# 

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

**Viện Công nghệ thông tin và Truyền thông**

-----🙞🙜🕮🙞🙜-----



BÁO CÁO

**Nhập môn An toàn thông tin**

# Mã lớp: 115655

Nhóm số 13

***Giảng viên hướng dẫn***: **PGS. TS. NGUYỄN LINH GIANG**

Sinh viên thực hiện:

1. Mai Thế Hưng : 20173161

2. Nguyễn Văn Huy : 20173175

3. Nguyễn Đức Thắng : 20173366

**Hà Nội, 2020**

# **LỜI NÓI ĐẦU**

Trong thời kỳ công nghệ phát triển nhanh chóng như ngày nay, cũng kéo theo những mặt không tốt. An ninh thông tin nói chung và an ninh mạng nói riêng đang là vấn đề được quan tâm không chỉ ở Việt Nam mà trên toàn thế giới. Cùng với sự phát triển nhanh chóng của mạng Internet, thì nhu cầu thiết yếu trong việc bảo vệ cơ sở hệ thống của doanh nghiệp từ các cuộc tấn công bảo mật bên ngoài là rất lớn.

Trong lĩnh vực an ninh mạng, phát hiện và phòng chống tấn công xâm nhập cho các mạng máy tính là một đề tài hay, thu hút được sự chú ý của nhiều nhà nghiên cứu với nhiều hướng nghiên cứu khác nhau. Trong xu hướng đó, bài tập lớn này chúng em mong muốn có thể tìm hiểu, nghiên cứu về phát hiện và phòng chống xâm nhập mạng với mục đích nắm bắt được các giải pháp, các kỹ thuật tiên tiến để chuẩn bị tốt cho hành trang của mình sau khi ra trường.

Mặc dù đã cố gắng hết sức nhưng do kiến thức và khả năng nhìn nhận vấn đề còn hạn chế nên bài làm không tránh khỏi thiếu sót, rất mong được sự quan tâm và góp ý thêm của thầy giáo . Để có thể hoàn thành đươc bài tập lớn này , chúng em xin gửi lời cảm ơn sâu sắc nhất tới thầy Nguyễn Linh Giang đã nhiệt tình hướng dẫn, chỉ bảo và cung cấp cho chúng em nhiều kiến thức rất bổ ích trong suốt quá trình làm. Nhờ sự giúp đỡ tận tâm của thầy, chúng em mới có thể hoàn thành được bài tập lớn này. Một lần nữa xin cảm ơn thầy rất nhiều !

# **CHƯƠNG I : TỔNG QUAN VỀ IDS**

* 1. **Giới thiệu tổng quan về IDS**

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

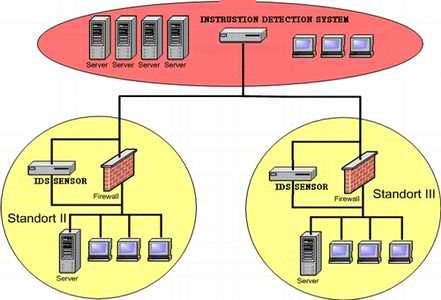
### **1.1.1 Định nghĩa**

Intrusion Detection system (IDS) là một hệ thống giám sát hoạt động trên hệ thống mạng và phân tích để tìm ra các dấu hiệu vi phạm đến các quy định bảo mật máy tính, chính sách sử dụng và các tiêu chuẩn an toàn thông tin. Các dấu hiệu này xuất phát từ rất nhiều nguyên nhân khác nhau, như lây nhiễm malwares, hackers xâm nhập trái phép, người dùng cuối truy nhập vào các tài nguyên không được phép truy cập..v.v

IDS phát hiện dựa trên các dấu hiệu đặc biệt về các nguy cơ đã biết (giống như cách các phần mềm diệt virus dựa vào các dấu hiệu đặc biệt để phát hiện và diệt virus) hay dựa trên so sánh lưu thông mạng hiện tại với baseline (thông số đo đạt chuẩn của hệ thống có thể chấp nhận được ngay tại thời điểm hiện tại) để tìm ra các dấu hiệu khác thường.

Hệ thống IDS sẽ thu thập thông tin từ rất nhiều nguồn trong hệ thống được bảo vệ sau đó tiến hành phân tích những thông tin đó theo các cách khác nhau để phát hiện những xâm nhập trái phép. Khi một hệ thống IDS có khả năng ngăn chặn các nguy cơ xâm nhập mà nó phát hiện được thì nó được gọi là một hệ thống phòng chống xâm nhập hay IPS.

Chức năng ban đầu của IDS chỉ là phát hiện các dấu hiện xâm nhập, do đó IDS chỉ có thể tạo ra các cảnh báo tấn công khi tấn công đang diễn ra hoặc thậm chí sau khi tấn công đã hoàn tất. Càng về sau, nhiều kỹ thuật mới được tích hợp vào IDS, giúp nó có khả năng dự đoán được tấn công (prediction) và thậm chí phản ứng lại các tấn công đang diễn ra (Active response).

Hình sau minh hoạ các vị trí thường cài đặt IDS trong mạng :

* 1. **Cấu trúc và kiến trúc**
     1. **Cấu trúc**

Các thành phần cơ bản :

* ***Sensor / Agent*** : giám sát và phân tích các hoạt động. “Sensor” thường được dùng cho dạng Network-base IDS trong khi “Agent” thường được dùng cho dạng Host-base IDS
* ***Management Server*** : là 1 thiết bị trung tâm dùng thu nhận các thông tin từ Sensor / Agent và quản lý chúng. 1 số Management Server có thể thực hiện việc phân tích các thông tin sự việc được cung cấp bởi Sensor / Agent và có thể nhận dạng được các sự kiện này dù các Sensor / Agent đơn lẻ không thể nhận diện.
* ***Database server*** : dùng lưu trữ các thông tin từ Sensor / Agent hay Management Server Console.

Là 1 chương trình cung cấp giao diện cho IDS users / Admins. Có thể cài đăt trên một máy tính bình thường dùng để phục vụ cho tác vụ quản trị, hoặc để giám sát, phân tích.

* + 1. **Kiến trúc**

Sensor là yếu tố cốt lõi trong một hệ thống IDS , nó mà có trách nhiệm phát hiện các xâm nhập nhờ chứa những cơ cấu ra quyết định đối với sự xâm nhập. Sensor nhận dữ liệu thô từ ba nguồn thông tin chính : kiến thức cơ bản ( knowledge base ) của IDS, syslog và audit trail .Các thông tin này tạo cơ sở cho quá trình ra quyết định sau này.

Sensor được tích hợp với các thành phần chịu trách nhiệm thu thập dữ liệu - một event generator. Dựa vào các chính sách tạo sự kiện nó xác định chế độ lọc thông tin thông báo sự kiện. Các event generator (hệ điều hành, mạng, ứng dụng) tạo ra một chính sách nhất quán tập các sự kiện có thể là log hoặc audit của các sự kiện của hệ thống, hoặc các gói tin. Điều này, thiết lập cùng với các thông tin chính sách có thể được lưu trữ hoặc là trong hệ thống bảo vệ hoặc bên ngoài. Trong những trường hợp nhất định, dữ liệu không được lưu trữ mà được chuyển trực tiếp đến các phân tích (thông thường áp dụng với các gói packet).

Các hệ thống IDS có thể được triển khai theo 2 hướng là tập trung và phân tán. Một ví dụ cụ thể cho hướng triển khai tập trung là tích hợp IDS cùng với các thành phần an ninh khác như firewall. Triển khai phân tán ( distributed IDS ) bao gồm nhiều hệ IDS trong 1 hệ thống mạng lớn, được kết nối với nhau nhằm nâng cao khả năng nhận diện chính xác xâm nhập và đưa ra phản ứng thích hợp.

