**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 VỀ CÀI ĐẶT, CẤU HÌNH NIDS

Sinh viên thực hiện:

B22DCAT199 Đỗ Duy Nam

Giảng viên hướng dẫn: TS.Đinh Trường Duy

**HỌC KỲ 2 NĂM HỌC 2024-2025**

# MỤC LỤC

# DANH MỤC CÁC HÌNH VẼ

# 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 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*

Hệ thống phát hiện tấn công và xâm nhập (Intrusion Detection System – IDS) là một công cụ hoặc giải pháp được thiết kế để giám sát các hoạt động trong mạng hoặc hệ thống máy tính, nhằm phát hiện các hành vi bất thường, đáng ngờ hoặc các cuộc tấn công mạng (như hacking, malware, truy cập trái phép). IDS đóng vai trò như một "hệ thống cảnh báo sớm", giúp bảo vệ cơ sở hạ tầng công nghệ thông tin bằng cách phát hiện và phản ứng với các mối đe dọa. IDS thường được tích hợp với các hệ thống bảo mật khác như tường lửa (firewall) hoặc hệ thống ngăn chặn xâm nhập (IPS - Intrusion Prevention System) để tăng cường khả năng bảo vệ.

*1.2.1.1 Phân loại hệ thống phát hiện xâm nhập*

Dựa trên vị trí triển khai:

- HIDS (Host-based Intrusion Detection System - Hệ thống phát hiện xâm nhập dựa trên máy chủ):
  Được cài đặt trên từng thiết bị riêng lẻ (máy chủ, máy trạm). HIDS giám sát hoạt động nội bộ của thiết bị, chẳng hạn như nhật ký hệ thống (logs), thay đổi tệp, hoặc hành vi của phần mềm.
  - Ưu điểm: Phát hiện tốt các tấn công nội bộ hoặc thay đổi trên máy chủ.
  - Nhược điểm: Không hiệu quả với các tấn công mạng diện rộng.
- NIDS (Network-based Intrusion Detection System - Hệ thống phát hiện xâm nhập dựa trên mạng):
  Được triển khai trên mạng để giám sát lưu lượng mạng (traffic). NIDS phân tích các gói dữ liệu (packets) để tìm kiếm dấu hiệu của tấn công.
  - Ưu điểm: Bảo vệ toàn bộ mạng, phát hiện các cuộc tấn công từ bên ngoài.
  - Nhược điểm: Khó phát hiện các mối đe dọa nội bộ hoặc mã hóa.
- Hệ thống lai (Hybrid IDS):
  Kết hợp cả HIDS và NIDS để tận dụng ưu điểm của cả hai, cung cấp khả năng bảo vệ toàn diện hơn.

Dựa trên chức năng:

- IDS (Intrusion Detection System): Chỉ phát hiện và cảnh báo, không trực tiếp ngăn chặn.
- IPS (Intrusion Prevention System): Không chỉ phát hiện mà còn chủ động chặn các hành vi xâm nhập.

*1.2.1.2 Các kỹ thuật phát hiện xâm nhập*

IDS sử dụng nhiều kỹ thuật khác nhau để nhận diện mối đe dọa. Các kỹ thuật chính bao gồm:

- Phát hiện dựa trên chữ ký (Signature-based Detection):
  So sánh các hoạt động hoặc lưu lượng mạng với cơ sở dữ liệu chứa "chữ ký" (mẫu hành vi) của các cuộc tấn công đã biết (ví dụ: mã độc, khai thác lỗ hổng).
  - Ưu điểm: Chính xác cao với các mối đe dọa đã biết.
  - Nhược điểm: Không phát hiện được các cuộc tấn công mới (zero-day attacks).
- Phát hiện dựa trên bất thường (Anomaly-based Detection):
  Xây dựng mô hình hành vi "bình thường" của hệ thống hoặc mạng, sau đó phát hiện các hoạt động bất thường so với mô hình này.
  - Ưu điểm: Có khả năng phát hiện các mối đe dọa chưa biết.
  - Nhược điểm: Dễ tạo ra cảnh báo giả (false positives) nếu hành vi bình thường thay đổi.
- Phát hiện dựa trên quy tắc (Rule-based Detection):
  Sử dụng các quy tắc hoặc chính sách được định nghĩa trước để phát hiện hành vi đáng ngờ (ví dụ: nhiều lần đăng nhập thất bại).
  - Ưu điểm: Dễ tùy chỉnh theo nhu cầu cụ thể.
  - Nhược điểm: Yêu cầu cập nhật thường xuyên và không linh hoạt với các cuộc tấn công phức tạp.
- Phát hiện dựa trên học máy (Machine Learning-based Detection):
  Sử dụng các thuật toán học máy để phân tích dữ liệu, học hỏi từ các mẫu tấn công và dự đoán mối đe dọa.
  - Ưu điểm: Linh hoạt, thích nghi với các mối đe dọa mới.
  - Nhược điểm: Đòi hỏi dữ liệu lớn để huấn luyện và có thể phức tạp trong triển khai.

### *1.2.2 Kiến trúc và tính năng của một số hệ thống phát hiện tấn công, xâm nhập*

*1.2.2.1 Snort*

Kiến trúc:

- Mô hình module: Snort hoạt động theo kiến trúc phân cấp gồm các module chính:
  - Packet Decoder: Giải mã các gói tin từ mạng (Ethernet, IP, TCP, UDP, v.v.).
  - Preprocessors: Xử lý trước dữ liệu (tái hợp gói tin, chuẩn hóa giao thức, phát hiện bất thường cơ bản).
  - Detection Engine: So sánh dữ liệu với các quy tắc (rules) để phát hiện xâm nhập.
  - Logging/Alerting: Ghi log hoặc gửi cảnh báo (syslog, file, cơ sở dữ liệu).
  - Output Modules: Xuất kết quả dưới nhiều định dạng (XML, tcpdump, v.v.).
- NIDS: Snort chủ yếu là hệ thống phát hiện xâm nhập dựa trên mạng (Network Intrusion Detection System).

Tính năng:

- Phát hiện dựa trên chữ ký (signature-based) với hàng nghìn quy tắc được cộng đồng cập nhật.
- Hỗ trợ phân tích giao thức và tìm kiếm nội dung (content matching).
- Có thể hoạt động ở 3 chế độ: sniffer (bắt gói tin), packet logger (ghi log), và NIDS (phát hiện xâm nhập).
- Tích hợp với IPS (Intrusion Prevention System) khi kết hợp với công cụ như SnortSam.
- Mã nguồn mở, chạy trên nhiều nền tảng (Linux, Windows, BSD, v.v.).

