

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG  
KHOA AN TOÀN THÔNG TIN**



**BÁO CÁO BÀI THỰC HÀNH  
HỌC PHẦN: THỰC TẬP CƠ SỞ  
MÃ HỌC PHẦN: INT13147**

**BÀI THỰC HÀNH 2.2  
TÌM HIỂU VÀ CÀI ĐẶT, CẤU HÌNH NIDS**

Sinh viên thực hiện:

B22DCAT251 Đặng Đức Tài

Giảng viên hướng dẫn: TS. Phạm Hoàng Duy

**HỌC KỲ 2 NĂM HỌC 2024-2025**

DANH MỤC CÁC BẢNG BIỂU .....	3
DANH MỤC CÁC HÌNH VẼ .....	3
<b>CHƯƠNG 1. GIỚI THIỆU CHUNG VỀ BÀI THỰC HÀNH</b> .....	4
1.1 Mục đích .....	4
1.2 Tìm hiểu lý thuyết.....	4
<b>1.2.1</b> Tổng quan về các hệ thống phát hiện tấn công & xâm nhập.....	4
<b>1.2.2</b> Kiến trúc và tính năng của một số hệ thống IDS/IPS phổ biến .....	10
<b>CHƯƠNG 2. NỘI DUNG THỰC HÀNH</b> .....	12
2.1 Chuẩn bị môi trường .....	12
2.2 Các bước thực hiện .....	12
<b>2.2.1</b> Cài đặt môi trường .....	12
<b>2.2.2</b> Tạo các luật trong Snort.....	15
<b>2.2.3</b> Thực thi tấn công và phát hiện sử dụng Snort .....	16
<b>TÀI LIỆU THAM KHẢO</b> .....	20

## DANH MỤC CÁC BẢNG BIỂU

Bảng 1. So sánh giữa IDS và IPS .....	4
---------------------------------------	---

## DANH MỤC CÁC HÌNH VẼ

Hình 1 Vị trí của hệ thống IDS/IPS trong sơ đồ mạng .....	6
Hình 2 Sử dụng kết hợp NIDS và HIDS để giám sát lưu lượng mạng và các máy.....	8
Hình 3 Lưu đồ giám sát phát hiện tấn công, xâm nhập dựa trên chữ ký .....	9
Hình 4 Giá trị entropy của IP nguồn của các gói tin từ lưu lượng hợp pháp (phần giá trị cao, đều) và entropy của IP nguồn của các gói tin từ lưu lượng tấn công DDoS (phần giá trị thấp) .....	10
Hình 5 Kiến trúc của Snort NIDS .....	10
Hình 6 Các thành phần chính của OSSEC.....	11
Hình 7 Cài đặt máy Kali Linux.....	13
Hình 8 Cài đặt máy Ubuntu (Snort).....	13
Hình 9 Cài đặt Snort .....	14
Hình 10 Chạy thử Snort .....	14
Hình 11 Kiểm tra file log của Snort.....	15
Hình 12 Tạo luật trong Snort .....	16
Hình 13 Cấu hình file Snort .....	16
Hình 14 Ping từ máy Kali Linux.....	17
Hình 15 Kiểm tra phát hiện ping trên Snort.....	17
Hình 16 Quét cổng dịch vụ (80) trên máy Kali Linux.....	18
Hình 17 Kiểm tra phát hiện rà quét trên Snort.....	18
Hình 18 Tấn công TCP SYN Flood trên máy Kali.....	18
Hình 19 Kiểm tra phát hiện TCP SYN Flood trên máy Snort .....	19
Hình 20 Lọc thông tin trên Snort .....	19

## CHƯƠNG 1. GIỚI THIỆU CHUNG VỀ BÀI THỰC HÀNH

### 1.1 Mục đích

- Tìm hiểu và luyện tập việc cài đặt và vận hành các hệ thống phát hiện xâm nhập cho host (HIDS) và cho mạng (NIDS).
- Luyện tập việc tạo và chỉnh sửa các luật phát hiện tấn công, xâm nhập cho các hệ thống phát hiện xâm nhập thông dụng.

### 1.2 Tìm hiểu lý thuyết

#### 1.2.1 Tổng quan về các hệ thống phát hiện tấn công & xâm nhập

##### 1.2.1.1 Khái niệm về hệ thống phát hiện xâm nhập (IDS) và hệ thống ngăn chặn xâm nhập (IPS)

- Hệ thống phát hiện xâm nhập (Intrusion Detection System - IDS) và hệ thống ngăn chặn xâm nhập (Intrusion Prevention System - IPS) là hai công nghệ quan trọng trong lĩnh vực an toàn thông tin, giúp giám sát và bảo vệ hệ thống trước các cuộc tấn công mạng.
- IDS (Intrusion Detection System): Là hệ thống có chức năng giám sát lưu lượng mạng hoặc hoạt động trên hệ thống để phát hiện các hành vi đáng ngờ hoặc dấu hiệu của cuộc tấn công. IDS chỉ có khả năng cảnh báo mà không can thiệp vào lưu lượng mạng.
- IPS (Intrusion Prevention System): Là hệ thống mở rộng của IDS, không chỉ phát hiện mà còn có thể chặn các cuộc tấn công bằng cách tự động thực hiện các biện pháp ngăn chặn, chẳng hạn như chặn địa chỉ IP độc hại hoặc hủy bỏ các gói tin đáng ngờ.
- Dưới đây là một số phân tích về IDS và IPS:

Tiêu chí	IDS (Intrusion Detection System)	IPS (Intrusion Prevention System)
Chức năng	Phát hiện và cảnh báo cuộc tấn công	Phát hiện và ngăn chặn cuộc tấn công
Cách thức hoạt động	Hoạt động như hệ thống giám sát	Hoạt động như một tường lửa nâng cao
Ảnh hưởng đến lưu lượng mạng	Không làm gián đoạn lưu lượng mạng	Có thể làm gián đoạn mạng khi chặn lưu lượng đáng ngờ
Vị trí triển khai	Đặt sau tường lửa để phân tích lưu lượng	Đặt trên đường truyền để kiểm soát lưu lượng dữ liệu

Bảng 1. So sánh giữa IDS và IPS

##### 1.2.1.2 Mục đích và vai trò của IDS/IPS trong bảo mật hệ thống

- Hệ thống phát hiện xâm nhập (IDS) và hệ thống ngăn chặn xâm nhập (IPS) đóng vai trò quan trọng trong việc bảo vệ hệ thống trước các mối đe dọa an ninh mạng. Chúng không chỉ giúp giám sát, phát hiện mà còn hỗ trợ kiểm soát và ngăn chặn các cuộc tấn công, đảm bảo tính an toàn và ổn định của hệ thống. Các chức năng chính của IDS và IPS bao gồm:

#### A. Phát hiện các mối đe dọa bảo mật

- IDS hoạt động bằng cách giám sát lưu lượng mạng để phát hiện các dấu hiệu bất thường như quét cổng, tấn công từ chối dịch vụ (DDoS), khai thác lỗ hổng hoặc truy cập trái phép. Khi phát hiện sự cố, IDS sẽ gửi cảnh báo đến quản trị viên để có biện pháp xử lý kịp thời.
- Trong khi đó, IPS không chỉ dừng lại ở việc phát hiện mà còn có khả năng tự động ngăn chặn các cuộc tấn công. Hệ thống này có thể vô hiệu hóa nguồn tấn công bằng cách chặn địa chỉ IP độc hại hoặc ngăn chặn các gói tin nguy hiểm trước khi chúng có thể gây hại cho hệ thống.

