

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG



BÁO CÁO ĐỒ ÁN

Môn: QUẢN TRỊ MẠNG VÀ HỆ THỐNG

Đề tài: Snort

Giảng viên hướng dẫn: Th.S. Trần Thị Dung

Nhóm: 09

Sinh viên thực hiện:

1. Nguyễn Xuân Huy – 22520568
2. Phan Thanh Hương – 22520531
3. Nguyễn Khang Hưng – 22520515
4. Nguyễn Thanh Kiệt – 22520720

MỤC LỤC

I. Tổng quan.....	4
1.1. Giới thiệu về IDS, IPS	5
1.1.1. Khái niệm Snort.....	6
1.1.2. Đặc điểm của Snort.....	6
1.2. Thành phần	6
1.3. Hoạt động	8
1.4. Bộ luật của Snort	10
II. Triển khai.....	11
2.1. Mô hình.....	11
2.1.1. Cơ bản.....	11
2.1.2. Nâng cao	12
2.2. Cài đặt.....	12
2.2.1. Ubuntu	12
2.2.2. Kali linux	13
2.2.3. Window	14
2.3. Cấu hình	16
III. Kết quả và kết luận.....	18
3.1. Kết quả.....	18
3.1.1. Cơ bản.....	18
3.1.2. Nâng cao	25
3.2. Kết luận.....	27
IV. Trả lời câu hỏi	28
V. Tài liệu tham khảo.....	29

Bảng phân chia công việc

Thành viên nhóm	Nhiệm vụ	Mức độ hoàn thành
Nguyễn Xuân Huy	<ul style="list-style-type: none"> - Tìm hiểu về Snort. - Làm ppt. - Thuyết trình. - Chạy và quay demo. - Viết report. 	100%
Nguyễn Khang Hưng	<ul style="list-style-type: none"> - Tìm hiểu về Snort. - Soạn nội dung. - Tìm hiểu demo. - Viết report. 	100%
Nguyễn Thanh Kiệt	<ul style="list-style-type: none"> - Tìm hiểu về Snort. - Làm ppt. - Kiến trúc Snort và một số lý thuyết cơ bản. 	100%
Phan Thanh Hương	<ul style="list-style-type: none"> - Bộ rule của Snort. - Các chế độ hoạt động của Snort. - Làm ppt. - Viết report. 	100%

Tiêu chí đánh giá

Tiêu chí	Mức điểm
Report format	1
Presentation	1
Theory	2
Demonstration	5
Quiz after presentation	1

I. Tổng quan

Để ngăn chặn những truy cập không mong muốn, chúng ta đã có tường lửa, tường lửa là một phần cơ bản của hệ thống bảo mật mạng, có chức năng kiểm soát và quản lý luồng dữ liệu giữa các mạng khác nhau hoặc giữa mạng nội bộ và mạng bên ngoài. Tường lửa sử dụng các quy tắc và chính sách để xác định quyền truy cập và ngăn chặn luồng dữ liệu không mong muốn hoặc nguy hiểm. Tuy nhiên, tường lửa không có khả năng phát hiện hoặc chống lại các hoạt động xâm nhập, tường lửa chỉ kiểm tra gói tin và không có khả năng phát hiện những mẫu tấn công tinh vi hoặc các hành vi bất thường. Để bổ sung thêm đó, chúng ta có kỹ thuật phát hiện xâm nhập IDS.

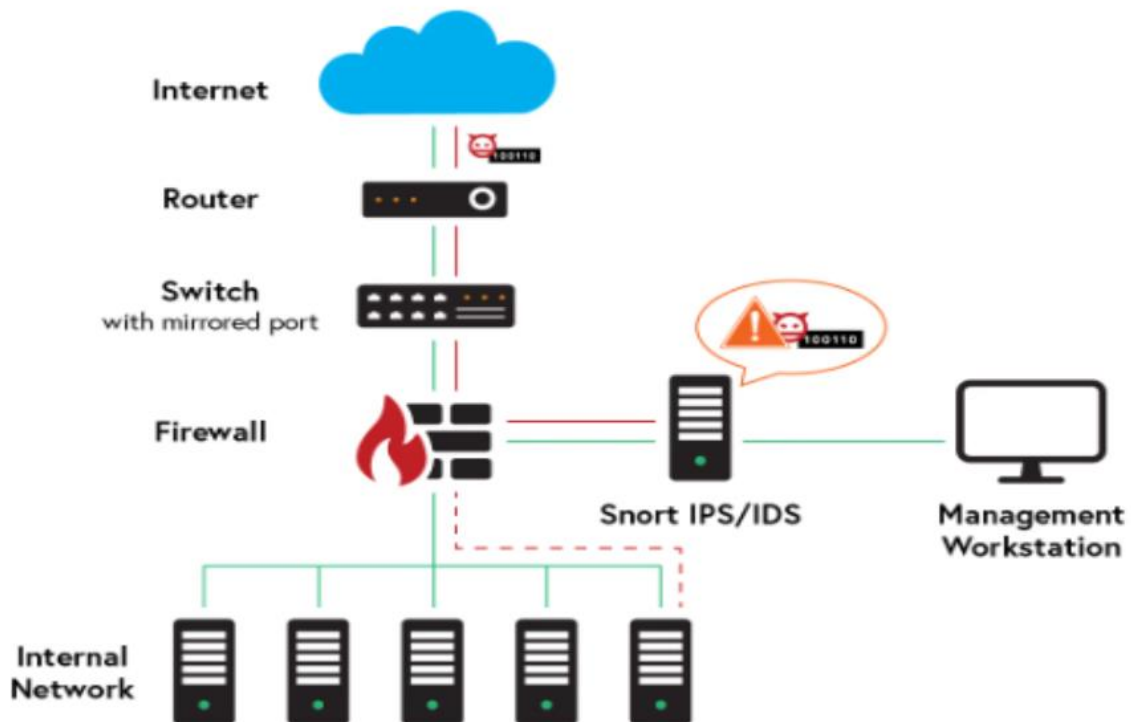
Trong đời sống, tường lửa được ví như hàng rào bảo vệ ngôi nhà, ổ khóa cửa. Tuy nhiên kẻ trộm có thể dùng bất cứ hành vi nào để có thể vượt qua những thứ trên và đột nhập nhà. IDS được ví như chuông báo động, camera quan sát giúp chúng ta biết được các phương pháp mà chúng có thể đột nhập. Tương tự IDS giúp chúng ta phát hiện và cảnh báo ra được các dấu hiệu bất thường, giám sát và phân tích các hoạt động ra vào của hệ thống để có thể ngăn chặn kịp thời.

Để nâng cao khả năng bảo vệ, ngoài IDS, hệ thống ngăn chặn xâm nhập (IPS - Intrusion Prevention System) được phát triển nhằm không chỉ phát hiện mà còn chủ động ngăn chặn các hành động xâm nhập nguy hiểm. IPS là một bước tiến so với IDS khi nó không chỉ giám sát và phân tích lưu lượng mạng mà còn tự động thực thi các biện pháp để ngăn chặn các mối đe dọa ngay lập tức, chẳng hạn như chặn các gói tin độc hại hoặc cắt đứt ngay lập tức nếu phát hiện kết nối đó đáng ngờ.

Nếu ví IDS là chuông báo động, thì IPS chính là người gác cổng chủ động phản ứng lại các mối đe dọa, không để chúng tiếp cận sâu hơn vào hệ thống. Điều này giúp bảo vệ hệ thống một cách toàn diện hơn, đặc biệt trong các môi trường yêu cầu tính bảo mật cao.

Snort là một công cụ mã nguồn mở nổi bật, được sử dụng rộng rãi nhờ khả năng kết hợp giữa IDS và IPS. Snort giúp phát hiện xâm nhập dựa trên các chữ ký tấn công và hành vi bất thường, đồng thời có khả năng ngăn chặn các gói tin nguy hiểm theo thời gian thực khi hoạt động. Điểm mạnh của Snort nằm ở tính linh hoạt cao, cho phép người dùng tùy chỉnh các quy tắc phù hợp với từng môi trường bảo mật, cũng như khả năng được cộng đồng hỗ trợ và cập nhật thường xuyên. Với những ưu điểm này, Snort trở thành một giải pháp mạnh mẽ, tiết kiệm chi phí và hiệu quả cho việc bảo vệ mạng trong nhiều tổ chức.

Simple Snort Network Topology



1.1. Giới thiệu về IDS, IPS

IDS các công cụ phần mềm hoặc thiết bị, được thiết kế cho việc phân tích các dòng dữ liệu hoặc các biến cố, nhằm xác định các hành động trái phép và các vi phạm chính sách an ninh tổ chức trên hệ thống mạng máy tính. IDS hoạt động bằng cách phân tích lưu lượng mạng hoặc hành vi hệ thống, sử dụng các thông tin được tập hợp, so sánh với các mẫu tấn công đã biết hoặc các hành vi bất thường để xác định các nguy cơ tiềm tàng. IDS sẽ gửi cảnh báo tới quản trị viên mạng để họ có thể xử lý kịp thời.

IDS phát hiện các xâm nhập bằng 2 phương pháp chính là dựa trên các dấu hiệu hay còn gọi là signature của gói tin, phương pháp này sẽ thực hiện bằng cách so sánh dòng dữ liệu với các mẫu tấn công đã biết trước, ví dụ như trong snort sẽ có tệp rule chứa các quy tắc mô tả các mẫu tấn công. Phương pháp thứ 2 là dựa trên các dấu hiệu về các hành động bất thường, gửi cảnh báo khi phát hiện ra các hoạt động vượt ra ngoài phạm vi bình thường, IDS sẽ giám sát mạng hoặc hệ thống trong một khoảng thời gian nhất định để thu thập dữ liệu: lưu lượng mạng, số lần truy cập máy chủ hoặc dịch vụ, tốc độ truy cập thông thường.

IDS có thể được phân loại dựa theo chức năng, gồm 2 loại: NIDS(network-based IDS) và HIDS(host-based IDS). Mỗi loại đều có cách tiếp cận riêng để theo dõi và bảo vệ dữ liệu.

NIDS: Là hệ thống phát hiện xâm nhập mạng. NIDS giám sát và phân tích lưu lượng mạng để phát hiện các hoạt động xâm nhập hoặc bất thường trên mạng.

HIDS: Là hệ thống phát hiện xâm nhập trên máy chủ có tài nguyên quan trọng và dễ bị tấn công, ví dụ như application server, web server, mail server. HIDS giám sát và phân

tích hoạt động của một máy chủ hoặc thiết bị đơn lẻ để phát hiện các hoạt động xâm nhập hoặc bất thường trên máy chủ.

Ưu điểm: giúp phát hiện sớm các hoạt động xâm nhập, ghi lại và phân tích, đảm bảo sự hoạt động ổn định và tin cậy của hệ thống.

Khác với IDS thì IPS ngoài là hệ thống phát hiện xâm nhập mà còn có thêm khả năng ngăn chặn các cuộc tấn công hoặc các hành vi bất thường mà nó bắt được, là một phiên bản mở rộng và nâng cao hơn của hệ thống IDS. Hoạt động theo kiểu tự động thực hiện các biện pháp phòng ngừa như ngăn chặn kết nối cắt ngắn chuỗi hoặc thậm chí xóa luôn các gói tin chứa mẫu xâm nhập hoặc hành vi không bình thường. IPS có thể triển khai dưới dạng Network IPS (NIPS) hoặc Host IPS (HIPS).

1.1.1. Khái niệm Snort

Snort là 1 hệ thống phòng chống xâm nhập (IPS) với mã nguồn mở, sử dụng 1 loạt các quy tắc để xác định được các hoạt động độc hại diễn ra trên hệ thống mạng, sử dụng các quy tắc để tìm gói tin khớp với chúng, sau đó tạo cảnh báo cho người dùng. Snort cung cấp khả năng phát hiện tấn công, ghi log gói tin, từ đó giúp người dùng có thể phát hiện sớm và tìm cách ngăn chặn các hoạt động gây nguy hiểm đến hệ thống.

1.1.2. Đặc điểm của Snort

Snort là một công cụ có mã nguồn mở, hoàn toàn miễn phí và có thể tải và cài đặt ở trên hầu hết các hệ điều hành phổ biến hiện nay: Linux/Unix, Ubuntu, MacOS, Window,...

Snort có một đặc điểm rất quan trọng là kiến trúc module hóa, kiến trúc của Snort bao gồm các module hoạt động riêng lẻ nhưng không độc lập với nhau, mỗi module thực hiện một nhiệm vụ cụ thể của nó, và Snort sẽ luôn vận hành các hệ thống module này sao cho chúng có khả năng phối hợp hiệu quả và liên kết với nhau tốt tạo nên bộ máy hoàn chỉnh, chính nhờ đặc điểm đó nên chúng ta có thể thêm, sửa, cập nhật các module mà không lo gây ảnh hưởng đến hệ thống.

1.2. Thành phần

Kiến trúc của snort bao gồm 4 phần cơ bản sau:

- a. Packet Sniffer (thụ thập gói tin)
- b. Preprocessors (tiền xử lý)
- c. Detection Engine (công cụ phát hiện)
- d. Output (ghi log và cảnh báo, giao diện người dùng)



The core component that collects and identifies packet structures from network traffic.

