



## BÁO CÁO LAB 2

*Môn: Hệ thống tìm kiếm, phát hiện và ngăn ngừa xâm nhập*

*GVTH: ĐỖ HOÀNG HIỀN*

Sinh viên thực hiện	<b>Sinh viên 1</b> MSSV: 21522573 Họ tên: Phạm Thanh Tâm <b>Sinh viên 2</b> MSSV: 21520514 Họ tên: Nguyễn Văn Anh Tú
Lớp	<b>NT204.O21.ATCL</b>
Tổng thời gian thực hiện Lab trung bình	
Phân chia công việc (nếu là nhóm)	<b>[Sinh viên 1]:</b> Yêu cầu 1,3  <b>[Sinh viên 2]:</b> Yêu cầu 2, 3
Link Video thực hiện (nếu có yêu cầu)	
Ý kiến (nếu có) + Khó khăn gặp phải + Đề xuất, góp ý...	
Điểm tự đánh giá (bắt buộc)	<b>10/10</b>



*[Nội dung báo cáo chi tiết – Trình bày tùy sinh viên, Xuất file .PDF khi nộp]*

## **BÁO CÁO CHI TIẾT**

**Yêu cầu 1: Sinh viên trả lời các câu hỏi bên dưới.**

### **1.1a. Tìm hiểu về Snort? Snort cho phép chạy trên những chế độ (mode) nào?**

- Snort là hệ thống ngăn chặn xâm nhập nguồn mở có khả năng phân tích lưu lượng truy cập và ghi nhật ký gói theo thời gian thực. Là hệ thống ngăn chặn xâm nhập mã nguồn mở hàng đầu thế giới. Snort sử dụng một loạt quy tắc giúp xác định hoạt động mạng độc hại và sử dụng các quy tắc đó để tìm các gói phù hợp với chúng và tạo cảnh báo cho người dùng.

- Snort chạy trên 3 mode:

+ Sniffer mode (chế độ ngấm): Trong chế độ này, Snort hoạt động như một công cụ giám sát mạng, chờ đợi và ghi lại các gói dữ liệu mạng mà nó nhận được. Chế độ này thường được sử dụng để xác định các vấn đề về bảo mật trong mạng, kiểm tra hoạt động mạng, và phát hiện xâm nhập.

+ Packet Logger Mode (Chế độ ghi gói dữ liệu): Trong chế độ này, Snort ghi lại toàn bộ hoặc một phần của các gói dữ liệu mạng mà nó nhận được vào một tập tin log. Thông thường, chế độ này được sử dụng để ghi lại lưu lượng mạng để phân tích sau này hoặc để duy trì một lịch sử hoạt động mạng.

+ Network Intrusion Detection System (NIDS) Mode (Chế độ phát hiện xâm nhập mạng): Đây là chế độ phổ biến nhất và mạnh mẽ nhất của Snort. Trong chế độ này, Snort sử dụng các quy tắc (rules) được định nghĩa trước để phát hiện các hoạt động không mong muốn hoặc các mẫu tấn công trong lưu lượng mạng. Khi Snort phát hiện một hoạt động không mong muốn, nó sẽ tạo ra cảnh báo để cảnh báo người quản trị hệ thống.

### **1.1b. Trình bày về những tính năng chính của Snort?**

- Snort có ba mục đích sử dụng chính:

+ Là một thám thính gói như tcpdump, như một trình ghi nhật kí gói – rất hữu ích cho việc gỡ lỗi lưu lượng truy cập mạng hoặc có thể sử dụng như một hệ thống ngăn chặn xâm nhập mạng toàn diện. Snort có thể được tải xuống và cấu hình để sử dụng cho cá nhân và doanh nghiệp

+ Snort có khả năng phát hiện các mẫu tấn công dựa trên các quy tắc và chính sách được cấu hình trước. Khi phát hiện một mẫu tấn công, nó sẽ tạo ra các cảnh báo để cảnh báo người quản trị hệ thống.

+ Snort có thể kiểm tra tất cả các gói tin trên mạng, phân tích chúng và kiểm tra xem chúng có chứa các biểu hiện của các cuộc tấn công hay không. Snort sử dụng một ngôn ngữ quy tắc mạnh mẽ để định nghĩa các biểu hiện của các loại tấn công. Người dùng có thể tạo ra các quy tắc tùy chỉnh dựa trên nhu cầu cụ thể của họ.

+ Snort có khả năng phát hiện tấn công dựa trên các chữ ký của chúng. Nó sử dụng cơ sở dữ liệu chữ ký để so sánh các gói tin mạng và xác định xem chúng có tương tự với các mẫu tấn công đã biết hay không.

