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. So với những công cụ kể trên, công cụ này còn nhiều thiếu sót, tuy nhiên nó có tùy chọn để xử lý gói tin dựa theo tình trạng của máy tính, cụ thể là mức độ sử dụng của CPU tại thời điểm hiện tại. Em tin đây là một hướng đi lạ 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

Ngày nay, nhu cầu trao đổi dữ liệu qua hệ thống mạng máy tính trở thành vô cùng quan trọng trong mọi hoạt động của xã hội. Vấn đề đảm bảo an ninh, an toàn cho thông tin trên mạng ngày càng là mối quan tâm hàng đầu của các công ty, tổ chức, nhà cung cấp dịch vụ. Cùng với thời gian, các kỹ thuật tấn công ngày càng tinh vi hơn khiến các hệ thống an ninh mạng trở nên không hiệu quả. Các hệ thống an ninh mạng truyền thống thuần túy dựa trên các tường lửa nhằm kiểm soát luồng thông tin vào ra hệ thống mạng một cách cứng nhắc dựa trên các luật bảo vệ cố định. Với kiểu phòng thủ này, các hệ thống sẽ bất lực trước các kỹ thuật tấn công mới, đặc biệt là các cuộc tấn công nhằm vào điểm yếu của hệ thống.

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).

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 an toàn mạng, các mô hình hệ thống phát hiện xâm nhập, 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 trên host và trong mạng.

CHƯƠNG 2. LÝ THUYẾT

2.1. Các giao thức ở tầng mạng và giao vận

2.1.1. Internet Protocol

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

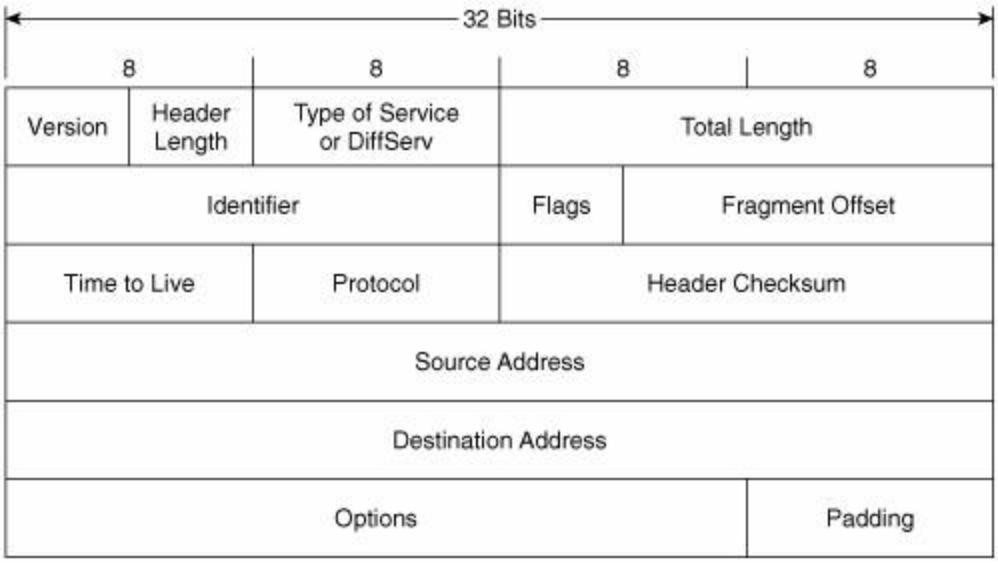
Dữ liệu trong một liên mạng IP được gửi theo các khối được gọi là các gói (packet hoặc datagram). Cụ thể, khi gửi một gói tin từ máy chủ đến một thiết bị khác chưa từng có hoạt động liên lạc trước đó, IP sẽ không cần phải thiết lập trươc tuyến đường truyền tin, việc định tuyến sẽ diễn ra khi gói tin được gửi đi.

IP cung cấp một dịch vụ gửi dữ liệu gọi là cố gắng cao nhất hay không đảm bảo, điều này nghĩa là nó hầu như không đảm bảo gì về tính toàn vẹn gói dữ liệu được gửi đi. Gói dữ liệu được gửi có thể đến điểm đích mà không còn nguyên vẹn về dữ liệu, cũng như nó có thể đến mà không theo thứ tự được gửi ở máy nguồn, nó có thể bị trùng lặp hoặc bị mất hoàn toàn.Khi một ứng dụng phần mềm cần có được sự bảo đảm cho gói tin gửi đi của mình, nó có thể nhận được sự bảo đảm này từ nơi khác, thường từ các giao thức giao vận nằm phía trên IP.

Các thiết bị định tuyến liên mạng chuyển tiếp các gói tin IP qua các mạng tầng liên kết dữ liệu được kết nối với nhau. Việc không có đảm bảo về gửi dữ liệu có nghĩa rằng các chuyển mạch gói có thiết kế đơn giản hơn. (Lưu ý rằng nếu mạng bỏ gói tin, làm đổi thứ tự hoặc làm hỏng nhiều gói tin, người dùng sẽ thấy hoạt động mạng trở nên kém đi. Hầu hết các thành phần của mạng đều cố gắng tránh để xảy ra tình trạng đó. Đó là lý do giao thức này còn được gọi là cố gắng cao nhất. Tuy nhiên, khi lỗi xảy ra không thường xuyên sẽ không có hiệu quả đủ xấu đến mức người dùng nhận thấy được.)

IP rất thông dụng trong mạng Internet công cộng ngày nay. Giao thức tầng mạng thông dụng nhất ngày nay là IPv4; đây là giao thức IP phiên bản 4. IPv6 được đề nghị sẽ kế tiếp IPv4: Internet đang hết dần địa chỉ IPv4, do IPv4 sử dụng 32 bit để đánh địa chỉ (tạo được khoảng 4 tỷ địa chỉ); IPv6 dùng địa chỉ 128 bit, cung cấp tối đa khoảng 3.4×1038 địa chỉ. Các phiên bản từ 0 đến 3 hoặc bị hạn chế, hoặc không được sử dụng. Phiên bản 5 được dùng làm giao thức dòng (stream) thử nghiệm. Còn có các phiên bản khác, nhưng chúng thường dành là các giao thức thử nghiệm và không được sử dụng rộng rãi.

Địa chỉ IP được chia thành 4 số giới hạn từ 0 - 255. Mỗi số được lưu bởi 1 byte, IP có kích thước là 4 byte, được chia thành các lớp địa chỉ. Có 5 lớp là A, B, C, D, E và loopback.



Hình 2.1. Header của gói tin IPv4

Header của gói tin IPv4 bao gồm 13 trường, trong đó 12 trường là bắt buộc. Trường thứ 13 là tùy chọn, đúng với tên của nó: options. Các trường này trong header được lưu trữ với byte có ý nghĩa cao (the most significant byte) ở địa chỉ thấp (big endian), nói cách khác bit có ý nghĩa cao luôn ở địa chỉ thấp. Bit quan trọng nhất là bit số 0, vì vậy trường phiên bản (version) được lưu trong 4 bits đầu tiên của byte đầu tiên.

Phiên bản (Version): Trường đầu tiên trong header của gói tin IP chính là trường Phiên bản (Version) dài 4 bit. Với IPv4, nó có giá trị bằng 4.

Độ lớn của header (Internet Header Length) (IHL): Trường thứ hai (4 bit) là độ lớn của header (Internet Header Length - IHL) cho biết số lượng các từ 32-bit trong header. Vì một header của gói tin IPv4 có thể chứa rất nhiều tùy chọn (options), trường này cho biết kích thước của header (nó cũng trùng với offset của data). Giá trị nhỏ nhất cho trường này là 5 (RFC 791), do đó gói tin có độ dài là 5×32 = 160 bit. Vì đây là số 4 bits nên độ dài lớn nhất có thể được của gói tin là 15 từ (15×32 bit) tức là 480 bitt.

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 mong muốn khi truyền các gói tin qua router. Trường này có 8 bit, xác định quyền ưu tiên, độ trễ, thông lượng, các đặc tính chỉ định độ tin cậy khác. 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, sử dụng 8 mức từ 0-7. Các công nghệ mới xuất hiện yêu cầu các dòng dữ liệu thời gian thực

(real-time data streaming) và sẽ sử dụng trường DS. Ví dụ Voice over IP (VoIP) được dùng để trao đổi dữ liệu là tiếng nói.

Total Length: Chỉ định tổng chiều dài gói tin IPv4 (cả phần mào đầu và phần dữ liệu). Kích thước 16 bit, chỉ định rằng gói tin IPv4 nhỏ nhất là 20 byte (chỉ có header không có dữ liệu) và có thể lớn tới 65.535 byte.

Identification: Định danh gói tin. Kích thước 16 bit. Định danh cho gói tin được cấp bởi nút nguồn 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 các phân mảnh đều có trường định danh gói tin này giúp nút đích có thể phục hồi gói tin.

2.1.2. Transmission Control Protocol

Transmission Control Protocol là một trong các giao thức cốt lõi của bộ giao thức TCP/IP. Các ứng dụng trên các máy chủ được kết nối mạng sẽ sử TCP để tạo các "kết nối" với nhau , mà thông qua kết nối đó chúng có thể trao đổi dữ liệu hoặc các gói tin. Đây là một giao thức đảm bảo chuyển giao dữ liệu tới nơi nhận một cách đáng tin cậy và theo đúng thứ tự các gói tin được gửi. TCP còn có thể phân biệt giữa dữ liệu của nhiều ứng dụng khác nhau (chẳng hạn, dịch vụ Web và dịch vụ thư điện tử) đồng thời chạy trên cùng một máy chủ.

TCP hỗ trợ nhiều giao thức ứng dụng phổ biến nhất trên Internet và các ứng dụng kết quả, trong đó có WWW, thư điện tử và Secure Shell.

Trong bộ giao thức TCP/IP, TCP đóng vai trò là tầng trung gian giữa giao thức IP ở bên dưới và một ứng dụng bên trên. Các ứng dụng thường cần các kết nối kiểu đường ồn đáng tin cậy để có thể liên lạc với nhau, trong khi đó giao thức IP không cung chỉ có thể cung cấp dịch vụ chuyển gói tin không đáng tin cậy. TCP làm nhiệm vụ của tầng giao vận trong mô hình OSI đơn giản của các mạng máy tính.

Các ứng dụng sẽ gửi dữ liệu cần truyền thành các dòng gồm các byte 8-bit tới TCP. TCP thưc hiện việc 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 các đoạn 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 đanng 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. Mô đun TCP tại bên nhận có nhiệm vụ gửi lại "tin báo nhận" (acknowledgement) cho các gói tin đã được 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 checksum (giá trị kiểm tra), 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.

