

BÁO CÁO BÀI TẬP

Môn học: Thực Hành Hệ Thống Tìm Kiếm, Phát Hiện Và Ngăn Ngừa Xâm Nhập

Lab 2: Triển khai Snort Inline

Lớp: NT204.P21.ANTT.2

GVHD: Trương Thị Hoàng Hảo

THÔNG TIN CHUNG

1. Thành viên

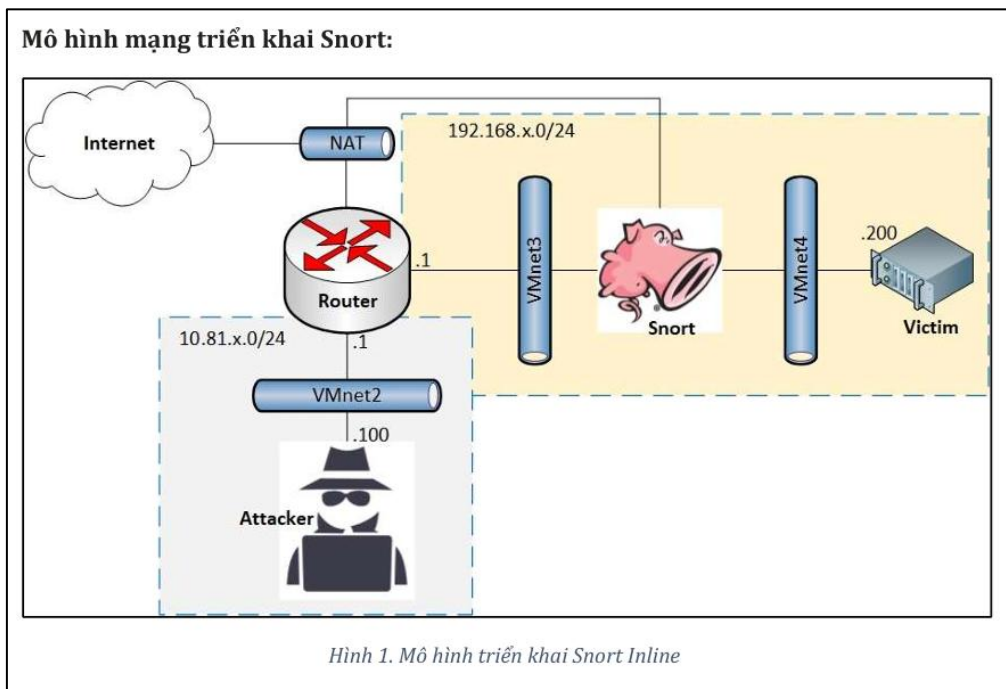
STT	Họ và tên	MSSV	Email
1	Lại Quan Thiên	22521385	22521385@gm.uit.edu.vn
2	Hồ Diệp Huy	22520541	22520541@gm.uit.edu.vn

2. Tiến Độ

STT	Nội dung	Tình Trạng	Trang
1	Yêu cầu 1	100%	2
2	Yêu cầu 2	100%	4
3	Yêu cầu 3	100%	17

Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.

3. Mô Hình Triển Khai (với x = 85)



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 gì? Snort là một hệ thống phát hiện xâm nhập (IDS) và ngăn chặn xâm nhập (IPS) mã nguồn mở, ban đầu được phát triển bởi Martin Roesch vào năm 1998 và hiện do Cisco Systems phát triển. Nó được thiết kế để giám sát lưu lượng mạng, bắt (capture) các gói tin, phân tích và so khớp với các “chữ ký” (signatures) được định nghĩa trong các luật (rules) nhằm phát hiện các hành vi xâm nhập, tấn công hoặc các sự cố bảo mật khác.

Snort cho phép chạy trên những chế độ nào? Snort có thể hoạt động ở 3 chế độ chính:

- **Sniffer mode:** Ở chế độ này, Snort chỉ đơn giản là bắt và hiển thị các gói tin đi qua mạng theo thời gian thực, tương tự như công cụ tcpdump. Đây là chế độ dùng để “nghe” lưu lượng mạng mà không ghi log hay phân tích sâu.

- **Packet logger mode:** Chế độ này ghi lại các gói tin bắt được vào các file log trên đĩa. Các file log này có thể lưu ở dạng ASCII hoặc nhị phân, phục vụ cho việc phân tích sau này.

- **Network Intrusion Detection System (NIDS) mode:** Đây là chế độ hoạt động “thông minh” nhất của Snort. Ở chế độ này, Snort sử dụng một tập hợp các luật (rules) được định nghĩa sẵn (hoặc do người dùng tự tạo) để phân tích, so khớp nội dung của các gói tin. Nếu phát hiện các mẫu tấn công (signature) hoặc hành vi bất thường, Snort sẽ ghi lại log và phát cảnh báo.

Ngoài ra, khi được cấu hình đúng, Snort có thể chạy ở **inline mode** – tức là hoạt động theo dạng IPS (Intrusion Prevention System) để không chỉ phát hiện mà còn chặn ngay các gói tin độc hại.

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

Các tính năng nổi bật của Snort bao gồm:

- Phát hiện xâm nhập (Intrusion Detection):

- + Snort giám sát lưu lượng mạng theo thời gian thực và sử dụng tập hợp các luật (rules) để so khớp với các “chữ ký” của các cuộc tấn công đã biết (ví dụ như quét cổng, tấn công buffer overflow, SQL injection, ...)

- + Khi có sự khớp giữa nội dung gói tin và quy tắc, hệ thống sẽ tạo ra cảnh báo và ghi lại các thông tin liên quan để phục vụ cho việc phân tích sự cố sau này.

- Ngăn chặn xâm nhập (Intrusion Prevention):

- + Khi Snort được cấu hình ở chế độ inline (IPS mode), ngoài việc phát hiện, nó còn có khả năng phản ứng ngay lập tức với các gói tin đáng ngờ.

- + Trong chế độ này, các hành động được thực hiện có thể bao gồm: **drop** (loại bỏ gói tin), **reject** (từ chối kết nối) hoặc **sdrop** (loại bỏ gói tin một cách “im lặng”), giúp ngăn chặn kịp thời các cuộc tấn công trước khi chúng có thể gây hại.

- Phản ứng và báo cáo (Alerting & Logging):

+ Sau khi phát hiện xâm nhập, Snort sẽ ghi log chi tiết và gửi cảnh báo tới quản trị viên (qua console, email, syslog hoặc lưu vào cơ sở dữ liệu).

+ Điều này cho phép người quản trị không chỉ nhận diện sớm các hành vi bất thường mà còn có thể thực hiện các biện pháp khắc phục hoặc phân tích sự cố một cách hiệu quả.

Một số tính năng khác như:

- **Mã nguồn mở và miễn phí:** Snort là phần mềm mã nguồn mở dưới giấy phép GNU GPL, cho phép người dùng tải về, sử dụng và tùy chỉnh theo nhu cầu.

- **Hệ thống luật (rules) mạnh mẽ:** Snort sử dụng các luật để so sánh nội dung của gói tin với các “chữ ký” của các cuộc tấn công đã biết. Người dùng có thể tự viết, chỉnh sửa và cập nhật các luật này để phù hợp với môi trường bảo mật của mình.

- **Kiến trúc module linh hoạt:** Snort được thiết kế theo kiến trúc module, với các thành phần chính:

+ **Module giải mã gói tin (Packet Decoder):** Nhận và giải mã các gói tin từ mạng.

+ **Module tiền xử lý (Preprocessors):** Chuẩn hóa, kết hợp và tái hợp các gói tin (ví dụ: xử lý phân mảnh) để chuẩn bị cho việc phân tích.

+ **Module phát hiện (Detection Engine):** So sánh các gói tin với các luật để phát hiện xâm nhập.

+ **Module log và cảnh báo (Logging and Alerting System):** Ghi log và phát ra các cảnh báo khi phát hiện sự bất thường.

+ **Module kết xuất thông tin (Output Module):** Cho phép xuất kết quả phân tích ra nhiều định dạng khác nhau (ví dụ: TCPDump, XML, cơ sở dữ liệu).

- **Khả năng chạy đa nền tảng:** Snort có thể chạy trên nhiều hệ điều hành như Linux, Windows và các hệ thống Unix, giúp dễ dàng triển khai trong nhiều môi trường mạng khác nhau.

- **Tính mở rộng và tích hợp:** Snort có thể được tích hợp với các công cụ và giao diện quản trị như BASE, Sguil, OSSIM... để hỗ trợ giám sát, phân tích log và báo cáo cảnh báo một cách trực quan.

