Wireshark to view network traffic

Completed	☑
Status	To Do: Theory

Objectives

Part 1: Capture and analyze local ICMP Data in Wireshark

Part 2: Capture and analyze Remote ICMP Data in Wireshark

Part 3: Allowing ICMP traffic through a firewall

Scenario

In this activity, traffic was generated from a source local network device to a destinated local network device as well as from a source local network device to a remote network. Using Wireshark, live traffic between the source and destination was ‘captured’ and examined.

Part 1: Capture and analyze local ICMP Data in Wireshark

The resources required for this activity are 1 PC with internet access and 2 or more PCs within the same network or virtual machines within the same NAT network, with at least 1 connected to the internet, as seen in Part 1 of this activity.

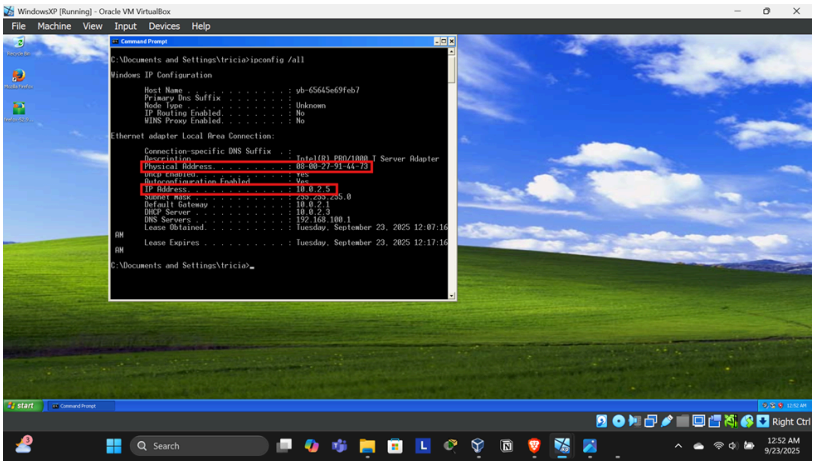
The two virtual machines were pinged, and using Wireshark, ICMP (Internet Control Message Protocol) Requests were captured. Ping, mostly used as a diagnostic tool, is used to clarify whether a device trying to be reached over the network is operational. The ICMP protocol is the underlying protocol of Ping,

which is connectionless as the devices do not need to be connected thus connectionless and operates in the network layer.

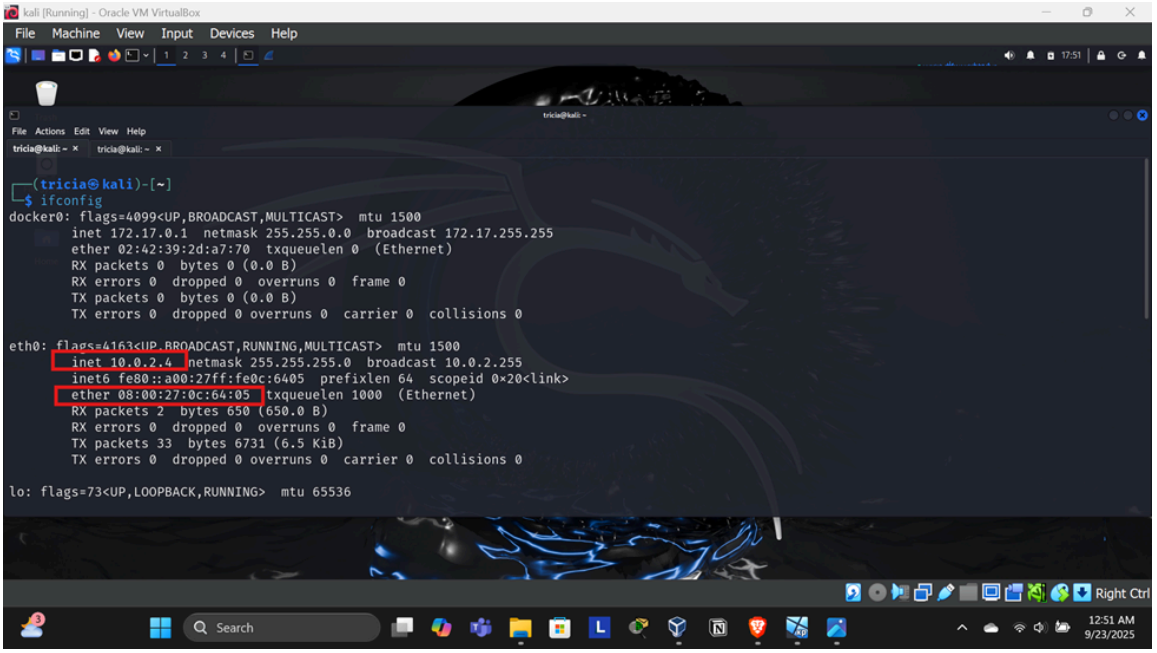
Step 1: Retrieve your PC interface addresses

For this activity, the Linux virtual machine, where the ping command will be conducted, its IP address and MAC address were retrieved as well as the Windows virtual machine which were within the same network.