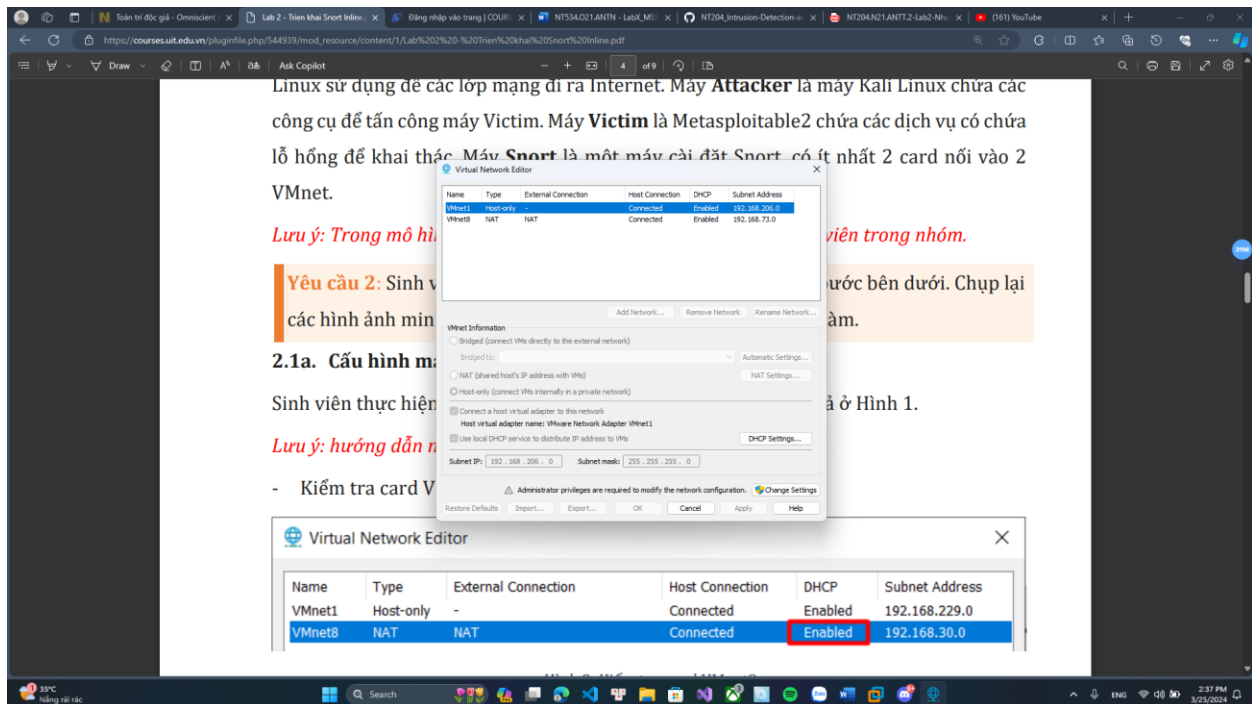
+ Snort không chỉ phát hiện tấn công dựa trên các chữ ký mẫu, mà còn dựa trên hành vi không bình thường của các gói tin mạng để xác định các mẫu tấn công mới hoặc không xác định trước. Snort cung cấp cơ chế cập nhật để cập nhật các quy tắc phát hiện mới và các chữ ký mới của các mẫu tấn công.

## 2. Cài đặt và cấu hình Snort để giám sát mạng

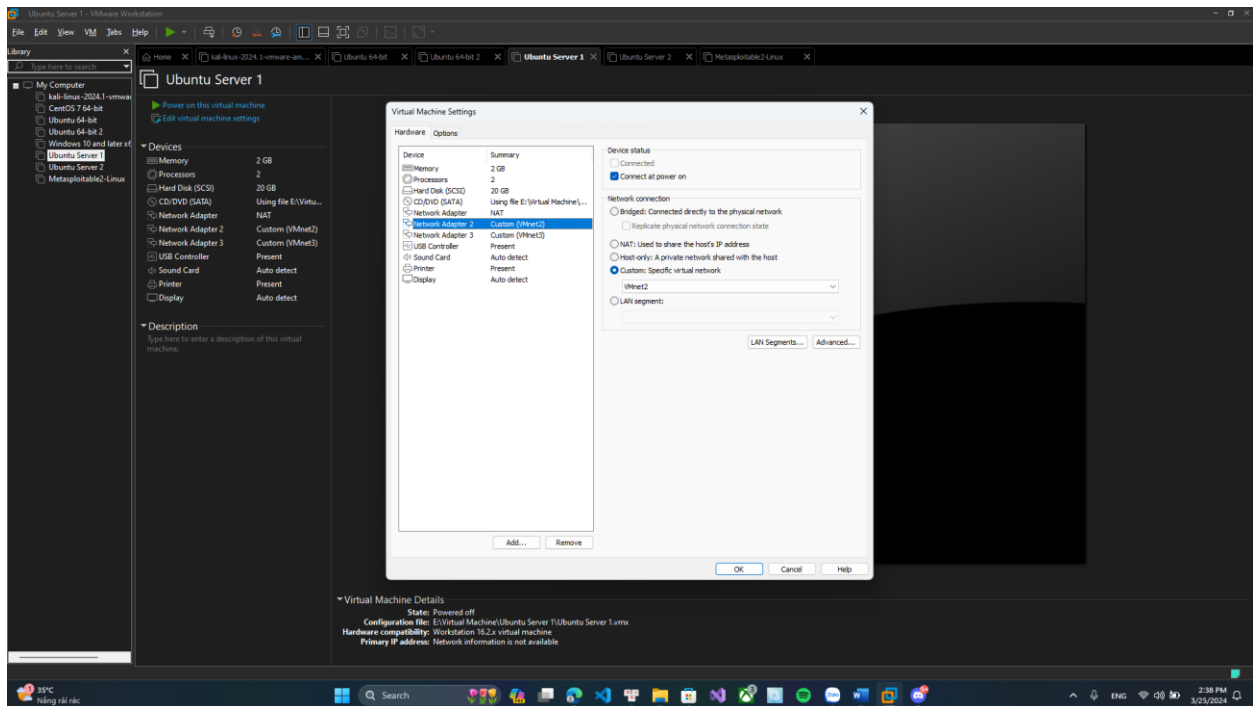
**Yêu cầu 2: Sinh viên cài đặt và cấu hình Snort Inline theo các bước bên dưới. Chụp lại các hình ảnh minh chứng (chụp full màn hình) cho từng bước làm**