*1.2.2.2 Suricata*

Kiến trúc:

- Đa luồng (multi-threaded): Suricata được thiết kế để tận dụng nhiều CPU, tăng hiệu suất so với Snort.
- Các thành phần chính:
  - Capture Layer: Thu thập gói tin từ mạng (sử dụng libpcap, PF_RING, hoặc AF_PACKET).
  - Decoding & Normalization: Giải mã và chuẩn hóa dữ liệu.
  - Detection Engine: Hỗ trợ cả chữ ký (signature) và bất thường (anomaly).
  - Output Layer: Ghi log (JSON, EVE, Unified2) hoặc cảnh báo.
- NIDS/IPS: Có thể hoạt động như cả IDS và IPS.

Tính năng:

- Hỗ trợ phát hiện dựa trên chữ ký (tương thích với quy tắc Snort) và phát hiện bất thường.
- Phân tích giao thức sâu (HTTP, TLS, DNS, SMB, v.v.).
- Hỗ trợ xử lý lưu lượng mã hóa (TLS/SSL inspection).
- Tích hợp với các công cụ SIEM (Security Information and Event Management) thông qua định dạng log JSON.
- Chặn tấn công thời gian thực khi ở chế độ IPS.

*1.2.2.3 Zeek*

Kiến trúc:

- Event-driven: Zeek không dựa trên chữ ký mà phân tích mạng theo sự kiện (event-based).
- Các thành phần:
    - Packet Capture: Thu thập lưu lượng mạng (libpcap).
    - Protocol Analysis: Phân tích chi tiết các giao thức (HTTP, DNS, SMTP, v.v.).
    - Scripting Engine: Sử dụng ngôn ngữ Zeek Script để định nghĩa logic phát hiện.
    - Logging: Ghi lại thông tin chi tiết dưới dạng log (text, JSON).
- Network Security Monitor: Zeek tập trung vào giám sát và phân tích hành vi mạng hơn là phát hiện tức thời.

Tính năng:

- Phân tích giao thức sâu, tạo ra log chi tiết về hoạt động mạng (connection logs, HTTP requests, DNS queries, v.v.).
- Không dựa trên chữ ký, mà dựa trên hành vi và ngữ cảnh (context-aware).
- Hỗ trợ tùy chỉnh qua script để phát hiện các mối đe dọa cụ thể.
- Tích hợp tốt với SIEM và các hệ thống phân tích dữ liệu lớn.

*1.2.2.4 OSSEC*

Kiến trúc:

- HIDS: OSSEC là hệ thống phát hiện xâm nhập dựa trên máy chủ (Host-based Intrusion Detection System).
- Các thành phần:
    - Agents: Cài đặt trên các máy chủ để giám sát log, file, và hoạt động hệ thống.
    - Server: Thu thập và phân tích dữ liệu từ các agent.
    - Analysis Engine: Phát hiện bất thường hoặc vi phạm dựa trên quy tắc.

    - Response Module: Thực hiện phản ứng (chặn, cảnh báo).
- Client-Server Model: Hỗ trợ triển khai phân tán.

Tính năng:

- Giám sát tính toàn vẹn của file (file integrity monitoring).
- Phân tích log hệ thống (system logs) và phát hiện rootkit.
- Phát hiện dựa trên quy tắc và bất thường.
- Hỗ trợ phản ứng chủ động (active response) như chặn IP hoặc chạy script.
- Chạy trên nhiều nền tảng (Linux, Windows, macOS, v.v.).

*1.2.2.5 Wazuh*

Kiến trúc:

- HIDS cải tiến từ OSSEC: Wazuh là một nhánh nâng cao của OSSEC, bổ sung nhiều tính năng.
- Các thành phần:
    - Agents: Thu thập dữ liệu từ máy chủ (log, file, hoạt động).
    - Manager: Phân tích dữ liệu và quản lý quy tắc.
    - RESTful API: Tích hợp với các hệ thống khác.
    - Kibana Dashboard: Giao diện trực quan hóa dữ liệu.
- Tích hợp với Elastic Stack: Kết hợp với Elasticsearch, Logstash, và Kibana.

Tính năng:

- Giám sát tính toàn vẹn file, phát hiện rootkit, và phân tích log.
- Hỗ trợ phát hiện dựa trên quy tắc (signature-based) và bất thường (anomaly-based).
- Tích hợp với các nguồn dữ liệu threat intelligence.
- Cung cấp dashboard trực quan và báo cáo chi tiết.
- Hỗ trợ phản ứng tự động (active response).

## 1.3 Kết luận

Ở chương này đã tìm hiểu khái quát về các hệ hệ thống phát hiện tấn công, xâm nhập, phân loại các hệ thống phát hiện xâm nhập, các kỹ thuật phát hiện xâm nhập. Bên cạnh đó còn tìm hiểu về kiến trúc và tính năng của một số hệ thống phát hiện tấn công, xâm nhập, như Snort, Suricata, Zeek, OSSEC, Wazuh.