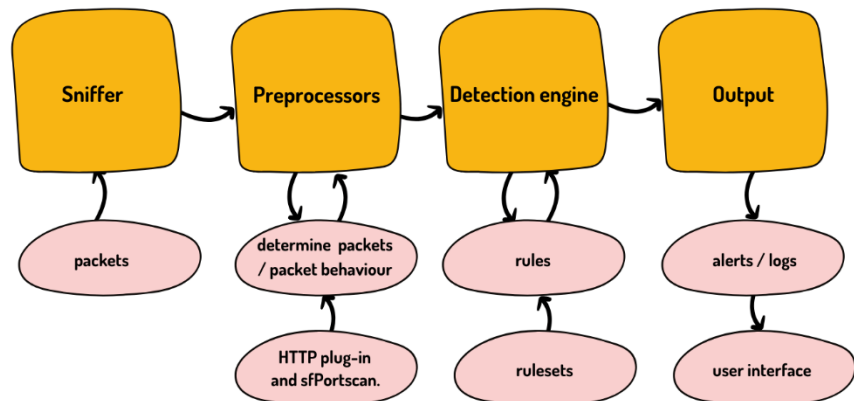
These analyze and modify packets to determine their type or behavior before passing them to the detection engine.

This compares packet data against a predefined ruleset to identify potential threats. Packets that match the rules are forwarded to the output.

Logs and triggers alerts based on detected threats. Logs can be saved in various formats and locations, and user interfaces like Snorby or ACID help manage and view this data.

Snort's architecture

consists of several key components working together to detect and analyze network traffic.



a. Packet Sniffer (thu thập gói tin)

Ở đây là nơi Snort sẽ sử dụng giao diện mạng mà do người dùng cấu hình muốn lắng nghe và bảo vệ để thu thập tất cả các gói tin đi qua giao diện đó. Các gói tin sau khi thu thập được sẽ được định nghĩa lại cấu trúc của nó, tách các thông tin cần thiết từ mỗi tầng trong lưu lượng mạng chẳng hạn như địa chỉ MAC hay loại giao thức của tầng mạng (IPv4, IPv6,...) của tầng data link (liên kết dữ liệu). Ở tầng mạng, các thông tin được định nghĩa bao gồm địa chỉ ip nguồn và đích của gói tin, loại giao thức của tầng vận chuyển (TCP, UDP, ICMP), TTL (time to live) là thời gian tồn tại của gói tin trong mạng. Ở tầng vận chuyển, xác định port nguồn và port đích của gói tin, trạng thái kết nối TCP, dữ liệu payload của gói tin do tầng ứng dụng gửi xuống,...sau khi cấu trúc của các gói tin được định nghĩa thì được tiếp tục chuyển đến module Preprocessors.

b. Preprocessors (tiền xử lý)

Ở đây được ví giống như người chuẩn bị dữ liệu, chuẩn bị gói tin ở định dạng sao cho module tiếp theo là Detection Engine dễ xử lý nhất có thể. Cụ thể ở đây các gói tin sẽ được phân tích và biến đổi để có thể xác định được loại gói tin và hành vi của nó là gì để chuyển tiếp gói tin. Để thực hiện được các tác vụ này thì có các Plugins sẽ thực hiện nhiệm vụ khác nhau mà nó đảm nhận. Các gói tin khi được truyền qua mạng với dữ liệu lớn thì thường sẽ không được toàn vẹn trong quá trình vận chuyển bởi nó đã bị phân mảnh, trong các giao thức phổ biến hiện nay chẳng hạn như Ethernet thì chỉ số MTU (maximum transmission unit) là chỉ số mà kích thước gói tin tối đa được truyền là 1500 bytes, vì thế sẽ có plugins Frag3 sẽ thực hiện nhiệm vụ tái cấu trúc lại các gói tin bị phân mảnh, ghép chúng lại với nhau để dễ phân tích, đồng thời cũng sẽ hạn chế được một số kỹ thuật tấn công né tránh như phân nhỏ gói tin truyền đi nhằm qua mặt các hệ thống phát hiện. Hay có gói tin stream5 để tái dựng luồng TCP vì thông thường trong các cuộc tấn công thì quá nhiều kết nối TCP trong 1 cổng. Ngoài ra còn có các plugins khác như HTTP để phân tích payload và phát hiện các cuộc tấn công sql hay xss. Dữ liệu sau khi đã được chuẩn bị ở định dạng dễ xử lý thì được chuyển tiếp đến Detection Engine.

c. Detection Engine (công cụ phát hiện)

Đây là module quan trọng và là xương sống của Snort. Các gói tin sẽ được so khớp với các quy tắc trong bộ quy tắc đã được định nghĩa sẵn hay do người dùng cấu hình để xác định các mẫu tấn công hoặc hành vi đáng ngờ trong lưu lượng mạng, nếu một gói tin khớp với một quy tắc thì nó sẽ được đánh dấu là có khả năng nguy hiểm và được gửi tiếp ra phần Output xử lý tiếp còn không thì sẽ được loại bỏ.

d. Output

Các gói tin bị phát hiện là nguy hiểm thì sẽ được ghi log lại trong nhật ký log và cảnh báo, nơi hệ thống lưu trữ thông tin chi tiết về sự kiện hoặc cảnh báo xảy ra để phân tích sau. Ngoài ra Snort còn cung cấp giao diện người dùng để có thể dễ dàng quan sát và thao tác như Snorby hay ACID.

1.3. Hoạt động

a. Sniffer mode

Tham số	Chức năng
-v	Hiển thị ở output TCP/IP
-d	Hiển thị dữ liệu gói tin (payload)
-e	Hiển thị dữ liệu tầng liên kết TCP/IP/UDP/ICMP
-X	Hiển thị tất cả dữ liệu gói tin dưới dạng hex
-i	Xác định một giao diện mạng cụ thể mà Snort sẽ lắng nghe hoặc sniff, tham số này cho phép chỉ định giao diện cụ thể mà Snort sẽ theo dõi lưu lượng mạng

Đây là chế độ cơ bản nhất của mọi hệ thống NIDS. Khi ở trong chế độ này, Snort sẽ phát hiện và hiển thị header của các gói tin: TCP, ICMP, UDP, IP, ... ra màn hình.

Các công cụ sniffer mạng như tcpdump, ethereal, và Tethereal có đầy đủ các đặc tính và phân tích gói tin một cách xuất sắc, tuy nhiên, có lúc cần xem lưu lượng mạng trên bộ cảm biến Snort. Trong trường hợp này, sử dụng Snort như là một sniffer là khả thi. Kết quả xuất của chế độ Snort sniffer hơi khác so với các sniffer dòng lệnh. Nó rất dễ để đọc và có thể thấy thích khả năng bắt giữ gói tin nhanh của nó. Một đặc tính hay của chế độ này là việc tóm tắt lưu lượng mạng khi kết thúc việc bắt giữ gói tin. Thỉnh thoảng, nó có thể là một công cụ gỡ rối hữu dụng cho nhà quản trị.

Bật chế độ sniffer cho Snort bằng -v :

snort -v

Trong lúc khởi động, Snort hiển thị chế độ, thư mục ghi log, và các giao diện mà nó đang lắng nghe. Khi việc khởi động hoàn tất, Snort bắt đầu xuất các gói tin ra màn hình. Kết quả xuất này khá cơ bản : nó chỉ hiển thị các header IP, TCP/UDP/ICMP và một số cái khác. Để thoát chế độ sniffer, sử dụng Ctrl-C. Snort thoát bằng cách tạo ra một bản tóm tắt các gói tin được bắt giữ, bao gồm các giao thức, thống kê phân mảnh và tái hợp gói tin. Để xem dữ liệu ứng dụng, sử dụng -d. Tùy chọn này cung cấp các kết quả chi tiết hơn:

snort -vd

Dữ liệu ứng dụng có thể thấy được và chúng ta có thể nhìn thấy các plain text trong gói tin. Trong trường hợp này, văn bản gửi từ một server DNS được thể hiện dưới dạng plain text. Để xem được chi tiết hơn, bao gồm các header lớp liên kết dữ liệu, sử dụng -e. Việc sử dụng cả hai tùy chọn -d và -e sẽ cho hiển thị hầu như tất cả các dữ liệu trong gói tin:

snort -vde

Các chuỗi thập lục phân hiển thị nhiều dữ liệu hơn. Có địa chỉ MAC và địa chỉ IP. Khi thực hiện kiểm tra trên một mạng hoặc bắt giữ dữ liệu bằng Snort, việc bật -vde cung cấp nhiều thông tin nhất.

Để lưu lại trong logfile thay vì xuất ra console, sử dụng snort -dve > temp.log.

b. Packet logger mode (chế độ ghi log gói tin)

Tham số	Chức năng
-l	Ghi lại các gói tin với thời gian chính xác khi chúng được nhận
-K ASCII	Ghi log các gói tin dưới dạng ASCII
-r	Đọc các bản ghi log bởi snort
-n	Chỉ định số lượng gói tin sẽ được xử lý

Đây là chế độ làm việc mà Snort sẽ thực hiện ghi log lại các gói tin đã phát hiện được rồi sau đó lưu trữ lại trong kho lưu trữ log của Snort hoặc một vị trí lưu trữ khác mà người dùng cấu hình chỉ định. Việc ghi log lại sẽ giúp cho chúng ta thực hiện theo dõi và truy vết sau này.

Để chạy snort ở chế độ logger sử dụng tham số -l

Snort -dev -l /home/user/log

Câu lệnh trên cho phép sau khi bắt các gói tin, lưu trữ chúng dưới dạng tập tin log. Ngoài ra có thể lưu trữ các tập tin log dựa trên các địa chỉ IP truy cập. Ví dụ câu lệnh sau sẽ cho phép ta bắt, in ra màn hình và lưu trữ lại các gói tin TCP/IP cũng với tiêu đề ở tầng data-link, dữ liệu của gói tin của tất cả các gói tin đi vào từ địa chỉ của lớp mạng C.

Snort -dev -l /home/user/log -h 192.168.1.0/24

Trường hợp muốn chạy snort ở chế độ logger lưu trữ các tập tin log ở dạng nhị phân có thể sử dụng tùy chọn -b, và sử dụng tùy chọn -r để đọc các tập tin nhị phân được ghi lại.

Snort -l /log -b

Snort -dv -r packet.log

c. NIDS mode (Network Intrusion Detection System Mode)

Tham số	Chức năng
-c	xác định tập tin cấu hình mà Snort sẽ sử dụng. Chỉ định đường dẫn đến tập tin cấu hình sau "-c"

-T	kiểm tra tính hợp lệ của file cấu hình
-N	tắt chế độ ghi log
-D	chạy snort trong chế độ nền
-A	Chế độ cảnh báo với 4 option theo sau: full : cảnh báo tất cả fast : hiển thị thông báo cảnh báo, timestamp, địa chỉ IP nguồn và đích, cùng với số cổng console : cung cấp các cảnh báo nhanh trên màn hình console cmg : cung cấp các thông tin cơ bản về tiêu đề với payload ở dạng hex nonce : tắt cảnh báo

Ở chế độ này, Snort không ghi lại từng gói tin đã bắt được như Sniffer mode. Thay vào đó Snort áp dụng các quy tắc trên tất cả các gói được bắt. Nếu một gói không khớp với bất kỳ quy tắc nào, gói tin đó sẽ bị loại bỏ (drop) và sẽ không thực hiện ghi lại log của gói này.

Để khởi chạy Snort ở chế độ phát hiện xâm nhập mạng không cần bắt tất cả các gói tin.

Snort –dev –1 ./log -h 192.168.1.0/24 -c snort.conf

Tham số -c được sử dụng để chỉ định tập tin cấu hình của Snort. Mặc định các tập tin log sẽ được lưu trữ tại /var/log/snort. Khi chạy ở chế độ NIDS có thể bỏ tùy chọn -v để tăng tốc độ, do không cần thiết phải bắt các gói tin và in ra màn hình.

1.4. Bộ luật của Snort

Hệ thống phát hiện của Snort hoạt động dựa trên các luật (rules) và các luật này lại được dựa trên các dấu hiệu nhận dạng tấn công. Các luật có thể được áp dụng cho tất cả các phần khác nhau của một gói tin dữ liệu.

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

a. Cấu trúc luật trong snort

[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.

Vd: alert tcp 192.168.0.0/22 23 -> any any (content:"confidential"; msg: "Detected confidential")

b. Phần header

Cấu trúc phần header như sau: |Action|Protocol|Address|port|Direction|Address|Port|

Vd: alert tcp 192.168.0.0/22 23 -> any any

- Các action có thể là: pass (bỏ qua gói tin), log (dùng để log gói tin), alert (gửi cảnh báo), activate (kích hoạt thêm các luật khác và kiểm tra điều kiện), dynamic (được gọi bởi luật khác khai báo bằng activate).
- Các protocol bao gồm: ICMP, TCP, UDP,...
- Phần address: có 2 phần là địa chỉ đích và địa chỉ nguồn, nó có thể là 1 ip đơn hoặc là 1 dải mạng, nếu là any thì áp dụng tất cả các địa chỉ trong mạng. Có thể loại trừ ip bằng dấu !.

- Phần port là số port để áp dụng cho các luật, và chỉ có 2 port là TCP và UDP.
- Ký hiệu sẽ chỉ ta đâu là nguồn và đâu là đích là -> hoặc <- hoặc <> .

