

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



BÁO CÁO BÀI THỰC HÀNH
THỰC TẬP CƠ SỞ
Bài 6: Cài đặt cấu hình HIDS/NIDS

Họ và tên: Nguyễn Huy Quang

Mã sinh viên: B20DCAT144

Giảng Viên: Nguyễn Hoa Cường

Hà Nội – 2023

MỤC LỤC

I.	Tìm hiểu lý thuyết.....	1
1.	Khái quát về hệ thống phát hiện tấn công, xâm nhập	1
2.	Phân loại các hệ thống phát hiện tấn công, xâm nhập.....	1
a.	NIDS	2
b.	HIDS	2
3.	Các kỹ thuật phát hiện xâm nhập	3
4.	Một số hệ thống phát hiện tấn công	4
a.	Snort.....	4
b.	Suricata.....	5
c.	OSSEC	6
d.	Wazuh	8
II.	Nội dung thực hành.....	9
1.	Chuẩn bị môi trường, công cụ.....	9
2.	Các bước thực hiện.....	9
a.	Cài đặt và cấu hình Snort	9
b.	Tạo các luật Snort để phát hiện 3 dạng rà quét, tấn công hệ thống	12
III.	Tổng kết.....	15
IV.	Tài liệu tham khảo	16

I. Tìm hiểu lý thuyết

1. Khái quát về các hệ thống phát hiện tấn công, xâm nhập

- Hệ thống phát hiện xâm nhập (Intrusion Detection System – IDS) là hệ thống phần cứng hoặc phần mềm có chức năng giám sát lưu lượng mạng, tự động theo dõi các sự kiện xảy ra trên hệ thống máy tính, phân tích để phát hiện ra các vấn đề liên quan đến an ninh, bảo mật và đưa ra các cảnh báo cho nhà quản trị.
- IDS bao gồm các thành phần chính:
 - + Thành phần thu thập thông tin gói tin
 - + Thành phần phát hiện gói tin
 - + Thành phần xử lý (phản hồi)
- Chức năng của IDS:
 - + Bảo vệ tính toàn vẹn của dữ liệu, đảm bảo sự nhất quán của dữ liệu trong hệ thống. Các biện pháp đưa ra ngăn chặn được việc thay đổi bất hợp pháp hoặc phá hoại dữ liệu.
 - + Bảo vệ tính bí mật, giữ cho thông tin không bị lộ ra ngoài.
 - + Bảo vệ tính khả dụng, tức là hệ thống luôn sẵn sàng thực hiện yêu cầu truy nhập thông tin của người dùng hợp pháp.
 - + Bảo vệ tính riêng tư, tức là đảm bảo cho người sử dụng khai thác tài nguyên hệ thống theo đúng chức năng, nhiệm vụ đã được phân cấp, ngăn chặn sự truy nhập thông tin bất hợp pháp.
 - + Cung cấp thông tin về sự xâm nhập, đưa ra những chính sách đối phó, khắc phục và sửa chữa.
 - + Ngăn chặn sự gia tăng của các cuộc tấn công.
 - + Bổ sung những điểm yếu mà hệ thống khác không làm được.
 - + Đánh giá chất lượng của việc thiết kế hệ thống.
- Kiến trúc của hệ thống phát hiện tấn công, xâm nhập gồm 3 thành phần chính:
 - + Thành phần thu thập gói tin (information collection).
 - + Thành phần phân tích gói tin và phát hiện xâm nhập (detection). Thành phần này là quan trọng nhất và ở thành phần này bộ cảm biến (sensor) đóng vai trò quyết định.
 - + Thành phần phản hồi (response) nếu hệ thống đó được phát hiện tấn công là của một hacker.

2. Phân loại các hệ thống phát hiện tấn công, xâm nhập

Hệ thống IDS được chia làm 2 loại cơ bản:

- Network-based IDS (NIDS): sử dụng dữ liệu trên toàn bộ lưu thông mạng cùng dữ liệu kiểm tra từ một hoặc một vài máy trạm để phát hiện xâm nhập.

- Host-based IDS (HIDS): sử dụng dữ liệu kiểm tra từ một máy trạm đơn để phát hiện xâm nhập.

