Week 1: Assignment 2: Using Wireshark to Examine Network Traffic

Report by: Tonny Odhiambo, CS-CNS06-24028

Introduction

Wireshark is a software protocol analyser, or "packet sniffer" application, used for network troubleshooting, analysis, software and protocol development, and education. As data streams travel back and forth over the network, the sniffer "captures" each protocol data unit (PDU) and can decode and analyse its content according to the appropriate RFC or other specifications.

Wireshark is a useful tool for anyone working with networks and can be used with most labs in the CCNA courses for data analysis and troubleshooting. In this lab, you will use Wireshark to capture ICMP data packet IP addresses and Ethernet frame MAC addresses.

Objectives

Part 1: Capture and Analyse Local ICMP Data in Wireshark

Part 2: Capture and Analyse Remote ICMP Data in Wireshark

Methodology

Part 1: Capture and Analysis of Local ICMP Data in Wireshark

Step 1: Retrieving PC interface addresses.

I retrieved my pc IP address and its network interface card (NIC) physical address using the **ipconfig /all** command on the command prompt window.

```
Microsoft Windows [Version 10.0.22621.3527]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Tyano>ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : DESKTOP-M0NG1SF
   Primary Dns Suffix  . . . . . . . :
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No

Ethernet adapter Ethernet:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel(R) Ethernet Connection (10) I219-LM
   Physical Address. . . . . . . . . : 48-2A-E3-A1-F0-77
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
```

Step 2: Start Wireshark and begin capturing data.

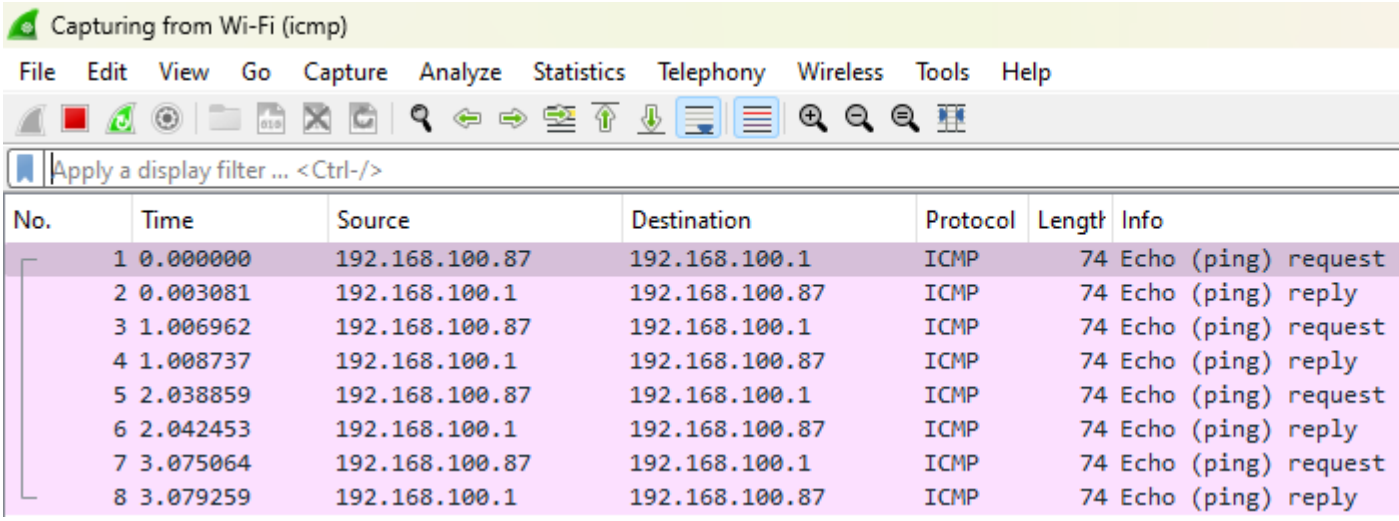
In this step, I navigated to wireshark and clicked the desired interface to start packet capture.

I applied the **icmp** filter to view only the ICMP (ping) PDUS.