#### B. Tăng cường bảo mật hệ thống

- Một trong những lợi ích quan trọng của IDS là hỗ trợ quản trị viên trong quá trình điều tra các cuộc tấn công. Hệ thống này ghi nhận nhật ký chi tiết và đưa ra cảnh báo kịp thời, giúp các chuyên gia bảo mật phân tích và đưa ra biện pháp khắc phục phù hợp.
- Mặt khác, IPS đóng vai trò chủ động hơn khi có thể ngăn chặn các mối đe dọa ngay từ đầu, giảm thiểu tối đa nguy cơ bị tấn công và giúp hệ thống duy trì trạng thái an toàn trước những cuộc xâm nhập trái phép.

#### C. Giám sát và kiểm soát lưu lượng mạng

- Hệ thống IDS có khả năng phân tích lưu lượng mạng để nhận diện các hoạt động đáng ngờ mà tường lửa (firewall) có thể không phát hiện được. Điều này giúp tăng cường lớp bảo mật và đảm bảo hệ thống luôn trong trạng thái giám sát chặt chẽ.
- Trong khi đó, IPS không chỉ phát hiện mà còn có thể lọc và kiểm soát lưu lượng mạng theo các chính sách bảo mật được thiết lập sẵn. Điều này giúp bảo vệ hệ thống khỏi các mối đe dọa tiềm ẩn, đồng thời tối ưu hóa hiệu suất mạng.

#### D. Hỗ trợ tuân thủ các tiêu chuẩn bảo mật

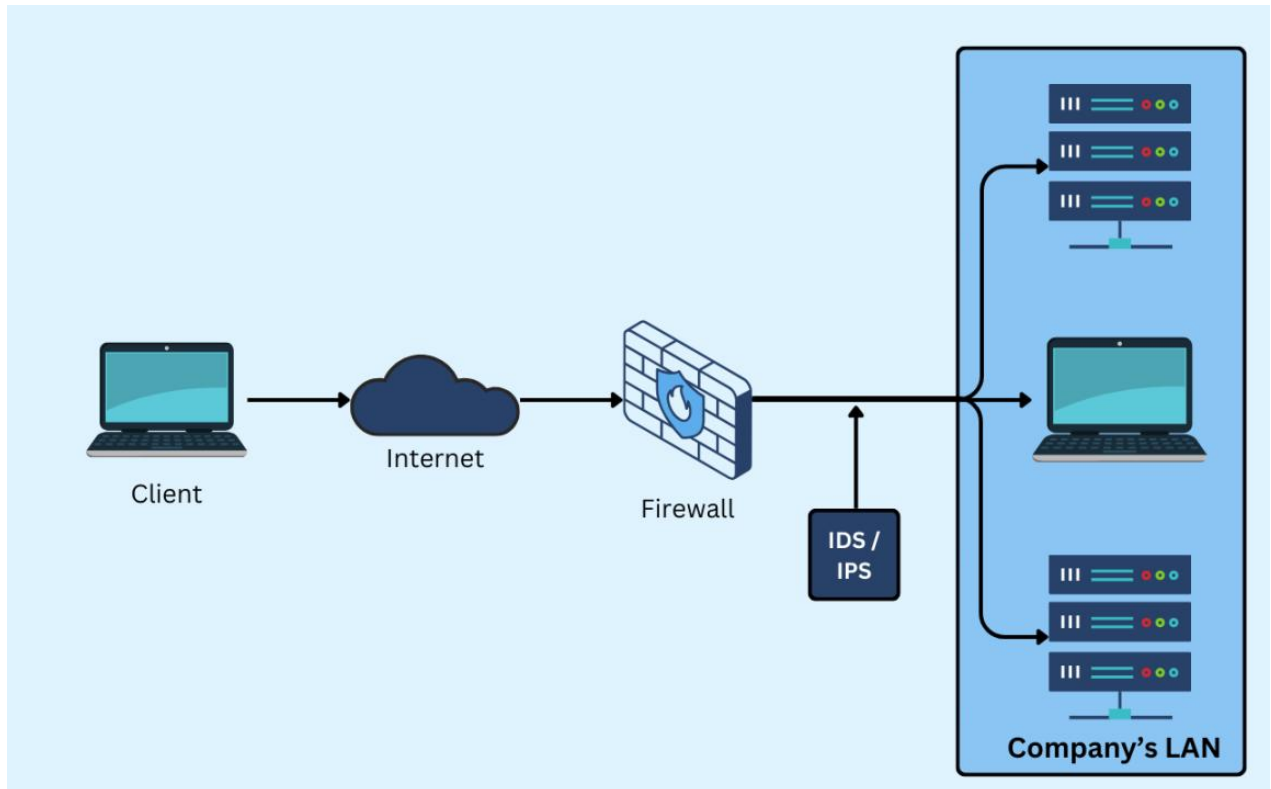
- Nhiều tổ chức yêu cầu triển khai IDS và IPS để đáp ứng các tiêu chuẩn bảo mật như ISO 27001, PCI DSS, NIST. Những tiêu chuẩn này quy định các biện pháp bảo vệ dữ liệu quan trọng, giúp doanh nghiệp và tổ chức duy trì tính bảo mật, toàn vẹn và sẵn sàng của hệ thống thông tin.

#### E. Bảo vệ hệ thống trước các cuộc tấn công zero-day

- Hệ thống IDS có thể sử dụng phương pháp phát hiện dựa trên bất thường (Anomaly-based Detection) để nhận diện các hành vi đáng ngờ mà chưa có trong cơ sở dữ liệu

chữ ký. Điều này đặc biệt hữu ích trong việc phát hiện các cuộc tấn công chưa từng được biết đến trước đây.

- Bên cạnh đó, IPS có thể kết hợp với các giải pháp bảo mật khác như tường lửa, hệ thống phân tích hành vi và trí tuệ nhân tạo để giảm thiểu tác động của các cuộc tấn công zero-day, giúp bảo vệ hệ thống trước những mối đe dọa chưa có dấu hiệu nhận diện rõ ràng.



Hình 1 Vị trí của hệ thống IDS/IPS trong sơ đồ mạng

#### 1.2.1.3 Phân loại hệ thống phát hiện xâm nhập

##### A. Các phương pháp phân loại

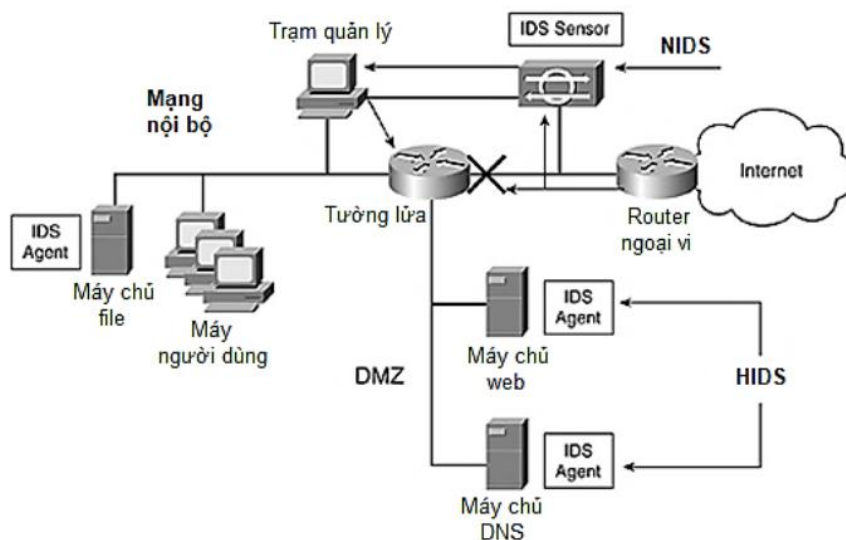
- Hệ thống phát hiện và ngăn chặn xâm nhập (IDS & IPS) có thể được phân loại theo hai tiêu chí chính, bao gồm phân loại theo nguồn dữ liệu và phân loại theo phương pháp phân tích dữ liệu. Theo nguồn dữ liệu, có hai loại hệ thống phát hiện xâm nhập phổ biến là hệ thống phát hiện xâm nhập mạng (NIDS – Network-based IDS) và hệ thống phát hiện xâm nhập cho máy (HIDS – Host-based IDS).
- NIDS là hệ thống giám sát và phân tích lưu lượng mạng để phát hiện các dấu hiệu tấn công, xâm nhập. Hệ thống này thường được triển khai tại các cổng mạng hoặc các phân đoạn mạng, nơi nó có thể quan sát toàn bộ lưu lượng đi qua. Ưu điểm của NIDS là khả năng giám sát và phát hiện các cuộc tấn công trên phạm vi toàn mạng. Tuy nhiên, hệ thống này gặp khó khăn khi phải xử lý lưu lượng mạng lớn, đặc biệt là khi dữ liệu được mã hóa hoặc khi các cuộc tấn công diễn ra trên các máy không tạo ra lưu lượng đáng kể qua cổng mạng.

