OLD DOMINION UNIVERSITY

CYSE 301 CYBERSECURITY TECHNIQUES AND OPERATIONS

Assignment #3: Sword vs. Shield

Antonio Shields

In this assignment, you will act as an attacker to identify the vulnerabilities in the LAN network and a defender to apply proper countermeasures. You need to provide a screenshot for each task below.

TASK A: SWORD – NETWORK SCANNING (20 + 20 = 40 POINTS)

Power on the listed VMs and complete the following steps from the **External Kali** (you can use either nmap or zenmap to complete the assignment)

- External Kali
- pfSense
- Ubuntu
- Windows Server 2008

Make sure you didn't add/delete any firewall policy before continuing.

1. Use Nmap to profile the basic information about the **subnet** topology (including open ports information, operation systems, etc.) You need to get the **service** and **backend software** information associated with each opening port in each VM.

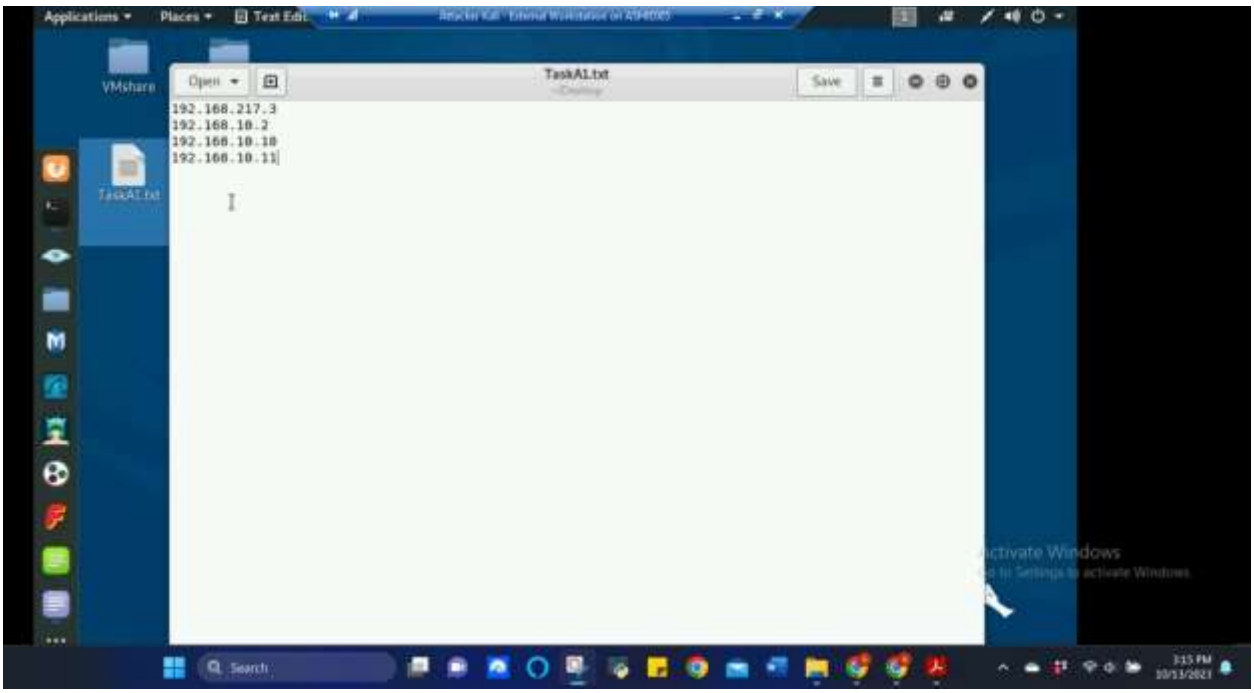

Figure 1 Screenshot of TaskA1.txt being created in Attacker Linux for using the nmap command for Task A.1

The above screenshot shows the .txt file "TaskA1" being created with the following four IP addresses: 192.168.217.3 (attacker kali), 192.168.10.2 (pfSense), 192.168.10.10 (Ubuntu), 192.168.10.11 (Windows Server 2008). This file will be used to search these ip addresses using the nmap command at one time instead of using the nmap command separately for each ip address.

