Lamar Roulhac

ICMP Traffic Analysis

Objectives

The objectives of this lab were to expand a virtual network by adding a Windows 10 virtual machine, configure the new VMs. install Wireshark on the Linux VMs, and analyze captured ICMP IPv4 and IPv6 network traffic.

Background

Networking is a critical component of modern computing environments, allowing multiple devices to communicate over local and wide-area networks. Virtual machines (VMs) are often used to simulate networked environments for testing and analysis.

Wireshark is a packet analyzer used for network troubleshooting and analysis. ICMP (Internet Control Message Protocol) is a fundamental protocol used for network diagnostics, such as the `ping` command. In this lab, we focus on capturing and analyzing ICMPv4 and ICMPv6 packets within a controlled virtual network environment.

Procedures

Task 1: Expanding the Virtual Environment

Acquired the Windows 10 ISO from the Canvas and installed Windows 10 as a new VM with 2 GBs of RAM, 2 CPU cores, 60 GBs of storage and NAT Networking adapter. The rest of the lab will be conducted on two Linux VMs.

I then powered my two VMs on and checked to see if they could communicate with each other, which they successfully did.

The assigned IP addresses:

Linux VM 1 IP: 192.168.192.128

Linux VM 2 IP: 192,168.192.129

I then installed Wireshark on my Linux VM 1 since i designated that to capture the traffic.

Task 2: Capturing ICMP Traffic

I opened Wireshark and started a packet capture on the relevant network interface. I then used the ping command on the Linux VM 2 to Generated ICMPv4 traffic as seen in **Figure 1**. After a few I saved the capture file with the packets to examine you can see in **Figure 2.**

```
lamar@lamar-VMware-Virtual-Platform:~$ ping -c 5 192.168.192.128
PING 192.168.192.128 (192.168.192.128) 56(84) bytes of data.
64 bytes from 192.168.192.128: icmp_seq=1 ttl=64 time=0.274 ms
64 bytes from 192.168.192.128: icmp_seq=2 ttl=64 time=0.182 ms
64 bytes from 192.168.192.128: icmp_seq=3 ttl=64 time=0.199 ms
64 bytes from 192.168.192.128: icmp_seq=4 ttl=64 time=0.168 ms
64 bytes from 192.168.192.128: icmp_seq=5 ttl=64 time=0.160 ms

--- 192.168.192.128 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4124ms
rtt min/avg/max/mdev = 0.160/0.196/0.274/0.040 ms
```