1. In the command prompt window, the command <ipconfig /all> for the Windows virtual machine and <ifconfig> for the Linux virtual machine were run to obtain the source PC's and the destination PC's IP Address and MAC address. Therefore, for the Windows Machine, the indicated IP Address: 10.0.2.5 and MAC address: 08-00-27-91-44-73, while for the Linux Machine, the indicated IP Address: 10.0.2.4 and MAC address: 08:00:27:0c:64:05



Windows Machine

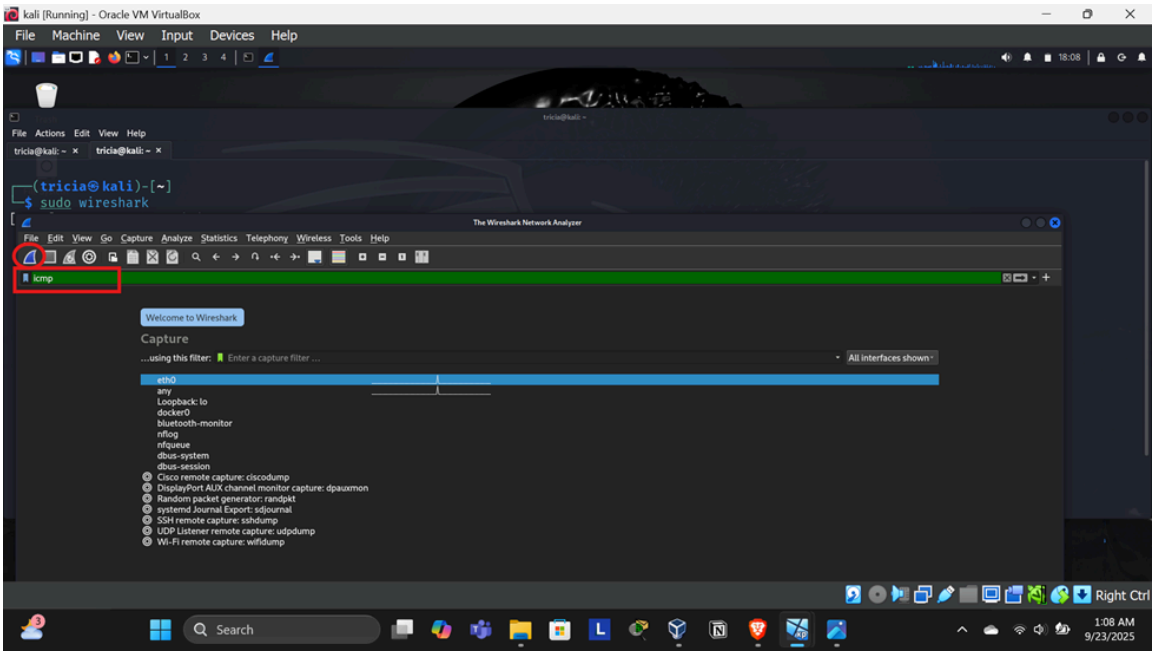


Linux Machine

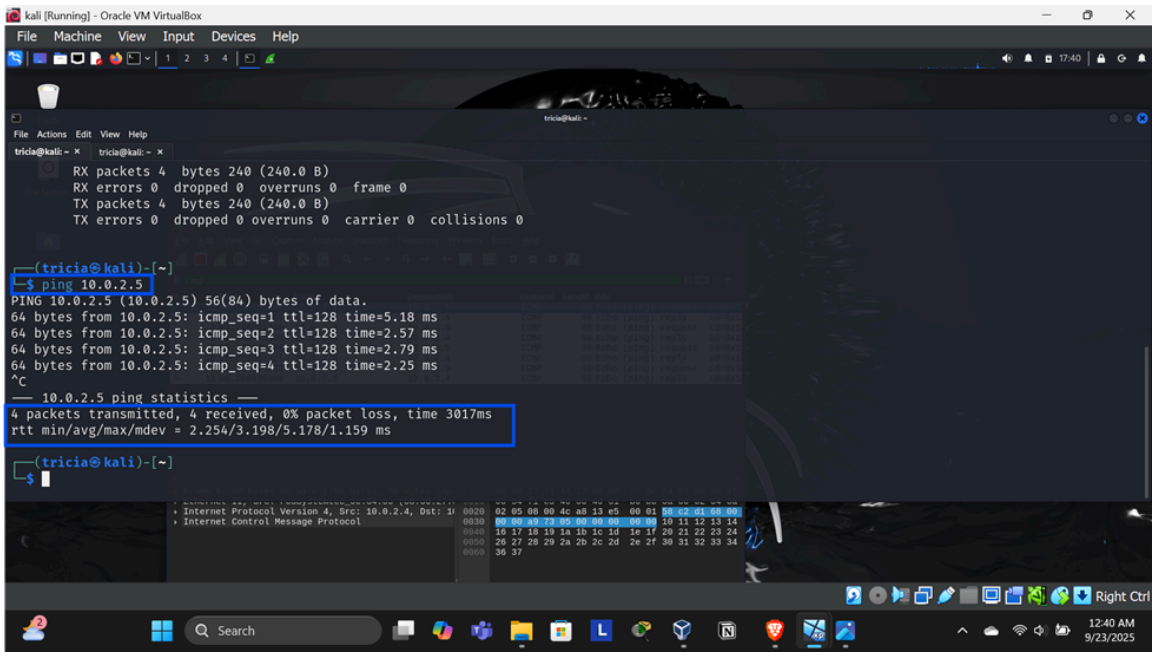
Step 2: Start Wireshark and begin capturing data

