	✓	✓	✓
nat (DNAT)	✓			✓	
(routing decision)	✓			✓	
filter		✓	✓	✓	
security		✓	✓	✓	
nat (SNAT)		✓			✓

Khi một gói kích hoạt netfilter hook, các chain liên kết sẽ được xử lý khi chúng được liệt kê trong bảng ở trên từ trên xuống dưới. Các hook (cột) mà một gói sẽ kích hoạt phụ thuộc vào việc nó là gói đến hay đi, các quyết định định tuyến được đưa ra và liệu gói tin có vượt qua các tiêu chí lọc hay không.

Một số sự kiện có thể sẽ khiến chains của bảng bị bỏ qua trong quá trình xử lý. Ví dụ, chỉ gói tin đầu tiên của kết nối được đánh giá theo các quy tắc NAT. Bất kỳ quyết định NAT nào được thực hiện cho gói tin đầu tiên sẽ được áp dụng cho tất cả các gói tin

tiếp theo trong cùng kết nối mà không cần đánh giá bổ sung. Phản hồi cho các kết nối NAT sẽ tự động áp dụng các quy tắc NAT ngược để định tuyến chính xác.

2.4.5. Thứ tự của các chain trong Iptables

Giả sử rằng máy tính biết cách định tuyến một gói tin và các luật tường lửa cho phép gói tin đi qua, các luồng sau đây biểu thị các đường dẫn sẽ đi qua trong các tình huống khác nhau:

- Gói tin đến với đích là cục bộ: PREROUTING → INPUT
- Gói tin đến với đích là host khác:
  PREROUTING → FORWARD → POSTROUTING
- Gói tin được tạo ra cục bộ: OUTPUT → POSTROUTING

Nếu kết hợp các thông tin trên với thứ tự được trình bày trong bảng trước, có thể thấy rằng một gói đến được định sẵn cho hệ thống cục bộ trước tiên sẽ được đánh giá theo các chuỗi PREROUTING của các bảng raw, mangle và nat. Sau đó, nó sẽ đi qua các chuỗi INPUT của các bảng mangle, filter, security và nat trước khi cuối cùng được chuyển đến local socket.

2.4.6. Luật của Iptables

Iptables rule bao gồm một hoặc nhiều tiêu chuẩn để xác định packets nào sẽ phải chịu ảnh hưởng và target để xác định hành động nào sẽ được thực thi với packet ấy.

Cả hai yếu tố của rules đó là match và target đều là tùy chọn. Như vậy, cấu trúc của Iptables như sau: Iptables → Tables → Chains → Rules

Để một rule trong Iptables được xem là matched thì gói tin đi qua phải đáp ứng các tiêu chí của rule đó để hành động tiếp theo hoặc target được thực thi.

Hệ thống matching của Iptables rất linh hoạt và có thể được mở rộng đáng kể với các tiện ích mở rộng(extension) của Iptables có sẵn trên hệ thống. Rule có thể xây dựng các tiêu chí để match bao gồm loại protocol, dest hoặc source address, dest hoặc source port, dest hoặc source network, input hoặc output interface, header, các trạng thái state của kết nối. Chúng có thể được kết hợp cùng nhau để tạo ra các bộ quy tắc để phân biệt giữa các gói tin khác nhau.

2.4.7. Target của Iptables

Target là một hành động sẽ được trigger ngay khi các tiêu chí của rule khớp (matched) hoàn toàn. Target trong Iptables được chia ra làm hai nhóm sau:

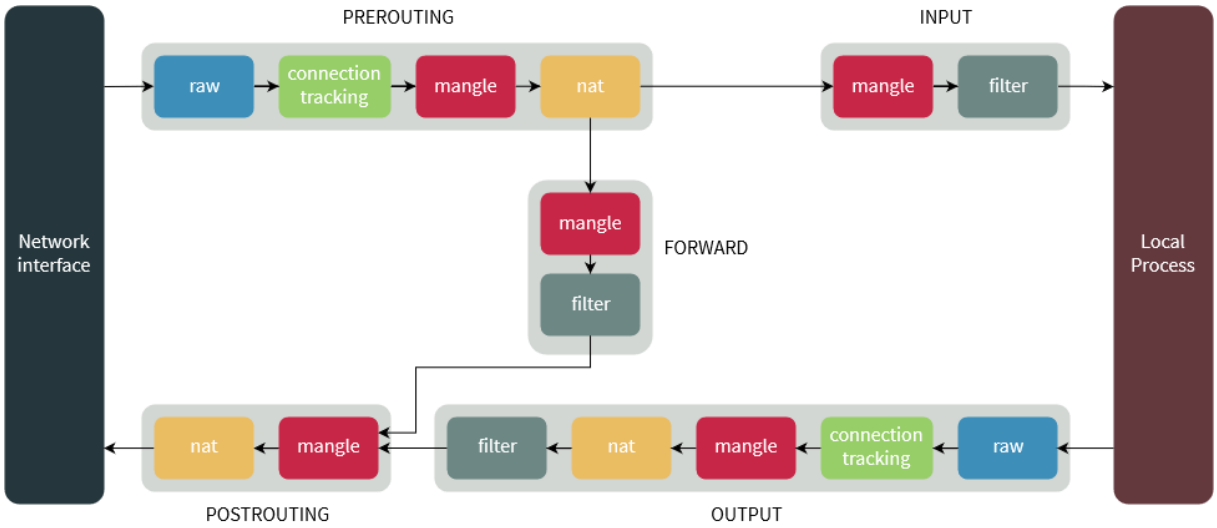
- Terminating targets: Chấm dứt các mục tiêu thực hiện một hành động chấm dứt đánh giá trong chuỗi và trả lại quyền kiểm soát cho netfilter hook. Tùy thuộc vào giá trị trả về được, hook có thể drop gói tin hoặc cho gói tin tiếp

tục đi đến giai đoạn xử lý tiếp theo. Tùy thuộc vào rule thiết lập, nó có thể DROP, ACCEPT hoặc REJECT gói tin.

- Non-terminating targets: Không chấm dứt thực hiện hành động và tiếp tục đánh giá trong chain. Mặc dù mỗi chain cuối cùng phải trả lại quyết định chấm dứt cuối cùng, bất kỳ số lượng mục tiêu không kết thúc nào cũng có thể được thực hiện trước. Là loại target mà nó thực thi hành động và vẫn tiếp tục việc kiểm tra gói tin dựa theo các rule khác. Ví dụ target LOG, nó ghi log vào file và packet đó vẫn chịu sự kiểm tra của các rule còn lại.