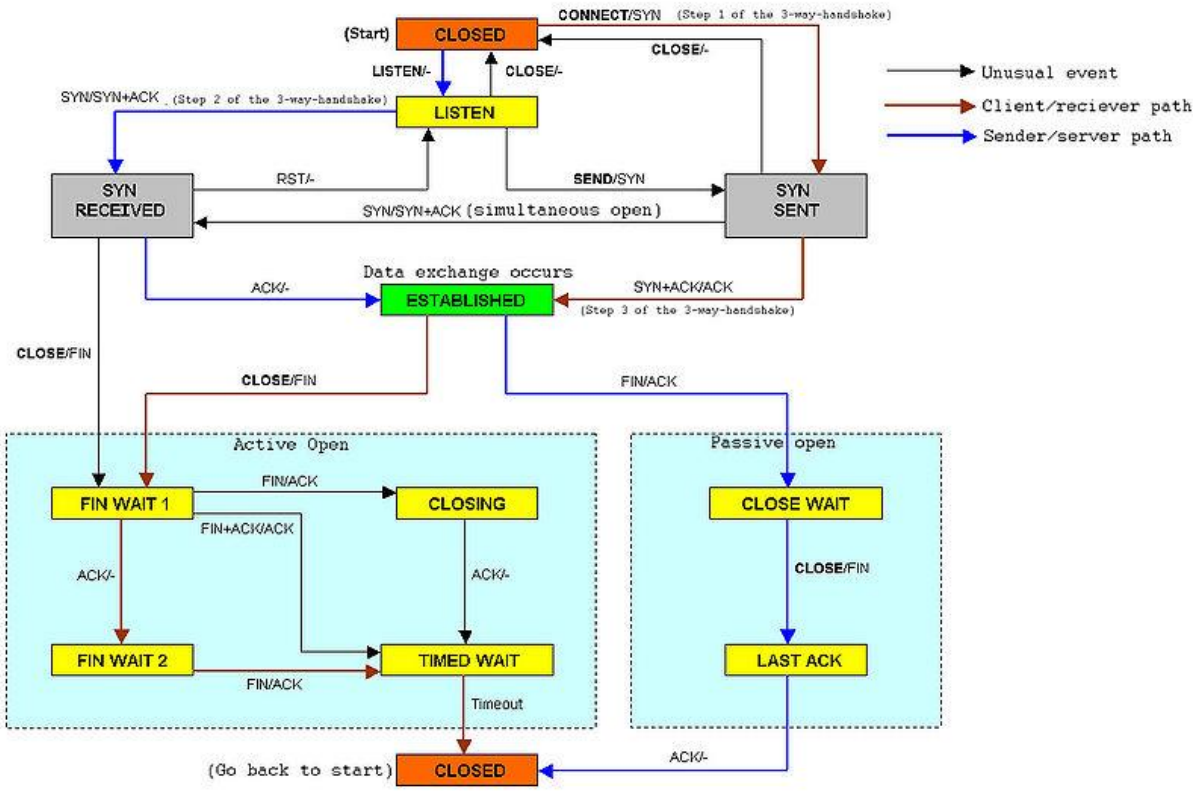
Hoạt động của giao thức

Nếu như giao thức UDP là một giao thực có thể thực hiện việc truyền tin ngay lập tức mà không cần thực hiện việc kết nối thì 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

Trước khi miêu tả các pha này, ta cần lưu ý các trạng thái khác nhau của một socket:

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



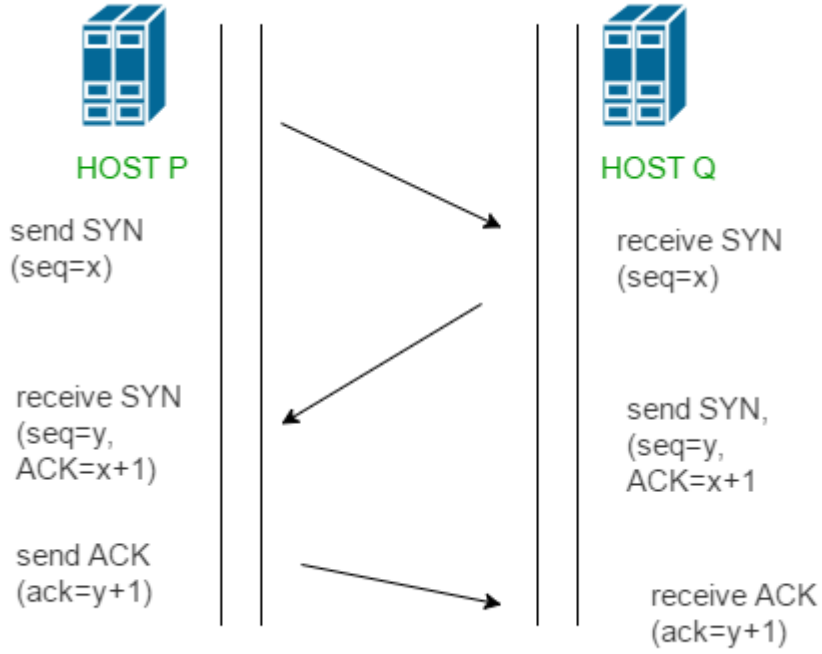
Hình 2.2. Sơ đồ trạng thái của TCP

Thiết lập kết nối

Để có thể tạo một kết nối, TCP sử dụng phương pháp bắt tay 3 bước (3-way handshake). Để client thử kết nối với một server, server phải đăng ký một cổng và mở cổng đó cho các kết nối: đây được gọi là mở bị động. Một khi mở bị động đã được thiết lập thì một client có thể bắt đầu mở chủ động. Để thiết lập một kết nối, quy trình bắt tay 3 bước 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.

Tại thời điểm này, cả client và server đều được xác nhận rằng, một kết nối đã được thiết lập.



Hình 2.3. Thiết lập kết nối TCP

Truyền dữ liệu

Một số đặc điểm cơ bản của TCP để phân biệt với UDP:

- Truyền dữ liệu không lỗi (do có cơ chế sửa lỗi/truyền lại)
- Truyền các gói dữ liệu theo đúng thứ tự
- Truyền lại các gói dữ liệu mất trên đường truyền
- Loại bỏ các gói dữ liệu trùng lặp
- Cơ chế hạn chế tắc nghẽn đường truyền

Ở hai bước đầu tiên trong ba bước bắt tay, hai máy tính trao đổi một số thứ tự gói ban đầu (Initial Sequence Number -ISN). Số này có thể 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 ta có thể sắp xếp lại chúng khi tới máy tính kia bất kể các gói tới nơi theo thứ tự thế nào.

Trên lý thuyết, mỗi byte gửi đi đều có một số thứ tự và khi nhận được thì máy tính nhận gửi lại tin báo nhận (ACK). Trong thực tế thì chỉ có byte dữ liệu đầu tiên được gán số thứ tự trong trường số thứ tự của gói tin và bên nhận sẽ gửi tin báo nhận bằng cách gửi số thứ tự của byte đang chờ.

Số thứ tự và tin báo nhận giải quyết được các vấn đề về lặp gói tin, truyền lại những gói bị hỏng/mất và các gói tin đến sai thứ tự. Để phục vụ mục đích kiểm tra, các gói tin có trường giá trị tổng kiểm (checksum).

Với trình độ hiện tại, kỹ thuật kiểm tra tổng trong TCP không đủ mạnh. Các tầng liên kết dữ liệu với xác suất lỗi bit cao có thể cần được bổ sung các khả năng phát hiện lỗi tốt hơn. Nếu như TCP được thiết kế vào thời điểm hiện tại, nhiều khả năng nó sẽ bao gồm trường kiểm tra độ dư tuần hoàn (cyclic redundancy check - CRC) với độ dài 32 bit. Điểm yếu này một phần được bù đắp bằng CRC hay những kỹ thuật khác tại tầng thứ 2 (trong mô hình 7 lớp OSI) ở bên dưới cả TCP và IP như trong các giao thức điểm-điểm (PPP) hoặc Ethernet. Tuy nhiên điều này cũng không có nghĩa là trường kiểm tra tổng của TCP là không cần thiết: thống kê cho thấy các sai sót do cả phần cứng và phần mềm gây ra giữa các điểm áp dụng kỹ thuật kiểm tra CRC là khá phổ biến và kỹ thuật kiểm tra tổng có khả năng phát hiện phần lớn các lỗi (đơn giản) này.

Điểm cuối cùng là khả năng hạn chế tắc nghẽn.

Tin báo nhận (hoặc không có tin báo nhận) là tín hiệu về tình trạng đường truyền giữa 2 máy tính. Từ đó, hai bên có thể thay đổi tốc độ truyền nhận dữ liệu phù hợp với điều kiện. Vấn đề này thường được đề cập là điều khiển lưu lượng, kiểm soát tắc nghẽn. TCP sử dụng một số cơ chế nhằm đạt được hiệu suất cao và ngăn ngừa khả năng nghẽn mạng. Các cơ chế này bao gồm: cửa sổ trượt (sliding window), thuật toán slow-start, thuật toán tránh nghẽn mạng (congestion avoidance), thuật toán truyền lại và phục hồi nhanh.

Kích thước cửa sổ TCP

Kích thước của cửa sổ là chiều dài (byte) của khối dữ liệu có thể lưu trong bộ đệm của bên nhận. Bên gửi chỉ có thể gửi tối đa lượng thông tin chứa trong cửa sổ này trước khi nhận được tin báo nhận.

Dãn kích thước cửa sổ

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

Do trường điều khiển không thể thay đổi nên người ta sử dụng một hệ số dãn nào đó. Hệ số này được định nghĩa trong tài liệu RFC 1323 có thể sử dụng để tăng kích thước tối đa của cửa sổ từ 65535 byte lên tới 1 gigabyte. Tăng kích thước cửa sổ lớn hơn nữa cũng cần thiết trong TCP Tuning.

Việc tăng kích thước cửa sổ chỉ được dùng trong giao thức bắt tay 3 pha. Giá trị của trường co giãn cửa sổ thể hiện số bit cần được dịch trái đối với trường kích thước cửa sổ. Hệ số dãn có thể thay đổi từ 0 (không dãn) tới 14 (dãn tối đa).

Kết thúc kết nối

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

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

Cấu trúc gói tin

Một gói tin TCP bao gồm 2 phần

- Header (có độ dài 20 byte)
- Dữ liệu

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

- Source port: Số hiệu của cổng tại máy tính gửi.
- Destination port: Số hiệu của cổng tại máy tính nhận.
- Sequence number: Trường này có 2 nhiệm vụ. Nếu cờ SYN bật thì nó 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 đây là số thứ tự của byte đầu tiên.
- Acknowledgement number: Nếu cờ ACK bật thì giá trị của trường chính là số thứ tự gói tin tiếp theo mà bên nhận cần.
- Data offset: Trường có độ dài 4 bit quy định độ dài của phần header (tính theo đơn vị từ 32 bit). Phần header có độ dài tối thiểu là 5 từ (160 bit) và tối đa là 15 từ (480 bit).
- Reserved: Dành cho tương lai và có giá trị là 0.
- Flags (hay 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 trường báo nhận (ACK)
- Checksum: 16 bit kiểm tra cho cả phần header và dữ liệu

- Urgent pointer: Nếu cờ URG bật thì giá trị trường này chính là số từ 16 bit mà số thứ tự gói tin (sequence number) cần dịch trái.
- Options: Đây là trường tùy chọn. Nếu có thì độ dài là bội số của 32 bit.