For this next step, Wireshark was installed on my Linux virtual machine.

1. Opened Wireshark on the desired interface to start the packet capture using the top left highlighted icon, as shown below.
2. Information can scroll by very quickly; therefore, to counter this, filtering was applied to make it easier to view and work with the data that is being captured. For this activity, ICMP ping PDUs are what we are interested in displaying; thus, ICMP was applied in the filtering section.



3. The IP address was pinged from the Linux virtual machine to the Windows virtual machine. The ping was successful as there were no packets lost during transmission, which indicated that the machine was operational. Using the Stop Capture icon, the capturing of data was halted.



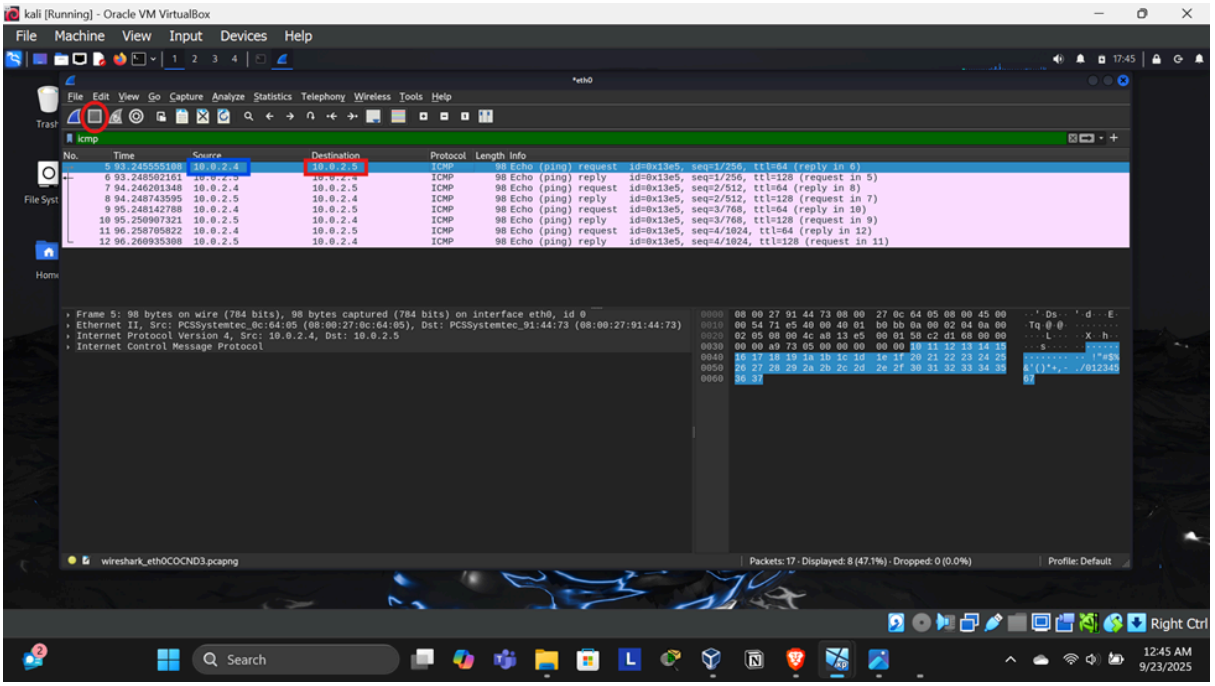
Step 3: Examine the captured data

In this section, the data generated by the ping requests to the Windows virtual machine was examined.

