Digital SkillUp Africa Final Capstone Project in Cybersecurity

Project Title: Building a Virtual Cybersecurity Laboratory and Conducting Android Forensics Investigations

Part Number: ONE

Part Title: Virtual Cybersecurity Lab Step

Student Name: ONAEKO, Emmanuel Oladipupo

Email: eonaeko778@gmail.com

Activites

- Install a Type 2 hypervisor
- Create and configure two virtual Machines
- Establish internal virtual networking between VMs
- Verify connectivity via ping tests, shared directories and service enumeration.

Tools used for this cybersecurity lab setup

Operations	Tools
Hypervisor	Virtual Box
Operating Systems	Kali Linux, Windows
Service Enumeration	Nmap

Installation of Type 2 hypervisor

This virtual cybersecurity lab was set up through the use of a Type 2 hypervisor **“ Virtual Box”** which was the platform to visulize two different operating systems used in the lab which are **Kali Linux** and **Windows 10**, the OS were allocated the memory and processor needed to perform effectively and efficiently and were configure with password and user profile as required and they were tested for some functionality that is also present in the host device like internet connectivity.

Internal Virtual networking between VMs

The VMs, although under the same hypervisor, are separate devices and would not normally communicate, so to enable them to communicate or perform internal networking, they must be under the network, and to do this, I perform the following:

1. **Change of network adapter settings**

   The network adapter configuration was configured on the hypervisor “Virtual Box “ to change from NAT to internal network, and this was done for both Kali Linux and Windows VMs, which meant that they were now on the same logical network.