Trường cuối cùng không thuộc về header. Giá trị của trường này là thông tin dành cho các tầng trên (trong mô hình 7 lớp OSI). Thông tin về giao thức của tầng trên không được chỉ rõ trong phần header mà phụ thuộc vào cổng được chọn.

+	Bit 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. Header của gói tin TCP

2.1.3. User Datagram Protocol

User Datagram Protocol là một trong những giao thức cốt lõi của giao thức TCP/IP. Dùng UDP, chương trình trên mạng máy tính có thể gửi những dữ liệu ngắn được gọi là datagram tới máy khác. UDP không cung cấp sự tin cậy và thứ tự truyền nhận như cách mà TCP làm; các gói dữ liệu có thể đến không đúng thứ tự hoặc bị mất mà không có thông báo. Tuy nhiên UDP nhanh và hiệu quả hơn khi dữ liệu có kích thước nhỏ và yêu cầu khắt khe về thời gian. Do bản chất không trạng thái của nó nên 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 DNS (Domain Name System), ứng dụng streaming media, Voice over IP, Trivial File Transfer Protocol (TFTP), và game trực tuyến.

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 cung cấp một giao diện rất đơn giản giữa tầng mạng bên dưới và tầng phiên làm việc hoặc tầng ứng dụng phía trên.

UDP không đảm bảo cho các tầng phía trên rằng thông điệp đã được gửi đi và phía gửi cũng không có trạng thái thông điệp UDP một khi đã được gửi (Vì lý do này đôi khi UDP còn được gọi là Unreliable Datagram Protocol).

UDP chỉ thêm các thông tin multiplexing và giao dịch. Các loại thông tin dùng cho việc truyền tin 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. Header của gói tin UDP

Source port: Trường này xác định cổng của phía gửi thông tin và có ý nghĩa nếu muốn nhận thông tin phản hồi từ phía nhận. Bằng 0 nếu không được dùng đến.

Destination port: Trường xác định cổng nhận thông tin, và trường này là bắt buộc.

Length: Trường có độ dài 16 bit xác định chiều dài của toàn bộ datagram: phần header và dữ liệu. Chiều dài tối thiểu là 8 byte khi gói tin không có dữ liệu, chỉ có header.

Checksum: Trường checksum 16 bit dùng cho việc kiểm tra lỗi của phần header và dữ liệu. Phương pháp tính checksum được định nghĩa trong RFC 768.

Do thiếu tính tin cậy, các ứng dụng UDP nói chung phải chấp nhận mất mát, lỗi hoặc trùng dữ liệu. Một số ứng dụng như TFTP có nhu cầu phải thêm những kỹ thuật làm tin cậy cơ bản vào tầng ứng dụng. Hầu hết các ứng dụng UDP không cần những kỹ thuật làm tin cậy này và đôi khi nó bị bỏ đi. Streaming media, game trực tuyến và voice over IP (VoIP) là ví dụ cho các ứng dụng thường dùng UDP. Nếu một ứng dụng đòi hỏi mức độ cao hơn về tính tin cậy, những giao thức như TCP hoặc mã erasure có thể được sử dụng để thay thế.

Thiếu những cơ chế kiểm soát tắc nghẽn và kiểm soát luồng, các kỹ thuật dựa trên mạng là cần thiết để giảm nguy hiệu ứng cơ tắc nghẽn dây chuyền do không kiểm soát, tỷ lệ tải UDP cao. Nói cách khác, vì phía gửi gói UDP không thể phát hiện tắc nghẽn, các thành phần dựa trên mạng như router dùng hàng đợi gói (packet queueing) hoặc kỹ thuật bỏ gói như là những công cụ để giảm tải của UDP. Giao thức Datagram Congestion Control Protocol (DCCP) được thiết kế như một giải pháp cho vấn đề bằng cách thêm

hành vi kiểm soát tắc nghẽn cho thiết bị đầu cuối cho các dòng dữ liệu UDP như streaming media.

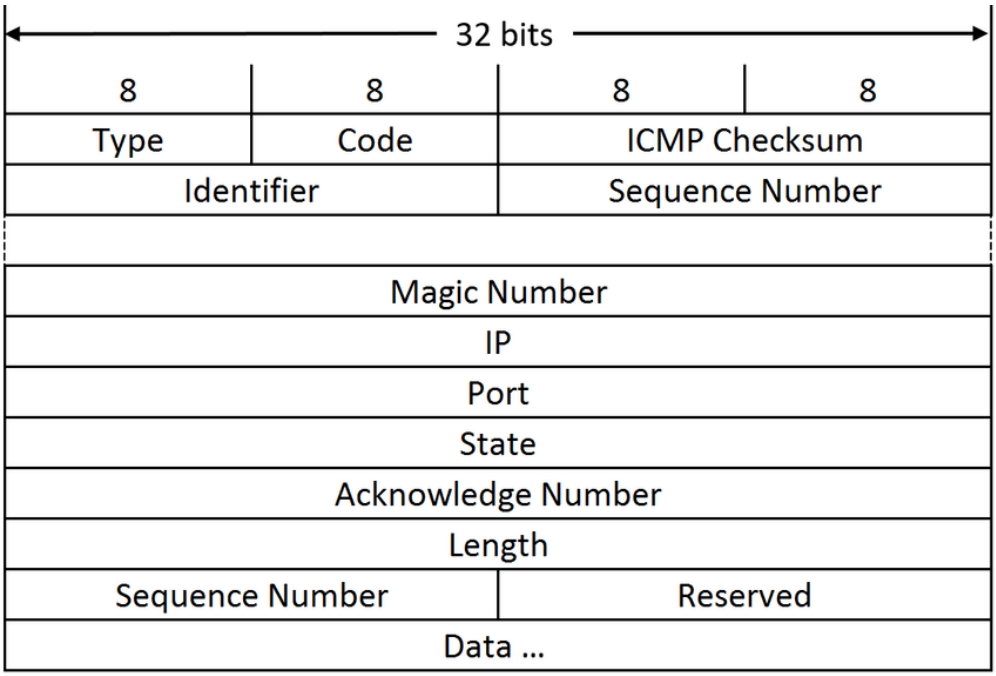
Mặc dù tổng lượng lưu thông của UDP trên mạng thường chỉ vài phần trăm, nhưng có nhiều ứng dụng quan trọng dùng UDP, bao gồm DNS, SNMP, DHCP và RIP.

2.1.4. Internet Control Message Protocol

Internet Control Message Protocol, là một giao thức của Internet Protocol. Giao thức này được các thiết bị mạng như router dùng để gửi đi các thông báo nhằm tìm ra một địa chỉ có tồn tại hay không. ICMP cũng có thể được sử dụng để chuyển tiếp các thông điệp truy vấn. Giao thức này được đánh số 1. ICMP khác với các giao thức vận chuyển như TCP và UDP ở chỗ nó không thường được sử dụng để trao đổi dữ liệu giữa các hệ thống, cũng không thường xuyên được sử dụng bởi các ứng dụng mạng của người dùng cuối (với ngoại lệ của một số công cụ chẩn đoán như ping và traceroute).

Nó tuân theo các nguyên tắc sau đây:

- ICMP sử dụng IP để làm cơ sở thông tin liên lạc bằng cách giải thích chính nó như là một lớp giao thức cao hơn, thông điệp ICMP được đóng gói trong các gói tin IP.
- ICMP nhận ra một số tình trạng lỗi, nhưng không làm IP trở thành một giao thức đáng tin cậy.
- ICMP phân tích sai sót trong mỗi gói IP, trừ các đối tượng mà mang một thông điệp ICMP.
- Thông điệp ICMP không trả lời các địa chỉ multicast hoặc broadcast.
- Thông điệp ICMP chỉ trả lời một địa chỉ IP định rõ.



Hình 2.6. Header của ICMP

2.2. Tổng quan 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 đánh cắp hoặc làm tổn hại đến phần cứng, phần mềm và dữ liệu, cũng như các nguyên nhân gây ra sự gián đoạn, chuyển lệch hướng của các dịch vụ trong hệ thống mạng [1]s.

An ninh mạng là bảo vệ các hệ thống mạng, máy tính, chương trình và dữ liệu khỏi những cuộc tấn công mạng. Tội phạm mạng có thể triển khai một loạt các cuộc tấn công nhắm vào các cá nhân hoặc doanh nghiệp; có thể kể đến như truy cập trái phép, làm thay đổi hoặc xóa bỏ dữ liệu; tống tiền; can thiệp vào các quy trình.

An ninh mạng máy tính bao gồm việc kiểm soát truy cập vật lý đến phần cứng, cũng như bảo vệ chống lại tác hại có thể xảy ra qua truy cập mạng máy tính, cơ sở dữ liệu và việc lợi dụng lỗ hổng phần mềm. Do sai lầm của những người điều hành, dù cố ý hoặc do bất cẩn, kẻ tấn công có thể sử dụng các phương pháp phi kỹ thuật để vượt qua các thủ tục an ninh.

An ninh mạng hoạt động thông qua một cơ sở hạ tầng chặt chẽ, 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 khỏi những kẻ xâm nhập trái phép.

- An ninh mạng: Là một tập hợp con của bảo mật công nghệ thông tin. An ninh mạng thực hiện nhiệm vụ đảm bảo an toàn cho dữ liệu số trên các mạng lưới, máy tính và thiết bị cá nhân khỏi sự truy cập, tấn công và phá hủy bất hợp pháp.
- An ninh máy tính: Là một tập hợp 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 từ máy tính cá nhân hoặc các thiết bị khác đến hệ thống mạng lưới thông tin. An ninh máy tính thực hiện chức năng bảo vệ cơ sở hạ tầng công nghệ thông tin và phòng chống việc các dữ liệu bị chặn, bị thay đổi hoặc đánh cắp bởi tội phạm mạng.

Lĩnh vực này dần trở nên quan trọng do sự phụ thuộc ngày càng nhiều vào các hệ thống máy tính và Internet trên khắp thế giới, cũng như sự phụ thuộc vào hệ thống mạng không dây như Bluetooth, Wi-Fi, cùng với sự phát triển của các thiết bị "thông minh", bao gồm điện thoại thông minh, TV và các thiết bị khác kết nối vào hệ thống Internet of Things.