Cách chia loại Target theo hành động:

- -j RETURN: Khiến gói tin hiện tại dừng di chuyển qua chuỗi hoặc chuỗi con
- -j ACCEPT: Luật được chấp nhận và sẽ không tiếp tục đi qua chuỗi hiện tại hoặc chuỗi khác trong bảng. Tuy nhiên gói tin được chấp nhận trong một chuỗi vẫn có thể di chuyển qua các chuỗi trong bảng khác và bị drop ở đó
- -j DNAT: Chỉ có trong chuỗi PREROUTING và OUTPUT trong bảng nat, và bất kỳ chuỗi nào được liệt kê ở đó
- -j SNAT: chỉ hợp lệ trong bảng nat, trong chuỗi POSTROUTING
- -j DROP: drop gói tin ở ngay đó
- -j REJECT: Gửi lại phản hồi (khác drop). Hợp lệ trong các chuỗi INPUT, FORWARD và OUTPUT hoặc chuỗi con
- -j LOG: Không hoạt động trên namespace
- -j ULOG: Thông tin của gói tin được multicast cùng với toàn bộ gói tin thông qua netlink socket. Chương trình chạy ở user-space có thể đăng ký và nhận gói tin
- -j MARK: chỉ hợp lệ trong bản mangle.
- -j MASQUERADE: Gần giống SNAT nhưng được sử dụng khi ip có thể thay đổi
- -j REDIRECT: chuyển hướng gói tin và streams đến máy tính. Hợp lệ với chains PREROUTING và OUTPUT của bảng nat. cũng hợp lệ với các chains do người dùng định nghĩa



Hình 2.14. Luồng đi của gói tin qua các chains trong các bảng

2.4.8. Mục tiêu nhảy giữa các chain

Mục tiêu nhảy là các hành động dẫn đến việc phải chuyển từ chain sang một chain khác để xử lý bổ sung. Iptables cũng cho phép quản trị viên tạo chain riêng cho mục đích tổ chức.

Các quy tắc có thể được đặt trong các chain do người dùng tự định nghĩa theo cùng cách mà chúng có thể được đặt vào các chain tích hợp sẵn. Sự khác biệt là các chain do người dùng xác định chỉ có thể đạt được bằng cách "nhảy" đến từ một rule (chúng không được tự đăng ký với netfilter hook).

2.4.9. Theo dõi kết nối trong Iptables

Theo dõi kết nối cho phép Iptables đưa ra quyết định về các gói tin trong ngữ cảnh của một kết nối. Hệ thống theo dõi kết nối cung cấp cho Iptables chức năng cần thiết để thực hiện các hoạt động stateful.

Theo dõi kết nối được tiến hành ngay sau khi các gói tin vào network stack. Các chains của table RAW và một số kiểm tra sơ bộ cơ bản là logic duy nhất được thực hiện trên các gói tin trước khi liên kết các gói tin với kết nối.

Hệ thống kiểm tra từng gói tin dựa trên một tập hợp các kết nối hiện có. Nó sẽ cập nhật trạng thái của kết nối nếu cần và sẽ thêm các kết nối mới vào hệ thống khi cần thiết. Các gói tin đã được đánh dấu là NOTRACK trong một trong các raw chains sẽ bỏ qua theo dõi kết nối.

Connection tracking cho phép Iptables đưa ra quyết định cho mỗi gói tin mà nó nhìn thấy dựa vào ngữ cảnh(context) của kết nối đang diễn ra. Quá trình Connection tracking diễn ra khá sớm trong vòng đời(lifecycle) của một gói tin. Hệ thống sẽ kiểm tra gói tin với tập hợp các kết nối đang có trên hệ thống, cập nhật trạng thái(state) nếu cần

hoặc thêm kết nối mới. Các gói tin được đánh dấu bằng target NOTRACK từ table raw sẽ được bypass quá trình tracking này.

2.4.10. Các trạng thái của kết nối

Đây là những trạng thái mà hệ thống connection tracking theo dõi:

- NEW: Khi có một gói tin mới được gửi tới và không nằm trong bất kỳ connection nào hiện có, hệ thống sẽ khởi tạo một kết nối mới và gắn cho kết nối này nhãn NEW. Nhãn này dùng cho cả TCP và UDP.
- ESTABLISHED: Trạng thái chuyển NEW to ESTABLISHED khi nhận được phản hồi hợp lệ từ phía đối diện của kết nối. Đối với kết nối TCP, phản hồi hợp lệ chính là SYN/ACK và với UDP/ICMP, là phản hồi mà ở đó địa chỉ nguồn và địa chỉ đích được hoán đổi.
- RELATED: Gói tin được gởi tới không thuộc về một kết nối hiện có nhưng có liên quan đến một kết nối đang có trên hệ thống. Đây có thể là một kết nối phụ hỗ trợ cho kết nối chính, ví dụ như giao thức FTP có kết nối chính dùng để chuyển lệnh và kết nối phụ dùng để truyền dữ liệu.
- INVALID: Gói tin được đánh dấu INVALID khi gói tin này không có bất cứ quan hệ gì với các kết nối đang có sẵn, không thích hợp để khởi tạo một kết nối mới hoặc đơn giản là không thể xác định được gói tin này, không tìm được kết quả trong bảng định tuyến.
- UNTRACKED: Gói tin có thể được gắn nhãn UNTRACKED nếu gói tin này đi qua bảng raw và được xác định là không cần theo dõi gói này trong bảng connection tracking.
- SNAT: Đó là trạng thái sẽ được đánh dấu khi gói tin được chỉnh sửa phần source address bởi quá trình NAT. Trạng thái này được dùng bởi hệ thống Connection tracking để thay đổi lại source address ở gói tin phản hồi lại.
- DNAT: Đó là trạng thái sẽ được đánh dấu khi gói tin được chỉnh sửa phần destination address bởi quá trình NAT. Trạng thái này được dùng bởi hệ thống Connection tracking để thay đổi lại destination address ở gói tin phản hồi lại.

Các trạng thái được theo dõi trong hệ thống theo dõi kết nối cho phép quản trị viên tạo ra các quy tắc nhắm vào mục tiêu là các điểm cụ thể trong vòng đời của kết nối. Điều này cung cấp các chức năng cần thiết cho các quy tắc an toàn và kỹ lưỡng hơn.