I then pinged the IP address that I received from another pc.

```
C:\Users\Tyano>ping 192.168.100.1

Pinging 192.168.100.1 with 32 bytes of data:
Reply from 192.168.100.1: bytes=32 time=3ms TTL=64
Reply from 192.168.100.1: bytes=32 time=2ms TTL=64
Reply from 192.168.100.1: bytes=32 time=3ms TTL=64
Reply from 192.168.100.1: bytes=32 time=4ms TTL=64

Ping statistics for 192.168.100.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 4ms, Average = 3ms
```

Immediately after the ping, data starts appearing in the top window of Wireshark again.

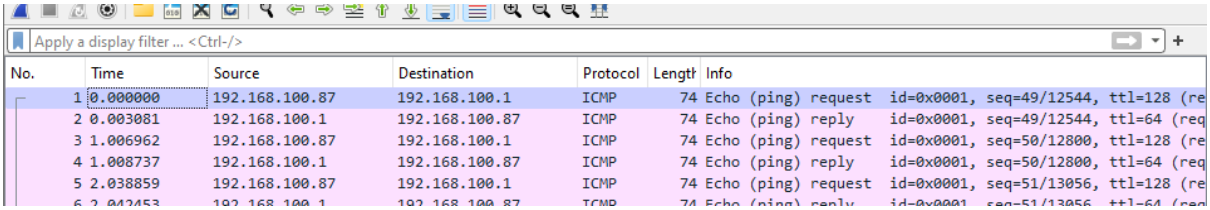


No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.100.87	192.168.100.1	ICMP	74	Echo (ping) request
2	0.003081	192.168.100.1	192.168.100.87	ICMP	74	Echo (ping) reply
3	1.006962	192.168.100.87	192.168.100.1	ICMP	74	Echo (ping) request
4	1.008737	192.168.100.1	192.168.100.87	ICMP	74	Echo (ping) reply
5	2.038859	192.168.100.87	192.168.100.1	ICMP	74	Echo (ping) request
6	2.042453	192.168.100.1	192.168.100.87	ICMP	74	Echo (ping) reply
7	3.075064	192.168.100.87	192.168.100.1	ICMP	74	Echo (ping) request
8	3.079259	192.168.100.1	192.168.100.87	ICMP	74	Echo (ping) reply

I then stopped capturing data so that I could begin examining the captured data.

Step 3: Examining the captured data.

On clicking the first request, the **source** column has my PC IP address, and the **destination** column has the IP address of the other PC I pinged.



On further examination, the **source** MAC address matches that of my PC interface and the **destination** MAC address in Wireshark matches that of the Pc which I pinged.

```
> Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{2DFFA
✔ Ethernet II, Src: Intel_da:aa:48 (04:6c:59:da:aa:48), Dst: HuaweiTechno_a8:f1:95 (c8:b6:d3:a8:f1:9
    > Destination: HuaweiTechno_a8:f1:95 (c8:b6:d3:a8:f1:95)
    > Source: Intel_da:aa:48 (04:6c:59:da:aa:48)
      Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 192.168.100.87, Dst: 192.168.100.1
> Internet Control Message Protocol
```

The MAC address of the pinged PC is obtained through the Address Resolution Protocol (ARP). When your PC wants to communicate with another device on the local network, it sends out an ARP request asking for the MAC address corresponding to the IP address it wants to ping. The device with that IP address responds with its MAC address, which your PC then stores for future reference. This dynamic process allows devices on the same local network to discover each other's MAC addresses.