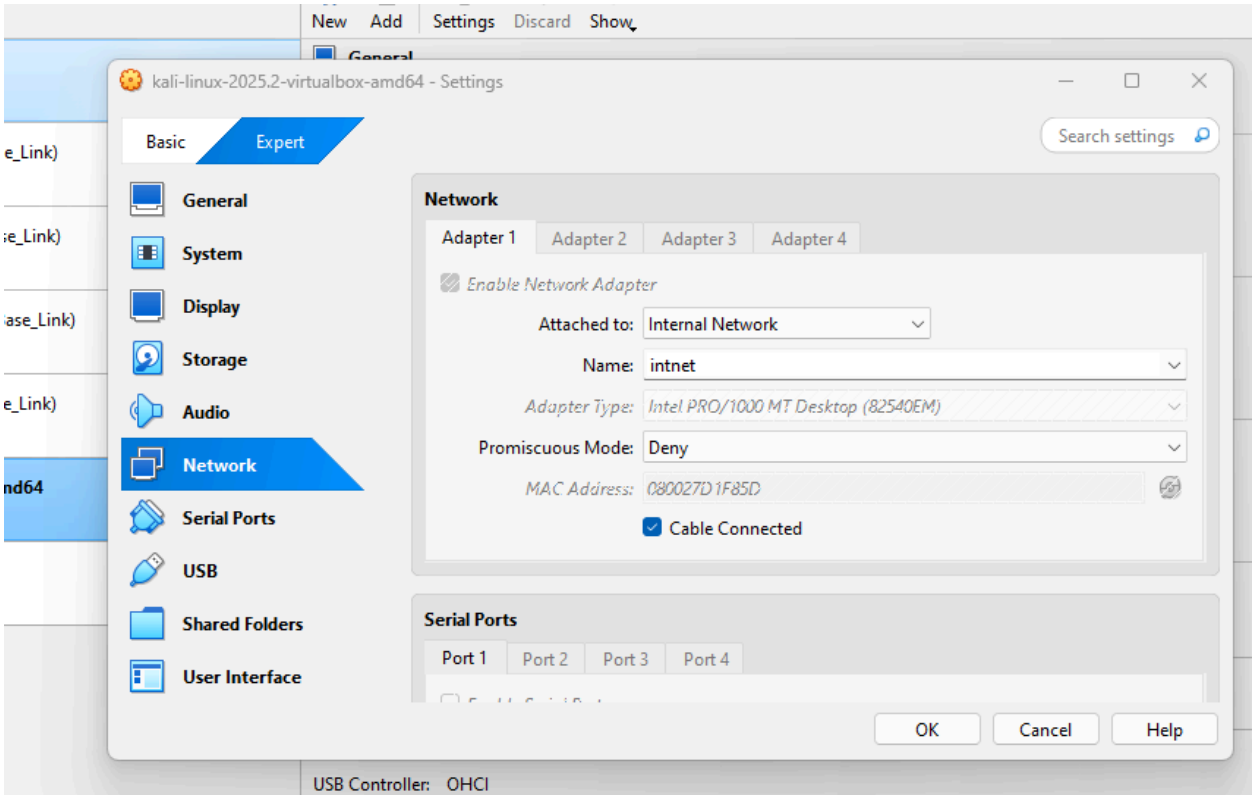

Fig. 1 shows the configuration of the network adapter on Kali Linux.

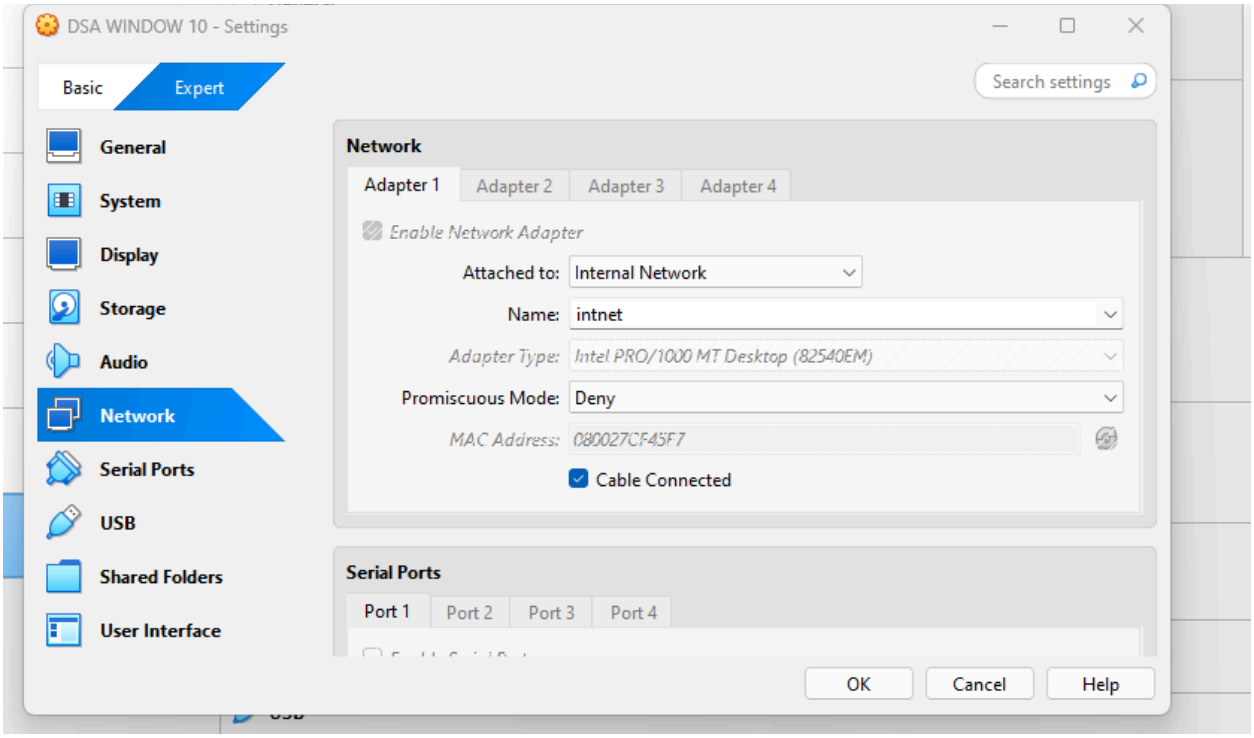

Fig. 2 shows the configuration of the network adapter on Windows 10

2. Configure private IP addresses from the same subnet to both VMs

For internal networking and communications to occur, both Kali Linux were configured with IP addresses from the same subnets. I configured Kali with IP address **192.168.1.10** with subnet mask: **255.255.255.0** and Windows VMs with IP address **192.168.1.20** with subnet mask: **255.255.255.0**. Kali Linux was configured from the terminal with the command: "*sudo ip addr add 192.168.1.10*" while Windows was configured through the GUI