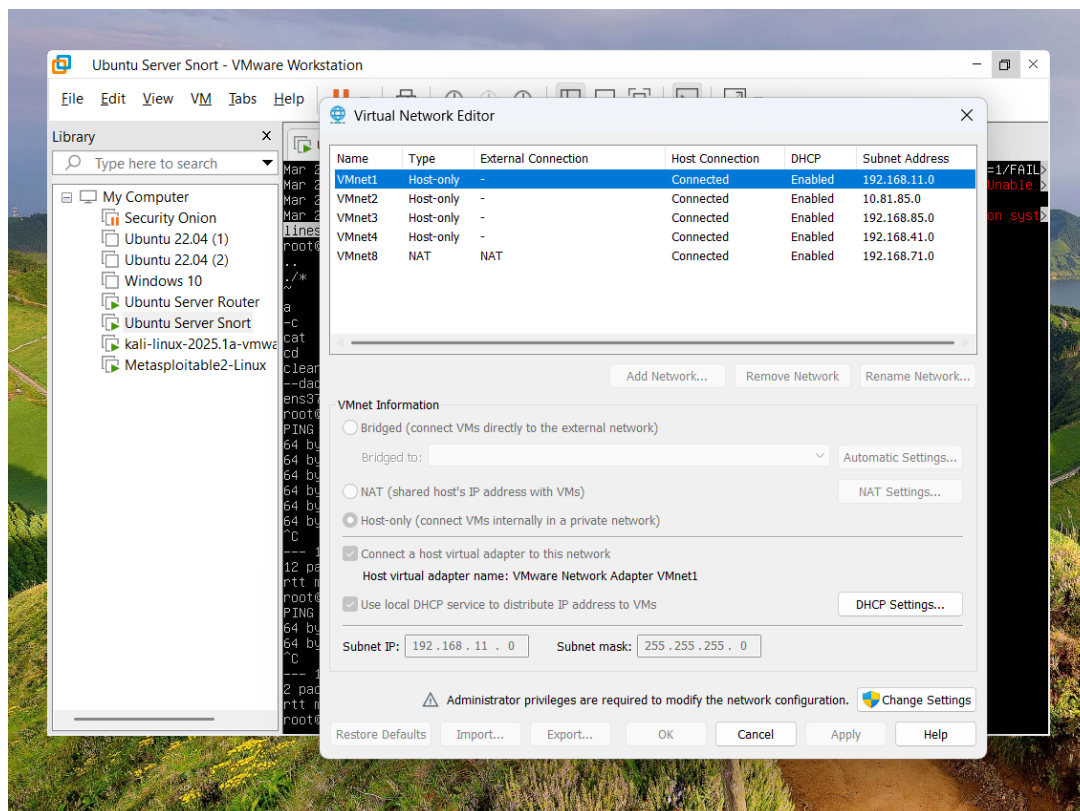
- **Khả năng hoạt động theo inline:** Khi cấu hình đúng, Snort có thể chạy ở chế độ inline để không chỉ phát hiện mà còn ngăn chặn các gói tin độc hại ngay trên đường đi, giúp tăng cường bảo vệ mạng.

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 và

2.1b. Cấu hình địa chỉ ip cho các máy

- Cấu hình các VMnet:



- Máy Router:

Ubuntu Server Router

Power on this virtual machine
[Edit virtual machine settings](#)

▼ Devices

Memory	4 GB
Processors	4
Hard Disk (SCSI)	96 GB
CD/DVD (SATA)	Using file C:\Use...
Network Adapter	NAT
Network Adapter 2	Custom (VMnet2)
Network Adapter 3	Custom (VMnet3)
USB Controller	Present
Sound Card	Auto detect
Display	Auto detect

▼ Description

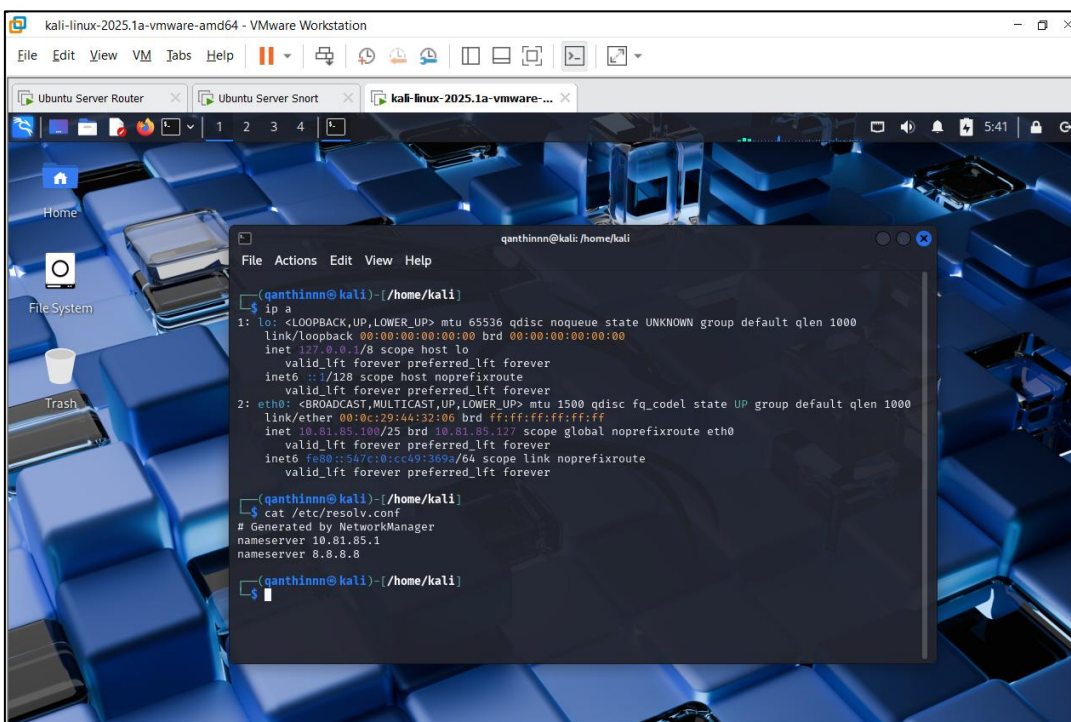
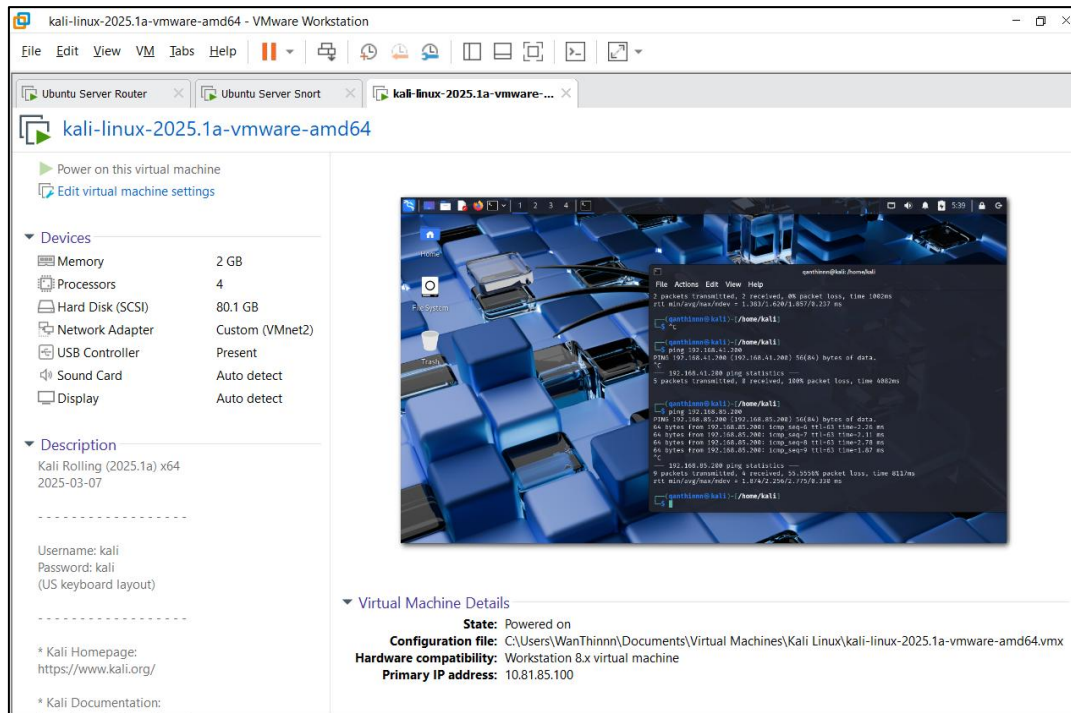
Type here to enter a description of this virtual machine.

