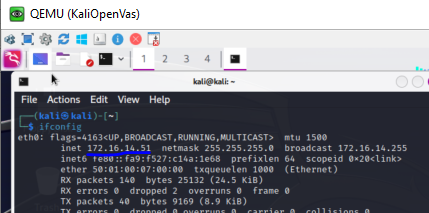
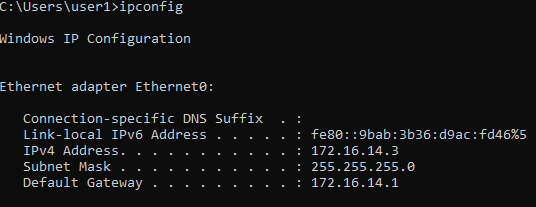
**Network Administration Project**

July 3, 2023

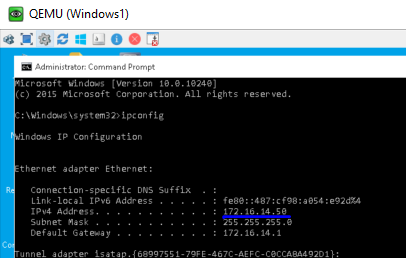
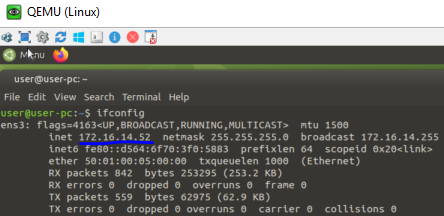
From the foregoing request to investigating all the endpoints (devices) within the network to confirm IP addresses, check open ports, and run nmap (Network map) scans, here are the steps followed and the finding also reported.

**Step 1:** I ensured that all the endpoints (devices) within the network connection (subnets) are switched on and running. The network subnet is 172.16.14.0/24 and the device that will be eventually used to run the entire scan of the network connection in this investigation (the Jump host) has the IP address of 172.16.14.3. An IP Address is a unique numerical identifier for every device or network that connects to the internet.

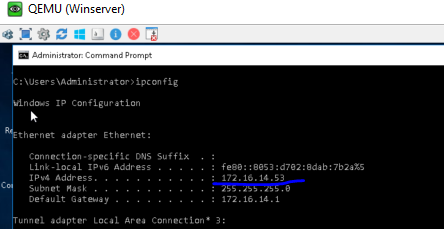
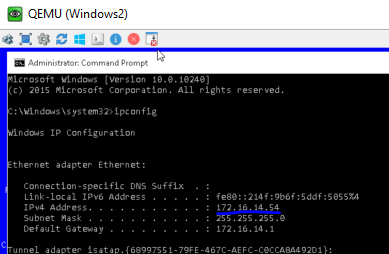
**Step 2**: I ran an *ipconfig/ifconfig* scan on each endpoint Command prompt/Terminal to confirm and record their IP addresses.



*Figure I: Scan for Jumphost (left) and Kali Linux machine (right) to confirm IP Address.*



*Figure II: Scan for Linux Machine (left) and Windows 1 machine (right) to confirm IP Address.*



*Figure III: Scan for Windows 2 machine (left) and Winserver machine (right) to confirm IP Address.*

*Details of the IP Addresses and other machine information are included in Table A below.*

We used 2 software tools to run these tests, namely Zenmap and Wireshark. The Zenmap Graphical software/application is used a command (nmap) to send queries (packets) to discover hosts (devices) and services on the computer network and analyzing the responses. While Wireshark is a graphical application that traffic (packets) from a network connection. Therefore, since the Jump host is on the same network connection with the other endpoints, we will run to following tests.