- Ngược lại, HIDS được cài đặt trực tiếp trên từng máy để giám sát các sự kiện và hoạt động bên trong hệ thống. Nhờ đó, HIDS có khả năng phát hiện chính xác các hành vi xâm nhập hoặc lạm dụng xảy ra trên từng thiết bị cụ thể. Tuy nhiên, nhược điểm của HIDS là phải triển khai trên từng máy, gây tốn kém chi phí cài đặt và bảo trì, đặc biệt trong các hệ thống mạng lớn.
- Bên cạnh cách phân loại theo nguồn dữ liệu, các hệ thống IDS còn được phân loại theo phương pháp phân tích dữ liệu. Hai kỹ thuật chính được sử dụng trong phát hiện xâm nhập bao gồm phát hiện dựa trên chữ ký (signature-based detection) và phát hiện dựa trên bất thường (anomaly-based detection).
- Phát hiện xâm nhập dựa trên chữ ký là phương pháp so sánh dữ liệu với các mẫu tấn công đã biết để nhận diện các hành vi xâm nhập. Kỹ thuật này có độ chính xác cao đối với các mối đe dọa đã được xác định từ trước, nhưng lại không thể phát hiện các cuộc tấn công mới chưa có trong cơ sở dữ liệu. Trong khi đó, phát hiện xâm nhập dựa trên bất thường sử dụng các mô hình học máy hoặc thống kê để nhận diện các hoạt động không bình thường so với hành vi thông thường của hệ thống. Phương pháp này có khả năng phát hiện các cuộc tấn công mới, nhưng có thể gặp vấn đề về tỷ lệ cảnh báo sai cao.

## B. NIDS và HIDS

- Hệ thống phát hiện xâm nhập mạng (NIDS – Network-based IDS)
  - NIDS là hệ thống giám sát cổng mạng, thu thập và phân tích lưu lượng mạng để phát hiện các cuộc tấn công hoặc xâm nhập trái phép. Hệ thống này có thể được triển khai tại cổng vào hệ thống mạng từ Internet hoặc trong từng phân đoạn mạng để theo dõi các hoạt động đáng ngờ.
  - Hiện nay, có nhiều hệ thống NIDS được phát triển dưới dạng phần cứng hoặc phần mềm, bao gồm cả các sản phẩm thương mại và mã nguồn mở. Một số hệ thống NIDS phổ biến có thể kể đến như Check Point IPS, McAfee Network Security Platform, Snort, Bro và Suricata. Trong đó, Snort là một trong những hệ thống phát hiện xâm nhập mạng được sử dụng rộng rãi nhất. Đây là một NIDS mã nguồn mở, miễn phí, hỗ trợ nhiều nền tảng hệ điều hành và cung cấp một bộ quy tắc phát hiện phong phú với khoảng 3000 luật để giám sát và phát hiện hầu hết các dạng tấn công mạng đã biết. Ngoài ra, Snort cho phép người dùng tùy chỉnh bộ quy tắc bằng cách thêm, bớt hoặc chỉnh sửa các luật phát hiện xâm nhập.
  - Kiến trúc của Snort NIDS bao gồm nhiều thành phần quan trọng, trong đó mô-đun phát hiện (detection engine) đóng vai trò cốt lõi. Thành phần này thực hiện việc đối sánh dữ liệu gói tin với các quy tắc đã thiết lập để nhận diện và cảnh báo các cuộc tấn công hoặc hành vi xâm nhập đáng ngờ.
- Hệ thống phát hiện xâm nhập cho máy (HIDS – Host-based IDS)

- HIDS là hệ thống giám sát các hoạt động trên một máy cụ thể, thu thập và phân tích các sự kiện xảy ra trong hệ thống hoặc trên các dịch vụ đang chạy để phát hiện các cuộc tấn công, xâm nhập hoặc hành vi lạm dụng.
- Một số hệ thống HIDS phổ biến bao gồm IBM QRadar, Tripwire, OSSEC, Security Onion và Wazuh. Trong đó, OSSEC là một trong những hệ thống phát hiện xâm nhập dựa trên máy được sử dụng rộng rãi nhất. OSSEC là một HIDS mã nguồn mở, miễn phí, bao gồm một máy chủ OSSEC (OSSEC Server) và các tác nhân OSSEC (OSSEC Agents) được cài đặt trên các thiết bị cần giám sát. Các OSSEC Agents có nhiệm vụ thu thập dữ liệu từ hệ thống và gửi về OSSEC Server để phân tích. Kết quả phát hiện có thể được ghi log và tạo cảnh báo gửi đến quản trị viên.
- Sự kết hợp giữa NIDS và HIDS
  - Trong thực tế, các hệ thống giám sát an ninh thường sử dụng kết hợp cả NIDS và HIDS để đảm bảo khả năng phát hiện và phản ứng hiệu quả trước các cuộc tấn công mạng. Một hệ thống kết hợp có thể bao gồm NIDS để giám sát lưu lượng tại cổng mạng và HIDS để giám sát các máy chủ quan trọng như máy chủ file, máy chủ web và máy chủ DNS.
  - Các IDS Agents được cài đặt trên từng máy chủ để theo dõi hoạt động trong hệ thống, trong khi một trạm quản lý tập trung sẽ thu thập và phân tích dữ liệu từ cả NIDS và HIDS. Trạm quản lý này giúp đưa ra quyết định cuối cùng về việc phát hiện và xử lý các mối đe dọa bảo mật, từ đó nâng cao hiệu quả giám sát và bảo vệ hệ thống.



Hình 2 Sử dụng kết hợp NIDS và HIDS để giám sát lưu lượng mạng và các máy

#### 1.2.1.4 Các kỹ thuật phát hiện xâm nhập

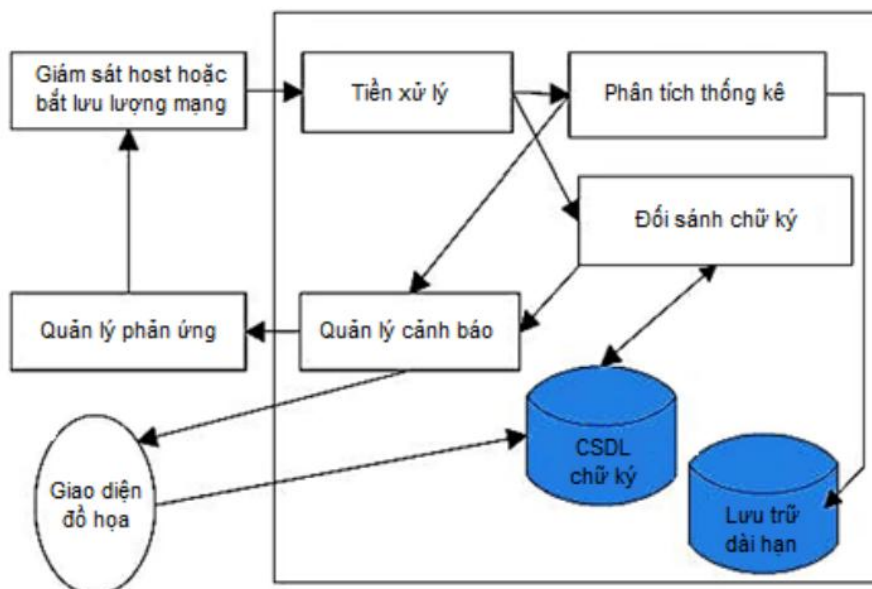
- Phát hiện xâm nhập là quá trình giám sát, thu thập và phân tích dữ liệu nhằm xác định các dấu hiệu tấn công hoặc xâm nhập hệ thống. Để làm được điều này, trước hết cần xây dựng cơ sở dữ liệu về các dấu hiệu tấn công đã biết hoặc thiết lập hồ sơ mô tả các hành vi bình thường của hệ thống. Dựa trên phương pháp tiếp cận này, có hai kỹ