Virtual Machine Details

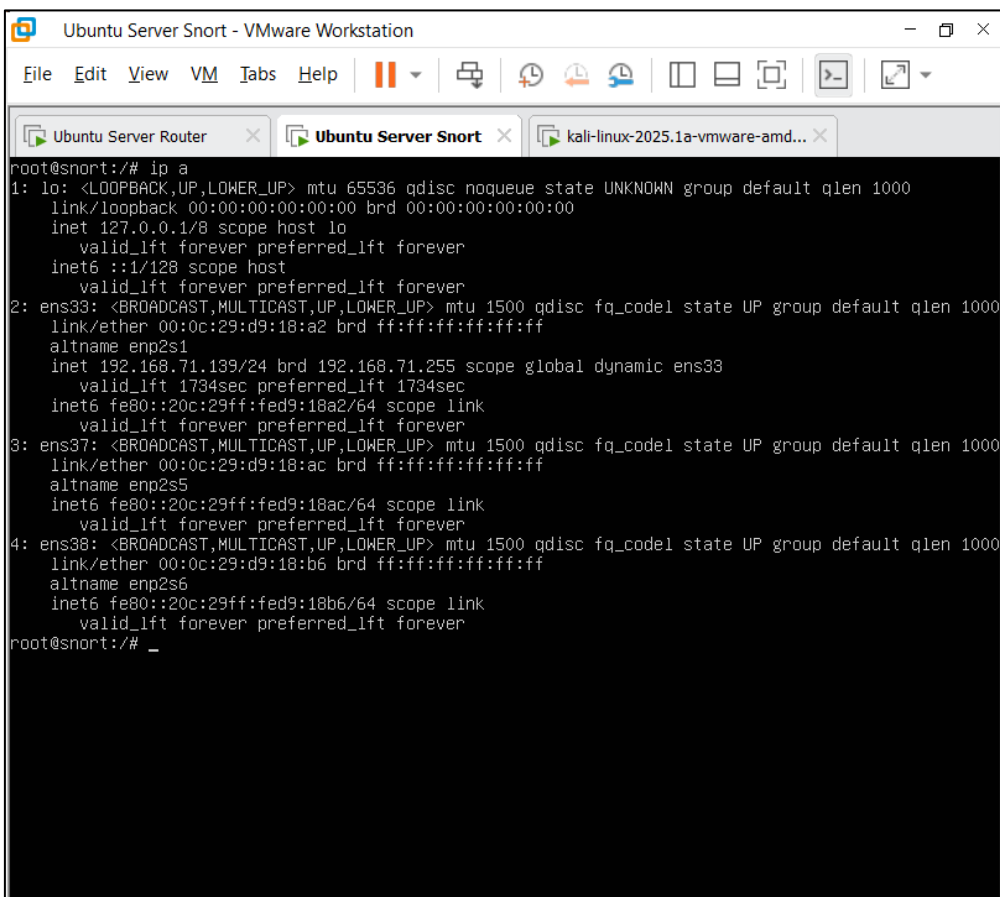
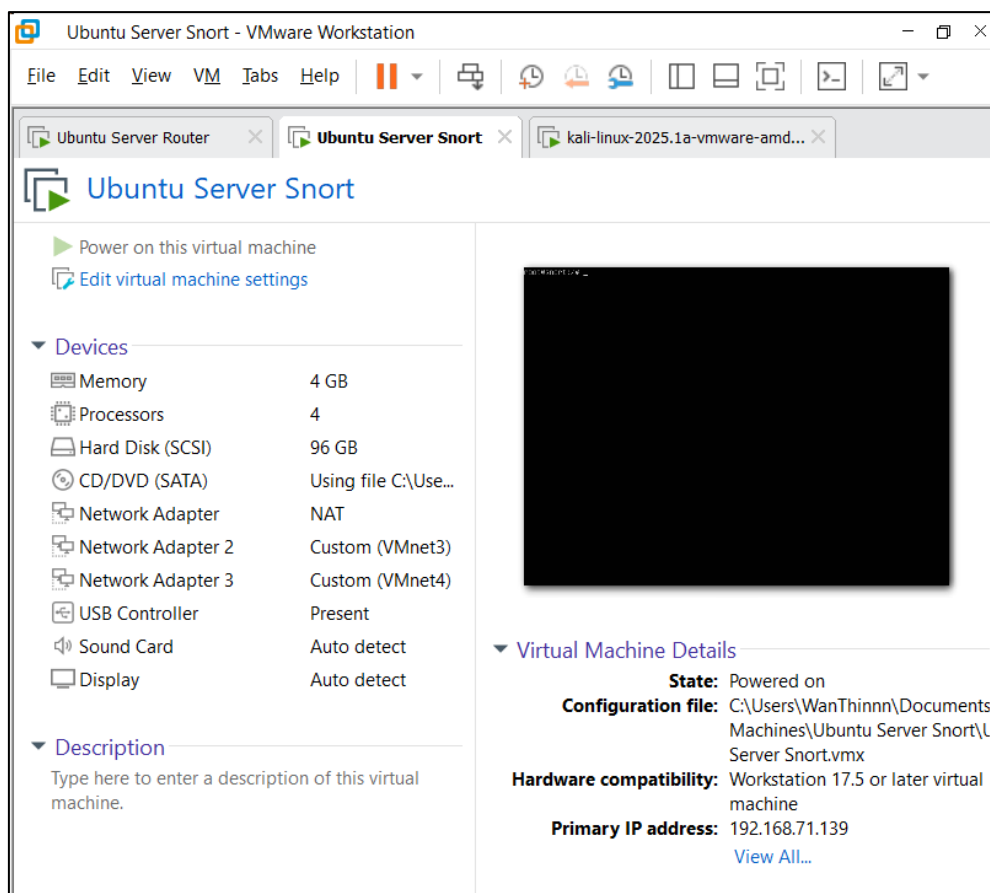
State: Powered on
Configuration file: C:\Users\WanThinnn\Document Machines\Ubuntu Server\Ubuntu Server.vmx
Hardware compatibility: Workstation 17.5 or later virtual machine
Primary IP address: 192.168.71.142
[View All...](#)

```
wanthinnn@router:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:fe:72:2e brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.71.142/24 brd 192.168.71.255 scope global dynamic ens33
        valid_lft 1711sec preferred_lft 1711sec
    inet6 fe80::20c:29ff:fefe:722e/64 scope link
        valid_lft forever preferred_lft forever
3: ens37: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:fe:72:38 brd ff:ff:ff:ff:ff:ff
    altname enp2s5
    inet 10.81.85.1/24 brd 10.81.85.255 scope global ens37
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fefe:7238/64 scope link
        valid_lft forever preferred_lft forever
4: ens38: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:fe:72:42 brd ff:ff:ff:ff:ff:ff
    altname enp2s6
    inet 192.168.85.1/24 brd 192.168.85.255 scope global ens38
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fefe:7242/64 scope link
        valid_lft forever preferred_lft forever
wanthinnn@router:~$
```


- Máy Attacker Kali Linux:



- Máy Snort:



- Máy Victim:

Metasploitable2-Linux - VMware Workstation

File Edit View VM Tabs Help

Metasploitable2-Linux

Power on this virtual machine
Edit virtual machine settings

▼ Devices

Memory	512 MB
Processors	1
Hard Disk (SCSI)	8 GB
CD/DVD (IDE)	Auto detect
Network Adapter	Custom (VMnet4)
USB Controller	Present
Display	Auto detect

▼ Description

This is Metasploitable2 (Linux)

Metasploitable is an intentionally vulnerable Linux virtual machine. This VM can be used to conduct security training, test security tools, and practice common penetration testing techniques.

The default login and password is
msfadmin:msfadmin.

▼ Virtual Machine Details

State: Powered on
Configuration file: C:\Users\WanThinn\Documents\Virtual Machines\Metasploitable2\Metasploitable.vmx
Hardware compatibility: Workstation 17.5 or later virtual machine
Primary IP address: Network information is not available

```
msfadmin@metasploitable:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:0c:29:cc:7a:4d brd ff:ff:ff:ff:ff:ff
    inet 192.168.85.200/24 brd 192.168.85.255 scope global eth0
        inet6 fe80::20c:29ff:fecc:7a4d/64 scope link
            valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$ cat /etc/resolv.conf
search localdomain
nameserver 8.8.8.8
nameserver 192.168.85.1
msfadmin@metasploitable:~$ _
```

Metasploitable2-Linux - VMware Workstation

Workstation

Metasploitable2-Linux

```
msfadmin@metasploitable:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:0c:29:cc:7a:4d brd ff:ff:ff:ff:ff:ff
    inet 192.168.85.200/24 brd 192.168.85.255 scope global eth0
        inet6 fe80::20c:29ff:fecc:7a4d/64 scope link
            valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$ cat /etc/resolv.conf
search localdomain
nameserver 8.8.8.8
nameserver 192.168.85.1
msfadmin@metasploitable:~$ _
```


2.1c. Cấu hình NAT outbound cho máy router

- Kích hoạt IP forwarding, cho phép máy chủ chuyển tiếp các gói tin giữa các mạng khác nhau, thường dùng trong NAT, VPN, hoặc router trên Linux.

```

GNU nano 6.2 /etc/sysctl.conf
#
# /etc/sysctl.conf - Configuration file for setting system variables
# See /etc/sysctl.d/ for additional system variables.
# See sysctl.conf(5) for information.
#
#kernel.domainname = example.com
#
# Uncomment the following to stop low-level messages on console
#kernel.printk = 3 4 1 3
#
#####
# Functions previously found in netbase
#
# Uncomment the next two lines to enable Spoof protection (reverse-path filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1
#
# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1
#
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1
#
# Uncomment the next line to enable packet forwarding for IPv6
# Enabling this option disables Stateless Address Autoconfiguration
# based on Router Advertisements for this host
#net.ipv6.conf.all.forwarding=1
  
```

- Cấu hình rule cho iptables trên Router:

+ Cho phép các máy trong mạng nội bộ (192.168.85.0/24 và 10.81.85.0/24) sử dụng IP của ens33 để truy cập Internet.

+ Kết hợp với lệnh **bật IP forwarding** (net.ipv4.ip_forward=1) để đảm bảo các gói tin được chuyển tiếp.

```

wanthinnn@router:~$ sudo iptables -t nat -S
-P PREROUTING ACCEPT
-P INPUT ACCEPT
-P OUTPUT ACCEPT
-P POSTROUTING ACCEPT
-A POSTROUTING -s 192.168.85.0/24 -o ens33 -j MASQUERADE
-A POSTROUTING -s 10.81.85.0/24 -o ens33 -j MASQUERADE
wanthinnn@router:~$ _
  
```

2.1d. Cài đặt và cấu hình Snort

- Cài đặt Snort:

```

root@snort:/# sudo apt-get install snort
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
snort is already the newest version (2.9.15.1-6build1).
0 upgraded, 0 newly installed, 0 to remove and 43 not upgraded.
root@snort:/# _

root@snort:/# snort --version
_*> 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

```

- Kiểm tra afpacket DAQ đã phải được cài đặt để sử dụng được mode inline.

```

wanthinn@snort:~$ sudo snort --daq-list
Available DAQ modules:
pcap(v3): readback live multi unpriv
nfq(v7): live inline multi
ipfw(v3): live inline multi unpriv
dump(v3): readback live inline multi unpriv
afpacket(v5): live inline multi unpriv

```

- Xóa tất cả file và tạo file rule nhóm9.rules

```

wanthinn@snort:~$ sudo su
root@snort:/home/wanthinn# cd /etc/snort/rules/
root@snort:/etc/snort/rules# ls
attack-responses.rules      community-web-dos.rules    policy.rules
backdoor.rules              community-web-iis.rules    pop2.rules
bad-traffic.rules           community-web-misc.rules   pop3.rules
chat.rules                  community-web-php.rules    porn.rules
community-bot.rules         ddos.rules                 rpc.rules
community-deleted.rules     deleted.rules              rservices.rules
community-dos.rules         dns.rules                  scan.rules
community-exploit.rules     dos.rules                  shellcode.rules
community-ftp.rules         experimental.rules         smtp.rules
community-game.rules        exploit.rules              snmp.rules
community-icmp.rules        finger.rules               sql.rules
community-imap.rules        ftp.rules                  telnet.rules
community-inappropriate.rules icmp-info.rules            tftp.rules
community-mail-client.rules icmp.rules                 virus.rules
community-misc.rules        imap.rules                 web-attacks.rules
community-nntp.rules        info.rules                 web-cgi.rules
community-oracle.rules      local.rules                web-client.rules
community-policy.rules      misc.rules                 web-coldfusion.rules
community-sip.rules         multimedia.rules           web-frontpage.rules
community-smtp.rules        mysql.rules                web-iis.rules
community-sql-injection.rules netbios.rules              web-misc.rules
community-virus.rules       nntp.rules                 web-php.rules
community-web-attacks.rules oracle.rules                 x11.rules
community-web-cgi.rules     other-ids.rules
community-web-client.rules  p2p.rules
root@snort:/etc/snort/rules# rm -rf ./*
root@snort:/etc/snort/rules# ls
root@snort:/etc/snort/rules# touch nhóm9.rules