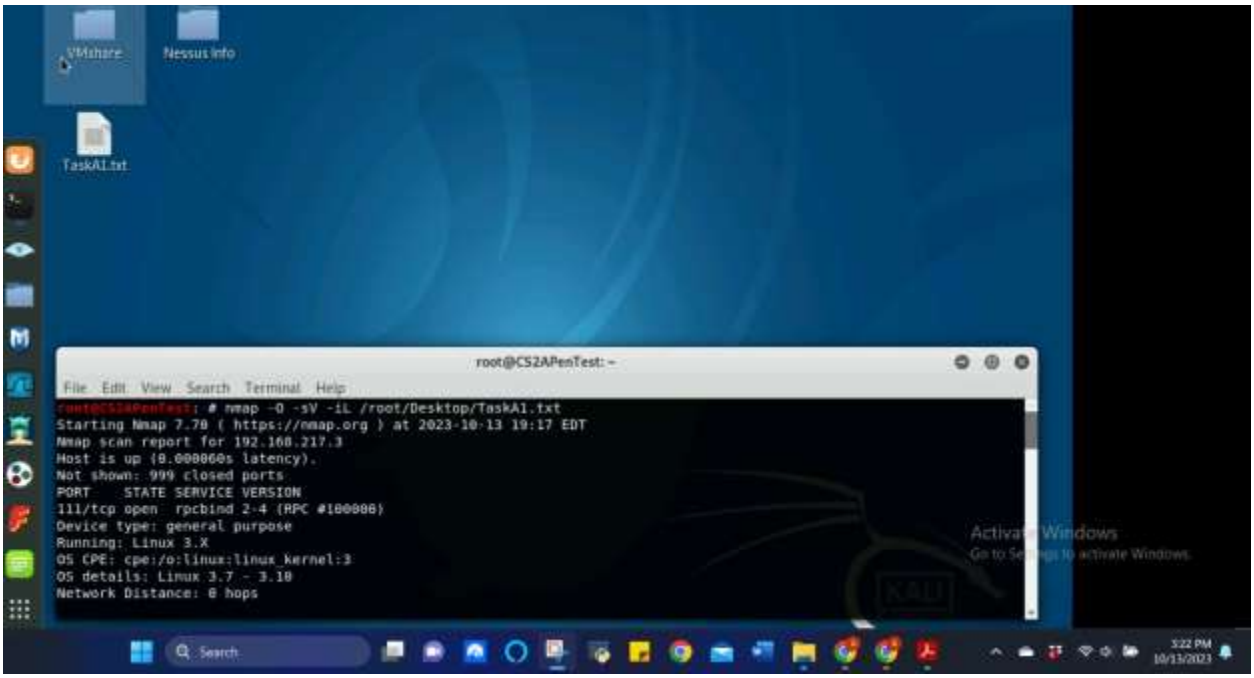

Figure 2 Screenshot of nmap command being used and the results returned in Attacker Linux for Task A.1

The above screenshot shows the nmap command being used to retrieve the subnet topology, service and backend software information from the open ports required for this Task. In the nmap command -O is used for OS detection, -sV is used to determine the service and version running on the port, and -iL allows me to scan from the ip addresses listed in the TaskA1.txt file in the previous screenshot. These are the results from 192.168.217.3 (Attacker Kali) which is the device currently on and initiating nmap.