CHƯƠNG 3. GIẢI PHÁP

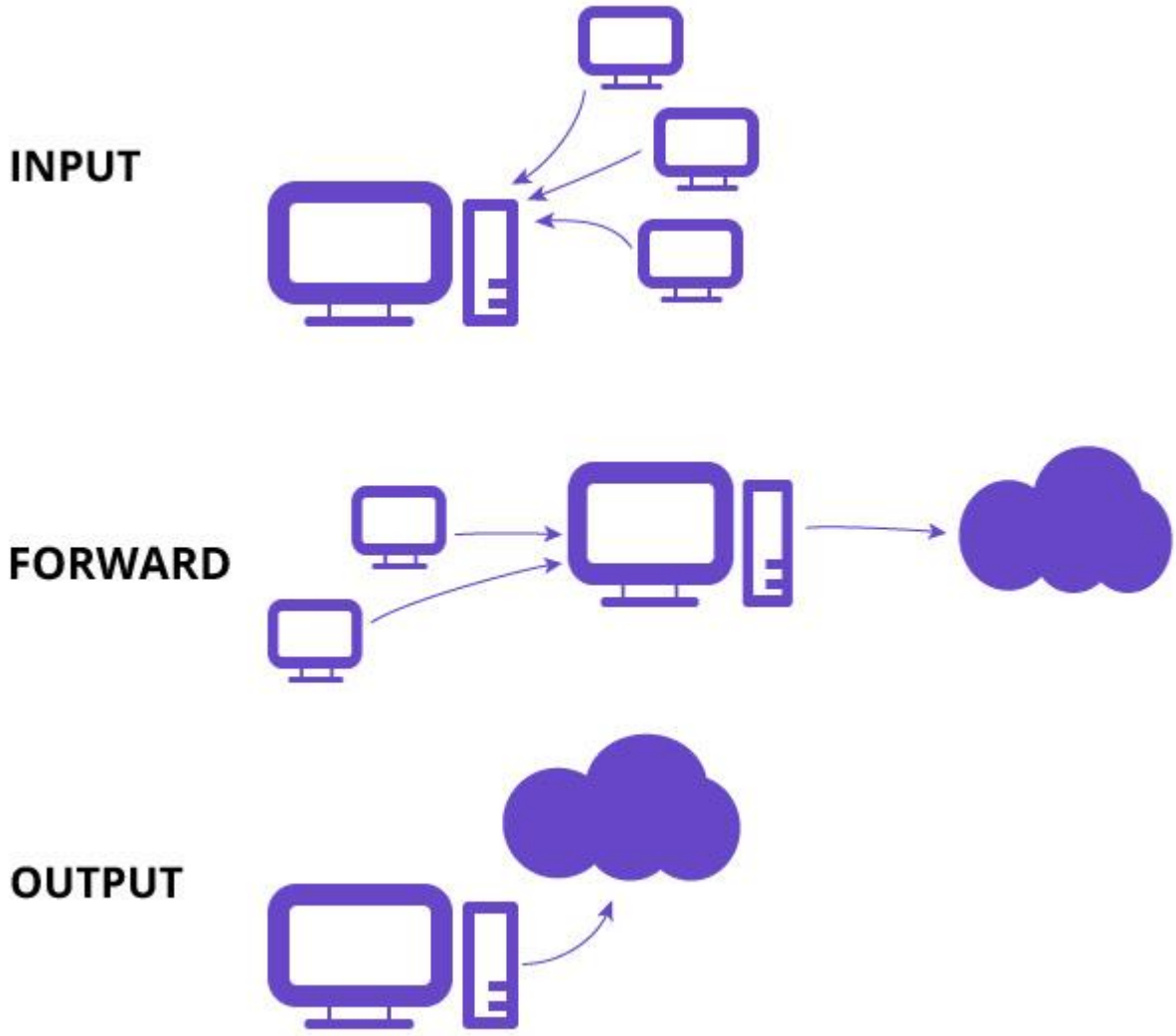
3.1. Xây dựng chương trình bằng C++

3.1.1. Ý tưởng và cấu trúc của chương trình

Ban đầu, thường thì khi muốn viết một chương trình để bắt các gói tin đi qua network interface, giải pháp sẽ là sử dụng libpcap hoặc các thư viện có khả năng tương tự. Tuy nhiên, thứ mà những thư viện này bắt được chỉ là bản sao của gói tin đã đi qua, nên nếu dùng các thư viện này thì chương trình chỉ có thể hoạt động được ở chế độ IDS. Vì vậy, trong đồ án này, chương trình sẽ tiếp cận theo một hướng khác, đó là sử dụng Netfilter để chương trình của người dùng có quyền lấy được thông tin và quyết định hành động đối với gói tin, theo hướng này, chương trình sẽ có thể hoạt động được ở chế độ IPS và có thể dễ dàng đưa ra hành động tương ứng với từng gói tin riêng biệt. Một ưu điểm nữa của cách làm này là các máy chạy chương trình cũng không cần cài đặt libpcap để có thể chạy được chương trình, làm cho chương trình dễ cài đặt hơn.

Như vậy, chương trình sẽ không tự thực hiện việc bắt gói tin mà sẽ lấy gói tin ra từ nfqueue (cấu hình để Iptables đẩy các gói tin đi đến hoặc đi qua network interface vào queue).

```
sudo iptables -I INPUT -j NFQUEUE
sudo iptables -I FORWARD -j NFQUEUE
sudo iptables -I OUTPUT -j NFQUEUE
```



Hình 3.1. Hướng đi của các gói tin được Iptables đẩy vào nfqueue ứng với các cấu hình trên

Các gói tin lấy ra sẽ được sẽ ở dạng unsigned char*, bằng cách dùng các struct có cấu trúc tương tự như header của gói tin IPv4, ICMP, TCP, UDP. Có thể ép kiểu để lấy ra các thông tin cần thiết như địa chỉ IP nguồn và IP đích, cổng nguồn và cổng đích, giao thức của gói tin.

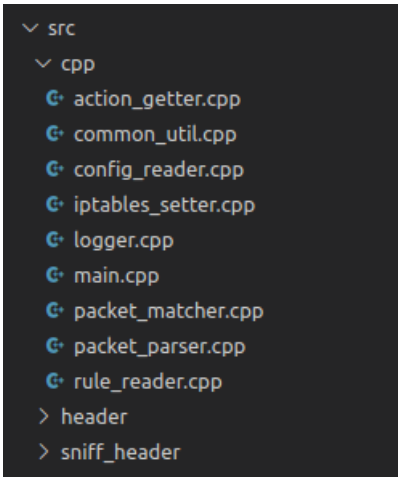
Sau khi có các thông tin trên, chương trình sẽ tiến hành so sánh gói tin với tập luật có sẵn, nếu khớp với bất kỳ luật nào, sẽ đưa ra hành động tương ứng với gói tin đó.

Mục tiêu là viết một chương trình C++ chạy trên Linux có khả năng bắt các gói tin đi qua hoặc đi đến, so sánh với tập luật, và đưa ra hành động tương ứng.

Kiến trúc của chương trình sẽ bao gồm các thành phần:

- Đọc tập luật (đọc và lấy ra tập luật lúc mới chạy chương trình)
- Thu thập gói tin (lấy từ nfqueue)

- Phân tích gói tin
- Quyết định hành động với gói tin
- Phản hồi gói tin (đưa ra hành động)
- Ghi log
- Thao tác với Iptables (người dùng không cần tự cấu hình Iptables)
- Lấy ra trạng thái CPU hiện tại của máy tính



Hình 3.2. Cấu trúc của chương trình

Mã nguồn của chương trình được chia làm 3 folder:

- Folder cpp chứa các thành phần kể trên và một số thành phần phụ trợ.
- Folder header chứa header của các thành phần.
- Folder sniff_header chứa header của các gói tin IP, TCP, UDP, ICMP.