* + 1. **Chức năng cơ bản**

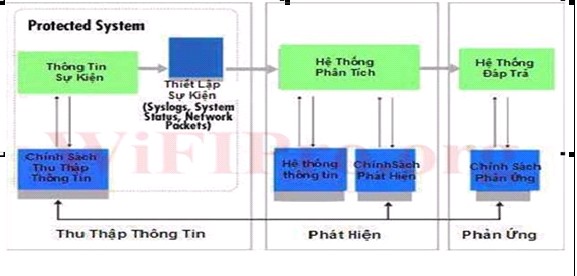
Nhìn chung, IDS không tự động cấm các cuộc tấn công hoặc là ngăn chặn những kẻ khai thác một cách thành công, tuy nhiên , một sự phát hiện mới nhất của IDS đó là hệ thống ngăn chặn xâm phập đã có thể thực hiện nhiều vai trò hơn và có thể ngăn chặn các cuộc tấn công khi nó xảy ra. Định nghĩa IDS khó hơn là chúng ta tưởng. Đầu tiên, IDS được nhìn nhận như là một cái chuông báo trộm mà có thể thông báo cho bạn biết khi nào thì bạn bị tấn công, Tuy nhiên những hệ thống IDS hiện đại thì phức tạp hơn nhiều và ít người có thể đồng ý rằng có mức độ giống như một cái chuông báo trộm truyền thống đáng tin cậy.Nếu sự giống nhau là cùng được sử dụng thì một hệ thống IDS trông giống như những chiếc camera chống trộm hơn là một cái chuông, những người có trách nhiệm có thể quan sát chúng và đáp trả cho những đe dọa xâm nhập.

Thực tế thì dường như IDS chỉ nói cho chúng ta biết rằng mạng đang bị nguy hiểm. Và điều quan trọng để nhận ra đó là một vài cuộc tấn công vào mạng đã thành công nếu hệ thống không có IDS.

Hệ thống phát hiện xâm nhập cho phép các tổ chức bảo vệ hệ thống của họ khỏi những đe doạ với việc gia tăng kết nối mạng và sự tin cậy của hệ thống thông tin, bổ sung những điểm yếu của hệ thống khác...Có nhiều tài liệu giới thiệu về những chức năng mà IDS đã làm được nhưng có thể đưa ra vài lý do tại sao nên sử dụng hệ thống IDS như sau:

* Bảo vệ tính toàn vẹn của dữ liệu, đảm bảo sự nhất quán của dữ liệu trong hệ thống . Các biện pháp đưa ra ngăn chặn được thay đổi bất hợp pháp hoặc phá hoại dữ liệu.
* Bảo vệ tính riêng tư, tức là đảm bảo cho người sử dụng khai thác tài nguyên của hệ thống theo đúng chức năng, nhiệm vụ đã được phân cấp, ngăn chặn được sự truy nhập thông tin bất hợp pháp.
* Bảo vệ tính bí mật, giữ cho thông tin không bị lộ ra ngoài. Bảo vệ tính khả dụng, tức la hệ thống luôn sẵn sàng thực hiện yêu cầu truy nhập thông tin của người dùng hợp pháp.
* Cung cấp thông tin về sự truy cập, đưa ra những chính sách đối phó, khôi phục , sửa chữa...

Hệ thống IDS có thể giúp chúng ta ngăn ngừa các sự kiện khi nó chưa xảy ra, cung cấp các giải pháp cho mạng và cho host, và thậm chí cũng có thể hoạt động như một cái chuông báo động. Tuy nhiên, chức năng chính của nó là thông báo cho bạn biết về các sự kiện có liên quan đến an ninh hệ thống đang sắp sửa xảy ra bên trong mạng và hệ thống mà bạn kiểm soát.



* 1. **Phân loại IDS**

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

• Host-based IDS (HIDS): 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 (host) cần theo dõi.

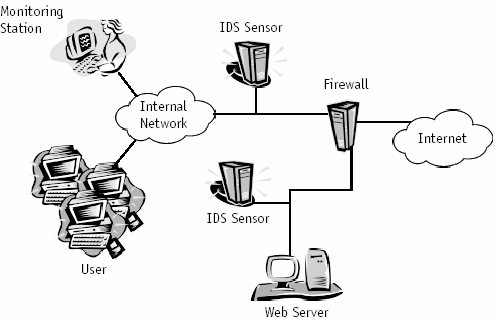
• Network-based IDS (NIDS): hệ thống phát hiện xâm nhập mạng.  Hệ thống sẽ tập hợp gói tin để phân tích sâu bên trong mà không làm thay đổi cấu trúc gói tin. NIDS có thể là phần mềm triển khai trên server hoặc dạng thiết bị tích hợp appliance.

* + 1. **Network based IDS – NIDS**

NIDS thường bao gồm có hai thành phần logic :

• Bộ cảm biến – Sensor : đặt tại một đoạn mạng, kiểm soát các cuộc lưu thông nghi ngờ trên đoạn mạng đó.

• Trạm quản lý : nhận các tín hiệu cảnh báo từ bộ cảm biến và thông báo cho một điều hành viên.



Một NIDS truyền thống với hai bộ cảm biến trên các đoạn mạng khác nhau cùng giao tiếp với một trạm kiểm soát. Thường dùng để giám sát, phân tích hoạt động hệ thống mạng trong 1 segment, phân tích mạng, các giao thức ứng dụng từ đó nhận diện các hoạt động khả nghi. Thường được triển khai ở các biên mạng ( network border ).  
 Hệ thống NIDS thường được triển khai trong 1 đoạn mạng con riêng phục vụ cho mục đích quản trị hệ thống ( management network ), trong trường hợp không có mạng quản trị riêng thì 1 mạng riêng ảo ( VLAN ) là cần thiết để bảo vệ các kết nối giữa các hệ NIDS.  
 Bên cạnh việc lựa chọn vị trí mạng phù hợp cho các thành phần của hệ NIDS, lựa chọn vị trí phù hợp cho các Sensor cũng là 1 vấn đề quan trọng ảnh hưởng đến khả năng detection của hệ NIDS. Trong hệ NIDS, các Sensor thường gặp ở 2 dạng là tích hợp phần cứng (appliance-based) và phần mềm ( software-only ).

***Hệ thống NIDS cung cấp các khả năng về bảo mật sau:***- Khả năng thu thập thông tin  
 + Nhận dạng host  
 + Nhận dạng hệ điều hành  
 + Nhận dạng ứng dụng  
 + Nhận dạng đặc điểm mạng  
- Khả năng ghi log  
- Khả năng nhận diện  
 + Hoạt động thăm dò và tấn công trên các lớp ứng dụng,vận chuyển và mạng  
 + Các dịch vụ ứng dụng không mong đợi ( unexpected application services)  
 + Vi phạm chính sách ( policy violations )  
-Khả năng ngăn chặn  
 + Kiểu thụ động: ngắt phiên TCP hiện tại  
 + Kiểu thẳng hàng: thực hiện tác vụ firewall thẳng hàng, điều tiết băng thông sử dụng, loại bỏ các nội dung gây hại  
 + Ngoài ra chức năng ngăn chặn còn có thể thay đổi cấu hình của 1 số thiết bị bảo mật cũng như thực thi các ứng dụng thứ 3 hoặc các script.  
  
***Lưu ý khi triển khai NIDS/IPS:*** Khi triển khai các hệ NIDS/IPS, 1 trong những điểm cần lưu ý là phải triển khai các Sensor ở dạng ẩn (Stealth mode). Trong dạng này, các interface của Sensor không được gán địa chỉ IP ( trừ interface quản lý ) để tránh việc khởi tạo kết nối từ các host khác nhằm ẩn Sensor khỏi sự phát hiện của kẻ tấn công.

