

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

---



**Môn học: Thực Tập Cơ Sở**  
**Báo Cáo Bài Thực Hành 6**  
**Cài Đặt Cấu Hình HIDS/NIDS**

**Họ và tên:** Trần Thị Thu Phương

**Mã sinh viên:** B21DCAT151

**Nhóm môn học:** 04

**Giảng viên:** Đinh Trường Duy

Hà Nội, 1/2024

## Mục lục

<b>1. Mục đích .....</b>	<b>3</b>
<b>2. Nội dung thực hành .....</b>	<b>3</b>
<b>2.1. Cơ sở lý thuyết.....</b>	<b>3</b>
<b>2.1.1. Tìm hiểu khái quát về các hệ thống phát hiện tấn công, xâm nhập, phân loại các hệ thống phát hiện xâm nhập, các kỹ thuật phát hiện xâm nhập. .3</b>	<b>3</b>
a. Giới thiệu về hệ thống phát hiện tấn công, xâm nhập .....	3
b. Phân loại hệ thống phát hiện tấn công, xâm nhập .....	4
c. Các kỹ thuật phát hiện xâm nhập .....	5
<b>2.1.2. Tìm hiểu về kiến trúc và tính năng của một số hệ thống phát hiện tấn công, xâm nhập: Snort, OSSEC .....</b>	<b>7</b>
2.1.2.1. Snort.....	7
2.1.2.2. Ossec.....	9
<b>2.2. Nội dung thực hành .....</b>	<b>12</b>
2.2.1. Chuẩn bị môi trường.....	12
2.2.2. Các bước thực hiện .....	13
<b>3. Kết luận .....</b>	<b>22</b>
<b>4. Tài liệu tham khảo.....</b>	<b>22</b>

## Danh mục hình ảnh

Hình 2.1.1.a.1. Vị trí hệ thống IDS trong sơ đồ mạng .....	3
Hình 2.1.1.b.1. Các NIDS được bố trí để giám sát phát hiện xâm nhập tại cổng vào và cho từng phân đoạn mạng .....	4
Hình 2.1.1.b.2. Sử dụng kết hợp NIDS và HIDS để giám sát lưu lượng mạng và các host .....	5
Hình 2.1.1.c.1. Lưu đồ giám sát phát hiện tấn công, xâm nhập dựa trên chữ ký .....	6
Hình 2.1.1.c.2. 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) .....	7
Hình 2.1.2.2.1. Kiến trúc của OSSEC .....	11
Hình 4.1.1. Trên máy Kali, thực hiện Ping đến máy cài Snort .....	18
Hình 4.1.2. Trên máy Snort hiện các cảnh báo .....	19
Hình 4.1.3. Kiểm tra địa chỉ IP máy Snort .....	20
Hình 4.2.1. Trên máy Kali, sử dụng công cụ nmap để rà quét máy Snort: nmap -sV -p80 -A 192.168.17.178 .....	20
Hình 4.2.2. Trên máy Snort, hiện cảnh báo .....	21
Hình 4.3.1. Kiểm tra IP máy Snort .....	21
Hình 4.3.2. Từ máy Kali, sử dụng công cụ hping3 để tấn công TCP SYN Flood máy Snort (dùng lệnh: hping3 -c 15000 -d 120 -S -w 64 -p 80 --flood --rand-source 192.168.17.178). .....	22
Hình 4.3.3. Trên máy Snort hiện các thông báo .....	22

## 1. Mục đích

- Luyện tập việc cài đặt và vận hành 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

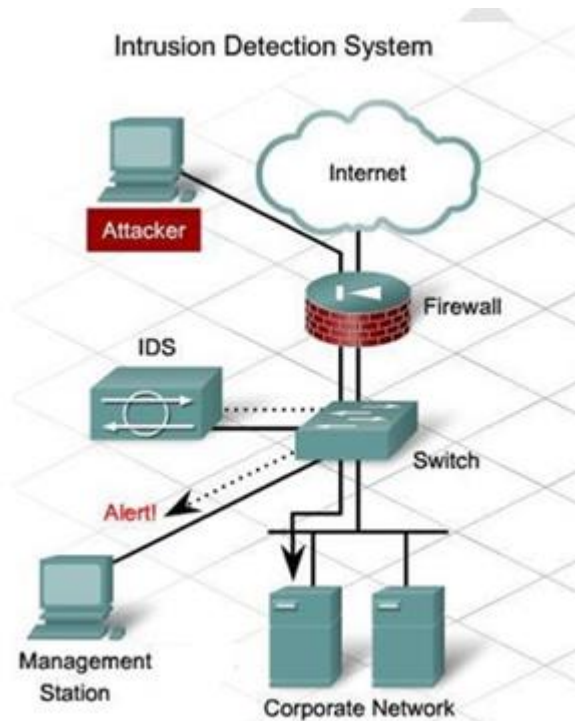
## 2. Nội dung thực hành

### 2.1. Cơ sở lý thuyết

#### 2.1.1. Tìm hiểu khái quát về các hệ thống phát hiện tấn công, xâm nhập, phân loại các hệ thống phát hiện xâm nhập, các kỹ thuật phát hiện xâm nhập.

##### a. Giới thiệu về hệ thống phát hiện tấn công, xâm nhập

Các hệ thống phát hiện tấn công, xâm nhập (IDS – Intrusion Detection System) là một lớp phòng vệ quan trọng trong các lớp giải pháp đảm bảo an toàn cho hệ thống thông tin và mạng theo mô hình phòng thủ có chiều sâu (defence in depth). Các hệ thống IDS có thể được đặt trước hoặc sau tường lửa trong mô hình mạng tùy theo mục đích sử dụng. IDS thường được kết nối vào bộ chuyển mạch (switch) phía sau tường lửa



Hình 2.1.1.a.1. Vị trí hệ thống IDS trong sơ đồ mạng

Nhiệm vụ chính của các hệ thống IDS bao gồm:

- Giám sát lưu lượng mạng hoặc các hành vi trên một hệ thống để nhận dạng các dấu hiệu của tấn công, xâm nhập;
- Khi phát hiện các hành vi tấn công, xâm nhập, thì ghi logs các hành vi này cho phân tích bổ sung sau này;

- Gửi thông báo cho người quản trị về các hành vi tấn công, xâm nhập đã phát hiện được.

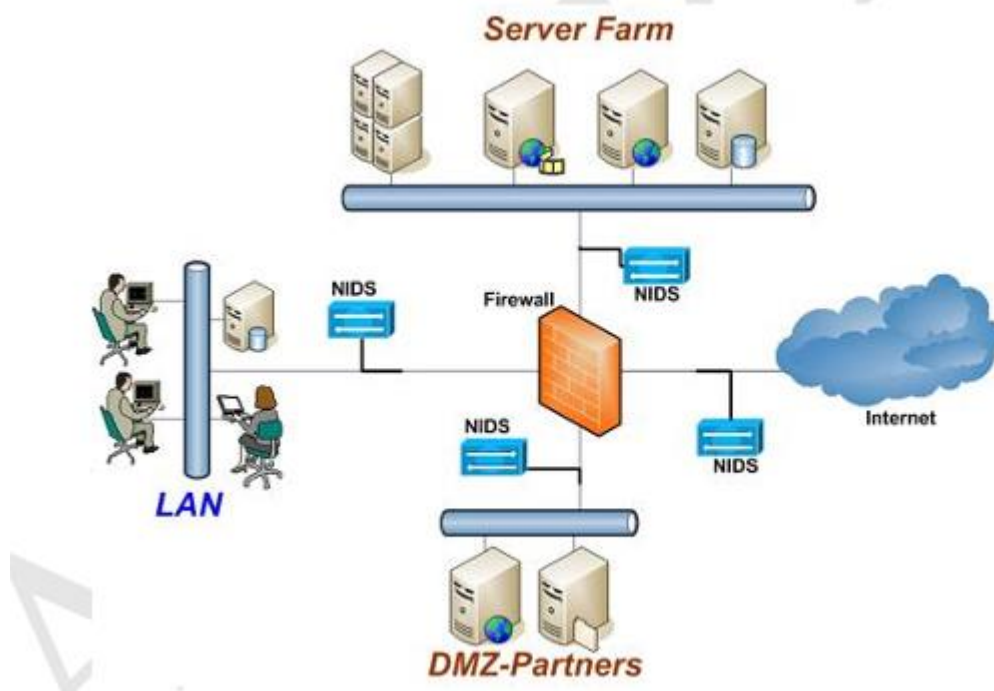
Nói tóm lại, IDS thường được kết nối vào các bộ định tuyến, switch, card mạng và chủ yếu làm nhiệm vụ giám sát và cảnh báo, không có khả năng chủ động ngăn chặn tấn công, xâm nhập.

### b. Phân loại hệ thống phát hiện tấn công, xâm nhập

Có 2 phương pháp phân loại chính các hệ thống IDS gồm (1) phân loại theo nguồn dữ liệu và (2) phân loại theo phương pháp phân tích dữ liệu.

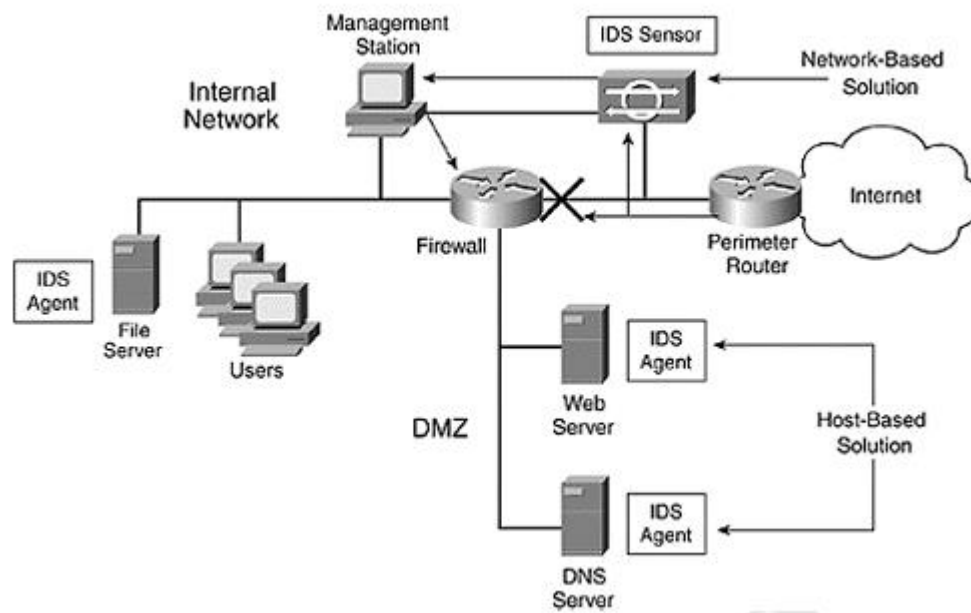
Theo nguồn dữ liệu, có 2 loại hệ thống phát hiện xâm nhập:

- Hệ thống phát hiện xâm nhập mạng (NIDS – Network-based IDS): NIDS phân tích lưu lượng mạng để phát hiện tấn công, xâm nhập cho cả mạng hoặc một phần mạng. Hình dưới đây biểu diễn một sơ đồ mạng, trong đó các NIDS được bố trí để giám sát phát hiện xâm nhập tại cổng vào và cho từng phân đoạn mạng.