```

- Tạo file cấu hình snort tại /etc/snort/nhom9-snort.conf

```
wanthinnn@snort:/etc/snort$ ls
attribute_table.dtd  file_magic.conf  reference.config  snort.debian.conf
classification.config  gen-msg.map      rules             threshold.conf
community-sid-msg.map  nhom9-snort.conf snort.conf        unicode.map
wanthinnn@snort:/etc/snort$ cat nhom9-snort.conf
config daq: afpacket
config daq_mode: inline

include /etc/snort/rules/nhom9.rules
wanthinnn@snort:/etc/snort$ _
```

- Kết quả kiểm tra file cấu hình snort:

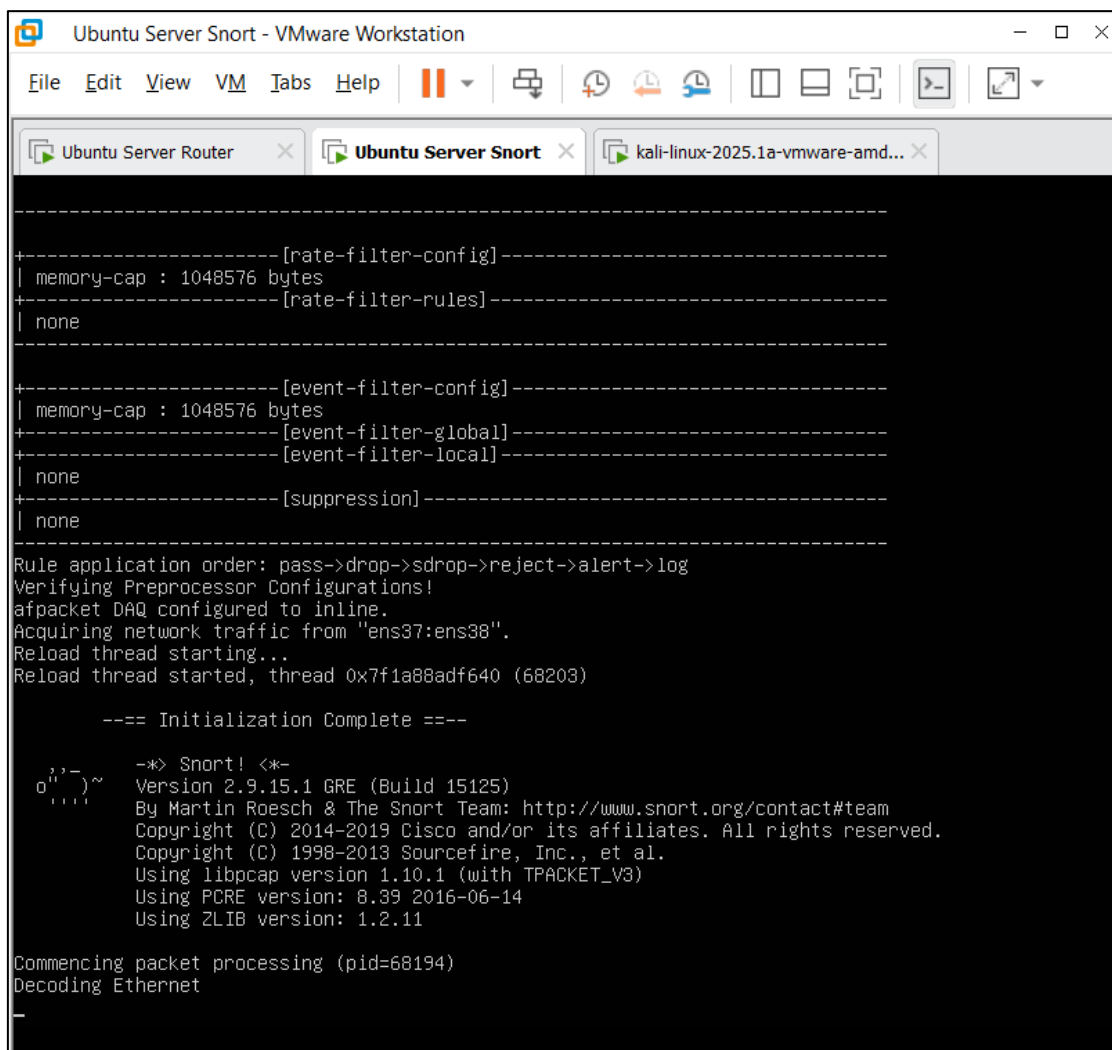
```
-----[rate-filter-config]-----
| memory-cap : 1048576 bytes
-----[rate-filter-rules]-----
| none
-----[event-filter-config]-----
| memory-cap : 1048576 bytes
-----[event-filter-global]-----
-----[event-filter-local]-----
| none
-----[suppression]-----
| none
-----
Rule application order: pass->drop->sdrop->reject->alert->log
Verifying Preprocessor Configurations!
afpacket DAQ configured to inline.
Acquiring network traffic from "ens37:ens38".
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) 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

Snort successfully validated the configuration!
Snort exiting
root@snort:/# _
```

- Chạy snort trong mode inline:



```
-----[rate-filter-config]-----
| memory-cap : 1048576 bytes
+-----[rate-filter-rules]-----
| none
-----

+-----[event-filter-config]-----
| memory-cap : 1048576 bytes
+-----[event-filter-global]-----
+-----[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 "ens37:ens38".
Reload thread starting...
Reload thread started, thread 0x7f1a88adf640 (68203)

---- Initialization Complete ----

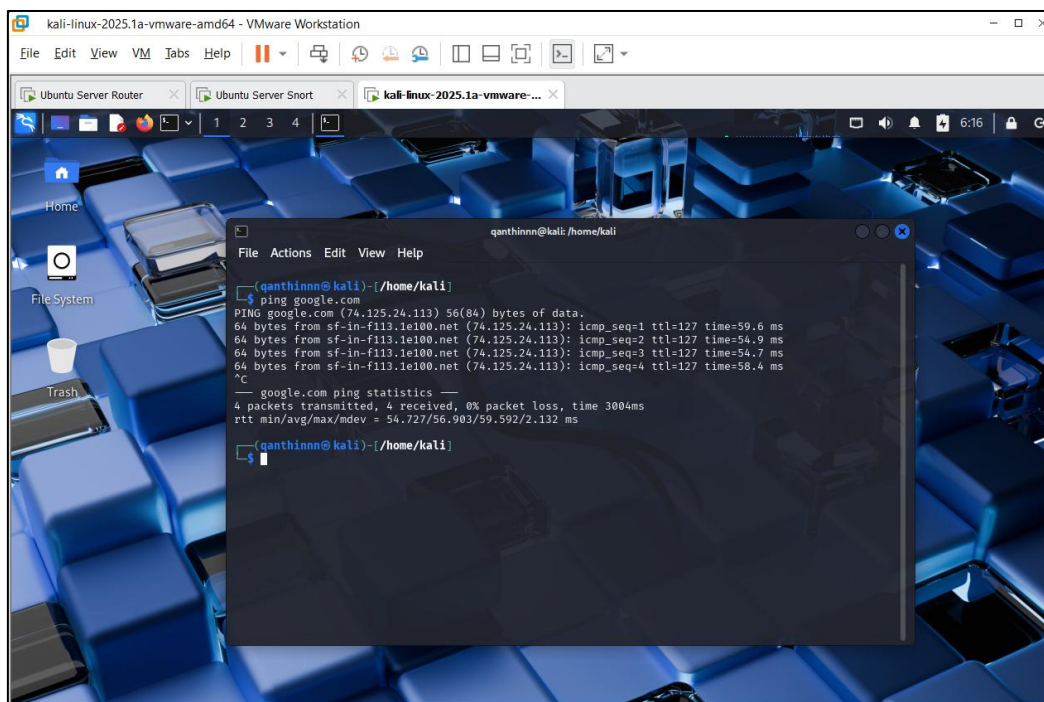
o''~
o''~
o''~
o''~

-*> Snort! <*-
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

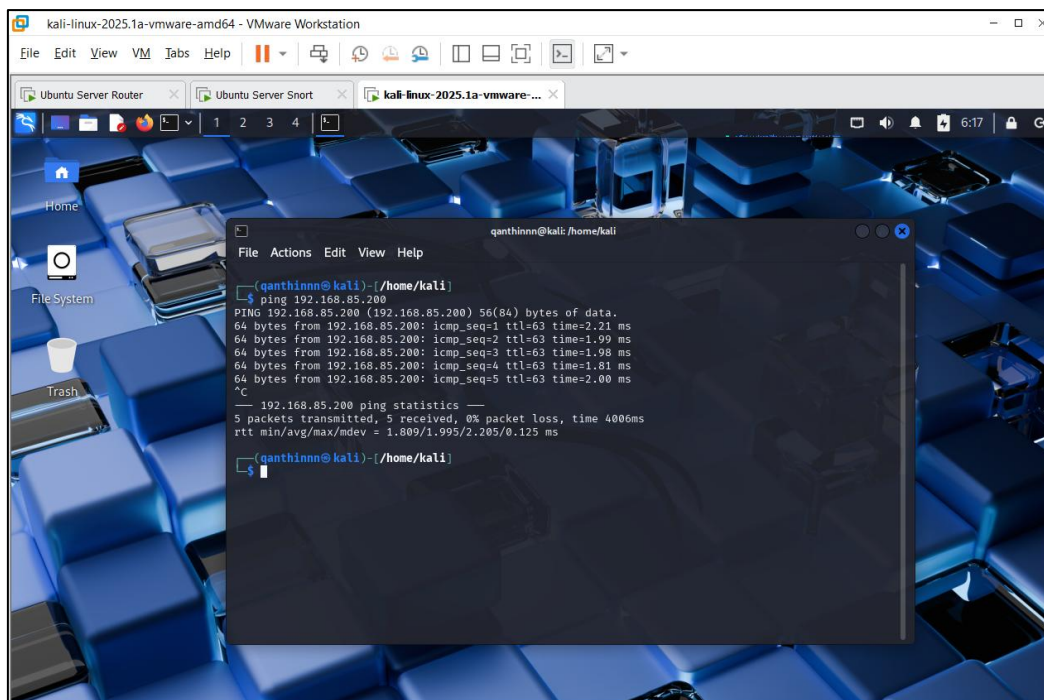
Commencing packet processing (pid=68194)
Decoding Ethernet
-
```