Wireshark data is displayed in 3 sections: The top section displays the list of PDU frames captured with a summary of the IP packet information, the left section lists PDU information for the frames selected and separates the PDU information by its protocol layer, and lastly, the right section displays raw data of each layer in hexadecimal and decimal form.

1. First ICMP request

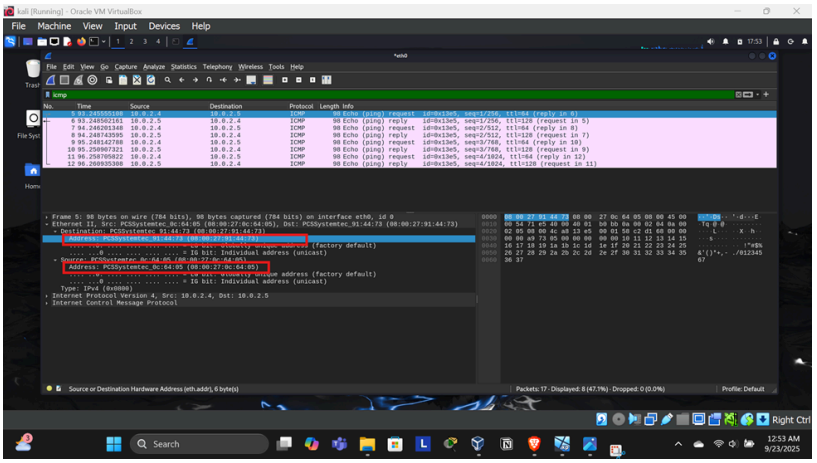
When the first ICMP request was clicked on, it was evident that the Source column was the Linux Machine's IP address, and the Destination column contained the Windows Machine's IP address.



1. OSI Model Layer 2 (Ethernet II)

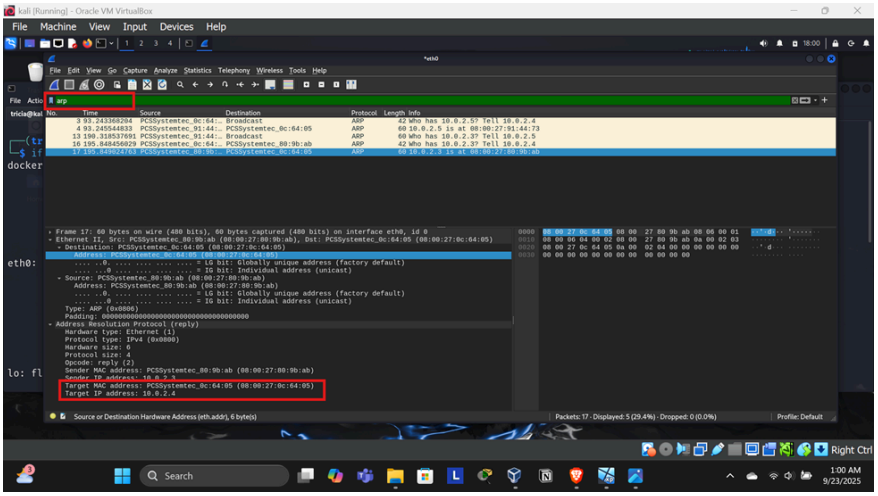
At Layer 2 of the OSI model, also known as the Data Link Layer, the packet is prepared for transmission across the local network. In this case, the ICMP data is first encapsulated inside an IPv4 packet at Layer 3. That IPv4 packet is then encapsulated within an Ethernet II frame, which includes an Ethernet II header. By adding this header, the network ensures that the frame can be properly addressed and delivered to the correct device on the same physical network segment.