### 2.1a. Cấu hình mạng cho các máy theo mô hình

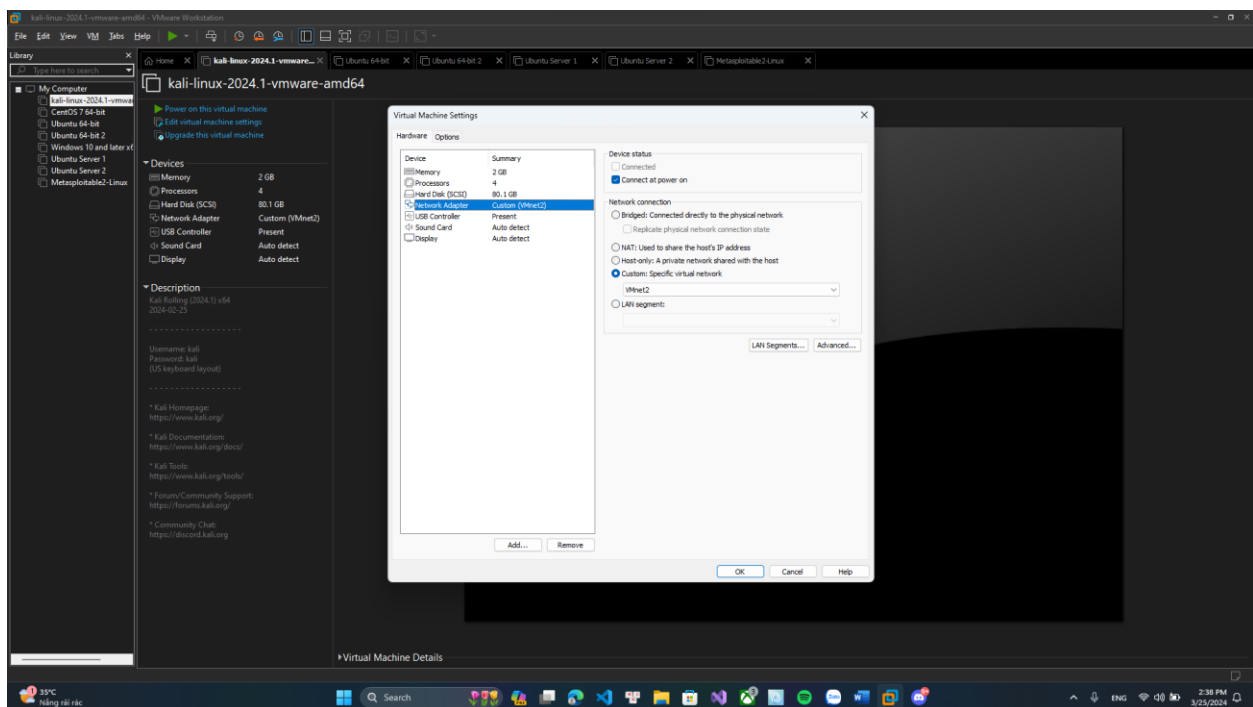
- Kiểm tra card VMnet8 (NAT) đã tồn tại và được bật DHCP



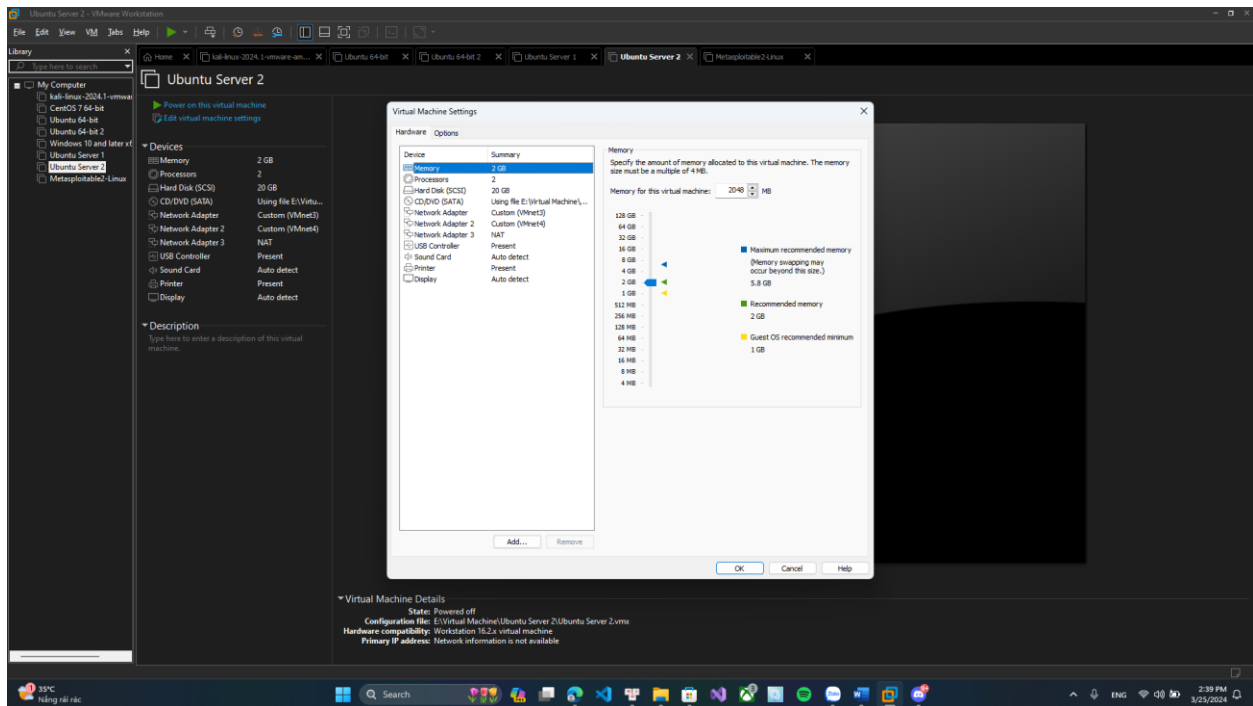
- Thêm card mạng cho Máy ảo Router:



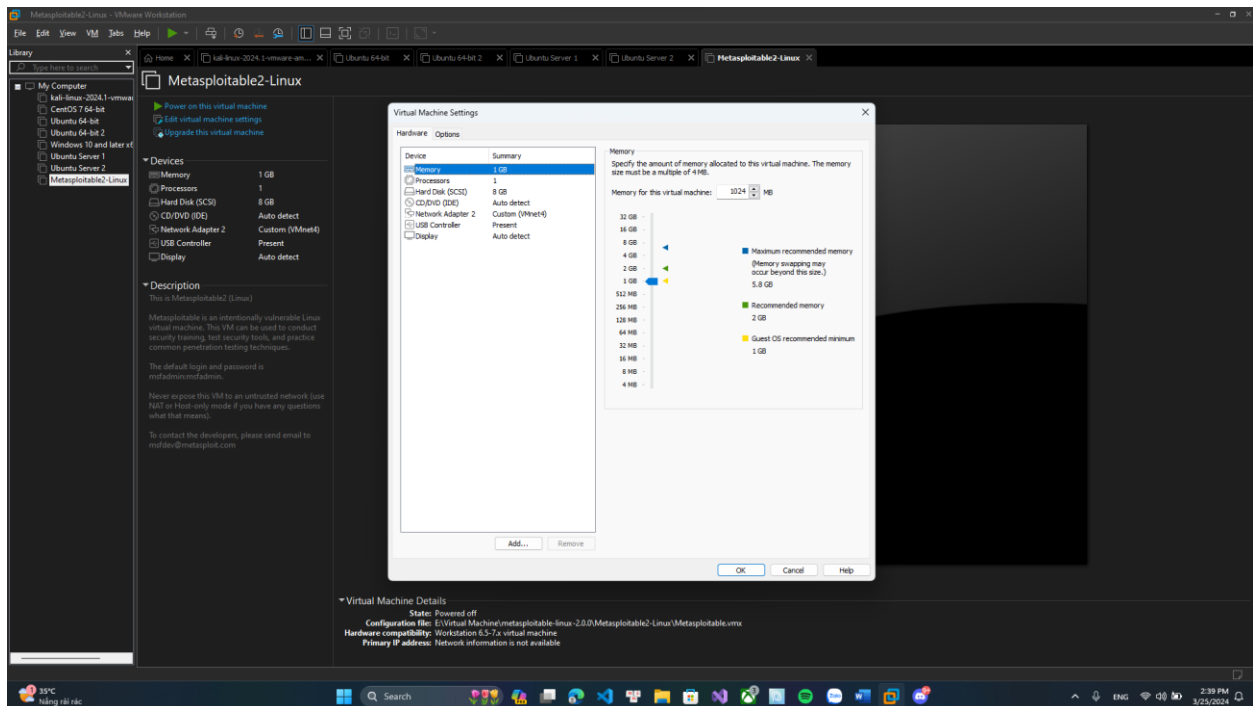
- Gán card mạng lên máy Kali:



- Gán card mạng lên máy ảo Snort:



- Gán card mạng lên máy ảo Metasploitable 2 (Victim):





## 2.1b. Cấu hình địa chỉ ip cho các máy

- Máy Victim:

```
GNU nano 2.0.7 File: /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet dhcp
iface eth0 inet static
address 192.168.73.200
netmask 255.255.255.0
gateway 192.168.73.1
```

- Máy Attacker:

```
kali@kali:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.01.73.100 netmask 255.255.255.0 broadcast 10.01.73.255
    inet6 fe80::6a59:1235:6b29:13e7 prefixlen 64 scopeid 0<link>
    ether 00:0c:29:a6:78:23 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 20 bytes 2442 (2.5 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 0<localhost>
    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

kali@kali:~$
```