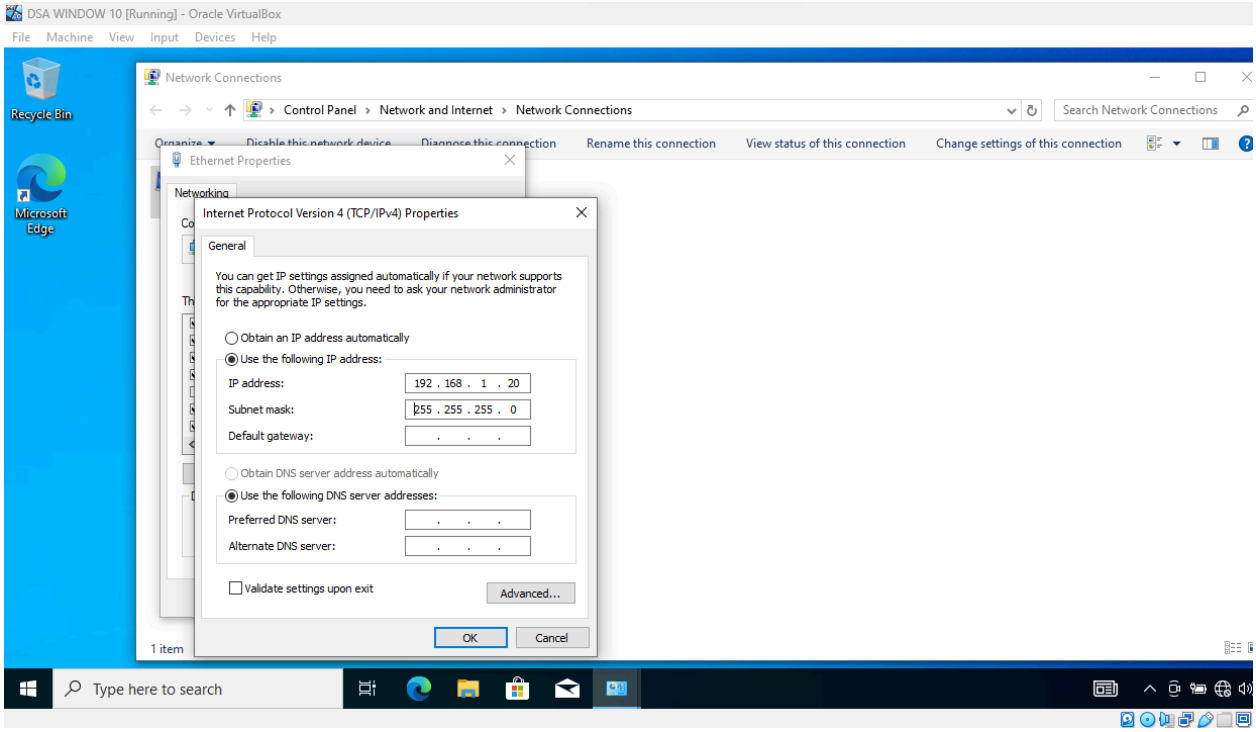

Fig. 3 shows the IP configuration on the Windows VM

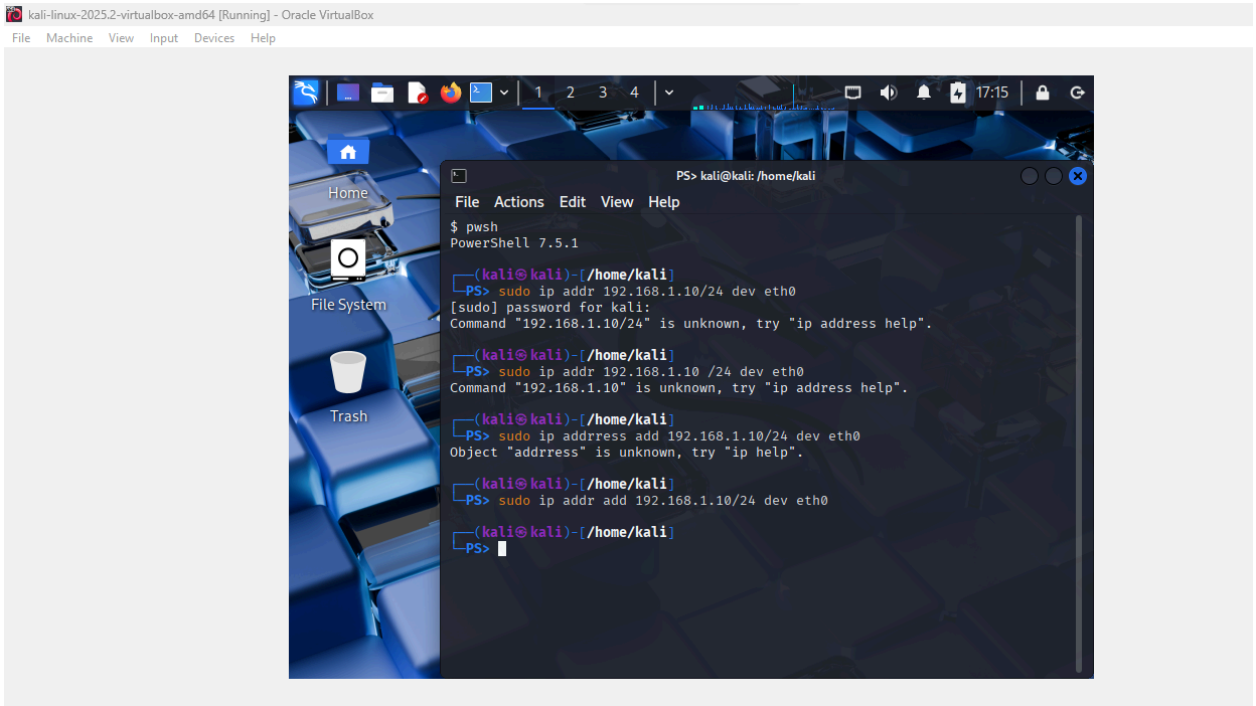

Fig. 4 shows the IP configuration on Kali Linux.

3. Verification of Internal networking via ping tests

I verify their internal networking and communications by carrying out ping tests on the terminal of both virtual machines. The ping fails at first, but after setting up the inbound rules on Windows Firewall to allow ICMP ECHO messages, I ping again, and the result was successful.