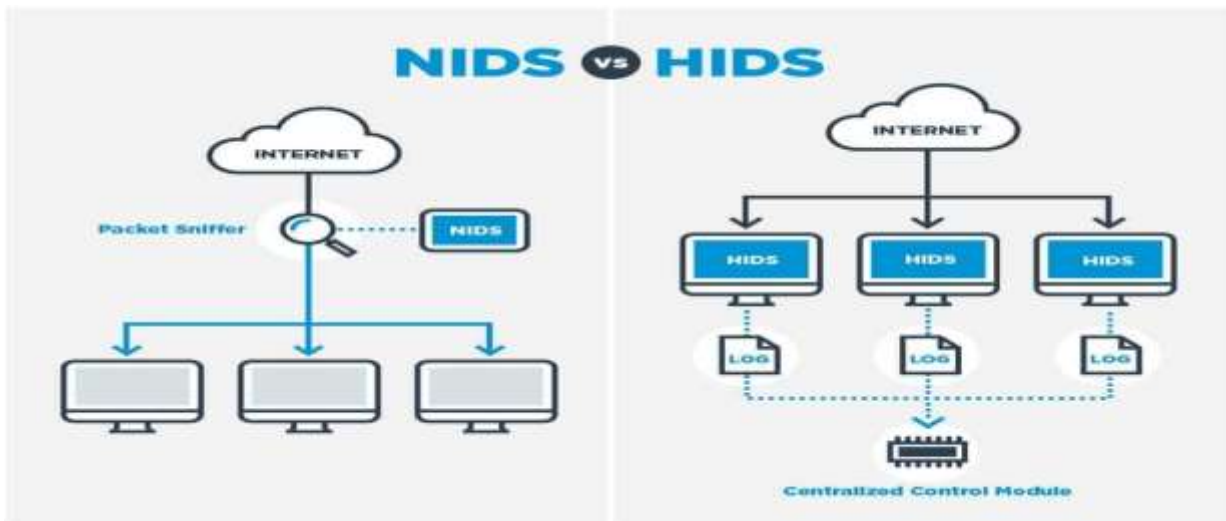
a. NIDS

- Hệ thống IDS dựa trên mạng sẽ kiểm tra các giao tiếp trên mạng với thời gian thực (real time).
- Nó kiểm tra các giao tiếp, quét header của các gói tin và có thể kiểm tra nội dung của các gói đó để phát hiện ra các đoạn mã nguy hiểm hay các dạng tấn công khác nhau.
- Một NIDS hoạt động tin cậy trong việc kiểm tra, phát hiện các dạng tấn công trên mạng như dựa vào băng thông của tấn công DoS.
- Ưu điểm:
 - + Quản lý được cả một network segment (gồm nhiều host).
 - + Trong suốt với người sử dụng lẫn kẻ tấn công.
 - + Cài đặt và bảo trì đơn giản, không ảnh hưởng tới mạng.
 - + Có khả năng xác định lỗi ở tầng network.
 - + Độc lập với hệ điều hành.
- Nhược điểm:
 - + Có thể xảy ra trường hợp báo động giả.
 - + Không thể phân tích dữ liệu đã được mã hóa (SSL,SSH,IPSec..)
 - + NIDS đòi hỏi phải cập nhật các signature mới nhất để thực sự an toàn.
 - + Có độ trễ giữa thời điểm bị tấn công với thời điểm phát báo động. Khi báo động được phát hiện, hệ thống có thể đã bị tổn hại.
 - + Không cho biết việc tấn công có thành công hay không.
 - + Hạn chế lớn nhất là giới hạn băng thông. Những bộ dò mạng phải nhận tất cả các lưu lượng mạng, sắp xếp lại những lưu lượng đó và phân tích chúng. Khi tốc độ mạng tăng lên thì khả năng của đầu dò cũng phải tăng theo.

b. HIDS

- Bằng cách cài đặt một phần mềm trên máy chủ, IDS dựa trên máy chủ quan sát tất cả những hoạt động về hệ thống và các file log, lưu lượng mạng thu thập.
- Hệ thống dựa trên máy chủ cũng theo dõi hệ điều hành, những cuộc gọi hệ thống, lịch sử và những thông điệp báo lỗi trên hệ thống máy chủ.
- HIDS giám sát hoạt động trên một máy tính và thường được cài trên các host quan trọng.
- Nhiệm vụ của HIDS là theo dõi các thay đổi trên hệ thống gồm:
 - + Các tiến trình
 - + Các entry

- + Mức độ sử dụng CPU
- + Tình trạng RAM
- + Tính toàn vẹn của hệ thống
- + Các thông số này khi vượt qua một ngưỡng nhất định hoặc có những thay đổi khả nghi sẽ gây ra báo động
- Ưu điểm:
 - + Có khả năng xác định user liên quan tới event.
 - + HIDS có khả năng phát hiện tấn công diễn ra trên một máy, NIDS thì không.
 - + Có thể phân tích các dữ liệu mã hóa.
 - + Cung cấp các thông tin về host trong lúc cuộc tấn công diễn ra trên host này.
- Nhược điểm:
 - + Thông tin từ HIDS là không đáng tin cậy ngay khi sự tấn công vào host này thành công.
 - + Khi hệ điều hành bị sập do tấn công, đồng thời HIDS cũng sập.
 - + HIDS phải được thiết lập trên từng host cần giám sát.
 - + HIDS không có khả năng phát hiện các cuộc dò quét mạng (Nmap, Netcat ...)
 - + HIDS cần tài nguyên trên host để hoạt động.
 - + HIDS có thể không hiệu quả khi bị tấn công DOS.



Hình 1: So sánh NIDS với HIDS

3. Các kỹ thuật phát hiện xâm nhập

Các kỹ thuật phát hiện xâm nhập gồm 2 loại chính là: dựa trên chữ ký và dựa trên bất thường.

- Dựa trên chữ ký:
 - + Là hình thức lâu đời nhất của phát hiện xâm nhập.
 - + Bằng cách duyệt qua dữ liệu để tìm ra các kết quả khớp với các mẫu đã biết. Ví dụ một địa chỉ IP hay một chuỗi văn bản, hoặc số lượng byte null xuất hiện sau một chuỗi xác định khi sử dụng một giao thức nào đó.
 - + Các mẫu được chia thành các mẫu nhỏ độc lập với nền tảng hoạt động.
 - + Mẫu được mô tả bằng ngôn ngữ cụ thể trong nền tảng của một cơ chế phát hiện xâm nhập, chúng trở thành chữ ký.
 - + Có hai cơ chế phát hiện dựa trên chữ ký phổ biến là Snort và Suricata.
- Dựa trên bất thường:
 - + Là một hình thức mới của phát hiện xâm nhập, phổ biến với công cụ Bro.
 - + Dựa vào quan sát sự cố mạng và nhận biết lưu lượng bất thường thông qua các chẩn đoán và thống kê.
 - + Có khả năng nhận ra các mẫu tấn công khác biệt với hành vi mạng thông thường.
 - + Đây là một cơ chế phát hiện rất tốt nhưng khó thực hiện.

4. Một số hệ thống phát hiện tấn công

a. Snort

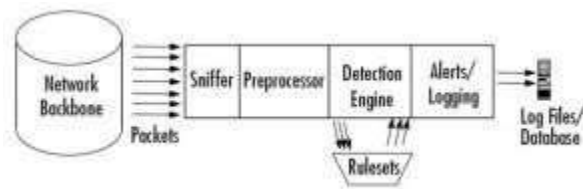
Snort là phần mềm IDS được phát triển dưới dạng mã nguồn mở. Snort ban đầu được xây dựng trên nền Unix sau đó phát triển sang nền tảng khác. Với kiến trúc kiểu modulo, người dùng có thể tự tăng cường tính năng cho hệ thống Snort của mình.

Kiến trúc của Snort: Bao gồm các thành phần sau

- Module giải mã gói tin: Snort chỉ sử dụng thư viện pcap để bắt mọi gói tin trên mạng lưu thông qua hệ thống. Một gói tin sau khi được giải mã sẽ được đưa tiếp vào module tiền xử lý.
- Module tiền xử lý: Chuẩn bị gói dữ liệu đưa vào module phát hiện và gồm 3 nhiệm vụ chính:
 - + Kết hợp lại các gói tin: Khi một dữ liệu lớn được gửi đi, thông tin sẽ không đóng gói toàn bộ vào một gói tin mà thực hiện phân mảnh, chia thành nhiều gói tin rồi mới gửi đi. Khi Snort nhận được các gói tin này, nó phải thực hiện kết nối lại để có gói tin ban đầu.
 - + Giải mã và chuẩn hóa giao thức (decode/normalize): công việc phát hiện xâm nhập dựa trên dấu hiệu nhận dạng nhiều khi thất bại khi kiểm tra

các giao thức có dữ liệu có thể được biểu diễn dưới nhiều dạng khác nhau.