Hình 2.1.1.b.1. Các NIDS được bố trí để giám sát phát hiện xâm nhập tại cổng vào và cho từng phân đoạn mạng

- Hệ thống phát hiện xâm nhập cho host (HIDS – Host-based IDS): HIDS phân tích các sự kiện xảy ra trong hệ thống/dịch vụ để phát hiện tấn công, xâm nhập cho hệ thống đó. Hình dưới đây minh họa một sơ đồ mạng, trong đó sử dụng NIDS để giám sát lưu lượng tại cổng mạng và HIDS để giám sát các host thông qua các IDS agent. Một trạm quản lý (Management station) được thiết lập để thu nhập các thông tin từ các NIDS và HIDS để xử lý và đưa ra quyết định cuối cùng.



Hình 2.1.1.b.2. Sử dụng kết hợp NIDS và HIDS để giám sát lưu lượng mạng và các host

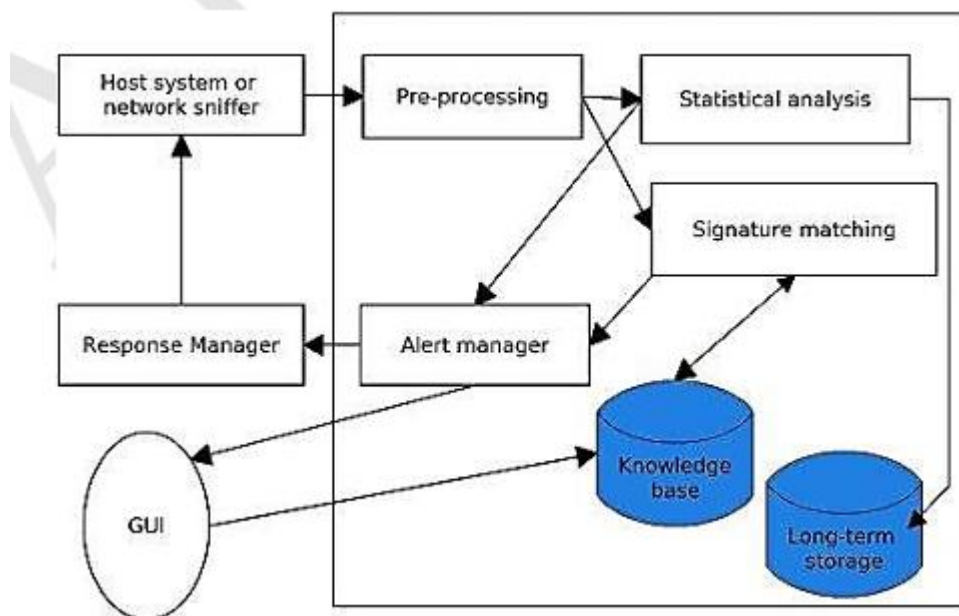
Theo phương pháp phân tích dữ liệu, có 2 kỹ thuật phân tích chính, gồm:

- Phát hiện xâm nhập dựa trên chữ ký, hoặc phát hiện sự lạm dụng (Signature-based / misuse intrusion detection)
- Phát hiện xâm nhập dựa trên các bất thường (Anomaly intrusion detection).

### c. Các kỹ thuật phát hiện xâm nhập

#### c.1. Phát hiện xâm nhập dựa trên chữ ký

Phát hiện xâm nhập dựa trên chữ ký trước hết cần xây dựng cơ sở dữ liệu các chữ ký, hoặc các dấu hiệu của các loại tấn công, xâm nhập đã biết. Hầu hết các chữ ký, dấu hiệu được nhận dạng và mã hóa thủ công và dạng biểu diễn thường gặp là các luật phát hiện (Detection rule). Bước tiếp theo là sử dụng cơ sở dữ liệu các chữ ký để giám sát các hành vi của hệ thống, hoặc mạng, và cảnh báo nếu phát hiện chữ ký của tấn công, xâm nhập. Hình dưới đây biểu diễn lưu đồ giám sát phát hiện tấn công, xâm nhập dựa trên chữ ký điển hình, trong đó Knowledge base là cơ sở dữ liệu lưu các chữ ký tấn công, xâm nhập.

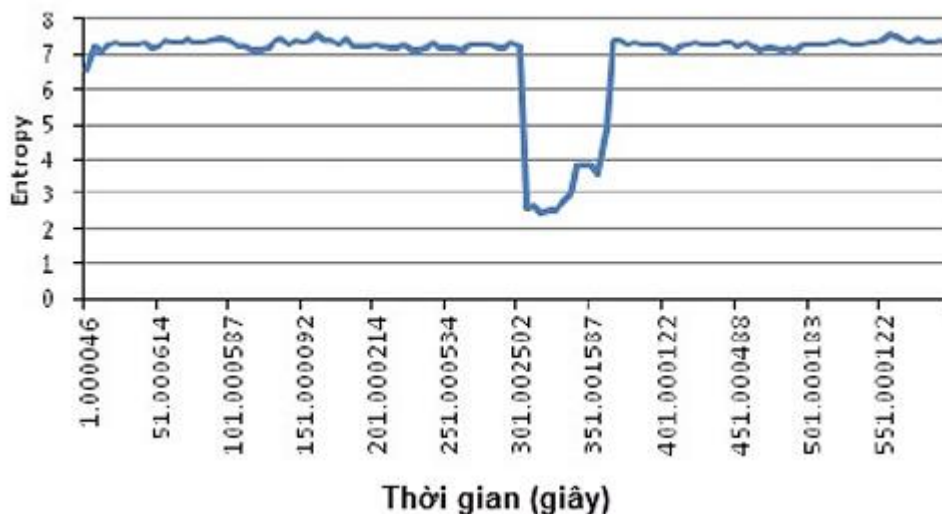


Hình 2.1.1.c.1. Lưu đồ giám sát phát hiện tấn công, xâm nhập dựa trên chữ ký

Ưu điểm lớn nhất của phát hiện xâm nhập dựa trên chữ ký là có khả năng phát hiện các tấn công, xâm nhập đã biết một cách hiệu quả. Ngoài ra, phương pháp này cho tốc độ xử lý cao, đồng thời yêu cầu tài nguyên tính toán tương đối thấp. Nhờ vậy, các hệ thống phát hiện xâm nhập dựa trên chữ ký được ứng dụng rộng rãi trong thực tế. Tuy nhiên, nhược điểm chính của phương pháp này là không có khả năng phát hiện các tấn công, xâm nhập mới, do chữ ký của chúng chưa tồn tại trong cơ sở dữ liệu các chữ ký. Hơn nữa, phương pháp này cũng đòi hỏi nhiều công sức xây dựng và cập nhật cơ sở dữ liệu chữ ký, dấu hiệu của các tấn công, xâm nhập.

### c.2. Phát hiện xâm nhập dựa trên bất thường

Phát hiện xâm nhập dựa trên bất thường dựa trên giả thiết: các hành vi tấn công, xâm nhập thường có quan hệ chặt chẽ với các hành vi bất thường. Quá trình xây dựng và triển khai một hệ thống phát hiện xâm nhập dựa trên bất thường thường gồm 2 giai đoạn: (1) huấn luyện và (2) phát hiện. Trong giai đoạn huấn luyện, hồ sơ (profile) của đối tượng trong chế độ làm việc bình thường được xây dựng. Để thực hiện giai đoạn huấn luyện, cần giám sát đối tượng trong một khoảng thời gian đủ dài để thu thập được đầy đủ dữ liệu mô tả các hành vi của đối tượng trong điều kiện bình thường làm dữ liệu huấn luyện. Tiếp theo, thực hiện huấn luyện dữ liệu để xây dựng mô hình phát hiện, hay hồ sơ của đối tượng. Trong giai đoạn phát hiện, thực hiện giám sát hành vi hiện tại của hệ thống và cảnh báo nếu có khác biệt rõ nét giữa hành vi hiện tại và các hành vi lưu trong hồ sơ của đối tượng.



Hình 2.1.1.c.2. 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)

Hình trên biểu diễn giá trị entropy của IP nguồn của các gói tin theo cửa sổ trượt từ lưu lượng bình thường và entropy của IP nguồn của các gói tin từ lưu lượng tấn công DDoS. Có thể thấy sự khác biệt rõ nét giữa giá trị entropy của lưu lượng bình thường và lưu lượng tấn công và như vậy nếu một ngưỡng entropy được chọn phù hợp ta hoàn toàn có thể phát hiện sự xuất hiện của cuộc tấn công DDoS dựa trên sự thay đổi đột biến của giá trị entropy. Ưu điểm của phát hiện xâm nhập dựa trên bất thường là có tiềm năng phát hiện các loại tấn công, xâm nhập mới mà không yêu cầu biết trước thông tin về chúng. Tuy nhiên, phương pháp này thường có tỷ lệ cảnh báo sai tương đối cao so với phương pháp phát hiện dựa trên chữ ký. Điều này làm giảm khả năng ứng dụng thực tế của phát hiện xâm nhập dựa trên bất thường. Ngoài ra, phương pháp này cũng tiêu tốn nhiều tài nguyên hệ thống cho việc xây dựng hồ sơ đối tượng và phân tích hành vi hiện tại.