thuật chính được sử dụng: phát hiện xâm nhập dựa trên chữ ký và phát hiện xâm nhập dựa trên bất thường. Dưới đây là chi tiết hai kỹ thuật trên:

- **Phát Hiện Xâm Nhập Dựa Trên Chữ Ký**

- Phương pháp này hoạt động dựa trên cơ sở dữ liệu chứa các chữ ký hoặc dấu hiệu của các loại tấn công đã biết. Thông thường, những chữ ký này được nhận dạng và mã hóa thủ công dưới dạng các quy tắc phát hiện. Khi triển khai, hệ thống sẽ sử dụng cơ sở dữ liệu chữ ký để giám sát hoạt động của hệ thống hoặc mạng và cảnh báo nếu phát hiện dấu hiệu trùng khớp với các mẫu tấn công có sẵn.
- Ưu điểm của phương pháp này là khả năng phát hiện hiệu quả các cuộc tấn công đã biết, tốc độ xử lý nhanh và yêu cầu tài nguyên tính toán thấp. Nhờ đó, phát hiện xâm nhập dựa trên chữ ký được ứng dụng rộng rãi trong thực tế. Tuy nhiên, nhược điểm lớn nhất của phương pháp này là không thể phát hiện các cuộc tấn công mới hoặc các biến thể chưa có trong cơ sở dữ liệu. Ngoài ra, việc xây dựng và cập nhật cơ sở dữ liệu chữ ký đòi hỏi nhiều công sức từ các chuyên gia bảo mật.

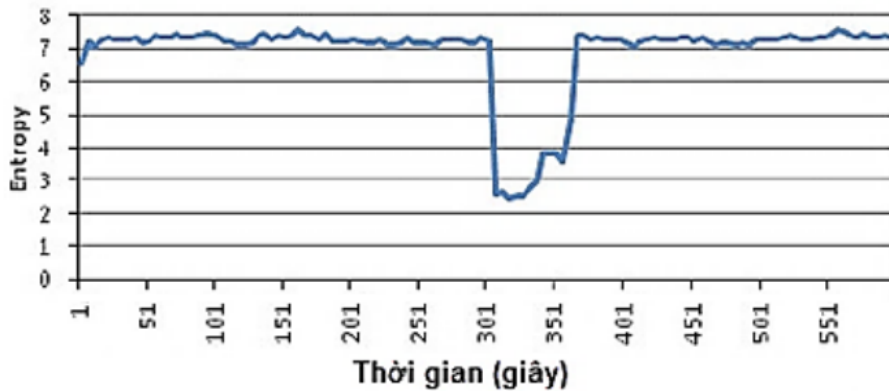


*Hình 3 Lưu đồ giám sát phát hiện tấn công, xâm nhập dựa trên chữ ký*

- **Phát Hiện Xâm Nhập Dựa Trên Bất Thường**

- Phương pháp này dựa trên giả định rằng các hành vi tấn công thường liên quan đến sự bất thường trong hoạt động hệ thống. Quá trình triển khai bao gồm hai giai đoạn chính:
- Giai đoạn huấn luyện: Hệ thống theo dõi và ghi nhận hoạt động bình thường của đối tượng trong một khoảng thời gian đủ dài để thu thập dữ liệu làm nền tảng huấn luyện.
- Giai đoạn phát hiện: Hệ thống tiếp tục giám sát và so sánh hành vi hiện tại với hồ sơ đã thiết lập. Nếu phát hiện sự khác biệt đáng kể, hệ thống sẽ đưa ra cảnh báo.
- Có nhiều phương pháp xử lý và phân tích dữ liệu để xây dựng hồ sơ phát hiện bất thường như phân tích thống kê, khai phá dữ liệu, học máy và phân tích tương quan. Ví dụ, khi phân tích giá trị entropy của địa chỉ IP nguồn trong các gói tin, có thể thấy

sự khác biệt rõ ràng giữa lưu lượng bình thường và lưu lượng trong các cuộc tấn công DDoS. Dựa vào sự thay đổi đột ngột của entropy, hệ thống có thể phát hiện tấn công DDoS một cách hiệu quả.



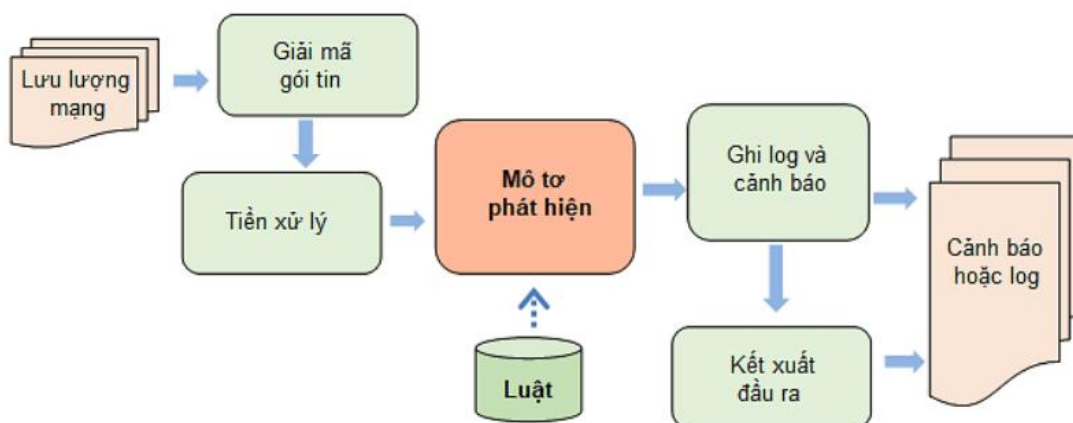
Hình 4 Giá trị entropy của IP nguồn của các gói tin từ lưu lượng hợp pháp (phần giá trị cao, đều) và entropy của IP nguồn của các gói tin từ lưu lượng tấn công DDoS (phần giá trị thấp)

- Ưu điểm lớn nhất của phương pháp này là khả năng phát hiện các cuộc tấn công mới mà không cần thông tin trước về chúng. Tuy nhiên, nó cũng tồn tại một số nhược điểm. Do không phải tất cả hành vi bất thường đều là tấn công, phương pháp này thường có tỷ lệ cảnh báo sai cao hơn so với phát hiện dựa trên chữ ký. Bên cạnh đó, việc xây dựng hồ sơ và phân tích hành vi yêu cầu nhiều tài nguyên tính toán. Mặc dù vậy, đây vẫn là một lĩnh vực nghiên cứu quan trọng nhằm cải thiện khả năng phát hiện xâm nhập, giảm tỷ lệ cảnh báo sai và tối ưu hóa hiệu suất hệ thống.