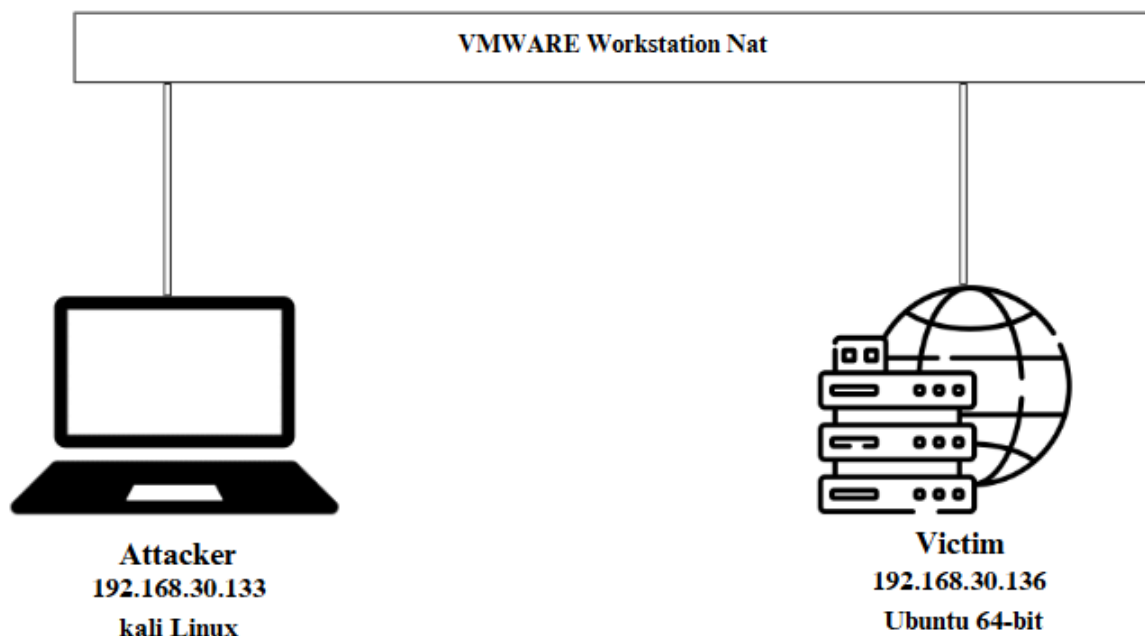
c. Phần option

Phần option được đặt trong dấu ngoặc đơn và các option sẽ cách nhau bởi dấu “;”. Các option sẽ bao gồm phần từ khóa và tham số, cấu trúc sẽ là từ khóa:tham số. Các từ khóa đó có thể là ack, flag, content, dsize,... với mỗi từ khóa thì mang 1 ý nghĩa khác nhau.

II. Triển khai

2.1. Mô hình

2.1.1. Cơ bản

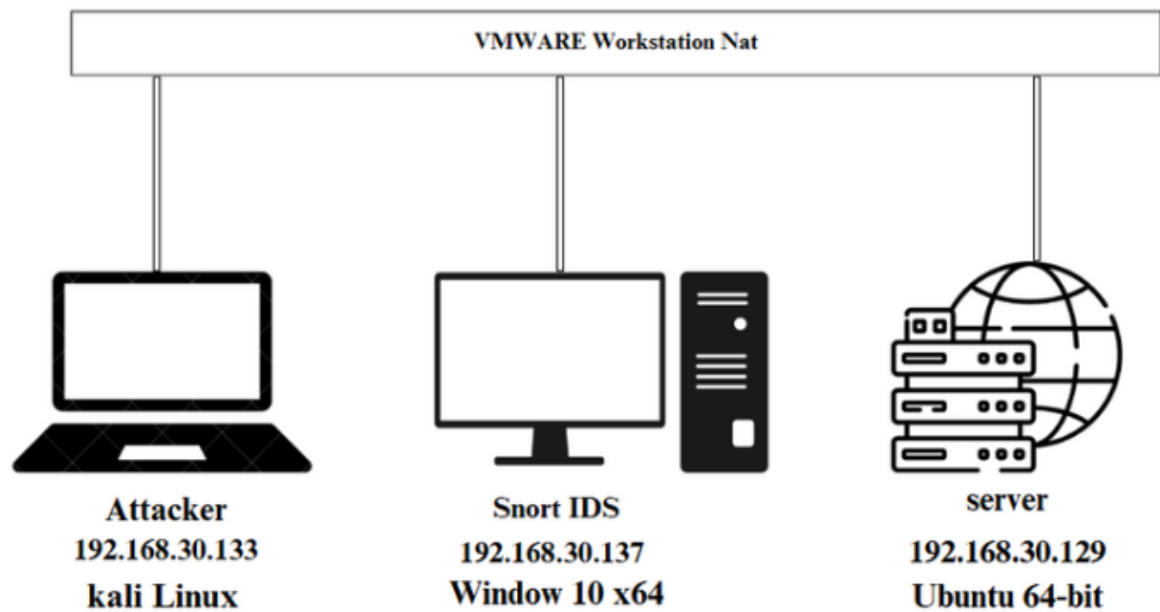


Máy	Hệ điều hành	ipv4	NIC
Attacker	Kali linux	192.168.30.133	NAT
victim	Ubuntu 64-bit	192.168.30.136	NAT

Mục tiêu của mô hình:

- Cài đặt Snort trên máy victim và cấu hình Snort để Snort lắng nghe trên card mạng 192.168.30.137.
- Tiến hành chạy Snort ở 2 chế độ là sniffer mode và packet logger mode.
- Test thử cách Snort lắng nghe các hoạt động dựa trên tấn công mạng giả lập cơ bản từ attacker: port scan, TCP SYN flood, Ping of Death.
- Sử dụng rule mặc định.

2.1.2. Nâng cao



Máy	Hệ điều hành	ipv4	NIC
Attacker	Kali linux	192.168.30.133	NAT
Snort IDS	Window 10 x64	192.168.30.137	NAT
Server	Ubuntu 64-bit	192.168.30.129	NAT

Mục tiêu của mô hình:

- Cài đặt cấu hình để phát hiện xâm nhập đến máy server.
- Cho snort hoạt động với chế độ HIDS mode.
- Viết một số rule cho các mẫu tấn công đã biết trước để dễ dàng phân biệt loại tấn công.

2.2. Cài đặt

2.2.1. Ubuntu

Để cài đặt đầu tiên ta sử dụng lệnh
sudo apt update

Sau đó ta sử dụng lệnh
sudo apt install -y snort

```

Processing triggers for libc-bin (2.35-0ubuntu3.8) ...
xuanhuy@xuanhuy-virtual-machine:~$ snort --v

,,-      -*> Snort! <*-
o"  )~   Version 2.9.15.1 GRE (Build 15125)
'    '   By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
          Copyright (C) 2014-2019 Cisco and/or its affiliates. All rights reserved.

          Copyright (C) 1998-2013 Sourcefire, Inc., et al.
          Using libpcap version 1.10.1 (with TPACKET_V3)
          Using PCRE version: 8.39 2016-06-14
          Using ZLIB version: 1.2.11

xuanhuy@xuanhuy-virtual-machine:~$ █

```

2.2.2. Kali linux

Cài đặt Snort trên kali linux thì có phần đơn giản hơn ở ubuntu bởi vì kali đã có sẵn kho phần mềm cho snort, nên việc cài đặt sẽ đơn giản và không cần phải cài đặt thêm nhiều. Nhưng muốn cài được snort ta cần phải chỉnh sửa lại 1 chút source list. Đầu tiên cần phải sao lưu source list trên kali khi chúng ta muốn chỉnh sửa file đó để đề phòng có chuyện xảy ra.

mv /etc/apt/sources.list /etc/apt/sources.list.bak

Chỉnh nội dung source list:

sudo nano /etc/apt/sources.list

Thêm nội dung sau vào:

**deb [arch=arm64] http://ports.ubuntu.com/ubuntu-ports focal main restricted
universe multiverse
**

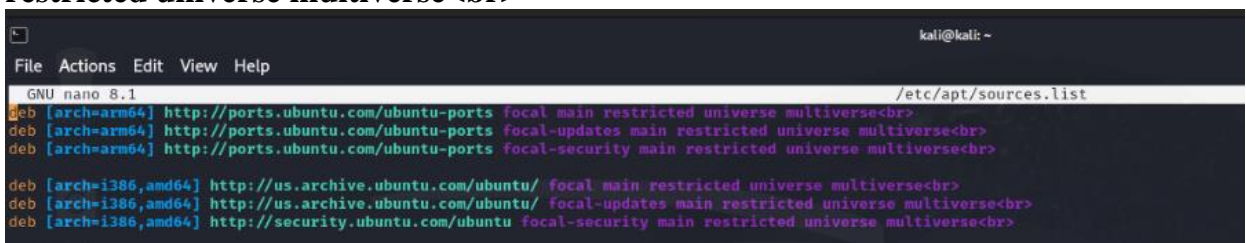
**deb [arch=arm64] http://ports.ubuntu.com/ubuntu-ports focal-updates main
restricted universe multiverse
**

**deb [arch=arm64] http://ports.ubuntu.com/ubuntu-ports focal-security main
restricted universe multiverse
**

**deb [arch=i386,amd64] http://us.archive.ubuntu.com/ubuntu/ focal main restricted
universe multiverse
**

**deb [arch=i386,amd64] http://us.archive.ubuntu.com/ubuntu/ focal-updates main
restricted universe multiverse
**

**deb [arch=i386,amd64] http://security.ubuntu.com/ubuntu focal-security main
restricted universe multiverse
**



```

kali@kali: ~
File Actions Edit View Help
GNU nano 8.1 /etc/apt/sources.list
deb [arch=arm64] http://ports.ubuntu.com/ubuntu-ports focal main restricted universe multiverse<br>
deb [arch=arm64] http://ports.ubuntu.com/ubuntu-ports focal-updates main restricted universe multiverse<br>
deb [arch=arm64] http://ports.ubuntu.com/ubuntu-ports focal-security main restricted universe multiverse<br>
deb [arch=i386,amd64] http://us.archive.ubuntu.com/ubuntu/ focal main restricted universe multiverse<br>
deb [arch=i386,amd64] http://us.archive.ubuntu.com/ubuntu/ focal-updates main restricted universe multiverse<br>
deb [arch=i386,amd64] http://security.ubuntu.com/ubuntu focal-security main restricted universe multiverse<br>

```

Chạy các khóa

**sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys
3B4FE6ACC0B21F32**

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys  
871920D1991BC93C
```

Cập nhật:

```
sudo apt update
```

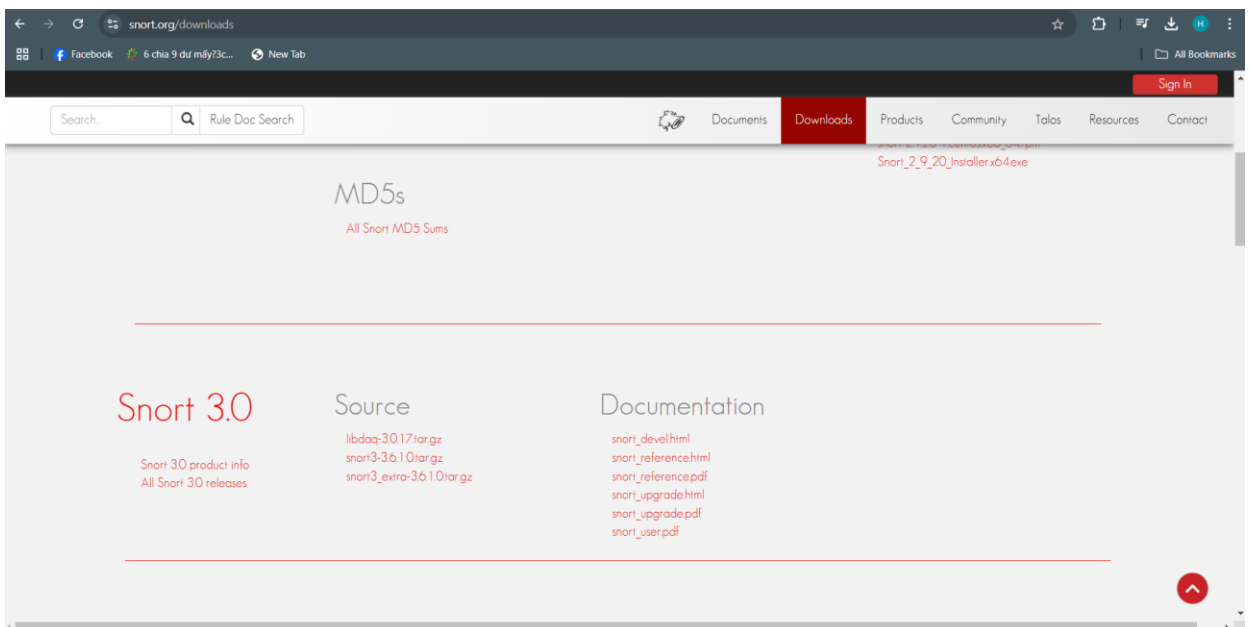
Cài đặt:

```
sudo apt install snort
```

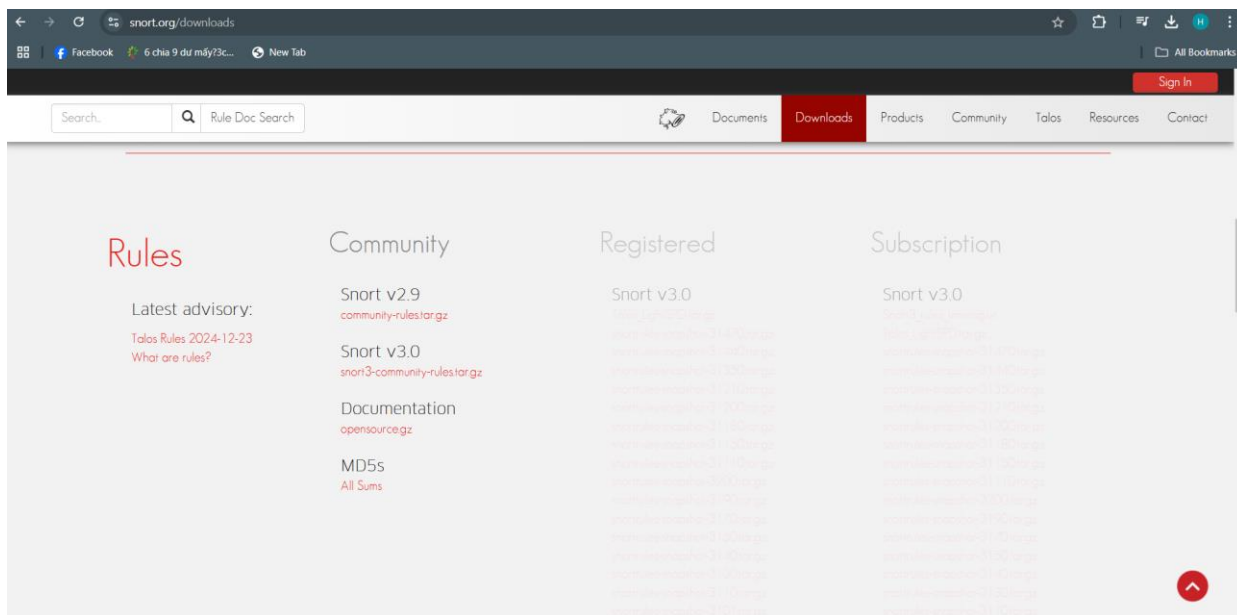
```
(kali@kali)-[~]  
$ snort -V  
  
-*> Snort! <*-  
Version 2.9.7.0 GRE (Build 149)  
By Martin Roesch & The Snort Team: http://www.snort.org/contact#team  
Copyright (C) 2014 Cisco and/or its affiliates. All rights reserved.  
Copyright (C) 1998-2013 Sourcefire, Inc., et al.  
Using libpcap version 1.10.5 (with TPACKET_V3)  
Using PCRE version: 8.39 2016-06-14  
Using ZLIB version: 1.3.1  
  
(kali@kali)-[~]  
$
```

2.2.3. Window

Để cài đặt được Snort ở trên window chúng ta có thể lên trang chính chủ của Snort là <https://www.snort.org> để tiến hành cài đặt phiên bản mới nhất của Snort về và giải nén.

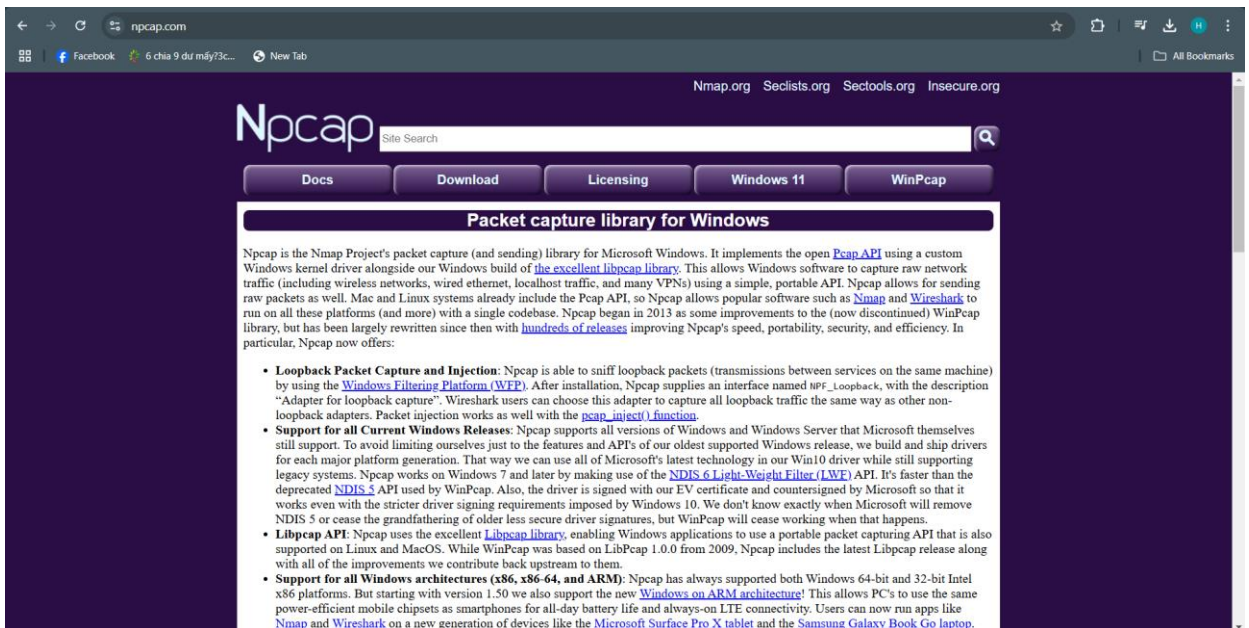


Cần phải tải thêm tệp rule đính kèm để sử dụng được các rule mặc định của Snort.

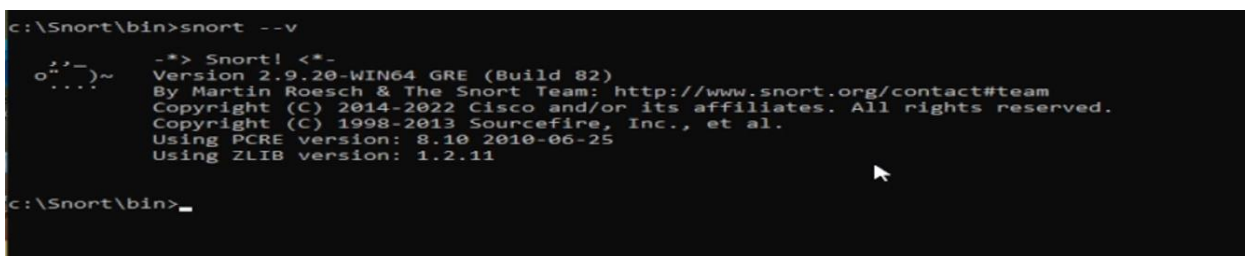


Tải thêm các thư viện kèm theo của Snort để tiến hành bắt được gói tin.



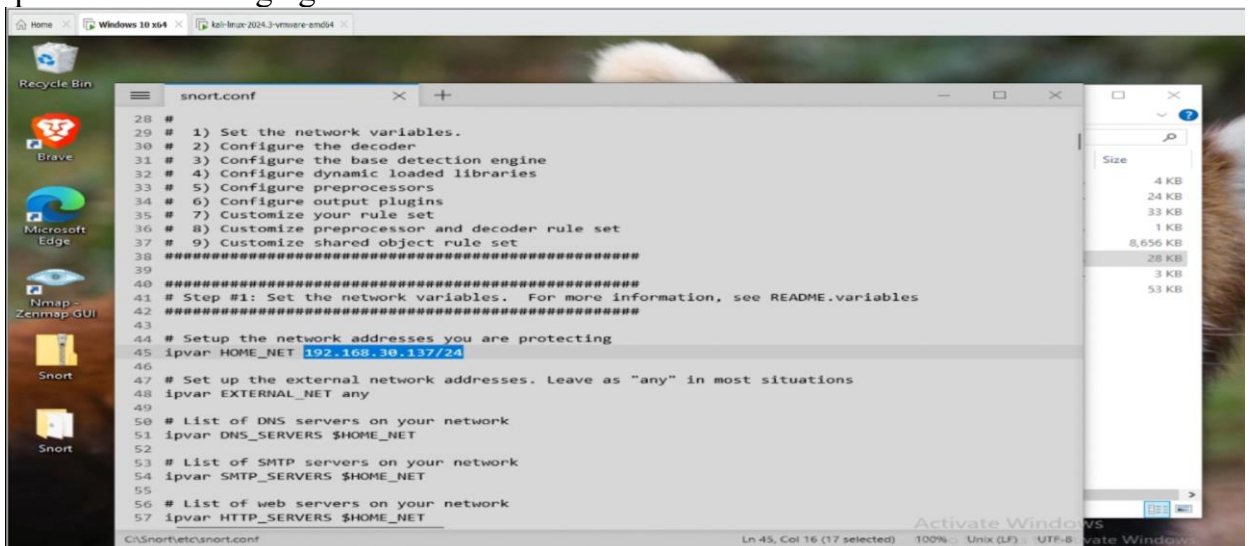


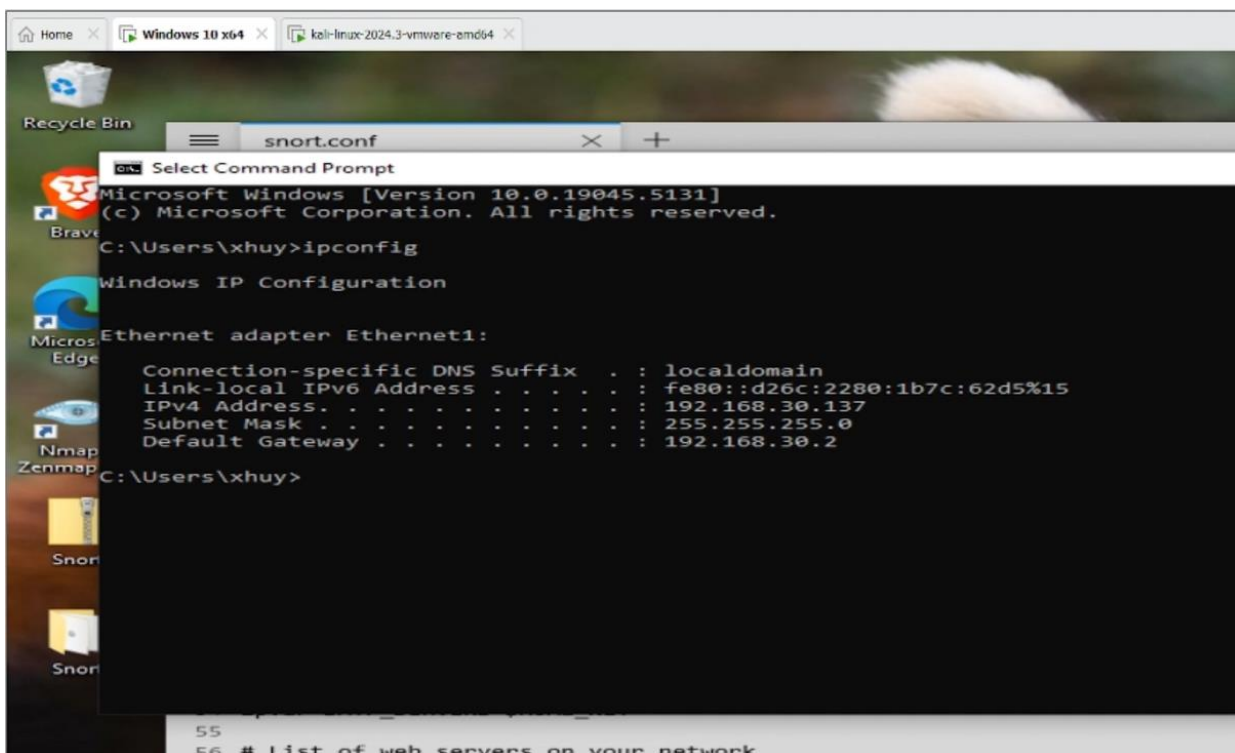
Kiểm tra Snort đã hoạt động sau khi cài đặt thành công.



2.3. Cấu hình

Sau khi cài đặt thành công Snort ta tiến hành vào file cấu hình Snort để cấu hình địa chỉ ip mà Snort sẽ lắng nghe.

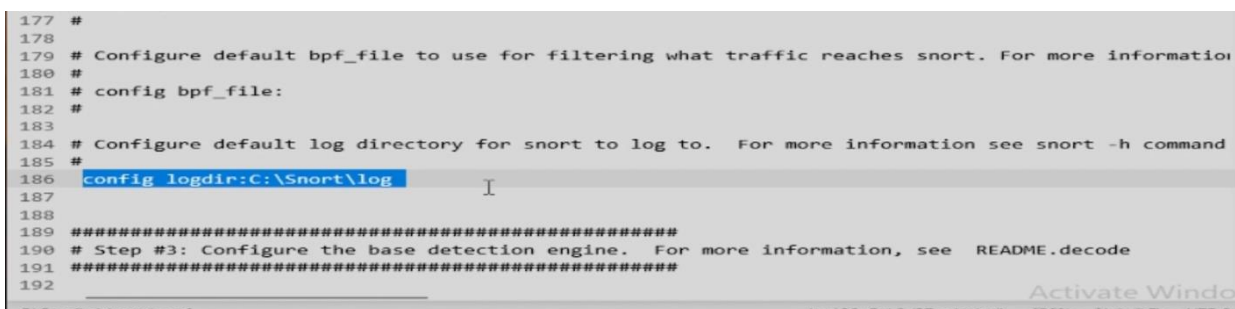




Tiếp theo ta cần chỉnh lại thư mục đường dẫn chứa các tệp rule của Snort.



Và cuối cùng ta cần chỉnh lại đường dẫn tới thư mục mà ta sẽ ghi lại log trong quá trình mà Snort hoạt động. Việc này khá quan trọng trong bước cấu hình.



III. Kết quả và kết luận

3.1. Kết quả

3.1.1. Cơ bản

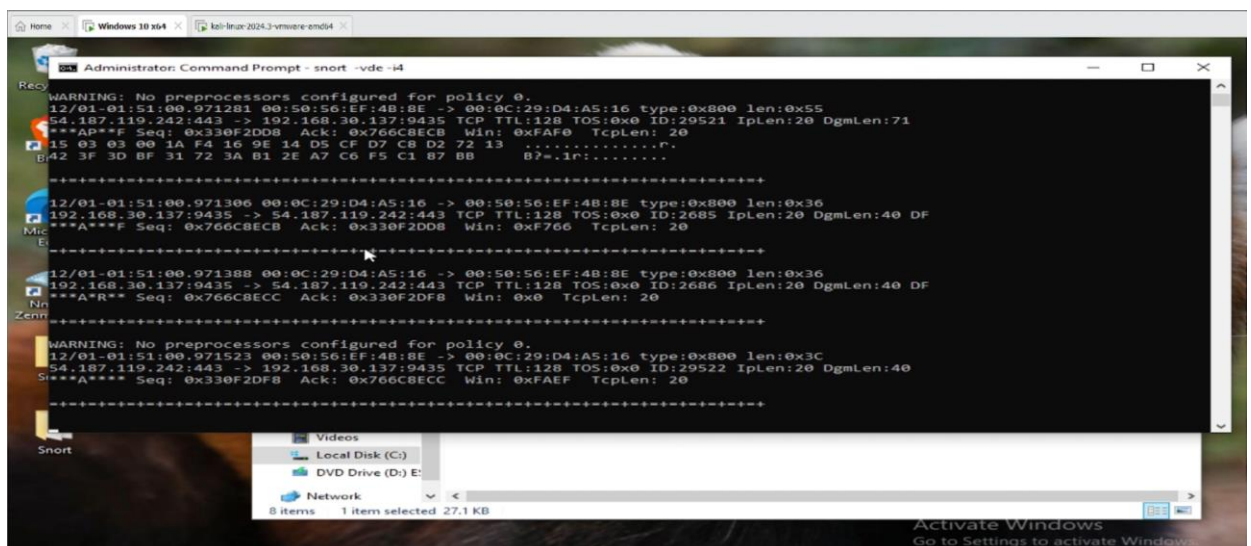
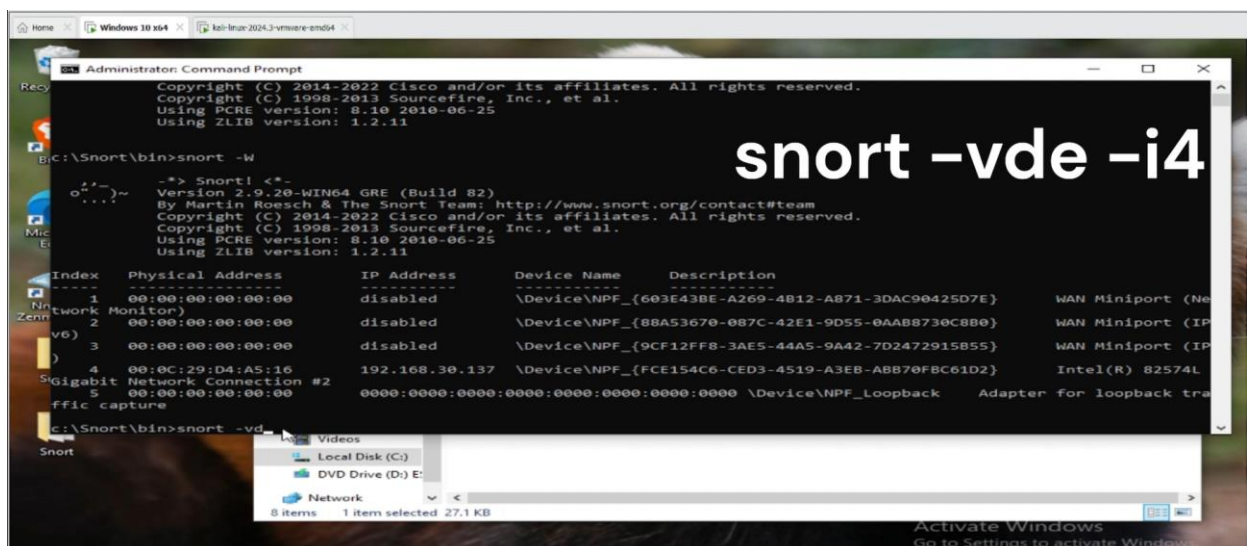
```
c:\Snort\bin>snort -W