## 2.1.2. Tìm hiểu về kiến trúc và tính năng của một số hệ thống phát hiện tấn công, xâm nhập: Snort, OSSEC

### 2.1.2.1. Snort

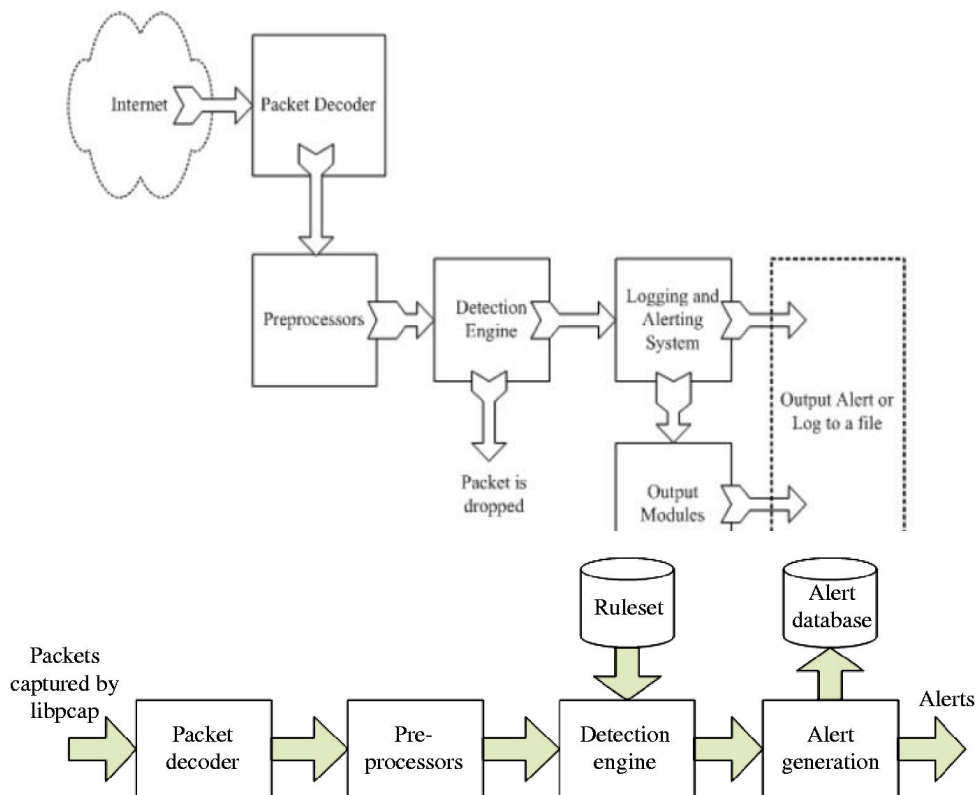
#### a. Giới thiệu

Snort là một công cụ IDS/IPS, thực hiện giám sát các gói tin ra vào hệ thống.

- Snort là một mã nguồn mở miễn phí với nhiều tính năng trong việc bảo vệ hệ thống bên trong, phát hiện sự tấn công từ bên ngoài vào hệ thống.
- Snort được viết bởi Martin Roesch vào năm 1998. Hiện tại, Snort được phát triển bởi Sourcefire, nơi mà Roesch đang là người sáng lập và CTO, và được sở hữu bởi Cisco từ năm 2013.



## b. Kiến trúc của Snort



Trong mô hình kiến trúc trên, hệ thống Snort được chia thành 4 phần:

- Module Decoder: Xử lý giải mã các gói tin
- Module Preprocessors: Tiền xử lý
- Module Detection Engine: Phát hiện
- Module Logging and Alerting System: Lưu log và cảnh báo

## c. Các luật của Snort

**Cấu trúc của một rule** được chia thành 02 phần: **|Rule header|Rule 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 tiêu chuẩn để áp dụng luật với gói tin đó.
- Phần Option: Chứa thông điệp cảnh báo và các thông tin về các phần của gói tin dùng để tạo nên cảnh báo. Phần Option này chứa các tiêu chuẩn phụ thêm để đối sánh với gói tin

**Cấu trúc phần Header:** |Action|Protocol|Address|port|Direction|Address|Port|

- Action: Thể hiện hành động sẽ được thực hiện khi một gói tin kích hoạt quy tắc. Trong đó:
  - + alert: Tạo một cảnh báo và ghi lại gói tin.
  - + log: Chỉ ghi lại gói tin mà không tạo cảnh báo.
  - + pass: Bỏ qua gói tin, không thực hiện hành động nào.

## Bài 6: Cài đặt cấu hình HIDS/NIDS

- + **activate**: Tạo ra cảnh báo và kích hoạt thêm các luật khác để kiểm tra thêm điều kiện của gói tin
- + **dynamic**: Đây là luật được gọi bởi các luật khác có Action khai báo là Activate
- **Protocol**: Xác định loại giao thức của gói tin, ví dụ: TCP, UDP, ICMP, hoặc any (tất cả).
- **Source IP Address**: Địa chỉ IP nguồn của gói tin.
- **Source Port**: Cổng nguồn của gói tin. Có thể là một số cụ thể hoặc từ khoảng cụ thể.
- **Direction Operator**: Thể hiện hướng của gói tin. Có thể là **->** (nguồn tới đích) hoặc **<-** (đích tới nguồn).
- **Destination IP Address**: Địa chỉ IP đích của gói tin.
- **Destination Port**: Cổng đích của gói tin. Cũng có thể là một số cụ thể hoặc từ khoảng cụ thể.

### Cấu trúc phần Option:

Phần Option nằm ngay sau phần Header và được bao bọc trong dấu ngoặc đơn. Nếu có nhiều Option thì sẽ phân biệt bởi dấu chấm phẩy ";". Một Option gồm có 2 phần: một là từ khóa và một là tham số. 02 phần này sẽ phân cách nhau bằng dấu hai chấm ":".

Các option có thể là: **msg** (tin nhắn cảnh báo), **content** (nội dung gói tin), **sid** (số nhận dạng duy nhất), **rev** (số phiên bản quy tắc), **content**: Chứa một chuỗi hoặc byte pattern để so khớp với dữ liệu gói tin ...

Ví dụ về cấu trúc một quy tắc Snort:

```
alert tcp any any -> any 80 (msg:"Potential Web Attack"; content:"/bin/bash";  
sid:100001;)
```

### 2.1.2.2. Ossec

#### a. Ossec là gì?

OSSEC là phần mềm mã nguồn mở giúp phát hiện xâm nhập dựa trên host (HIDS)

Nó đa nền tảng, có thể mở rộng và có nhiều cơ chế bảo mật khác nhau.

b. Các tính năng của Ossec

- Log based Intrusion Detection (LIDs) and Log Monitoring:
  - + Chủ động theo dõi và phân tích dữ liệu real-time từ nhiều nguồn sinh log.
  - + Ngoài ra, Ossec sẽ thu thập, phân tích và kiểm tra mối tương quan các log và cho ta biết những điều đáng ngờ đang xảy ra trong hệ thống (bị tấn công, lỗi, sử dụng sai,...), các phần mềm được cài đặt thêm, các rule firewall bị đổi.
- Compliance Auditing:

## Bài 6: Cài đặt cấu hình HIDS/NIDS

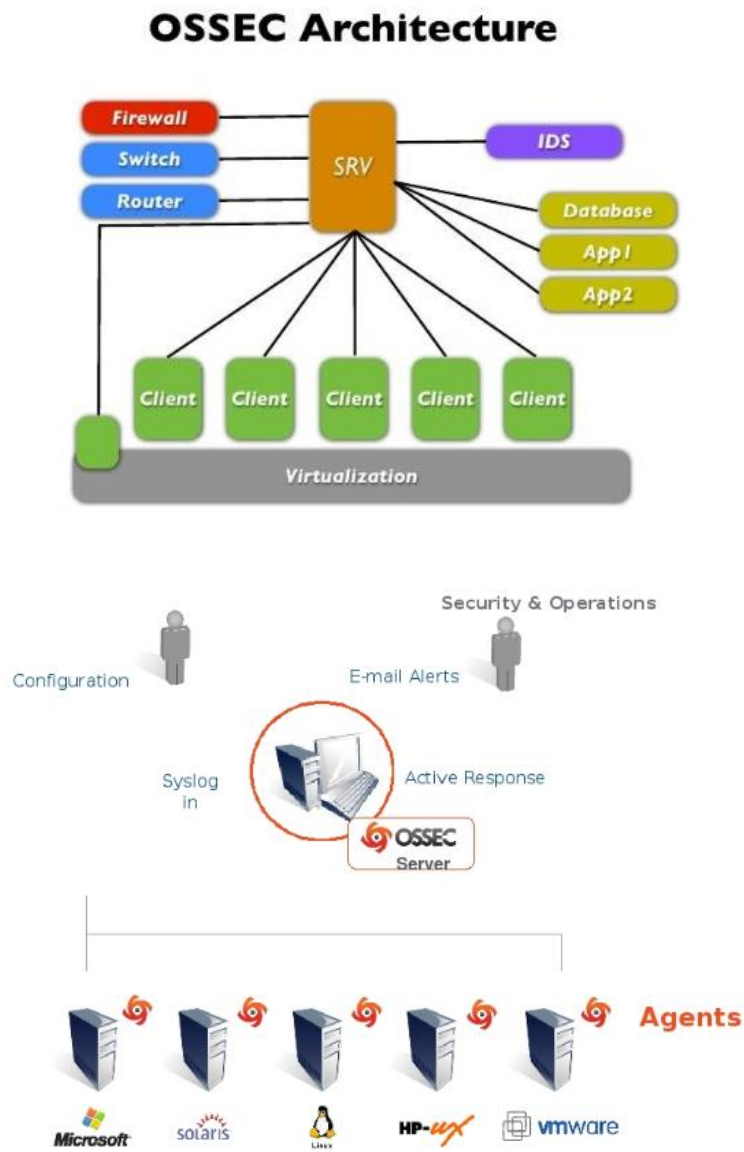
- + Kiểm soát các ứng dụng và hệ thống nhằm tuân thủ các yêu cầu, tiêu chuẩn về bảo mật như PCI-DSS và CIS.
- Rootkit and Malware Detection:
  - + Tin tặc thường muốn che dấu hành động và quay lại hệ thống đã xâm nhập được
  - + Ossec phân tích ở cấp độ file và tiến trình nhằm phát hiện các ứng dụng độc hại, các rootkit hay các file hệ thống bị sửa đổi theo cách phổ biến với rootkit
- File Integrity Monitoring (FIM):
  - + Phát hiện các thay đổi đối với hệ thống.
- Active Response:
  - + Các hành vi ứng phó lại các cuộc tấn công vào hệ thống trong thời gian thực.
  - + Giúp ngăn sự cố lan rộng trước khi admin có thể hành động
- System Inventory:
  - + Thu thập các thông tin hệ thống như phần mềm được cài đặt, hardware,...

### c. Điểm nổi trội của Ossec

- Đa nền tảng (Linux, Mac OS, Window, Solaris)
- Real-time Alert (Cảnh báo thời gian thực)
  - + Kết hợp với smtp, sms, syslog sẽ cho phép người dùng nhận cảnh báo trên các thiết bị có hỗ trợ email
  - + Ngoài ra tính năng Active-response có thể giúp block 1 cuộc tấn công ngay lập tức.
- Có thể tích hợp với các hệ thống hiện đại (SIM/SEM)
- Mô hình Server – Agent/Agentless, cho phép Server dễ dàng quản lý tập trung các chính sách trên nhiều OS.
- Giám sát trên agent, agentless (Client không cài đặt được gói agent) như router, firewall

### d. Kiến trúc và mô hình hoạt động của Ossec

Ossec hoạt động theo mô hình Server-Agent/Agentless



Hình 2.1.2.2.1. Kiến trúc của OSSEC

### ❖ Manager (Server)

Lưu trữ cơ sở dữ liệu của việc kiểm tra tính toàn vẹn file

Kiểm tra các log, event.

Quản lý, lưu tất cả các rule, decoder (bộ giải mã), cấu hình chính. Điều này giúp dễ dàng quản lý, dù cho có lượng lớn Agent

Server không chạy trên Windows OS.