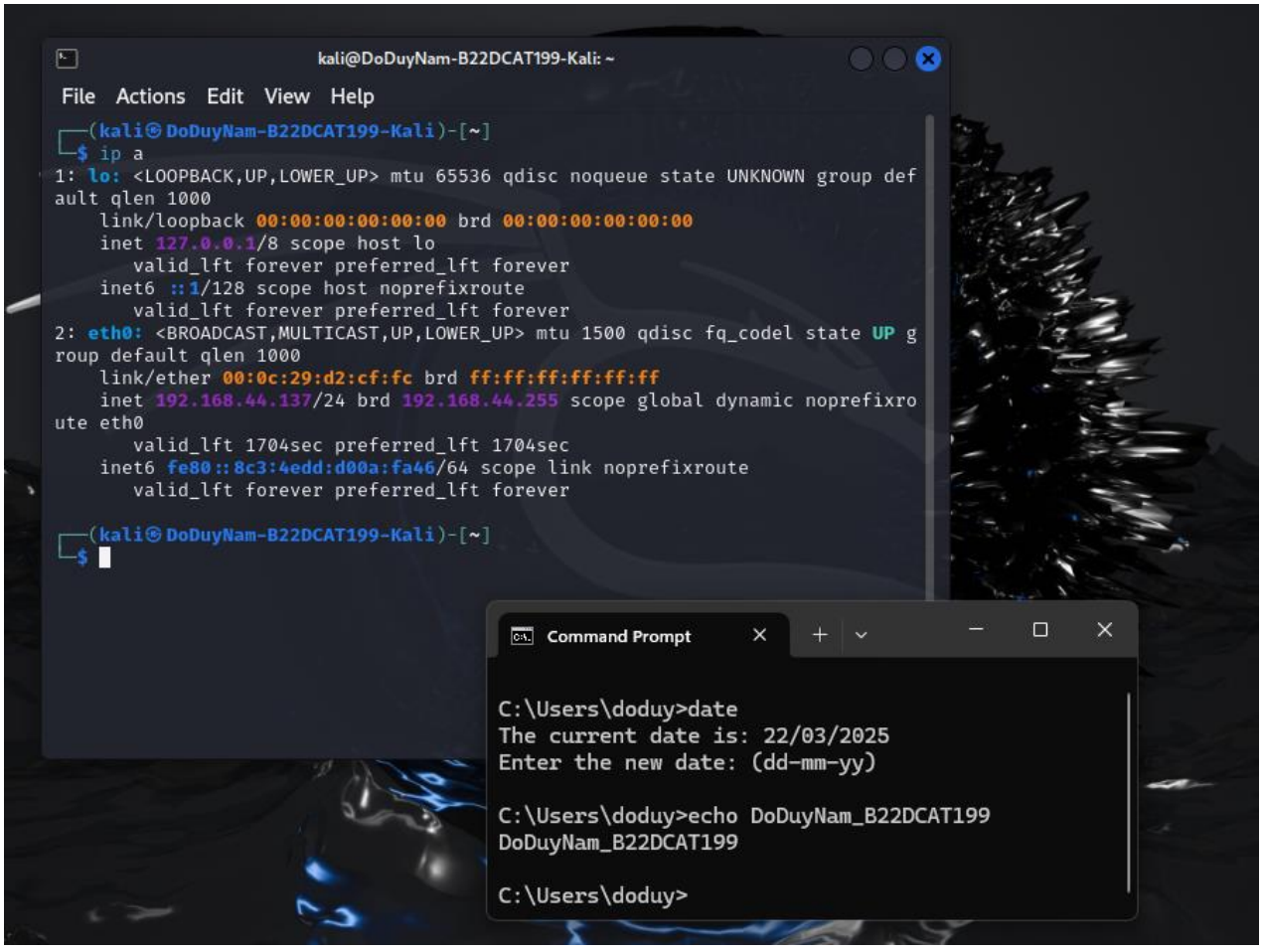
# CHƯƠNG 2. NỘI DUNG THỰC HÀNH

## 2.1 Chuẩn bị môi trường.

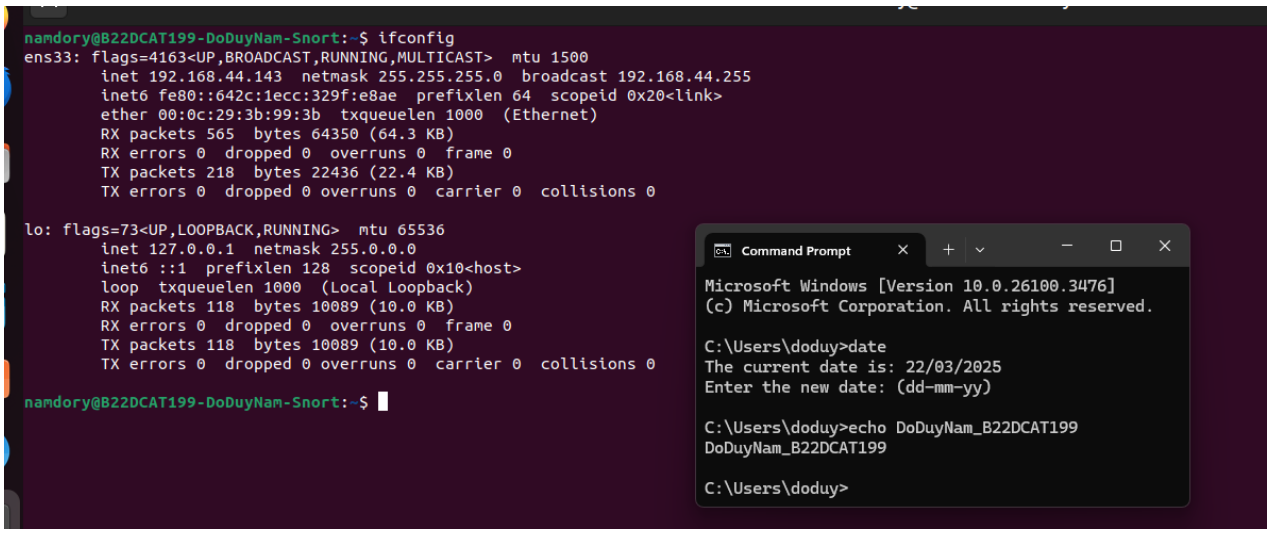
- Máy Linux với RAM tối thiểu 2GB, 10GB đĩa cứng có kết nối mạng (LAN hoặc Internet)
- Máy Kali Linux (bản 2021 trở lên)
- Bộ phần mềm Snort

## 2.2 Các bước thực hiện

- Bước 1: Chuẩn bị các máy tính như mô tả trong mục 2.1. Máy Kali Linux được đổi tên thành B22DCAT199_DoDuyNam-Kali và máy cài Snort thành B22DCAT199_DoDuyNam-Snort. Các máy có địa chỉ IP và kết nối mạng LAN.

