

Portfolio Project

# **Packet Analysis and Simple DoS using Wireshark And Hping**

**Ranilo John M. Delos Angeles  
University of the East - Caloocan**

## I. Introduction

From learning the topology creations in Cisco Packet Tracer, I wanted to use a more complicated network simulator and so in this scenario, I simulated topology where I can access the computers through Virtual Machines and this was made possible by GNS3 and Virtual Box.

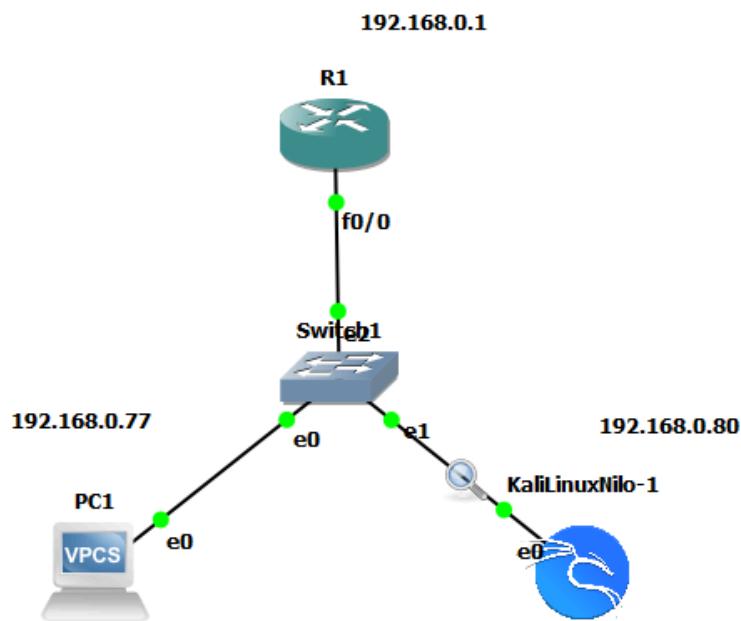
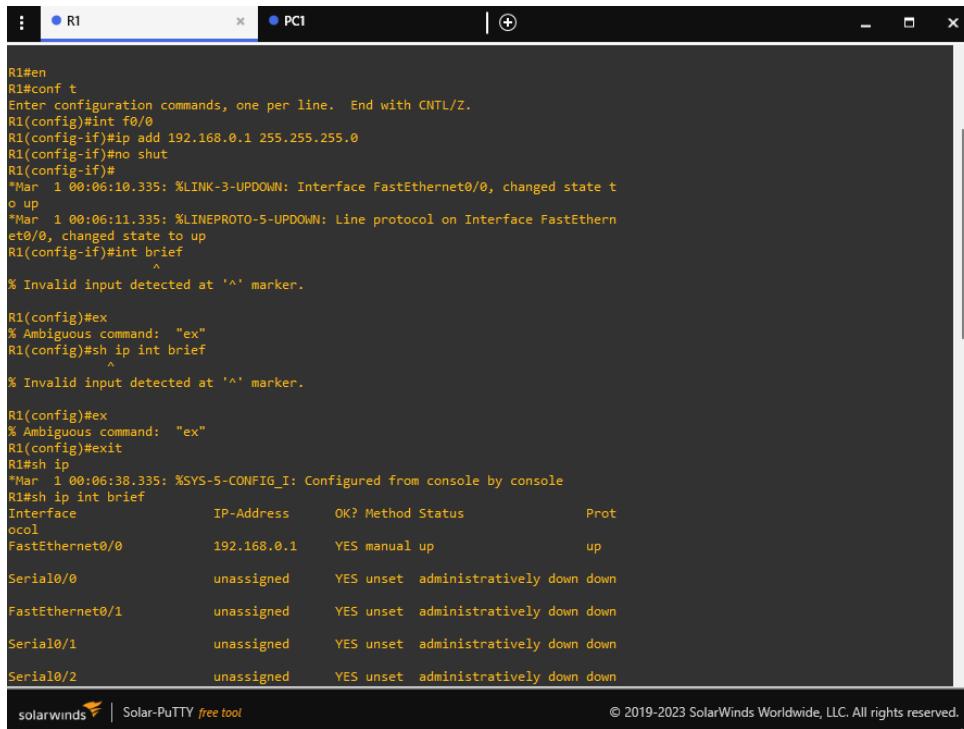