***Ưu điểm:***

• Chi phí thấp : Do chỉ cần cài đặt NIDS ở những vị trí trọng yếu là có thể giám sát lưu lượng toàn mạng nên hệ thống không cần phải nạp các phần mềm và quản lý trên các máy toàn mạng.

• Phát hiện được các cuộc tấn công mà HIDS bỏ qua: Khác với HIDS, NIDS kiểm tra header của tất cả các gói tin vì thế nó không bỏ sót các dấu hiệu xuất phát từ đây. Ví dụ: nhiều cuộc tấn công DoS, TearDrop (phân nhỏ) chỉ bị phát hiện khi xem header của các gói tin lưu chuyển trên mạng.

• Khó xoá bỏ dấu vết (evidence): Các thông tin lưu trong log file có thể bị kẻ đột nhập sửa đổi để che dấu các hoạt động xâm nhập, trong tình huống này HIDS khó có đủ thông tin để hoạt động. NIDS sử dụng lưu thông hiện hành trên mạng để phát hiện xâm nhập. Vì thế, kẻ đột nhập không thể xoá bỏ được các dấu vết tấn công. Các thông tin bắt được không chỉ chứa cách thức tấn công mà cả thông tin hỗ trợ cho việc xác minh và buộc tội kẻ đột nhập.

• Phát hiện và đối phó kịp thời : NIDS phát hiện các cuộc tấn công ngay khi xảy ra, vì thế việc cảnh báo và đối phó có thể thực hiện được nhanh hơn. VD: Một hacker thực hiện tấn công DoS dựa trên TCP có thể bị NIDS phát hiện và ngăn chặn ngay bằng việc gửi yêu cầu TCP reset nhằm chấm dứt cuộc tấn công trước khi nó xâm nhập và phá vỡ máy bị hại.

• Có tính độc lập cao: Lỗi hệ thống không có ảnh hưởng đáng kể nào đối với công việc của các máy trên mạng. Chúng chạy trên một hệ thống chuyên dụng dễ dàng cài đặt; đơn thuần chỉ mở thiết bị ra, thực hiện một vài sự thay đổi cấu hình và cắm chúng vào trong mạng tại một vị trí cho phép nó kiểm soát các cuộc lưu thông nhạy cảm.

***Nhược điểm***

• Bị hạn chế với Switch: Nhiều lợi điểm của NIDS không phát huy được trong các mạng chuyển mạch hiện đại. Thiết bị switch chia mạng thành nhiều phần độc lập vì thế NIDS khó thu thập được thông tin trong toàn mạng. Do chỉ kiểm tra mạng trên đoạn mà nó trực tiếp kết nối tới, nó không thể phát hiện một cuộc tấn công xảy ra trên các đoạn mạng khác.Vấn đề này dẫn tới yêu cầu tổ chức cần phải mua một lượng lớn các bộ cảm biến để có thể bao phủ hết toàn mạng gây tốn kém về chi phí cài đặt.

• Hạn chế về hiệu năng: NIDS sẽ gặp khó khăn khi phải xử lý tất cả các gói tin trên mạng rộng hoặc có mật độ lưu thông cao, dẫn đến không thể phát hiện các cuộc tấn công thực hiện vào lúc "cao điểm". Một số nhà sản xuất đã khắc phục bằng cách cứng hoá hoàn toàn IDS nhằm tăng cường tốc độ cho nó. Tuy nhiên, do phải đảm bảo về mặt tốc độ nên một số gói tin được bỏ qua có thể gây lỗ hổng cho tấn công xâm nhập.

• Tăng thông lượng mạng: Một hệ thống phát hiện xâm nhập có thể cần truyền một dung lượng dữ liệu lớn trở về hệ thống phân tích trung tâm, có nghĩa là một gói tin được kiểm soát sẽ sinh ra một lượng lớn tải phân tích. Để khắc phục người ta thường sử dụng các tiến trình giảm dữ liệu linh hoạt để giảm bớt số lượng các lưu thông được truyền tải. Họ cũng thường thêm các chu trình tự ra các quyết định vào các bộ cảm biến và sử dụng các trạm trung tâm như một thiết bị hiển thị trạng thái hoặc trung tâm truyền thông hơn là thực hiện các phân tích thực tế. Điểm bất lợi là nó sẽ cung cấp rất ít thông tin liên quan cho các bộ cảm biến; bất kỳ bộ cảm biến nào sẽ không biết được việc một bộ cảm biến khác dò được một cuộc tấn công. Một hệ thống như vậy sẽ không thể dò được các cuộc tấn công hiệp đồng hoặc phức tạp.

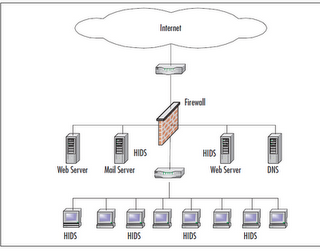
• Có thể xảy ra trường hợp báo động giả

• Không cho biết việc tấn công có thành công hay không

• Một hệ thống NIDS thường gặp khó khăn trong việc xử lý các cuộc tấn công trong một phiên được mã hoá. Lỗi này càng trở nên trầm trọng khi nhiều công ty và tổ chức đang áp dụng mạng riêng ảo VPN.

• Một số hệ thống NIDS cũng gặp khó khăn khi phát hiện các cuộc tấn công mạng từ các gói tin phân mảnh. Các gói tin định dạng sai này có thể làm cho NIDS hoạt động sai và đổ vỡ.

* + 1. **Host based IDS – HIDS**



Bằng cách cài đặt một phần mềm trên máy chủ, IDS dựa trên máy chủ quan sát tất cả những hoạt động về hệ thống và các file log, lưu lượng mạng thu thập. Hệ thống dựa trên máy chủ cũng theo dói OS, những cuộc gọi hệ thống, lịch sử và những thông điệp báo lỗi trên hệ thống máy chủ. HIDS thường được cài đặt trên một máy tính nhất định thay vì giám sát hoạt động của một network, HIDS chỉ giám sát các hoạt động trên một máy tính. HIDS thường được đặt trên các host quan trọng và các server trong vùng DMS. Nhiệm vụ của HIDS là theo dõi các thay đổi trên hệ thống gồm:

* Các tiến trình
* Các entry
* Mức độ sử dụng CPU
* Tình trạng RAM
* Tính toàn vẹn hệ thống
* Các thông số này khi vượt qua một ngưỡng nhất định hoặc có những thay đổi khả nghi sẽ gây báo động

***Hệ thống HIDS/IPS cung cấp các khả năng bảo mật sau:***  
- Khả năng ghi log.  
- Khả năng phát hiện  
 + Phân tích mã (phân tích hành vi mã, nhận diện buffer-overflow, giám sát hàm gọi hệ thống, giám sát danh sách ứng dụng và hàm thư viện)  
 + Phân tích và lọc lưu lượng mạng  
 + Giám sát filesystem ( kiểm tra tính toàn vẹn,thuộc tính,truy cập của file)  
 + Phân tích log  
 + Giám sát cấu hình mạng  
- Khả năng ngăn chặn  
 + Phân tích mã: ngăn chặn thực thi mã độc  
 + Phân tích và lọc lưu lượng mạng: ngăn chặn truy cập, lưu mã độc, chặn các dịch vụ hoặc giao thức không được phép  
Giám sát filesystem: ngăn chặn việc truy cập, thay đổi filesystem  
- Các khả năng bảo vệ khác: ngăn chặn truy cập đến các removeable-media, củng cố bảo mật cho host, giám sát trạng thái các tiến trình…  
Các sản phẩm đại diện: Tripware, OSSEC, BroIDS, ISS, Samhain, Prelude - LML,Snort.