## 1.2.2 Kiến trúc và tính năng của một số hệ thống IDS/IPS phổ biến

### 1.2.2.1 Snort

- Kiến trúc
  - Snort là một hệ thống phát hiện xâm nhập mạng mã nguồn mở, hoạt động dựa trên phương pháp phát hiện theo chữ ký. Nó giám sát lưu lượng mạng và so sánh với các mẫu tấn công đã biết để phát hiện các mối đe dọa. Snort có thể được cấu hình để ghi lại lưu lượng hoặc ngăn chặn các kết nối đáng ngờ.



Hình 5 Kiến trúc của Snort NIDS

- Tính năng chính:
  - Phát hiện dựa trên chữ ký với cơ sở dữ liệu mẫu tấn công phong phú.
  - Khả năng ghi lại và phân tích lưu lượng mạng.
  - Cấu hình linh hoạt cho các quy tắc phát hiện.

#### 1.2.2.2 Suricata

- Kiến trúc
  - Suricata là một hệ thống IDS/IPS mã nguồn mở hiệu suất cao, cung cấp khả năng phát hiện dựa trên cả chữ ký và hành vi. Nó hỗ trợ xử lý đa luồng và kiểm tra gói tin sâu, cho phép phân tích các giao thức tầng ứng dụng để tăng cường bảo mật.
- Tính năng chính:
  - Hỗ trợ phát hiện dựa trên chữ ký và hành vi.
  - Sử dụng ngôn ngữ quy tắc tương tự như Snort, nhưng có khả năng mở rộng hơn
  - Hỗ trợ thực thi quy tắc dựa trên nhiều luồng, cải thiện hiệu suất
  - Có thể thực hiện các chức năng phân tích lưu lượng mạng, bao gồm cả việc tái lập lưu lượng.
  - Hỗ trợ nhiều đầu vào như pcap, Netflow, IPFW, NFQUEUE

#### 1.2.2.3 Zeek (Bro IDS)

- Kiến trúc
  - Zeek, trước đây gọi là Bro, là một hệ thống phân tích lưu lượng mạng mã nguồn mở, tập trung vào việc ghi lại và phân tích chi tiết các hoạt động mạng. Khác với Snort và Suricata, Zeek không chỉ dựa trên chữ ký mà còn cung cấp khả năng phân tích hành vi và phát hiện bất thường
- Tính năng chính:
  - Phân tích sâu các giao thức tầng ứng dụng.
  - Ghi lại chi tiết các sự kiện mạng để hỗ trợ điều tra và giám sát.
  - Cung cấp nền tảng mạnh mẽ cho việc phát hiện bất thường và săn lùng mối đe dọa.

#### 1.2.2.4 OSSEC

- Kiến trúc
  - OSSEC là một hệ thống phát hiện xâm nhập dựa trên máy chủ (HIDS) mã nguồn mở, tập trung vào việc giám sát và phân tích các tệp nhật ký, kiểm tra tính toàn vẹn của tệp, phát hiện rootkit và cảnh báo thời gian thực



Hình 6 Các thành phần chính của OSSEC

- Tính năng chính:
  - Giám sát và phân tích tệp nhật ký từ nhiều nguồn khác nhau.
  - Kiểm tra tính toàn vẹn của tệp để phát hiện các thay đổi không mong muốn.
  - Phát hiện rootkit và các mối đe dọa khác trên máy chủ.
  - Cảnh báo thời gian thực và khả năng phản ứng tự động.

#### 1.2.2.5 Wazuh

- Kiến trúc
  - Wazuh là một nền tảng bảo mật mã nguồn mở, phát triển từ OSSEC, cung cấp khả năng giám sát bảo mật toàn diện cho cả máy chủ và điểm cuối. Nó tích hợp với các công cụ như Elasticsearch và Kibana để cung cấp giao diện quản lý và phân tích mạnh mẽ.
- Tính năng chính
  - Giám sát tính toàn vẹn của tệp và phân tích tệp nhật ký.
  - Phát hiện rootkit và giám sát cấu hình.
  - Tích hợp với Elasticsearch và Kibana để hiển thị và phân tích dữ liệu bảo mật.
  - Hỗ trợ triển khai linh hoạt, từ mô hình đơn lẻ đến mô hình phân tán.
- Mỗi hệ thống IDS/IPS trên có những ưu điểm và hạn chế riêng, phù hợp với các nhu cầu và môi trường triển khai khác nhau. Việc lựa chọn hệ thống phù hợp cần dựa trên yêu cầu cụ thể của tổ chức và khả năng tích hợp với các công cụ hiện có.

## CHƯƠNG 2. NỘI DUNG THỰC HÀNH

### 2.1 Chuẩn bị môi trường

- 01 máy tính (máy thật hoặc máy ảo) chạy Linux với RAM tối thiểu 2GB, 10GB đĩa cứng có kết nối mạng (LAN hoặc Internet).
- 01 máy tính (máy thật hoặc máy ảo) chạy Kali Linux (bản 2021 trở lên)
- Bộ phần mềm Snort tải tại <https://www.snort.org/downloads>

### 2.2 Các bước thực hiện

#### 2.2.1 Cài đặt môi trường

- Cài đặt, đổi tên máy ảo theo format Máy Kali Linux được đổi tên thành <Mã SV-Tên SV>-Kali và máy cài Snort thành <Mã SV-Tên SV>-Snort. Các máy có địa chỉ IP và kết nối mạng LAN (cùng card mạng trên VMware).
- Cài đặt máy Kali Linux

```
Command Prompt
C:\Users\jayce>echo %DATE% %TIME% Dang Duc Tai B22DCAT251
21/03/2025 15:54:54.72 Dang Duc Tai B22DCAT251
C:\Users\jayce>

(jayce@B22DCAT251-DangDucTai-Kali)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:b6:9d:38 brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.3/24 brd 192.168.100.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:feb6:9d38/64 scope link proto kernel_ll
        valid_lft forever preferred_lft forever

(jayce@B22DCAT251-DangDucTai-Kali)-[~]
$
```

Hình 7 Cài đặt máy Kali Linux

- Cài đặt máy Ubuntu (Snort)

```
Command Prompt
C:\Users\jayce>echo %DATE% %TIME% Dang Duc Tai B22DCAT251
21/03/2025 15:54:54.72 Dang Duc Tai B22DCAT251
C:\Users\jayce>

jayce@B22DCAT251-DangDucTai-Snort:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:2c:5d:37 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.100.147/24 brd 192.168.100.255 scope global noprefixroute ens33
        valid_lft forever preferred_lft forever
    inet 192.168.100.4/24 brd 192.168.100.255 scope global secondary dynamic noprefixroute ens33
        valid_lft 1786sec preferred_lft 1786sec
jayce@B22DCAT251-DangDucTai-Snort:~$
```

Hình 8 Cài đặt máy Ubuntu (Snort)