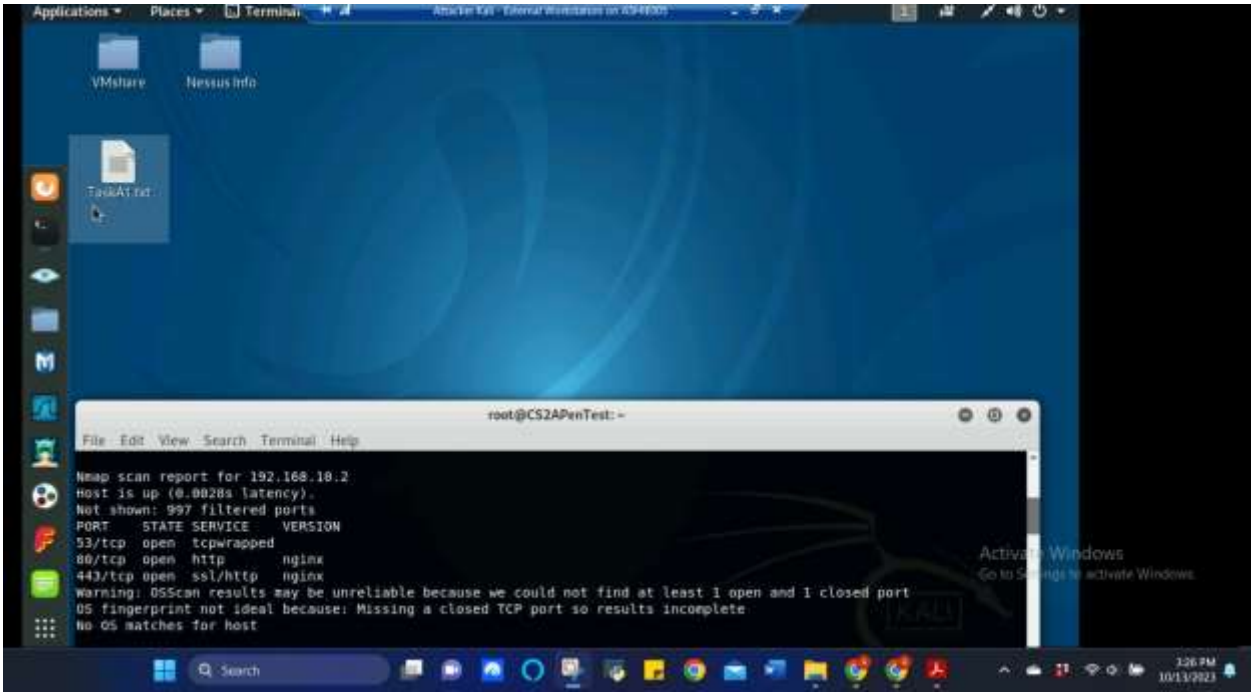

Figure 3 Screenshot of nmap command results for 192.168.10.2 (pfSense) returned in Attacker Linux for Task A.1

The above screenshot shows the nmap command results for 192.168.10.2 (pfSense) and shows the open port information, the service and version. OS version came back with no match.