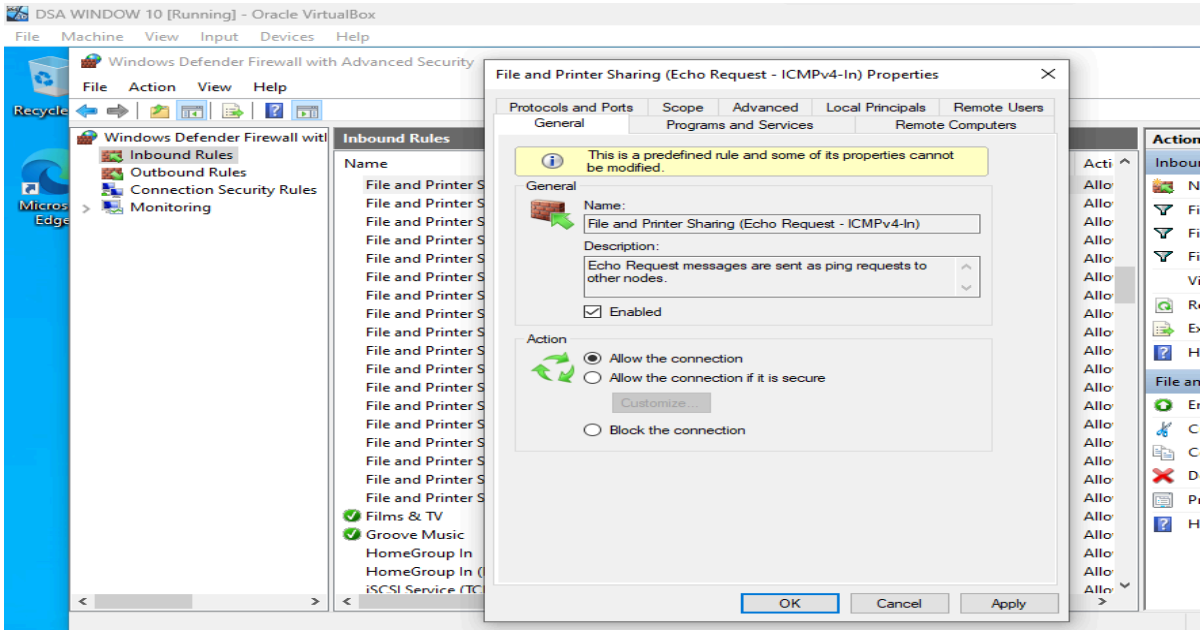

Fig. .5 shows a capture of the Windows Defender firewall being configured to allow transport.

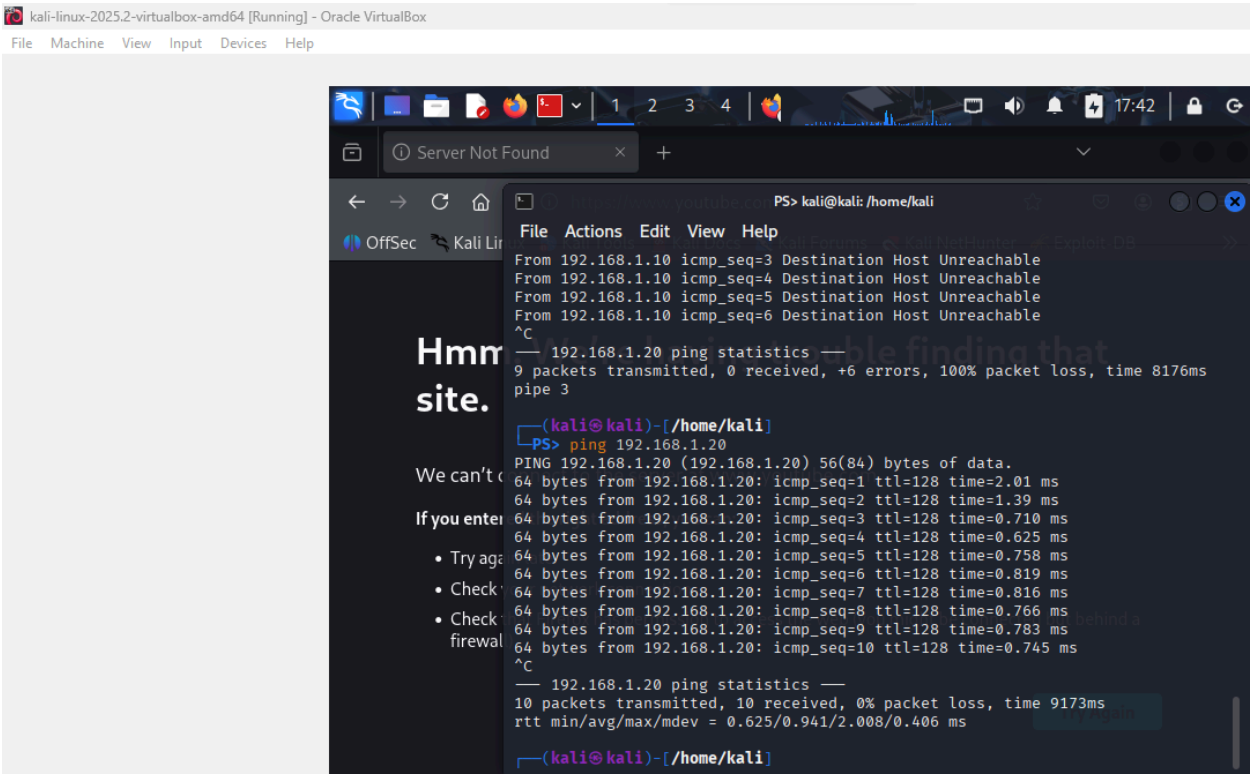

Fig. .6 shows a capture of the successful ping tests to 192,168.1.20 from Kali Linux.