- + Phát hiện các xâm nhập bất thường (nonrule/anormal): các plugin dạng này thường để xử lý với các xâm nhập không thể hoặc rất khó phát hiện bằng các luật thông thường.
- Module phát hiện: Đây là phần quan trọng nhất của Snort
 - + Chịu trách nhiệm phát hiện các dấu hiệu xâm nhập.
 - + Module phát hiện sử dụng các luật được định nghĩa trước để so sánh với dữ liệu thu thập được, từ đó xác định xem có xâm nhập xảy ra hay không.
- Module log và cảnh báo: Tùy thuộc vào module phát hiện có nhận dạng được xâm nhập hay không mà gói tin có thể bị ghi log hay đưa ra cảnh báo. Các file log là các file dữ liệu có thể ghi dưới nhiều định dạng khác nhau như tcpdump.
- Module kết xuất thông tin: Module này thực hiện các thao tác khác nhau tùy thuộc vào việc cấu hình lưu kết quả xuất ra như thế nào như ghi log file, ghi syslog, ghi cảnh báo vào cơ sở dữ liệu, tạo file log XML....



Hình 1: Kiến trúc Snort

Tính năng của Snort: Bên cạnh hoạt động như một ứng dụng bắt gói tin thông thường, Snort còn được cấu hình để chạy như một NIDS.

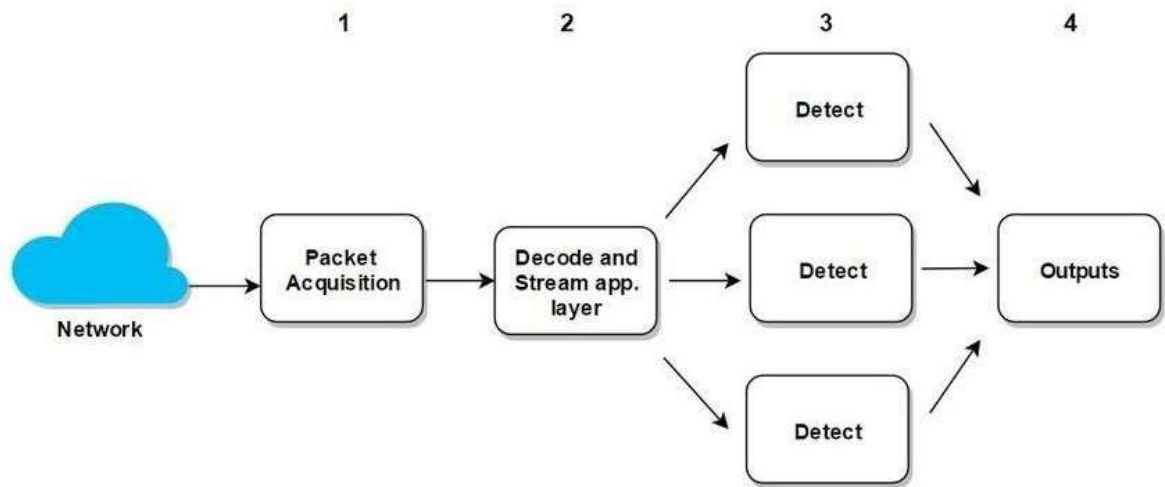
b. Suricata

Suricata là giải pháp IDS/IPS mã nguồn mở hiệu quả cho các hệ thống mạng chưa được đầu tư các giải pháp IDS/IPS thương mại. Nó được xây dựng từ các thành phần khác nhau và khả năng hoạt động của nó tùy thuộc vào cách thức cấu hình, cài đặt cho hệ thống. Ở chế độ mặc định được xem là cơ chế hoạt động tương đối tối ưu cho việc phát hiện các dạng tấn công mạng.

Kiến trúc của Suricata:

- Bước đầu tiên trong quá trình xử lý là thu thập các gói tin với module Packet Acquisition.
- Module này có chức năng thu thập gói tin từ công mạng và chuyển tiếp chúng đến để giải mã gói tin(decoder), nơi chịu trách nhiệm cho việc xác định các loại liên kết và chuẩn hóa dữ liệu cho các tiến trình khác.

- Tiếp theo, dữ liệu sẽ được chuyển tới stream module. Stream làm nhiệm vụ nhóm các dạng dữ liệu và reassembly các gói dữ liệu.
- Kế tiếp dữ liệu được đưa vào module phát hiện, nơi phân tích gói tin để phát hiện các tấn công mạng dựa trên các dấu hiệu. Cuối cùng, cảnh báo được đưa ra khi có các dấu hiệu được phát hiện và được gửi tới output module, dữ liệu đầu ra có thể được xác định ở nhiều dạng khác nhau.



Hình 2: Kiến trúc của Suricata

Phát hiện các dạng tấn công mạng

c. OSSEC

- OSSEC là hệ thống phát hiện dựa trên host (HIDS) dựa trên log mã nguồn mở, miễn phí, đa nền tảng có thể mở rộng và có nhiều cơ chế bảo mật khác nhau.
- OSSEC có thể phát hiện xâm nhập cả bằng chữ ký và dấu hiệu bất thường.

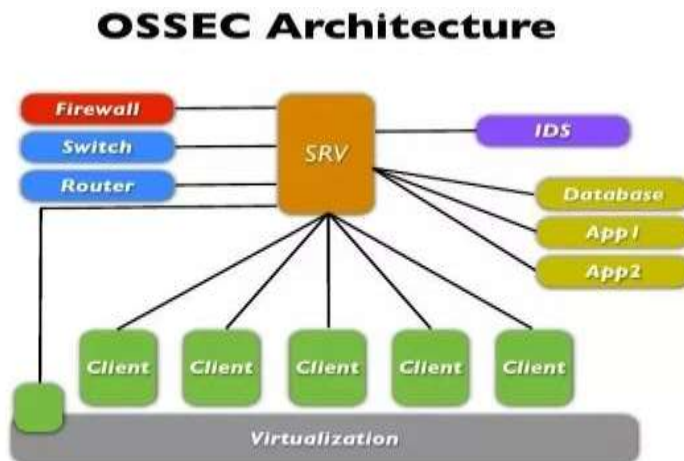
Kiến trúc của OSSEC: OSSEC được thiết kế theo mô hình client-server, gồm 2 thành phần chính là OSSEC Server và OSSEC Agent.