-*> Snort! <*-
o'~
...~
Version 2.9.20-WIN64 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 PCRE version: 8.10 2010-06-25
Using ZLIB version: 1.2.11

Index  Physical Address      IP Address      Device Name      Description
-----
1  00:00:00:00:00:00      disabled      \Device\NPF_{603E43BE-A269-4B12-AB71-3DAC90425D7E}      WAN Miniport (Ne
2  00:00:00:00:00:00      disabled      \Device\NPF_{88A53670-087C-42E1-9D55-0AAB8730C8B0}      WAN Miniport (IP
3  00:00:00:00:00:00      disabled      \Device\NPF_{9CF12FF8-3AE5-44A5-9A42-7D2472915B55}      WAN Miniport (IP
4  00:0C:29:D4:A5:16      192.168.30.137 \Device\NPF_{FCE154C6-CED3-4519-A3EB-AB870FBC61D2}      Intel(R) 82574L
Gigabit Network Connection #2
5  00:00:00:00:00:00      0000:0000:0000:0000:0000:0000 \Device\NPF_Loopback      Adapter for loopback tra
ff capture

c:\Snort\bin>
```

Bước đầu tiên kiểm tra xem card mạng mà chúng ta muốn Snort lắng nghe.



Tiến hành chạy Snort ở chế độ sniffer, lắng nghe và hiện tất cả những gói tin mà nó bắt được ra màn hình console.

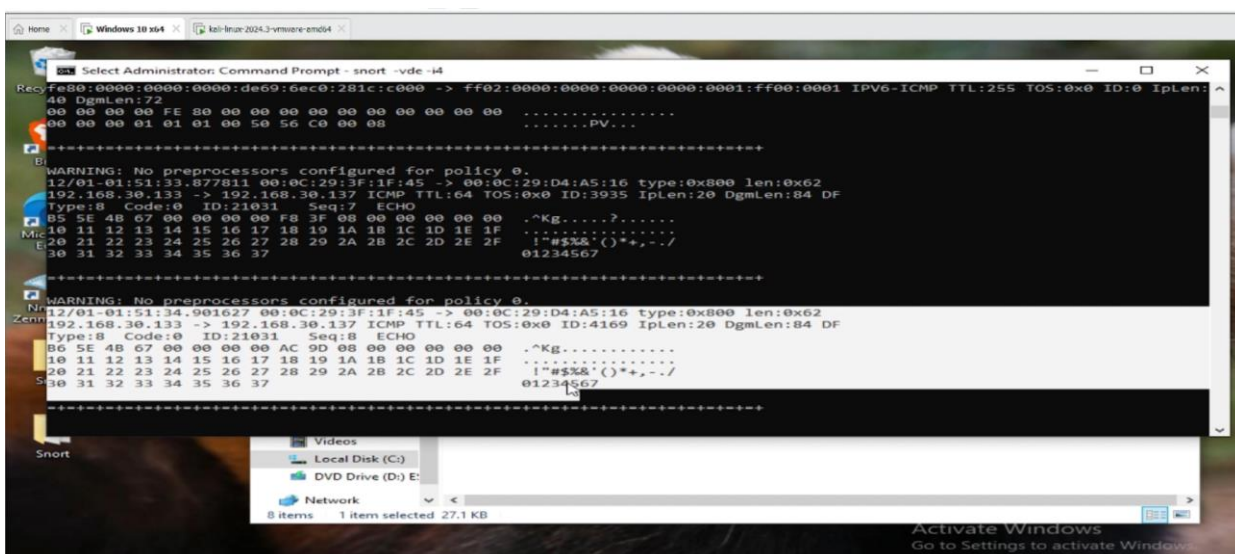
```
File Actions Edit View Help
zsh: corrupt history file /home/kali/.zsh_history
(kali@kali)-[~]
$ ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
    ether 02:42:6f:e9:8f:2e txqueuelen 0 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.30.133 netmask 255.255.255.0 broadcast 192.168.30.255
    inet6 fe80::7f15:3d6f:f5f2:e196 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:3f:1f:45 txqueuelen 1000 (Ethernet)
    RX packets 1796263 bytes 107968509 (102.9 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 40906404 bytes 2478961572 (2.3 GiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 769978 bytes 30800936 (29.3 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 769978 bytes 30800936 (29.3 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