Figure 1: ICMPv4 Ping

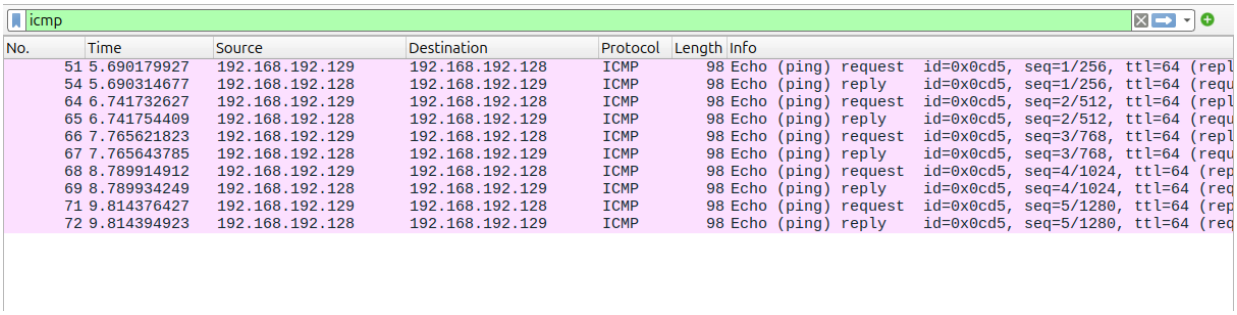

Figure 2: ICMPv4 Traffic Capture

I then used the ping command to create exclusively ICMPv6 traffic as seen in **Figure 3**. and again after a few I stopped the capture and saved the file to see the traffic captured as seen in **Figure 4**.

```
lamar@lamar-VMware-Virtual-Platform:~$ ping6 -c 5 fe80::20c:29ff:fe32:3bc5%ens33
PING fe80::20c:29ff:fe32:3bc5%ens33 (fe80::20c:29ff:fe32:3bc5%ens33) 56 data byt
es
64 bytes from fe80::20c:29ff:fe32:3bc5%ens33: icmp_seq=1 ttl=64 time=0.299 ms
64 bytes from fe80::20c:29ff:fe32:3bc5%ens33: icmp_seq=2 ttl=64 time=0.153 ms
64 bytes from fe80::20c:29ff:fe32:3bc5%ens33: icmp_seq=3 ttl=64 time=0.180 ms
64 bytes from fe80::20c:29ff:fe32:3bc5%ens33: icmp_seq=4 ttl=64 time=0.180 ms
64 bytes from fe80::20c:29ff:fe32:3bc5%ens33: icmp_seq=5 ttl=64 time=0.156 ms

--- fe80::20c:29ff:fe32:3bc5%ens33 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4108ms
rtt min/avg/max/mdev = 0.153/0.193/0.299/0.053 ms
```

Figure 3: ICMPv6 Ping

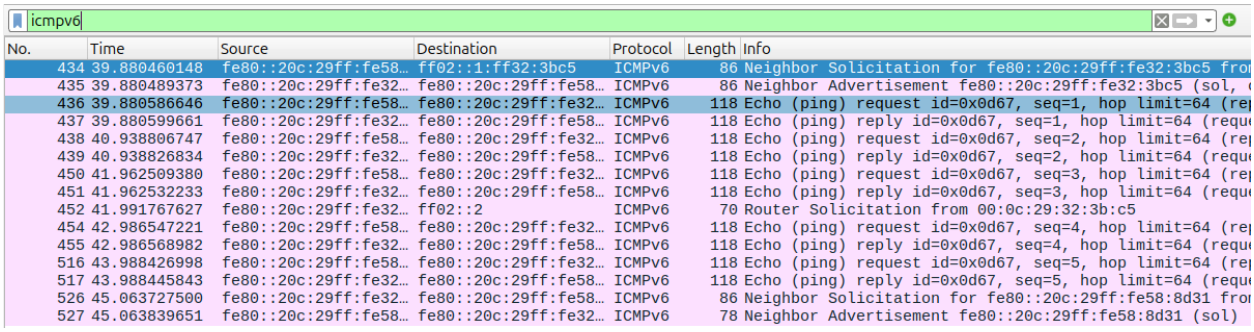

Figure 4: ICMPv6 Traffic Capture

Task 3: Analyzing Packet Captures

With both of the pings done I analyzed the traffic captured in both files starting with ICMPv4 traffic as seen in **Figure 5** you can see the Source Ip and Destination Ip, and the ICMP type.