- Máy Router:





```
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.73.128 netmask 255.255.255.0 broadcast 192.168.73.255
    inet6 fe80::20c:29ff:fe68:2f51 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:68:2f:51 txqueuelen 1000 (Ethernet)
    RX packets 11 bytes 1456 (1.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 17 bytes 1866 (1.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ens37: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.81.73.1 netmask 255.255.255.0 broadcast 10.81.73.255
    inet6 fe80::20c:29ff:fe68:2f5b prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:68:2f:5b txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 12 bytes 906 (906.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ens38: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.73.1 netmask 255.255.255.0 broadcast 192.168.73.255
    inet6 fe80::20c:29ff:fe68:2f5b prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:68:2f:5b txqueuelen 1000 (Ethernet)
    RX packets 1 bytes 329 (329.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 15 bytes 1086 (1.0 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 89 bytes 6824 (6.8 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 89 bytes 6824 (6.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

kop@kop1:~$
```

- Máy Snort:

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
kop@kop1:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet6 fe80::20c:29ff:fe68:2f51 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:68:2f:51 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 20 bytes 4555 (4.5 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ens34: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.73.130 netmask 255.255.255.0 broadcast 192.168.73.255
    inet6 fe80::20c:29ff:fe68:2f51 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:68:2f:51 txqueuelen 1000 (Ethernet)
    RX packets 579 bytes 648009 (648.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 253 bytes 27470 (27.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

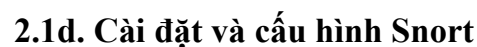
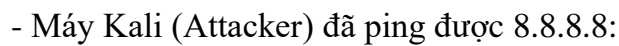
ens35: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet6 fe80::20c:29ff:fe68:2f51 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:68:2f:51 txqueuelen 1000 (Ethernet)
    RX packets 9 bytes 540 (540.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 20 bytes 4555 (4.5 KB)
    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 144 bytes 13204 (13.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 144 bytes 13204 (13.2 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

kop@kop1:~$
```

## 2.1c. Cấu hình NAT outbound cho máy router

- Set up Nat outbound:



- Cài đặt Snort:





```
Setting up libnss-perl (2.37-1) ...  
Setting up oinkmaster (2.0-4.1) ...  
Processing triggers for libc-bin (2.35-0ubuntu6) ...  
Processing triggers for man-db (2.10.2-1) ...  
Scanning processes...  
Scanning Linux images...  
  
Running kernel seems to be up-to-date.  
  
No services need to be restarted.  
  
No containers need to be restarted.  
  
No user sessions are running outdated binaries.  
  
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
kop@kop:~$ snort -v  
Running in packet dump mode  
  
== Initializing Snort ==  
Initializing Output Plugins!  
pcap DAQ configured to passive.  
Acquiring network traffic from "ens33".  
ERROR: Can't start DAQ (-1) - socket: operation not permitted!  
Fatal Error, Quitting..  
kop@kop:~$ snort -v  
  
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  
  
kop@kop:~$
```

- Kiểm tra afpacket DAQ:

```
No services need to be restarted.  
  
No containers need to be restarted.  
  
No user sessions are running outdated binaries.  
  
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
kop@kop:~$ snort -v  
Running in packet dump mode  
  
== Initializing Snort ==  
Initializing Output Plugins!  
pcap DAQ configured to passive.  
Acquiring network traffic from "ens33".  
ERROR: Can't start DAQ (-1) - socket: operation not permitted!  
Fatal Error, Quitting..  
kop@kop:~$ snort -v  
  
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  
  
kop@kop:~$ sudo su  
root@kop:/home/kop# snort --daq-list  
available DAQ modules:  
pcap(v3): readback live multi unpriv  
nfp(v7): live inline multi  
iota(v3): live inline multi unpriv  
dumplib(v3): readback live inline multi unpriv  
afpacket(v3): live inline multi unpriv  
root@kop:/home/kop#
```

- Xóa file rule mặc định, tạo file rule mới và cấu hình snort:



```
root@op:/home/kop# history
1 snort --daq-list
2 fw -rf /etc/snort/rules/r
3 touch /etc/snort/rules/nhm13.rules
4 cat /etc/snort/nhm13-snort.conf
5 touch /etc/snort/nhm13-snort.conf
6 nano /etc/snort/nhm13-snort.conf
7 history
root@op:/home/kop# cat /etc/snort/nhm13-snort.conf
config daq: afpacket
config daq_mode: inline
include /etc/snort/rules/nhm13.rules
root@op:/home/kop#
```

- Kiểm tra file cấu hình snort:

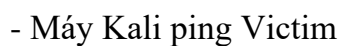
```
[rate-filter-config]-----
| memory-cap : 1048576 bytes
|-----[rate-filter-rules]-----
| none
|-----[event-filter-config]-----
| memory-cap : 1048576 bytes
|-----[event-filter-global]-----
| none
|-----[event-filter-local]-----
| none
|-----[suppression]-----
| none
|-----
Rule application order: pass->drop->sdrops->reject->alert->log
Verifying Preprocessor Configurations!
afpacket daq configured to inline
Acquiring network traffic from "ens34:ens35".
Decoding Ethernet

--- Initialization Complete ---

--> Snort! <--
Version 2.9.15.1 GRE (Build 15125)
By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
Copyright (C) 2004-2019 Cisco and/or its affiliates. All rights reserved.
Copyright (C) 1998-2019 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