3.1.2. Các tính năng chính mà chương trình hỗ trợ

Chương trình có thể hoạt động với một trong hai chế độ IDS (giám sát và cảnh báo) hoặc IPS (xử lý xâm nhập).

Chế độ IDS cho phép chương trình giám sát và đưa ra cảnh bảo với tất cả các gói tin đi qua.

Chế độ IPS bao gồm tất cả các tính năng của chế độ IDS, cộng thêm khả năng có thể drop gói tin (ngăn chặn xâm nhập).

Khi đặt chương trình ở trước gateway, nó sẽ theo dõi tất cả các gói tin đi qua mạng. Khi đặt chương trình ở một máy tính, nó sẽ theo dõi các gói tin vào ra máy tính đó. Điều này cho phép chương trình có thể hoạt động như NIDS/NIPS hoặc HIDS/HIPS.

Chương trình sẽ hoạt động dựa trên signature của gói tin, tức là sẽ so sánh gói tin với tập luật để đưa ra hành động, gói tin có thông tin khớp với luật nào sẽ được xử lý theo luật đó.

Tập luật do người dùng tùy chỉnh, cần phải theo cấu trúc, thứ tự ưu tiên của các luật là từ trên xuống dưới.

Các hành động với gói tin mà chương trình hỗ trợ:

- Pass: gói tin đi qua bình thường
- Alert: ghi log về thông tin của gói tin, cho đi qua
- Drop: ghi log về thông tin của gói tin, không cho đi qua

3.1.3. Cấu trúc luật của chương trình

Luật sẽ bao gồm 2 phần chính là header và option.

Phần header trong một luật bao gồm các thông tin như hành động được thực thi (action), giao thức mạng khớp với luật (protocol), các thông tin về như địa chỉ IP nguồn và IP đích, cổng nguồn và cổng đích.

Action:

- Pass: Gói tin có signature trùng khớp với hành động này sẽ được đi qua.
- Alert: Gói tin có signature trùng khớp với hành động này sẽ được đi qua và ghi lại thông tin.
- Drop: Gói tin có signature trùng khớp với hành động này sẽ không được đi qua và ghi lại thông tin.

Protocol:

- Ip: Tất cả các gói tin IPv4 đều ứng với protocol này
- Tcp: Các gói tin TCP đều ứng với protocol này
- Udp: Các gói tin UDP đều ứng với protocol này
- Icmp: Các gói tin ICMP đều ứng với protocol ày

Địa chỉ IP nguồn và IP đích sẽ tuân theo định dạng như bảng sau:

Bảng 3.1. Định dạng địa chỉ ip của rule

Định dạng	Ý nghĩa
x.x.x.x	Địa chỉ chính xác, ví dụ 10.11.12.13
x.x.x.x/n	Địa chỉ subnet, ví dụ 10.10.0.0/16
!	Phủ định, ví dụ !10.11.12.13
any	Tất cả các địa chỉ

Địa chỉ cổng nguồn và cổng đích sẽ tuân theo định dạng như bảng sau:

Bảng 3.2. Định dạng địa chỉ port của rule

Định dạng	Ý nghĩa
x	Cổng chính xác, ví dụ 80

!	Phủ định, ví dụ !80
any	Tất cả các cổng

Option là phần quan trọng chỉ ra những dấu hiệu bất thường để phát hiện sự xâm nhập. Các Option được bọc bởi dấu ngoặc đơn và cách nhau dấu chấm phẩy.

Bảng 3.3. Định dạng option của rule

Tên	Ý nghĩa
Second, count và timeout	Sẽ thực hiện action khi có c gói tin khớp với rule trong vòng s giây, và sẽ thôi thực hiện action đó sau t giây, thường dùng để chống dos, ví dụ: (count: 100; second: 5; timeout: 5)
cpu	Sẽ thực hiện action khi có gói tin khớp với rule và tại thời điểm đó cpu usage đang lớn hơn mức này, ví dụ: (cpu: 70;)

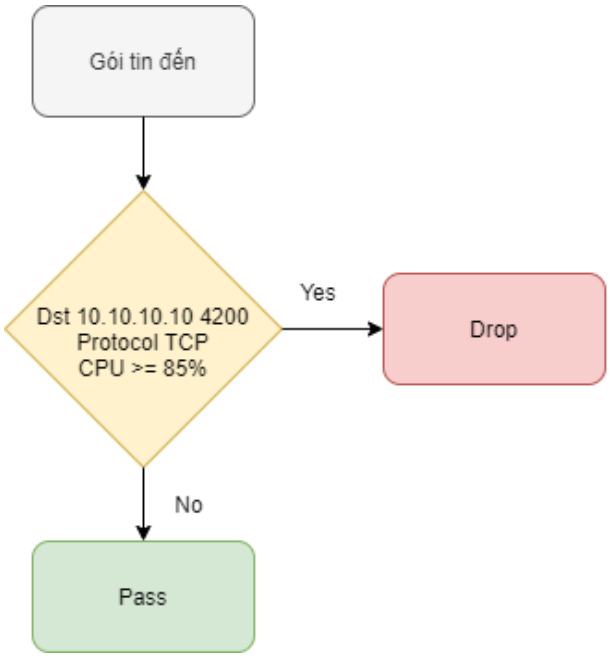
Rule format:

```
action protocol src_ip src_port → dst_ip dst_port (option)
```