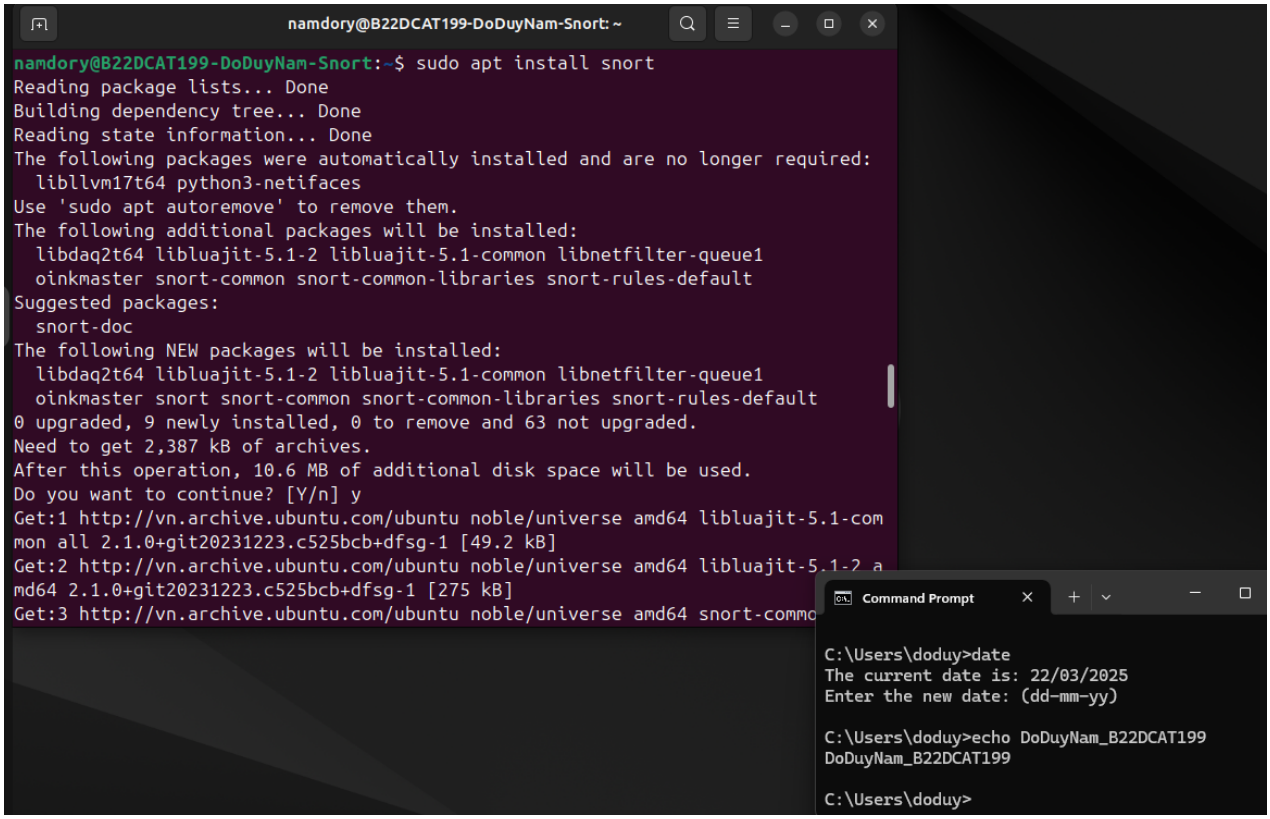



*Hình 1 Máy Kali Linux*



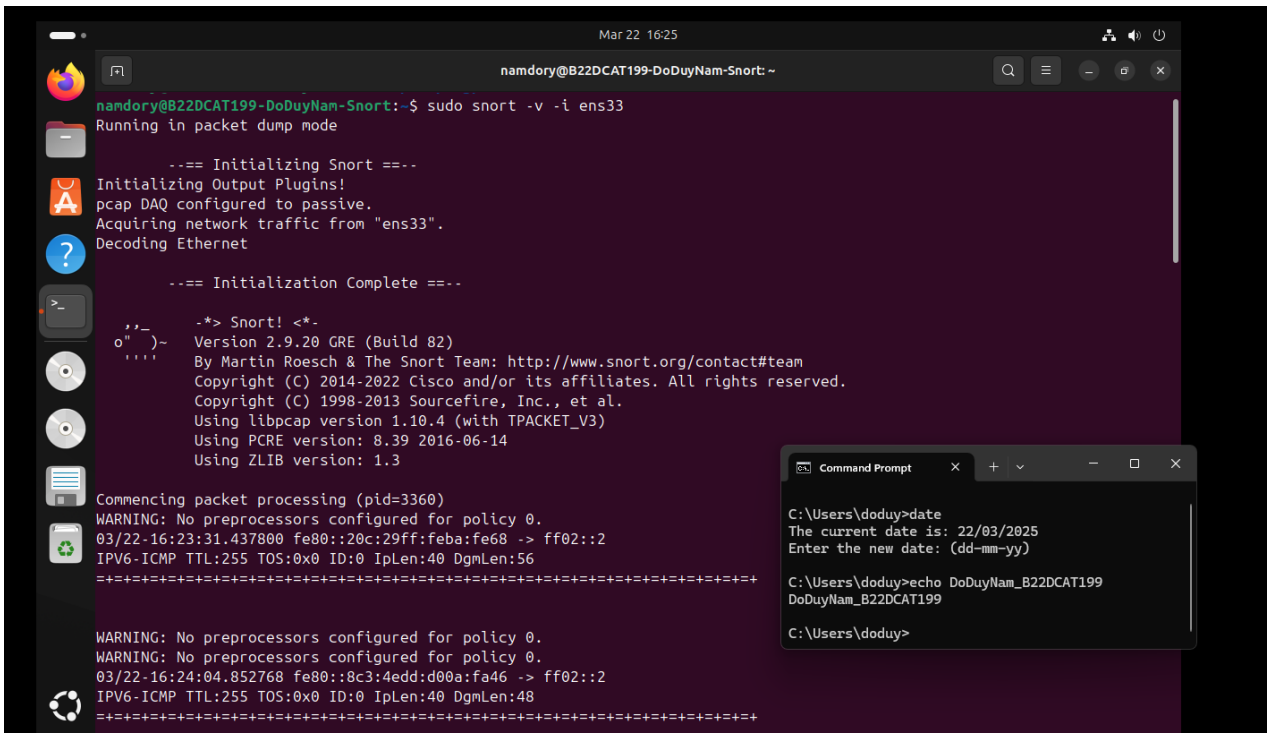

*Hình 2 Máy cài Snort*

- Bước 2: Tải, cài đặt Snort và chạy thử Snort. Kiểm tra log của Snort để đảm bảo Snort hoạt động bình thường.
  + Sử dụng câu lệnh *sudo apt install snort* để tiến hành cài đặt Snort