- OSSEC Server

- + Đây là phần trung tâm và quan trọng nhất của OSSEC, server là nơi lưu trữ dữ liệu. Tất cả bộ luật, bộ giải mã đều được lưu trữ trên server.
- + Server còn đảm nhận nhiệm vụ quản lý các agent. Các agent kết nối với máy chủ trên cổng 1514 hoặc 514, giao thức UDP. Kết nối các cổng này phải được cho phép để các agent kết nối với manager.
- + Nhiệm vụ quan trọng nhất của server là phân tích các log nhận từ các agent hay agentless và xuất ra cảnh báo. Các cảnh báo này có thể xuất ra cho các công cụ xử lý log và hiển thị cho người dùng, lưu trữ trong cơ sở dữ liệu

- OSSEC Agent

- + Agent là một chương trình nhỏ hoặc tập hợp các chương trình được cài đặt trên các hệ thống được giám sát.
- + Agent sẽ thu thập thông tin và gửi về cho manager để phân tích và so sánh. Một số thông tin được thu thập theo thời gian thực, một số thông tin khác thu thập theo định kỳ.
- + Agent có một bộ nhớ rất nhỏ và sử dụng rất ít CPU, không ảnh hưởng đến việc sử dụng của hệ thống. Server cấu hình cho các agent. Các agent được cài đặt trên các host và chúng gửi lại các log cho server thông qua giao thức thông điệp được mã hóa OSSEC.
- + Các modul chức năng của agent là: giám sát host, kiểm tra tính toàn vẹn file trên máy host mà nó được cài, phát hiện rootkit trên máy host, đọc các log và gửi các log cho server.
- + Agentless là tính năng hỗ trợ cho các thiết bị không cài đặt được agent theo cách bình thường như router, switch, tường lửa. Nó có chức năng như agent. Agentless kết nối để gửi thông điệp, log cho manager bằng các phương thức RPC.



Hình 3: Kiến trúc OSSEC Tính năng của OSSEC:

- Theo dõi và phân tích các log
 - + OSSEC thu thập log theo thời gian thực từ nhiều nguồn khác nhau để phân tích (giải mã, lọc và phân loại) và đưa ra cảnh báo dựa trên bộ luật đã được xây dựng trước.

- + Một số loại log mà OSSEC có thể phân tích là log proxy, log web, log ghi lại xác thực, system log.
- Kiểm tra tính toàn vẹn của file: Sử dụng hàm băm mật mã, có thể tính toán giá trị băm của mỗi file trong hệ điều hành dựa trên tên file, nội dung file và giá trị băm này là duy nhất.
- Giám sát Registry
 - + Hệ thống Registry là danh sách thư mục tất cả các cài đặt phần cứng và phần mềm, các cấu hình hệ điều hành, người dùng, nhóm người dùng, và các preference trên một hệ thống Microsoft Windows.
 - + Các thay đổi được thực hiện bởi người dùng và quản trị viên đối với hệ thống được ghi lại trong các khóa registry để các thay đổi được lưu khi người dùng đăng xuất hoặc hệ thống được khởi động lại.
- Phát hiện Rootkit
 - + Rootkit là công cụ cho phép kẻ đột nhập khả năng xâm nhập trở máy tính bị cài rootkit và xóa dấu vết về sự tồn tại của nó.
 - + OSSEC có khả năng phát hiện rootkit bằng cách đọc file cơ sở dữ liệu về rootkit và tiến hành quét hệ thống định kỳ, thực hiện các lời gọi hệ thống để phát hiện các file không bình thường, các tiến trình ẩn, các dấu hiệu vượt quyền, các cổng ẩn và so sánh chúng với cơ sở dữ liệu để phát hiện rootkit.
- Phản ứng chủ động: Phản ứng chủ động cho phép các IDS nói chung và OSSEC nói riêng tự động thực thi các lệnh hoặc phản ứng khi một sự kiện hoặc tập hợp sự kiện cụ thể được kích hoạt.

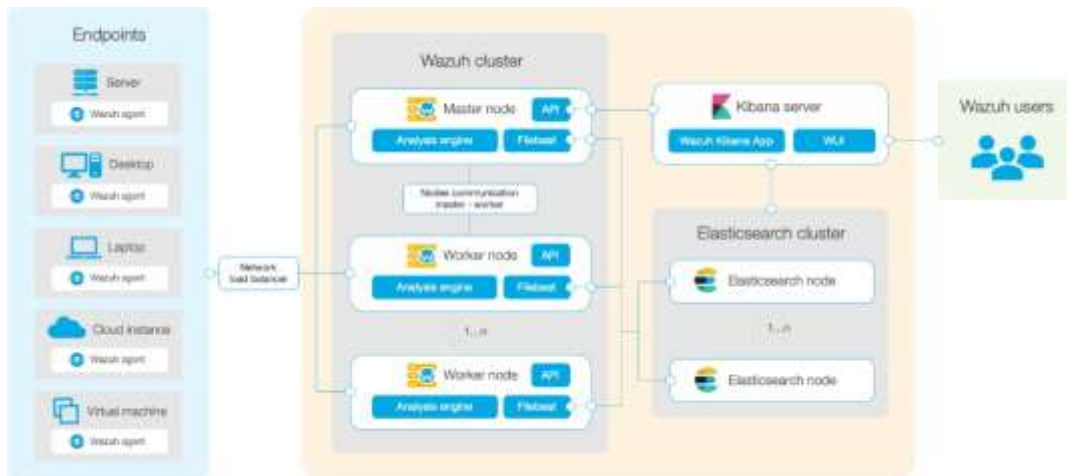
d. Wazuh

Wazuh là 1 project mã nguồn dùng cho việc bảo vệ an ninh. Được xây dựng từ các thành phần : OSSEC HIDS, OpenSCAP và Elastic Stack.

Kiến trúc của Wazuh:

- Wazuh agent:
 - + Dùng thu thập các dạng khác nhau của dữ liệu hệ thống và ứng dụng. Dữ liệu được chuyển tới Wazuh server thông qua 1 kênh được mã hóa và xác thực.
 - + Các agent có thể dùng để giám sát server vật lý, máy ảo, cloud instance (AWS, Azure hoặc Google cloud).
 - + Các agent task hoặc process khác nhau được dùng để giám sát hệ thống theo các cách khác nhau (giám sát sự thay đổi về file, đọc log, quét các thay đổi hệ thống).
- Wazuh server

- + Thành phần server phụ trách việc phân tích dữ liệu nhận từ agent, tạo các ngưỡng cảnh báo khi 1 event ánh xạ với rule (phát hiện xâm nhập, thay đổi file, cấu hình không tương thích với policy, rootkit...)
- + Server thông thường chạy các thành phần agent với mục tiêu giám sát chính nó.



Hình 4: Kiến trúc của Wazuh

Tính năng của Wazuh: Do được xây dựng từ các thành phần gồm OSSEC HIDS, OpenSCAP và Elastic Stack nên Wazuh sẽ có các tính năng sau:

- OSSEC HIDS:
 - + Phát hiện xâm nhập, hiển thị và giám sát.
 - + Cung cấp syslog server trung tâm và hệ thống giám sát không cần agent.
- OpenSCAP: được dùng để kiểm tra cấu hình hệ thống và phát hiện các ứng dụng dễ bị tấn công.
- ELK Stack Sử dụng cho việc thu thập, phân tích, lưu trữ, tìm kiếm và hiển thị dữ liệu log.