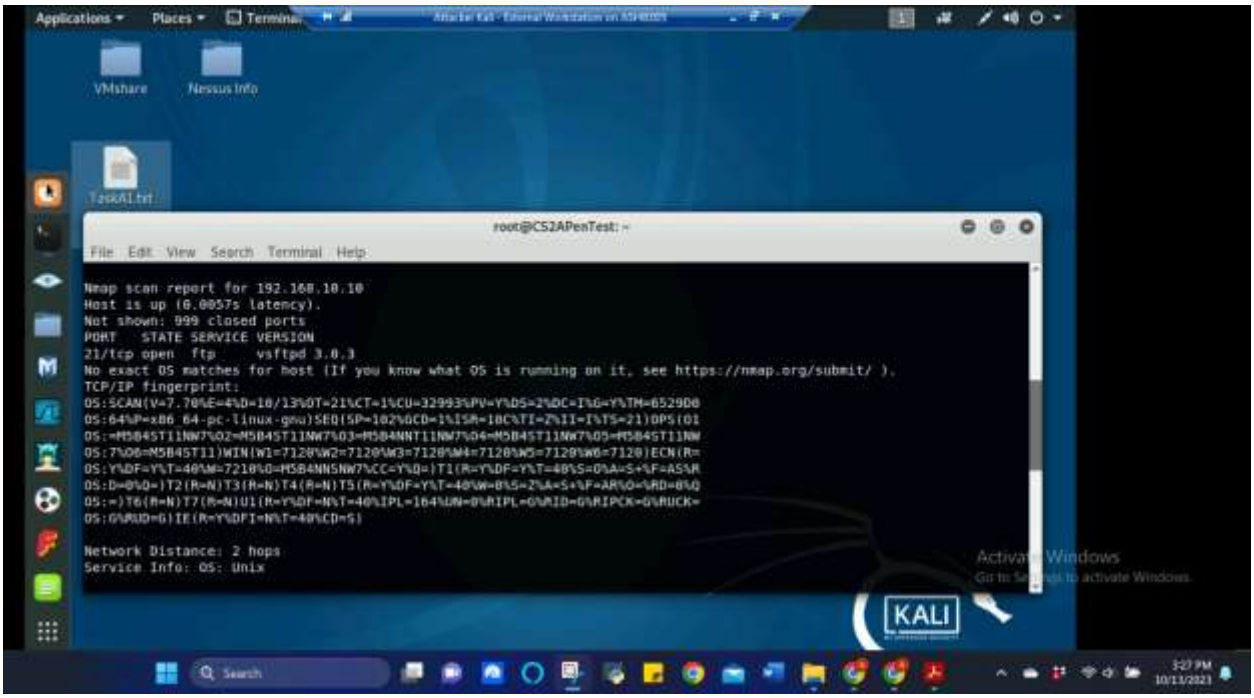

Figure 4 Screenshot of nmap command results for 192.168.10.10 (Ubuntu) returned in Attacker Linux for Task A.1

The above screenshot shows the nmap command results for 192.168.10.10 (Ubuntu) and shows the open port information, the service and version. OS version came back as Unix.

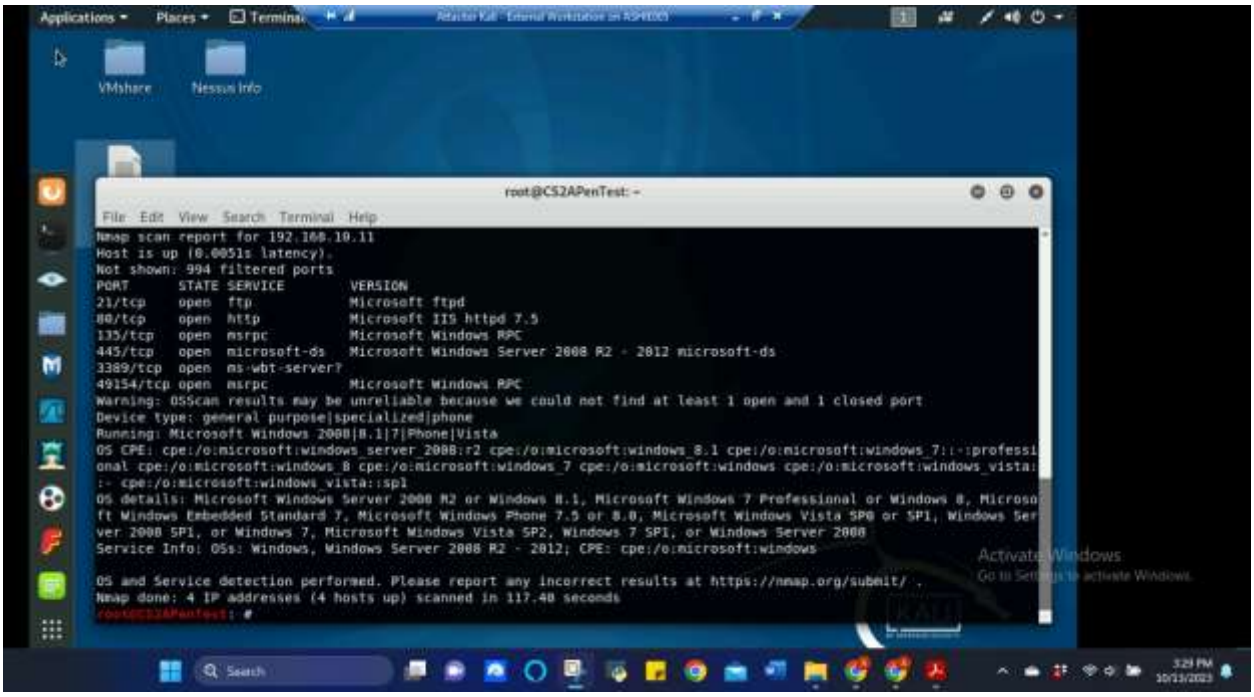

Figure 5 Screenshot of nmap command results for 192.168.10.11 (Windows Server 2008) returned in Attacker Linux for Task A.1

The above screenshot shows the nmap command results for 192.168.10.11 (Windows Server 2008) and shows the open port information, the service and version. OS version came back as Microsoft Windows Server 2008 R2.