*Hình 3 Cài đặt Snort*

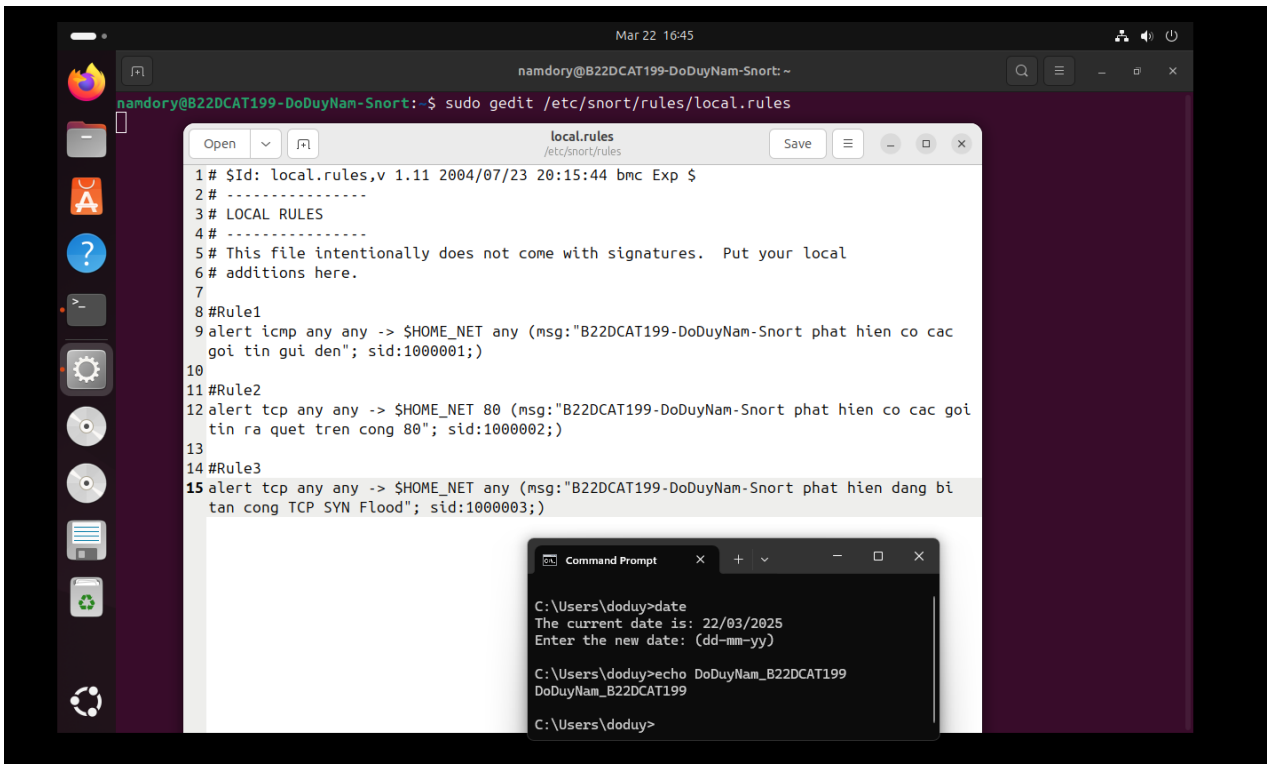
+ Sử dụng câu lệnh *sudo snort -v -i ens33* để chạy thử Snort




*Hình 4 Chạy thử Snort*

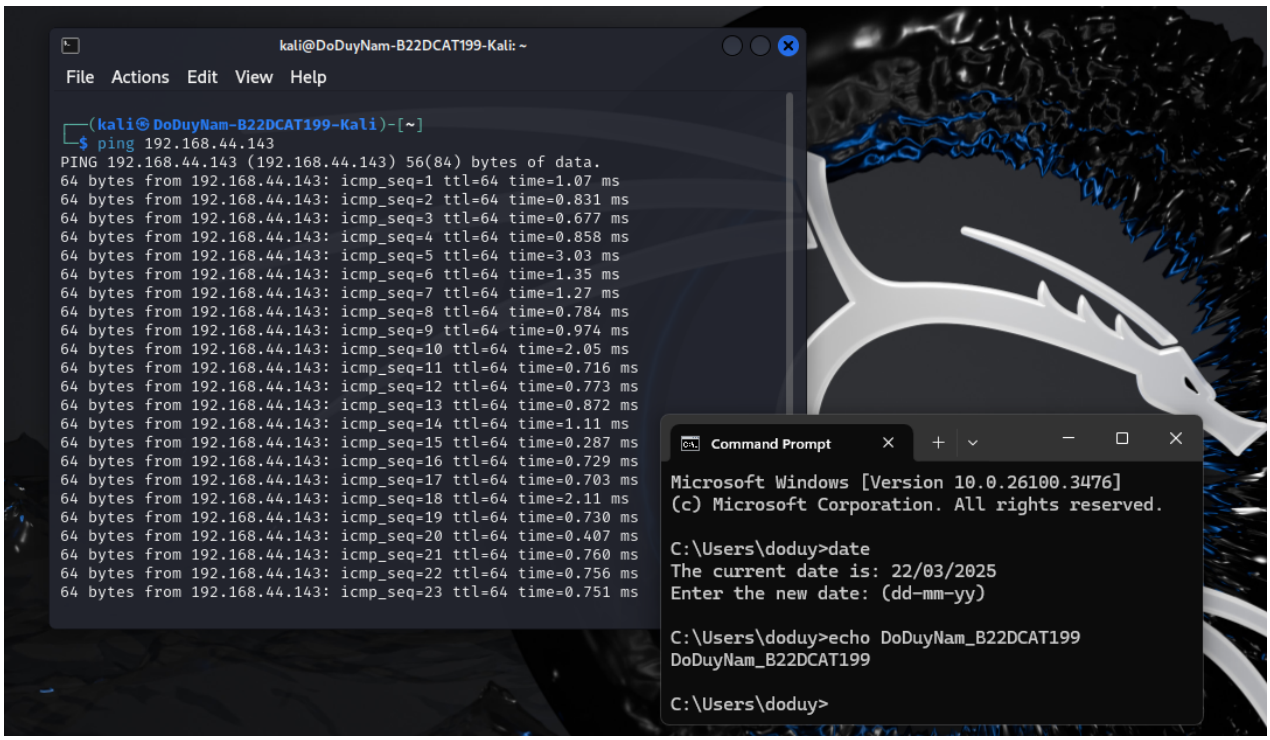
- Bước 3: Tạo các luật Snort để phát hiện 3 dạng rà quét, tấn công hệ thống:
  + Phát hiện các gói tin ping từ bất kỳ một máy nào gửi đến máy chạy Snort. Hiển thị thông điệp khi phát hiện: “B22DCAT199-DoDuyNam-Snort phát hiện có các gói Ping gửi đến.”
  + Phát hiện các gói tin rà quét từ bất kỳ một máy nào gửi đến máy chạy Snort trên cổng 80. Hiển thị thông điệp khi phát hiện: B22DCAT199-DoDuyNam-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. Hiển thị thông điệp khi phát hiện: “B22DCAT199-DoDuyNam-Snort phát hiện đang bị tấn công TCP SYN Flood.”