Snort successfully validated the configuration!
Snort exiting
root@op:/home/kop#
```

- Chạy Snort inline:





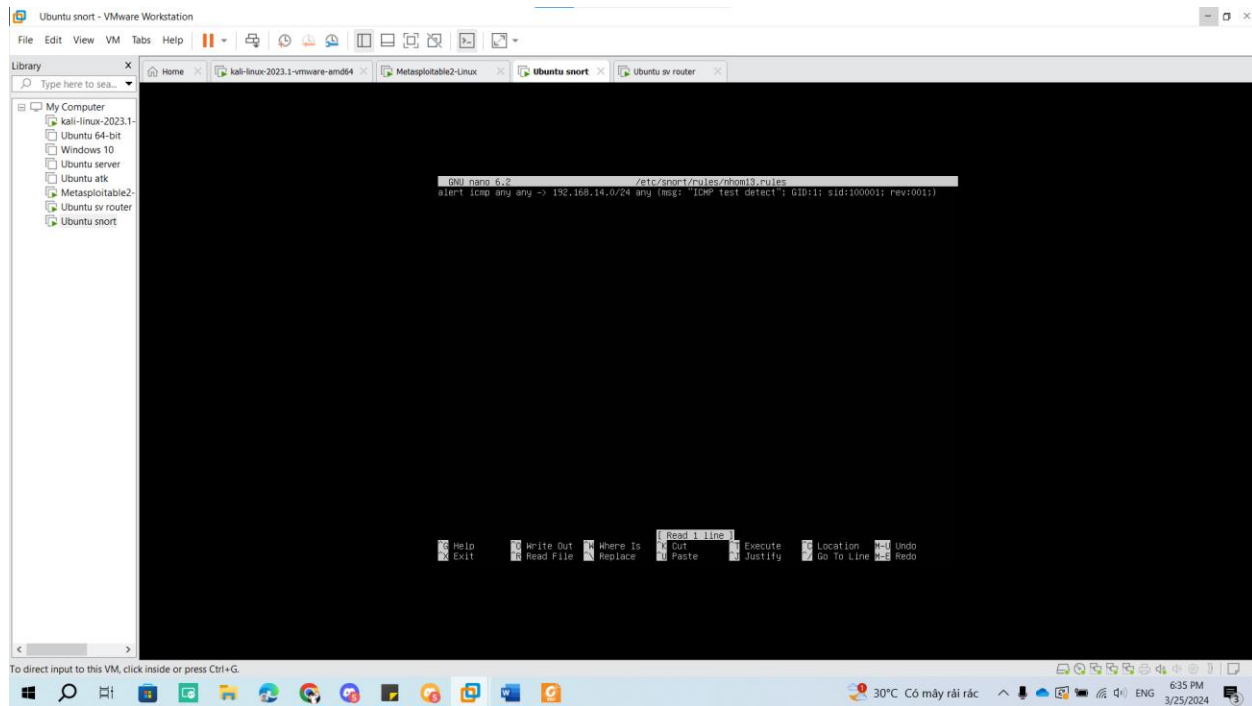
```
kali@kali:~$ ping 192.168.14.200
PING 192.168.14.200 (192.168.14.200) 56(84) bytes of data:
64 bytes from 192.168.14.200: icmp_seq=1 ttl=63 time=1.73 ms
64 bytes from 192.168.14.200: icmp_seq=2 ttl=63 time=1.68 ms
64 bytes from 192.168.14.200: icmp_seq=3 ttl=63 time=2.27 ms
64 bytes from 192.168.14.200: icmp_seq=4 ttl=63 time=2.38 ms
64 bytes from 192.168.14.200: icmp_seq=5 ttl=63 time=9.36 ms
64 bytes from 192.168.14.200: icmp_seq=6 ttl=63 time=2.09 ms
^C
--- 192.168.14.200 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 501ms
rtt min/avg/max/mdev = 1.682/3.251/9.356/2.741 ms
kali@kali:~$
```

## - Máy Victim ping google.com

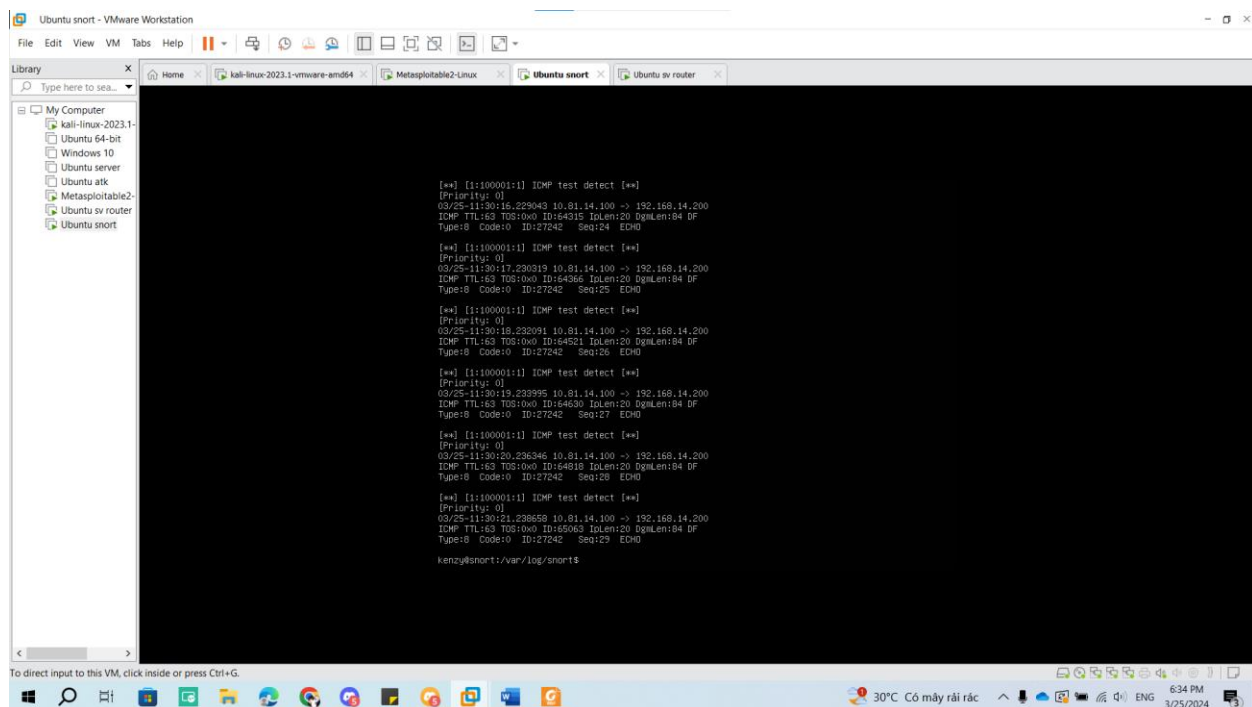
```
msfadmin@metasploitkali:/etc/network$ ping google.com
PING google.com (172.217.24.78) 56(84) bytes of data:
64 bytes from 78.24.217.172.in-addr.arpa (172.217.24.78): icmp_seq=1 ttl=127 time=53.0 ms
64 bytes from 78.24.217.172.in-addr.arpa (172.217.24.78): icmp_seq=2 ttl=127 time=52.4 ms
64 bytes from 78.24.217.172.in-addr.arpa (172.217.24.78): icmp_seq=3 ttl=127 time=57.0 ms
64 bytes from 78.24.217.172.in-addr.arpa (172.217.24.78): icmp_seq=4 ttl=127 time=51.9 ms
64 bytes from 78.24.217.172.in-addr.arpa (172.217.24.78): icmp_seq=5 ttl=127 time=53.3 ms
^C
--- google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 51.976/54.765/59.375/2.926 ms
msfadmin@metasploitkali:/etc/network$
```

## 2.1e. Viết rule cho Snort

- Viết rule phát hiện gói ICMP gửi đến lớp mạng 192.168.x.0/24 trong file `/etc/snort/rules/nhom13.rules` như sau:



- Kiểm tra log của snort



**Yêu cầu 3: Sinh viên viết rule drop các gói ICMP đi đến máy Victim (rule #1) sử dụng tcmdump trên máy Victim kiểm tra các trường hợp sau:**

- Trước khi viết áp dụng rule #1



- Sau khi áp dụng rule #1.

## Kiểm tra alert log của snort để xem kết quả

- Trước khi viết áp dụng rule #1