1. As shown below, the source's as well as the destination's MAC addresses matched to their assigned physical address.



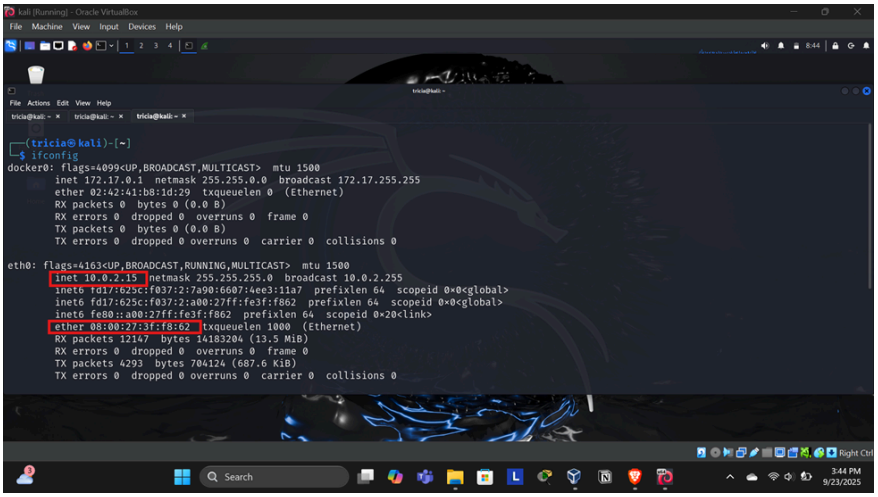
2. Obtaining the MAC address

Through the ARP process: sending a broadcast ARP request to all devices within the LAN, associating hosts with their IP addresses, finding the IP address associated with the MAC address by finding IP address that matches it, and storing the results in the MAC address table, we obtained the MAC address for the source PC.



Part 2: Capture and analyze remote ICMP data in Wireshark

In this section, remote hosts, hosts not on the LAN, were pinged, and from those pings, the data generated was examined using Wireshark in the Linux virtual machine.

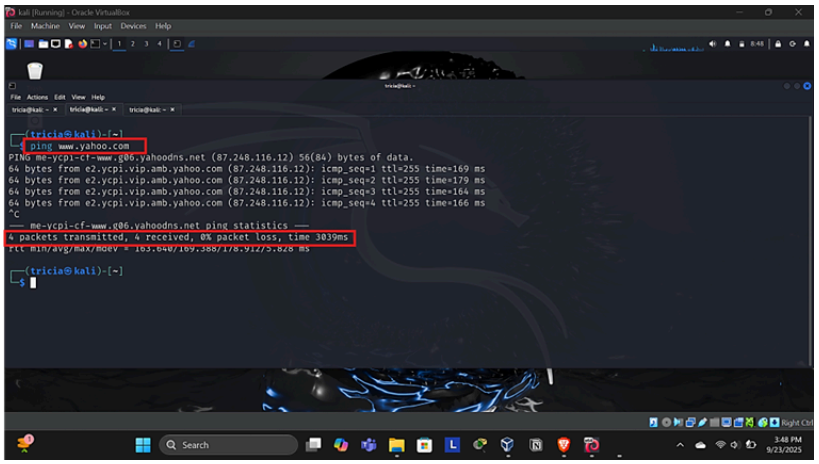


Step 1: Start capturing data on the interface

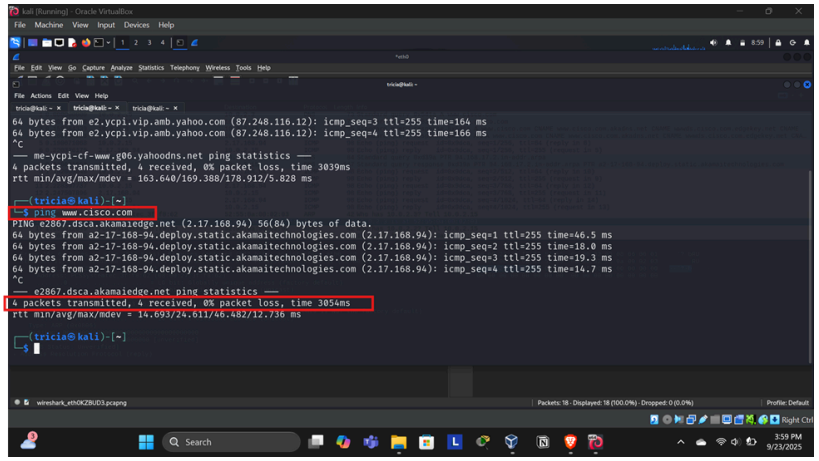
1. Restarted the data capture and pinged the target domain names as shown below.
    a. Ping

    The URLs were first pinged to ensure their servers were operational, as well as to generate traffic, which was captured by Wireshark, and as shown below, the pings were successful as there were no lost packets.