***Ưu điểm:***

* Có khả năng xác định user liên quan tới event
* HIDS có khả năng phát hiện tấn công diễn ra trên một máy, NIDS thì không
* Có thể phân tích các dữ liệu mã hóa
* Cung cấp các thông tin về host trong lúc cuộc tấn công diễn ra trên host này

***Nhược điểm:***

* Thông tin từ HIDS là không đáng tin cậy ngay khi sự tấn công vào host này thành công
* Khi OS bị sập do tấn công, đồng thời HIDS cũng sập
* HIDS phải được thiết lập trên từng host cần giám sát
* HIDS không có khả năng phát hiện các cuộc dò quét mạng (Nmap, Netcat ...)
* HIDS cần tài nguyên trên host để hoạt động
* HIDS có thể không hiệu quả khi bị DOS.
* Hệ thống host-based tương đối đắt : nhiều tổ chức không có đủ nguồn tài chính để bảo vệ toàn bộ các đoạn mạng của mình sử dụng các hệ thống host-based. Những tổ chức đó phải rất thận trọng trong việc chọn các hệ thống nào để bảo vệ. Nó có thể để lại các lỗ hổng lớn trong mức độ bao phủ phát hiện xâm nhập. Ví dụ như một kẻ tấn công trên một hệ thống láng giềng không được bảo vệ có thể đánh hơi thấy các thông tin xác thực hoặc các tài liệu dễ bị xâm phạm khác trên mạng.
  1. **Cơ chế hoạt động của hệ thống IDS**

Có hai cách tiếp cận cơ bản đối với việc phát hiện và phòng chống xâm nhập là :

* Phát hiện sự lạm dụng (Misuse Detection Model): Hệ thống sẽ phát hiện các xâm nhập bằng cách tìm kiếm các hành động tương ứng với các kĩ thuật xâm nhập đã được biết đến (dựa trên các dấu hiệu - signatures) hoặc các điểm dễ bị tấn công của hệ thống. P
* Phát hiện sự bất thường (Anomaly Detection Model): Hệ thống sẽ phát hiện các xâm nhập bằng cách tìm kiếm các hành động khác với hành vi thông thường của người dùng hay hệ thống.
  + 1. **Phát hiện sự lạm dụng**

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

Ở đây, ta sử dụng thuật ngữ ***kịch bản xâm nhập (intrusion scenario)***. Một hệ thống phát hiện sự lạm dụng điển hình sẽ liên tục so sánh hành động của hệ thống hiện tại với một tập các kịch bản xâm nhập để cố gắng dò ra kịch bản đang được tiến hành. Hệ thống này có thể xem xét hành động hiện tại của hệ thống được bảo vệ trong thời gian thực hoặc có thể là các bản ghi kiểm tra được ghi lại bởi hệ điều hành.

Các kỹ thuật để phát hiện sự lạm dụng khác nhau ở cách thức mà chúng mô hình hoá các hành vi chỉ định một sự xâm nhập. Các hệ thống phát hiện sự lạm dụng thế hệ đầu tiên sử dụng các luật (rules) để mô tả những gì mà các nhà quản trị an ninh tìm kiếm trong hệ thống. Một lượng lớn tập luật được tích luỹ dẫn đến khó có thể hiểu và sửa đổi bởi vì chúng không được tạo thành từng nhóm một cách hợp lý trong một kịch bản xâm nhập.

Để giải quyết khó khăn này, các hệ thống thế hệ thứ hai đưa ra các biểu diễn kịch bản xen kẽ, bao gồm các tổ chức luật dựa trên mô hình và các biểu diễn về phép biến đổi trạng thái. Điều này sẽ mang tính hiệu quả hơn đối với người dùng hệ thống cần đến sự biểu diễn và hiểu rõ ràng về các kịch bản. Hệ thống phải thường xuyên duy trì và cập nhật để đương đầu với những kịch bản xâm nhập mới được phát hiện.

Do các kịch bản xâm nhập có thể được đặc tả một cách chính xác, các hệ thống phát hiện sự lạm dụng sẽ dựa theo đó để theo vết hành động xâm nhập. Trong một chuỗi hành động, hệ thống phát hiện có thể đoán trước được bước tiếp theo của hành động xâm nhập. Bộ dò tìm phân tích thông tin hệ thống để kiểm tra bước tiếp theo, và khi cần sẽ can thiệp để làm giảm bởi tác hại có thể.

* + 1. **Phát hiện sự bất thường**

Dựa trên việc định nghĩa và mô tả đặc điểm của các hành vi có thể chấp nhận của hệ thống để phân biệt chúng với các hành vi không mong muốn hoặc bất thường, tìm ra các thay đổi, các hành vi bất hợp pháp.

Như vậy, bộ phát hiện sự không bình thường phải có khả năng phân biệt giữa những hiện tượng thông thường và hiện tượng bất thường.

Ranh giới giữa dạng thức chấp nhận được và dạng thức bất thường của đoạn mã và dữ liệu lưu trữ được định nghĩa rõ ràng (chỉ cần một bit khác nhau), còn ranh giới giữa hành vi hợp lệ và hành vi bất thường thì khó xác định hơn.

Phát hiện sự không bình thường được chia thành hai loại tĩnh và động.

#### **1.4.2.1 Phát hiện tĩnh**

Dựa trên giả thiết ban đầu là phần hệ thống được kiểm soát phải luôn luôn không đổi. Ở đây, ta chỉ quan tâm đến phần mềm của vùng hệ thống đó (với giả sử là phần cứng không cần phải kiểm tra). Phần tĩnh của một hệ thống bao gồm 2 phần con: mã hệ thống và dữ liệu của phần hệ thống đó. Hai thông tin này đều được biểu diễn dưới dạng một xâu bit nhị phân hoặc một tập các xâu. Nếu biểu diễn này có sự sai khác so với dạng thức gốc thì hoặc có lỗi xảy ra hoặc một kẻ xâm nhập nào đó đã thay đổi nó. Lúc này, bộ phát hiện tĩnh sẽ được thông báo để kiểm tra tính toàn vẹn dữ liệu.

Cụ thể là: bộ phát hiện tĩnh đưa ra một hoặc một vài xâu bit cố định để định nghĩa trạng thái mong muốn của hệ thống. Các xâu này giúp ta thu được một biểu diễn về trạng thái đó, có thể ở dạng nén. Sau đó, nó so sánh biểu diễn trạng thái thu được với biểu diễn tương tự được tính toán dựa trên trạng thái hiện tại của cùng xâu bit cố định. Bất kỳ sự khác nhau nào đều là thể hiện lỗi như hỏng phần cứng hoặc có xâm nhập.

Biểu diễn trạng thái tĩnh có thể là các xâu bit thực tế được chọn để định nghĩa cho trạng thái hệ thống, tuy nhiên điều đó khá tốn kém về lưu trữ cũng như về các phép toán so sánh. Do vấn đề cần quan tâm là việc tìm ra được sự sai khác để cảnh báo xâm nhập chứ không phải chỉ ra sai khác ở đâu nên ta có thể sử dụng dạng biểu diễn được nén để giảm chi phí. Nó là giá trị tóm tắt tính được từ một xâu bit cơ sở. Phép tính toán này phải đảm bảo sao cho giá trị tính được từ các xâu bit cơ sở khác nhau là khác nhau. Có thể sử dụng các thuật toán checksums, message-digest (phân loại thông điệp), các hàm băm.