- Cài đặt snort trên máy Ubuntu

*sudo apt install snort*

```
Command Prompt
C:\Users\jaye>echo %DATE% %TIME% Dang Duc Tai B22DCAT251
16/03/2025 22:46:02.13 Dang Duc Tai B22DCAT251
C:\Users\jaye>

jaye@B22DCAT251-DangDucTai-Snort:~$ sudo apt install snort
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  liblvm17t64
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  libdaq2t64 libdumbnet1 liblua5.1-2 liblua5.1-common libnetfilter-queue1 libpcrc3 net-tools oinkmaster snort-common snort-common-libraries snort-rules-default
Suggested packages:
  snort-doc
The following NEW packages will be installed:
  libdaq2t64 libdumbnet1 liblua5.1-2 liblua5.1-common libnetfilter-queue1 libpcrc3 net-tools oinkmaster snort snort-common snort-common-libraries snort-rules-def
0 upgraded, 12 newly installed, 0 to remove and 21 not upgraded.
Need to get 2,869 kB of archives.
After this operation, 12.2 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://vn.archive.ubuntu.com/ubuntu noble/universe amd64 liblua5.1-2-common all 2.1.0+git20231223.c525bcb+dfsg-1 [49.2 kB]
Get:2 http://vn.archive.ubuntu.com/ubuntu noble/universe amd64 liblua5.1-2 amd64 2.1.0+git20231223.c525bcb+dfsg-1 [275 kB]
Get:3 http://vn.archive.ubuntu.com/ubuntu noble/universe amd64 libpcrc3 amd64 2:8.39-15build1 [248 kB]
Get:4 http://vn.archive.ubuntu.com/ubuntu noble/universe amd64 snort-common-libraries amd64 2.9.20-0+deb11u1ubuntu1 [899 kB]
Get:5 http://vn.archive.ubuntu.com/ubuntu noble/universe amd64 snort-rules-default all 2.9.20-0+deb11u1ubuntu1 [144 kB]
Get:6 http://vn.archive.ubuntu.com/ubuntu noble/universe amd64 snort-common all 2.9.20-0+deb11u1ubuntu1 [47.7 kB]
Get:7 http://vn.archive.ubuntu.com/ubuntu noble/main amd64 net-tools amd64 2.10-0.1ubuntu4 [284 kB]
Get:8 http://vn.archive.ubuntu.com/ubuntu noble/universe amd64 libdumbnet1 amd64 1.17.0-1ubuntu2 [30.7 kB]
Get:9 http://vn.archive.ubuntu.com/ubuntu noble/universe amd64 libnetfilter-queue1 amd64 1.0.5-4build1 [15.1 kB]
Get:10 http://vn.archive.ubuntu.com/ubuntu noble/universe amd64 libdaq2t64 amd64 2.0.7-5.1build3 [92.9 kB]
Get:11 http://vn.archive.ubuntu.com/ubuntu noble/universe amd64 snort amd64 2.9.20-0+deb11u1ubuntu1 [791 kB]
Get:12 http://vn.archive.ubuntu.com/ubuntu noble/universe amd64 oinkmaster all 2.0.4.2 [71.9 kB]
Fetched 2,869 kB in 2s (1,572 kB/s)
Preconfiguring packages ...
Snort configuration: interface default not set, using 'ens33'
Selecting previously unselected package liblua5.1-2-common.
(Reading database ... 167871 files and directories currently installed.)
Preparing to unpack .../00-liblua5.1-2-common_2.1.0+git20231223.c525bcb+dfsg-1_all.deb ...
Unpacking liblua5.1-2-common (2.1.0+git20231223.c525bcb+dfsg-1) ...
```

Hình 9 Cài đặt Snort

- Chạy thử snort

*sudo snort -v -I ens33*

```
Command Prompt
C:\Users\jaye>echo %DATE% %TIME% Dang Duc Tai B22DCAT251
16/03/2025 22:46:02.13 Dang Duc Tai B22DCAT251
C:\Users\jaye>

jaye@B22DCAT251-DangDucTai-Snort:~$ sudo snort -v -i ens33
Running in packet dump mode

--== Initializing Snort ==--
Initializing Output Plugins!
pcap DAQ configured to passive.
Acquiring network traffic from "ens33".
Decoding Ethernet

--== Initialization Complete ==--

--> Snort! <*-
o" )~ Version 2.9.20 GRE (Build 82)
    '~~ By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
        Copyright (C) 2014-2022 Cisco and/or its affiliates. All rights reserved.
        Copyright (C) 1998-2013 Sourcefire, Inc., et al.
        Using libpcap version 1.10.4 (with TPACKET_V3)
        Using PCRE version: 8.39 2016-06-14
        Using ZLIB version: 1.3

Commencing packet processing (pid=13518)

03/16-22:49:14.950241 192.168.127.136:55300 -> 185.125.190.56:123
UDP TTL:64 TOS:0x0 ID:20517 IpLen:20 DgmLen:76 DF
Len: 48
=====

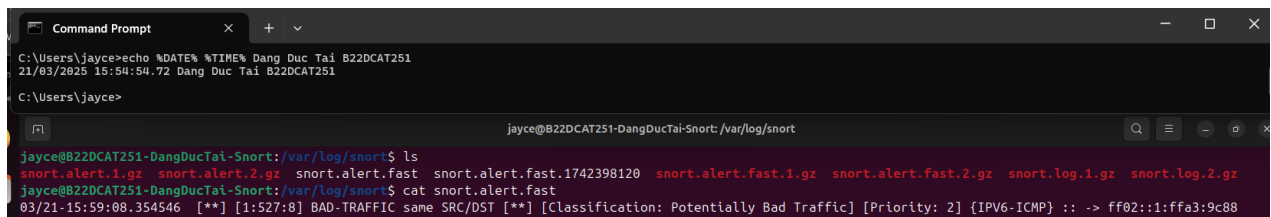
WARNING: No preprocessors configured for policy 0.
03/16-22:49:15.255985 185.125.190.56:123 -> 192.168.127.136:55300
UDP TTL:128 TOS:0x0 ID:1735 IpLen:20 DgmLen:76
Len: 48
=====

WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
```

Hình 10 Chạy thử Snort

- Kiểm tra log để đảm bảo snort tại đường dẫn */var/log/snort*





The image shows two overlapping windows. The top window is a Windows Command Prompt with the following text:  
C:\Users\jayce>echo %DATE% %TIME% Dang Duc Tai B22DCAT251  
21/03/2025 15:54:54.72 Dang Duc Tai B22DCAT251  
C:\Users\jayce>  
The bottom window is a terminal window titled 'jayce@B22DCAT251-DangDucTai-Snort: /var/log/snort'. It shows the following commands and output:  
jayce@B22DCAT251-DangDucTai-Snort:/var/log/snort\$ ls  
snort.alert.1.gz snort.alert.2.gz snort.alert.fast snort.alert.fast.1742398120 snort.alert.fast.1.gz snort.alert.fast.2.gz snort.log.1.gz snort.log.2.gz  
jayce@B22DCAT251-DangDucTai-Snort:/var/log/snort\$ cat snort.alert.fast  
03/21-15:59:08.354546 [\*\*] [1:527:8] BAD-TRAFFIC same SRC/DST [\*\*] [Classification: Potentially Bad Traffic] [Priority: 2] {IPV6-ICMP} :: -> ff02::1:ffa3:9c88

Hình 11 Kiểm tra file log của Snort

### 2.2.2 Tạo các luật trong Snort

- Tạo các luật trong snort để phát hiện 3 dạng quét, tấn công hệ thống.

- Rule 1: Ping detect

Phát hiện các gói tin ping từ bất kỳ một máy nào gửi đến máy chạy Snort. Hiện thị thông điệp khi phát hiện: “<Mã SV-Tên SV>-Snort phát hiện có các gói Ping gửi đến.”

*alert icmp any any -> \$HOME\_NET any (msg: “B22DCAT251-TaiDD phat hien co cac goi ping icmp gui den”; sid: 1000001; rev:1;)*

- Rule 2: TCP scan port 80 detect

Phát hiện các gói tin rà quét từ bất kỳ một máy nào gửi đến máy chạy Snort trên cổng 80. Hiện thị thông điệp khi phát hiện: “<Mã SV-Tên SV>-Snort phát hiện có các gói tin rà quét trên cổng 80.”

*alert icmp any any -> \$HOME\_NET 80 (msg: “B22DCAT251-TaiDD phat hien co cac goi tin ra quet tren cong 80”; sid: 1000002; rev:1;)*

- Rule 3: TCP SYN FLOOD detect

Phát hiện tấn công TCP SYN Flood từ bất kỳ một máy nào gửi đến máy chạy Snort. Hiện thị thông điệp khi phát hiện: “<Mã SV-Tên SV>-Snort phát hiện đang bị tấn công TCP SYN Flood.”