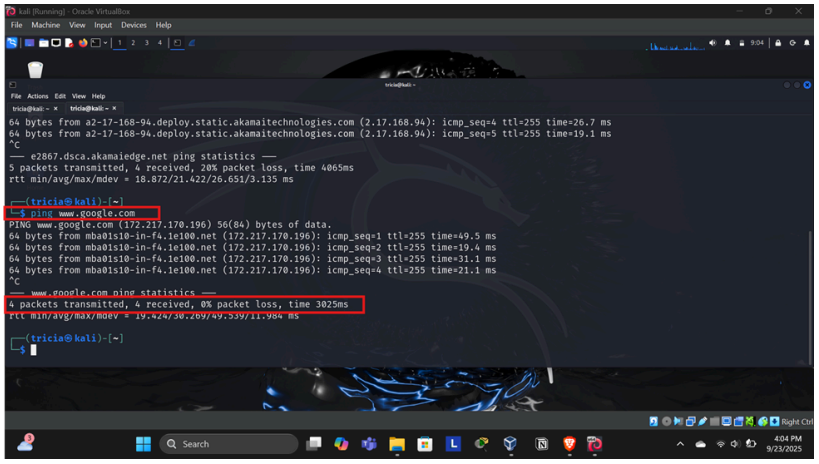
    - yahoo.com



    - cisco.com



- google.com

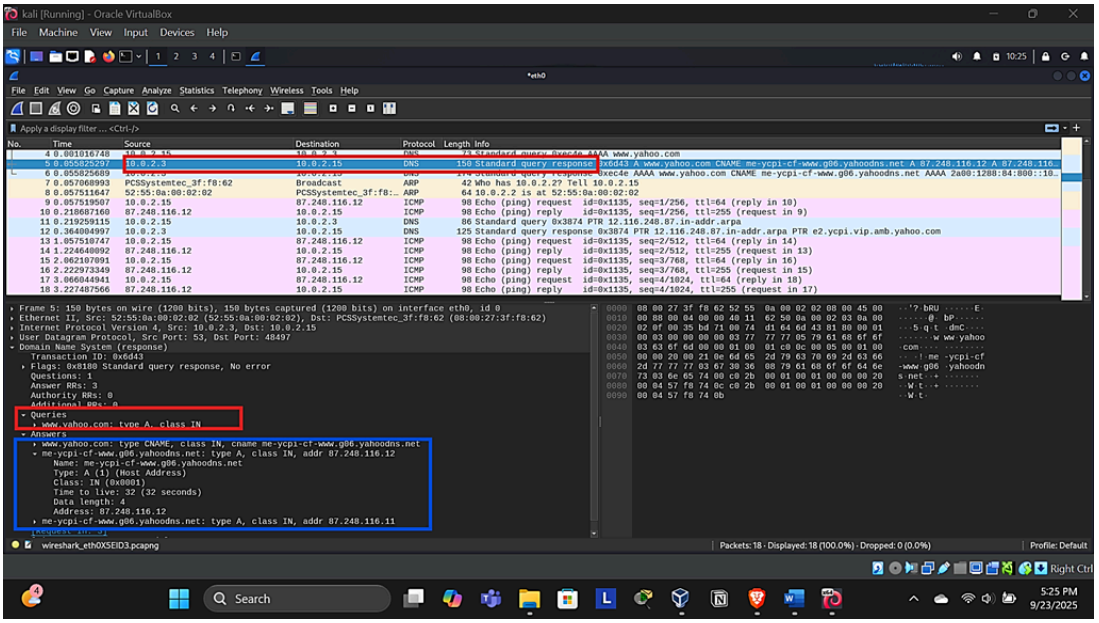


2. Domain Name Server Protocol

   DNS is used to translate a human-readable domain name into a machine-readable IP address. In this activity, Wireshark was used to capture and analyze the DNS PDU information generated when domain names were queried through the ping command. The captured results showed that for each query, the DNS response included the resolved IP address of the domain name, and in some cases, a CNAME record, where a domain name is mapped to another canonical name instead of directly mapping to the domain IP address.