Figure 1.1 GNS3 Simple Topology

The devices I used was a Cisco Router C3745, a Simple Virtual PC, Ethernet Switch, and a Kali Linux simulated using VirtualBox.

## II. Configuration

I first started configuring the router, giving the switch its IP Address of 192.168.0.1  
255.255.255.0. It is an IP Address because I wanted to check first how I can properly  
configure and simulate the network. I also went to configure the PC1 with the IP  
Address of 192.168.0.77 and Kali Linux VM IP Address of 192.168.0.77



The screenshot shows a terminal window titled 'R1' with two tabs: 'R1' and 'PC1'. The 'R1' tab displays the following configuration commands:

```
R1#en
R1>conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#int f0/0
R1(config-if)#ip add 192.168.0.1 255.255.255.0
R1(config-if)#no shut
R1(config-if)#
*Mar 1 00:06:10.335: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar 1 00:06:11.335: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R1(config-if)#int brief
^
% Invalid input detected at '^' marker.

R1(config)#ex
% Ambiguous command: "ex"
R1(config)#sh ip int brief
^
% Invalid input detected at '^' marker.

R1(config)#x
% Ambiguous command: "ex"
R1(config)#exit
R1>sh ip
*Mar 1 00:06:38.335: %SYS-5-CONFIG_I: Configured from console by console
R1>sh ip int brief
Interface          IP-Address      OK? Method Status        Prot
ocel
FastEthernet0/0      192.168.0.1    YES manual up           up
Serial0/0            unassigned     YES unset administratively down down
FastEthernet0/1      unassigned     YES unset administratively down down
Serial0/1            unassigned     YES unset administratively down down
Serial0/2            unassigned     YES unset administratively down down
```

The 'PC1' tab is currently inactive.

Figure 2.1 Router IP Address

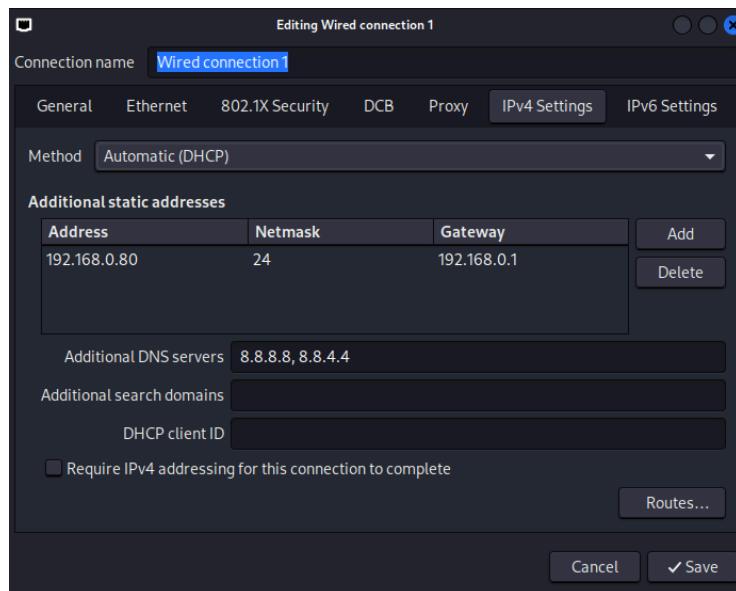


Figure 2.2 Kali Linux IP Address

### III. Simulation

So first from the VPCS, I went to ICMP ping the destination IP of 192.168.0.80 (Kali Linux PC) for error checking (to see if it works or not)