II. Nội dung thực hành

1. Chuẩn bị môi trường, công cụ

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

2. Các bước thực hiện

a. Cài đặt và cấu hình Snort

- Kiểm tra địa chỉ IP của máy tính Kali Linux và máy tính Ubuntu
- Địa chỉ IP máy tính Ubuntu

```

Processing triggers for libnss-ldap2 (2:1.1.0-1)
quangnh@Quang-B20DCAT144-Snort:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.150.136 netmask 255.255.255.0 broadcast 192.168.150.255
    inet6 fe80::518b:498f:711b:2660 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:28:1d:81 txqueuelen 1000 (Ethernet)
    RX packets 1413 bytes 1611926 (1.6 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 581 bytes 63242 (63.2 KB)
    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 160 bytes 22899 (22.8 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 160 bytes 22899 (22.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

- Địa chỉ IP máy tính Kali Linux

```

File Actions Edit View Help
(quangnh@Quang-B20DCAT144-Kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.150.137 netmask 255.255.255.0 broadcast 192.168.150.255
    inet6 fe80::20c:29ff:fe32:cd88 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:32:cd:88 txqueuelen 1000 (Ethernet)
    RX packets 4 bytes 808 (808.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 19 bytes 3124 (3.0 KiB)
    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 4 bytes 240 (240.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4 bytes 240 (240.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(quangnh@Quang-B20DCAT144-Kali)-[~]
$

```

- Tiến hành cài đặt Snort
 - sudo apt-get update
 - sudo apt-get upgrade
 - sudo apt-get install snort*

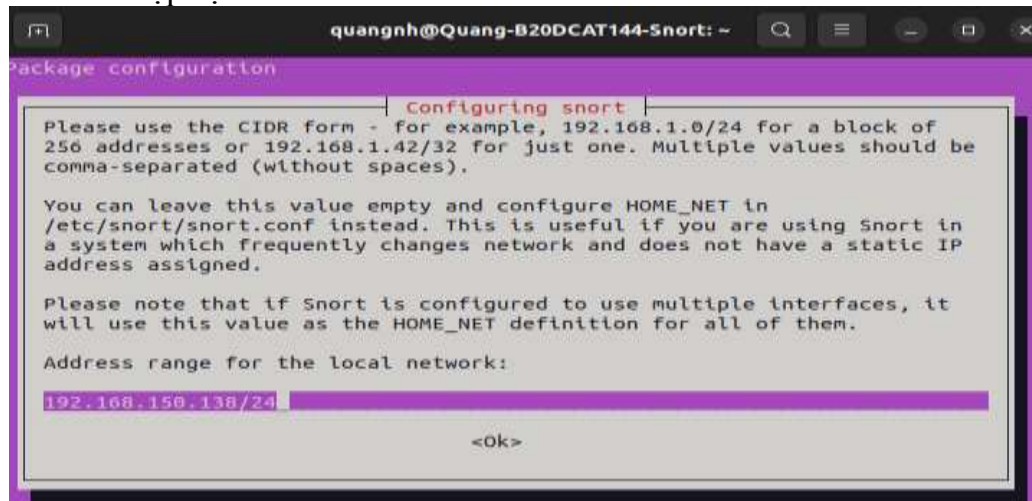
```

quangnh@Quang-B20DCAT144-Snort:~$ sudo apt-get install snort*
[sudo] password for quangnh:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'snort-pgsql' for glob 'snort*'
Note, selecting 'snort-rules-default' for glob 'snort*'
Note, selecting 'snort-mysql' for glob 'snort*'
Note, selecting 'snort-doc' for glob 'snort*'
Note, selecting 'snort-common-libraries' for glob 'snort*'
Note, selecting 'snort-rules' for glob 'snort*'
Note, selecting 'snort' for glob 'snort*'
Note, selecting 'snort-common' for glob 'snort*'
snort is already the newest version (2.9.15.1-6build1).
snort-common is already the newest version (2.9.15.1-6build1).
snort-common-libraries is already the newest version (2.9.15.1-6build1).
snort-doc is already the newest version (2.9.15.1-6build1).
snort-rules-default is already the newest version (2.9.15.1-6build1).
0 upgraded, 0 newly installed, 0 to remove and 4 not upgraded.
1 not fully installed or removed.
After this operation, 0 B of additional disk space will be used.
Do you want to continue? [Y/n] y
Setting up snort (2.9.15.1-6build1) ...
Snort configuration: interface default set, using ens33
Job for snort.service failed because the control process exited with error code.

```

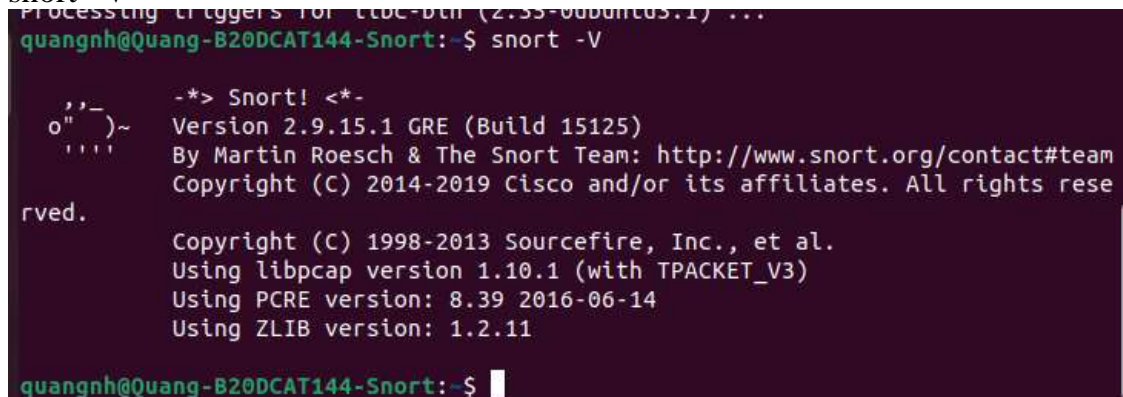
- Nhấn Y để tiếp tục
 - Cấu hình Snort

Nhập địa chỉ IP



- Kiểm tra snort đã cài đặt thành công hay chưa

snort -V



b. 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

Chỉnh sửa luật trong Snort

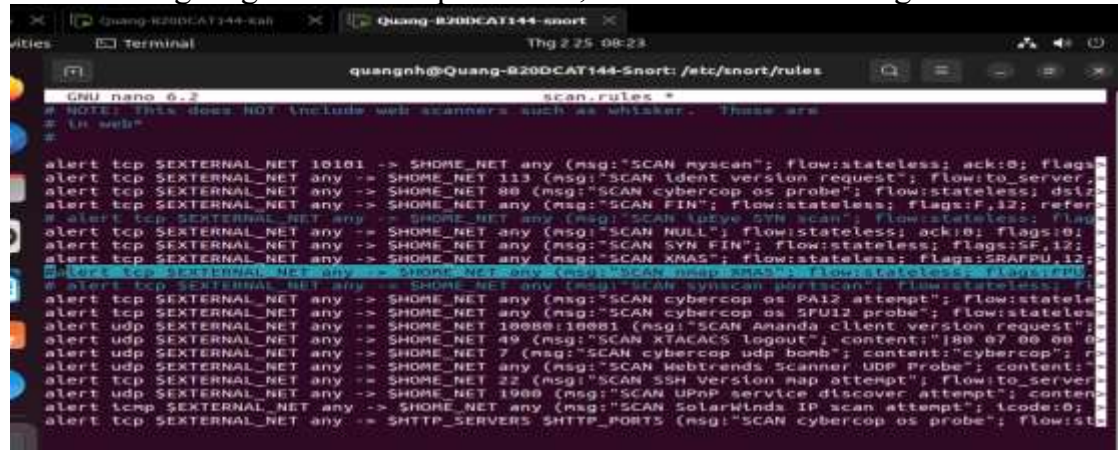
cd /etc/snort/rules/

ls



sudo nano scan.rules

Tìm tới dòng msg: SCAN nmap XMAS", đánh dấu # ở đầu dòng