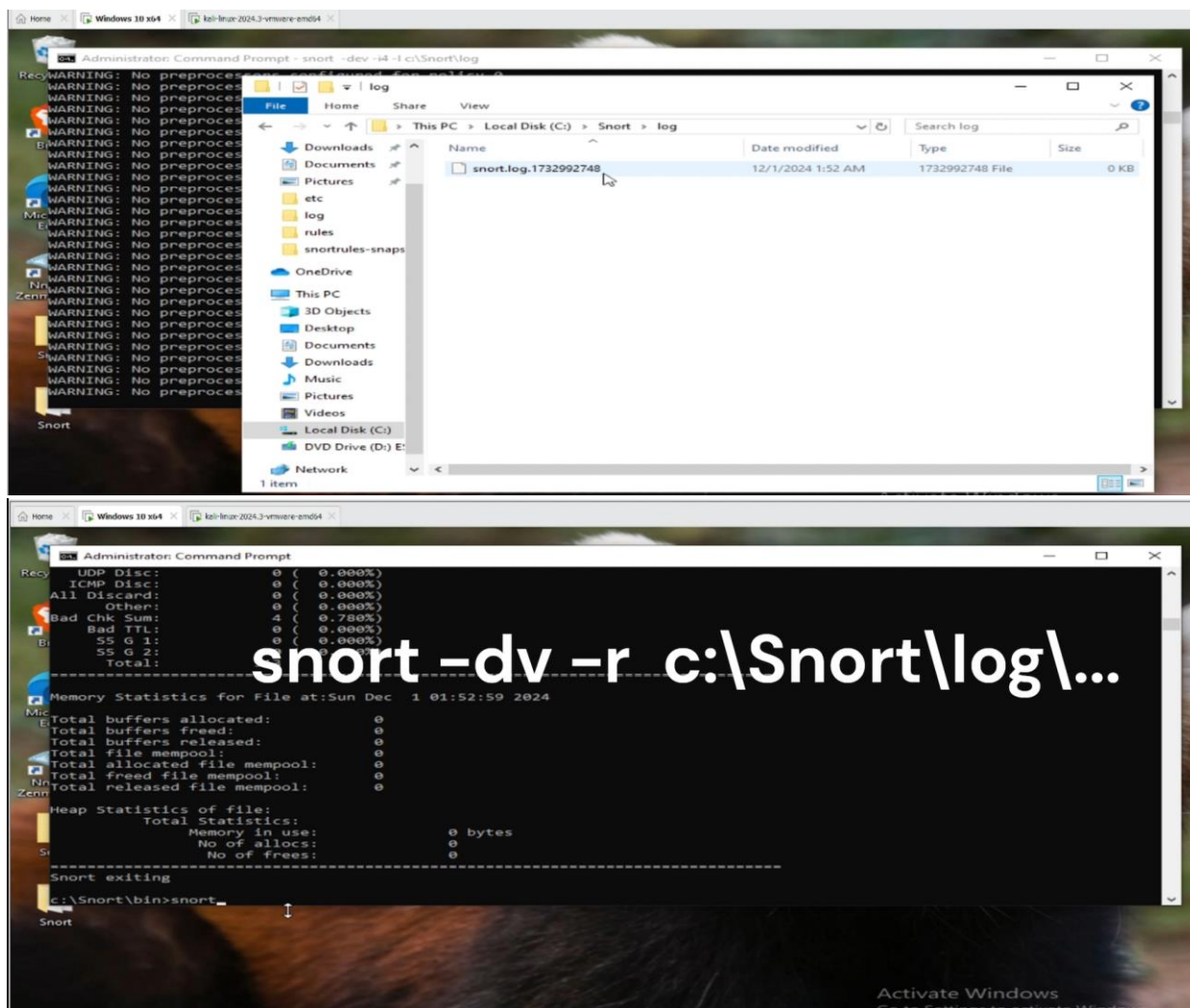
(kali@kali)-[~]
$ ping 192.168.30.137
PING 192.168.30.137 (192.168.30.137) 56(84) bytes of data.
```

Thử ping từ máy attacker đến máy nạn nhân và xem kết quả mà chúng ta nhận được Snort.

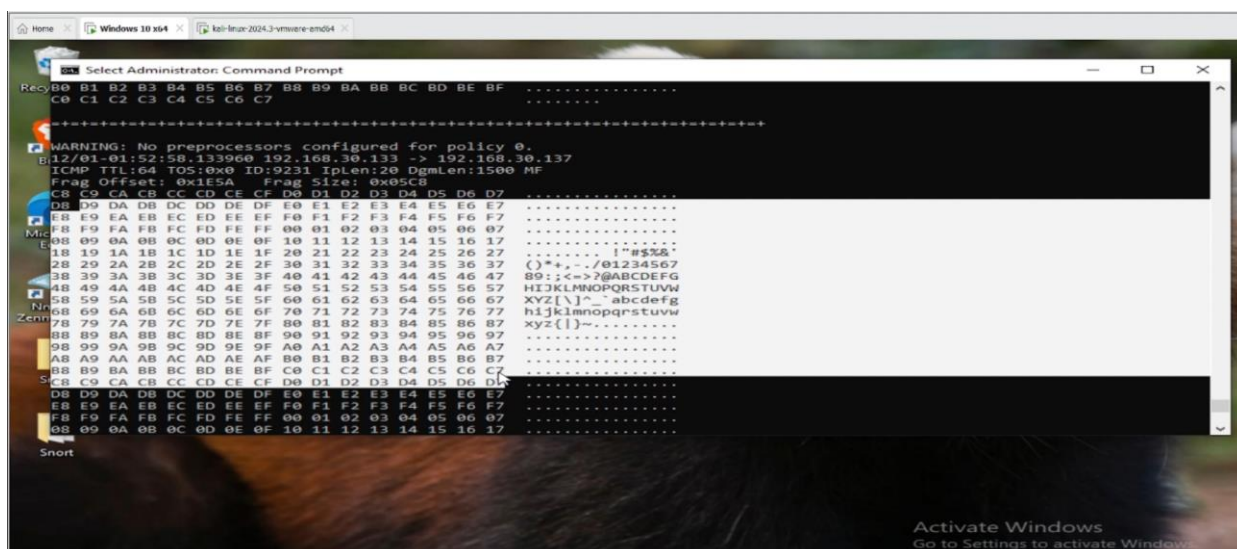


Chúng ta đã nhận được các gói tin ICMP có địa chỉ ip nguồn là máy attacker đến máy nạn nhân.

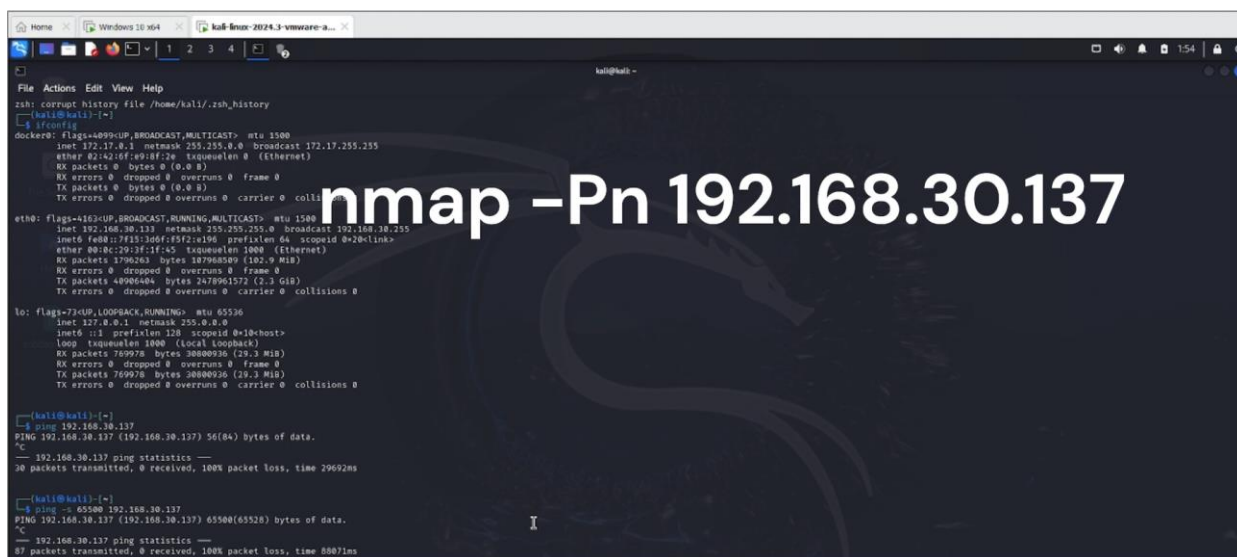
Khi kiểm tra bên máy victim thì thấy dòng thông báo NO PREPROCESSORS CONFIGURED FOR POLICY 0 thì do chưa cấu hình cho module preprocessors.



Tiến hành vào thư mục mà từ đầu ta đã chỉ đường dẫn để Snort ghi log vào và tiến hành đọc nó ra màn hình console.



Ở đây ta đã đọc được những gói tin với kích thước payload rất lớn là 1500 bytes, gói tin đã bị phân mảnh trong quá trình truyền qua mạng với giao thức là ethernet có kích thước MTU (Maximum transmission unit) tối đa là 1500 bytes, và cộng thêm việc module preprocessors chưa được chúng ta cấu hình nên không thể tái cấu trúc nó lại được.



```
zsh: corrupt history file /home/kali/.zsh_history
kali@kali:~$ ifconfig
docker0: flags=4096<UP,BROADCAST,MULTICAST> mtu 1500
    inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
    ether 02:42:1f:4e:99:2e txqueuelen 0 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    TX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

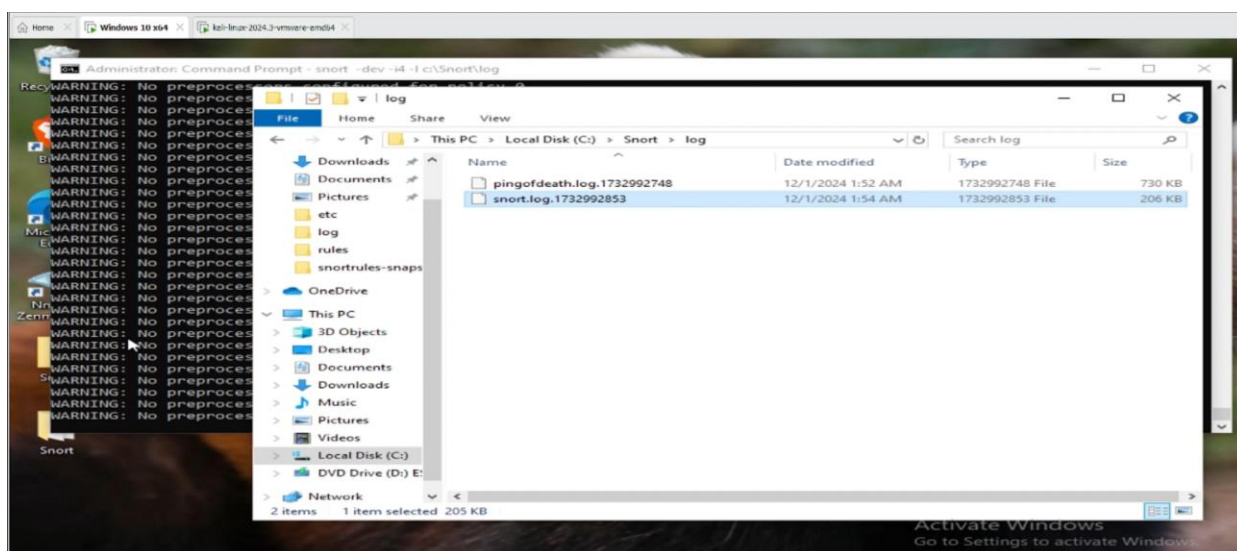
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.30.137 netmask 255.255.255.0 broadcast 192.168.30.255
    inet6 fe80::f23:3d6f:f572:e196 prefixlen 64 scopeid 0<20<link>
    ether 08:00:27:3f:c1:f5 txqueuelen 1000 (Ethernet)
    RX packets 1796263 bytes 187968589 (182.9 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 499684 bytes 247961572 (23.3 GiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

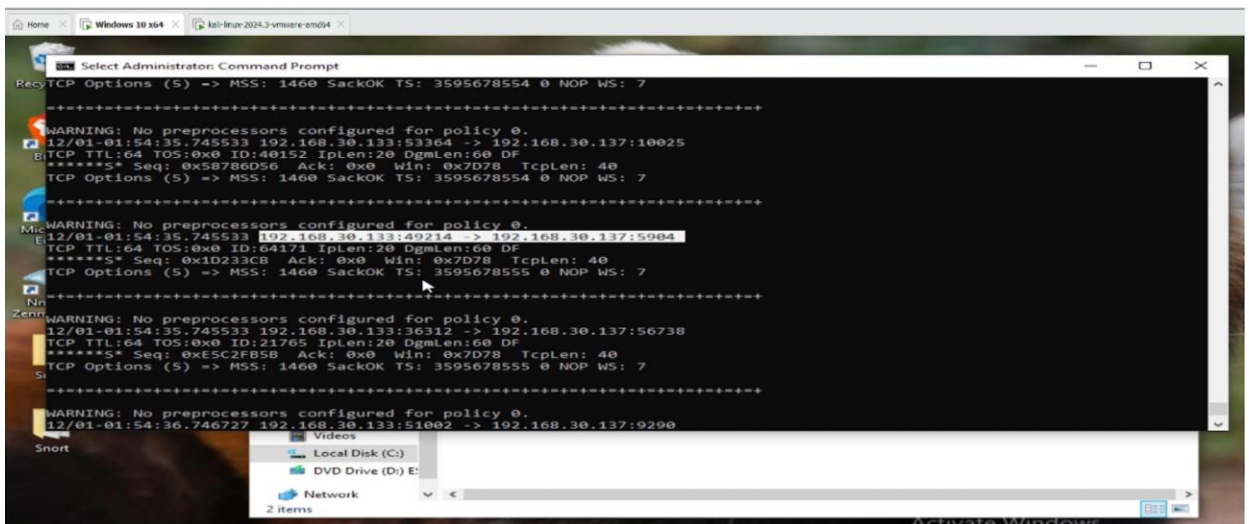
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0<1<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 769978 bytes 388000936 (38.3 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 769978 bytes 388000936 (38.3 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

kali@kali:~$ ping 192.168.30.137
PING 192.168.30.137 (192.168.30.137) 56(84) bytes of data.
^C
 192.168.30.137 ping statistics:
 30 packets transmitted, 0 received, 100% packet loss, time 29692ms