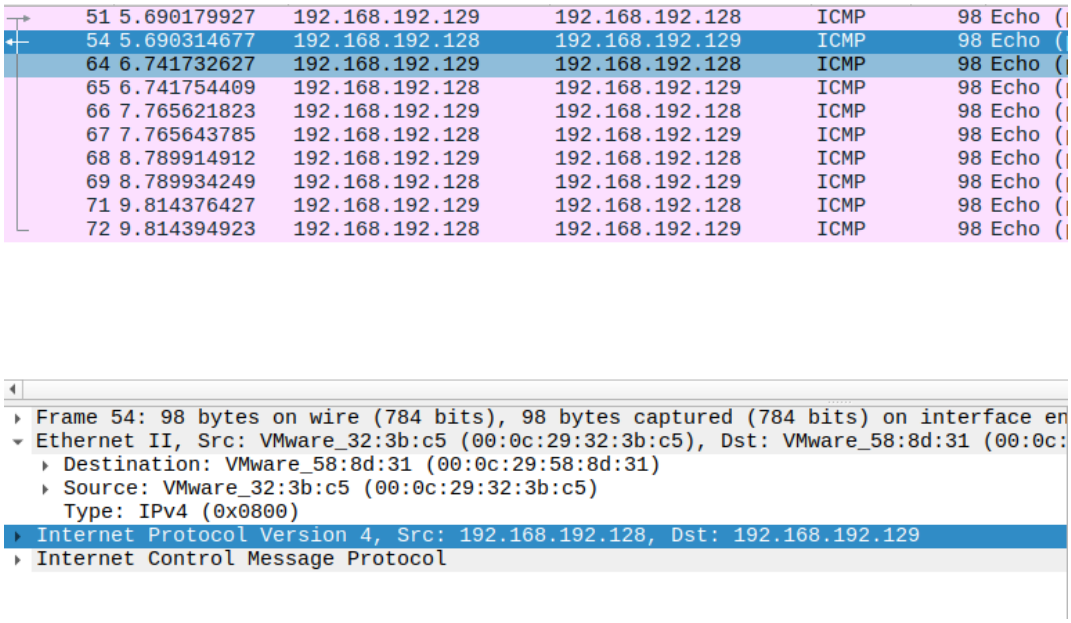

Figure 5: iCMPv4 information

The ICMPv6 traffic had a little more information, the echo pings which had the source and destination Ips (**Figure 6 and Figure 7**) it also had Neighbor Solicitation messages as seen in **Figure 8** you can see the type and code of it.

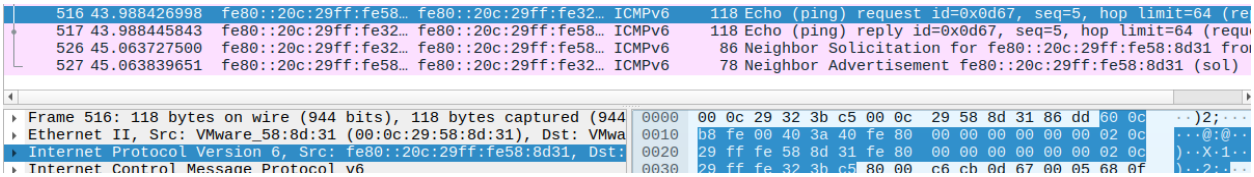

Figure 6: Source IP

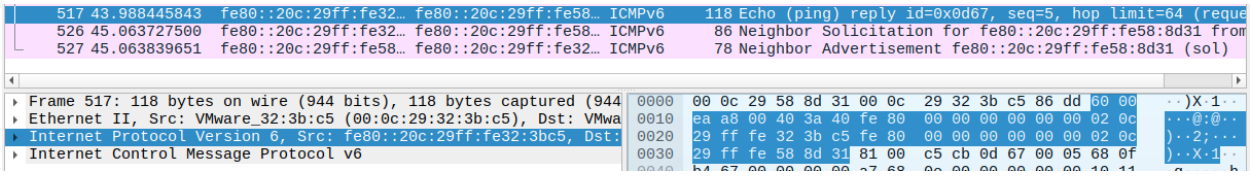

Figure 7: Destination IP

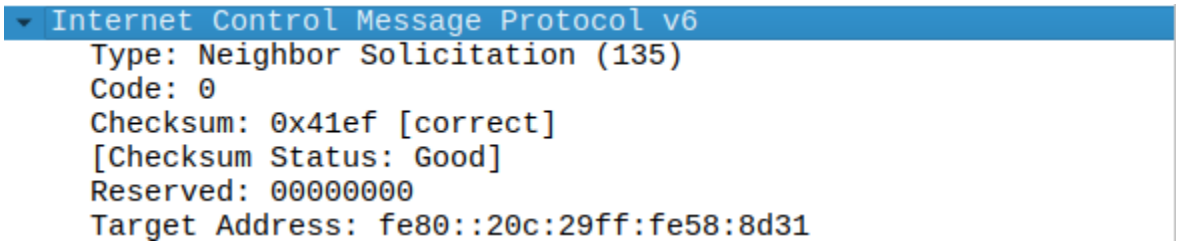

Figure 8: Neighbor Solicitation Message Details

Observations:

- The ICMPv4 packets were in the format of an Echo Request and Echo Reply exchange.
- The ICMPv6 packets included Neighbor Solicitation and Advertisement messages, which IPv6-specific communication since it does not come up with the ICMPv4 traffic.
- The captured packets confirmed successful communication between the virtual machines.

Conclusion

This lab I successfully added a Windows 10 workstation and verified the network connectivity between the Linux virtual machines. Using Wireshark, ICMPv4 and ICMPv6 traffic was captured and analyzed, which allowed me to see the way network communication and protocols are used.