2.2.1. Mục tiêu của an ninh mạng

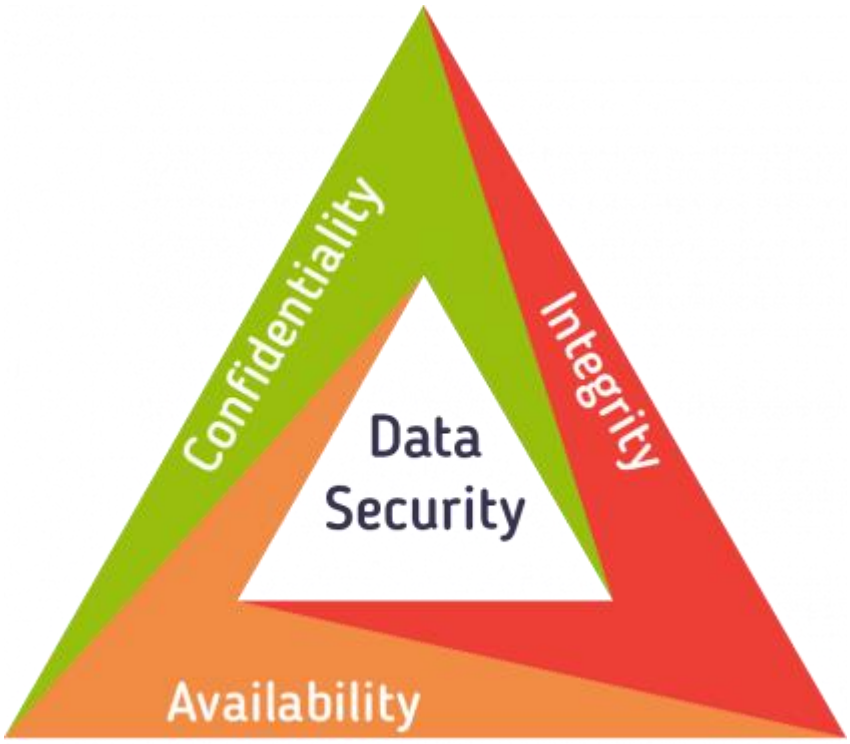
Mục tiêu của an ninh mạng là bảo vệ thông tin khỏi bị đánh cắp, xâm phạm hoặc bị tấn công. Độ bảo mật an ninh mạng có thể được đo lường bằng ít nhất một trong ba mục tiêu sau:

- Bảo vệ tính bảo mật của dữ liệu.
- Bảo toàn tính toàn vẹn của dữ liệu.
- Thúc đẩy sự sẵn có của dữ liệu cho người dùng được ủy quyền.

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. Tam giác CIA là một mô hình bảo mật được thiết kế để hướng dẫn thực thi các chính sách bảo mật thông tin trong khuôn khổ nội bộ một tổ chức hoặc một công ty.

Các tiêu chí của CIA được hầu hết các tổ chức sử dụng khi bắt đầu cài đặt một ứng dụng mới, tạo lập cơ sở dữ liệu hoặc khi muốn đảm bảo quyền truy cập vào một số dữ liệu. Để dữ liệu được bảo mật hoàn toàn, tất cả các tiêu chí này phải có hiệu lực, đây là những chính sách bảo mật mà mọi thành phần cấu tạo nên nó đều phải cùng nhau hoạt động, và do đó, có thể sẽ xảy ra sai sót khi bỏ quên một trong những thành phần của CIA.

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

Tính bảo mật (Confidentiality)

Bảo mật có thể coi là tương đương với quyền riêng tư và việc tránh tiết lộ thông tin trái phép. Liên quan đến việc bảo vệ dữ liệu, bảo mật cung cấp quyền truy cập cho những người được phép và ngăn chặn người khác tiếp xúc với bất kỳ thông tin nào về nội dung của chủ sở hữu. 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. Mã hóa dữ liệu là một ví dụ điển hình để đảm bảo tính bảo mật.

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. Sử dụng khóa bí mật (khóa mã hóa) để dữ liệu được chuyển đổi, chỉ có thể được đọc bằng cách sử dụng một khóa bí mật khác (khóa giải mã). Công cụ này nhằm bảo vệ những dữ liệu nhạy cảm, bằng cách mã hóa và chuyển đổi dữ liệu thành một kiểu dữ liệu không thể đọc được, dữ liệu này chỉ có thể được đọc bằng cách giải mã nó. Khóa bất đối xứng (asymmetric-key) và khóa đối xứng (symmetric-key) là hai loại mã hóa chính phổ biến nhất.

- 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. Trong các hệ thống kiểm soát quyền truy cập, người dùng cần xuất trình thông tin đăng nhập trước khi có thể được cấp phép. Trong các hệ thống vận hành vật lý, các thông tin đăng nhập này có thể tồn tại dưới nhiều dạng, nhưng với các thông tin không thể được chuyển giao sẽ cung cấp tính bảo mật cao nhất.
- Xác thực (Authentication): Xác thực là quá trình sxác nhận danh tính hoặc vai trò của người dùng. Công cụ này có thể được thực hiện theo một số cách khác nhau, nhưng thường dựa trên sự kết hợp giữa: một thứ gì đó mà cá nhân sở hữu (như thẻ thông minh hoặc khóa radio để lưu trữ các khóa bí mật), một thứ gì đó mà cá nhân biết (như mật khẩu) hoặc một thứ gì đó dùng để nhận dạng cá nhân (như dấu vân tay). Xác thực đóng vai trò cấp thiết đối với mọi tổ chức, vì công cụ này cho phép họ giữ an toàn cho mạng lưới thông tin của mình bằng cách chỉ cho phép người dùng được xác thực truy cập vào các tài nguyên dưới sự bảo vệ, giám sát của nó. Những tài nguyên này có thể bao gồm các hệ thống máy tính, mạng, cơ sở dữ liệu, website và các ứng dụng hoặc dịch vụ dựa trên mạng lưới khác.
- 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. Ủy quyền thường được đi trước xác thực để xác minh danh tính người dùng. Quản trị viên hệ thống thường là người chỉ định cấp phép hoặc từ chối quyền truy cập đối với cá nhân khi muốn đăng nhập vào hệ thống và tiếp cận thông tin dữ liệu.
- Bảo mật vậy 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. Công cụ này bảo vệ các tài sản nêu trên khỏi các mối đe dọa vật lý như: trộm cắp, phá hoại, hỏa hoạn và thiên tai.

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. Sao lưu cũng được sử dụng để tạo các bản sao phục vụ cho các mục đích lưu lại lịch sử dữ liệu.
- Tổng kiểm tra (Checksums): Tổng kiểm tra là một giá trị số được sử dụng để xác minh tính toàn vẹn của dữ liệu được truyền đi. Nói cách khác, đó là sự tính toán của một hàm chuyển nội dung của dữ liệu thành một giá trị số. 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 phục vụ cho mục đích riêng của nó và thông tin phải luôn 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

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.

Nạn nhân của tấn công mạng:

- Tấn công mạng không mục tiêu: Trong các cuộc tấn công không mục tiêu, đối tượng mà tội phạm mạng nhắm đến là số lượng thiết bị, dịch vụ hoặc

người dùng bị ảnh hưởng càng tốt. Kẻ tấn công không quan tâm ai là nạn nhân vì luôn có một số lượng lớn máy móc hoặc dịch vụ tồn tại lỗ hổng. Để thực hiện các cuộc tấn công này, kẻ tấn công 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. Lý giải cho những nguyên do đằng sau cuộc tấn công này, tội phạm mạng, hoặc là có chủ đích rõ ràng với cá nhân/tổ chức; hoặc là được trả tiền để thực hiện tấn công vào cá nhân/tổ chức đó. Việc chuẩn bị cho một cuộc tấn công mạng với mục tiêu xác định có thể mất nhiều tháng để tìm ra cách tốt nhất tác động đến cá nhân/tổ chức. 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.

Nhìn chung, nạn nhân của tấn công mạng có thể là một cá nhân, doanh nghiệp, các tổ chức chính phủ hoặc phi chính phủ, cơ quan nhà nước, thậm chí, đối tượng có thể là cả một quốc gia. Tuy nhiên, đối tượng phổ biến nhất của các cuộc tấn công mạng là các doanh nghiệp.

Mục đích tấn công mạng

Bên cạnh những mục đích phổ biến như trục lợi phi pháp, tống tiền, hiện quảng cáo kiếm tiền, thì còn tồn tại một số mục đích khác phức tạp và nguy hiểm hơn: cạnh tranh không lành mạnh giữa các doanh nghiệp, tấn công an ninh hoặc kinh tế của một quốc gia, tấn công đánh sập một tổ chức tôn giáo, v.v. Ngoài ra, một số tội phạm mạng tấn công mạng chỉ để mua vui, thử sức, hoặc tò mò muốn khám phá các vấn đề về an ninh mạng.

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

Lỗ hổng bảo mật là một điểm yếu của hệ thống trong quá trình thiết kế, thi công và quản trị. Phần lớn các lỗ hổng bảo mật được đã phát hiện ngày nay đều được ghi lại trong cơ sở dữ liệu Common Vulnerabilities and Exposures (CVE). Một lỗ hổng bị khai thác là một lỗ hổng mà đã bị lợi dụng để thực hiện hoạt động tấn công ít nhất một lần hoặc đã bị khai thác (exploit).

Để đảm bảo một hệ thống máy tính, điều quan trọng là phải hiểu các cuộc tấn công có thể được thực hiện chống lại nó, và các mối đe dọa thường được xếp vào một trong các mục dưới đây:

Tấn công bằng phần mềm độc hại (Malware Attack)

Malware là phần mềm độc hại, được kết hợp giữa hai từ "malicious" và "software". Đây là một trong những hình thức tấn công mạng phổ biến nhất. Tội phạm mạng tạo ra malware với mục đích làm hư hỏng máy tính của người dùng. Thông thường, tội phạm mạng sẽ tấn công người dùng thông qua các lỗ hổng bảo mật, dụ dỗ người dùng click vào một đường link đính kèm trong thư điện tử hoặc tải các tệp tin được ngụy trang để phần mềm độc hại tự động cài đặt vào máy tính. Malware thường được sử dụng nhằm phục vụ cho mục đích kiếm tiền hoặc tham gia vào các cuộc tấn công mạng có động cơ.

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 với nhiều mức độ phá hủy khác nhau.
- Trojan Horse: Khác với virus, phần mềm này không có chức năng tự sao chép nhưng lại có khả năng phá hoại tương đương. Trojan Horse sẽ được ngụy trang thành các phần mềm hợp pháp, tội phạm mạng lừa người dùng cài đặt Trojan Horse vào máy tính của họ, nơi 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 mà không để lại "di chứng", bí mật lưu lại những gì người dùng làm, dựa vào đó, tội phạm mạng có thể sử dụng các thông tin này để đem đến bất lợi cho chủ sở hữu chúng.
- 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 phát tán, cài đặt các phần mềm độc hại khác.
- Botnets: Mạng lưới các máy tính bị nhiễm phần mềm độc hại được tội phạm mạng sử dụng để thực hiện các hành động mà không có sự cho phép của người dùng.

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 nhắm đến việc đánh cắp dữ liệu cá nhân nhạy cảm như thông tin thẻ tín dụng, mật khẩu, tài khoản đăng nhập hoặc cài đặt các phần mềm độc hại vào máy tính nạn nhân. Phishing thường được thực hiện bằng cách sử dụng thư điện tử (email) hoặc tin nhắn.

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

Tấn công trung gian (MitM), hay còn gọi là tấn công nghe lén, xảy ra khi kẻ tấn công mạng 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. Thông thường, khi sử dụng Wi-Fi công cộng thiếu bảo mật, kẻ tấn công có thể tự "chen" vào giữa thiết bị của người truy cập và mạng Wi-Fi đó, tất cả dữ liệu cá nhân mà người dùng gửi đi sẽ bị kẻ tấn công thu thập.