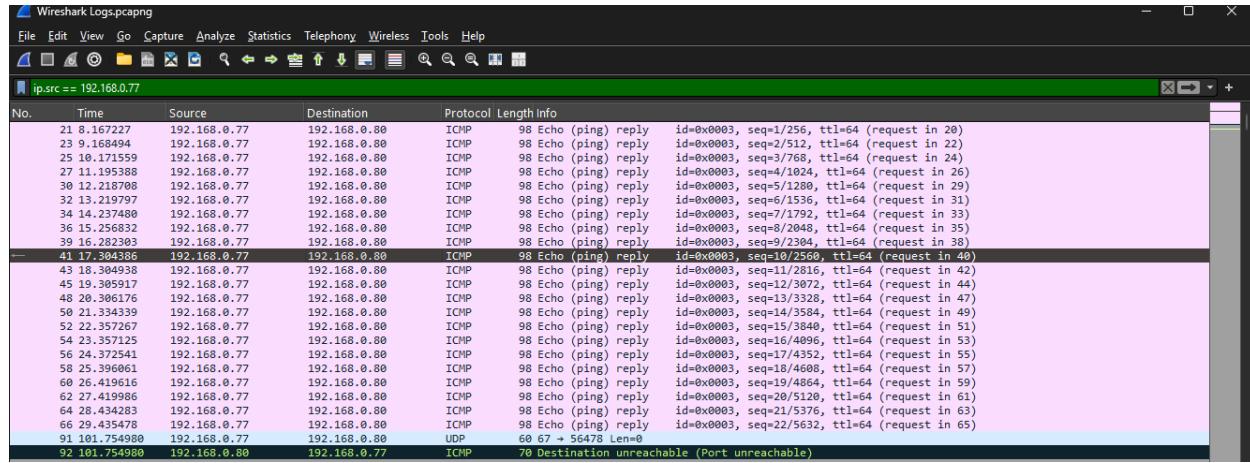


Figure 3.1 ICMPing VPCS to Kali Linux

I also did the same for Kali, I ICMP ping from Kali with IP Address of 192.168.0.80 to VPCS 192.168.0.77.