```
quangnh@Quang-B20DCAT144-Snort: /etc/snort/rules
GNU nano 6.2 scan.rules
# NOTE: This does NOT include web scanners such as whisker. These are
# in web*.
#
alert tcp $EXTERNAL_NET 10101 -> $HOME_NET any (msg:'SCAN nmapscan'; flow:stateless; ack:0; flags:
alert tcp $EXTERNAL_NET any -> $HOME_NET 113 (msg:'SCAN ident version request'; flow:to_server;
alert tcp $EXTERNAL_NET any -> $HOME_NET 80 (msg:'SCAN cybercop os probe'; flow:stateless; dsiz:
# alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:'SCAN FIN'; flow:stateless; flags:F,32; refer
# alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:'SCAN deye SYN scan'; flow:stateless; flags:
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:'SCAN NULL'; flow:stateless; ack:0; flags:0;
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:'SCAN SYN FIN'; flow:stateless; flags:SF,12;
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:'SCAN XMAS'; flow:stateless; flags:SRAPPU,12;
# alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:'SCAN nmap XMAS'; flow:stateless; flow:cpu;
# alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:'SCAN synscan portscan'; flow:stateless; fl
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:'SCAN cybercop os PA12 attempt'; flow:statele
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:'SCAN cybercop os SPUI2 probe'; flow:stateles
alert udp $EXTERNAL_NET any -> $HOME_NET 10000:10001 (msg:'SCAN amanda client version request';
alert udp $EXTERNAL_NET any -> $HOME_NET 49 (msg:'SCAN XTACACS logout'; content:"100 07 00 00 0
alert udp $EXTERNAL_NET any -> $HOME_NET 7 (msg:'SCAN cybercop udp bomb'; content:"cybercop"; r
alert udp $EXTERNAL_NET any -> $HOME_NET any (msg:'SCAN Webtrends Scanner UDP Probe'; content:"
alert tcp $EXTERNAL_NET any -> $HOME_NET 22 (msg:'SCAN SSH Version map attempt'; flow:to_server
alert udp $EXTERNAL_NET any -> $HOME_NET 1900 (msg:'SCAN UPnP service discover attempt'; conten
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:'SCAN SolarWinds IP scan attempt'; lcode:0;
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:'SCAN cybercop os probe'; flow:st
```

Hiện thị thông điệp khi phát hiện có các gói ping gửi đến
sudo nano local.rules

Thêm dòng sau vào cuối file:

*alert icmp any any -> \$HOME_NET any (msg: "NHQB20DCAT144_Snort
detected ping packets are sending"; sid: 1000001;)*

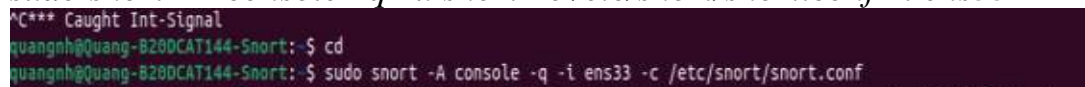


```
quangnh@Quang-B20DCAT144-Snort: /etc/snort/rules
GNU nano 6.2 local.rules
# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
# -----
# LOCAL RULES
# -----
# This file intentionally does not come with signatures. Put your local
# additions here.
alert icmp any any -> 192.168.150.139 any (msg: "NHQ_B20DCAT144_Snort
detected ping packets are sending"; sid: 1000001;)
```

Tại máy cài Snort ta gõ lệnh sau:

cd

sudo snort -A console -q -u snort -c /etc/snort/snort.conf -i ens33



```
quangnh@Quang-B20DCAT144-Snort: $ cd
quangnh@Quang-B20DCAT144-Snort: $ sudo snort -A console -q -i ens33 -c /etc/snort/snort.conf
```

Trên máy Kali, ta tiến hành ping
ping 192.168.150.136

```
(quangnh@Quang-B20DCAT144-Kali)-[~]
$ ping 192.168.150.136
PING 192.168.150.136 (192.168.150.136) 56(84) bytes of data.
64 bytes from 192.168.150.136: icmp_seq=1 ttl=64 time=0.570 ms
64 bytes from 192.168.150.136: icmp_seq=2 ttl=64 time=0.297 ms
64 bytes from 192.168.150.136: icmp_seq=3 ttl=64 time=0.470 ms
64 bytes from 192.168.150.136: icmp_seq=4 ttl=64 time=0.478 ms
64 bytes from 192.168.150.136: icmp_seq=5 ttl=64 time=0.405 ms
64 bytes from 192.168.150.136: icmp_seq=6 ttl=64 time=0.352 ms
64 bytes from 192.168.150.136: icmp_seq=7 ttl=64 time=0.410 ms
64 bytes from 192.168.150.136: icmp_seq=8 ttl=64 time=0.357 ms
```

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