Tấn công từ chối dịch vụ (Denial of Service)

Các cuộc tấn công từ chối dịch vụ (DoS) được thiết kế để làm cho tài nguyên mạng không sẵn sàng để phục vụ cho người dùng. Kẻ tấn công có thể từ chối dịch vụ cho từng nạn nhân, chẳng hạn như cố tình nhập sai mật khẩu đủ lần liên tục để khiến tài khoản nạn nhân bị khóa hoặc làm quá tải máy tính hoặc mạng và chặn tất cả người dùng cùng một lúc. Mặc dù một cuộc tấn công mạng từ một địa chỉ IP duy nhất có thể bị chặn bằng cách sử dụng tường lửa, nhiều hình thức tấn công từ chối dịch vụ phân tán (DDoS) là có thể, trong đó cuộc tấn công đến từ một số lượng lớn máy tính. Các cuộc tấn công như vậy có thể bắt nguồn từ các máy tính zombie của botnet, nhưng một loạt các kỹ thuật khác có thể bao gồm các cuộc tấn công phản xạ và khuếch đại, trong đó các hệ thống bình thường bị lừa gửi dữ liệu đến máy nạn nhân.

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 máy chủ 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, kẻ tấn công có thể tấn công bằng cách chèn một đoạn mã truy vấn vào ô “Tìm kiếm” là đã có thể dễ dàng tấn công những website với mức bảo mật yếu.

Tấn công "cửa hậu" (Backdoor Attack)

Trong một hệ thống máy tính, Backdoor ("cửa hậu") là một phương pháp bí mật vượt qua thủ tục chứng thực người dùng thông thường hoặc để giữ đường truy nhập từ xa tới một máy tính, trong khi cố gắng không bị phát hiện bởi việc giám sát thông thường. Chúng tồn tại vì một số lý do, bao gồm từ thiết kế ban đầu hoặc từ cấu hình kém. Chúng có thể đã được thêm vào bởi một nhóm có thẩm quyền để cho phép một số truy cập hợp pháp, hoặc bởi những kẻ tấn công vì lý do độc hại; nhưng bất kể động cơ đưa tới sự tồn tại của chúng, chúng tạo ra một lỗ hổng.

Khai thác lỗ hổng (Zero-day Exploits)

Lỗ hổng Zero-day (hay còn gọi là 0-day) là thuật ngữ để chỉ những lỗ hổng phần mềm hoặc phần cứng chưa được biết đến và chưa được khắc phục. Các hacker có thể

tận dụng lỗ hổng này để tấn công xâm nhập vào hệ thống máy tính của doanh nghiệp, tổ chức nhằm đánh cắp hoặc thay đổi dữ liệu.

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 phân biệt được những tấn công từ bên trong và từ bên ngoài [2].

IDS phát hiện tấn công dựa trên các dấu hiệu khác thường (tương tự như các 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.

Các tính năng quan trọng nhất của IDS là:

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

Phân loại IDS:

- Network-based Intrusion Detection System: Hệ thống phát hiện xâm nhập mạng. Theo dõi hoạt động bất thường trên trong toàn mạng. NIDS có thể là phần mềm triển khai trên server hoặc dạng thiết bị tích hợp appliance.
- Host-based Intrusion Detection System: Hệ thống phát hiện xâm nhập host. Theo dõi các hoạt động bất thường trên các host riêng biệt. HIDS được cài đặt trực tiếp trên các máy cần theo dõi.

IDS là một trong những thành phần quan trọng trong các giải pháp bảo vệ hệ thống. Khi triển khai có thể giúp hệ thống:

- Theo dõi các hoạt động bất thường đối với hệ thống.
- Xác định ai đang tác động đến hệ thống và cách thức như thế nào.
- Xác định vị trí các hoạt động động xâm nhập xảy ra nàos mạng.

Ưu điểm của IDS:

- Cung cấp một cách nhìn toàn diện về toàn bộ lưu lượng mạng.
- Giúp kiểm tra các sự cố xảy ra với hệ thống mạng.
- Sử dụng để thu thập bằng chứng cho điều tra và ứng cứu sự cố.

Hạn chế của IDS:

- Có thể gây ra tình trạng báo động nhầm nếu cấu hình không hợp lý.
- Khả năng phân tích lưu lượng bị mã hóa tương đối thấp.

- Chi phí triển khai và vận hành hệ thống tương đối lớn.

Hệ thống luật của IDS

Tập luật là thành phần quan trọng nhất của một hệ thống phát hiện xâm nhập. Đây là tập sẽ định ra dấu hiệu để so sánh, đối chiếu với dữ liệu ở đầu vào. Thông thường, tập luật bao gồm rất nhiều luật, mỗi luật sẽ gồm 2 thành phần cơ bản: Rule Header và Rule Options.

Rule header bao gồm các thông tin sau:

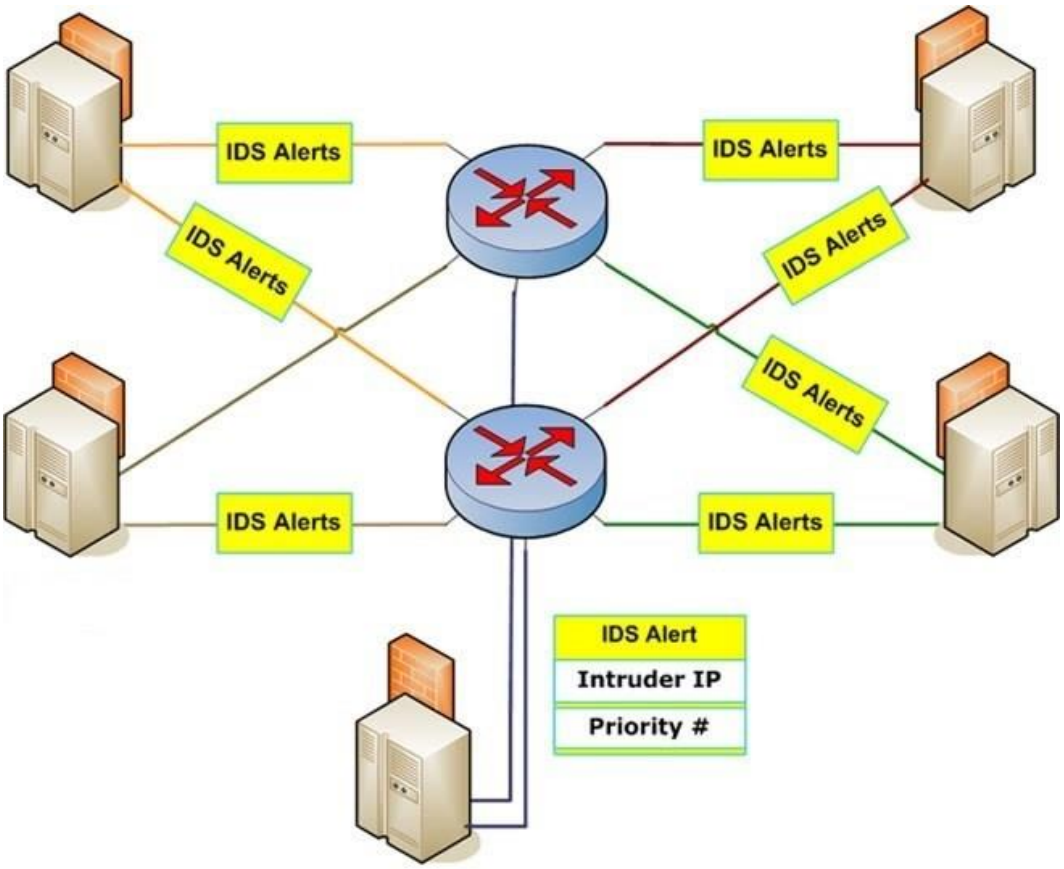
- Rule Action: Cho biết các hoạt động sẽ được thực thi khi "khớp" luật (alert, log, pass, active, dynamic, drop…).
- Protocol: Cho biết giao thức sẽ kiểm tra (TCP, UDP, ICMP, IP…)
- IP address: Cho biết thông tin về địa chỉ ip.
- Port number: Cho biết thông tin về cổng.
- Direction: Cho biết hướng của dữ liệu mà được so khớp.

Rule options chia làm 4 danh mục:

- General: cung cấp 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: cung cấp các phương pháp thực thi kế tiếp (logto, session, tag…).

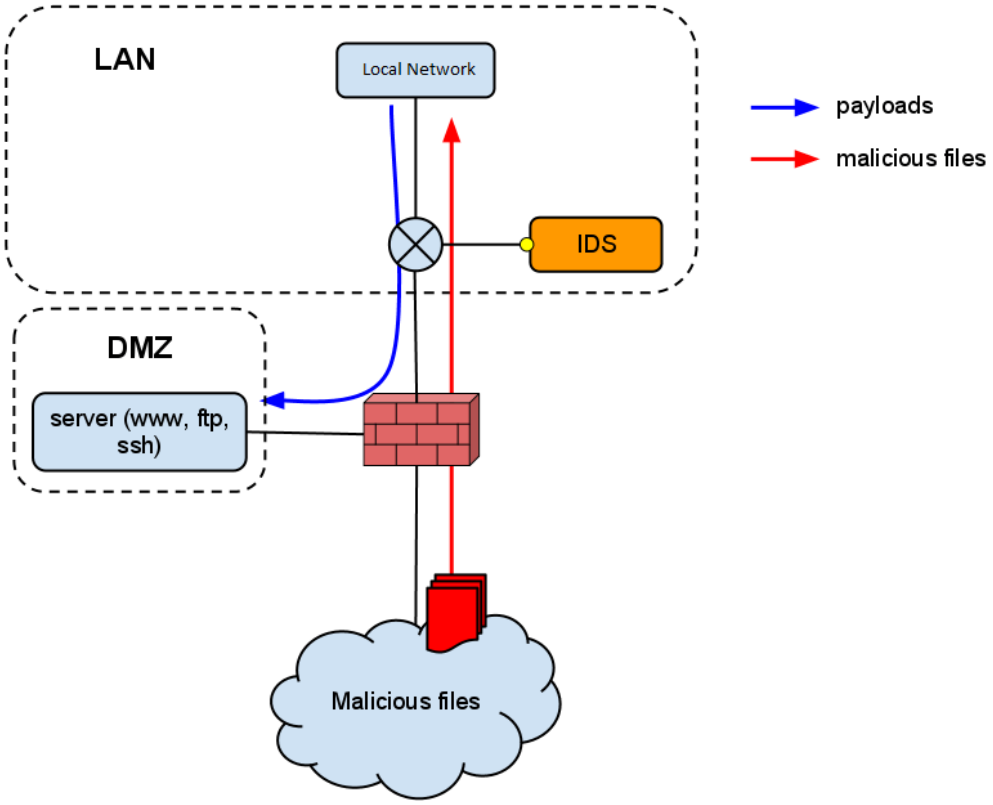
Thiết kế IDS trong mô hình mạng doanh nghiệp

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.