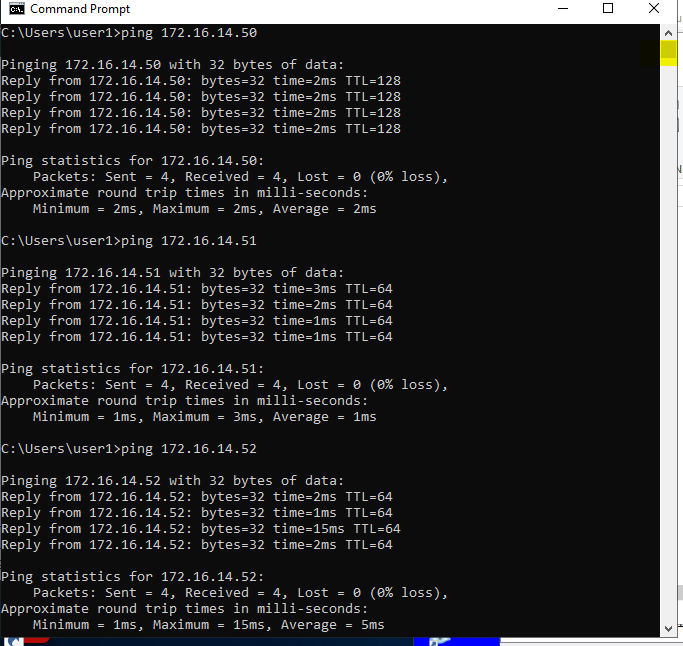
**Step 3**:

Launched Command Prompt, Zenmap and Wireshark applications. I ensure that these are functioning on the Jumphost.

**Step 4:**

Performed a Ping Test for each endpoint device following this procedure.

* Click on the Ethernet session on the Wireshark to ensure it’s running and stop the capture.
* Type ping <IP Address> in the command prompt.
* Start the Wireshark scan again and immediately press the “enter” key to run the ping command in the command prompt.
* After the ping command is completed, I stopped the Wireshark capture and [save in this folder](https://drive.google.com/open?id=11EYwMCORmvUJX6cnO0w40uPwZ3jnJpFS&usp=drive_fs).
* I repeated this process for the 4 remaining devices and saved the Wireshark capture.
* All devices ping requests were successful and below is a screenshot of a few of the ping results displayed.



*Figure IV: Command prompt display of 3 of the endpoint pings.*

**Step 5:**

I then entered 172.16.14.0/24 in the target session on the Zenmap of the Jumphost, selected “Regular scan” in the “Profile” dropdown. Is will display nmap 172.16.14.0/24 in the “command” box.