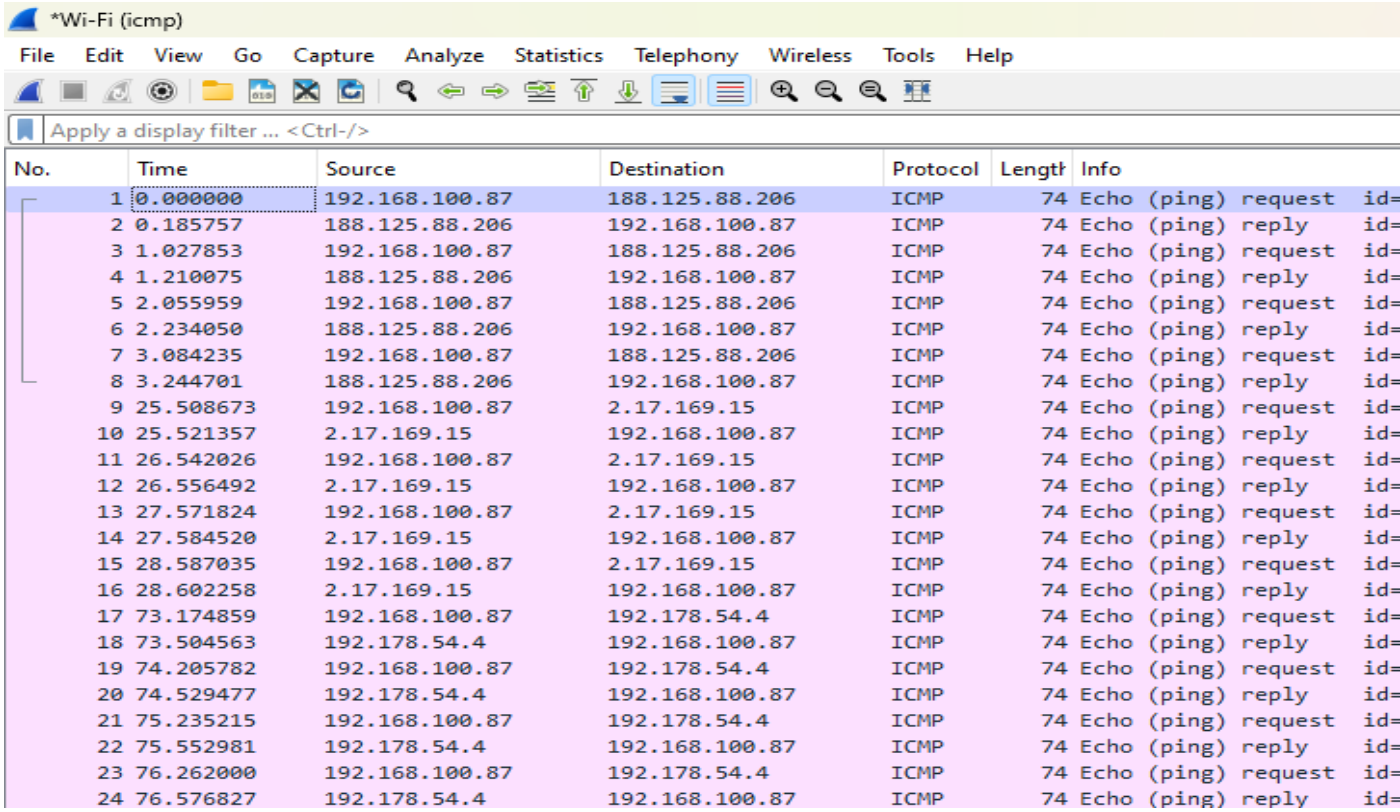
Part 2: Capture and Analysis of Remote ICMP Data in Wireshark

Step 1: Start capturing data on the interface.