- Kiểm tra kết nối của các máy:

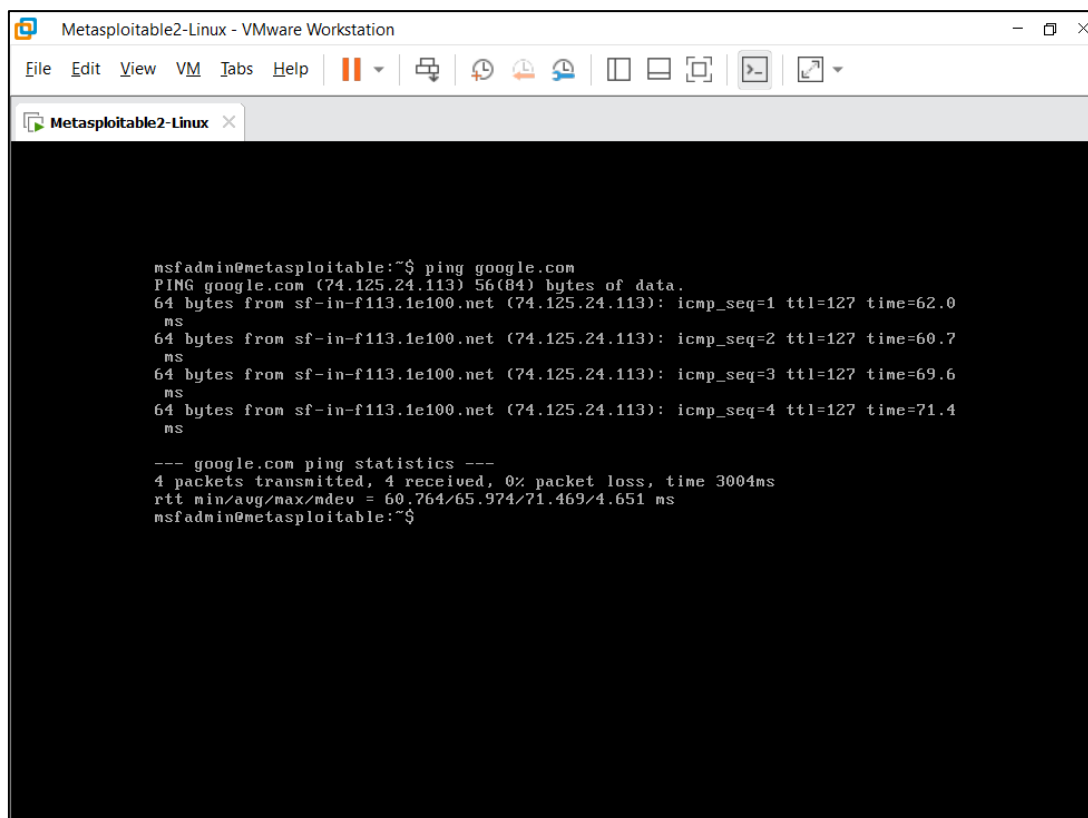
+ Máy Kali ping google.com



+ Máy Kali ping máy Victim



+ Máy Victim ping google.com



```
Metasploitable2-Linux - VMware Workstation
File Edit View VM Tabs Help
Metasploitable2-Linux x

msfadmin@metasploitable:~$ ping google.com
PING google.com (74.125.24.113) 56(84) bytes of data:
64 bytes from sf-in-f113.1e100.net (74.125.24.113): icmp_seq=1 ttl=127 time=62.0
ms
64 bytes from sf-in-f113.1e100.net (74.125.24.113): icmp_seq=2 ttl=127 time=60.7
ms
64 bytes from sf-in-f113.1e100.net (74.125.24.113): icmp_seq=3 ttl=127 time=69.6
ms
64 bytes from sf-in-f113.1e100.net (74.125.24.113): icmp_seq=4 ttl=127 time=71.4
ms

--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 60.764/65.974/71.469/4.651 ms
msfadmin@metasploitable:~$
```


2.1e. Viết rule cho Snort

- Viết rule phát hiện gói ICMP gửi đến lớp mạng 192.168.85.0/24 trong file /etc/snort/rules/nhom9.rules như sau:

```
root@snort:/# cat /etc/snort/rules/nhom9.rules
alert icmp any any -> 192.168.85.0/24 any (msg: "ICMP test detected";
GID:1; sid:10000001; rev:001;)
root@snort:/#
```

- Kiểm tra log của snort trên console:.

```

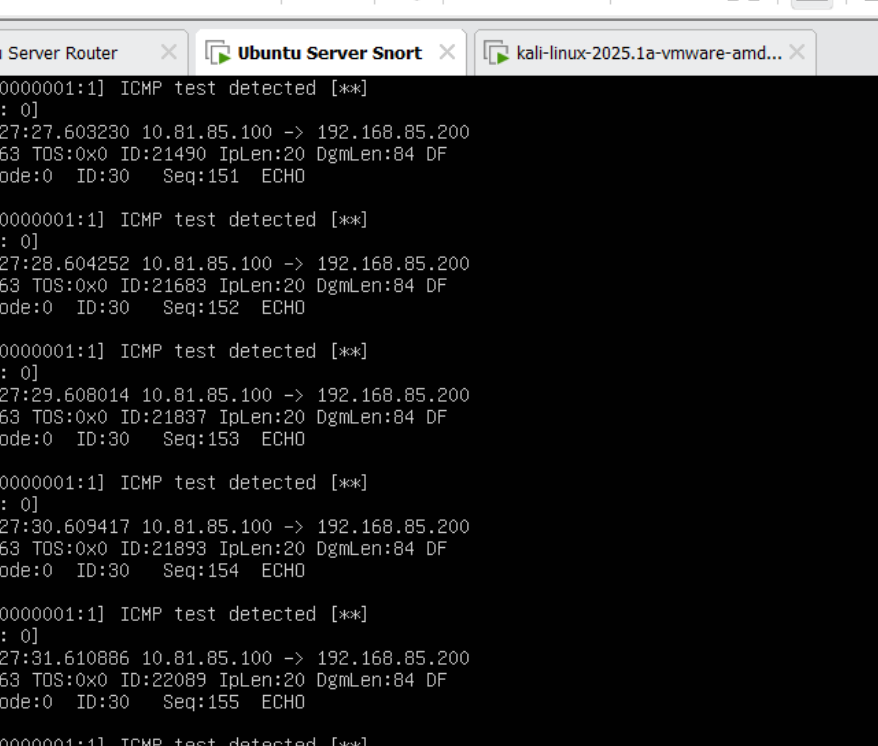
Ubuntu Server Snort - VMware Workstation
File Edit View VM Tabs Help
[Icons]
Ubuntu Server Router x Ubuntu Server Snort x kali-linux-2025.1a-vmware-amd... x
Reload thread starting...
Reload thread started, thread 0x7fa465496640 (68301)

---= Initialization Complete ===

o''~
'''~
-*> Snort! <*-
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

Commencing packet processing (pid=68292)
Decoding Ethernet
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
03/29-10:28:07.249554  [*] [1:10000001:1] ICMP test detected [*] [Priority: 0] {ICMP} 10.81.85.100
-> 192.168.85.200
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
03/29-10:28:08.249057  [*] [1:10000001:1] ICMP test detected [*] [Priority: 0] {ICMP} 10.81.85.100
-> 192.168.85.200
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
03/29-10:28:09.251598  [*] [1:10000001:1] ICMP test detected [*] [Priority: 0] {ICMP} 10.81.85.100
-> 192.168.85.200
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
03/29-10:28:10.252062  [*] [1:10000001:1] ICMP test detected [*] [Priority: 0] {ICMP} 10.81.85.100
-> 192.168.85.200
WARNING: No preprocessors configured for policy 0.