### ❖ Agent

Bản chất thì là 1 phần mềm được cài đặt trên máy client giúp thu thập các thông tin và gửi cho Server để phân tích, thống kê.

- Chiếm lượng memory và CPU nhỏ, không đáng kể
- 1 số thông tin được thu thập theo thời gian thực
- 1 số thông tin thì lại được thu thập định kỳ
- Nhưng khi nói Agent thì là để chỉ máy Client được cài gói Ossec-agent.

*Chú ý:* Windows OS chỉ có thể làm Agent chứ không làm Server được.

#### ❖ **Agentless**

Là các hệ thống không cài được gói agent

Trên các Agentless này có thể thực hiện việc kiểm tra tính toàn vẹn

Giúp monitor firewall, router hay thậm chí cả hệ thống Unix

#### ❖ **Ảo hóa/ VMware**

Cho phép cài đặt agent trên các guest OS (Máy ảo)

Ngoài ra cũng được cài đặt trong VMware ESX nhưng có thể dẫn đến sự cố không hỗ trợ.

Khi cài đặt trong VMware ESX giúp nhận được thời điểm các VM guest được khởi tạo, xóa đi, khởi động,... Ossec cũng giám sát việc login, logouts và các lỗi bên trong ESX server

Ngoài ra nó cũng cảnh báo nếu bất kỳ tùy chọn cấu hình không an toàn nào được bật.

#### ❖ **Firewalls, switches and routers**

Chính là các Agentless

Ossec có thể nhận và phân tích nhật ký hệ thống từ nhiều firewall, switch, router.

Nó support tất cả Cisco routers, Cisco PIX, Cisco FWSM, Cisco ASA, Juniper Routers, Netscreen firewall, Checkpoint và nhiều thiết bị khác.

## 2.2. Nội dung thực hành

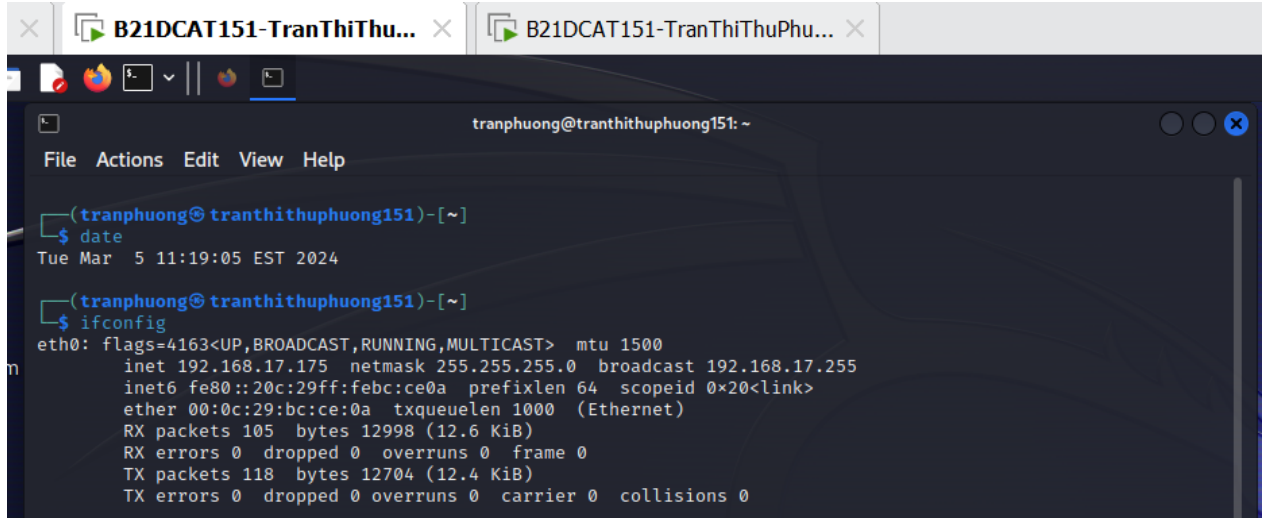
### 2.2.1. Chuẩn bị môi trường

- 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): IP 192.168.17.176
- 01 máy tính (máy thật hoặc máy ảo) chạy Kali Linux (bản 2021 trở lên): IP 192.168.17.175

- Bộ phần mềm Snort

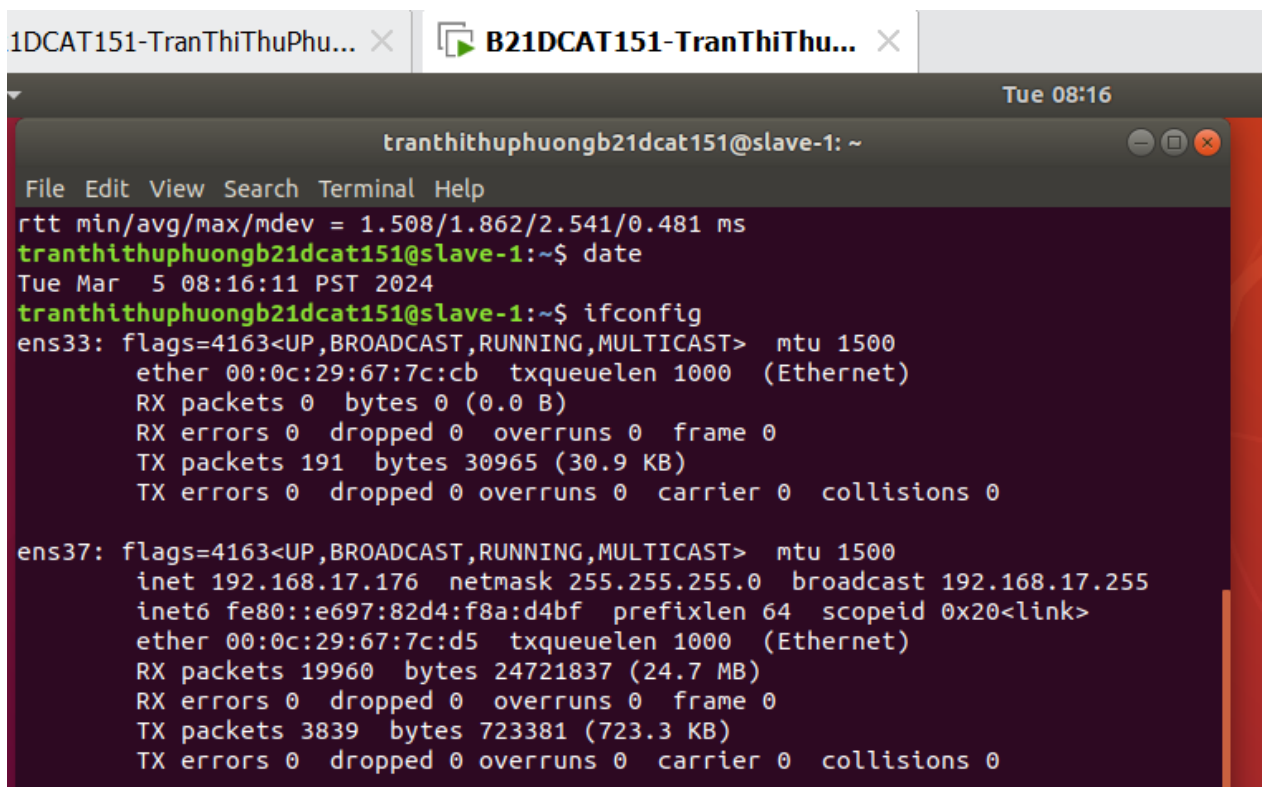
### 2.2.2. Các bước thực hiện

- a. **Bước 1:** Chuẩn bị các máy tính như mô tả trong mục 2.2.1 Máy Kali Linux được đổi tên thành B21DCAT151-TranThiThuPhuong-Kali và máy cài Snort thành B21DCAT151-TranThiThuPhuong-Snort. Các máy có địa chỉ IP và kết nối mạng LAN.



```
(tranphuong@tranthithuphuong151)-[~]
$ date
Tue Mar 5 11:19:05 EST 2024

(tranphuong@tranthithuphuong151)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.17.175 netmask 255.255.255.0 broadcast 192.168.17.255
    inet6 fe80::20c:29ff:febc:ce0a prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:bc:ce:0a txqueuelen 1000 (Ethernet)
    RX packets 105 bytes 12998 (12.6 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 118 bytes 12704 (12.4 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```



```
tranthithuphuongb21dcat151@slave-1: ~
File Edit View Search Terminal Help
rtt min/avg/max/mdev = 1.508/1.862/2.541/0.481 ms
tranthithuphuongb21dcat151@slave-1:~$ date
Tue Mar 5 08:16:11 PST 2024
tranthithuphuongb21dcat151@slave-1:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    ether 00:0c:29:67:7c:cb txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 191 bytes 30965 (30.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ens37: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.17.176 netmask 255.255.255.0 broadcast 192.168.17.255
    inet6 fe80::e697:82d4:f8a:d4bf prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:67:7c:d5 txqueuelen 1000 (Ethernet)
    RX packets 19960 bytes 24721837 (24.7 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 3839 bytes 723381 (723.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

- b. **Bước 2:** Tải, cài đặt Snort và chạy thử Snort. Kiểm tra log của Snort để đảm bảo Snort hoạt động bình thường.