2. Run Wireshark in Ubuntu VM while External Kali is scanning the network. Discuss the traffic pattern you observed. What do you find? **Please write a 200-word essay to discuss your findings.**

In running Wireshark in Ubuntu VM while External Kali scanned the network using the nmap command, the results that came back, which can be seen in figures 6, 7, 8, 9, and 10 below were as follows: In Figure 6 it shows, the TCP protocols and DNS protocols that were conducted multiple times between 192.168.10.10 (Ubuntu) and 192.168.10.2 (pfSense) with the three-way handshake between the two ip addresses for TCP and a DNS query sent for DNS. Near the bottom of Figure 6, it shows where 192.168.217.3 (Attacker Kali) and 192.168.10.10 begin TCP protocol. Figure 7 shows the protocol hierarchy statistics during the capture, where 2452 packets total were captured with 5 packets being UDP protocol, 2422 being TCP protocol with 1 of those 2422 being FTP and 31 being DNS, the remaining 25 of the total 2452 came from ICMP protocol. In figure 8 when scrolling further down in the captured packets, it was discovered that interactions between 192.168.217.3 (Attacker Kali) and 192.168.10.10 (Ubuntu) started appearing red in color and "RST" began appearing in the three-way handshake during the TCP protocol. It looks like the connection kept resetting or forcing the connection to terminate multiple times. In figure 9, it shows RST between 192.168.10.10 (Ubuntu) and 192.168.10.2 (pfSense) where the connection is being forcefully terminated and reset and can also be seen are black TCP Retransmission responses from the firewall back to Ubuntu. Figure 10 shows TCP, ICMP, and UDP protocols being used with some RST messages or indicators that ports are being reused or are unreachable. These errors can indicate possible suspicious activity like an outside actor scanning for open ports.