  Sử dụng câu lệnh *sudo gedit /etc/snort/rules/local.rules* để tạo các luật trên

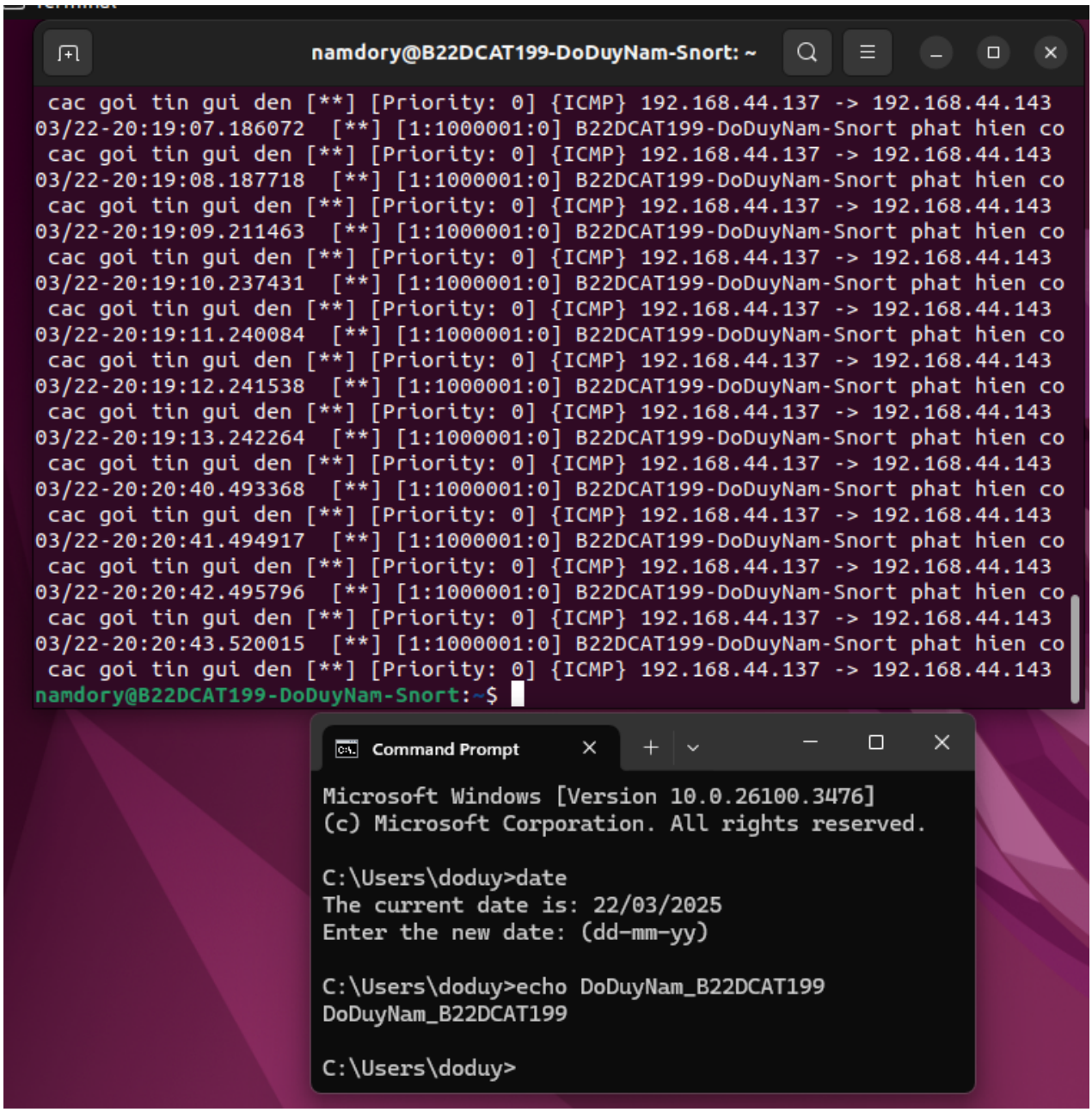



*Hình 5 Tạo luật cho Snort*

- 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.