- Tải snort:
  - + `sudo apt update`

## Bài 6: Cài đặt cấu hình HIDS/NIDS

- + `sudo apt install snort`
- Kiểm tra phiên bản của snort

```
Processing triggers for systemd (237-3ubuntu10.57) ...
tranthithuphuongb21dcat151@slave-1:~$ snort --version

o''~
o''~)~
o''~)~
o''~)~

-*> Snort! <*-
Version 2.9.7.0 GRE (Build 149)
By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
Copyright (C) 2014 Cisco and/or its affiliates. All rights reserved.
Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using libpcap version 1.8.1
Using PCRE version: 8.39 2016-06-14
Using ZLIB version: 1.2.11

tranthithuphuongb21dcat151@slave-1:~$ date
Tue Mar  5 08:29:49 PST 2024
tranthithuphuongb21dcat151@slave-1:~$ clear
```

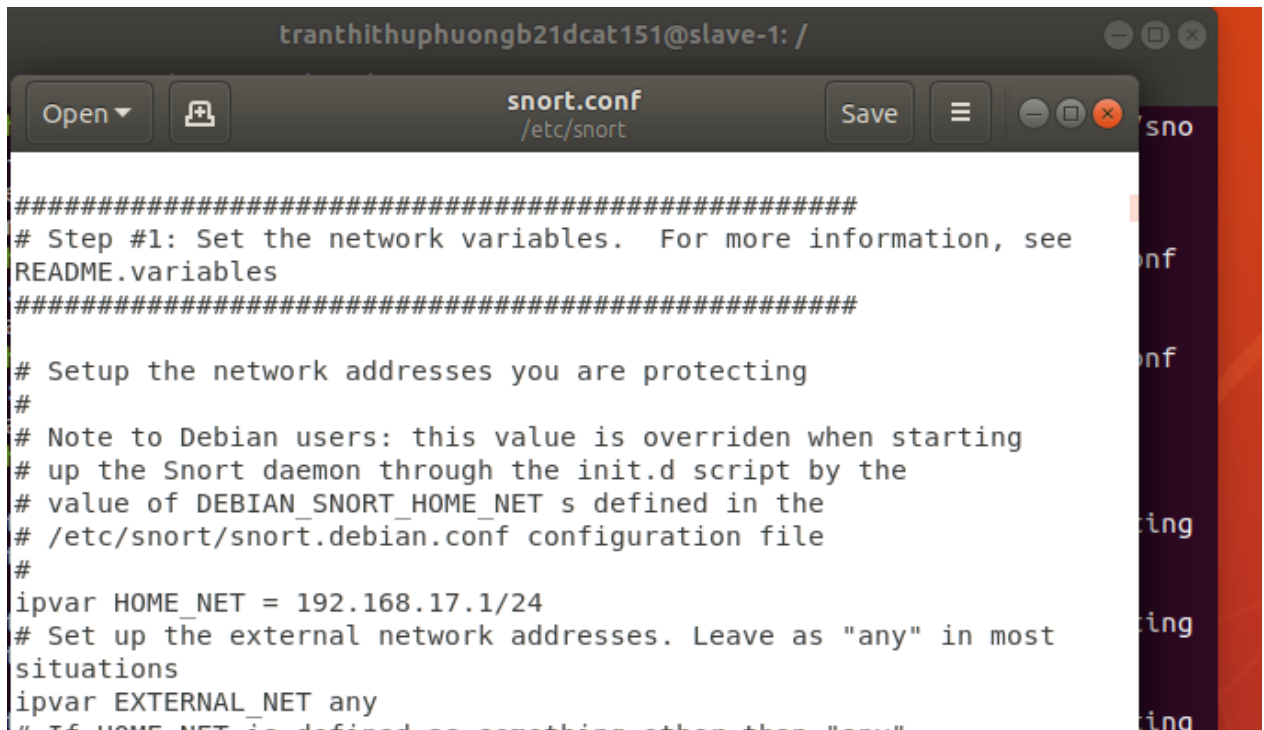
- Kiểm tra trạng thái của snort

```
tranthithuphuongb21dcat151@slave-1: ~
File Edit View Search Terminal Help
tranthithuphuongb21dcat151@slave-1:~$ date
Tue Mar  5 08:34:40 PST 2024
tranthithuphuongb21dcat151@slave-1:~$ systemctl status snort
● snort.service - LSB: Lightweight network intrusion detection system
   Loaded: loaded (/etc/init.d/snort; generated)
   Active: active (running) since Tue 2024-03-05 08:29:22 PST; 5min ago
     Docs: man:systemd-sysv-generator(8)
    Tasks: 2 (limit: 2281)
   CGroup: /system.slice/snort.service
           └─3738 /usr/sbin/snort -m 027 -D -d -l /var/log/snort -u snort -g sno

Mar 05 08:29:23 slave-1 snort[3738]: Preprocessor Object: SF_IMAP Ve
Mar 05 08:29:23 slave-1 snort[3738]: Preprocessor Object: SF_DNP3 Ve
Mar 05 08:29:23 slave-1 snort[3738]: Preprocessor Object: SF_SIP Ver
Mar 05 08:29:23 slave-1 snort[3738]: Preprocessor Object: SF_MODBUS
Mar 05 08:29:23 slave-1 snort[3738]: Preprocessor Object: SF_REPUTATI
Mar 05 08:29:23 slave-1 snort[3738]: Preprocessor Object: SF_SDF Ver
Mar 05 08:29:23 slave-1 snort[3738]: Preprocessor Object: SF_DNS Ver
Mar 05 08:29:23 slave-1 snort[3738]: Preprocessor Object: SF_FTPTELNE
Mar 05 08:29:23 slave-1 snort[3738]: Preprocessor Object: SF_DCERPC2
Mar 05 08:29:23 slave-1 snort[3738]: Commencing packet processing (pid=3738)
```

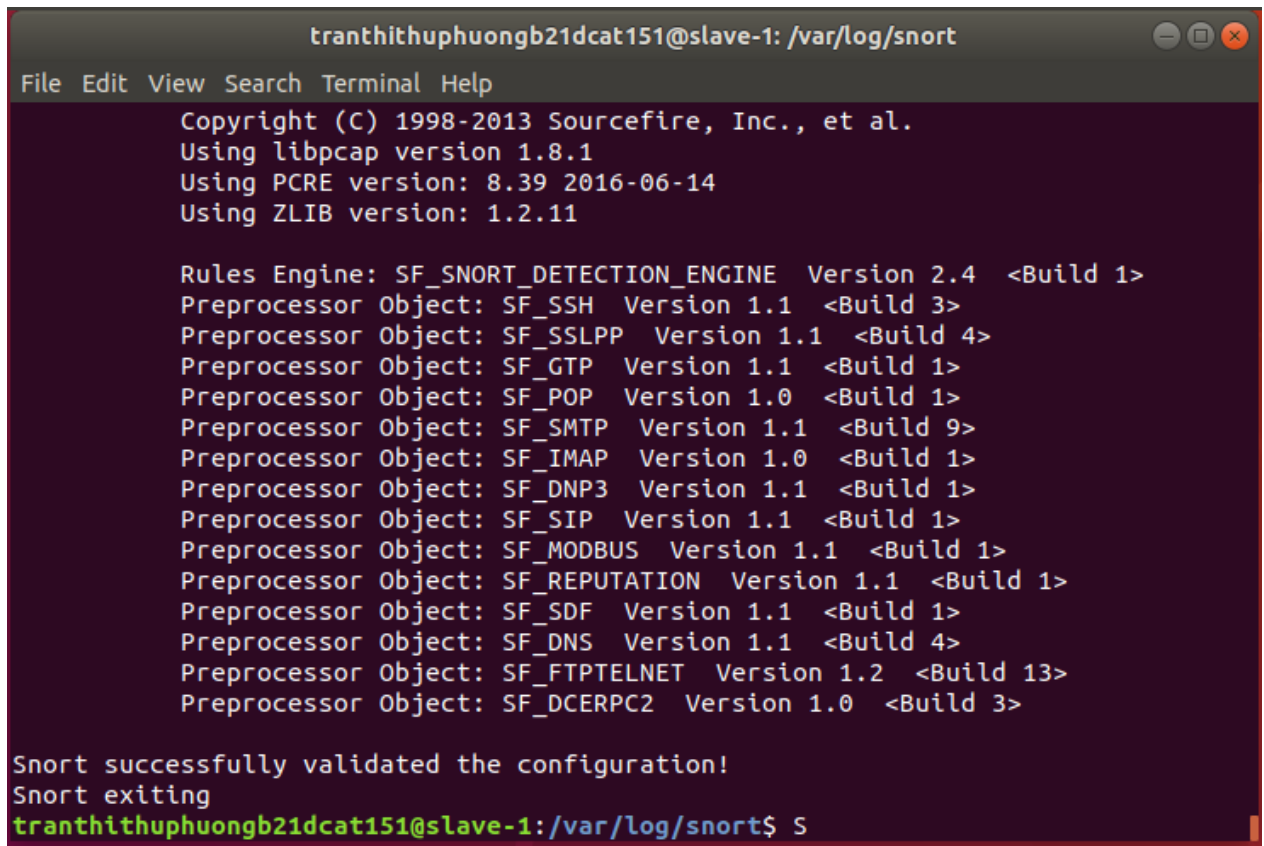
- Khởi chạy snort và giám sát trên card mạng ens37



A screenshot of a text editor window titled 'snort.conf' with the path '/etc/snort'. The window shows the configuration file content. The user's terminal prompt is 'tranthithuphuongb21dcat151@slave-1: /'.

```
#####  
# Step #1: Set the network variables. For more information, see  
# README.variables  
#####  
  
# Setup the network addresses you are protecting  
#  
# Note to Debian users: this value is overridden when starting  
# up the Snort daemon through the init.d script by the  
# value of DEBIAN_SNORT_HOME_NET s defined in the  
# /etc/snort/snort.debian.conf configuration file  
#  
ipvar HOME_NET = 192.168.17.1/24  
# Set up the external network addresses. Leave as "any" in most  
# situations  
ipvar EXTERNAL_NET any  
# If HOME_NET is defined as something other than "any"
```

- Kiểm tra cấu hình snort: `sudo -T -i ens37 -c /etc/snort/snort.conf`

A screenshot of a terminal window titled 'tranthithuphuongb21dcat151@slave-1: /var/log/snort'. The terminal shows the output of the command 'sudo -T -i ens37 -c /etc/snort/snort.conf'. The output displays the Snort version and various preprocessor objects and their versions. The user's terminal prompt is 'tranthithuphuongb21dcat151@slave-1: /var/log/snort\$'.

```
Copyright (C) 1998-2013 Sourcefire, Inc., et al.  
Using libpcap version 1.8.1  
Using PCRE version: 8.39 2016-06-14  
Using ZLIB version: 1.2.11  
  
Rules Engine: SF_SNORT_DETECTION_ENGINE Version 2.4 <Build 1>  
Preprocessor Object: SF_SSH Version 1.1 <Build 3>  
Preprocessor Object: SF_SSLPP Version 1.1 <Build 4>  
Preprocessor Object: SF_GTP Version 1.1 <Build 1>  
Preprocessor Object: SF_POP Version 1.0 <Build 1>  
Preprocessor Object: SF_SMTP Version 1.1 <Build 9>  
Preprocessor Object: SF_IMAP Version 1.0 <Build 1>  
Preprocessor Object: SF_DNP3 Version 1.1 <Build 1>  
Preprocessor Object: SF_SIP Version 1.1 <Build 1>  
Preprocessor Object: SF_MODBUS Version 1.1 <Build 1>  
Preprocessor Object: SF_REPUTATION Version 1.1 <Build 1>  
Preprocessor Object: SF_SDF Version 1.1 <Build 1>  
Preprocessor Object: SF_DNS Version 1.1 <Build 4>  
Preprocessor Object: SF_FTPTELNET Version 1.2 <Build 13>  
Preprocessor Object: SF_DCERPC2 Version 1.0 <Build 3>  
  
Snort successfully validated the configuration!  
Snort exiting  
tranthithuphuongb21dcat151@slave-1: /var/log/snort$
```

- Khởi chạy giám sát (hiển thị thông báo):  
`sudo snort -A console -q -c /etc/snort/snort.conf -i ens37`