*alert icmp any any -> \$HOME\_NET any (msg: “B22DCAT251-TaiDD phat hien dang bi tan cong TCP SYN Flood”; sid: 1000003; rev:1;)*

```
Command Prompt
C:\Users\jayce>echo %DATE% %TIME% Dang Duc Tai B22DCAT251
19/03/2025 22:11:15.35 Dang Duc Tai B22DCAT251
C:\Users\jayce>

GNU nano 7.2 /etc/snort/rules/local.rules
# $Id: Local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
# -----
# LOCAL RULES
# -----
# This file intentionally does not come with signatures. Put your local
# additions here.

#Rule 1: Ping detect
alert icmp any any -> $HOME_NET any (msg: "B22DCAT251-TaiDD phat hien co cac goi ping icmp gui den"; sid: 1000001; rev:1;)

#Rule 2: TCP scan port 80 detect
alert tcp any any -> $HOME_NET 80 (msg: "B22DCAT251-TaiDD phat hien co cac goi tin ra quet tren cong 80"; sid:1000002; rev:1;)

#Rule 3: TCP SYN FLOOD detect
alert tcp any any -> $HOME_NET any (msg: "B22DCAT251-TaiDD phat hien dang bi tan cong TCP SYN Flood"; sid:1000003; rev:1;)
```

Hình 12 Tạo luật trong Snort

- Đảm bảo file cấu hình của snort có dòng *include \$RULE\_PATH/local.rules* được bật

```
Command Prompt
C:\Users\jayce>echo %DATE% %TIME% Dang Duc Tai B22DCAT251
19/03/2025 17:12:51.37 Dang Duc Tai B22DCAT251
C:\Users\jayce>

#
# NOTE: All categories are enabled in this conf file
#####

# Note to Debian users: The rules preinstalled in the system
# can be *very* out of date. For more information please read
# the /usr/share/doc/snort-rules-default/README.Debian file

#
# If you install the official VRT Sourcefire rules please review this
# configuration file and re-enable (remove the comment in the first line) those
# rules files that are available in your system (in the /etc/snort/rules
# directory)

# site specific rules
include $RULE_PATH/local.rules

# The include files commented below have been disabled
# because they are not available in the stock Debian
# rules. If you install the Sourcefire VRT please make
# sure you re-enable them again:

#include $RULE_PATH/app-detect.rules
#include $RULE_PATH/attack-responses.rules
#include $RULE_PATH/backdoor.rules
#include $RULE_PATH/bad-traffic.rules
#include $RULE_PATH/blacklist.rules
#include $RULE_PATH/botnet-cnc.rules
#include $RULE_PATH/browser-chrome.rules
#include $RULE_PATH/browser-firefox.rules
```

Hình 13 Cấu hình file Snort

### 2.2.3 Thực thi tấn công và phát hiện sử dụng Snort

- Phát hiện ping
- Từ máy Kali, ping tới máy Snort  
*ping 192.168.100.147*



```
Command Prompt
C:\Users\jayce>echo %DATE% %TIME% Dang Duc Tai B22DCAT251
19/03/2025 22:11:15.35 Dang Duc Tai B22DCAT251
C:\Users\jayce>

(jayce@B22DCAT251-DangDucTai-Kali)-[~]
$ ping -c 4 192.168.100.147
PING 192.168.100.147 (192.168.100.147) 56(84) bytes of data.
64 bytes from 192.168.100.147: icmp_seq=1 ttl=64 time=2.77 ms
64 bytes from 192.168.100.147: icmp_seq=2 ttl=64 time=1.96 ms
64 bytes from 192.168.100.147: icmp_seq=3 ttl=64 time=3.98 ms
64 bytes from 192.168.100.147: icmp_seq=4 ttl=64 time=1.33 ms

— 192.168.100.147 ping statistics —
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 1.329/2.509/3.983/0.992 ms

(jayce@B22DCAT251-DangDucTai-Kali)-[~]
$
```

Hình 14 Ping từ máy Kali Linux

- Trên máy Snort, kiểm tra (Rule 1)  
*cat snort.alert.fast | grep "ping icmp"*

```
Command Prompt
C:\Users\jayce>echo %DATE% %TIME% Dang Duc Tai B22DCAT251
21/03/2025 15:54:54.72 Dang Duc Tai B22DCAT251
C:\Users\jayce>

jayce@B22DCAT251-DangDucTai-Snort:/var/log/snort$ cat snort.alert.fast.1742398120 | grep "ping icmp"
03/19-22:12:07.311542  [**] [1:1000001:1] B22DCAT251-TaiDDo phát hiện có các gói ping icmp gửi đến [**] [Priority: 0] {ICMP} 192.168.100.3 -> 192.168.100.147
03/19-22:12:08.313862  [**] [1:1000001:1] B22DCAT251-TaiDDo phát hiện có các gói ping icmp gửi đến [**] [Priority: 0] {ICMP} 192.168.100.3 -> 192.168.100.147
03/19-22:12:09.316519  [**] [1:1000001:1] B22DCAT251-TaiDDo phát hiện có các gói ping icmp gửi đến [**] [Priority: 0] {ICMP} 192.168.100.3 -> 192.168.100.147
03/19-22:12:10.316774  [**] [1:1000001:1] B22DCAT251-TaiDDo phát hiện có các gói ping icmp gửi đến [**] [Priority: 0] {ICMP} 192.168.100.3 -> 192.168.100.147
```

Hình 15 Kiểm tra phát hiện ping trên Snort

- Phát hiện rà quét tự động
- Trên máy Kali Linux, sử dụng nmap để quét các cổng dịch vụ (80)  
*nmap -sS -p80 192.168.100.147*

```
Command Prompt
C:\Users\jayce>echo %DATE% %TIME% Dang Duc Tai B22DCAT251
19/03/2025 22:11:15.35 Dang Duc Tai B22DCAT251
C:\Users\jayce>

(jayce@B22DCAT251-DangDucTai-Kali)-[~]
$ sudo nmap -sS -p80 192.168.100.147

sudo: unable to resolve host B22DCAT251-DangDucTai-Kali: Temporary failure in name resolution

Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-19 11:18 EDT
Stats: 0:00:00 elapsed; 0 hosts completed (0 up), 1 undergoing ARP Ping Scan
ARP Ping Scan Timing: About 100.00% done; ETC: 11:18 (0:00:00 remaining)
Nmap scan report for 192.168.100.147
Host is up (0.0014s latency).

PORT      STATE SERVICE
80/tcp    open  http
MAC Address: 00:0C:29:2C:5D:37 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 13.84 seconds

(jayce@B22DCAT251-DangDucTai-Kali)-[~]
$
```

Hình 16 Quét cổng dịch vụ (80) trên máy Kali Linux

- Trên máy Snort, kiểm tra kết quả phát hiện rà quét (Rule 2)