I pinged three websites, www.yahoo.com, www.cisco.com and www.google.com

```
C:\Users\Tyano>ping www.yahoo.com

Pinging me-ycpi-cf-www.g06.yahoodns.net [188.125.88.206] with 32 bytes of data:
Reply from 188.125.88.206: bytes=32 time=185ms TTL=52
Reply from 188.125.88.206: bytes=32 time=182ms TTL=52
Reply from 188.125.88.206: bytes=32 time=178ms TTL=52
Reply from 188.125.88.206: bytes=32 time=160ms TTL=52

Ping statistics for 188.125.88.206:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 160ms, Maximum = 185ms, Average = 176ms

C:\Users\Tyano>ping www.cisco.com

Pinging e2867.dsca.akamaiedge.net [2.17.169.15] with 32 bytes of data:
Reply from 2.17.169.15: bytes=32 time=12ms TTL=57
Reply from 2.17.169.15: bytes=32 time=14ms TTL=57
Reply from 2.17.169.15: bytes=32 time=12ms TTL=57
Reply from 2.17.169.15: bytes=32 time=15ms TTL=57

Ping statistics for 2.17.169.15:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 12ms, Maximum = 15ms, Average = 13ms

C:\Users\Tyano>ping www.google.com

Pinging www.google.com [192.178.54.4] with 32 bytes of data:
Reply from 192.178.54.4: bytes=32 time=329ms TTL=109
Reply from 192.178.54.4: bytes=32 time=323ms TTL=109
Reply from 192.178.54.4: bytes=32 time=318ms TTL=109
Reply from 192.178.54.4: bytes=32 time=315ms TTL=109

Ping statistics for 192.178.54.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 315ms, Maximum = 329ms, Average = 321ms
```

I captured data from the three pings from the websites as shown below.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.100.87	188.125.88.206	ICMP	74	Echo (ping) request id=
2	0.185757	188.125.88.206	192.168.100.87	ICMP	74	Echo (ping) reply id=
3	1.027853	192.168.100.87	188.125.88.206	ICMP	74	Echo (ping) request id=
4	1.210075	188.125.88.206	192.168.100.87	ICMP	74	Echo (ping) reply id=
5	2.055959	192.168.100.87	188.125.88.206	ICMP	74	Echo (ping) request id=
6	2.234050	188.125.88.206	192.168.100.87	ICMP	74	Echo (ping) reply id=
7	3.084235	192.168.100.87	188.125.88.206	ICMP	74	Echo (ping) request id=
8	3.244701	188.125.88.206	192.168.100.87	ICMP	74	Echo (ping) reply id=
9	25.508673	192.168.100.87	2.17.169.15	ICMP	74	Echo (ping) request id=
10	25.521357	2.17.169.15	192.168.100.87	ICMP	74	Echo (ping) reply id=
11	26.542026	192.168.100.87	2.17.169.15	ICMP	74	Echo (ping) request id=
12	26.556492	2.17.169.15	192.168.100.87	ICMP	74	Echo (ping) reply id=
13	27.571824	192.168.100.87	2.17.169.15	ICMP	74	Echo (ping) request id=
14	27.584520	2.17.169.15	192.168.100.87	ICMP	74	Echo (ping) reply id=
15	28.587035	192.168.100.87	2.17.169.15	ICMP	74	Echo (ping) request id=
16	28.602258	2.17.169.15	192.168.100.87	ICMP	74	Echo (ping) reply id=
17	73.174859	192.168.100.87	192.178.54.4	ICMP	74	Echo (ping) request id=
18	73.504563	192.178.54.4	192.168.100.87	ICMP	74	Echo (ping) reply id=
19	74.205782	192.168.100.87	192.178.54.4	ICMP	74	Echo (ping) request id=
20	74.529477	192.178.54.4	192.168.100.87	ICMP	74	Echo (ping) reply id=
21	75.235215	192.168.100.87	192.178.54.4	ICMP	74	Echo (ping) request id=
22	75.552981	192.178.54.4	192.168.100.87	ICMP	74	Echo (ping) reply id=
23	76.262000	192.168.100.87	192.178.54.4	ICMP	74	Echo (ping) request id=
24	76.576827	192.178.54.4	192.168.100.87	ICMP	74	Echo (ping) reply id=

Step 2: Examining and analysing the data from the remote hosts.

A review of the captured data shows the following destination IP and MAC addresses for the tree locations

IP address for www.yahoo.com: 188.125.88.206

MAC address for www.yahoo.com: c8:b6:d3:a8:f1:95

IP address for www.cisco.com: 2.17.169.15

MAC address for www.cisco.com: c8:b6:d3:a8:f1:95

IP address for www.google.com: 192.178.54.4

MAC address for www.google.com: c8:b6:d3:a8:f1:95

What's significant About this information is that the MAC addresses can't be accessed unlike the previous part. So, the MAC address displayed is that of my router which last forwarded the packet and not the actual MAC address of the remote hosts.