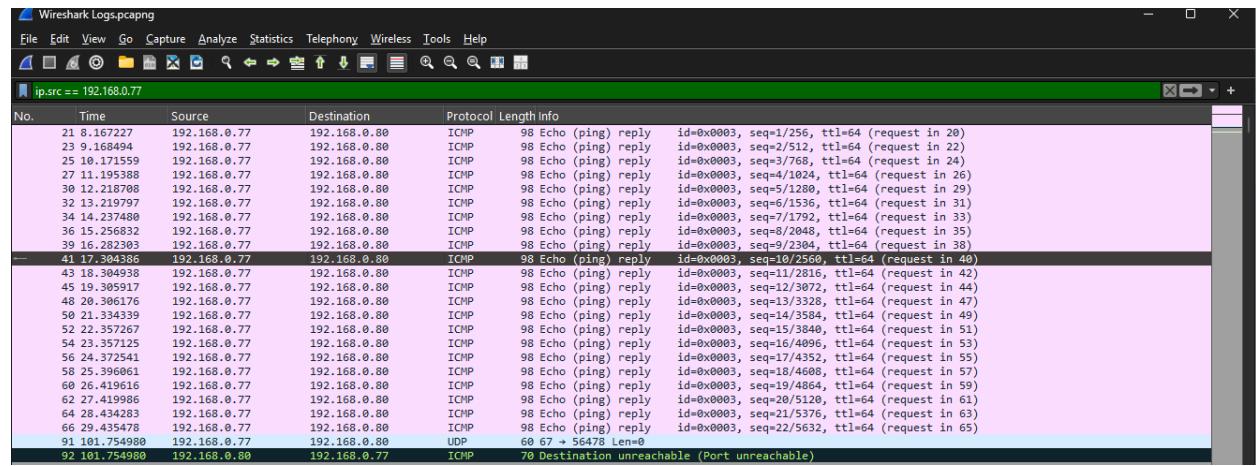


Figure 3.2 ICMPing Kali to VPCS

After that, I started doing nbtscan (netbios scan) on the network to check the ip addresses.

```
└─(root㉿kali)-[~/home/kali]
  # nbtscan -r 192.168.0.1/24
  Doing NBT name scan for addresses from 192.168.0.1/24

  IP address      NetBIOS Name      Server      User      MAC address
  ─────────────────────────────────────────────────────────────────────────────────
  192.168.0.80    <unknown>          <unknown>
  192.168.0.77    <unknown>          <unknown>
  ^C
```

Figure 3.3 NBT scan in the network.

Finishing the nbtscan, I went ahead and used the NMAP command for the scanning of the VPCS, the report below shows the open ports (All ports are open because the PC is unsecured) the -sS does this by getting a SYN/ACK reply and then gets to RST to not complete the three way handshake.

```
Session Actions Edit View Help
root@kali: /home/kali
└─(kali㉿kali)-[~]
  $ nmap -sS 192.168.0.77
  Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-02 08:25 EST
  Nmap scan report for 192.168.0.77
  Host is up (0.0016s latency).

  PORT      STATE      SERVICE
  1/tcp      open       tcpmux
  3/tcp      open       compressnet
  4/tcp      open       unknown
  6/tcp      open       unknown
  7/tcp      open       echo
  9/tcp      open       discard
  13/tcp     open       daytime
  17/tcp     open       qotd
  19/tcp     open       chargen
  20/tcp     open       ftp-data
  21/tcp     open       ftp
  22/tcp     open       ssh
  23/tcp     open       telnet
  24/tcp     open       priv-mail
  25/tcp     open       smtp
  26/tcp     open       rsftp
  30/tcp     open       unknown
  32/tcp     open       unknown
  33/tcp     open       dsp
  37/tcp     open       time
```

Figure 3.4 NMAP the ports to check which is open.

Here below, shows all the captured packet from the attacker's machine (Kali) the displayed packet are 2018. It shows all the ports from 1 to 65389. It also shows the MAC Address of the VPCS, 00:50:79:66:68:00.

```
root@kali: /home/kali
Session Actions Edit View Help
56737/tcp open    unknown
56738/tcp open    unknown
57294/tcp open    unknown
57797/tcp open    unknown
58080/tcp open    unknown
60020/tcp open    unknown
60443/tcp open    unknown
61532/tcp open    unknown
61900/tcp open    unknown
62078/tcp open    iphone-sync
63331/tcp open    unknown
64623/tcp open    unknown
64680/tcp open    unknown
65000/tcp open    unknown
65129/tcp open    unknown
65389/tcp open    unknown
MAC Address: 00:50:79:66:68:00 (Private)
```

Figure 3.5 NMAP scan finished.