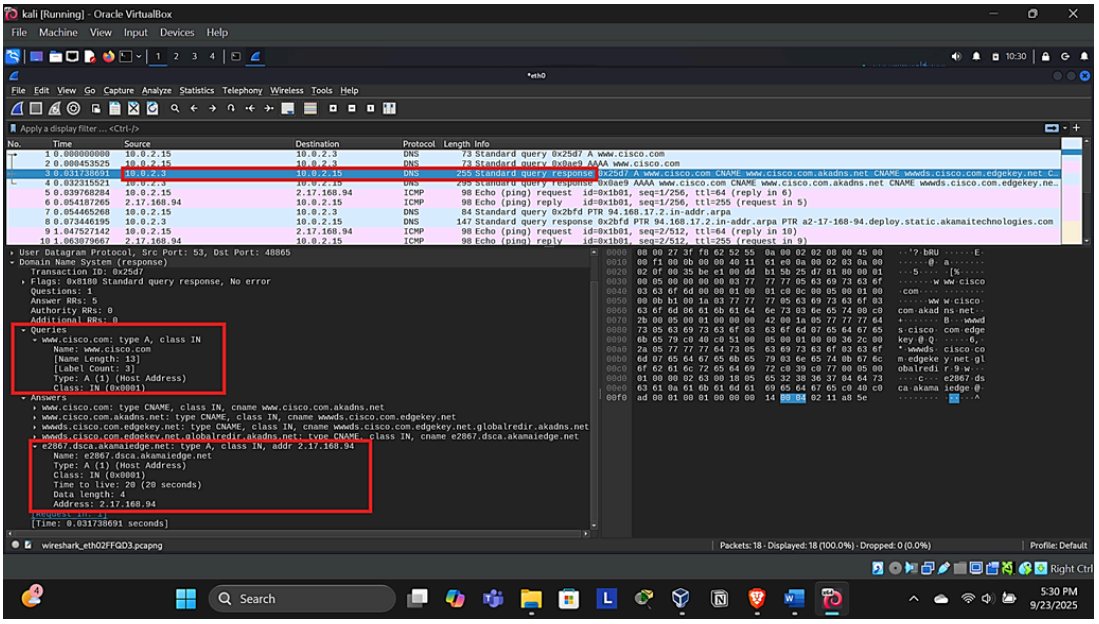
   - yahoo.com

     The IP address for the domain name is: 87.248.116.12



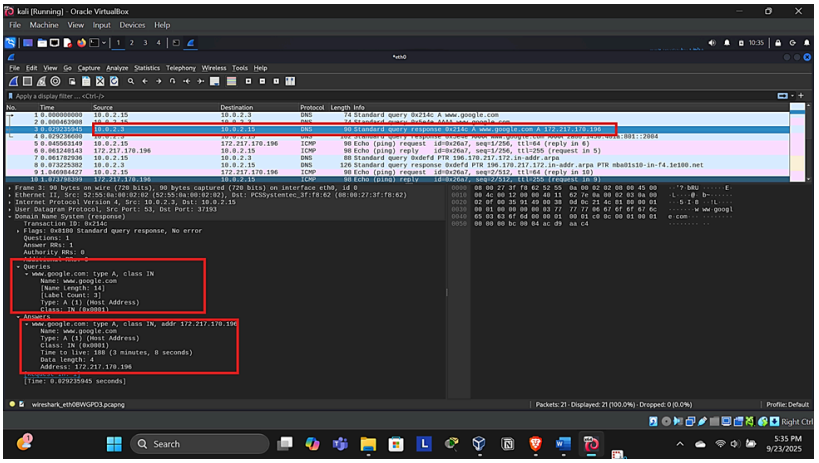
- cisco.com

  The IP address for the domain name is: 2.17.168.94



- google.com

  IP address for the domain name is: 172.217.170.196



Step 2: Examining and analyzing the data from the remote hosts

In this section, the captured data was reviewed, and the IP and MAC addresses of the pinged domain names were examined as shown below.

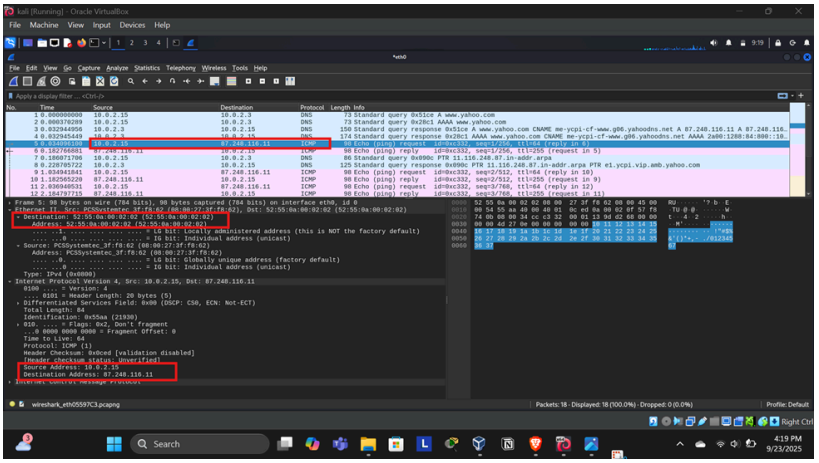
1. Remote hosts' IP addresses and MAC addresses

   Successful pings to remote hosts indicated that the virtual machine was properly connected to the internet. The results also confirmed that the DNS process had successfully resolved the URL to the correct IP address, allowing the destination host to be reached. Ultimately, this showed that the remote host was operational and responsive.

   Note: Despite there being an MAC address captured below, they do not belong to the remote host but the local router. This will be discussed further in (b) and Part 3.