```
Fatal Error, Quitting..
quangnh@Quang-B20DCAT144-Snort: $ sudo snort -A console -q -l em33 -c /etc/snort/snort.conf
02/27/13:18:52.005123 ** [1:1000001:0] "MQ_B20DCAT144_Snort detected ping packets are sending" ** [Priority: 0] [ICMP] 192.168.150.137 -> 192.168.150.136
02/27/13:18:52.005153 ** [1:1000001:0] "MQ_B20DCAT144_Snort detected ping packets are sending" ** [Priority: 0] [ICMP] 192.168.150.136 -> 192.168.150.137
02/27/13:18:53.016170 ** [1:1000001:0] "MQ_B20DCAT144_Snort detected ping packets are sending" ** [Priority: 0] [ICMP] 192.168.150.137 -> 192.168.150.136
02/27/13:18:53.016205 ** [1:1000001:0] "MQ_B20DCAT144_Snort detected ping packets are sending" ** [Priority: 0] [ICMP] 192.168.150.136 -> 192.168.150.137
02/27/13:18:54.042806 ** [1:1000001:0] "MQ_B20DCAT144_Snort detected ping packets are sending" ** [Priority: 0] [ICMP] 192.168.150.137 -> 192.168.150.136
02/27/13:18:54.042907 ** [1:1000001:0] "MQ_B20DCAT144_Snort detected ping packets are sending" ** [Priority: 0] [ICMP] 192.168.150.136 -> 192.168.150.137
02/27/13:18:55.004379 ** [1:1000001:0] "MQ_B20DCAT144_Snort detected ping packets are sending" ** [Priority: 0] [ICMP] 192.168.150.137 -> 192.168.150.136
02/27/13:18:55.004400 ** [1:1000001:0] "MQ_B20DCAT144_Snort detected ping packets are sending" ** [Priority: 0] [ICMP] 192.168.150.136 -> 192.168.150.137
02/27/13:18:56.093250 ** [1:1000001:0] "MQ_B20DCAT144_Snort detected ping packets are sending" ** [Priority: 0] [ICMP] 192.168.150.137 -> 192.168.150.136
02/27/13:18:56.093270 ** [1:1000001:0] "MQ_B20DCAT144_Snort detected ping packets are sending" ** [Priority: 0] [ICMP] 192.168.150.136 -> 192.168.150.137
02/27/13:18:57.121701 ** [1:1000001:0] "MQ_B20DCAT144_Snort detected ping packets are sending" ** [Priority: 0] [ICMP] 192.168.150.137 -> 192.168.150.136
02/27/13:18:57.121801 ** [1:1000001:0] "MQ_B20DCAT144_Snort detected ping packets are sending" ** [Priority: 0] [ICMP] 192.168.150.136 -> 192.168.150.137
02/27/13:18:58.147150 ** [1:1000001:0] "MQ_B20DCAT144_Snort detected ping packets are sending" ** [Priority: 0] [ICMP] 192.168.150.137 -> 192.168.150.136
02/27/13:18:58.147190 ** [1:1000001:0] "MQ_B20DCAT144_Snort detected ping packets are sending" ** [Priority: 0] [ICMP] 192.168.150.136 -> 192.168.150.137
```

- Phát hiện các gói tin rà quét từ bất kỳ một máy nào gửi đến máy chạy Snort trên cổng 80.

Tương tự như trên, ta cấu hình file local.rules để hiển thị thông báo khi phát hiện có các gói tin trên cổng 80

Thêm dòng sau vào cuối file:

alert tcp \$EXTERNAL_NET any -> \$HOME_NET 80 (msg: "B20DCAT144_Snort detected packets scanning on gate 80"; sid: 1001002; rev: 1;)

```
quangnh@Quang-B20DCAT144-Snort: /etc/snort/rules
GNU nano 2.2.1 local.rules
# This file intentionally does not come with signatures. Put your local
# additions here.
alert icmp any any -> $HOME_NET any (msg: "MQ_B20DCAT144_Snort detected ping packets are sending"; sid: 1000001;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 80 (msg: "MQ_B20DCAT144_Snort detected packets scanning on gate 80"; sid: 1001002; rev: 1;)
```


Tại máy cài Snort ta gõ lệnh sau:

`cd`

`sudo snort -A console -q -u snort -c /etc/snort/snort.conf -i ens33`

```
^C*** Caught Int-Signal
quangnh@Quang-B200CAT144-Snort:~$ cd
quangnh@Quang-B200CAT144-Snort:~$ sudo snort -A console -q -i ens33 -c /etc/snort/snort.conf
```

Trên máy Kali ta gõ lệnh sau

`sudo -i` (chuyển sang root)

`nmap 192.168.150.136` (dùng nmap để rà quét Snort)

```
(root@Quang-B200CAT144-Kali)-[~]
# nmap 192.168.150.136
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-27 01:54 EST
Nmap scan report for 192.168.150.136
Host is up (0.00037s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    closed http
MAC Address: 00:0C:29:28:1D:81 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 5.25 seconds
```

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