This information differs in that, in Part 1, local ping information includes MAC addresses from devices in the same network, which Wireshark can capture. Part 2 captures data from remote hosts like Yahoo and Google, whose MAC addresses are typically inaccessible as they're outside the local network.

Reflection Question

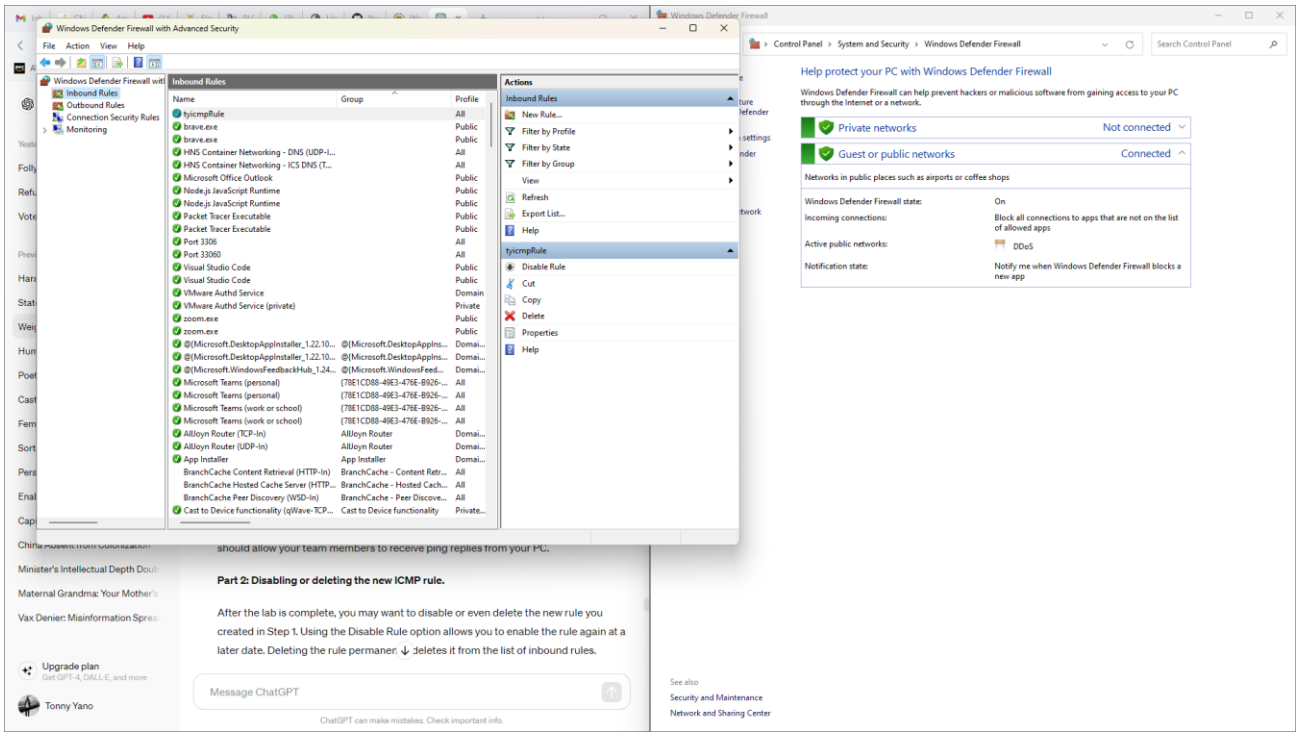
Why does Wireshark show the actual MAC address of the local hosts, but not the actual MAC address for the remote hosts?

Wireshark displays the MAC addresses of local hosts because these addresses are directly available in the data packets your computer receives from them. However, for remote hosts, the situation differs. As data packets pass through routers, the source MAC address gets replaced by each router's MAC address. So, when the packet reaches your computer, it shows the MAC address of the last router, typically your default gateway. Therefore, Wireshark can only display the MAC address of the last device.