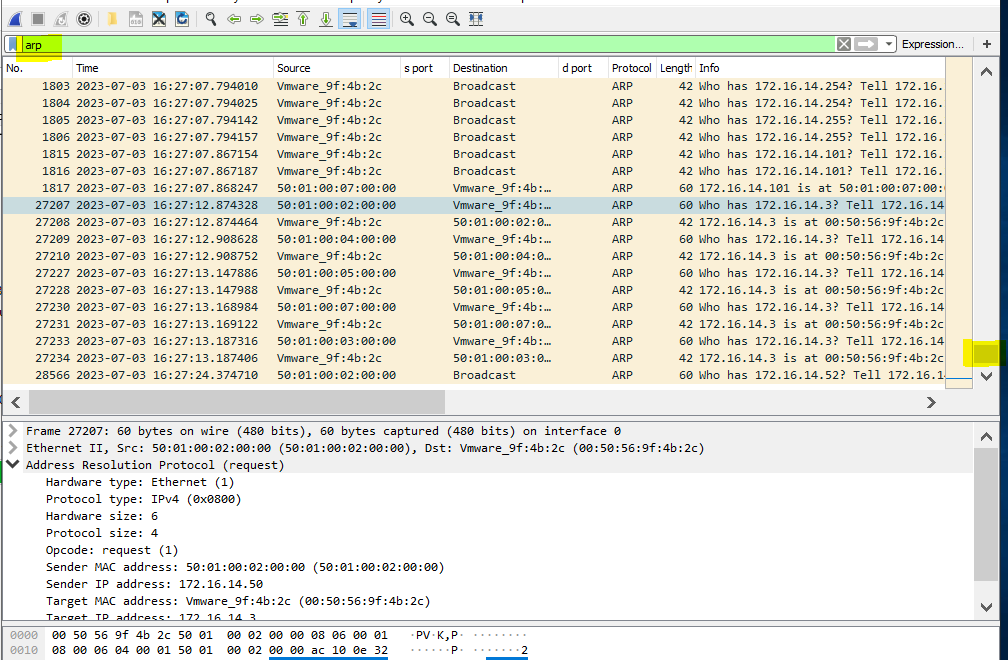
A screenshot of a computer

Description automatically generated with low confidence

*Figure V: Zenmap regular scan illustration*

After this, I started the Wireshark to capture the flow of traffic on the Jumphost, and immediately clicked on the “Scan” button. When Zenmap finished sending packet/queries to discover hosts and services on the network, I stopped the Wireshark capture and [save it in this folder.](https://drive.google.com/open?id=11Zuj8gnOZVsqi3qITgqz6U0NUmdiYtoJ&usp=drive_fs)

I studied the analysis in the Zenmap “host”, and “nmap output” panes to record the output and the Wireshark output as well.



*Figure VI: Example screenshot of Wireshark displaying the result of nmap scan, with ARP filter implemented on the display.*

**Step 6**:

I ran another scan on Zenmap of the Jumphost to discover the Operating System, open ports, and available/listening services running on the discovered endpoints/devices/Virtual machine. For this scan, I input the following command line in the “command” box for each device according to the IP addresses.

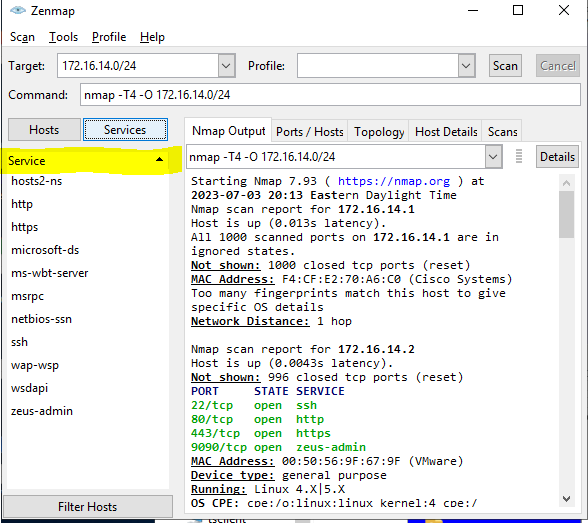
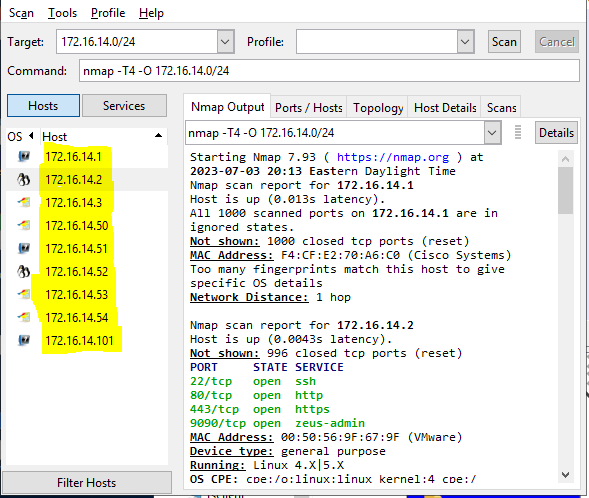
nmap -T4 -O <IP Address>

Follow this same step for Wireshark scan and started in **step 5**, capture and saved PCAP (Wireshark file) for 172.16.14.50 and 172.16.14.54 [devices in this folder](https://drive.google.com/open?id=11Zuj8gnOZVsqi3qITgqz6U0NUmdiYtoJ&usp=drive_fs). Here is also [the links for Zenmap files](https://drive.google.com/open?id=11TowllwjGbDaO9S2WuI0M7Klp0yp9KsG&usp=drive_fs).

I also run a comprehensive scan on the entire network as well and save the Wireshark capture in the folder. See figure below. More detailed information in Table A below.