```



The screenshot shows a VMware Workstation window titled "Ubuntu Server Snort - VMware Workstation". The interface includes a menu bar (File, Edit, View, VM, Tabs, Help) and a toolbar with icons for file operations and window management. Below the toolbar, there are three tabs: "Ubuntu Server Router", "Ubuntu Server Snort", and "kali-linux-2025.1a-vmware-amd...". The "Ubuntu Server Snort" tab is active, displaying a terminal window with the following output:

```
root@snort:/#   
[**] [1:10000001:1] ICMP test detected [**]  
[Priority: 0]  
03/29-10:27:27.603230 10.81.85.100 -> 192.168.85.200  
ICMP TTL:63 TOS:0x0 ID:21490 IpLen:20 DgmLen:84 DF  
Type:8 Code:0 ID:30 Seq:151 ECHO  
  
[**] [1:10000001:1] ICMP test detected [**]  
[Priority: 0]  
03/29-10:27:28.604252 10.81.85.100 -> 192.168.85.200  
ICMP TTL:63 TOS:0x0 ID:21683 IpLen:20 DgmLen:84 DF  
Type:8 Code:0 ID:30 Seq:152 ECHO  
  
[**] [1:10000001:1] ICMP test detected [**]  
[Priority: 0]  
03/29-10:27:29.608014 10.81.85.100 -> 192.168.85.200  
ICMP TTL:63 TOS:0x0 ID:21837 IpLen:20 DgmLen:84 DF  
Type:8 Code:0 ID:30 Seq:153 ECHO  
  
[**] [1:10000001:1] ICMP test detected [**]  
[Priority: 0]  
03/29-10:27:30.609417 10.81.85.100 -> 192.168.85.200  
ICMP TTL:63 TOS:0x0 ID:21893 IpLen:20 DgmLen:84 DF  
Type:8 Code:0 ID:30 Seq:154 ECHO  
  
[**] [1:10000001:1] ICMP test detected [**]  
[Priority: 0]  
03/29-10:27:31.610886 10.81.85.100 -> 192.168.85.200  
ICMP TTL:63 TOS:0x0 ID:22089 IpLen:20 DgmLen:84 DF  
Type:8 Code:0 ID:30 Seq:155 ECHO  
  
[**] [1:10000001:1] ICMP test detected [**]  
[Priority: 0]  
03/29-10:27:32.611992 10.81.85.100 -> 192.168.85.200  
ICMP TTL:63 TOS:0x0 ID:22186 IpLen:20 DgmLen:84 DF  
Type:8 Code:0 ID:30 Seq:156 ECHO  
  
root@snort:/# _
```

Yêu cầu 3: Sinh viên viết rule drop các gói ICMP đi đến máy **Victim** (rule #1). Sử dụng **tcpdump** 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 áp dụng Rule #1:

- Viết rule alert ping từ bên ngoài:

```
root@snort:/# cat /etc/snort/rules/nhom9.rules
alert icmp any any -> 192.168.85.0/24 any (msg: "ICMP PING detected";
root@snort:/# _
```

- Khởi chạy Snort:

```

-----[rate-filter-config]-----
| memory-cap : 1048576 bytes
-----[rate-filter-rules]-----
| none
-----

-----[event-filter-config]-----
| memory-cap : 1048576 bytes
-----[event-filter-global]-----
-----[event-filter-local]-----
| none
-----[suppression]-----
| none
-----

Rule application order: pass->drop->sdrops->reject->alert->log
Verifying Preprocessor Configurations!

[ Port Based Pattern Matching Memory ]
afpacket DAQ configured to inline.
Acquiring network traffic from "ens37:ens38".
Reload thread starting...
Reload thread started, thread 0x7fbc09938640 (68342)

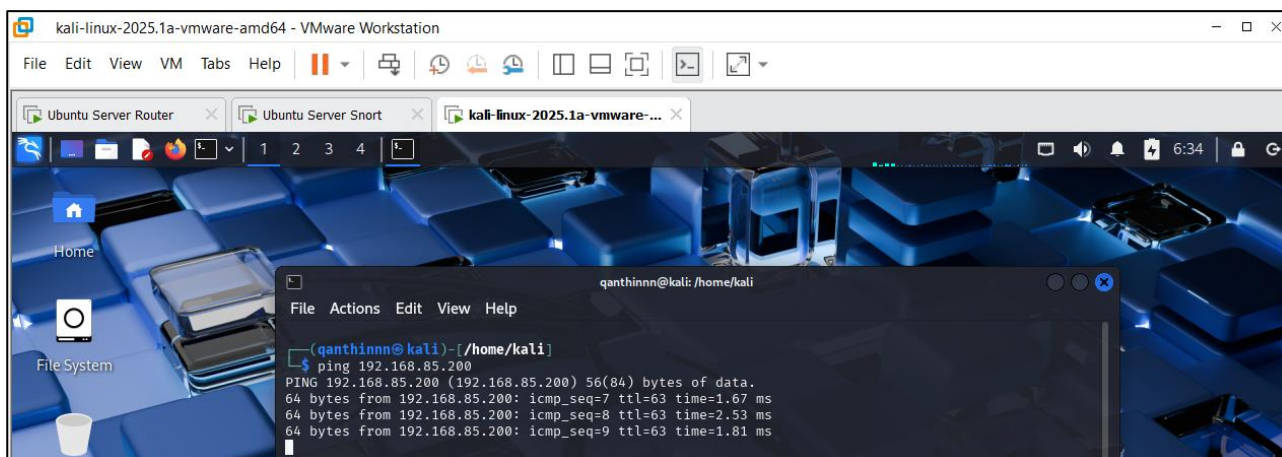
==== Initialization Complete ====

o''')~
  ''')~

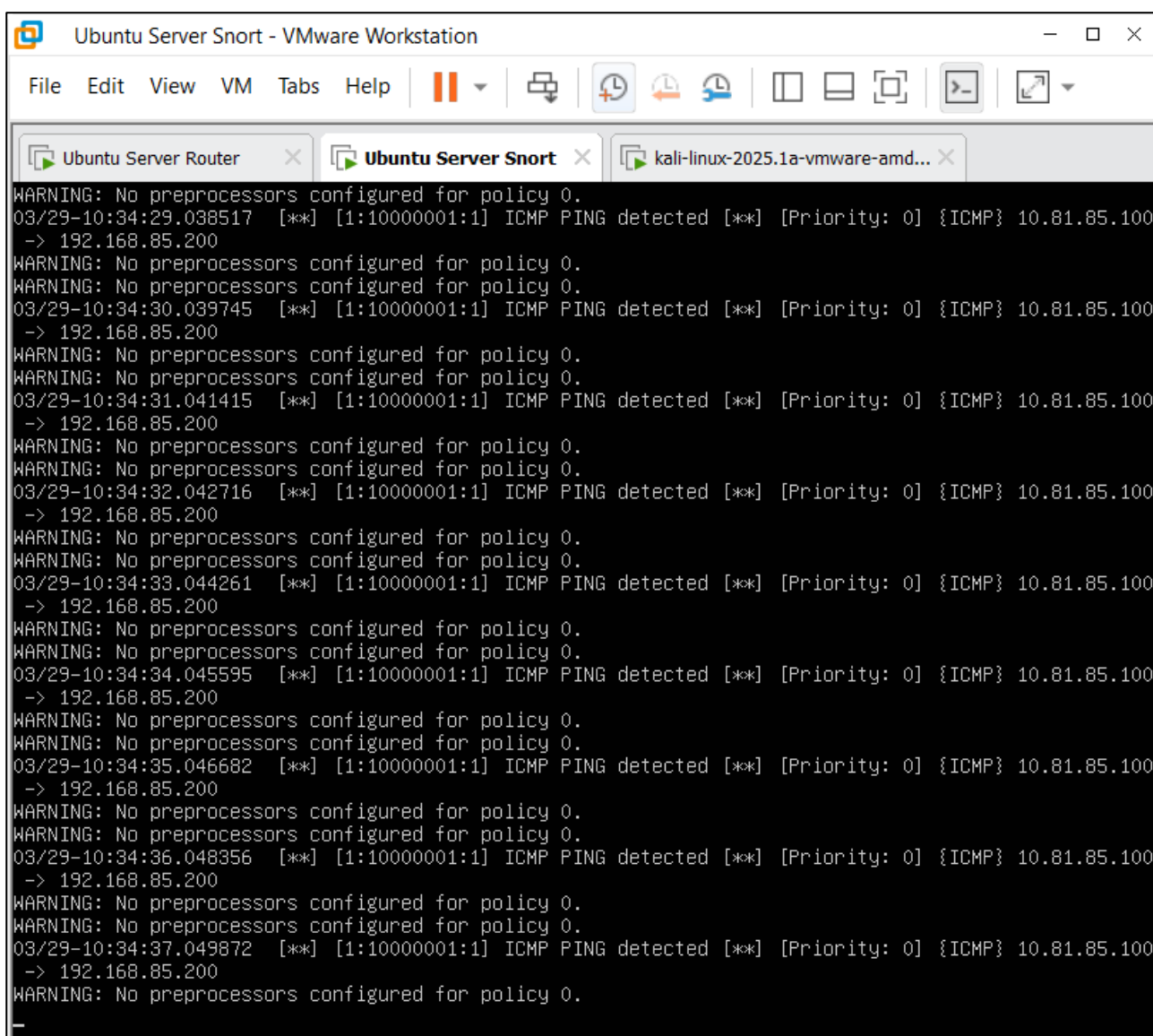
-*> Snort! <*-
Version 2.9.15.1 GRE (Build 15125)
By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
Copyright (C) 2014-2019 Cisco and/or its affiliates. All rights reserved.
Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using libpcap version 1.10.1 (with TPACKET_V3)
Using PCRE version: 8.39 2016-06-14
Using ZLIB version: 1.2.11

Commencing packet processing (pid=68333)
Decoding Ethernet
_
```