*grep "ra quet" snort.alert.fast*

```
Command Prompt
C:\Users\jayce>echo %DATE% %TIME% Dang Duc Tai B22DCAT251
21/03/2025 15:54:54.72 Dang Duc Tai B22DCAT251
C:\Users\jayce>

(jayce@B22DCAT251-DangDucTai-Snort)-[~]
$ grep "192.168.100.3" snort.alert.fast.1742398120 | grep "ra quet"
03/19-22:15:43.625548  [**] [1:1000002:1] B22DCAT251-TaiDD phat hien co cac goi tin ra quet tren cong 80 [**] [Priority: 0] (TCP) 192.168.100.3:51866 -> 192.168.100.147:80
03/19-22:15:43.627492  [**] [1:1000002:1] B22DCAT251-TaiDD phat hien co cac goi tin ra quet tren cong 80 [**] [Priority: 0] (TCP) 192.168.100.3:51866 -> 192.168.100.147:80
03/19-22:15:43.627505  [**] [1:1000002:1] B22DCAT251-TaiDD phat hien co cac goi tin ra quet tren cong 80 [**] [Priority: 0] (TCP) 192.168.100.3:51866 -> 192.168.100.147:80
03/19-22:17:14.633472  [**] [1:1000002:1] B22DCAT251-TaiDD phat hien co cac goi tin ra quet tren cong 80 [**] [Priority: 0] (TCP) 192.168.100.3:36084 -> 192.168.100.147:80
03/19-22:17:14.636122  [**] [1:1000002:1] B22DCAT251-TaiDD phat hien co cac goi tin ra quet tren cong 80 [**] [Priority: 0] (TCP) 192.168.100.3:36084 -> 192.168.100.147:80
03/19-22:18:31.812421  [**] [1:1000002:1] B22DCAT251-TaiDD phat hien co cac goi tin ra quet tren cong 80 [**] [Priority: 0] (TCP) 192.168.100.3:47244 -> 192.168.100.147:80
03/19-22:18:31.813909  [**] [1:1000002:1] B22DCAT251-TaiDD phat hien co cac goi tin ra quet tren cong 80 [**] [Priority: 0] (TCP) 192.168.100.3:47244 -> 192.168.100.147:80
(jayce@B22DCAT251-DangDucTai-Snort)-[~]
$
```

Hình 17 Kiểm tra phát hiện rà quét trên Snort

- Phát hiện tấn công TCP SYN Flood
- Sử dụng công cụ hping3 để thực hiện yêu cầu tấn công TCP SYN Flood trên máy Kali Linux tới máy Snort

*hping3 -c 15000 -d 120 -S -w 64 -p 80 --flood --rand-source 192.168.100.147*

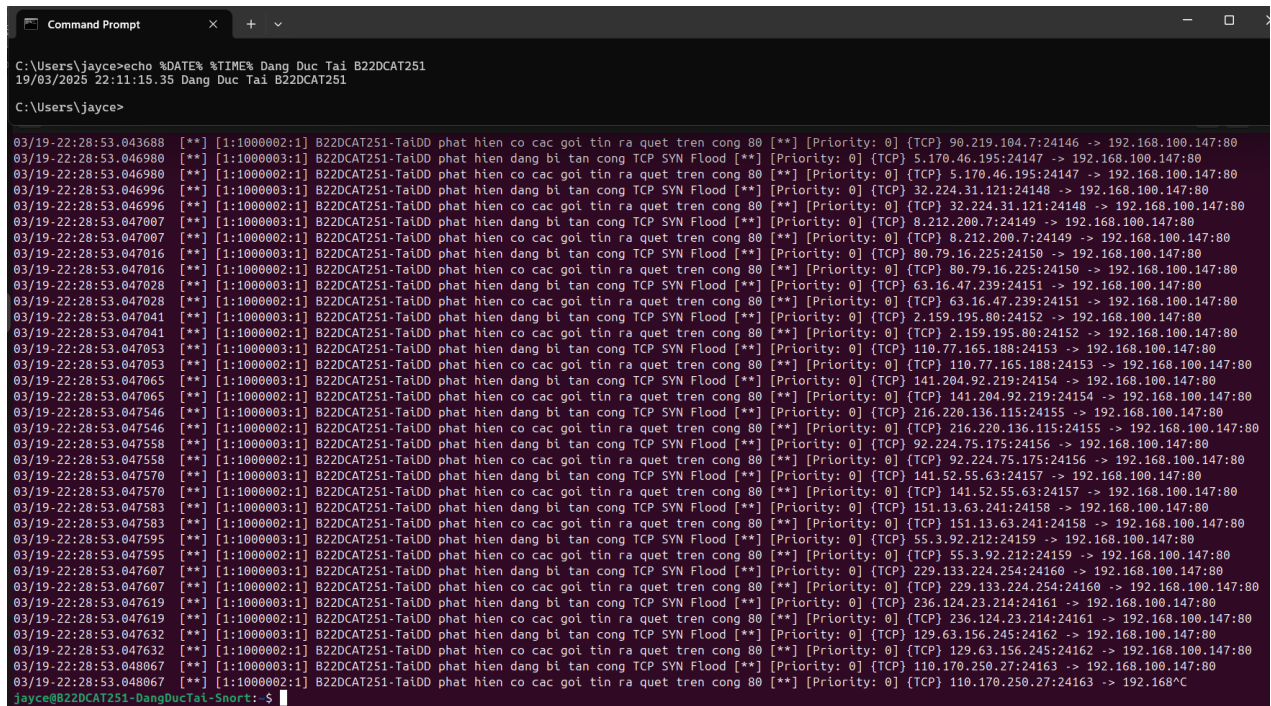
```
Command Prompt
C:\Users\jayce>echo %DATE% %TIME% Dang Duc Tai B22DCAT251
19/03/2025 22:11:15.35 Dang Duc Tai B22DCAT251
C:\Users\jayce>

(jayce@B22DCAT251-DangDucTai-Kali)-[~]
$ sudo hping3 -c 15000 -d 120 -S -w 64 -p 80 --flood --rand-source 192.168.100.147
HPING 192.168.100.147 (eth0 192.168.100.147): S set, 40 headers + 120 data bytes
hping in flood mode, no replies will be shown
^C
— 192.168.100.147 hping statistic —
180472 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms

(jayce@B22DCAT251-DangDucTai-Kali)-[~]
$
```

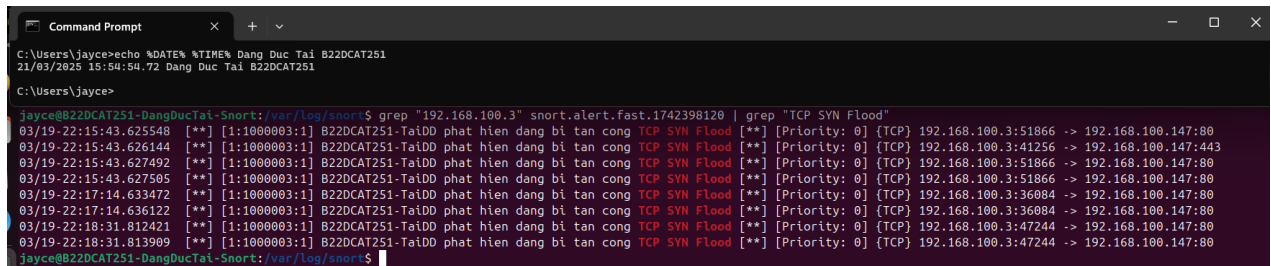
Hình 18 Tấn công TCP SYN Flood trên máy Kali

- Trên máy Snort, kiểm tra kết quả phát hiện TCP SYN Flood trên file log của máy Snort. Từ kết quả, ta có thể thấy 1 lượng lớn request được gửi từ rất nhiều địa chỉ ip khác nhau tới máy Snort.  
→ Rất có thể là 1 cuộc tấn công SYN Flood làm gián đoạn hệ thống phân tán (DDoS)



*Hình 19 Kiểm tra phát hiện TCP SYN Flood trên máy Snort*

- Trên thực tế (Đã biết địa chỉ ip attacker), ta có thể lọc chính xác địa chỉ ip



*Hình 20 Lọc thông tin trên Snort*

## **TÀI LIỆU THAM KHẢO**

- [1] Chương 5, Giáo trình Cơ sở an toàn thông tin, Học viện Công nghệ BVCT, 2020.
- [2] Suricata: <https://suricata.io/documentation/>
- [3] Snort: <https://www.snort.org/#documents>
- [4] OSSEC: <https://www.ossec.net/docs/>
- [5] Wazuh: <https://documentation.wazuh.com/current/index.html>