Ví dụ:

```
drop tcp any any → 10.10.10.10 4200 (cpu: 85;)
```

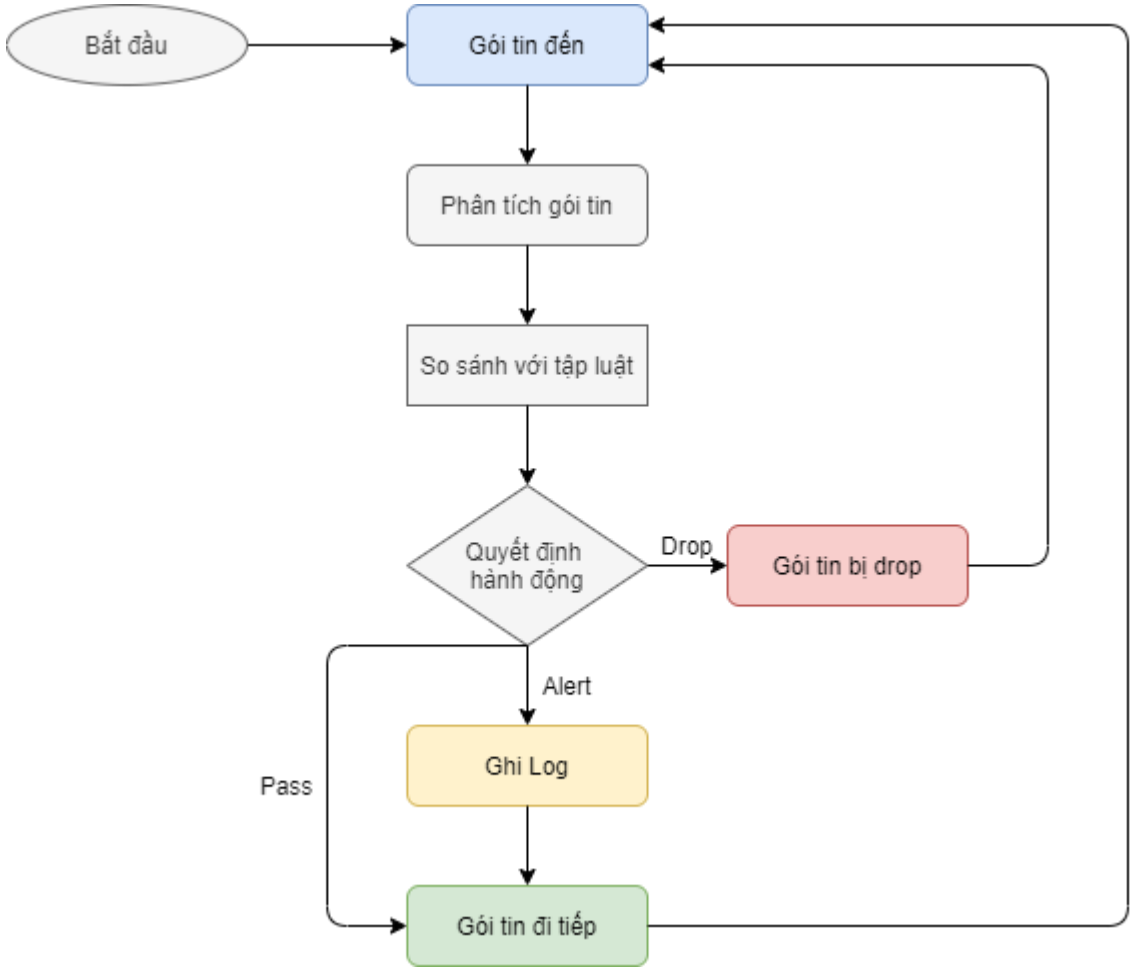
Có nghĩa là sẽ drop các gói tin tcp từ tất cả các nguồn đến host 10.10.10.10 ở port 4200 khi cpu đang cao hơn mức 85%.



Hình 3.3. Ví dụ luồng chạy với rule trên

Thông thường packet sẽ bị drop khi CPU usage đang ở mức cao, tuy nhiên, khi CPU usage đạt tới mức phải drop packet thì sẽ gây ra ảnh hưởng đến toàn bộ hệ thống, vì vậy, khi được cấu hình hợp lý, việc drop packet khi CPU usage đạt tới một mức nào đó sẽ có ý nghĩa lớn trong việc đảm bảo cho toàn hệ thống hoạt động ổn định (đánh đổi với việc đáp ứng ít request hơn).

3.1.4. Lập trình và luồng chạy của chương trình



Hình 3.4. Luồng chạy của chương trình

Khi mới bắt đầu, chương trình sẽ tiến hành các bước chuẩn bị, bao gồm:

- Đọc ra tập luật và lưu vào bộ nhớ của chương trình
- Cấu hình để Iptables gửi gói tin đến nfqueue
- Binding chương trình với nfqueue để lấy ra gói tin

Sau khi hoàn thành bước chuẩn bị, mỗi gói tin đi đến hoặc đi qua máy chạy chương trình sẽ được chuyển đến nfqueue, chương trình sẽ lấy gói tin ra từ queue và phân tích. Các bước phân tích gói tin bao gồm:

- Ép kiểu gói tin nhận được thành struct sniff_ip, lấy ra thông tin về giao thức và địa chỉ nguồn, địa chỉ đích.

- Sau khi có thông tin về giao thức, tùy vào đó là ICMP, TCP, UDP sẽ tiếp tục ép kiểu để lấy ra thông tin cổng.

Sau khi có các thông tin cần thiết của gói tin, chương trình sẽ lần lượt so sánh với tập luật. Khi gặp một luật phù hợp, sẽ dừng so sánh và xử lý gói tin theo hành động trong luật đó. Bao gồm các bước:

- So sánh với header của luật, nếu khớp thì sẽ tiến hành so sánh với option
- So sánh với option của luật, tùy vào các kiểu option mà sẽ có cách làm khác nhau

Cuối cùng là bước đưa ra hành động:

- Nếu hành động là pass, gói tin sẽ được đi qua bình thường.
- Nếu hành động là alert, chương trình sẽ ghi log về thông tin của gói tin và cho gói tin đi tiếp.
- Nếu hành động là drop, gói tin sẽ không được đi tiếp nữa.

Thành phần thu thập gói tin (gói tin đi, đến và đi qua)

Binding chương trình với nfqueue, mỗi gói tin đi được lấy ra từ queue sẽ được gọi hàm callback để phân tích gói tin [6].

```
h = nfq_open();
if (!h)
{
    exit(1);
}
if (nfq_unbind_pf(h, AF_INET) < 0)
{
    exit(1);
}
if (nfq_bind_pf(h, AF_INET) < 0)
{
    exit(1);
}
qh = nfq_create_queue(h, 0, &callback, NULL);
if (!qh)
{
    exit(1);
}
if (nfq_set_mode(qh, NFQNL_COPY_PACKET, 0xffff) < 0)
{
    exit(1);
}
fd = nfq_fd(h);
while ((rv = recv(fd, buf, sizeof(buf), 0)))
{
    nfq_handle_packet(h, buf, rv);
}
nfq_destroy_queue(qh);
```

Hình 3.5. Binding với nfqueue để lấy ra gói tin

Thành phần phân tích gói tin (gói tin IPv4)

Gói tin nhận được có kiểu u_char* sẽ được ép thành kiểu sniff_ip để lấy ra thông tin, từ đó biết được giao thức và địa chỉ ip, từ đó tiếp tục phân tích để lấy ra port [7].

```
/* Ip header */
struct sniff_ip {
    u_char  ip_vhl;                 /* version << 4 | header length >> 2 */
#define IP_HL(ip)               (((ip)->ip_vhl) & 0x0f)
#define IP_V(ip)                (((ip)->ip_vhl) >> 4)
    u_char  ip_tos;                 /* type of service */
    u_short ip_len;                 /* total length */
    u_short ip_id;                  /* identification */
    u_short ip_off;                 /* fragment offset field */
    #define IP_RF 0x8000            /* reserved fragment flag */
    #define IP_DF 0x4000            /* dont fragment flag */
    #define IP_MF 0x2000            /* more fragments flag */
    #define IP_OFFMASK 0x1fff       /* mask for fragmenting bits */
    u_char  ip_ttl;                 /* time to live */
    u_char  ip_p;                   /* protocol */
    u_short ip_sum;                 /* checksum */
    struct  in_addr ip_src,ip_dst;  /* source and dest address */
};
```