4. Shared Files

After Connectivity was established between the two VMs, I created a folder on the Windows VM, named WannaShare, and configured it to be a shared folder accessible to computers on the network. I accessed the file through the Kali Linux VMs after proper research on how to access the files through the terminal. I use the command: *“sudo smbclient //192.168.1.20/WannaShare -u 'emmanuel difr dsa'” to* access the content of the folder, which was successful.

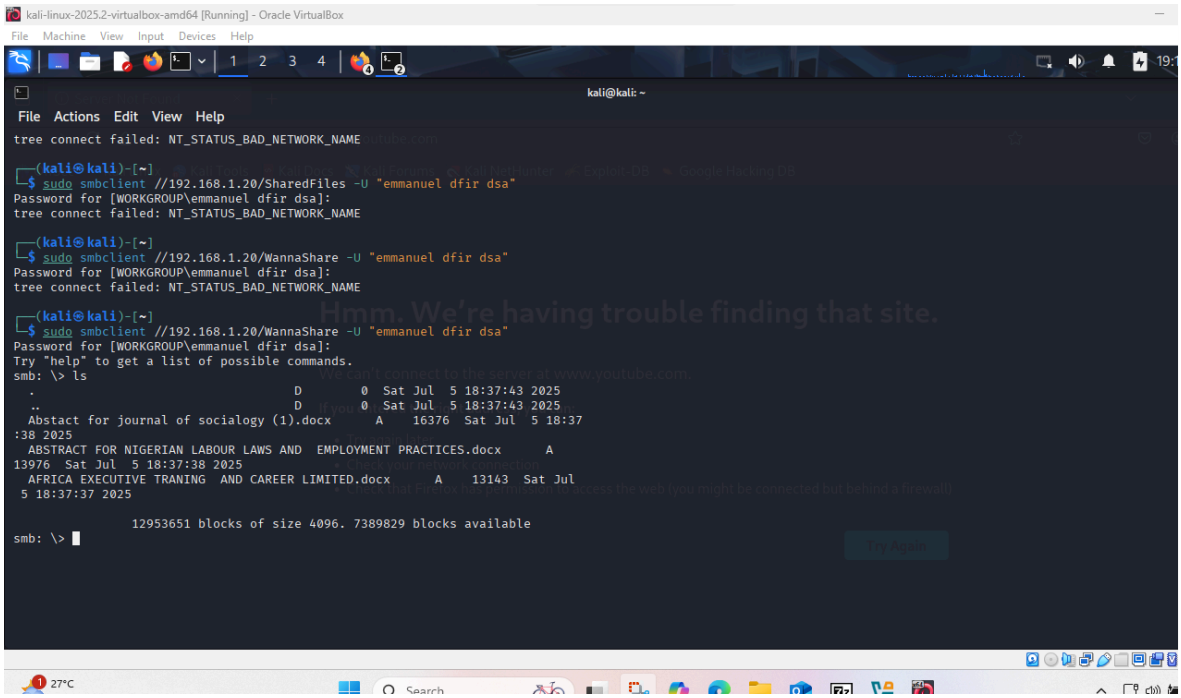

Fig. .7 shows a capture of the shared folders being accessed, with some previous errors on Kali Linux.