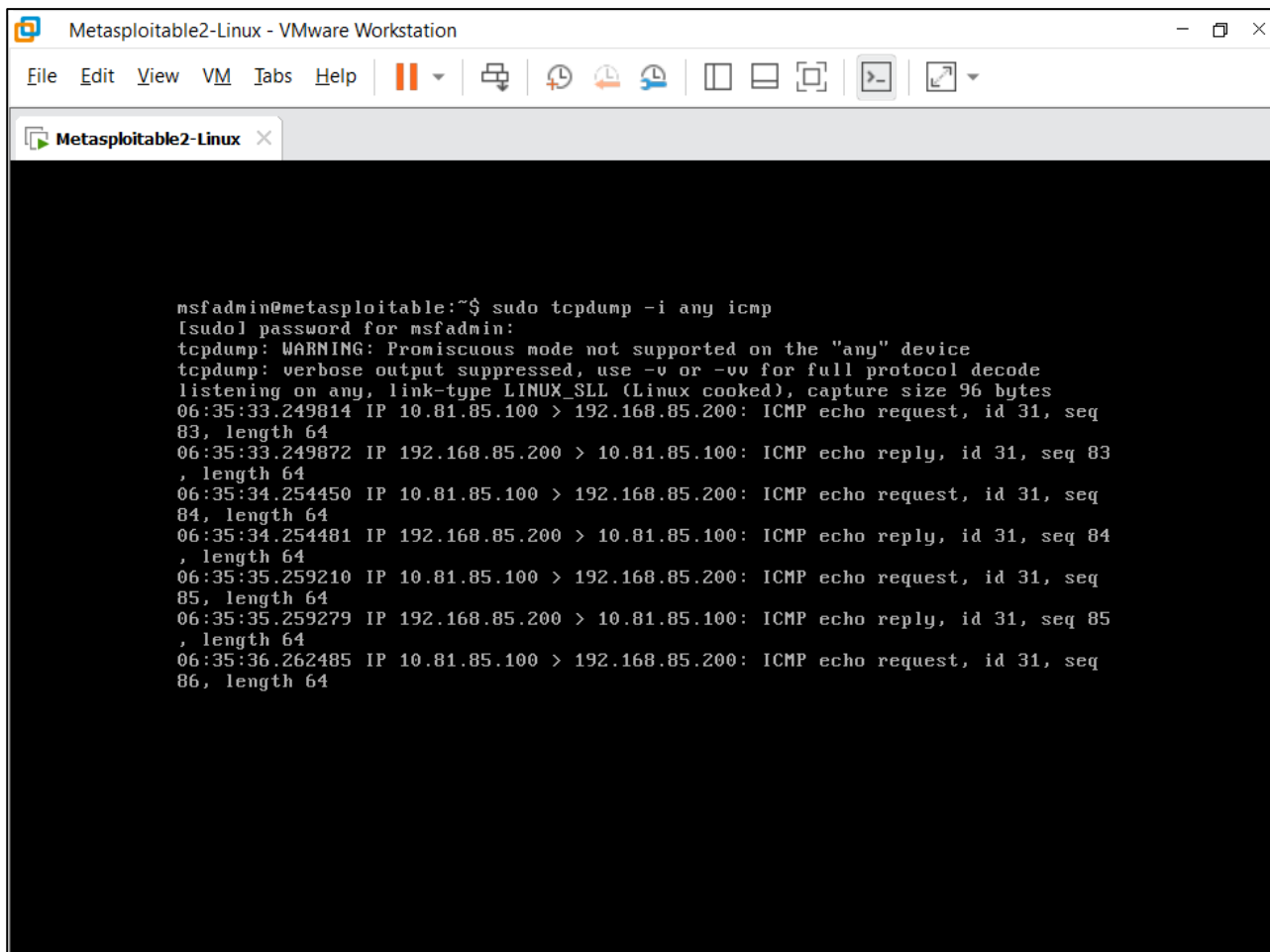
- Ping từ Attacker tới Victim



- Log của snort phát hiện ping từ Attacker:



- Dừng tcpdump trước khi áp dụng rule



The screenshot shows a terminal window titled "Metasploitable2-Linux - VMware Workstation". The terminal prompt is `msfadmin@metasploitable:~$`. The user enters `sudo tcpdump -i any icmp`. The terminal output shows the following:

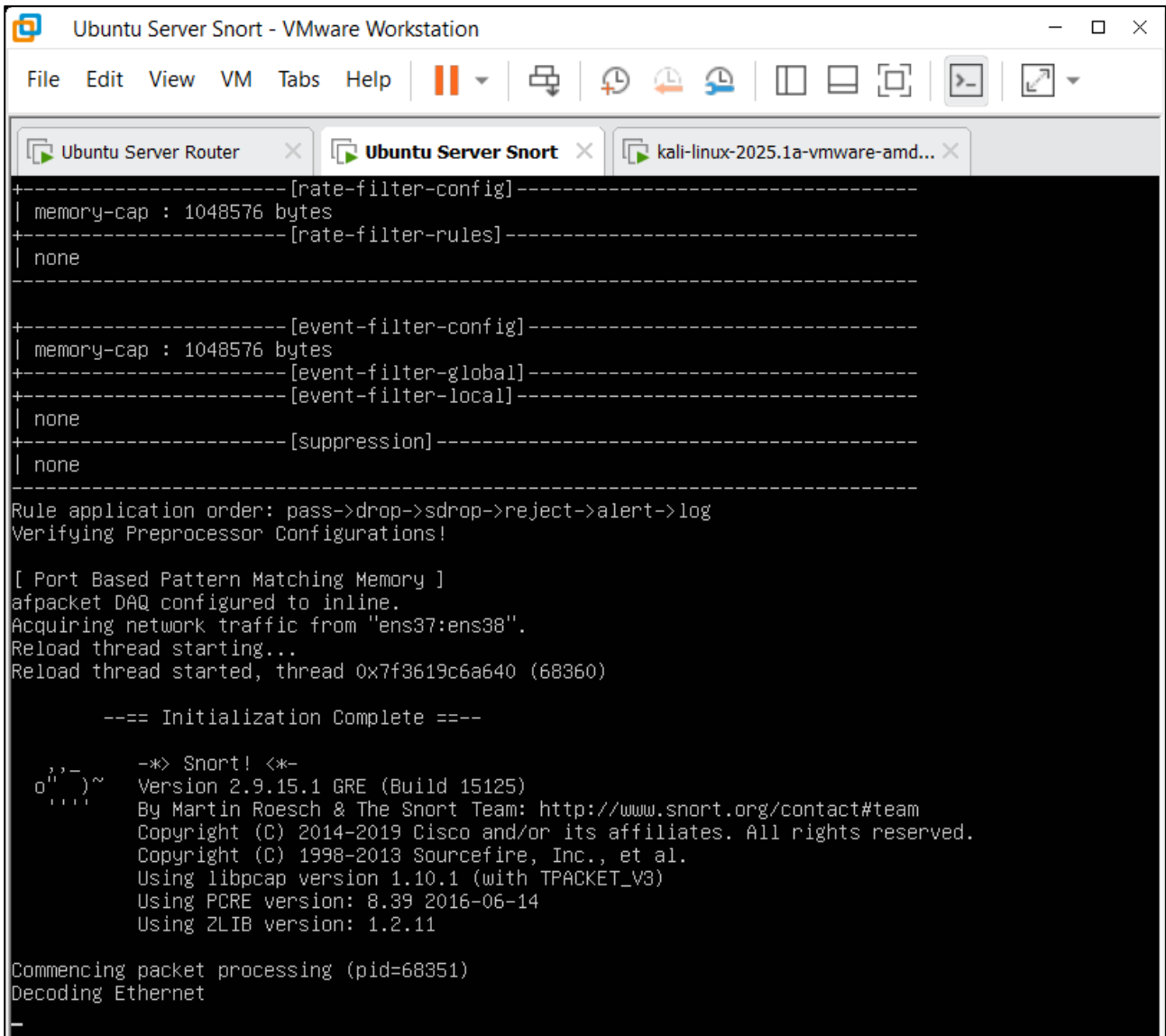
```
msfadmin@metasploitable:~$ sudo tcpdump -i any icmp
[sudo] password for msfadmin:
tcpdump: WARNING: Promiscuous mode not supported on the "any" device
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked), capture size 96 bytes
06:35:33.249814 IP 10.81.85.100 > 192.168.85.200: ICMP echo request, id 31, seq
83, length 64
06:35:33.249872 IP 192.168.85.200 > 10.81.85.100: ICMP echo reply, id 31, seq 83
, length 64
06:35:34.254450 IP 10.81.85.100 > 192.168.85.200: ICMP echo request, id 31, seq
84, length 64
06:35:34.254481 IP 192.168.85.200 > 10.81.85.100: ICMP echo reply, id 31, seq 84
, length 64
06:35:35.259210 IP 10.81.85.100 > 192.168.85.200: ICMP echo request, id 31, seq
85, length 64
06:35:35.259279 IP 192.168.85.200 > 10.81.85.100: ICMP echo reply, id 31, seq 85
, length 64
06:35:36.262485 IP 10.81.85.100 > 192.168.85.200: ICMP echo request, id 31, seq
86, length 64
```

***Sau khi áp dụng Rule #1:**

- Dừng rule chặn icmp:

```
root@snort:/# cat /etc/snort/rules/nhom9.rules
drop icmp any any -> 192.168.85.200 (msg: "Drop all ICMP traffic to 192.168.85.200"; GID:1; sid:10000001; rev:001;)
root@snort:/#
```