Hình 3.6. Struct sniff_ip, có cấu trúc tương tự với header của gói tin IPv4

Thành phần phản hồi

Sau khi có được thông tin của gói tin, sẽ so sánh với tập luật, và đưa ra quyết định với gói tin.

```
    u_int32_t id;
    struct nfqnl_msg_packet_hdr *ph;
    ph = nfq_get_msg_packet_hdr(nfa);
    id = ntohl(ph->packet_id);
    return nfq_set_verdict(qh, id, nf_action, 0, NULL);
```

Hình 3.7. Quyết định hành động đối với gói tin

Chi tiết mã nguồn của chương trình: https://github.com/chutichnuoc/magic

3.2. Triển khai chương trình trên Linux

```
sudo ./target/magic IPS NET /home/hung/config.ini
```

Hình 3.8. Ví dụ về cách chạy chương trình

Vì chương trình có tác động lên Iptables nên yêu cầu quyền sudo.

Các tham số của chương trình lần lượt là là:

- Chế độ IDS hoặc IPS
- Bảo vệ HOST hoặc NET
- Đường dẫn đến file config

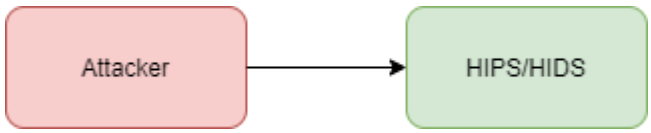
```
ruleFile = /home/hung/magic/rules/test.rules
logFile = /home/hung/magic/log/log.txt
iptablesFile = /home/hung/rules.v4
```

Hình 3.9. Nội dung file chứa các config của chương trình

File config sẽ chứa:

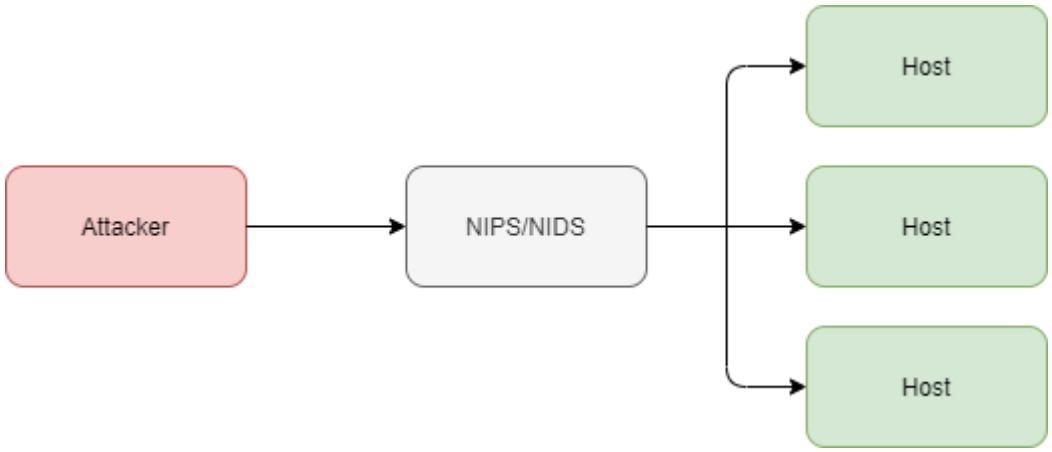
- Đường dẫn đến file chứa tập luật
- Đường dẫn đến file ghi log
- Đường dẫn đến file để backup Iptables

Có 2 lựa chọn về vị trí để đặt chương trình:



Hình 3.10. Chương trình hoạt động ở host

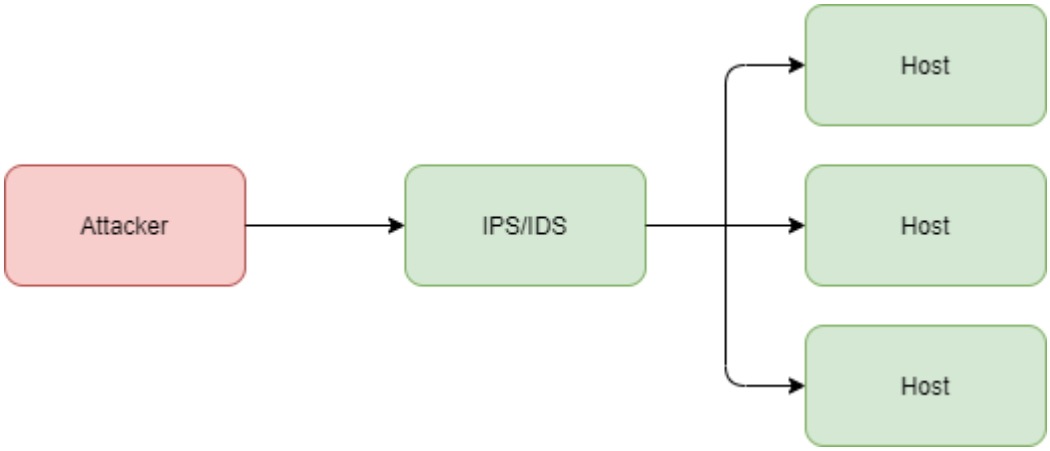
Chương trình hoạt động như HIDS/HIPS, theo dõi và xử lý các gói tin đến và đi ở máy đặt chương trình (màu xanh).



Hình 3.11. Chương trình hoạt động ở mạng.

Chương trình hoạt động như NIDS/NIPS, theo dõi và xử lý các gói tin đến và đi ở các máy đặt phía sau máy chương trình (màu xanh).

Về mặt kỹ thuật, hoàn toàn có thể có một công cụ bảo vệ cả mạng lẫn chính máy đặt nó, tuy nhiên trong phạm vi báo cáo này sẽ không đề cập đến.

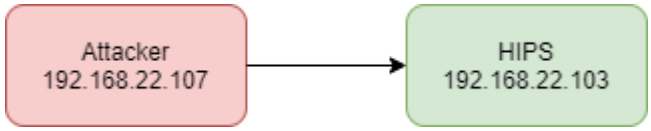


Hình 3.12. Chương trình hoạt động ở cả host và mạng.