Command used: nmap -T4 -O 172.16.14.0/24

I studied the analysis in the Zenmap “host”, “services”, “nmap output”, “Ports/hosts” and “host details” panes to record the output and the Wireshark output for Step 5 and Step 6 as well.



*Figure VII: Zenmap for network subnet.*

*Table A: Full machine documentation in the Main Lab environment*

| **Machine Name** | **OS Version**  **(%accuracy)** | **IP Address** | **MAC Address** | **Open ports** | **Listening service** |
| --- | --- | --- | --- | --- | --- |
| **Network Gateway** | N/A | **172.16.14.1** | FA:CF:EC:70:A6:CO | N/A |  |
| **EVE Server** | Linux 4.15 - 5.6 (*100*%) | **172.16.14.2** | 00:50:56:9F:67:9F | = tcp/**22**  = tcp/**80**  = tcp/**443**  = tcp/**9090** | SSH  HTTP  HTTPS  ZUES-ADMINS |
| **Jump host Machine** | MS Windows 10 1809 -1909 (*97*%) | **172.16.14.3** | 00:50:56:9F:4B:2C | = tcp/**135**  = tcp/**139**  = tcp/**445**  = tcp/**3389** | MSRPC  NETBIOS-SSN  MICROSOFT-DS  MS-WBT-SERVER |
| **Winserver** | MS Windows 10 1507 - 1607 (*100*%) | **172.16.14.50** | 50:01:00:02:00:00 | = tcp/**135**  = tcp/**139**  = tcp/**445**  = tcp/**3389** | MSRPC  NETBIOS-SSN  MICROSOFT-DS  MS-WBT-SERVER |
| **Kali Linux Machine** | Too many fingerprints match this host to give specific OS details | **172.16.14.51** | 50:01:00:07:00:00 | All 1000 scanned ports on 172.16.14.51 are in ignored states. |  |
| **Linux Machine** | Linux 4.15 - 5.6 (*100*%) | **172.16.14.52** | 50:01:00:05:00:00 | = **80**/tcp  = **3389**/tcp  = **9200**/tcp | HTTP  MS-WBT-SERVER  WAP-WSP |
| **Windows1 machine** | Microsoft Windows Server 2016 build 10586 - 14393 (*100*%) | **172.16.14.53** | 50:01:00:04:00:00 | = tcp/**135**  = tcp/**139**  = tcp/**445**  = tcp/**3389** | MSRPC  NETBIOS-SSN  MICROSOFT-DS  MS-WBT-SERVER |
| **Windows2 machine** | Microsoft Windows 10 1507 - 1607 (*100*%) | **172.16.14.54** | 50:01:00:03:00:00 | = tcp/**135**  = tcp/**139**  = tcp/**445**  = tcp/**3389**  = tcp/**5357**  = tcp/**81** | HOSTS2-NS  MSRPC  NETBIOS-SSN  MICROSOFT-DS  MS-WBT-SERVER  WSDAPI |
| **Unknown Endpoint** | All 1000 scanned ports on 172.16.14.101 are in ignored states. | **172.16.14.101** | 50:01:00:07:00:00 | Too many fingerprints match this host to give specific OS details. |  |

**Conclusions:**

* I discovered that the MAC Address for the unknown endpoint is similar to the Kali Linux MAC Address = 50:01:00:07:00:00.
* Zenmap/nmap gathers information by sending queries to all available IP addresses in the network subnet or in a specific IP Address scan, seeking for the Layer 2 device/network that has the active/running devices. After it establishes a true/active connection, it then sends packets requesting for all available Layer 2 – 7 information as requested in the command. It does encapsulation and decapsulation within milliseconds as it sends packets across the network.
* I observed that ARP (Layer 2), ICMP, ICMPv6 (Layer 3), TLSv1.2 (Layer 4 - 7), and TCP (Layer 4) port interaction during the scan in Wireshark capture (both for network subnet scan and devices scan for information about OS (Operating system) with -O command).

*Prepared by Titobiloluwa (Tito) Ilori*