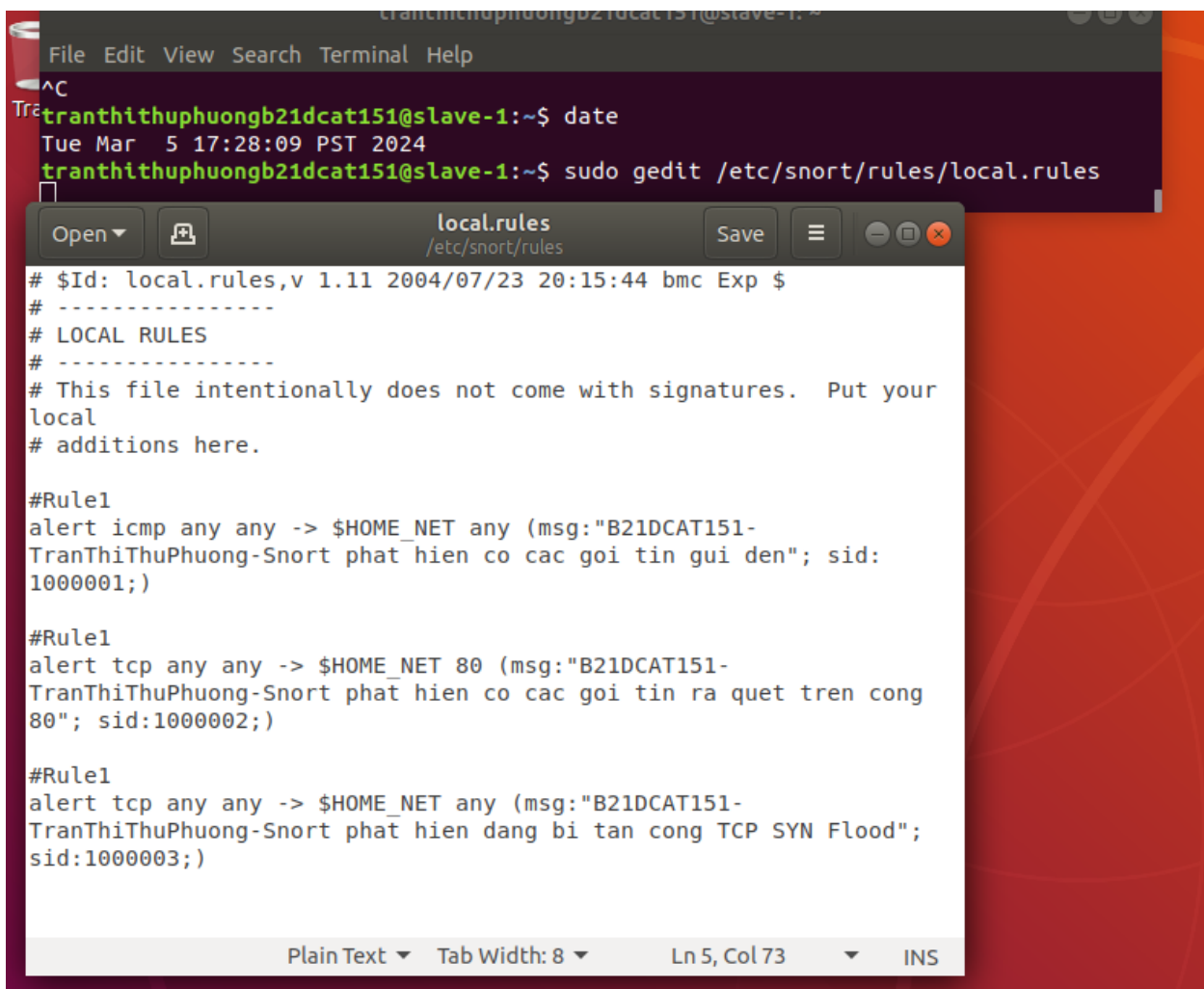


```
tranthithuphuongb21dcat151@slave-1:/var/log/snort$ sudo snort -A console -q -c /etc/snort
/snort.conf -i ens37
03/05-09:20:55.208331  [**] [1:1917:6] SCAN UPnP service discover attempt [**] [Classification: Detection of a Network Scan] [Priority: 3] {UDP} 192.168.17.1:54624 -> 239.255.255.250:1900
03/05-09:20:56.216099  [**] [1:1917:6] SCAN UPnP service discover attempt [**] [Classification: Detection of a Network Scan] [Priority: 3] {UDP} 192.168.17.1:54624 -> 239.255.255.250:1900
03/05-09:20:57.218616  [**] [1:1917:6] SCAN UPnP service discover attempt [**] [Classification: Detection of a Network Scan] [Priority: 3] {UDP} 192.168.17.1:54624 -> 239.255.255.250:1900
03/05-09:20:58.233141  [**] [1:1917:6] SCAN UPnP service discover attempt [**] [Classification: Detection of a Network Scan] [Priority: 3] {UDP} 192.168.17.1:54624 -> 239.255.255.250:1900
```

**c. Bước 3:** Tạo các luật Snort để phát hiện 3 dạng rà quét, tấn công hệ thống

- Yêu cầu:
  - + Phát hiện các gói tin ping từ bất kỳ một máy nào gửi đến máy chạy Snort. Hiện thị thông điệp khi phát hiện: “B21DCAT151-TranThiThuPhuong-Snort phát hiện có các gói Ping gửi đến.”
  - + 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: “ B21DCAT151-TranThiThuPhuong-Snort phát hiện có các gói tin rà quét trên cổng 80.”
  - + 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: “B21DCAT151-TranThiThuPhuong-Snort phát hiện đang bị tấn công TCP SYN Flood.”
- Mở file để tạo thêm luật: *sudo gedit /etc/snort/rules/local.rules*

## Bài 6: Cài đặt cấu hình HIDS/NIDS



The screenshot shows a Kali Linux desktop environment. In the background, a terminal window displays the following commands and output:

```
tranthithuphuongb21dcat151@slave-1:~$ date
Tue Mar  5 17:28:09 PST 2024
tranthithuphuongb21dcat151@slave-1:~$ sudo gedit /etc/snort/rules/local.rules
```

In the foreground, a text editor window titled "local.rules" is open, showing the following content:

```
# $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.

#Rule1
alert icmp any any -> $HOME_NET any (msg:"B21DCAT151-
TranThiThuPhuong-Snort phat hien co cac goi tin gui den"; sid:
1000001;)

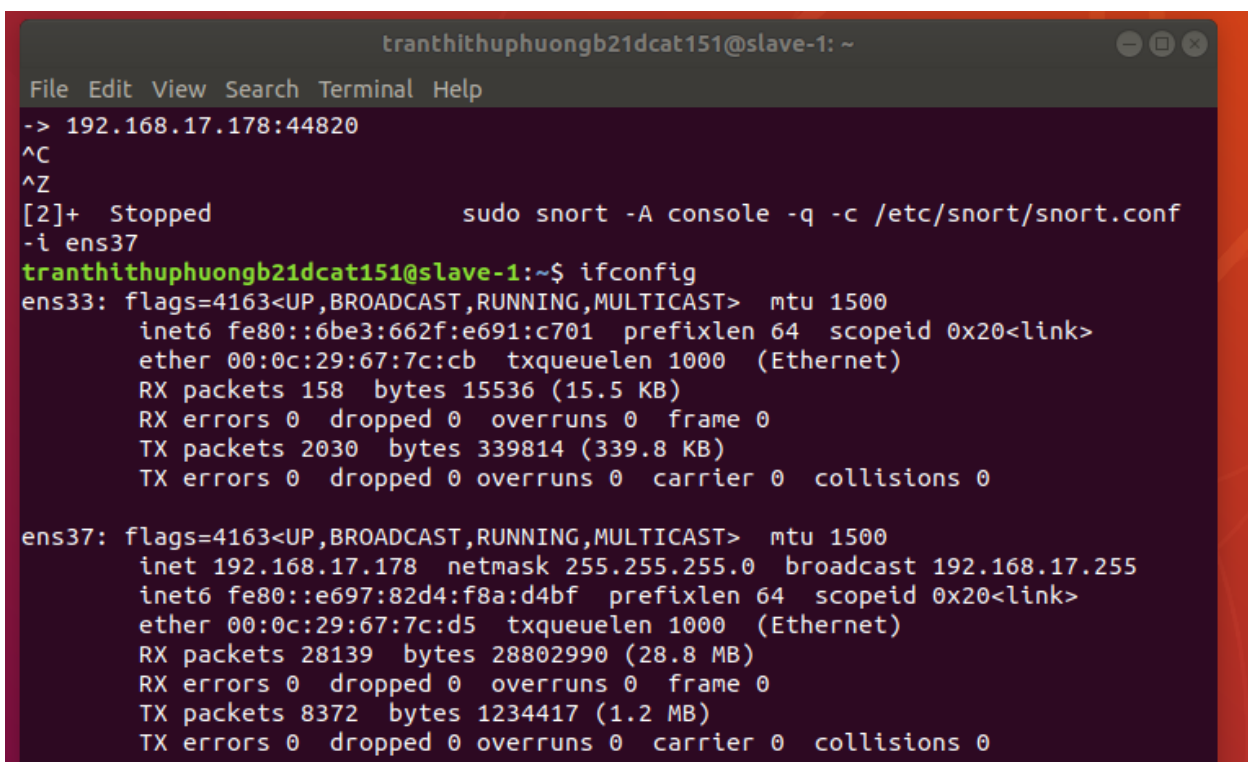
#Rule1
alert tcp any any -> $HOME_NET 80 (msg:"B21DCAT151-
TranThiThuPhuong-Snort phat hien co cac goi tin ra quet tren cong
80"; sid:1000002;)

#Rule1
alert tcp any any -> $HOME_NET any (msg:"B21DCAT151-
TranThiThuPhuong-Snort phat hien dang bi tan cong TCP SYN Flood";
sid:1000003;)
```

The text editor's status bar at the bottom indicates "Plain Text", "Tab Width: 8", "Ln 5, Col 73", and "INS".