CHƯƠNG 4. KẾT QUẢ

Kịch bản 1: Host IPS

Trên web server 192.168.22.103 đang có web service hoạt động ở cổng 80. Tiến hành siege web server từ host 192.168.22.107 trong hai trường hợp không chạy chương trình và có chạy chương trình ở chế độ HIPS, theo dõi CPU của web server trong cả hai trường hợp và so sánh.



Hình 4.1. Sơ đồ các máy

```
drop tcp any any -> 192.168.22.103 80 (cpu: 80;)
alert ip any any -> 192.168.22.103 any
```

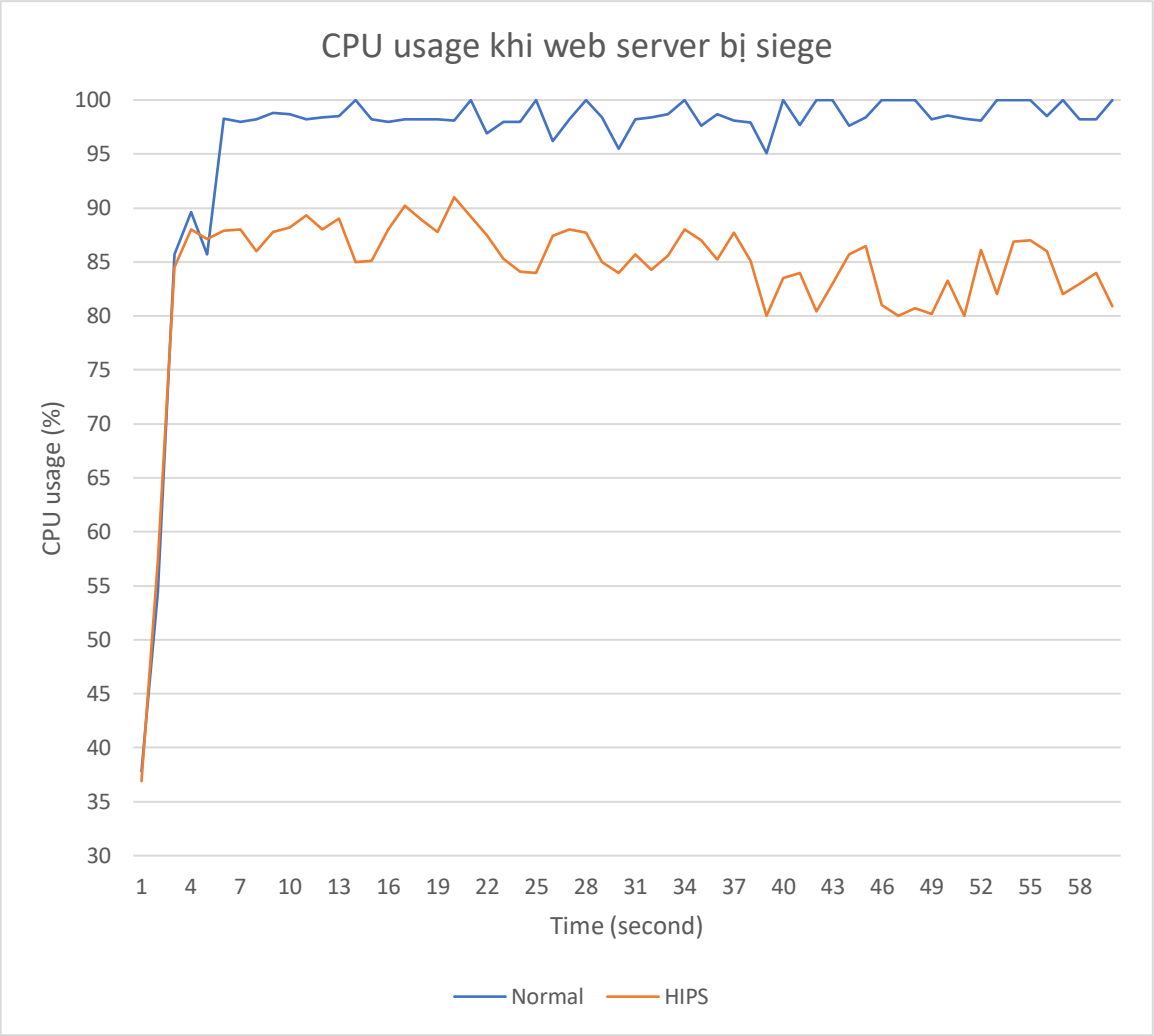
Hình 4.2. Rules của chương trình

```
sudo ./target/magic IPS HOST /home/hung/config.ini
```

Hình 4.3. Chạy chương trình với với mode HIPS

```
siege -c250 192.168.22.103
```

Hình 4.4. Siege web server đặt trên 192.168.22.103



Biểu đồ 4.1 Trạng thái CPU của server trong vòng 60 giây từ khi siege

Nhận xét

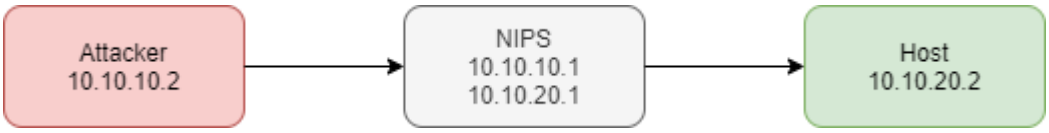
Ở trạng thái bình thường, CPU của web server hoạt động ở mức 35%.

Khi chạy bình thường (hoặc chạy các IPS không có option cpu), CPU của server luôn ở trên mức 95% và thường xuyên chạm mức 100%. Web server bị quá tải, xảy ra hiện tượng "treo".

Khi chạy chương trình với rule drop gói tin khi cpu đang cao hơn 80%, CPU của server thường nằm trong khoảng từ 80%-90%. Điều này xảy ra do chương trình sẽ drop tất cả các gói tin đến khi CPU đang cao hơn 80%, web server sẽ không phải tiếp nhận và xử lý những request này nữa. Lúc này, web server sẽ hoạt động ổn định và không xảy ra hiện tượng "treo". Đánh đổi với việc server sẽ tiếp nhận ít request hơn.

Kịch bản 2: Network IPS

Dùng từ host 10.10.10.2, dùng hping để dos host 10.10.20.2. Chạy chương trình và theo dõi kết quả.