```
quangnh@Quang-B200CAT144-Snort:~$ sudo snort -A console -q -i ens33 -c /etc/snort/snort.conf
02/27-13:54:03.944726  ** [1:1001002:1] "NHQB20DCAT144_Snort detected packets scanning on gate 80" ** [Priority: 0] [TCP] 192.168.150.137:36582 -> 192.168.150.136:80
02/27-13:54:05.229946  ** [1:1001002:1] "NHQB20DCAT144_Snort detected packets scanning on gate 80" ** [Priority: 0] [TCP] 192.168.150.137:36587 -> 192.168.150.136:80
02/27-13:54:05.679499  ** [1:1418:1] SWMP request tcp ** [Classification: Attempted Information Leak] [Priority: 2] [TCP] 192.168.150.137:36582 -> 192.168.150.136:161
02/27-13:54:05.787352  ** [1:1418:1] SWMP request tcp ** [Classification: Attempted Information Leak] [Priority: 2] [TCP] 192.168.150.137:36584 -> 192.168.150.136:161
02/27-13:54:06.326264  ** [1:1001002:1] "NHQB20DCAT144_Snort detected packets scanning on gate 80" ** [Priority: 0] [TCP] 192.168.150.137:36589 -> 192.168.150.136:80
02/27-13:54:06.970970  ** [1:1421:1] SWMP AgentX/tcp request ** [Classification: Attempted Information Leak] [Priority: 2] [TCP] 192.168.150.137:36582 -> 192.168.150.136:705
02/27-13:54:07.879136  ** [1:1421:1] SWMP AgentX/tcp request ** [Classification: Attempted Information Leak] [Priority: 2] [TCP] 192.168.150.137:36584 -> 192.168.150.136:705
```

- Sử dụng công cụ hping3 để tấn công TCP SYN Flood máy Snort

Tương tự như trên, ta cấu hình file local.rules để hiển thị thông báo khi phát hiện đang bị tấn công TCP SYN Flood

Thêm dòng sau vào cuối file

`alert tcp any any -> $HOME_NET any (msg: "NHQB20DCAT144_Snort detected being attacked by TCP SYN Flood"; sid: 1001003; rev: 1;)`

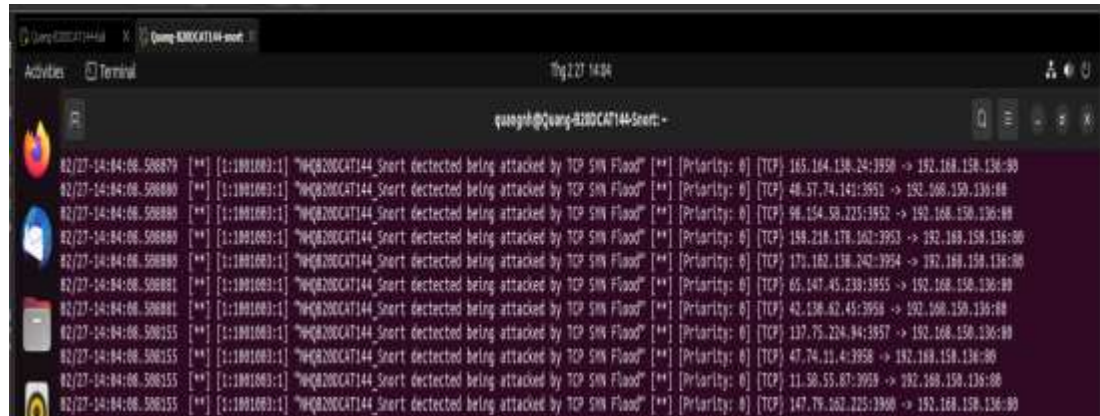
Tại máy cài Snort ta gõ lệnh sau:

`cd`

`sudo snort -A console -q -u snort -c /etc/snort/snort.conf -i ens33`


```
(root@Quang-B20DCAT144-Kali)-[~]
# sudo hping3 -c 15000 -d 120 -S -w 64 -p 80 --flood --rand-source 192.168.150.136
HPING 192.168.150.136 (eth0 192.168.150.136): S set, 40 headers + 120 data bytes
hping in flood mode, no replies will be shown
^C
— 192.168.150.136 hping statistic —
198034 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

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



```
02/27-14:04:06.508879 ** [1:1001003:1] "MQB20DCAT144_Snort detected being attacked by TCP SYN Flood" ** [Priority: 0] [TCP] 165.164.138.24:3958 -> 192.168.150.136:80
02/27-14:04:06.508880 ** [1:1001003:1] "MQB20DCAT144_Snort detected being attacked by TCP SYN Flood" ** [Priority: 0] [TCP] 48.37.74.141:3951 -> 192.168.150.136:80
02/27-14:04:06.508880 ** [1:1001003:1] "MQB20DCAT144_Snort detected being attacked by TCP SYN Flood" ** [Priority: 0] [TCP] 96.154.58.225:3952 -> 192.168.150.136:80
02/27-14:04:06.508880 ** [1:1001003:1] "MQB20DCAT144_Snort detected being attacked by TCP SYN Flood" ** [Priority: 0] [TCP] 199.218.178.162:3953 -> 192.168.150.136:80
02/27-14:04:06.508880 ** [1:1001003:1] "MQB20DCAT144_Snort detected being attacked by TCP SYN Flood" ** [Priority: 0] [TCP] 173.162.138.242:3954 -> 192.168.150.136:80
02/27-14:04:06.508881 ** [1:1001003:1] "MQB20DCAT144_Snort detected being attacked by TCP SYN Flood" ** [Priority: 0] [TCP] 65.147.45.238:3955 -> 192.168.150.136:80
02/27-14:04:06.508881 ** [1:1001003:1] "MQB20DCAT144_Snort detected being attacked by TCP SYN Flood" ** [Priority: 0] [TCP] 42.136.82.45:3958 -> 192.168.150.136:80
02/27-14:04:06.508153 ** [1:1001003:1] "MQB20DCAT144_Snort detected being attacked by TCP SYN Flood" ** [Priority: 0] [TCP] 137.75.229.84:3957 -> 192.168.150.136:80
02/27-14:04:06.508155 ** [1:1001003:1] "MQB20DCAT144_Snort detected being attacked by TCP SYN Flood" ** [Priority: 0] [TCP] 47.74.11.4:3958 -> 192.168.150.136:80
02/27-14:04:06.508155 ** [1:1001003:1] "MQB20DCAT144_Snort detected being attacked by TCP SYN Flood" ** [Priority: 0] [TCP] 11.30.55.87:3959 -> 192.168.150.136:80
02/27-14:04:06.508155 ** [1:1001003:1] "MQB20DCAT144_Snort detected being attacked by TCP SYN Flood" ** [Priority: 0] [TCP] 147.79.162.225:3960 -> 192.168.150.136:80
```

III. Tổng kết

Qua bài báo cáo trên, chúng ta đã cùng nhau tìm hiểu khái quát về hệ thống phát hiện tấn công xâm nhập (IDS), phân loại các hệ thống phát hiện xâm nhập và các kỹ thuật phát hiện xâm nhập. Ngoài ra, chúng ta cũng tìm hiểu về kiến trúc và tính năng của một số hệ thống phát hiện tấn công, xâm nhập như Snort, OSSEC, Suricata, Wazuh. Bài thực hành cũng cung cấp một số kỹ năng cài đặt Snort, tạo các luật trong Snort để phát hiện các gói tin ping từ một máy khác gửi đến Snort, phát hiện các gói tin rà quét từ bất kỳ một máy nào gửi đến Snort trên cổng 80, cấu hình Snort để thông báo về một cuộc tấn công TCP.

IV. Tài liệu tham khảo

1. [IDS Snort Installation & Rules Set-Up Guide | Step by Step in Ubuntu OS | Network Security Project - YouTube](#)
2. [CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG PHÁT HIỆN XÂM NHẬP - TaiLieu.VN](#)
3. [nguyen ngoc diepchuong 3 phat hien xam nhap 486.pdf \(tailieu.vn\)](#)
4. [TÌM HIỂU VỀ HỆ THỐNG PHÁT HIỆN XÂM NHẬP OSSEC \(Phần I\) \(viblo.asia\)](#)
5. [\[Network\] Tìm hiểu cơ chế, cách hoạt động của IDS \(phần 2\) \(viblo.asia\)](#)
6. [Ghi-chep-Suricata-/Tong quan ve Suricata.md at master · huongbn/Ghi-chep- Suricata- · GitHub](#)