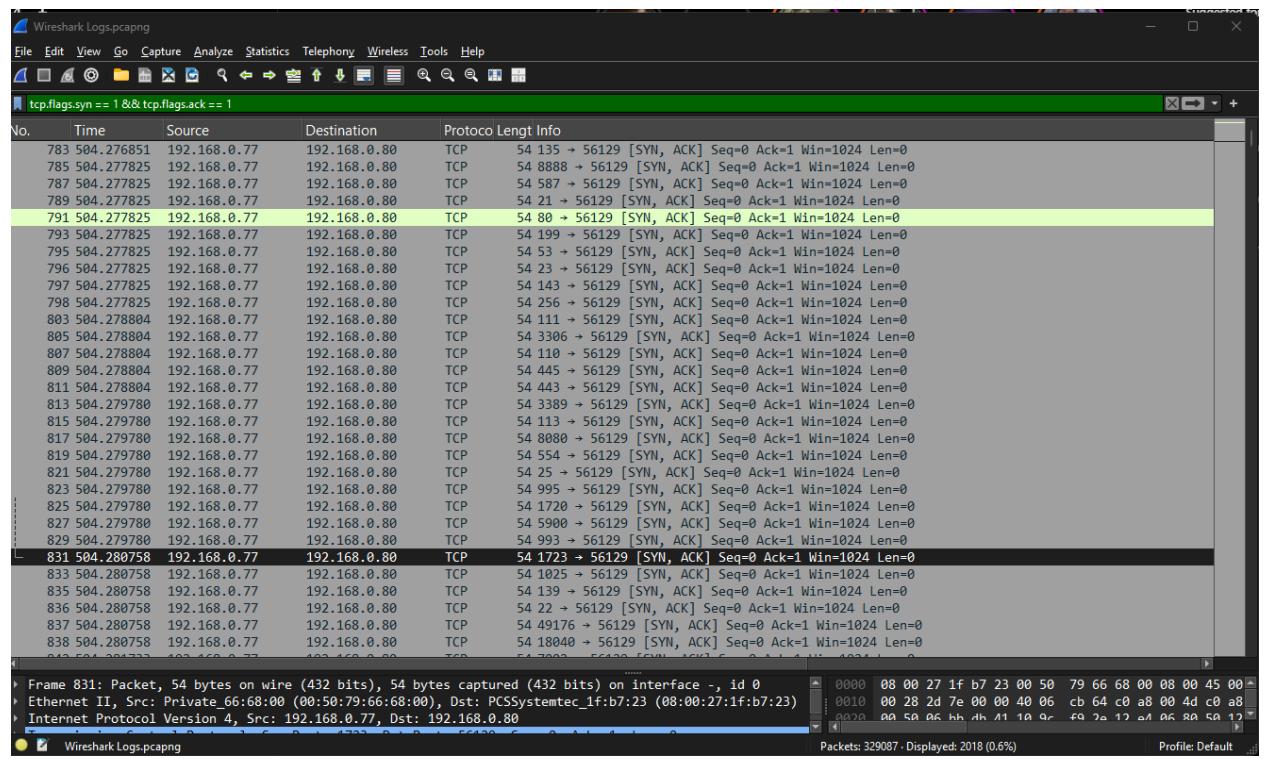


Figure 3.6 Packets captured from NMAP scanning from Port 1 to 65389.

Now since I'm curious about Denial of Service, I tried flooding the VPCS with command hping3 –icmp –flood 192.168.0.77, As shown below, there's about 245665 packets transmitted to the other end device it also did put the PC into a session timeout due to overwhelming amounts of ICMP.