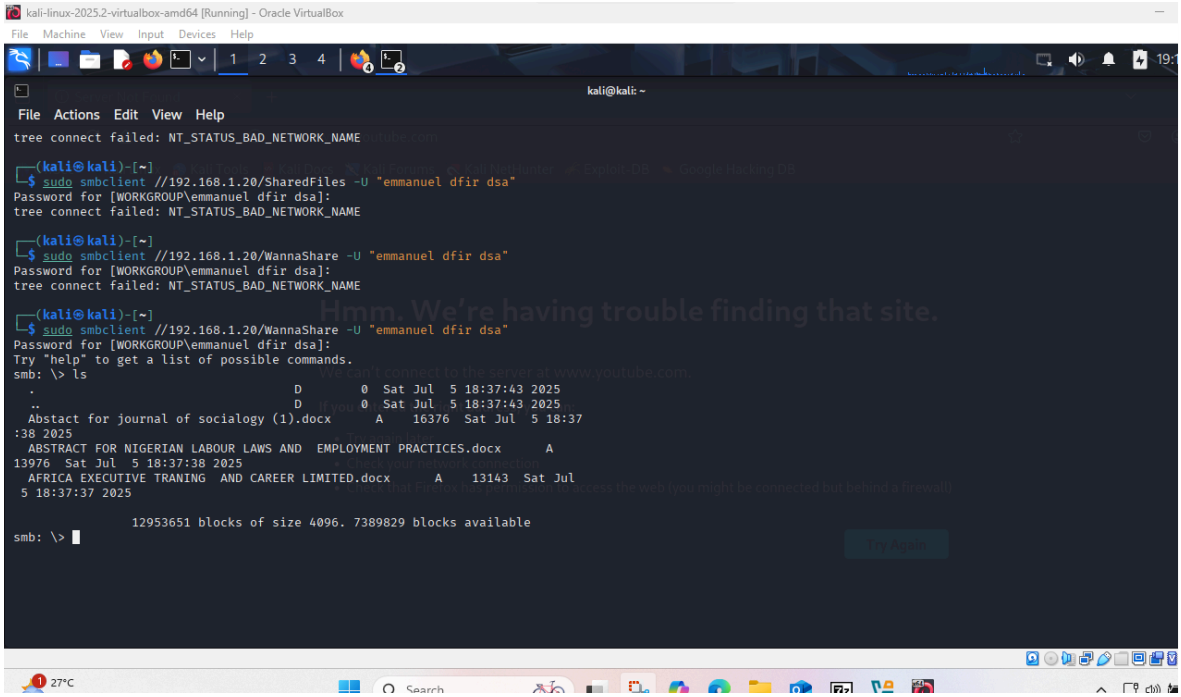

Fig. .8 shows a capture of the files being accessed through the terminal.

5. Service Enumeration

To perform service enumeration, I used the nmap function to do a number of vulnerability assessments on the Windows OS.

nmap 192.168.1.20 to scan for the IP address 192.168.1.20

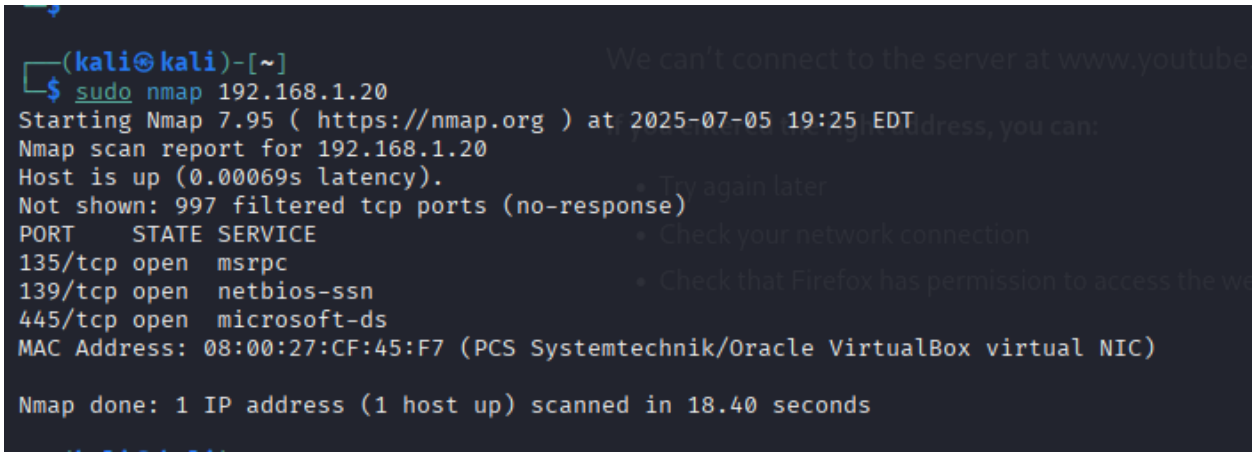

Fig. .9 shows a capture of nmap scanning the IP address of 192.168.1.20

nmap -p- 192.168.1.20 scans all 65535 ports on the target IP address: 192.168.1.20