Một số bộ phát hiện xâm nhập kết hợp chặt chẽ với meta-data (dữ liệu mô tả các đối tượng dữ liệu) hoặc thông tin về cấu trúc của đối tượng được kiểm tra. Ví dụ, meta-data cho một log file bao gồm kích cỡ của nó. Nếu kích cỡ của log file tăng thì có thể là một dấu hiệu xâm nhập.

#### **1.4.2.2 Phát hiện động**

Trước hết ta đưa ra khái niệm ***hành vi*** của hệ thống ***(behavior)***. Hành vi của hệ thống được định nghĩa là một chuỗi các sự kiện phân biệt, ví dụ như rất nhiều hệ thống phát hiện xâm nhập sử dụng các bản ghi kiểm tra (audit record), sinh ra bởi hệ điều hành để định nghĩa các sự kiện liên quan, trong trường hợp này chỉ những hành vi mà kết quả của nó là việc tạo ra các bản ghi kiểm tra của hệ điều hành mới được xem xét.

Các sự kiện có thể xảy ra theo trật tự nghiêm ngặt hoặc không và thông tin phải được tích luỹ. Các ngưỡng được định nghĩa để phân biệt ranh giới giữa việc sử dụng tài nguyên hợp lý hay bất thường.

Nếu không chắc chắn hành vi là bất thường hay không, hệ thống có thể dựa vào các tham số được thiết lập trong suốt quá trình khởi tạo liên quan đến hành vi. Ranh giới trong trường hợp này là không rõ ràng do đó có thể dẫn đến những cảnh báo sai.

Cách thức thông thường nhất để xác định ranh giới là sử dụng các phân loại thống kê và các độ lệch chuẩn. Khi một phân loại được thiết lập, ranh giới có thể được vạch ra nhờ sử dụng một số độ lệch chuẩn. Nếu hành vi nằm bên ngoài thì sẽ cảnh báo là có xâm nhập.

Cụ thể là: các hệ thống phát hiện động thường tạo ra một profile (dữ liệu) cơ sở để mô tả đặc điểm các hành vi bình thường, chấp nhận được. Một dữ liệu bao gồm tập các đo lường được xem xét về hành vi, mỗi đại lượng đo lường gồm nhiều chiều:

• Liên quan đến các lựa chọn: thời gian đăng nhập, vị trí đăng nhập,… • Các tài nguyên được sử dụng trong cả quá trình hoặc trên một đơn vị thời gian: chiều dài phiên giao dịch, số các thông điệp gửi ra mạng trong một đơn vị thời gian,…

• Chuỗi biểu diễn các hành động.

Sau khi khởi tạo dữ liệu cơ sở, quá trình phát hiện xâm nhập có thể được bắt đầu. Phát hiện động lúc này cũng giống như phát hiện tĩnh ở đó chúng kiểm soát hành vi bằng cách so sánh mô tả đặc điểm hiện tại về hành vi với mô tả ban đầu của hành vi được mong đợi (chính là dữ liệu cơ sở), để tìm ra sự khác nhau. Khi hệ thống phát hiện xâm nhập thực hiện, nó xem xét các sự kiện liên quan đến thực thể hoặc các hành động là thuộc tính của thực thể. Chúng xây dựng thêm một dữ liệu hiện tại.

Các hệ thống phát hiện xâm nhập thế hệ trước phải phụ thuộc vào các bản ghi kiểm tra (audit record) để bắt giữ các sự kiện hoặc các hành động liên quan. Các hệ thống sau này thì ghi lại một cơ sở dữ liệu đặc tả cho phát hiện xâm nhập. Một số hệ thống hoạt động với thời gian thực, hoặc gần thời gian thực, quan sát trực tiếp sự kiện trong khi chúng xảy ra hơn là đợi hệ điều hành tạo ra bản ghi mô tả sự kiện.

Khó khăn chính đối với các hệ thống phát hiện động là chúng phải xây dựng các dữ liệu cơ sở một cách chính xác, và sau đó nhận dạng hành vi sai trái nhờ các dữ liệu.

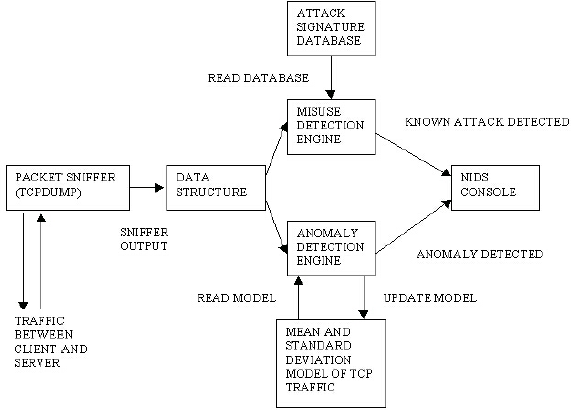
Các dữ liệu cơ sở có thể xây dựng nhờ việc giả chạy hệ thống hoặc quan sát hành vi người dùng thông thường qua một thời gian dài.

* + 1. **So sánh giữa hai mô hình**

|  |  |
| --- | --- |
| **Phát hiện sự lạm dụng** | **Phát hiện sự bất thường** |
| Bao gồm:  • Cơ sở dữ liệu các dấu hiệu tấn công.  • Tìm kiếm các so khớp mẫu đúng. | Bao gồm:  • Cơ sở dữ liệu các hành động thông thường.  • Tìm kiếm độ lệch của hành động thực tế so với hành động thông thường. |
| Hiệu quả trong việc phát hiện các dạng tấn công đã biết, hay các biến thể (thay đổi nhỏ) của các dạng tấn công đã biết. Không phát hiện được các dạng tấn công mới. | Hiệu quả trong việc phát hiện các dạng tấn công mới mà một hệ thống phát hiện sự lạm dụng bỏ qua. |
| Dễ cấu hình hơn do đòi hỏi ít hơn về thu thập dữ liệu, phân tích và cập nhật | Khó cấu hình hơn vì đưa ra nhiều dữ liệu hơn, phải có được một khái niệm toàn diện về hành vi đã biết hay hành vi được mong đợi của hệ thống |
| Đưa ra kết luận dựa vào phép so khớp mẫu (pattern matching). | Đưa ra kết quả dựa vào tương quan bằng thống kê giữa hành vi thực tế và hành vi được mong đợi của hệ thống (hay chính là dựa vào độ lệch giữa thông tin thực tế và ngưỡng cho phép). |
| Có thể kích hoạt một thông điệp cảnh báo nhờ một dấu hiệu chắc chắn, hoặc cung cấp dữ liệu hỗ trợ cho các dấu hiệu khác | Có thể hỗ trợ việc tự sinh thông tin hệ thống một cách tự động nhưng cần có thời gian và dữ liệu thu thập được phải rõ ràng. |

Để có được một hệ thống phát hiện xâm nhập tốt nhất ta tiến hành kết hợp cả hai phương pháp trên trong cùng một hệ thống. Hệ thống kết hợp này sẽ cung cấp khả năng phát hiện nhiều loại tấn công hơn và hiệu quả hơn.