```
Metasploit2-Linux - VMware Workstation
File Edit View VM Tabs Help
Library
Type here to search
My Computer
kali-linux-2023.1-
  Ubuntu 64-bit
  Windows 10
  Ubuntu server
  Ubuntu atk
  Metasploit2-Linux
  Ubuntu sv router
  Ubuntu snort

06:40:06.091571 IP 192.168.14.200 > 10.01.14.100: ICMP echo reply, id 32569, seq
4, length 64
06:40:07.893730 IP 10.01.14.100 > 192.168.14.200: ICMP echo request, id 32569, s
eq 5, length 64
06:40:07.895922 IP 192.168.14.200 > 10.01.14.100: ICMP echo reply, id 32569, seq
5, length 64
06:40:08.885150 arp who-has 192.168.14.1 tell 192.168.14.200
06:40:08.885751 arp reply 192.168.14.1 is-at 00:0c:29:44:44:cb (oui Unknown)
06:40:08.895862 IP 10.01.14.100 > 192.168.14.200: ICMP echo request, id 32569, s
eq 6, length 64
06:40:08.895926 IP 192.168.14.200 > 10.01.14.100: ICMP echo reply, id 32569, seq
6, length 64
06:40:09.897709 IP 10.01.14.100 > 192.168.14.200: ICMP echo request, id 32569, s
eq 7, length 64
06:40:09.897853 IP 192.168.14.200 > 10.01.14.100: ICMP echo reply, id 32569, seq
7, length 64
06:40:10.900731 IP 10.01.14.100 > 192.168.14.200: ICMP echo request, id 32569, s
eq 8, length 64
06:40:10.900795 IP 192.168.14.200 > 10.01.14.100: ICMP echo reply, id 32569, seq
8, length 64
06:40:11.901771 IP 10.01.14.100 > 192.168.14.200: ICMP echo request, id 32569, s
eq 9, length 64
06:40:11.901856 IP 192.168.14.200 > 10.01.14.100: ICMP echo reply, id 32569, seq
9, length 64
-
```

- Sau khi viết áp dụng rule #1