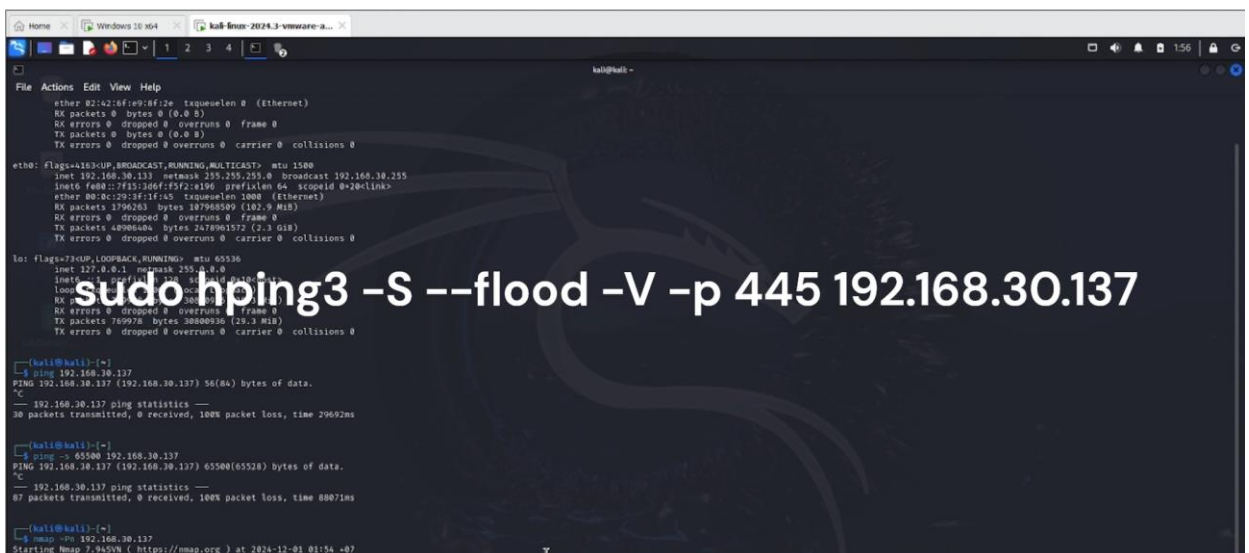
kali@kali:~$ ping -c 65500 192.168.30.137
PING 192.168.30.137 (192.168.30.137) 65500(65528) bytes of data.
^C
 192.168.30.137 ping statistics:
 87 packets transmitted, 0 received, 100% packet loss, time 88871ms
```

Tiếp theo ta sẽ thử quét cổng của máy victim với lệnh nmap và -Pn để quét bất kể máy nạn nhân có hoạt động hay là có phản hồi lại gói tin cho chúng ta hay không.

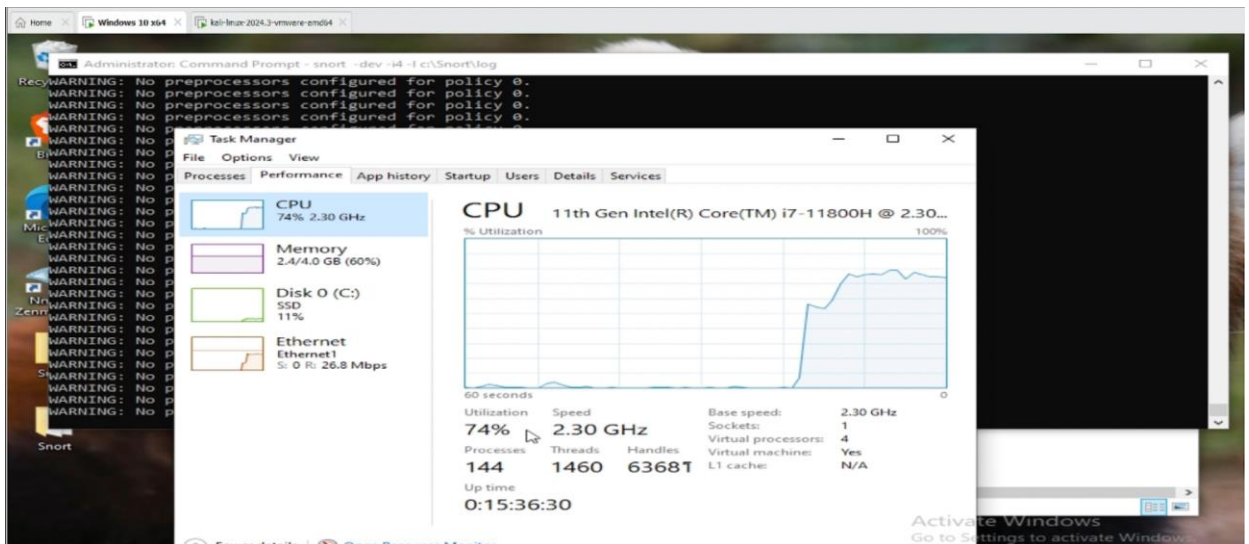




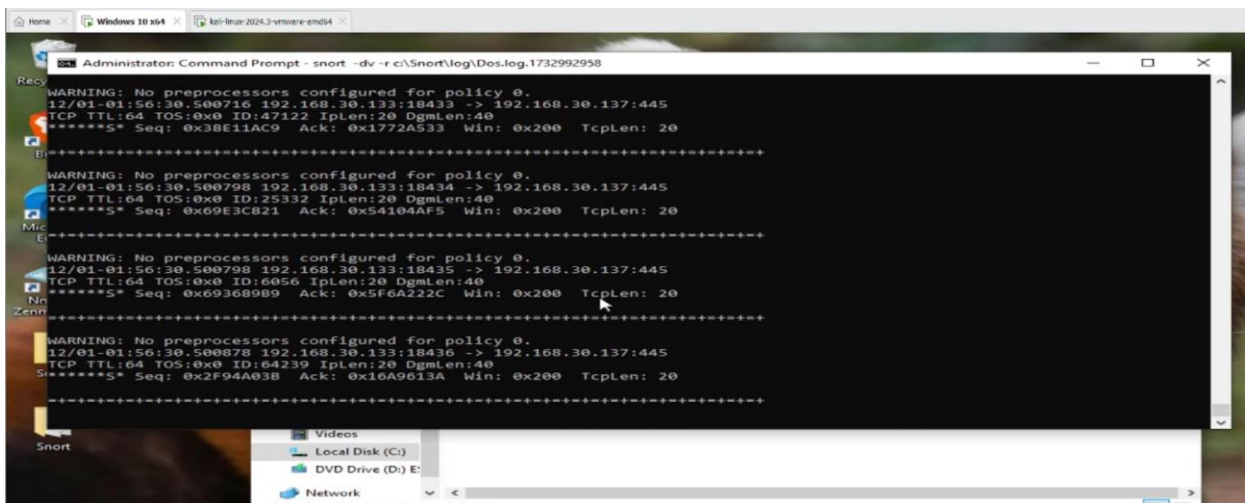
Tiến hành kiểm tra thì đã thấy log mới, đọc thì sẽ thấy rất nhiều gói tin TCP đến từ cùng 1 địa chỉ và đến 1 địa chỉ với rất nhiều cổng, đây là dấu hiệu dễ nhận thấy nhất của việc đang bị quét cổng.



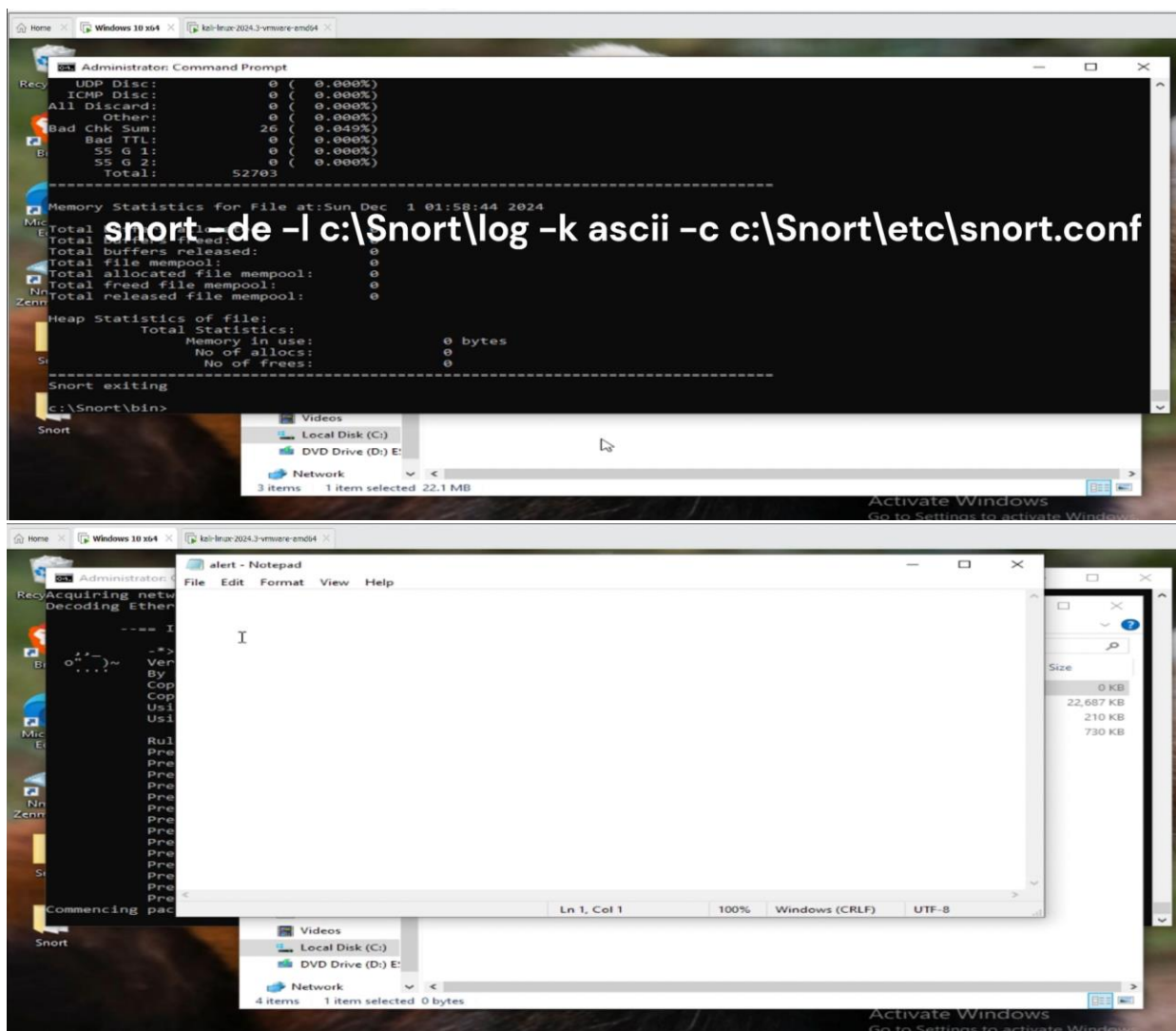
Và cuối cùng sau khi quét cổng nhưng có vẻ victim đã có tường lửa chặn và chúng ta không thể tìm được cổng nào đang mở của máy victim nên ta sẽ tiến hành tấn công ngập lụt với công cụ hping3 vào cổng 445, cổng SMB (server message block) của window để chia sẻ tệp tin, các gói tin này sẽ được gửi đi với hình thức là không hoàn tất quá trình bắt tay 3 bước để hoàn tất kết nối TCP, gửi giả lập các gói tin TCP để làm cho máy victim phải đợi quá trình kết nối hoàn tất trong khoảng thời gian dài và làm đầy hàng đợi của máy victim.



Kiểm tra CPU bên máy victim thì thấy hoạt động của nó tăng cao bất thường và có dấu hiệu bị quá tải.

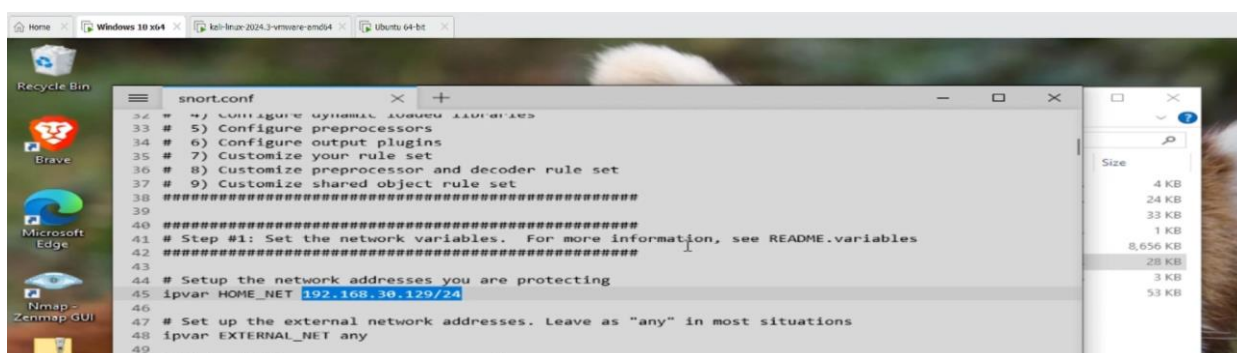


Tiến hành đọc log thì thấy rất nhiều gói tin TCP được gửi đi trong thời gian ngắn, đây là kiểu tấn công ngập lụt làm quá tải CPU của máy nạn nhân và có thể làm công bị tấn công tê liệt.



Tiến hành chạy thử Snort ở chế độ IDS và đọc file IDS thì thấy chưa có gì xuất hiện có thể là do chưa cấu hình các rule cho Snort nên vẫn chưa tìm thấy được gói tin nào khớp với rule để hiện ra thông báo.

3.1.2. Nâng cao



Ở đây ta sẽ chỉnh lại địa chỉ ip của máy server mà Snort sẽ làm nhiệm vụ IDS.

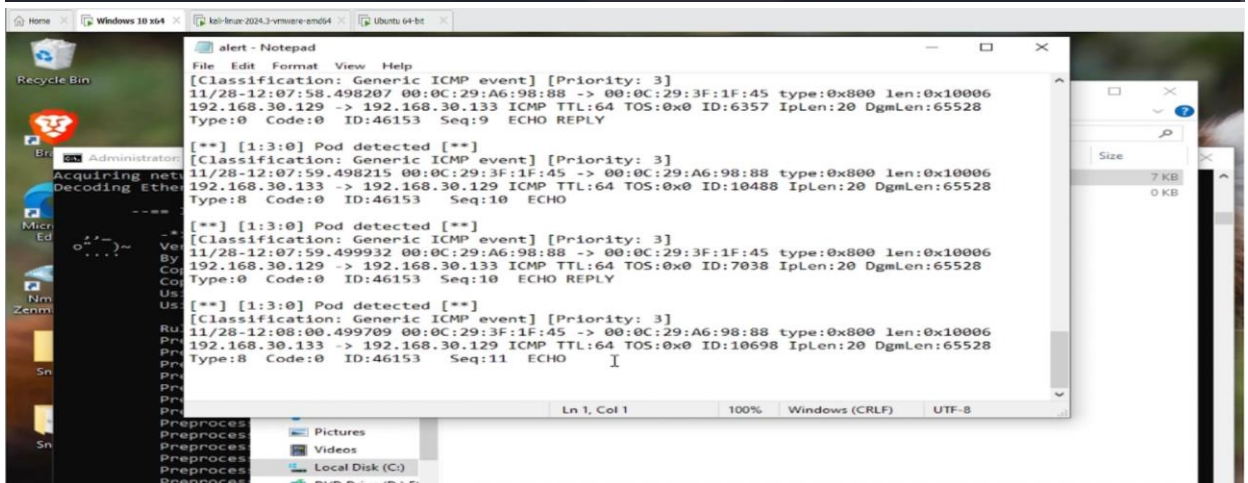
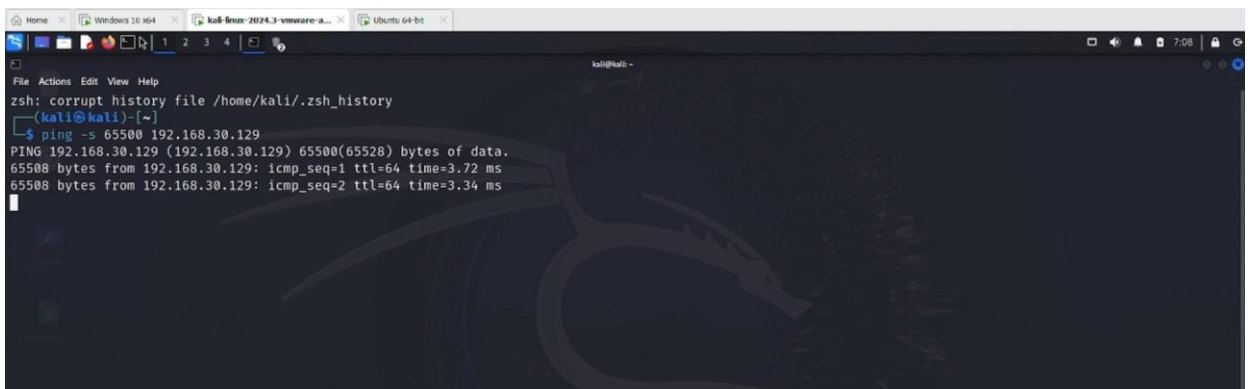
```
# to the VRT Certified Rules License Agreement (v2.0).
#
#-----
# ICMP RULES
#-----
alert icmp any any -> $HOME_NET any (msg:"Pod
detected";dsize:>1000;sid:3;classtype:icmp-event;priority:3;) I
```

Ln 21, Col 108

100%

Unix (LF)

Ta truy cập vào tệp rule của Snort mà ta đã tải về và thêm một một số rule cho nó, ở đây thêm rule để phát hiện và cảnh báo những gói tin có kích thước lớn hơn 1000 bytes.



Ở đây ta sẽ kiểm tra file IDS và đã xuất hiện một số cảnh báo về kích thước gói tin rất lớn là 65528 bytes với 28 bytes header. Ở đây ta đã nhận được kích thước gói tin đúng với kích thước của nó vì chúng ta đang chạy Snort ở chế độ IDS và module preprocessors đã được cấu hình nên các gói tin sẽ được tái cấu trúc lại cho chúng ta dễ xử lý.