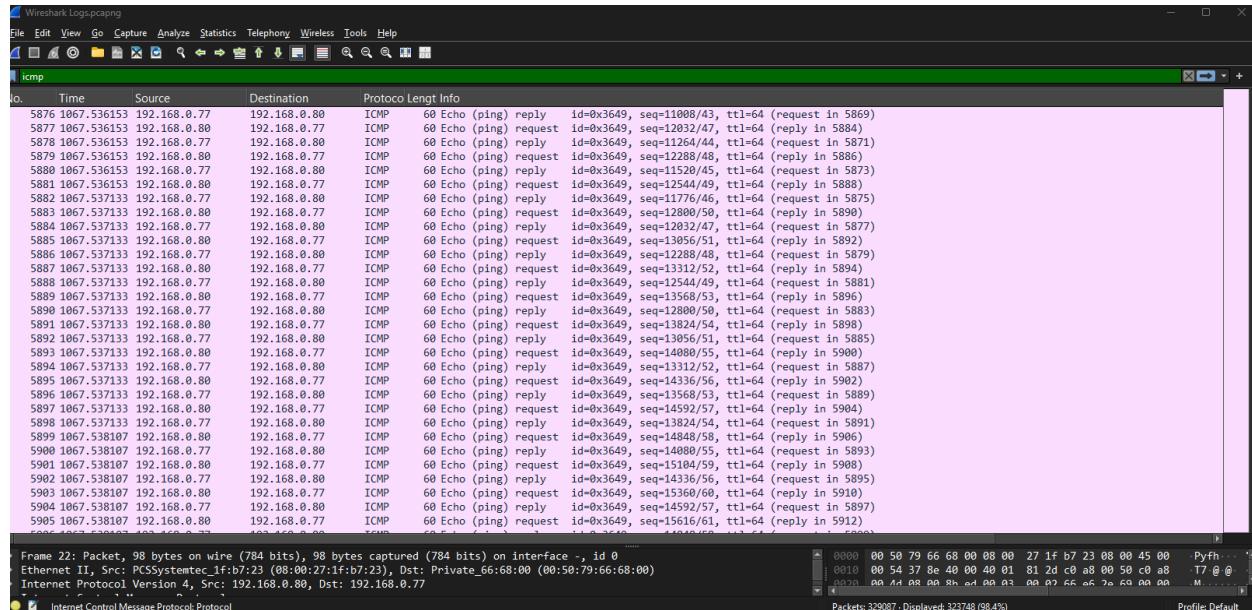


Figure 3.7 Packet Capture of DoS ICMP Flooding

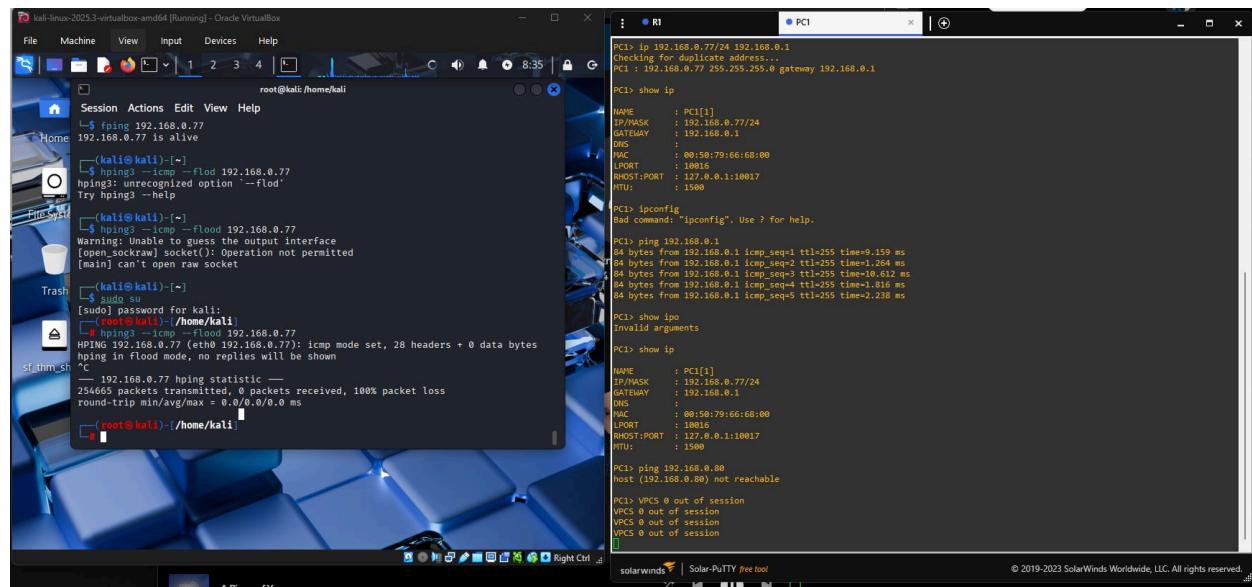


Figure 3.8 Kali command and transmitted packets and putting VPCS down.

## **IV. Conclusion**

This has been a fun experiment to play with, because now I can finally simulate penetration testing via Kali Linux, but for this kind of simulation I want to explore more on how I will defend organizations via IDS and IPS.