Appendix A: Allowing ICMP Traffic Through a Firewall

Part 1: Create a new inbound rule allowing ICMP traffic through the firewall.

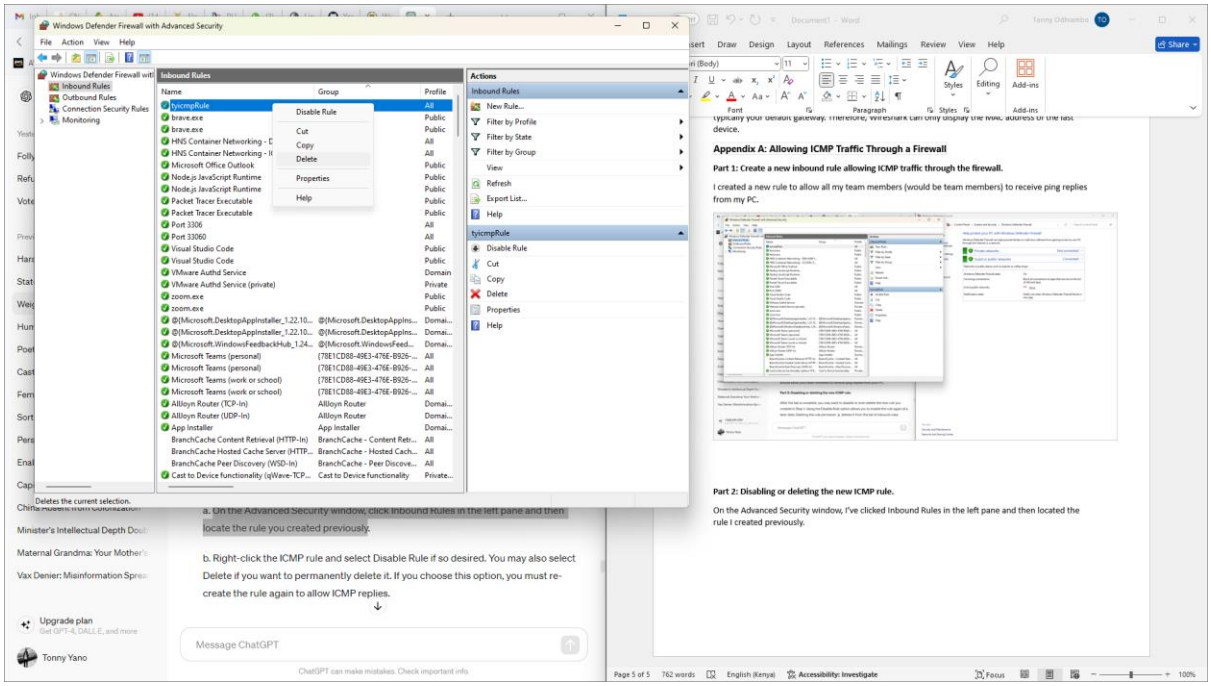
I created a new rule to allow all my team members (would be team members) to receive ping replies from my PC.



Part 2: Disabling or deleting the new ICMP rule.

On the Advanced Security window, I've clicked Inbound Rules in the left pane and then located the rule I created previously.

I have opted to delete the rule I created by right-clicking and the choosing delete.



Conclusion

This Wireshark exercise was a comprehensive exploration of network traffic analysis, offering me practical insights into the intricate workings of data transmission. Through capturing and analysing local ICMP data, I gained a deeper understanding of MAC and IP addresses, packet headers, and the mechanics of communication within a local network segment.

Additionally, the examination of remote ICMP data highlighted the differences in data transmission between local and remote hosts, shedding light on the role of routers in altering MAC addresses along the transmission path.

The accompanying appendix on configuring firewalls for ICMP traffic provided valuable knowledge applicable to real-world network administration scenarios. Overall, this exercise provided me with hands-on experience and enhanced my skills in network troubleshooting and management, empowering me with practical insights into network traffic analysis.