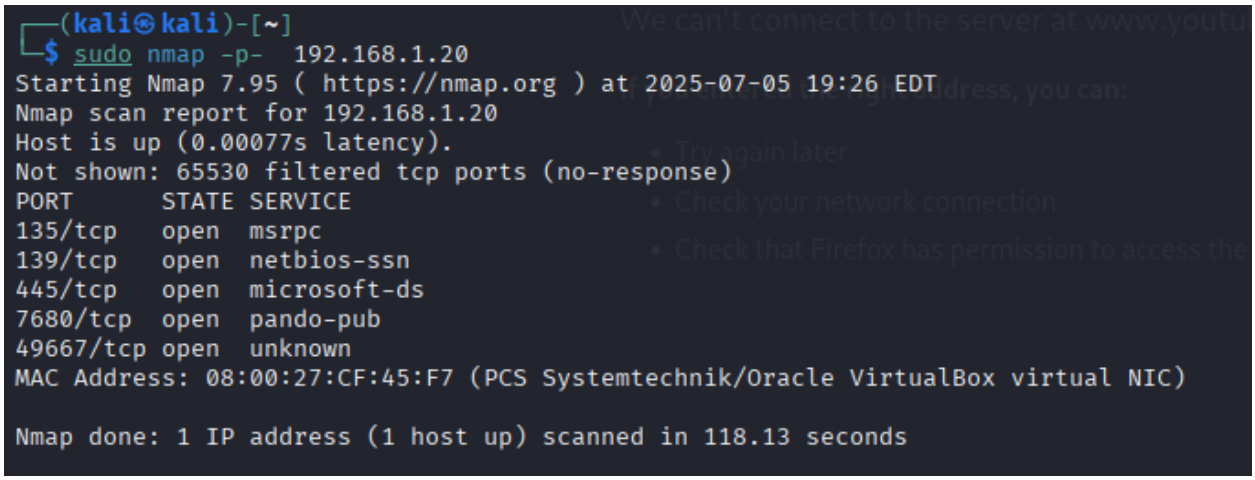

Fig. .10 shows a capture of nmap scanning the IP address of 192.168.1.20 for open ports

Nmap-sS -sV 192.168.1.20 scans for all services running on open ports.

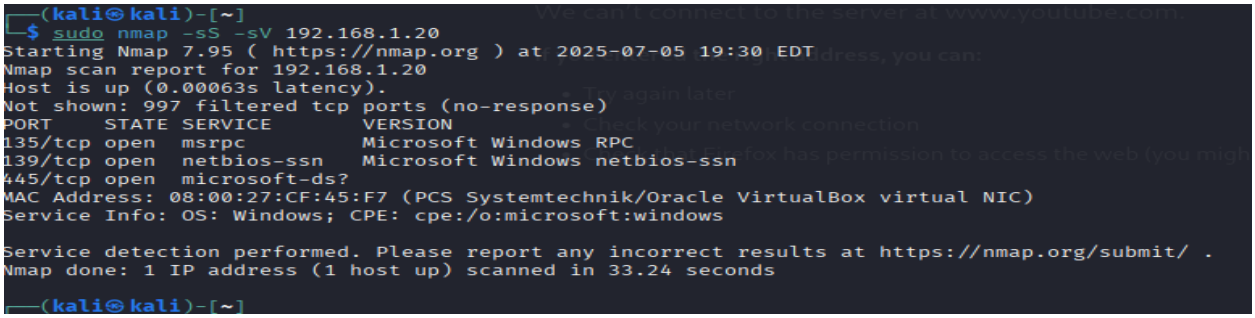

Fig. . .11 shows a capture of nmap scanning the IP address of 192.168.1.20 for all services on open ports