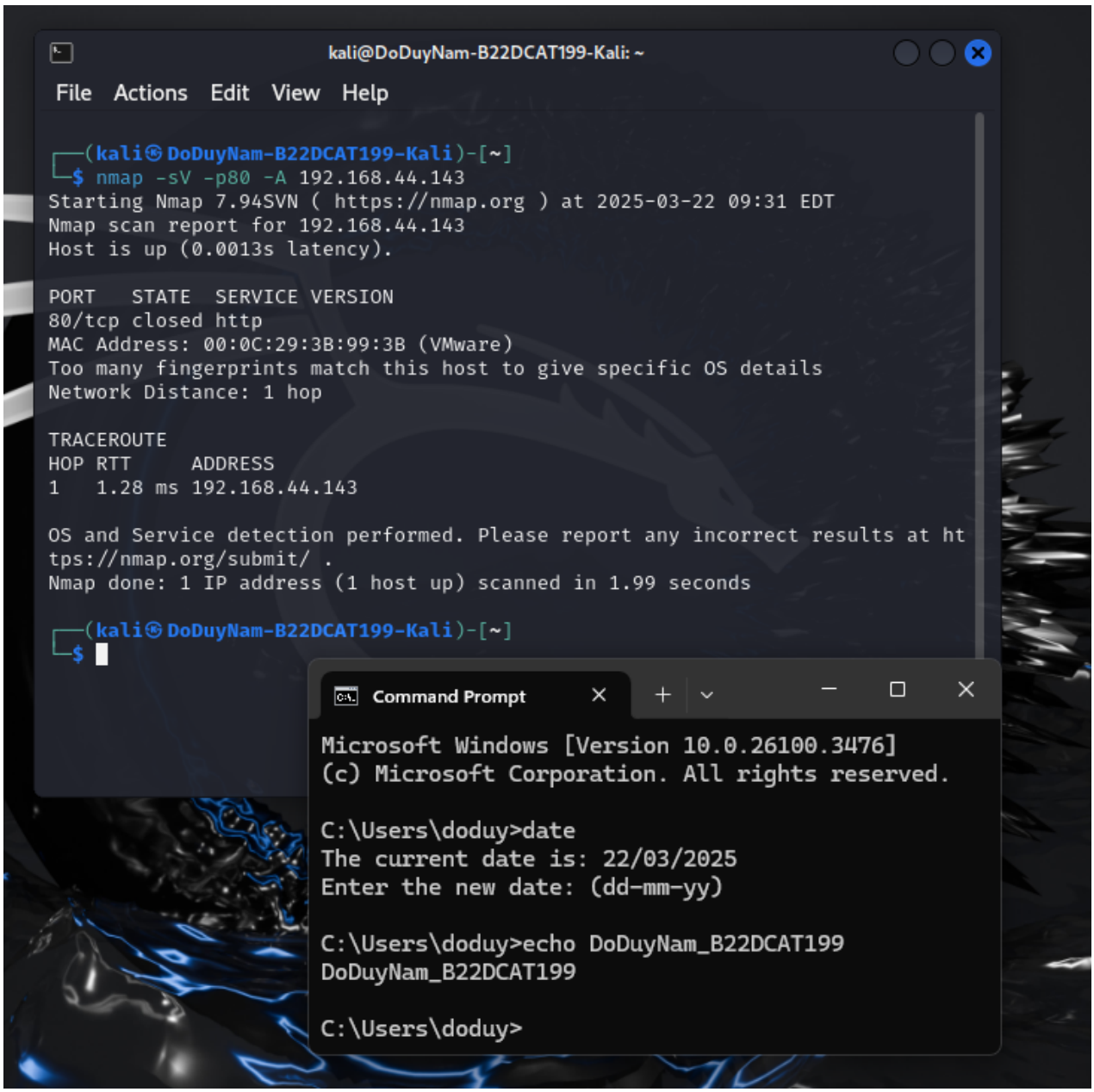



*Hình 6 Trên máy Kali ping tới máy Snort*

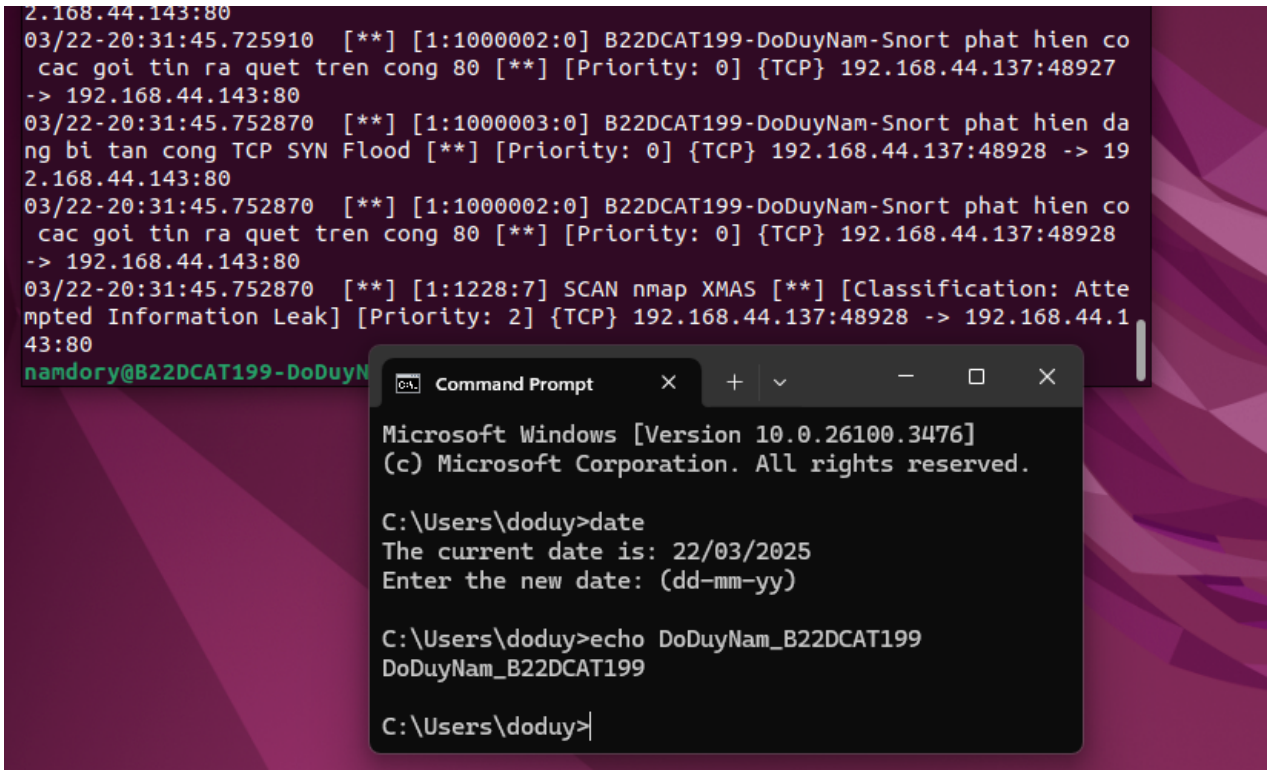



*Hình 7 Trên máy Snort xuất hiện các cảnh báo*

+ Từ máy Kali, sử dụng công cụ nmap để rà quét máy Snort (dùng lệnh: nmap -sV -p80 -A 192.168.44.143). Trên máy Snort kiểm tra kết quả phát hiện trên giao diện terminal hoặc log của Snort.