- Khởi chạy Snort:



```
-----[rate-filter-config]-----
| memory-cap : 1048576 bytes
+-----[rate-filter-rules]-----
| none
-----

+-----[event-filter-config]-----
| memory-cap : 1048576 bytes
+-----[event-filter-global]-----
+-----[event-filter-local]-----
| none
+-----[suppression]-----
| none
-----

Rule application order: pass->drop->sdrops->reject->alert->log
Verifying Preprocessor Configurations!

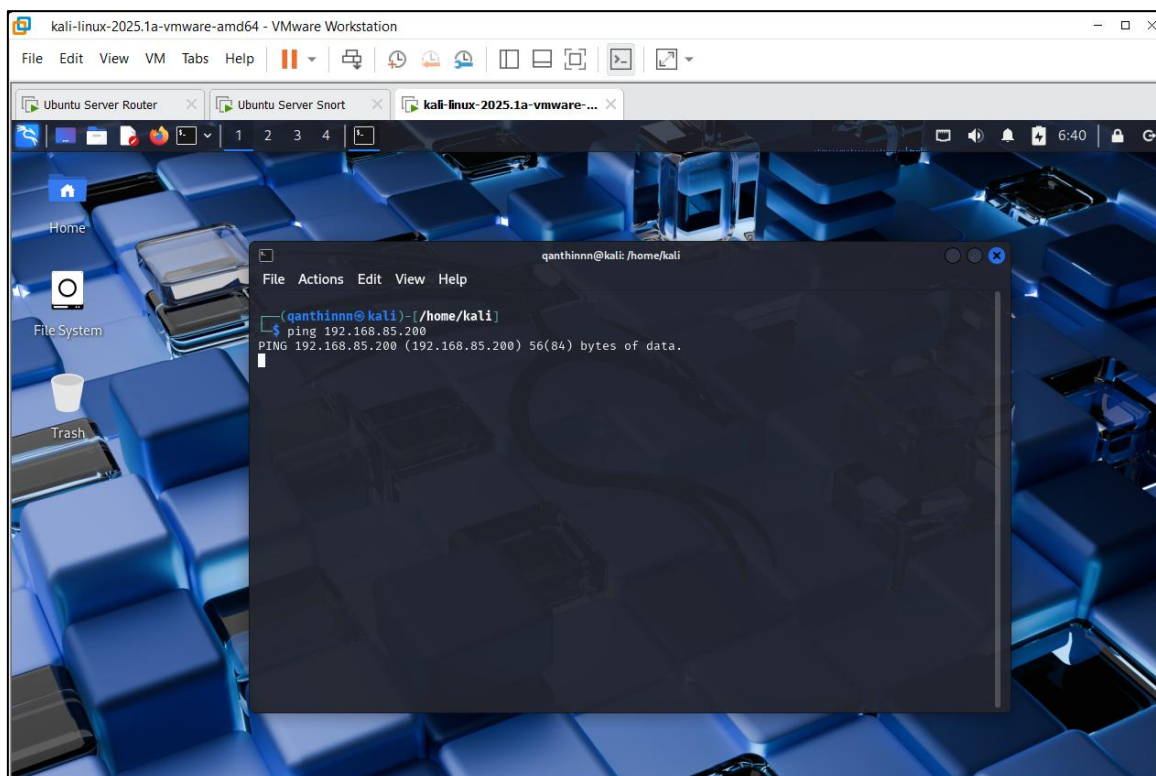
[ Port Based Pattern Matching Memory ]
afpacket DAQ configured to inline.
Acquiring network traffic from "ens37:ens38".
Reload thread starting...
Reload thread started, thread 0x7f3619c6a640 (68360)

==== Initialization Complete ====

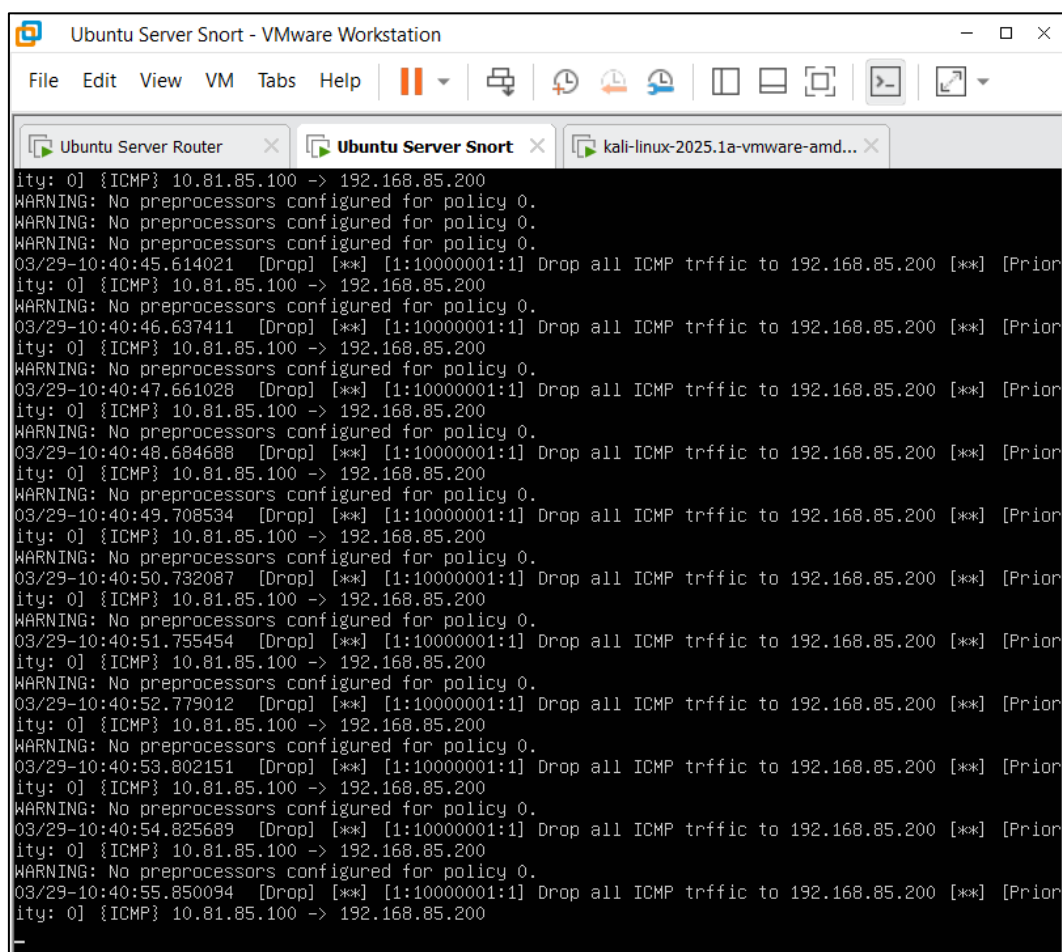
o''~
''''~
-*> Snort! <*-
Version 2.9.15.1 GRE (Build 15125)
By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
Copyright (C) 2014-2019 Cisco and/or its affiliates. All rights reserved.
Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using libpcap version 1.10.1 (with TPACKET_V3)
Using PCRE version: 8.39 2016-06-14
Using ZLIB version: 1.2.11

Commencing packet processing (pid=68351)
Decoding Ethernet
-
```

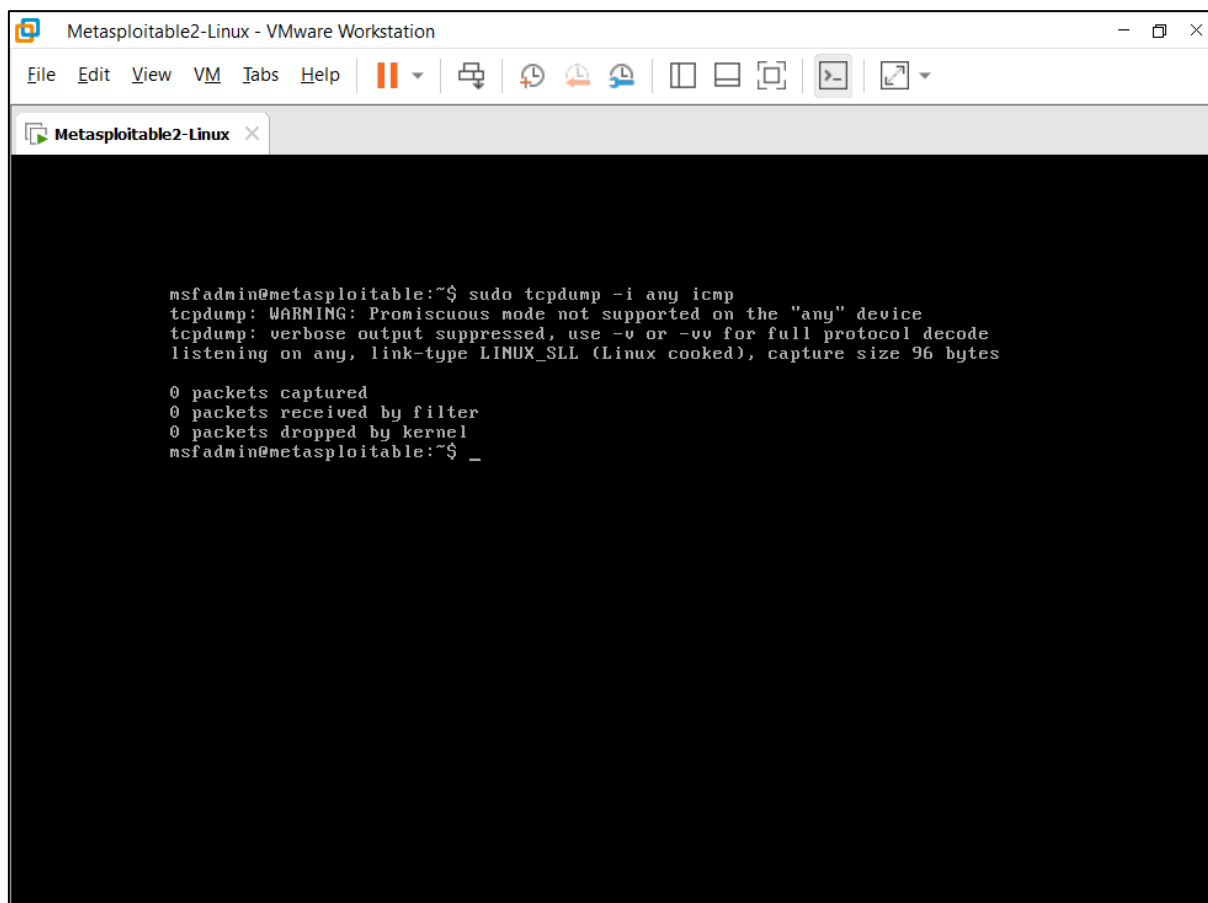

- Máy Attacker ping không được:



- Check log trên Snort, ta thấy các gói ICMP đã được chặn thành công:



- Dừng tcpdump và không phát hiện gói tin nào được gửi tới:



The screenshot shows a terminal window titled "Metasploitable2-Linux - VMware Workstation". The terminal output is as follows:

```
msfadmin@metasploitable:~$ sudo tcpdump -i any icmp
tcpdump: WARNING: Promiscuous mode not supported on the "any" device
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked), capture size 96 bytes

0 packets captured
0 packets received by filter
0 packets dropped by kernel
msfadmin@metasploitable:~$ _
```