Hình 4.5. Sơ đồ các máy

```
drop ip any any -> 10.10.20.2 8001 (count: 100; second: 5;)
alert ip any any -> any any
```

Hình 4.6. Rules của chương trình

```
sudo ./target/magic IPS NET /home/hung/config.ini
```

Hình 4.7. Chạy chương trình

```
sudo hping3 10.10.10.2 -q -p 8001 --flood --rand-source
```

Hình 4.8. Tiến hành dos host 10.10.20.2

```
tcp    96.52.94.63:2505 -> 10.10.20.2:8001
tcp    120.45.229.134:2506 -> 10.10.20.2:8001
tcp    98.229.129.202:2507 -> 10.10.20.2:8001
tcp    152.158.137.96:2509 -> 10.10.20.2:8001
tcp    248.221.12.104:2510 -> 10.10.20.2:8001
tcp    100.158.30.70:2512 -> 10.10.20.2:8001
tcp    123.185.124.30:2513 -> 10.10.20.2:8001
tcp    120.247.155.255:2514 -> 10.10.20.2:8001
tcp    13.197.216.170:2515 -> 10.10.20.2:8001
tcp    74.83.124.89:2517 -> 10.10.20.2:8001
tcp    115.204.49.153:2518 -> 10.10.20.2:8001 (dropped)
tcp    182.194.10.45:2519 -> 10.10.20.2:8001 (dropped)
tcp    38.152.177.249:2520 -> 10.10.20.2:8001 (dropped)
tcp    253.248.38.22:2521 -> 10.10.20.2:8001 (dropped)
tcp    203.167.8.169:2522 -> 10.10.20.2:8001 (dropped)
tcp    59.155.54.8:2523 -> 10.10.20.2:8001 (dropped)
tcp    80.229.59.128:2524 -> 10.10.20.2:8001 (dropped)
tcp    253.98.21.239:2525 -> 10.10.20.2:8001 (dropped)
tcp    98.182.176.20:2526 -> 10.10.20.2:8001 (dropped)
tcp    45.151.12.129:2527 -> 10.10.20.2:8001 (dropped)
tcp    98.104.52.42:2528 -> 10.10.20.2:8001 (dropped)
tcp    179.52.210.43:2529 -> 10.10.20.2:8001 (dropped)
```

Hình 4.9. Output của chương trình

```
17-11-2020 20:41:36     tcp    96.52.94.63:2505 -> 10.10.20.2:8001
17-11-2020 20:41:36     tcp    120.45.229.134:2506 -> 10.10.20.2:8001
17-11-2020 20:41:36     tcp    98.229.129.202:2507 -> 10.10.20.2:8001
17-11-2020 20:41:36     tcp    152.158.137.96:2509 -> 10.10.20.2:8001
17-11-2020 20:41:37     tcp    248.221.12.104:2510 -> 10.10.20.2:8001
17-11-2020 20:41:37     tcp    100.158.30.70:2512 -> 10.10.20.2:8001
17-11-2020 20:41:37     tcp    123.185.124.30:2513 -> 10.10.20.2:8001
17-11-2020 20:41:37     tcp    120.247.155.255:2514 -> 10.10.20.2:8001
17-11-2020 20:41:37     tcp    13.197.216.170:2515 -> 10.10.20.2:8001
17-11-2020 20:41:38     tcp    74.83.124.89:2517 -> 10.10.20.2:8001
17-11-2020 20:41:38     tcp    115.204.49.153:2518 -> 10.10.20.2:8001 (dropped)
17-11-2020 20:41:38     tcp    182.194.10.45:2519 -> 10.10.20.2:8001 (dropped)
17-11-2020 20:41:38     tcp    38.152.177.249:2520 -> 10.10.20.2:8001 (dropped)
17-11-2020 20:41:38     tcp    253.248.38.22:2521 -> 10.10.20.2:8001 (dropped)
17-11-2020 20:41:38     tcp    203.167.8.169:2522 -> 10.10.20.2:8001 (dropped)
17-11-2020 20:41:39     tcp    59.155.54.8:2523 -> 10.10.20.2:8001 (dropped)
17-11-2020 20:41:39     tcp    80.229.59.128:2524 -> 10.10.20.2:8001 (dropped)
17-11-2020 20:41:39     tcp    253.98.21.239:2525 -> 10.10.20.2:8001 (dropped)
17-11-2020 20:41:39     tcp    98.182.176.20:2526 -> 10.10.20.2:8001 (dropped)
17-11-2020 20:41:39     tcp    45.151.12.129:2527 -> 10.10.20.2:8001 (dropped)
17-11-2020 20:41:39     tcp    98.104.52.42:2528 -> 10.10.20.2:8001 (dropped)
17-11-2020 20:41:40     tcp    179.52.210.43:2529 -> 10.10.20.2:8001 (dropped)
```

Hình 4.10. Log của chương trình

Nhận xét

Từ thời điểm 20:41:38, chương trình đã đếm được 100 gói tin đến 10.10.20.2 ở cổng 8001 trong vòng 5 giây, nên các gói tin khớp với luật này sẽ xử lý theo rule, trong trường hợp này là drop, tất cả các gói tin cùng loại đến sau đó trong lần chạy này của chương trình cũng sẽ bị drop.

Chương trình đã có khả năng giúp phòng chống dos nếu được cấu hình đúng, các máy ở phía sau NIPS được bảo vệ khỏi cuộc tấn công.

Khi áp dụng tùy chọn này, chương trình chỉ đơn giản là loại bỏ các gói tin khi phát hiện dấu hiện của dos, chứ không liên quan đến trạng thái của máy tính.

CHƯƠNG 5. KẾT LUẬN

Trong đồ án này em đã nghiên cứu, tìm hiểu về IDS, IPS, Netfilter, Iptables, C++ và xây dựng công cụ phát hiện và xử lý gói tin theo tập luật tùy chỉnh.

Em đã áp dụng các kiến thức đã học của các môn như mạng máy tính, an toàn và an ninh mạng, lập trình mạng, …

Đồ án đã thực hiện thành công việc xây dựng một công cụ có tính năng tương tự như IDS/IPS và đáp ứng được yêu cầu có thể xử lý gói tin dựa trên trạng thái hiện tại của máy tính.

Tuy nhiên với kinh nghiệm và kiến thức còn hạn chế, trong quá trình thực hiện đồ án em không thể tránh khỏi những thiếu sót.

Trong thời gian tới em sẽ tiếp tục phát triển chương trình, bao gồm các tính năng:

- Bổ sung thêm các tùy chọn mới cho luật.
- Bổ sung thêm tính năng xác thực dữ liệu đầu vào cho chương trình.
- Bổ sung chức năng cảnh báo tới người dùng trong trường hợp cần thiết.

TÀI LIỆU THAM KHẢO

Tiếng Việt

[1] https://securitydaily.net/an-toan-thong-tin-mang/

[2] https://securitydaily.net/network-hieu-ve-he-thong-phat-hien-xam-nhap-ids/

[3] https://securitydaily.net/network-security-he-thong-ngan-ngua-xam-nhap-ips/

Tiếng Anh

[4] https://www.w3.org/People/Frystyk/thesis/TcpIp.html

[5] https://www.digitalocean.com/community/tutorials/a-deep-dive-into-Iptables-and-netfilter-architecture

[6] https://github.com/irontec/netfilter-nfqueue-samples

[7] http://yuba.stanford.edu/~casado/pcap/section4.html