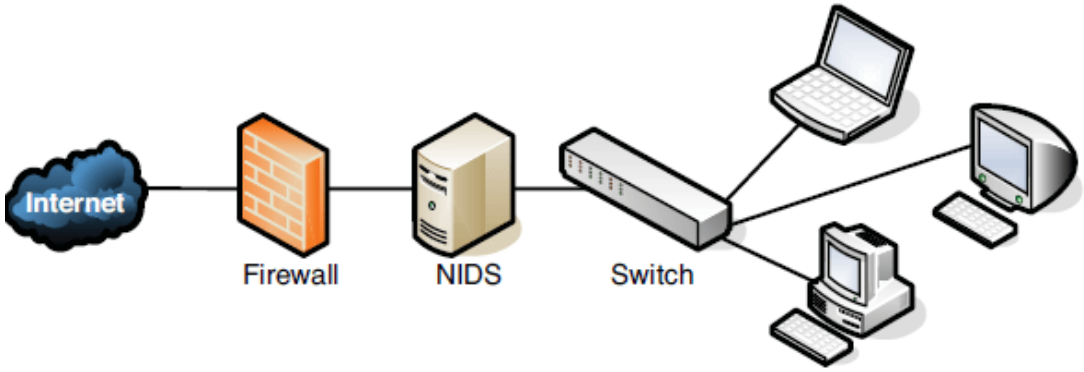
Hình 2.8. IDS được đặt giữa router và firewall

Khi đặt trong trường hợp này, 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 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 đặt trong trường hợp này, IDS sẽ theo dõi tất cả lưu lượng vào/ra trong miền DMZ.



Hình 2.10. IDS được đặt sau firewall

Khi đặt trong trường hợp này, IDS sẽ theo dõi tất cả lưu lượng trao đổi phía sau firewall như:

- Dữ liệu trao đổi trong LAN.
- Dữ liệu từ LAN vào/ra DMZ và ngược lại.

2.3.2. Intrusion Prevention System

Intrusion Prevention Systems là hệ thống theo dõi, ngăn ngừa kịp thời các hoạt động xâm nhập không mong muốn. Chức năng chính của IPS là xác định các hoạt động nguy hại, lưu giữ các thông tin này. Sau đó kết hợp với firewall để dừng ngay các hoạt động này, và cuối cùng đưa ra các báo cáo chi tiết về các hoạt động xâm nhập trái phép trên. Hệ thống IPS được xem là trường hợp mở rộng của hệ thống IDS, cách thức hoạt động cũng như đặc điểm của 2 hệ thống này tương tự nhau. Điểm khác nhau duy nhất là hệ thống IPS ngoài khả năng theo dõi, giám sát thì còn có chức năng ngăn chặn kịp thời các hoạt động nguy hại đối với hệ thống. Hệ thống IPS sử dụng tập luật tương tự như hệ thống IDS [3].

Phân loại IPS:

- Network-based Intrusion Prevention: Hệ thống ngăn ngừa xâm nhập mạng, thường được triển khai trước hoặc sau firewall. Khi triển khai IPS trước firewall là có thể bảo vệ được toàn bộ hệ thống bên trong kể cả firewall, vùng DMZ. Có thể giảm thiểu nguy cơ bị tấn công từ chối dịch vụ đối với firewall. Khi triển khai IPS sau firewall có thể phòng tránh 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 Prevention: Hệ thống ngăn ngừa xâm nhập host, thường được triển khai với mục đích phát hiện và ngăn chặn kịp thời các hoạt động thâm nhập trên các host. Để có thể ngăn chặn ngay các tấn công, HIPS sử dụng công nghệ tương tự như các giải pháp antivirus. 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 tham gia trong kiến trúc mạng đều có chức năng, điểm mạnh, điểm yếu khác nhau. Sử dụng, khai thác 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 hệ thống:

- Theo dõi các hoạt động bất thường đối với hệ thống.
- Xác định ai đang tác động đến hệ thống và cách thức như thế nào, các hoạt động xâm nhập xảy ra tại vị trí nào trong cấu trúc mạng.
- Tương tác với hệ thống firewall để ngăn chặn kịp thời các hoạt động thâm nhập hệ thống.

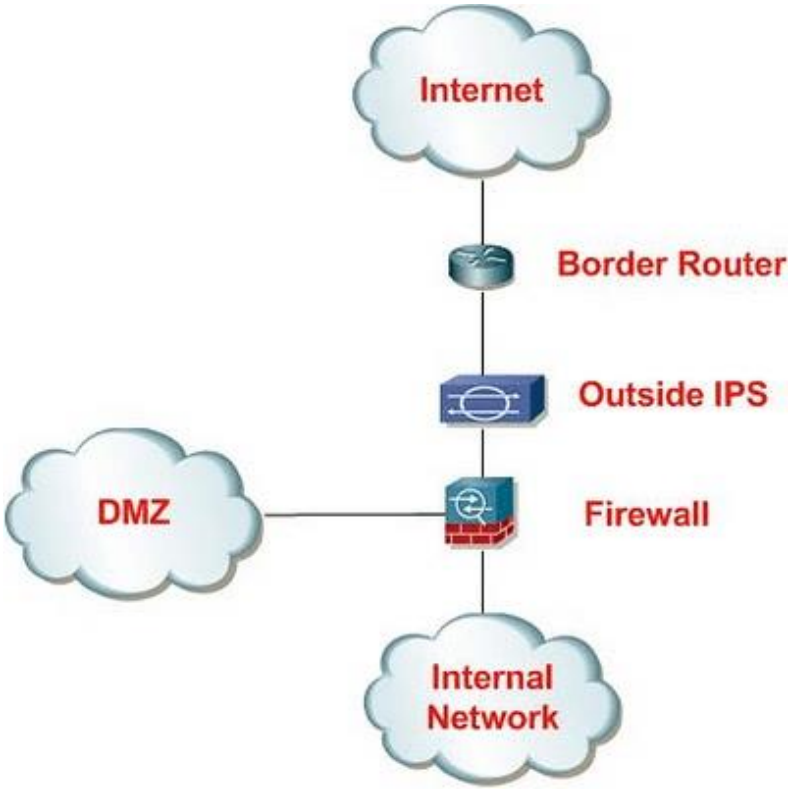
Ưu điểm của IPS:

- Cung cấp giải pháp bảo vệ toàn diện hơn đối với tài nguyên hệ thống.

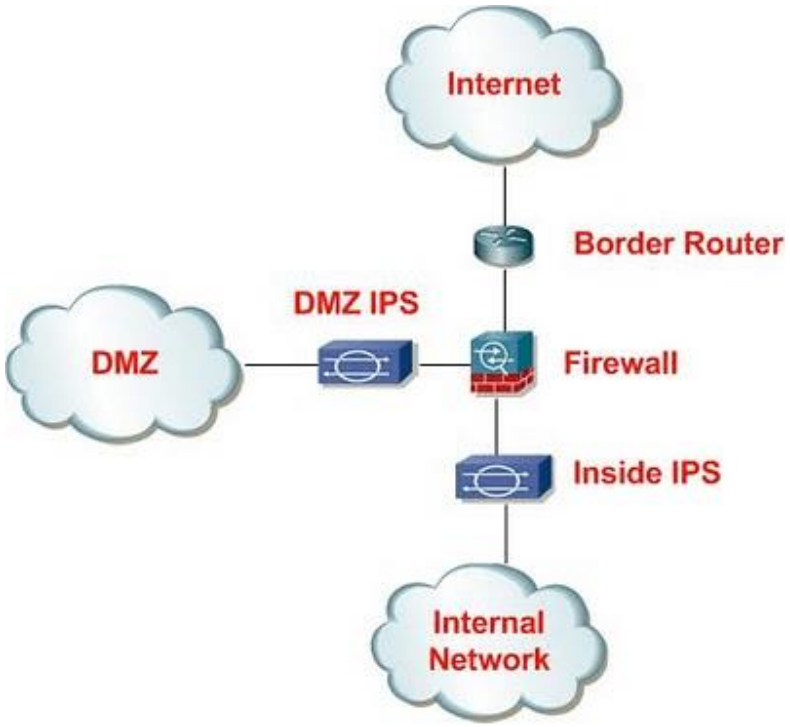
- Ngăn chặn kịp thời các tấn công đã biết hoặc chưa được biết.

Hạn chế của IPS:

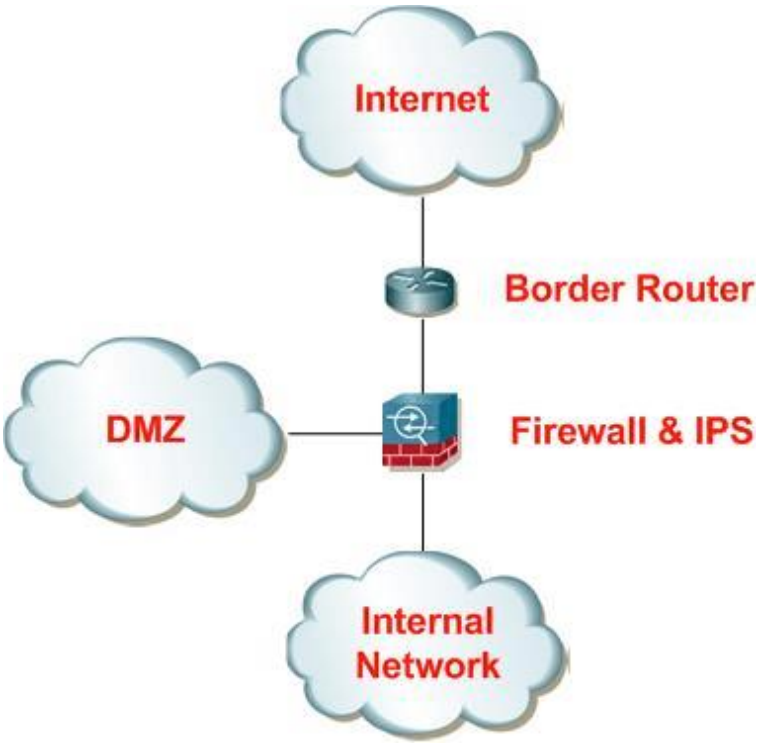
- Có thể gây ra tình trạng phát hiện nhầm, có thể không cho phép các truy cập hợp lệ tới hệ thống.



Hình 2.11. IPS được đặt trước firewall



Hình 2.12. IPS được đặt giữa firewall và miền DMZ



Hình 2.13. IPS là một module trong giải pháp UTM

2.4. Iptables và Netfilter

Iptables là phần mềm tường lửa cơ bản được sử dụng phổ biến nhất trong Linux. Tường lửa 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 bất kỳ quyết định định tuyến nào được đưa ra.
- 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 được định tuyến đã diễn ra và ngay trước khi 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

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

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. Điều này thường được sử dụng để định tuyến các gói đến các mạng khi không thể truy cập trực tiếp.
Mangle Table	Bảng Mangle được sử dụng để thay đổi các tiêu đề IP của gói 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. Bảng raw có chức năng được xác định rất hẹp. Mục đích duy nhất của nó 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 trên cơ sở 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 sẽ khiến chains của bảng bị bỏ qua trong quá trình xử lý. Chẳng hạn, chỉ gói đầu tiên trong kết nối sẽ được đánh giá theo các quy tắc NAT. Bất kỳ quyết định nat nào được thực hiện cho gói đầu tiên sẽ được áp dụng cho tất cả các gói tiếp theo trong 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

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 để tạo ra các bộ quy tắc khá phức tạp để phân biệt giữa các gói tin khác nhau.