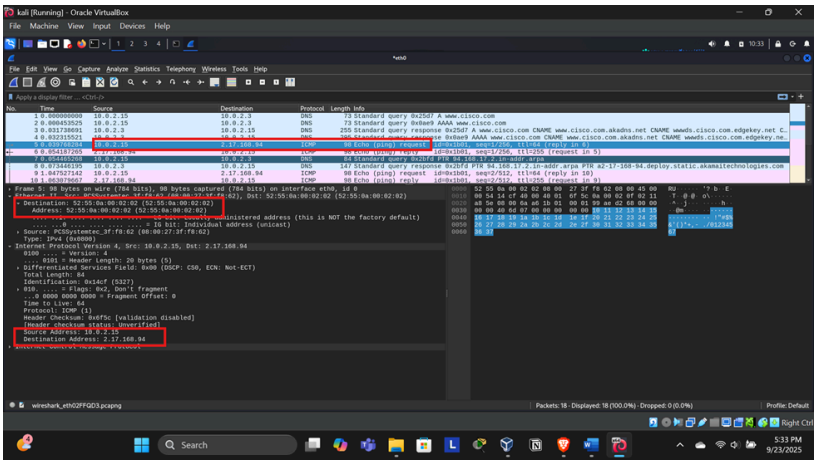
   a. yahoo.com

      As shown below, the IP address of this domain is: 87.248.116.11, and the MAC address: 52:55:0a:00:02:02



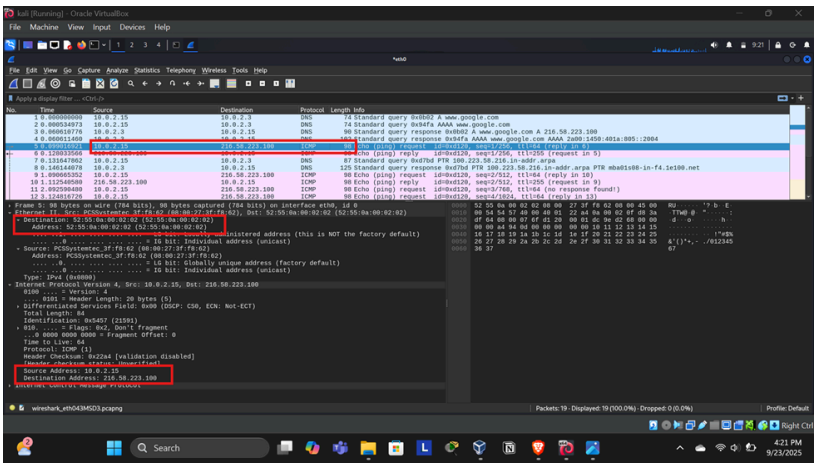
b. cisco.com

As shown below, the IP address of this domain is: 2.17.168.94 and the MAC address: 52:55:0a:00:02:02



c. google.com

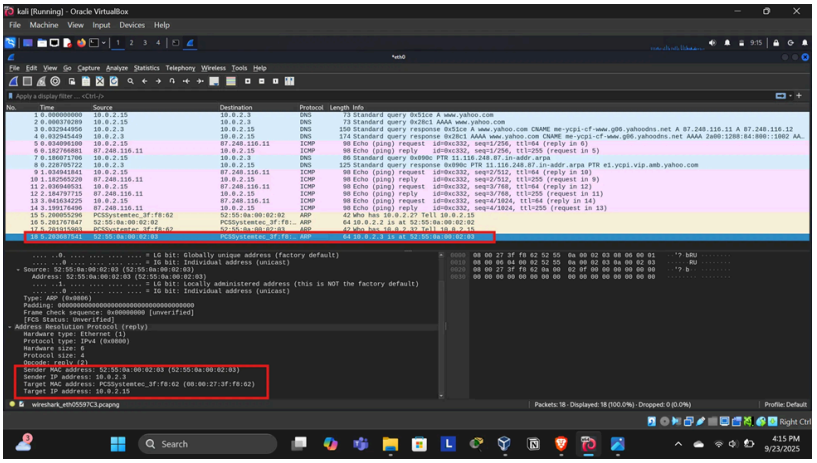
As shown below, the IP address of this domain is: 216.58.223.100 and the MAC address: 52:55:0a:00:02:02



2. Difference from pinging local hosts within the same LAN network

   The results showed that when pinging remote hosts, Wireshark did not capture their actual MAC addresses. Instead, only the MAC address of the local router was visible, since Layer 2 information is reassigned at each hop across the network. In contrast, when pinging local hosts within the same LAN, Wireshark captured the true physical MAC addresses of those devices.

   As shown below in the ARP process, it was evident that the MAC address for the local router is similar to what was captured in part (a).



Part 3: Capture remote hosts' MAC addresses with Wireshark

The previous results indicated that Wireshark was unable to capture the MAC addresses of remote hosts. MAC addresses are “physical” identifiers assigned to network interface hardware, and they enable communication within a local network segment. When a packet is sent to a remote host, it goes through multiple routers. At each hop, the Layer 2 information (including the MAC address) is stripped and replaced with new information specific to that link. As a result, the original MAC address of the remote host is never visible to the local network or captured by Wireshark.