```
Metasploit2-Linux - VMware Workstation
File Edit View VM Tabs Help
Library
Type here to search
My Computer
kali-linux-2023.1-
  Ubuntu 64-bit
  Windows 10
  Ubuntu server
  Ubuntu atk
  Metasploit2-Linux
  Ubuntu sv router
  Ubuntu snort

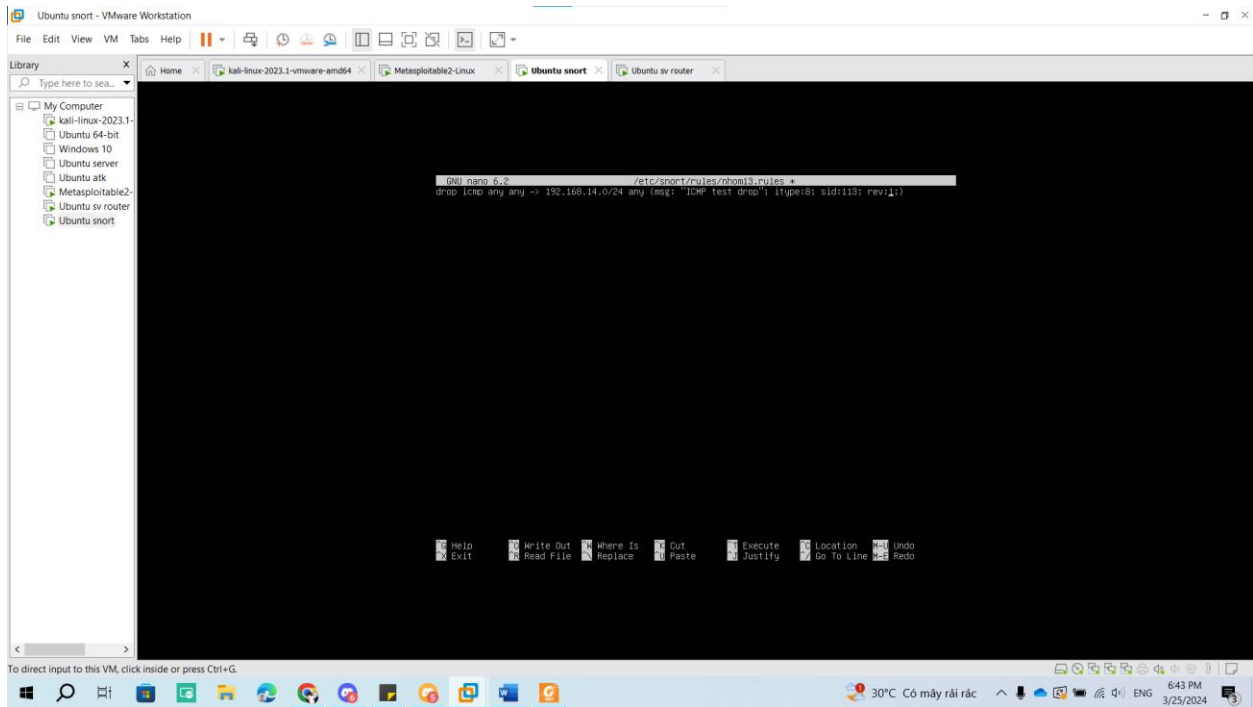
main
06:43:10.595101 arp who-has 192.168.14.1 tell 192.168.14.200
06:43:10.596341 arp reply 192.168.14.1 is-at 00:0c:29:44:44:cb (oui Unknown)

10 packets captured
10 packets received by filter
0 packets dropped by kernel
msfadmin@metasploit2-Linux:~/network$ sudo tcpdump -i eth0
tcpdump: verbose output suppressed, use -v or -ss for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
06:44:06.329566 arp who-has 192.168.14.200 tell 192.168.14.1
06:44:06.329624 arp reply 192.168.14.200 is-at 00:0c:29:44:44:cb (oui Unknown)
06:44:06.329976 IP 192.168.14.200.57792 > 10.5.14.2.domain: 22163* PTR 200.14.1
68.192.in-addr.arpa. (45)
06:44:06.334345 IP 10.5.14.2.domain > 192.168.14.200.57792: 22163* 1/0/0 (73)
06:44:06.334528 IP 192.168.14.200.52560 > 10.5.14.2.domain: 53300* PTR 1.14.168
.192.in-addr.arpa. (43)
06:44:06.340015 IP 10.5.14.2.domain > 192.168.14.200.52560: 53300* 1/0/0 (69)
06:44:06.342444 IP 192.168.14.200.44391 > 10.5.14.2.domain: 4949* PTR 2.14.5.10
.in-addr.arpa. (40)
06:44:06.346593 IP 10.5.14.2.domain > 192.168.14.200.44391: 4949* 1/0/0 PTR(1000
ain)
06:44:11.325630 arp who-has 192.168.14.1 tell 192.168.14.200
06:44:11.327340 arp reply 192.168.14.1 is-at 00:0c:29:44:44:cb (oui Unknown)
-
```





## - Rule #1 drop các gói ICMP



## - Alert log của Snort