Sơ đồ hệ thống kết hợp như sau:



# **Chương II: Nghiên cứu ứng dụng Snort trong IDS**

## **2.1 Giới thiệu về Snort**

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

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

## **2.2 Kiến trúc của Snort**

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

• Môđun giải mã gói tin (Packet Decoder)

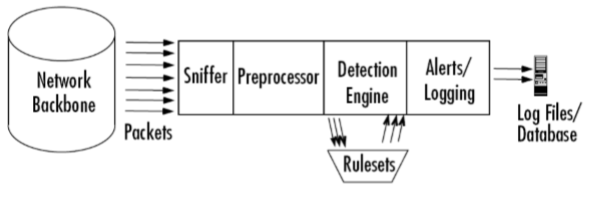
• Môđun tiền xử lý (Preprocessors)

• Môđun phát hiện (Detection Engine)

• Môđun log và cảnh báo (Logging and Alerting System)

• Môđun kết xuất thông tin (Output Module)

Kiến trúc của Snort được mô tả trong hình sau:



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

### ***2.2.1 Modun giải mã gói tin***

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

Ảnh có chứa văn bản, bản đồ

Mô tả được tạo tự động

### ***2.2.2 Mô đun tiền xử lý***

Ba nhiệm vụ chính của các môđun loại này là:

* Kết hợp lại các gói tin: Khi một lượng dữ liệu lớn được gửi đi, thông tin sẽ không đóng gói toàn bộ vào một gói tin mà phải thực hiện việc phân mảnh, chia gói tin ban đầu thành nhiều gói tin rồi mới gửi đi. Khi Snort nhận được các gói tin này nó phải thực hiện việc ghép nối lại để có được dữ liệu nguyên dạng ban đầu, từ đó mới thực hiện được các công việc xử lý tiếp.
* Giải mã và chuẩn hóa giao thức (decode/normalize): công việc phát hiện xâm nhập dựa trên dấu hiệu nhận dạng nhiều khi bị thất bại khi kiểm tra các giao thức có dữ liệu có thể được thể hiện dưới nhiều dạng khác nhau. Ví dụ: một web server có thể chấp nhận nhiều dạng URL như URL được viết dưới dạng mã hexa/Unicode, URL chấp nhận cả dấu \ hay / hoặc nhiều ký tự này liên tiếp cùng lúc. Chẳng hạn ta có dấu hiệu nhận dạng “scripts/iisadmin”, kẻ tấn công có thể vượt qua được bằng cách tùy biến các yêu cấu gửi đến web server như sau:

“scripts/./iisadmin”

“scripts/examples/../iisadmin”

“scripts\iisadmin”

“scripts/.\iisadmin”

Hoặc thực hiện việc mã hóa các chuỗi này dưới dạng khác. Nếu Snort chỉ thực hiện đơn thuần việc so sánh dữ liệu với dấu hiệu nhận dạng sẽ xảy ra tình trạng bỏ sót các hành vi xâm nhập. Hiện nay Snort đã hỗ trợ việc giải mã và chuẩn hóa cho các giao thức: telnet, http, rpc, arp.

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

### ***2.2.3 Môđun phát hiện***

Đây là môđun quan trọng nhất của Snort. Nó chịu trách nhiệm phát hiện các dấu hiệu xâm nhập. Môđun phát hiện sử dụng các luật được định nghĩa trước để so sánh với dữ liệu thu thập được từ đó xác định xem có xâm nhập xảy ra hay không. Rồi tiếp theo mới có thể thực hiện một số công việc như ghi log, tạo thông báo và kết xuất thông tin. Một vấn đề rất quan trọng trong môđun phát hiện là vấn đề thời gian xử lý các gói tin: một IDS thường nhận được rất nhiều gói tin và bản thân nó cũng có rất nhiều các luật xử lý. Có thể mất những khoảng thời gian khác nhau cho việc xử lý các gói tin khác nhau. Và khi thông lượng mạng quá lớn có thể xảy ra việc bỏ sót hoặc không phản hồi được đúng lúc. Khả năng xử lý của môđun phát hiện dựa trên một số yếu tố như: số lượng các luật, tốc độ của hệ thống đang chạy Snort, tải trên mạng. Một số thử nghiệm cho biết, phiên bản hiện tại của Snort khi được tối ưu hóa chạy trên hệ thống có nhiều bộ vi xử lý và cấu hình máy tính tương đối mạnh thì có thể hoạt động tốt trên cả các mạng cỡ Giga.

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

• IP header

• Header ở tầng giao vận: TCP, UDP

• Header ở tầng ứng dụng: DNS header, HTTP header, FTP header, …

• Phần tải của gói tin (bạn cũng có thể áp dụng các luật lên các phần dữ liệu được truyền đi của gói tin)

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

***2.2.4 Môđun log và cảnh báo***

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

***2.2.5 Mô đun kết xuất thông tin***

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

• Ghi log file

• Ghi syslog: syslog và một chuẩn lưu trữ các file log được sử dụng rất nhiều trên các hệ thống Unix, Linux.

• Ghi cảnh báo vào cơ sở dữ liệu.

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

• Cấu hình lại Router, firewall.

• Gửi các cảnh báo được gói trong gói tin sử dụng giao thức SNMP. Các gói tin dạng SNMP này sẽ được gửi tới một SNMP server từ đó giúp cho việc quản lý các cảnh báo và hệ thống IDS một cách tập trung và thuận tiện hơn.

• Gửi các thông điệp SMB (Server Message Block) tới các máy tính Windows. Nếu không hài lòng với các cách xuất thông tin như trên, ta có thể viết các môđun kết xuất thông tin riêng tuỳ theo mục đích sử dụng.

**2.3 Bộ luật của snort**

***2.3.1 Giới thiệu***

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

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

***2.3.2 Cấu trúc luật của Snort***

Hãy xem xét một ví dụ đơn giản :

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

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

Rule Header + Rule Option

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

• Phần Header chứa thông tin về hành động mà luật đó sẽ thực hiện khi phát hiện ra có xâm nhập nằm trong gói tin và nó cũng chứa các tiêu chuẩn để áp dụng luật với gói tin đó.

• Phần Option chứa một thông điệp cảnh báo và các thông tin về các phần của gói tin dùng để tạo nên cảnh báo. Phần Option chứa các tiêu chuẩn phụ thêm để đối sánh luật với gói tin. Một luật có thể phát hiện được một hay nhiều hoạt động thăm dò hay tấn công. Các luật thông minh có khả năng áp dụng cho nhiều dấu hiệu xâm nhập.

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

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

* Action: là phần qui định loại hành động nào được thực thi khi các dấu hiệu của gói tin được nhận dạng chính xác bằng luật đó. Thông thường, các hành động tạo ra một cảnh báo hoặc log thông điệp hoặc kích hoạt một luật khác.
* Protocol: là phần qui định việc áp dụng luật cho các packet chỉ thuộc một giao thức cụ thể nào đó. Ví dụ như IP, TCP, UDP …
* Address: là phần địa chỉ nguồn và địa chỉ đích. Các địa chỉ có thể là một máy đơn, nhiều máy hoặc của một mạng nào đó. Trong hai phần địa chỉ trên thì một sẽ là địa chỉ nguồn, một sẽ là địa chỉ đích và địa chỉ nào thuộc loại nào sẽ do phần Direction “->” qui định.
* Port: xác định các cổng nguồn và đích của một gói tin mà trên đó luật được áp dụng.
* Direction: phần này sẽ chỉ ra đâu là địa chỉ nguồn, đâu là địa chỉ đích.

Ví dụ:

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

Phần đứng trước dấu mở ngoặc là phần Header của luật còn phần còn lại là phần Option. Chi tiết của phần Header như sau:

* Hành động của luật ở đây là “alert” : một cảnh báo sẽ được tạo ra nếu như các điều kiện của gói tin là phù hợp với luật(gói tin luôn được log lại mỗi khi cảnh báo được tạo ra).
* Protocol của luật ở đây là ICMP tức là luật chỉ áp dụng cho các gói tin thuộc loại ICMP. Bởi vậy, nếu như một gói tin không thuộc loại ICMP thì phần còn lại của luật sẽ không cần đối chiếu.
* Địa chỉ nguồn ở đây là “any”: tức là luật sẽ áp dụng cho tất cả các gói tin đến từ mọi nguồn còn cổng thì cũng là “any” vì đối với loại gói tin ICMP thì cổng không có ý nghĩa. Số hiệu cổng chỉ có ý nghĩa với các gói tin thuộc loại TCP hoặc UDP thôi.
* Còn phần Option trong dấu đóng ngoặc chỉ ra một cảnh báo chứa dòng “Ping with TTL=100” sẽ được tạo khi tìm thấy điều kiện TTL=100. TTL là Time To Live là một trường trong Header IP.

