

**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

icmp									
No.	Time	Source	Destination	Protocol	Length	Info			
51	5.690179927	192.168.192.129	192.168.192.128	ICMP	98	Echo (ping) request	id=0x0cd5, seq=1/256, ttl=64	(repl	
54	5.690314677	192.168.192.128	192.168.192.129	ICMP	98	Echo (ping) reply	id=0x0cd5, seq=1/256, ttl=64	(requ	
64	6.741732627	192.168.192.129	192.168.192.128	ICMP	98	Echo (ping) request	id=0x0cd5, seq=2/512, ttl=64	(repl	
65	6.741754409	192.168.192.128	192.168.192.129	ICMP	98	Echo (ping) reply	id=0x0cd5, seq=2/512, ttl=64	(requ	
66	7.765621823	192.168.192.129	192.168.192.128	ICMP	98	Echo (ping) request	id=0x0cd5, seq=3/768, ttl=64	(repl	
67	7.765643785	192.168.192.128	192.168.192.129	ICMP	98	Echo (ping) reply	id=0x0cd5, seq=3/768, ttl=64	(requ	
68	8.789914912	192.168.192.129	192.168.192.128	ICMP	98	Echo (ping) request	id=0x0cd5, seq=4/1024, ttl=64	(rep	
69	8.789934249	192.168.192.128	192.168.192.129	ICMP	98	Echo (ping) reply	id=0x0cd5, seq=4/1024, ttl=64	(req	
71	9.814376427	192.168.192.129	192.168.192.128	ICMP	98	Echo (ping) request	id=0x0cd5, seq=5/1280, ttl=64	(rep	
72	9.814394923	192.168.192.128	192.168.192.129	ICMP	98	Echo (ping) reply	id=0x0cd5, seq=5/1280, ttl=64	(req	

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

No.	Time	Source	Destination	Protocol	Length	Info
434	39.880460148	fe80::20c:29ff:fe58...	ff02::1:ff32:3bc5	ICMPv6	86	Neighbor Solicitation for fe80::20c:29ff:fe32:3bc5 from
435	39.880489373	fe80::20c:29ff:fe32...	fe80::20c:29ff:fe58...	ICMPv6	86	Neighbor Advertisement fe80::20c:29ff:fe32:3bc5 (sol,
436	39.880586646	fe80::20c:29ff:fe32...	fe80::20c:29ff:fe32...	ICMPv6	118	Echo (ping) request id=0x0d67, seq=1, hop limit=64 (req
437	39.880599661	fe80::20c:29ff:fe32...	fe80::20c:29ff:fe58...	ICMPv6	118	Echo (ping) reply id=0x0d67, seq=1, hop limit=64 (requ
438	40.938806747	fe80::20c:29ff:fe58...	fe80::20c:29ff:fe32...	ICMPv6	118	Echo (ping) request id=0x0d67, seq=2, hop limit=64 (req
439	40.938826834	fe80::20c:29ff:fe32...	fe80::20c:29ff:fe58...	ICMPv6	118	Echo (ping) reply id=0x0d67, seq=2, hop limit=64 (requ
450	41.962509380	fe80::20c:29ff:fe58...	fe80::20c:29ff:fe32...	ICMPv6	118	Echo (ping) request id=0x0d67, seq=3, hop limit=64 (req
451	41.962532233	fe80::20c:29ff:fe32...	fe80::20c:29ff:fe58...	ICMPv6	118	Echo (ping) reply id=0x0d67, seq=3, hop limit=64 (requ
452	41.991767627	fe80::20c:29ff:fe32...	ff02::2	ICMPv6	70	Router Solicitation from 00:0c:29:32:3b:c5
454	42.986547221	fe80::20c:29ff:fe58...	fe80::20c:29ff:fe32...	ICMPv6	118	Echo (ping) request id=0x0d67, seq=4, hop limit=64 (req
455	42.986568982	fe80::20c:29ff:fe32...	fe80::20c:29ff:fe58...	ICMPv6	118	Echo (ping) reply id=0x0d67, seq=4, hop limit=64 (requ
516	43.988426998	fe80::20c:29ff:fe58...	fe80::20c:29ff:fe32...	ICMPv6	118	Echo (ping) request id=0x0d67, seq=5, hop limit=64 (req
517	43.988445843	fe80::20c:29ff:fe32...	fe80::20c:29ff:fe58...	ICMPv6	118	Echo (ping) reply id=0x0d67, seq=5, hop limit=64 (requ
526	45.063727500	fe80::20c:29ff:fe32...	fe80::20c:29ff:fe58...	ICMPv6	86	Neighbor Solicitation for fe80::20c:29ff:fe58:8d31 from
527	45.063839651	fe80::20c:29ff:fe58...	fe80::20c:29ff:fe32...	ICMPv6	78	Neighbor Advertisement fe80::20c:29ff:fe58:8d31 (sol)

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.