### d. Bước 4: Thực thi tấn công và phát hiện sử dụng Snort.

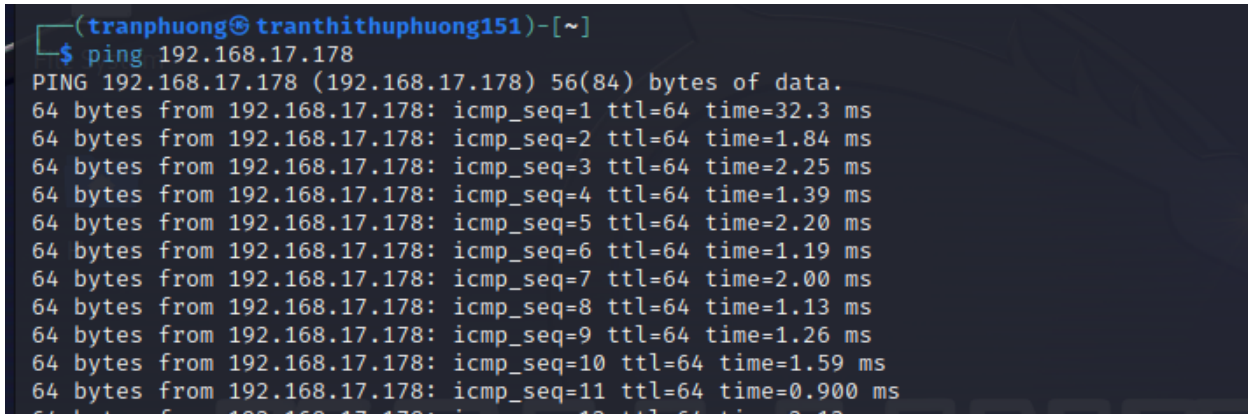
- Từ máy Kali, sử dụng lệnh ping để ping máy Snort. Trên máy Snort kiểm tra kết quả phát hiện trên giao diện terminal hoặc log của Snort.



```
tranthithuphuongb21dcat151@slave-1: ~
File Edit View Search Terminal Help
-> 192.168.17.178:44820
^C
^Z
[2]+  Stopped                  sudo snort -A console -q -c /etc/snort/snort.conf
-i ens37
tranthithuphuongb21dcat151@slave-1:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 fe80::6be3:662f:e691:c701  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:67:7c:cb  txqueuelen 1000  (Ethernet)
        RX packets 158  bytes 15536 (15.5 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2030  bytes 339814 (339.8 KB)
        TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

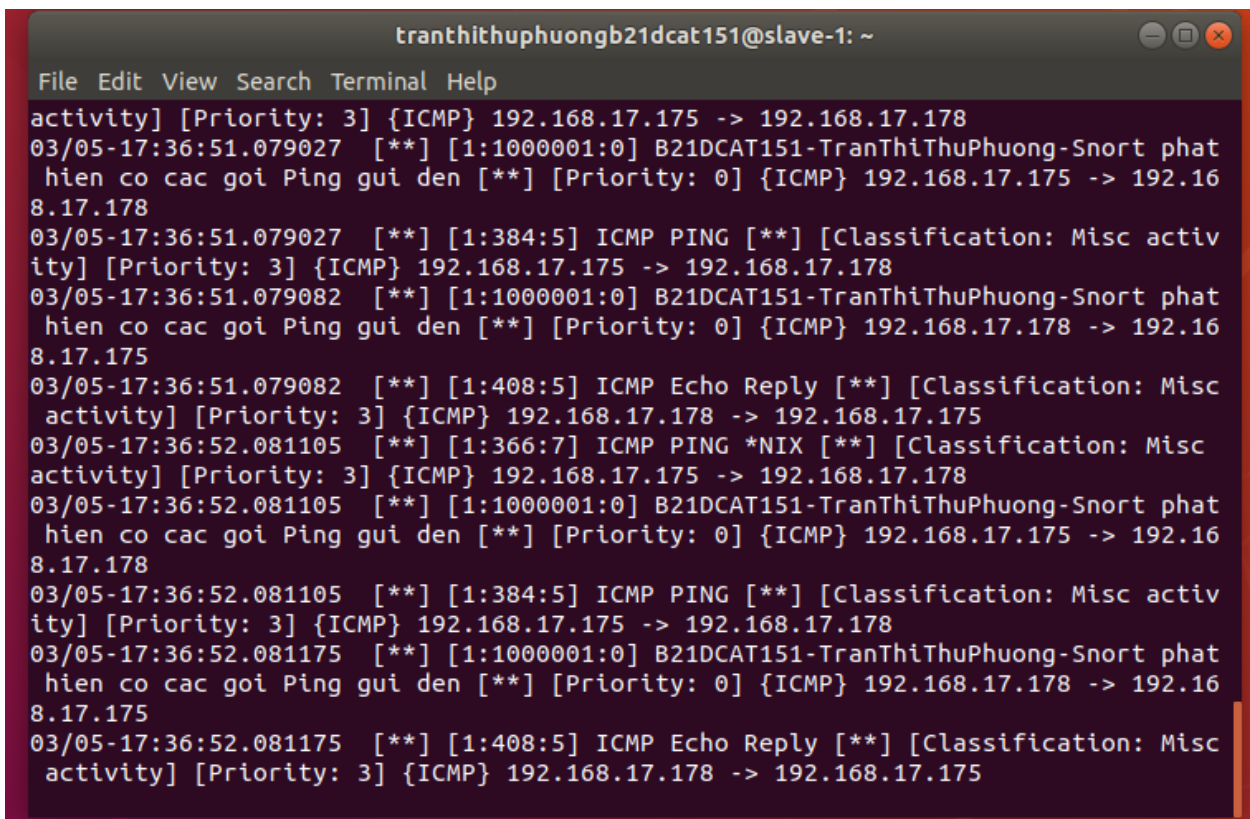
ens37: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.17.178  netmask 255.255.255.0  broadcast 192.168.17.255
        inet6 fe80::e697:82d4:f8a:d4bf  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:67:7c:d5  txqueuelen 1000  (Ethernet)
        RX packets 28139  bytes 28802990 (28.8 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8372  bytes 1234417 (1.2 MB)
        TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
```

Hình 4.1. Kiểm tra địa chỉ IP máy Snort



```
(tranhithuphuong@tranthithuphuong151)-[~]
$ ping 192.168.17.178
PING 192.168.17.178 (192.168.17.178) 56(84) bytes of data.
 64 bytes from 192.168.17.178: icmp_seq=1 ttl=64 time=32.3 ms
 64 bytes from 192.168.17.178: icmp_seq=2 ttl=64 time=1.84 ms
 64 bytes from 192.168.17.178: icmp_seq=3 ttl=64 time=2.25 ms
 64 bytes from 192.168.17.178: icmp_seq=4 ttl=64 time=1.39 ms
 64 bytes from 192.168.17.178: icmp_seq=5 ttl=64 time=2.20 ms
 64 bytes from 192.168.17.178: icmp_seq=6 ttl=64 time=1.19 ms
 64 bytes from 192.168.17.178: icmp_seq=7 ttl=64 time=2.00 ms
 64 bytes from 192.168.17.178: icmp_seq=8 ttl=64 time=1.13 ms
 64 bytes from 192.168.17.178: icmp_seq=9 ttl=64 time=1.26 ms
 64 bytes from 192.168.17.178: icmp_seq=10 ttl=64 time=1.59 ms
 64 bytes from 192.168.17.178: icmp_seq=11 ttl=64 time=0.900 ms
```

Hình 4.1.1. Trên máy Kali, thực hiện Ping đến máy cài Snort



```
tranthithuphuongb21dcat151@slave-1: ~
File Edit View Search Terminal Help
activity] [Priority: 3] {ICMP} 192.168.17.175 -> 192.168.17.178
03/05-17:36:51.079027  [**] [1:1000001:0] B21DCAT151-TranThiThuPhuong-Snort phat
  hien co cac goi Ping gui den [**] [Priority: 0] {ICMP} 192.168.17.175 -> 192.16
8.17.178
03/05-17:36:51.079027  [**] [1:384:5] ICMP PING [**] [Classification: Misc activ
ity] [Priority: 3] {ICMP} 192.168.17.175 -> 192.168.17.178
03/05-17:36:51.079082  [**] [1:1000001:0] B21DCAT151-TranThiThuPhuong-Snort phat
  hien co cac goi Ping gui den [**] [Priority: 0] {ICMP} 192.168.17.178 -> 192.16
8.17.175
03/05-17:36:51.079082  [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc
activity] [Priority: 3] {ICMP} 192.168.17.178 -> 192.168.17.175
03/05-17:36:52.081105  [**] [1:366:7] ICMP PING *NIX [**] [Classification: Misc
activity] [Priority: 3] {ICMP} 192.168.17.175 -> 192.168.17.178
03/05-17:36:52.081105  [**] [1:1000001:0] B21DCAT151-TranThiThuPhuong-Snort phat
  hien co cac goi Ping gui den [**] [Priority: 0] {ICMP} 192.168.17.175 -> 192.16
8.17.178
03/05-17:36:52.081105  [**] [1:384:5] ICMP PING [**] [Classification: Misc activ
ity] [Priority: 3] {ICMP} 192.168.17.175 -> 192.168.17.178
03/05-17:36:52.081175  [**] [1:1000001:0] B21DCAT151-TranThiThuPhuong-Snort phat
  hien co cac goi Ping gui den [**] [Priority: 0] {ICMP} 192.168.17.178 -> 192.16
8.17.175
03/05-17:36:52.081175  [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc
activity] [Priority: 3] {ICMP} 192.168.17.178 -> 192.168.17.175
```

Hình 4.1.2. Trên máy Snort hiện các cảnh báo

- Từ máy Kali, sử dụng công cụ nmap để rà quét máy Snort (dùng lệnh: nmap -sV -p80 -A 192.168.17.178). Trên máy Snort kiểm tra kết quả phát hiện trên giao diện terminal hoặc log của Snort.

```
tranthithuphuongb21dcat151@slave-1: ~
File Edit View Search Terminal Help
-> 192.168.17.178:44820
^C
^Z
[2]+ Stopped                  sudo snort -A console -q -c /etc/snort/snort.conf
-i ens37
tranthithuphuongb21dcat151@slave-1:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet6 fe80::6be3:662f:e691:c701 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:67:7c:cb txqueuelen 1000 (Ethernet)
    RX packets 158 bytes 15536 (15.5 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2030 bytes 339814 (339.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ens37: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.17.178 netmask 255.255.255.0 broadcast 192.168.17.255
    inet6 fe80::e697:82d4:f8a:d4bf prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:67:7c:d5 txqueuelen 1000 (Ethernet)
    RX packets 28139 bytes 28802990 (28.8 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8372 bytes 1234417 (1.2 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Hình 4.1.3. Kiểm tra địa chỉ IP máy Snort

```
(tranhithuphuong151@tranhithuphuong151)~$ nmap -sV -p80 -A 192.168.17.178
Starting Nmap 7.94 ( https://nmap.org ) at 2024-03-05 20:39 EST
Nmap scan report for 192.168.17.178
Host is up (0.040s latency).