```
#
#-----
# SCAN RULES
#-----
alert tcp any any -> $HOME_NET any (msg:"Port-scan
detected";sid:2;classtype:network-scan;priority:2;flags:S)
```

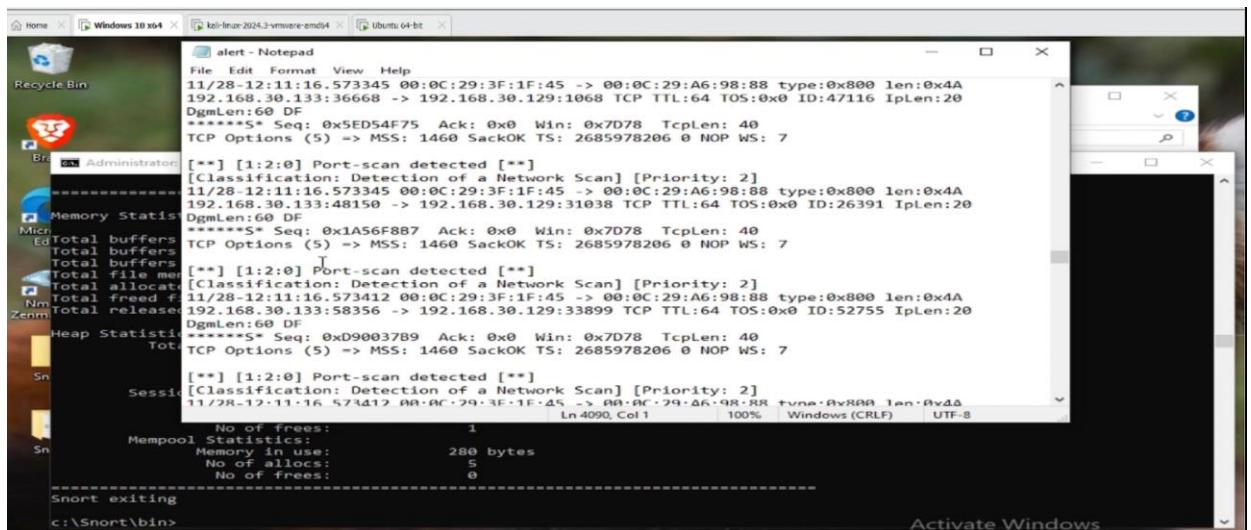
Ln 21, Col 62

100%

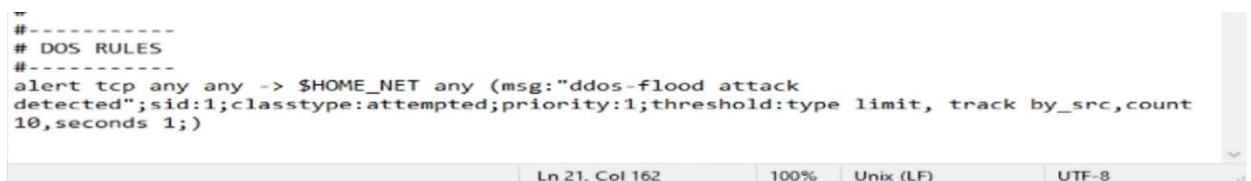
Unix (LF)

UTF-8

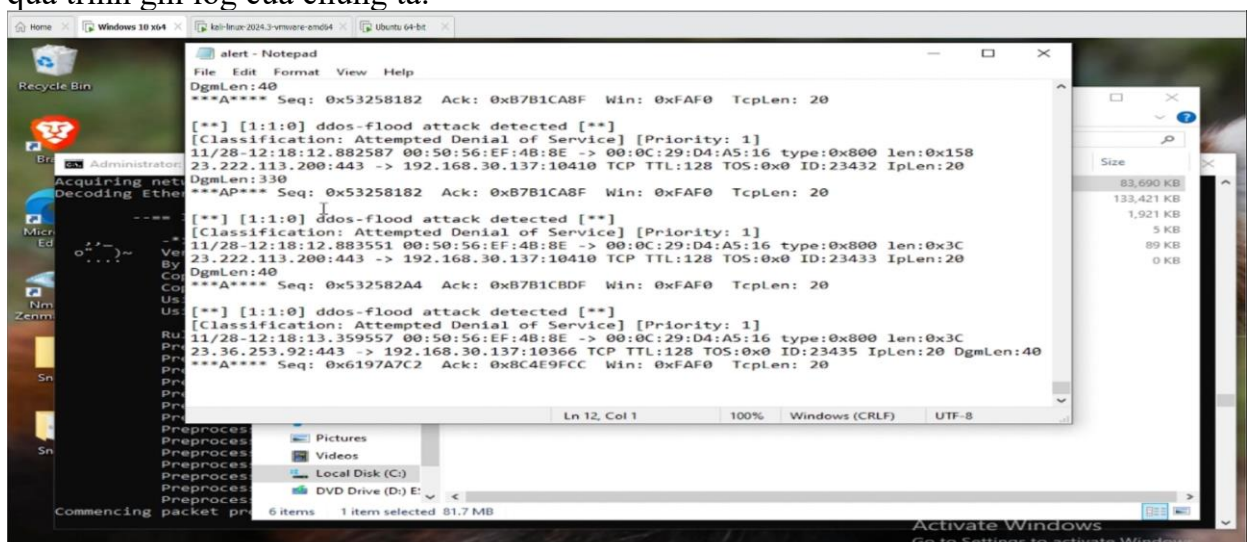
Ta sẽ thêm vào rule với dòng cảnh báo là port-scan để phát hiện trường hợp quét cổng từ máy attacker với cờ S, gửi các gói tin yêu cầu kết nối TCP để kiểm tra xem cổng có mở không.



Sau khi tiến hành quét cổng thì file IDS đã xuất hiện những cảnh báo mới về port-scan đến địa chỉ ip của máy attacker.



Cuối cùng là thiết lập rule cho kiểu tấn công ngập lụt. Với loại tấn công này thì cần thêm điều kiện là các gói tin đối với cùng 1 địa chỉ ip thì cần giới hạn thông báo của nó lại là 10 gói tin trong 1s được gửi đến sẽ hiện 1 lần thông báo để tránh tình trạng sẽ bị lag trong quá trình ghi log của chúng ta.



Ở đây đã nhận được các cảnh báo về cuộc tấn công ngập lụt.

3.2. Kết luận

Snort là một hệ thống phát hiện xâm nhập (IDS) mạnh mẽ và linh hoạt, được sử dụng rộng rãi trong lĩnh vực an ninh mạng nhờ khả năng phát hiện các mối đe dọa dựa trên phân tích gói tin thời gian thực. Với ưu điểm nổi bật là mã nguồn mở, Snort cho phép người dùng

tùy chỉnh và xây dựng các quy tắc phù hợp với nhu cầu cụ thể. Ngoài ra, khả năng phát hiện các kiểu tấn công phổ biến như DDoS, SQLi, XSS, và CSRF giúp Snort trở thành một công cụ không thể thiếu trong việc bảo vệ mạng và hệ thống. Tuy nhiên, Snort cũng có nhược điểm, bao gồm hiệu suất có thể giảm khi xử lý lượng lớn dữ liệu trên các hệ thống quy mô lớn và yêu cầu kỹ năng chuyên môn cao để cấu hình và quản lý hiệu quả. Tổng kết lại, Snort là một giải pháp lý tưởng cho các tổ chức muốn tăng cường bảo mật với chi phí thấp, đồng thời đòi hỏi sự đầu tư vào nguồn lực kỹ thuật để khai thác tối đa tiềm năng của nó.

Hãy thường xuyên cập nhật quy tắc và tối ưu cấu hình Snort để đảm bảo phát hiện chính xác các mối đe dọa mới nhất, đồng thời giám sát hiệu suất để phù hợp với quy mô hệ thống của chúng ta.

IV. Trả lời câu hỏi

Snort xử lý các cuộc tấn công từ chối dịch vụ (DoS) bằng cách dựa vào bộ quy tắc được định nghĩa trước của nó hay còn gọi là ruleset, từ đó nhận ra được dấu hiệu của cuộc tấn công này như lưu lượng lớn bất thường từ một nguồn duy nhất, các yêu cầu lặp đi lặp lại hoặc việc khai thác các lỗ hổng giao thức. Sau đó sẽ kích hoạt cảnh báo cho quản trị viên hoặc phối hợp với các hệ thống khác để thực hiện các biện pháp phòng ngừa như chặn địa chỉ IP nguồn thông qua tường lửa. Nhưng rule cần phải được cập nhật và tối ưu thường xuyên, quan trọng hơn là phải giám sát hệ thống liên tục tránh tình trạng quá tải khi xử lý.

Hiện tại thì Snort gặp hạn chế trong việc xử lý các cuộc tấn công sử dụng mã hóa (encryption) vì nó không thể phân tích nội dung đã mã hóa. Tuy nhiên, Snort có thể phát hiện các hành vi bất thường liên quan đến lưu lượng mã hóa như lưu lượng tăng đột biến, sử dụng giao thức mã hóa không phổ biến, hoặc các kết nối mã hóa đến các địa chỉ IP đáng ngờ, giúp quản trị viên nhận biết và điều tra sâu hơn.

Để giảm thiểu false positive, Snort cần được tối ưu hóa bằng cách chỉ sử dụng các quy tắc phù hợp với môi trường, tùy chỉnh ngưỡng phát hiện và định nghĩa ngoại lệ cho các luồng dữ liệu hợp lệ. Đối với false negative, việc cập nhật thường xuyên bộ rule và chữ ký và phân tích kỹ lưỡng lưu lượng mạng giúp đảm bảo không bỏ sót các mối đe dọa nguy hiểm khác. Sự cân bằng giữa độ chính xác và khả năng phát hiện là yếu tố quan trọng trong việc giảm cả hai loại sai lệch này.

Snort được sử dụng vì khả năng phát hiện xâm nhập thời gian thực, giúp phát hiện và cảnh báo về các mối đe dọa dựa trên quy tắc tùy chỉnh. So với Wireshark, Snort tự động phát hiện tấn công thay vì phân tích thủ công. So với tường lửa, như đã nói lúc đầu thì tường lửa được ví như hàng rào bảo vệ ngôi nhà, ổ khóa cửa. Tuy nhiên kẻ trộm có thể dùng bất cứ hành vi nào để có thể vượt qua những thứ trên và đột nhập nhà. IDS được ví như chuông báo động, camera quan sát giúp chúng ta biết được các phương pháp mà chúng có thể đột nhập. Tương tự IDS giúp chúng ta phát hiện và cảnh báo ra được các dấu hiệu bất thường, giám sát và phân tích các hoạt động ra vào của hệ thống để có thể ngăn chặn kịp thời.

Để tích hợp Snort với Wazuh, cấu hình Snort để gửi log đầu ra ở định dạng Unified2 hoặc syslog. Sau đó, cài đặt Filebeat hoặc ossec-agent trên máy chạy Snort để chuyển log đến Wazuh Manager. Trong Wazuh, bật các quy tắc phát hiện liên quan đến Snort để phân tích log và hiển thị cảnh báo trên giao diện SIEM của Wazuh.

V. Tài liệu tham khảo

<https://www.snort.org/>

https://www.researchgate.net/figure/The-Snort-architecture_fig1_325851146

<https://github.com/hocchudong/ghichep-IDS-IPS-SIEM.git>

<https://users.soict.hust.edu.vn/hoangph/textbook/ch05-3.html>