#### 2.3.2.1 Phần tiêu đề

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

**Hành động của luật (Rule Action)**

Là phần đầu tiên của luật, chỉ ra hành động nào được thực hiện khi mà các điều kiện của luật được thoã mãn. Một hành động được thực hiện khi và chỉ khi tất cả các điều kiện đều phù hợp. Có 5 hành động đã được định nghĩa nhưng ta có thể tạo ra các hành động riêng tuỳ thuộc vào yêu cầu của mình.

* Pass: Hành động này hướng dẫn Snort bỏ qua gói tin này. Hành động này đóng vai trò quan trọng trong việc tăng cường tốc độ hoạt động của Snort khi mà ta không muốn áp dụng các kiểm tra trên các gói tin nhất định.
* Log: Hành động này dùng để log gói tin. Có thể log vào file hay vào cơ sở dữ liệu tuỳ thuộc vào nhu cầu của mình.
* Alert: Gửi một thông điệp cảnh báo khi dấu hiệu xâm nhập được phát hiện. Có nhiều cách để gửi thông điệp như gửi ra file hoặc ra một Console. Tất nhiên là sau khi gửi thông điệp cảnh báo thì gói tin sẽ được log lại.
* Activate: sử dụng để tạo ra một cảnh báo và kích hoạt một luật khác kiểm tra thêm các điều kiện của gói tin.
* Dynamic: chỉ ra đây là luật được gọi bởi các luật khác có hành động là Activate.

Hành động do người dùng định nghĩa:

ruletype action\_name { action definition } với ruletype là từ khoá.

Ví dụ như:

ruletype smb\_db\_alert {

type alert

output alert\_smb: workstation.list

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

dbname=snort host = localhost

}

Hành động này có tên là smb\_db\_alert dùng để gửi thông điệp cảnh báo dưới dạng cửa sổ pop-up SMB tới các máy có tên trong danh sách liệt kê trong file workstation.list và tới cơ sở dữ liệu MySQL tên là snort.

**Protocols**

Là phần thứ hai của một luật có chức năng chỉ ra loại gói tin mà luật sẽ được áp dụng. Hiện tại Snort hiểu được các protocol sau :

• IP

• ICMP

• TCP

• UDP

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

**Address**

Có hai phần địa chỉ trong một luật của Snort. Các địa chỉ này được dùng để kiểm tra nguồn sinh ra và đích đến của gói tin. Địa chỉ có thể là địa chỉ của một IP đơn hoặc là địa chỉ của một mạng. Ta có thể dùng từ any để áp dụng luật cho tất cả các địa chỉ.

Ví dụ như luật :

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

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

**Ngăn chặn địa chỉ hay loại trừ địa chỉ**

Snort cung cấp cho ta kĩ thuật để loại trừ địa chỉ bằng cách sử dụng dấu phủ định (dấu !). Ví dụ, luật sau sẽ áp dụng cho tất cả các gói tin ngoại trừ các gói có nguồn xuất phát từ mạng lớp C 192.168.2.0. alert icmp ![192.168.2.0/24] any -> any any (msg: “Ping with TTL=100”; ttl: 100;)

**Danh sách địa chỉ**

Ví dụ nếu bạn muốn áp dụng luật cho tất cả các gói tin trừ các gói xuất phát từ hai mạng lớp C 192.168.2.0 và 192.168.8.0 thì luật được viết như sau:

alert icmp ![192.168.2.0/24, 192.168.8.0/24] any -> any any (msg: “Ping with TTL=100”; ttl: 100;)

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

**Cổng (Port Number)**

Số hiệu cổng dùng để áp dụng luật cho các gói tin đến từ hoặc đi đến một cổng hay một phạm vi cổng cụ thể nào đó. Ví dụ ta có thể sử dụng số cổng nguồn là 23 để áp dụng luật cho tất cả các gói tin đến từ một server Telnet. Từ any cũng được dùng để đại diện cho tất cả các cổng. Chú ý là số hiệu cổng chỉ có ý nghĩa trong các giao thức TCP và UDP thôi. Nếu protocol của luật là IP hay ICMP thì số hiệu cổng không đóng vai trò gì cả.

Ví dụ :

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

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

**Dãy cổng hay phạm vi cổng:**

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

Ví dụ :

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

Ta cũng có thể dùng cổng theo kiểu cận trên và cận dưới, tức là chỉ sử dụng cổng bắt đầu hoặc cổng kết thúc mà thôi. Ví dụ sau sẽ log tất cả các gói tin ngoại trừ các gói tin xuất phát từ cổng 53.

log udp any !53 -> any any log udp

**Hướng – Direction**

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

#### 2.3.2.2 Các tùy chọn

Phần Rule Option nằm ngay sau phần Rule Header và được bao bọc trong dấu ngoặc đơn. Các option sẽ được phân cách với nhau bằng dấu chấm phẩy ”,”.

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

**Từ khoá ack**

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

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

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

**Từ khoá classtype**

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

config classification: name, description, priority

trong đó:

• name: là tên dùng để phân loại, tên này sẽ được dùng với từ khoá classtype trong các luật Snort.

• description: mô tả về loại lớp này

• priority: là một số chỉ độ ưu tiên mặc định của lớp này. Độ ưu tiên này có thể được điều chỉnh trong từ khoá priority của phần option trong luật của Snort.

Ví dụ : config classification: DoS , Denial of Service Attack, 2

và trong luật:

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

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

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

**Từ khoá content**

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

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

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

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

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

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

Tuy nhiên khi sử dụng từ khoá content ta cần nhớ rằng: Đối sánh nội dung sẽ phải xử lý tính toán rất lớn và ta phải hết sức cân nhắc khi sử dụng nhiều luật có đối sánh nội dung.

Ta có thể sử dụng nhiều từ khoá content trong cùng một luật để tìm nhiều dấu hiệu trong cùng một gói tin. Đối sánh nội dung là công việc rất nhạy cảm. Có 3 từ khoá khác hay được dùng cùng với từ khoá content dùng để bổ sung thêm các điều kiện để tìm kiếm là :

• offset: dùng để xác định vị trí bắt đầu tìm kiếm (chuỗi chứa trong từ khoá content) là offset tính từ đầu phần dữ liệu của gói tin. Ví dụ sau sẽ tìm chuỗi “HTTP” bắt đầu từ vị trí cách đầu đoạn dữ liệu của gói tin là 4 byte:

alert tcp 192.168.1.0/24 any -> any any (content: “HTTP”; offset: 4; msg: “HTTP matched”;)

• dept : dùng để xác định vị trí mà từ đó Snort sẽ dừng việc tìm kiếm.Từ khoá này cũng thường được dùng chung với từ khoá offset vừa nêu trên.

• Ví dụ:

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

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

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

• “test”

“Snort”

“NIDS”

và ta có luật sau:

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

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

**Từ khoá dsize**

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

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

**Từ khoá flags**

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

|  |  |
| --- | --- |
| **Flag** | **Ký tự tham số** |
| FIN | F |
| SYN | S |
| RST (Reset) | R |
| PSH (Push) | P |
| ACK | A |
| URG (Urgent) | U |
| Reseved Bit 1 | 1 |
| Reserved Bit 2 | 2 |
| No Flag set | 0 |

**Từ khoá fragbits**

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

• Reserved Bit (RB) dùng để dành cho tương lai.