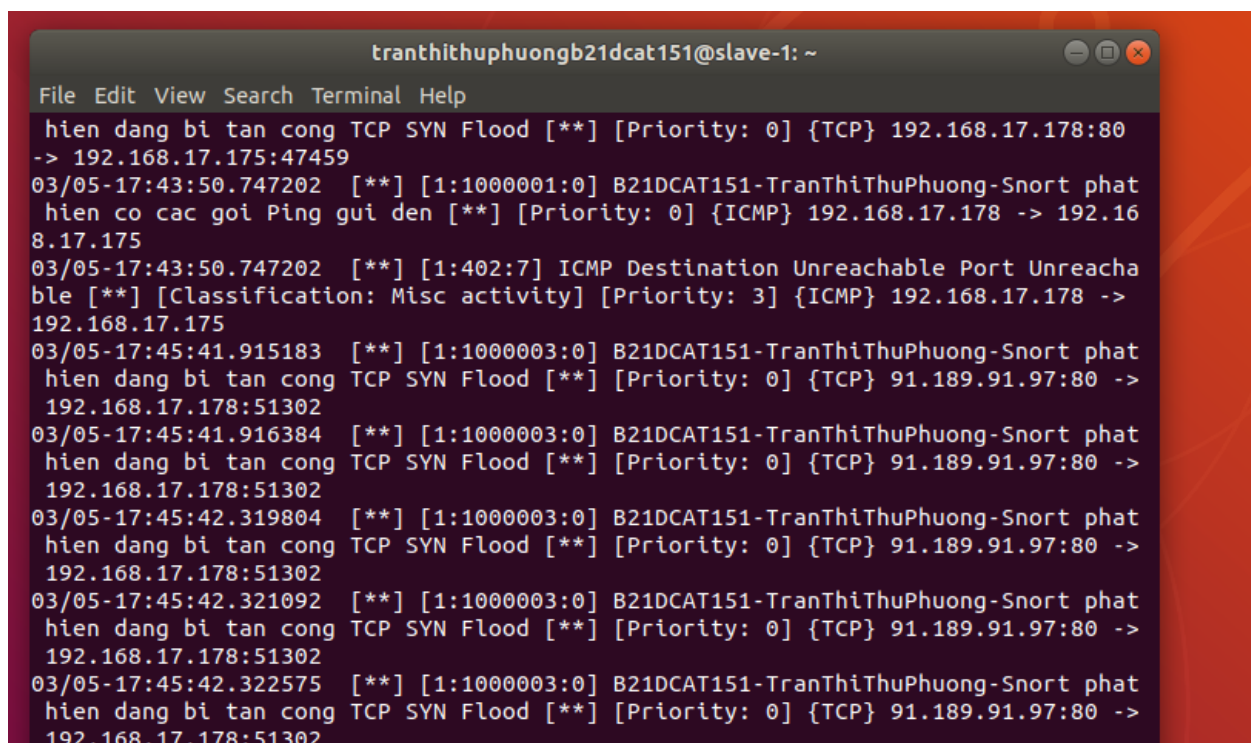
PORT      STATE SERVICE VERSION
80/tcp    closed http

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.87 seconds

(tranhithuphuong151@tranhithuphuong151)~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.17.175 netmask 255.255.255.0 broadcast 192.168.17.255
    inet6 fe80::20c:29ff:febc:ce0a prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:bc:ce:0a txqueuelen 1000 (Ethernet)
    RX packets 1148 bytes 208343 (203.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
```

Hình 4.2.1. Trên máy Kali, sử dụng công cụ nmap để rà quét máy Snort: `nmap -sV -p80 -A 192.168.17.178`

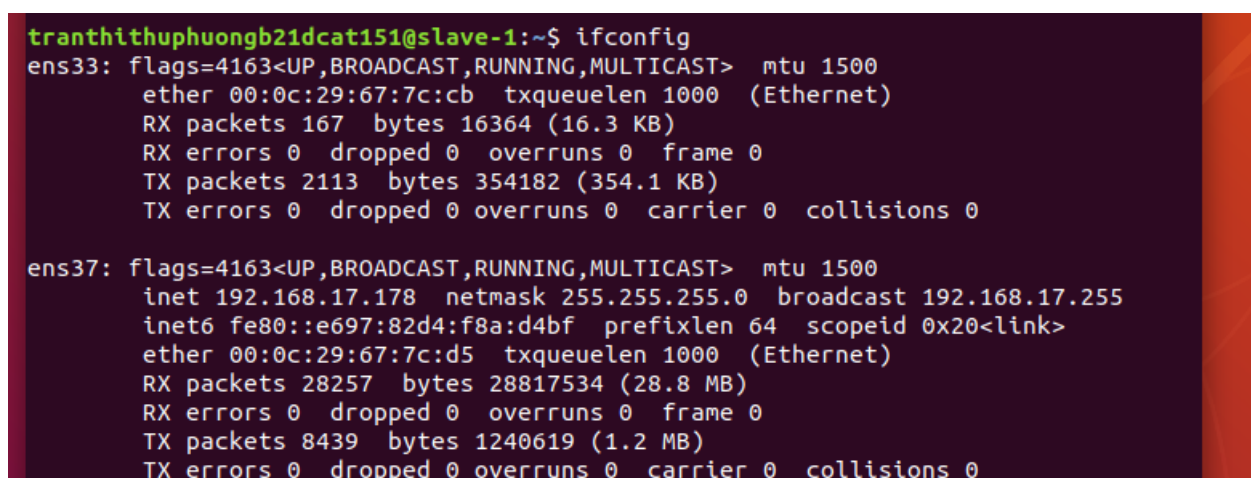




```
tranthithuphuongb21dcat151@slave-1: ~  
File Edit View Search Terminal Help  
hien dang bi tan cong TCP SYN Flood [**] [Priority: 0] {TCP} 192.168.17.178:80  
-> 192.168.17.175:47459  
03/05-17:43:50.747202 [**] [1:1000001:0] B21DCAT151-TranThiThuPhuong-Snort phat  
hien co cac goi Ping gui den [**] [Priority: 0] {ICMP} 192.168.17.178 -> 192.16  
8.17.175  
03/05-17:43:50.747202 [**] [1:402:7] ICMP Destination Unreachable Port Unreacha  
ble [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.17.178 ->  
192.168.17.175  
03/05-17:45:41.915183 [**] [1:1000003:0] B21DCAT151-TranThiThuPhuong-Snort phat  
hien dang bi tan cong TCP SYN Flood [**] [Priority: 0] {TCP} 91.189.91.97:80 ->  
192.168.17.178:51302  
03/05-17:45:41.916384 [**] [1:1000003:0] B21DCAT151-TranThiThuPhuong-Snort phat  
hien dang bi tan cong TCP SYN Flood [**] [Priority: 0] {TCP} 91.189.91.97:80 ->  
192.168.17.178:51302  
03/05-17:45:42.319804 [**] [1:1000003:0] B21DCAT151-TranThiThuPhuong-Snort phat  
hien dang bi tan cong TCP SYN Flood [**] [Priority: 0] {TCP} 91.189.91.97:80 ->  
192.168.17.178:51302  
03/05-17:45:42.321092 [**] [1:1000003:0] B21DCAT151-TranThiThuPhuong-Snort phat  
hien dang bi tan cong TCP SYN Flood [**] [Priority: 0] {TCP} 91.189.91.97:80 ->  
192.168.17.178:51302  
03/05-17:45:42.322575 [**] [1:1000003:0] B21DCAT151-TranThiThuPhuong-Snort phat  
hien dang bi tan cong TCP SYN Flood [**] [Priority: 0] {TCP} 91.189.91.97:80 ->  
192.168.17.178:51302
```

Hình 4.2.2. Trên máy Snort, hiện cảnh báo

- Từ máy Kali, sử dụng công cụ hping3 để tấn công TCP SYN Flood máy Snort (dùng lệnh: hping3 -c 15000 -d 120 -S -w 64 -p 80 --flood --rand-source 192.168.17.178). Trên máy Snort kiểm tra kết quả phát hiện trên giao diện terminal hoặc log của Snort.



```
tranthithuphuongb21dcat151@slave-1:~$ ifconfig  
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
ether 00:0c:29:67:7c:cb txqueuelen 1000 (Ethernet)  
RX packets 167 bytes 16364 (16.3 KB)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 2113 bytes 354182 (354.1 KB)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
ens37: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
inet 192.168.17.178 netmask 255.255.255.0 broadcast 192.168.17.255  
inet6 fe80::e697:82d4:f8a:d4bf prefixlen 64 scopeid 0x20<link>  
ether 00:0c:29:67:7c:d5 txqueuelen 1000 (Ethernet)  
RX packets 28257 bytes 28817534 (28.8 MB)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 8439 bytes 1240619 (1.2 MB)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Hình 4.3.1. Kiểm tra IP máy Snort

```
(tranhthuphuong@tranhthuphuong151)-[~]
$ sudo hping3 -c 15000 -d 120 -S -w 64 -p 80 --flood --rand-source 192.168.17.178
HPING 192.168.17.178 (eth0 192.168.17.178): S set, 40 headers + 120 data bytes
hping in flood mode, no replies will be shown
81: ^C
— 192.168.17.178 hping statistic —
366752 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Hình 4.3.2. Từ máy Kali, sử dụng công cụ hping3 để tấn công TCP SYN Flood máy Snort (dùng lệnh: `hping3 -c 15000 -d 120 -S -w 64 -p 80 --flood --rand-source 192.168.17.178`).

```
tranhthuphuongb21dcat151@slave-1: ~
File Edit View Search Terminal Help
hien dang bi tan cong TCP SYN Flood [**] [Priority: 0] {TCP} 168.199.31.43:3492
9 -> 192.168.17.178:81
03/05-17:51:12.643528 [**] [1:1000003:0] B21DCAT151-TranThiThuPhuong-Snort phat
hien dang bi tan cong TCP SYN Flood [**] [Priority: 0] {TCP} 184.43.213.111:349
30 -> 192.168.17.178:81
03/05-17:51:12.644992 [**] [1:1000003:0] B21DCAT151-TranThiThuPhuong-Snort phat
hien dang bi tan cong TCP SYN Flood [**] [Priority: 0] {TCP} 29.90.21.75:34931
-> 192.168.17.178:81
03/05-17:51:12.644996 [**] [1:1000003:0] B21DCAT151-TranThiThuPhuong-Snort phat
hien dang bi tan cong TCP SYN Flood [**] [Priority: 0] {TCP} 171.132.18.11:3493
2 -> 192.168.17.178:81
03/05-17:51:12.644999 [**] [1:1000003:0] B21DCAT151-TranThiThuPhuong-Snort phat
hien dang bi tan cong TCP SYN Flood [**] [Priority: 0] {TCP} 49.219.74.31:34933
-> 192.168.17.178:81
03/05-17:51:12.645001 [**] [1:1000003:0] B21DCAT151-TranThiThuPhuong-Snort phat
hien dang bi tan cong TCP SYN Flood [**] [Priority: 0] {TCP} 107.88.62.27:34934
-> 192.168.17.178:81
03/05-17:51:12.645003 [**] [1:1000003:0] B21DCAT151-TranThiThuPhuong-Snort phat
hien dang bi tan cong TCP SYN Flood [**] [Priority: 0] {TCP} 147.47.165.72:3493
5 -> 192.168.17.178:81
```

Hình 4.3.3. Trên máy Snort hiện các thông báo

### 3. Kết luận

- Hệ thống phát hiện xâm nhập Snort hoạt động ổn định.
- Các luật mới được tạo và lưu vào trong file luật của Snort.
- Snort phát hiện thành công các rà quét tấn công kẻ trên.

### 4. 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>