51	5.690179927	192.168.192.129	192.168.192.128	ICMP	98 Echo (I
54	5.690314677	192.168.192.128	192.168.192.129	ICMP	98 Echo (I
64	6.741732627	192.168.192.129	192.168.192.128	ICMP	98 Echo (I
65	6.741754409	192.168.192.128	192.168.192.129	ICMP	98 Echo (I
66	7.765621823	192.168.192.129	192.168.192.128	ICMP	98 Echo (I
67	7.765643785	192.168.192.128	192.168.192.129	ICMP	98 Echo (I
68	8.789914912	192.168.192.129	192.168.192.128	ICMP	98 Echo (I
69	8.789934249	192.168.192.128	192.168.192.129	ICMP	98 Echo (I
71	9.814376427	192.168.192.129	192.168.192.128	ICMP	98 Echo (I
72	9.814394923	192.168.192.128	192.168.192.129	ICMP	98 Echo (I

▶ Frame 54: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface en
▼ Ethernet II, Src: VMware_32:3b:c5 (00:0c:29:32:3b:c5), Dst: VMware_58:8d:31 (00:0c:
▶ Destination: VMware_58:8d:31 (00:0c:29:58:8d:31)
▶ Source: VMware_32:3b:c5 (00:0c:29:32:3b:c5)
Type: IPv4 (0x0800)
▶ Internet Protocol Version 4, Src: 192.168.192.128, Dst: 192.168.192.129
▶ Internet Control Message Protocol

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.

516	43.988426998	fe80::20c:29ff:fe58...	fe80::20c:29ff:fe32...	ICMPv6	118 Echo (ping) request id=0x0d67, seq=5, hop limit=64 (re
517	43.988445843	fe80::20c:29ff:fe32...	fe80::20c:29ff:fe58...	ICMPv6	118 Echo (ping) reply id=0x0d67, seq=5, hop limit=64 (reque
526	45.063727500	fe80::20c:29ff:fe32...	fe80::20c:29ff:fe58...	ICMPv6	86 Neighbor Solicitation for fe80::20c:29ff:fe58:8d31 fro
527	45.063839651	fe80::20c:29ff:fe58...	fe80::20c:29ff:fe32...	ICMPv6	78 Neighbor Advertisement fe80::20c:29ff:fe58:8d31 (sol)

▶ Frame 516: 118 bytes on wire (944 bits), 118 bytes captured (944
▶ Ethernet II, Src: VMware_58:8d:31 (00:0c:29:58:8d:31), Dst: VMwa
▶ Internet Protocol Version 6, Src: fe80::20c:29ff:fe58:8d31, Dst:
▶ Internet Control Message Protocol v6

Figure 6: Source IP

517	43.988445843	fe80::20c:29ff:fe32...	fe80::20c:29ff:fe58...	ICMPv6	118 Echo (ping) reply id=0x0d67, seq=5, hop limit=64 (reque
526	45.063727500	fe80::20c:29ff:fe32...	fe80::20c:29ff:fe58...	ICMPv6	86 Neighbor Solicitation for fe80::20c:29ff:fe58:8d31 from
527	45.063839651	fe80::20c:29ff:fe58...	fe80::20c:29ff:fe32...	ICMPv6	78 Neighbor Advertisement fe80::20c:29ff:fe58:8d31 (sol)

▶ Frame 517: 118 bytes on wire (944 bits), 118 bytes captured (944
▶ Ethernet II, Src: VMware_32:3b:c5 (00:0c:29:32:3b:c5), Dst: VMwa
▶ Internet Protocol Version 6, Src: fe80::20c:29ff:fe32:3bc5, Dst:
▶ Internet Control Message Protocol v6

Figure 7: Destination IP

▼ Internet Control Message Protocol v6
Type: Neighbor Solicitation (135)
Code: 0
Checksum: 0x41ef [correct]
[Checksum Status: Good]
Reserved: 00000000
Target Address: fe80::20c:29ff:fe58:8d31

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 are 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.