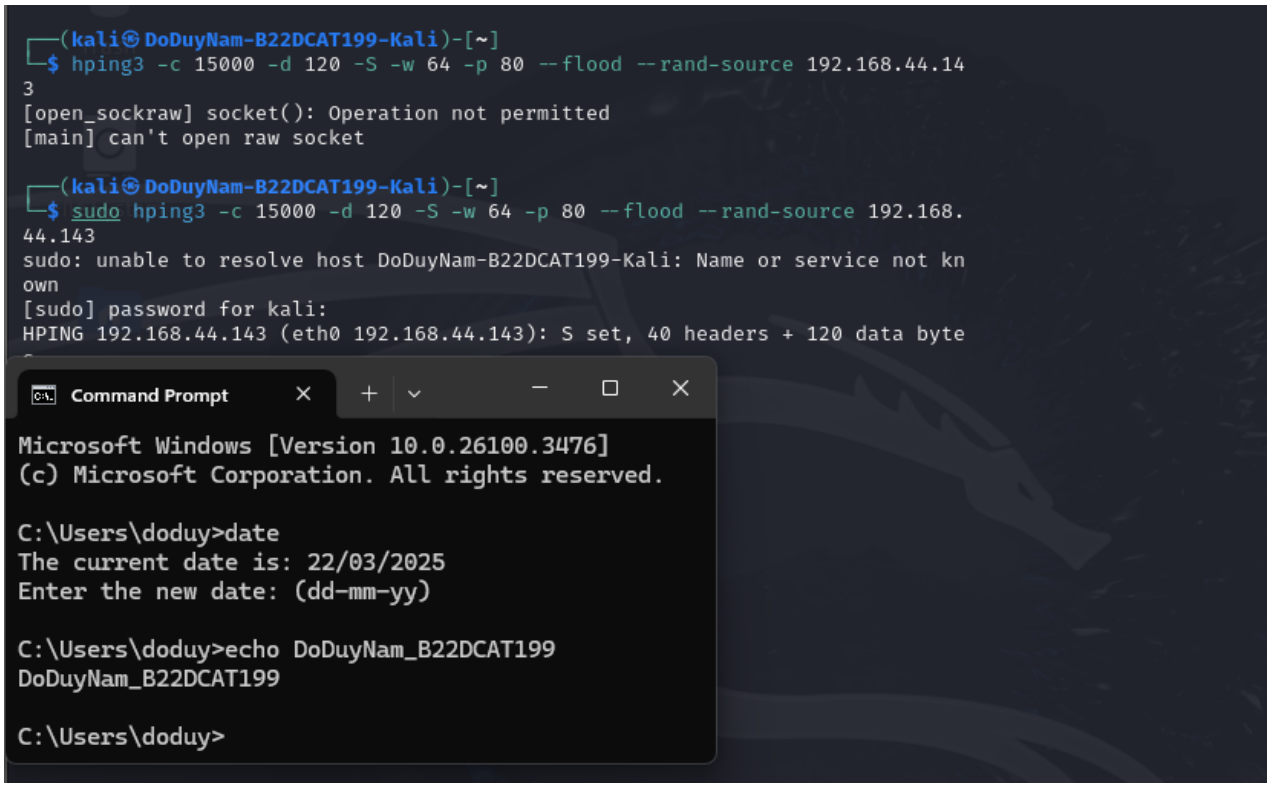



*Hình 8 Trên máy Kali sử dụng công cụ nmap để rà quét trên máy Snort*

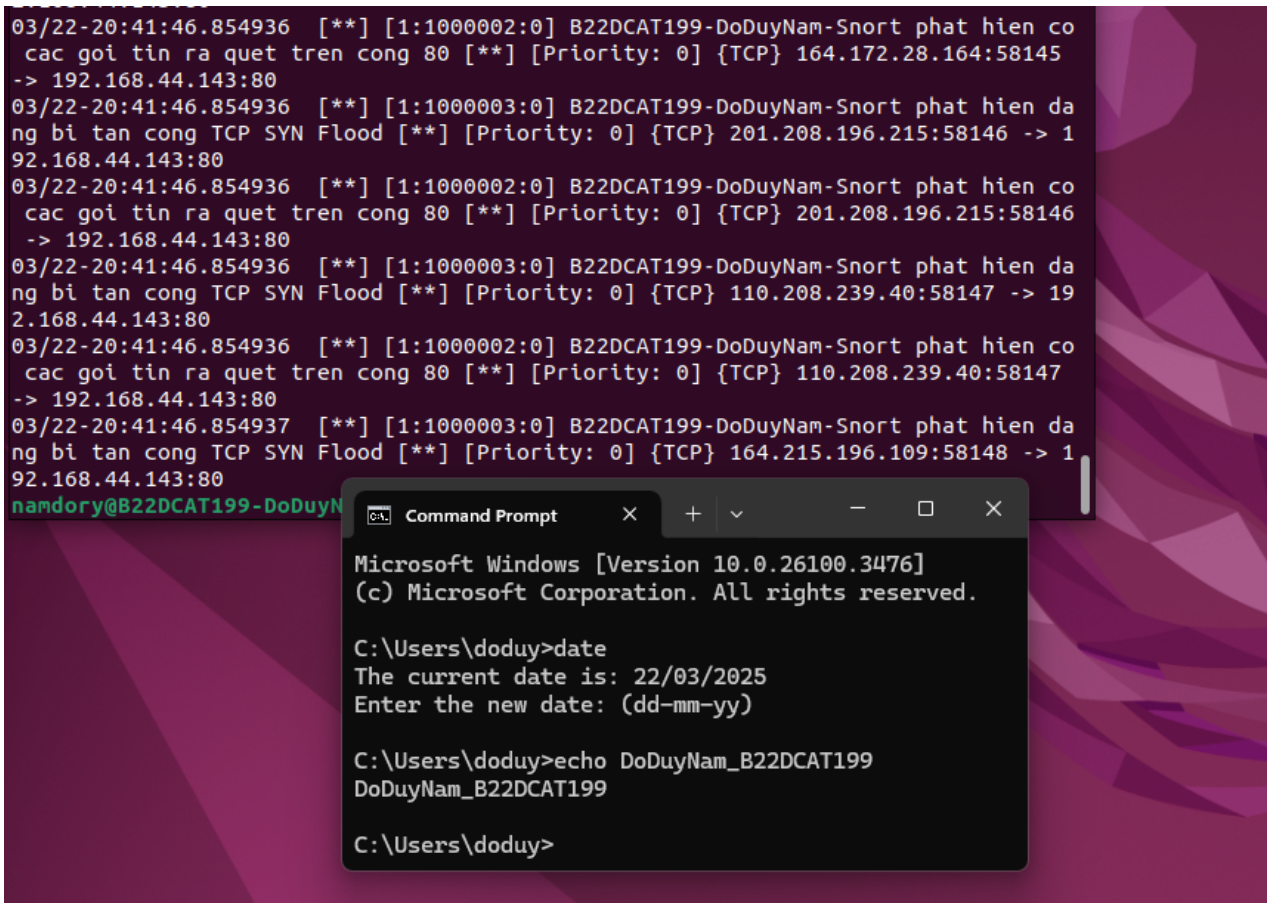



*Hình 9 Trên máy Snort xuất hiện các cảnh báo*

+ 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 192.168.44.143). Trên máy Snort kiểm tra kết quả phát hiện trên giao diện terminal hoặc log của Snort.




*Hình 10 Trên máy Kali sử dụng công cụ hping3 để tấn công TCP SYN Flood máy Snort*




*Hình 11 Trên máy Snort xuất hiện các cảnh báo*

## 2.3 Kết luận

Ở chương này đã hướng dẫn cài đặt và chạy thử Snort và tạo các luật để phát hiện 3 dạng rà quét, tấn công hệ thống, dựa vào đó để thử thi tấn công và phát hiện bằng việc sử dụng Snort.

# KẾT LUẬN

- Tìm hiểu khát quát về các hệ thống phát hiện tấn công, xâm nhập, phân loại các các hệ thống phát hiện xâm nhập, các kỹ thuật phát hiện xâm nhập.
- Tìm hiểu về kiến trúc và tính năng của một số hệ thống phát hiện tấn công, xâm nhập, như Snort, Suricata, Zeek, OSSEC, Wazuh...
- Cài đặt thành công
- 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

# 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