2.4.7. Targets

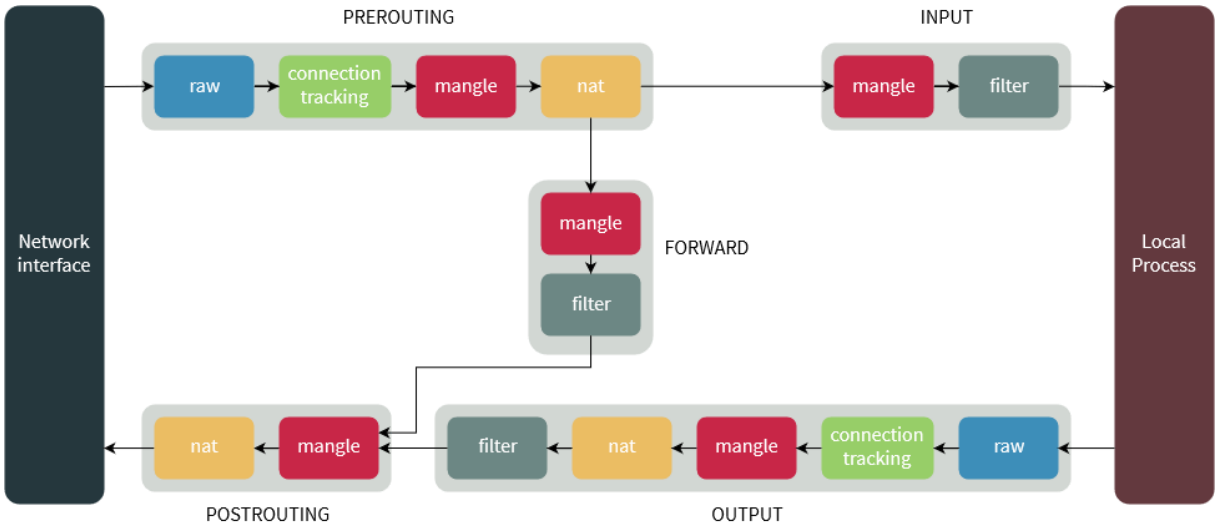
Target là một hành động sẽ được trigger ngay khi các tiêu chí của rule khớp(matched) hoàn toàn. Target được chia ra làm 2 nhóm sau:

- 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 cung cấp, hook có thể thả gói hoặc cho phép gói tiếp tục 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: Các mục tiêu không chấm dứt thực hiện một 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

Mục tiêu nhảy là các hành động dẫn đến việc chuyển 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 nhãn NEW cho kết nối này. 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. Với kết nối TCP, nó 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. Nó 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. Nó 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 các quy tắc nhắm mục tiêu 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

3.1.1. Ý tưởng

Chương trình sẽ không tự thực hiện việc bắt gói tin mà sẽ lấy cái gói tin ra từ nfqueue (cấu hình để Iptables đẩy các gói tin vào fqueue).

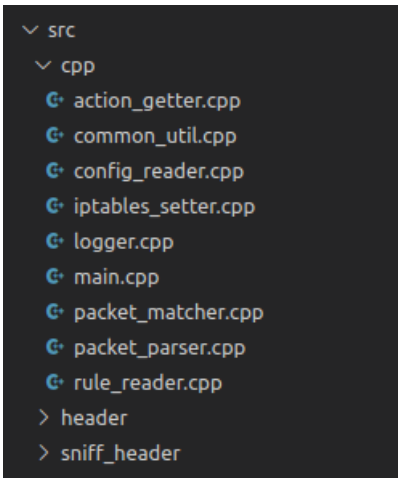
Các gói tin lấy ra sẽ được ép kiểu thành một struct có các trường tương tự như header của gói tin IPv4, từ đó lấy ra các thông tin như giao thức, địa chỉ nguồn, địa chỉ đích, tiếp tục làm vậy với các struct đại diện cho header của TCP và UDP để lấy ra cổng nguồn và cổng đích.

Sau khi có các thông tin trên, sẽ so sánh 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



Hình 3.1. Cấu trúc thư mục của chương trình

Mã nguồn của chương trình bao gồm 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 của chương trình

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

Khi đặt chương trình ở một 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.

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

Hoạt động trên các máy Linux.

Các hành động 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

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 địa chỉ như địa chỉ ip nguồn/đích số port.

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 IP
- Tcp: Các gói tin TCP
- Udp: Các gói tin UDP
- Icmp: Các gói tin ICMP

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

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 và count	Sẽ thực hiện action khi có n gói tin khớp với rule trong vòng x giây, thường dùng để chống dos
cpu	Sẽ thực hiện action khi có gói tin khớp với rule và tại thời điểm đó cpu usage đang lớn hơn mức này

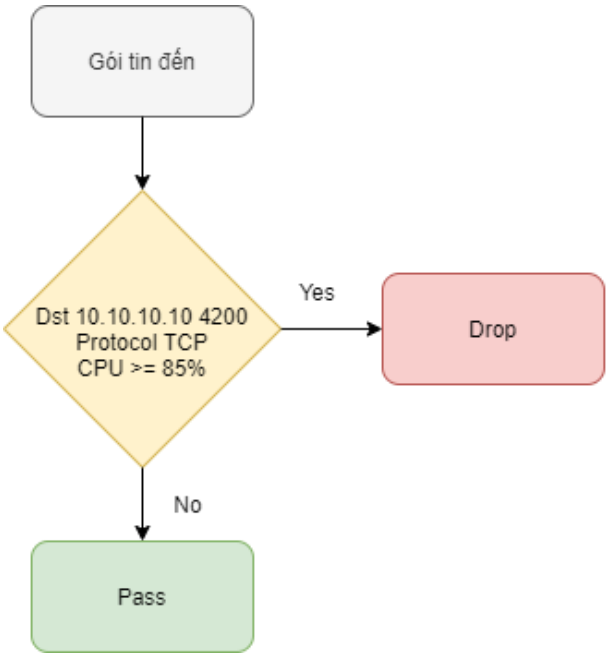
Rule format:

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

Ví dụ:

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

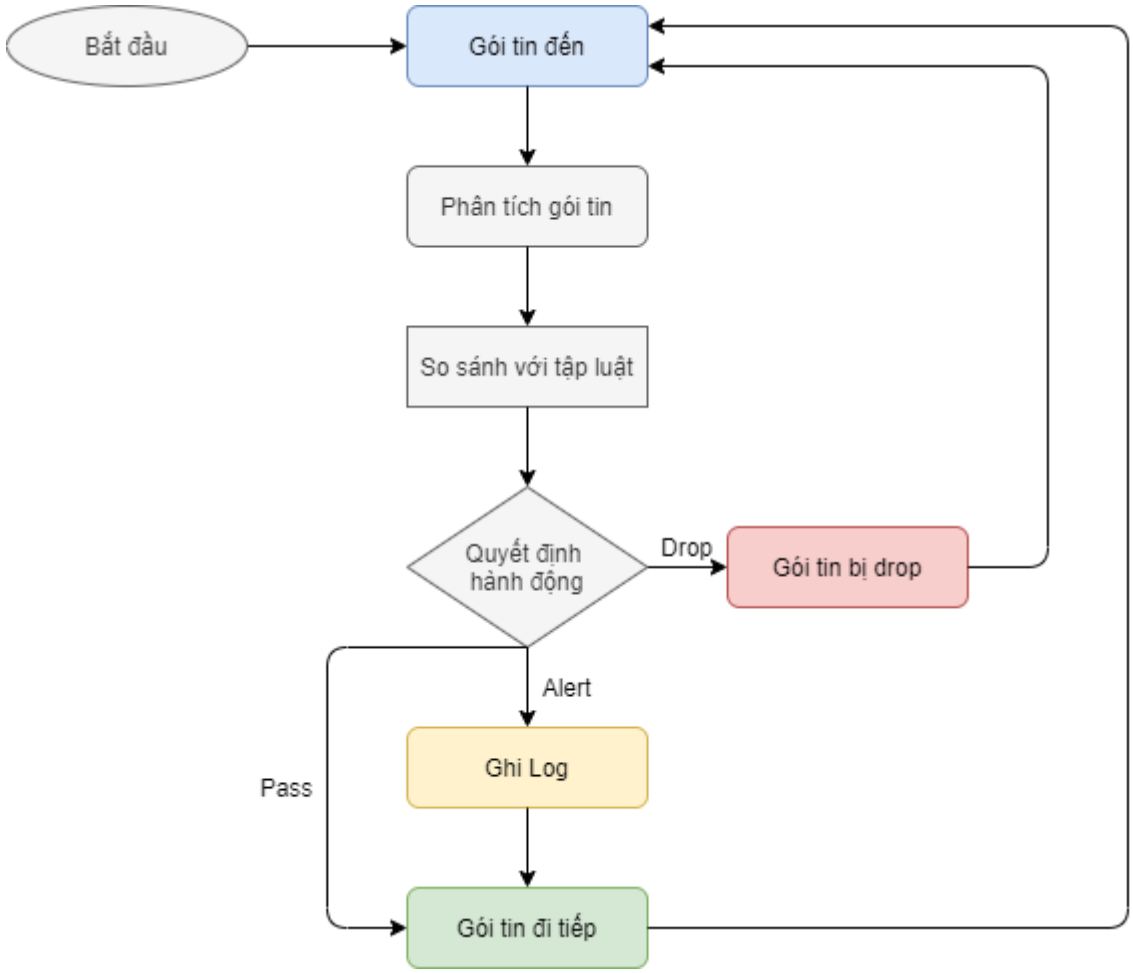
Có nghĩa là sẽ drop các gói tin tcp từ tất cả các nguồn đến host 10.10.10.10 ở port 4200 khi cpu đang cao hơn mức 85%.



Hình 3.2. 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



Hình 3.3. Luồng chạy của chương trình

Khi mới bắt đầu, chương trình sẽ tiến hành các bước chuẩn bị, bao gồm việc đọc tập luật, cấu hình cho iptables gửi gói tin đến nfqueue, binding chương trình với queue.

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

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.

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

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 3.4. Binding với nfqueue để lấy ra gói tin

Thành phần phân tích gói tin

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 3.5. 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 3.6. 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

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

Hình 3.7. Chạy chương trình

Chạy chương trình với quyền sudo.