• Don’t Fragment Bit (DF): nếu bit này được thiết lập thì tức là gói tin đó không bị phân mảnh.

• More Fragments Bit (MF): nếu được thiết lập thì tức là các phần khác (gói tin bị phân mảnh) của gói tin vẫn đang còn trên đường đi mà chưa tới đích. Nếu bit này không được thiết lập thì có nghĩa là đây là phần cuối cùng của gói tin (hoặc là gói duy nhất). Bit MF này giúp bên đích có thể tổng hợp lại các phần khác nhau thành một gói tin hoàn chỉnh.

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

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

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

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

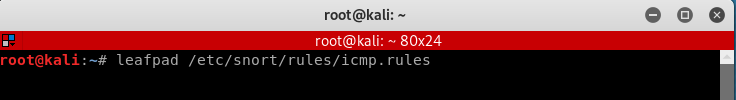
alert icmp any any -> 192.168.1.0/24 any (fragbits: !D; msg: “Dont Fragment bit not set”;)

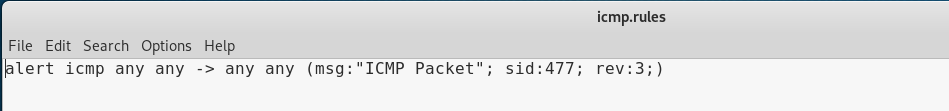
## **2.4 Thử nghiệm khả năng phản ứng của Snort IDS/IPS**

### **2.4.1 Phát hiện có máy đang ping**

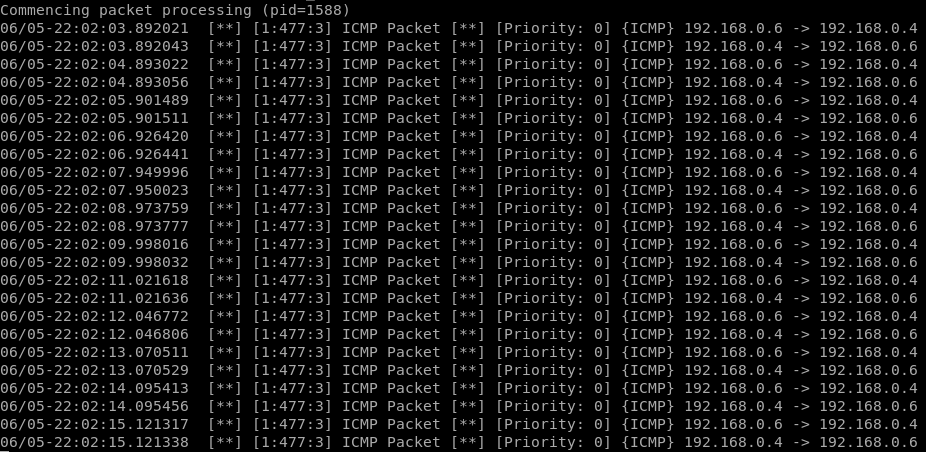
Mở file **icmp.rules** và thêm vào rule sau:

*alert icmp any any -> any any (msg:”ICMP Packet”; sid:447; rev:3;)*





Kết quả khi có máy khác (192.168.0.6) ping tới máy cài Snort (192.168.0.4):



### **2.4.2 Phát hiện tấn công DDOS**

**Khái niệm:**

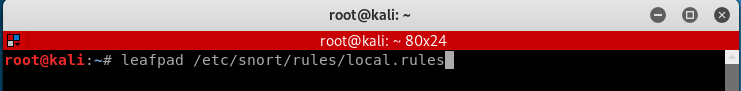
SYN flood ( half-open attack) là một kiểu tấn công từ chối dịch vụ (DDos), tấn công này với mục đích làm cho Server không có  lưu lượng để truy cập hợp pháp bằng cách tiêu thụ tất cả tài nguyên server đang có sẵn. Bằng việc gửi liên tục gửi các packet tin yêu cầu kết nối ban đầu (SYN). Người tấn công có thể áp đảo tất cả các cổng có sẵn trên [Server](https://vnso.vn/thue-may-chu-rieng/) được chọn muc tiêu, làm cho thiết bị Client đáp ứng lưu lượng hợp pháp một cách chậm chạp hoặc không đáp ứng kịp thời.

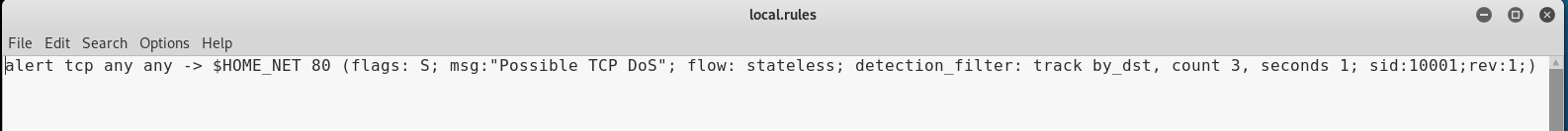
Demo gồm hai máy trong cùng một mạng LAN. Máy 1 có địa chỉ IP là 192.168.0.4, là máy bị tấn công và có cài đặt Snort. Máy 2 có địa chỉ IP là 192.168.0.6 là máy tấn công.

**Cài đặt Snort:**

Mở file **local.rules** và thêm vào rule sau:

*alert tcp any any -> $HOME\_NET 80 (flags: S; msg:”Possible TCP DoS”; flow: stateless; detection\_filter: track by\_dst, count 3, second 1; sid:10001; rev:1;)*

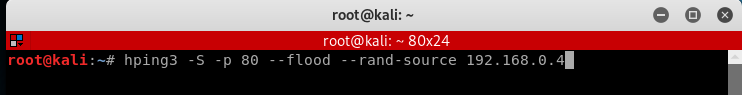




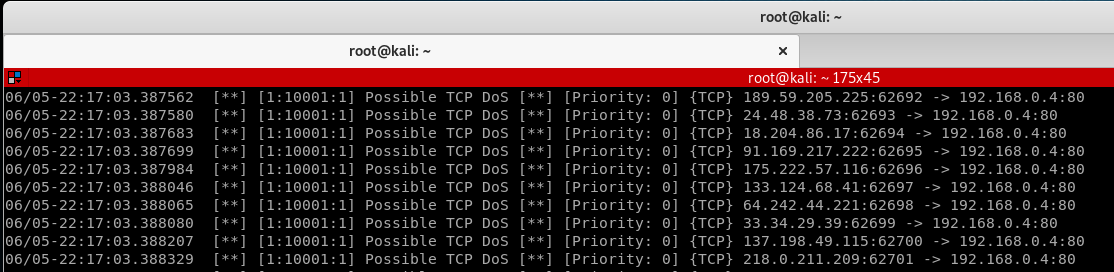
Trên máy tấn công (192.168.0.6) ta chạy dòng lệnh sau:

*hping3 -S -p 80 –flood –rand-source 192.168.0.4*

Giải thích: Máy tấn công sẽ gửi một khối lượng lớn các packet tin SYN đến cổng 80 của Server được nhắm là mục tiêu và thường là các địa chỉ IP giả mạo.



Kết quả:



Mức CPU sử dụng trên máy bị tấn công tăng cao:



**Giải thích kết quả:**

Sau khi máy bị tấn công gửi gói tin SYN/ACK nó sẽ phải đợi cho đến khi máy tấn công trả lời hoặc hết thời gian timeout. Khi đó máy tấn công không gửi gói tin ACK trả lời mà liên tục gửi các gói tin SYN, khi đó máy bị tấn công tiếp tục chờ khiến tài nguyên hệ thống được sử dụng tăng lên, CPU đạt mức cao như trong hình.