Các tham số lần lượt là:

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

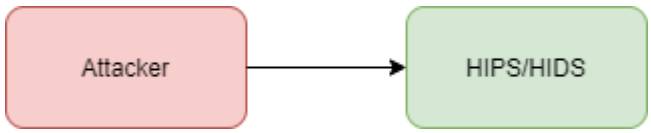
```
ruleFile = /home/hung/magic/rules/test.rules
logFile = /home/hung/magic/log/log.txt
iptablesFile = /home/hung/rules.v4
```

Hình 3.8. Nội dung file config của chương trình

File config sẽ chứa:

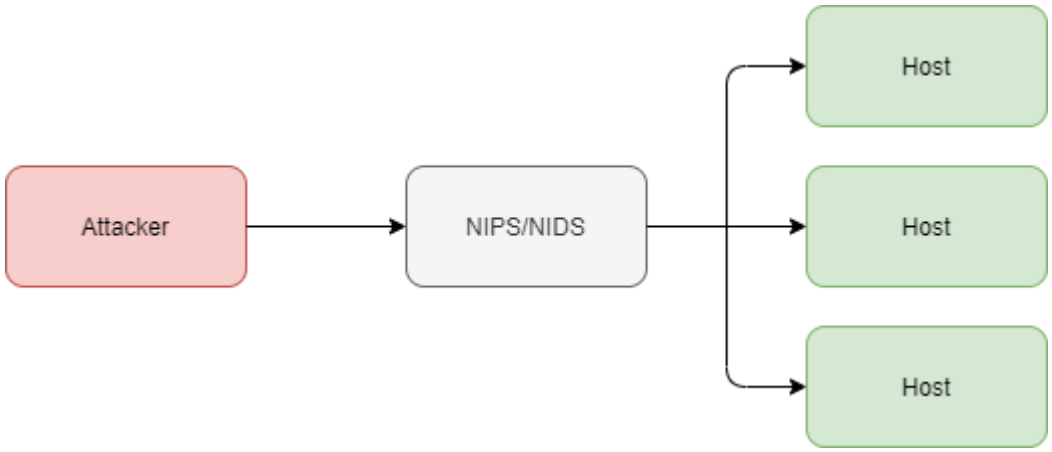
- Đường dẫn đến file chứa tập luật
- Đường dẫn đến file ghi log
- Đường dẫn đến file để backup iptables

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



Hình 3.9. Chương trình hoạt động ở host

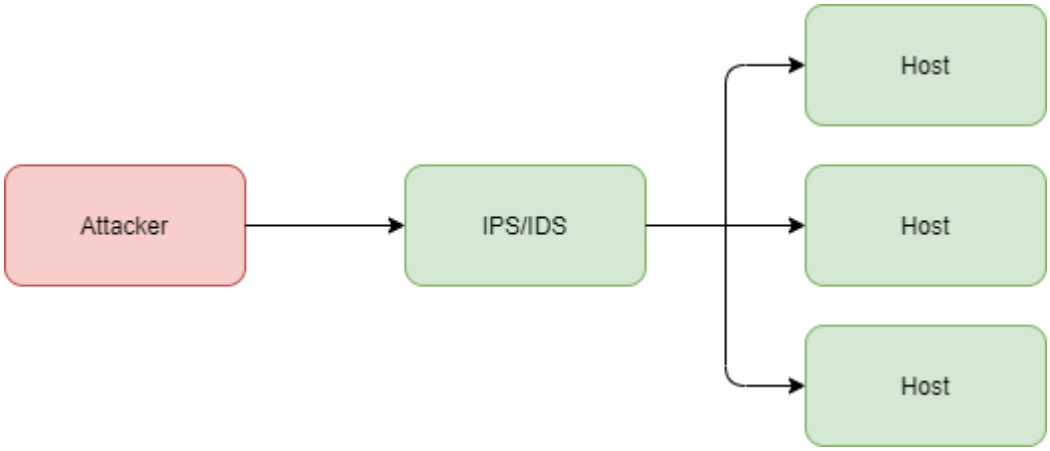
Chương trình hoạt động như HIDS/HIPS, theo dõi và xử lý các gói tin đến và đi ở máy đặt chương trình (màu xanh).



Hình 3.10. 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.

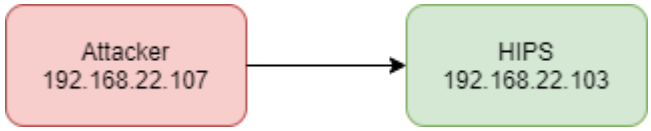


Hình 3.11. Chương trình hoạt động ở cả host và mạng.

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
```

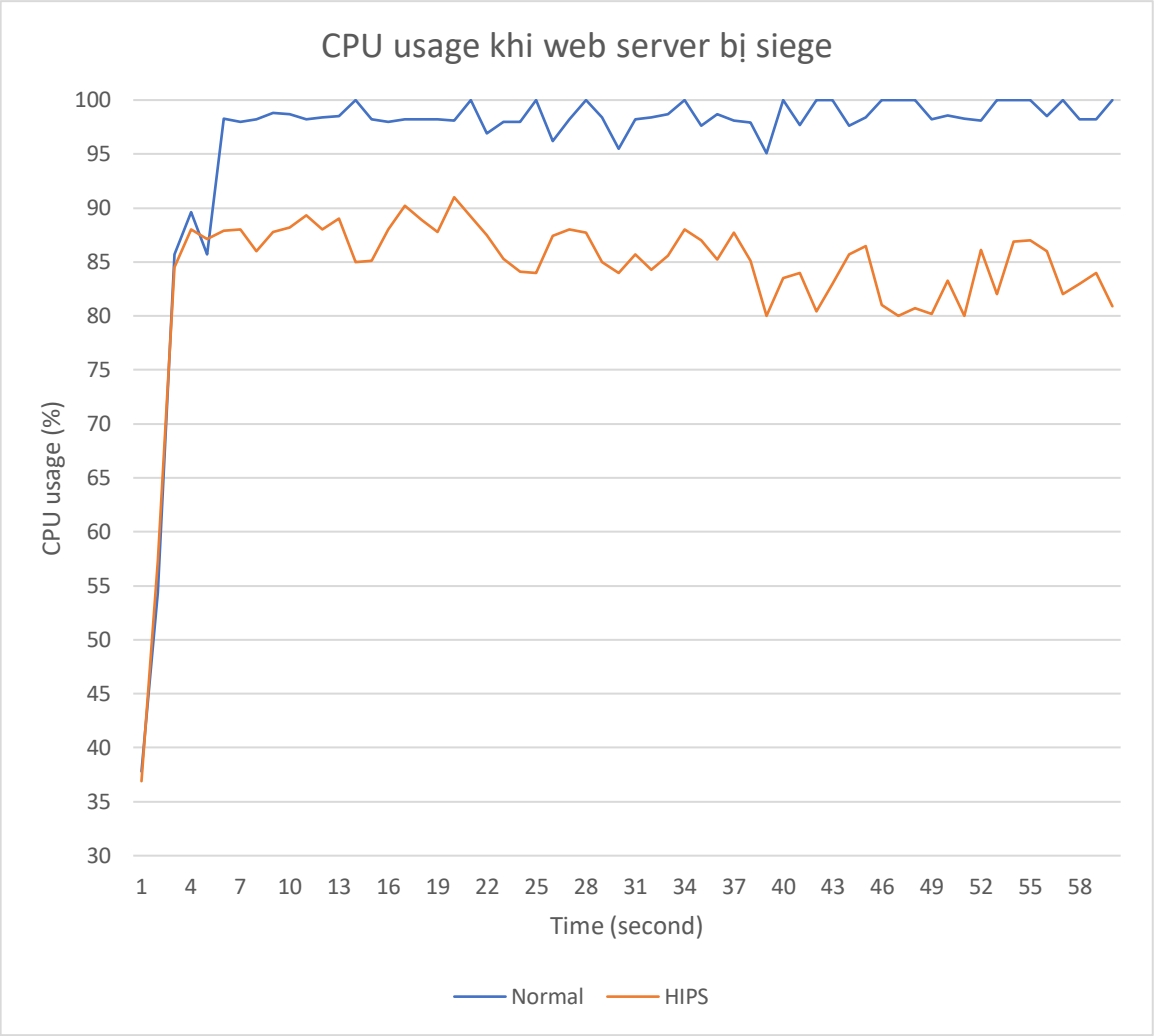
Hình 4.2. Rules của chương trình

```
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

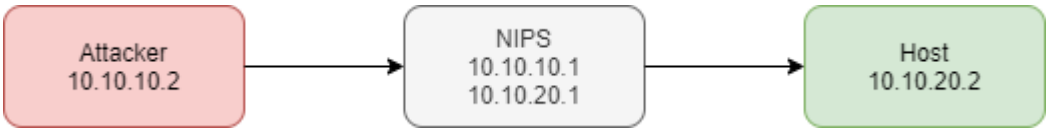
Nhận xét

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
```

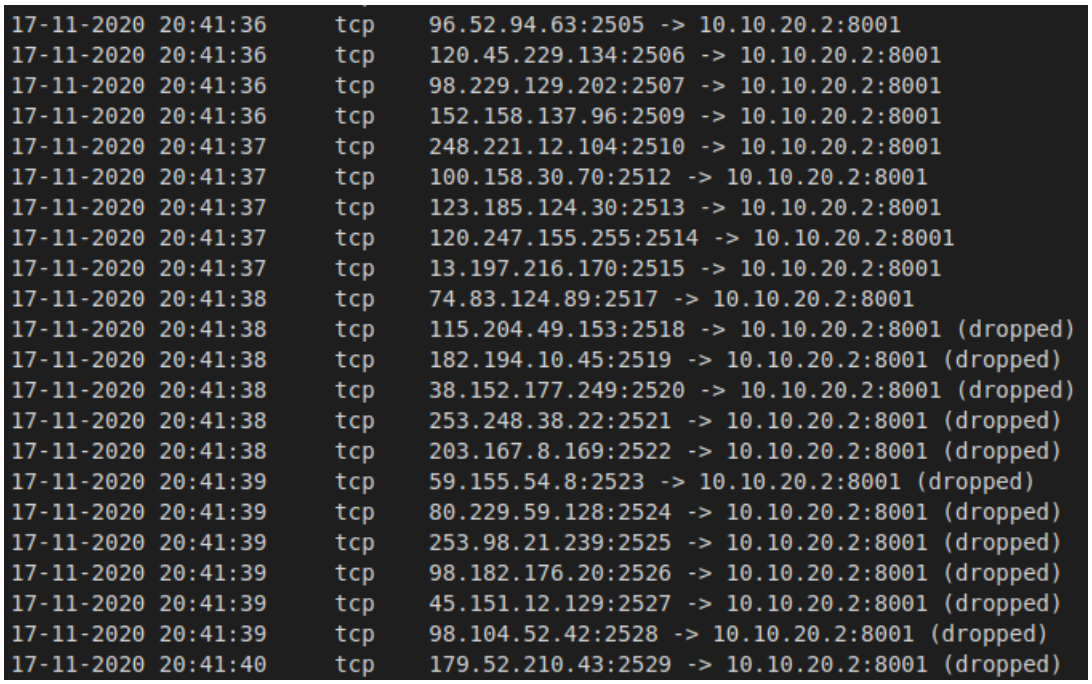
Hình 4.7. Chạy chương trình

```
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



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 (trong tương lai sẽ bổ sung thêm option timeout cho phép các gói tin đi qua sau một khoảng thời gian kể từ thời điểm phát hiện dos).

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.

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 tùy chọn 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.

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