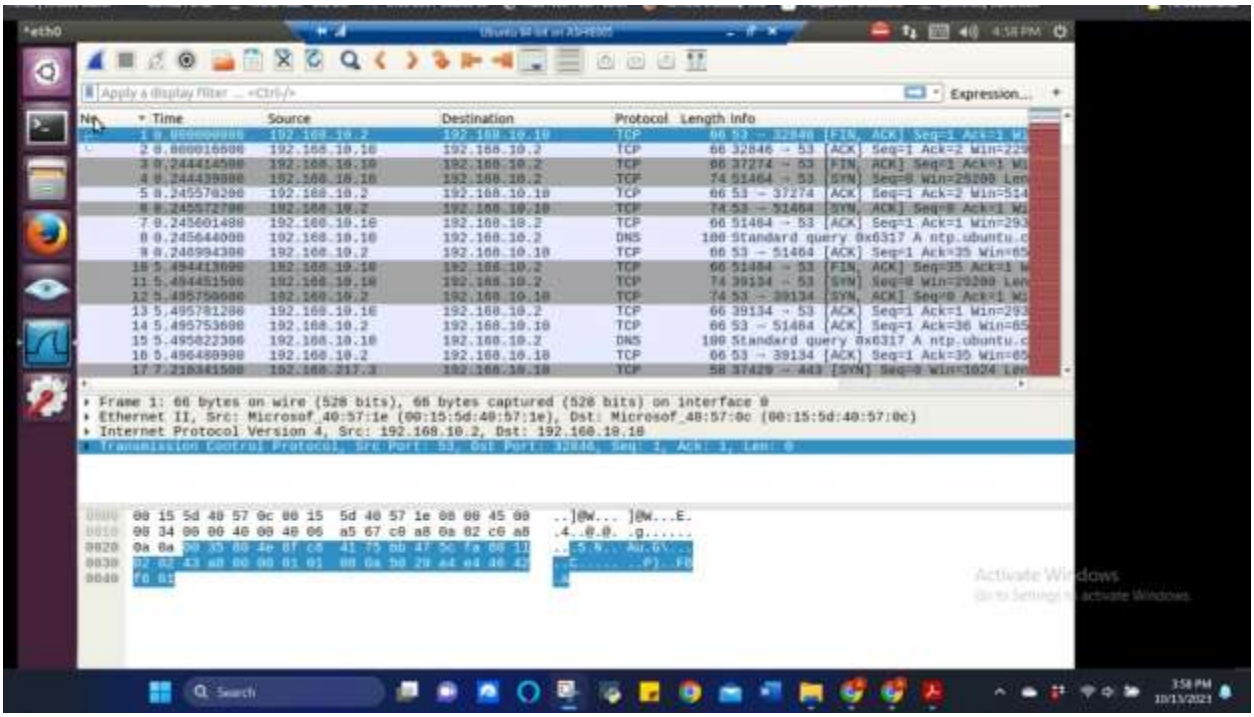

Figure 6 Screenshot of wireshark results ran on 192.168.10.10 (Ubuntu) while 192.168.217.3 (Attacker Linux) scanned the network for Task A.2

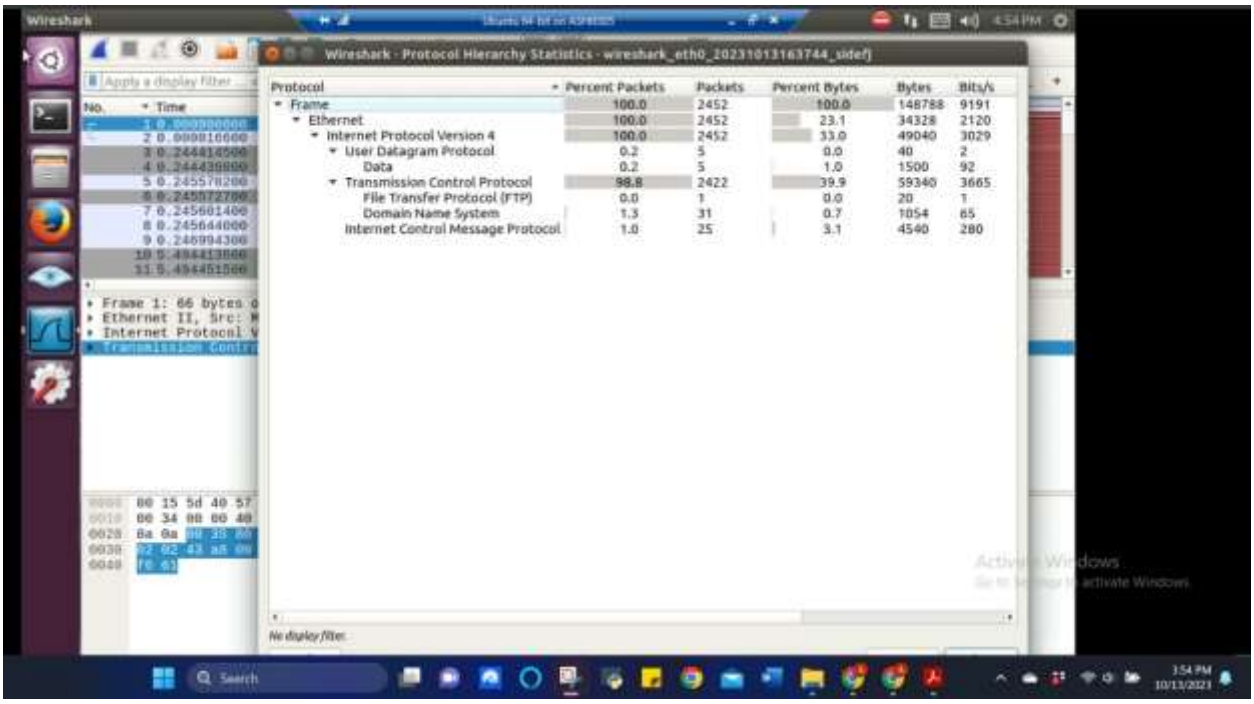

Figure 7 Continued screenshot of wireshark results ran on 192.168.10.10 (Ubuntu) while 192.168.217.3 (Attacker Linux) scanned the network for Task A.2

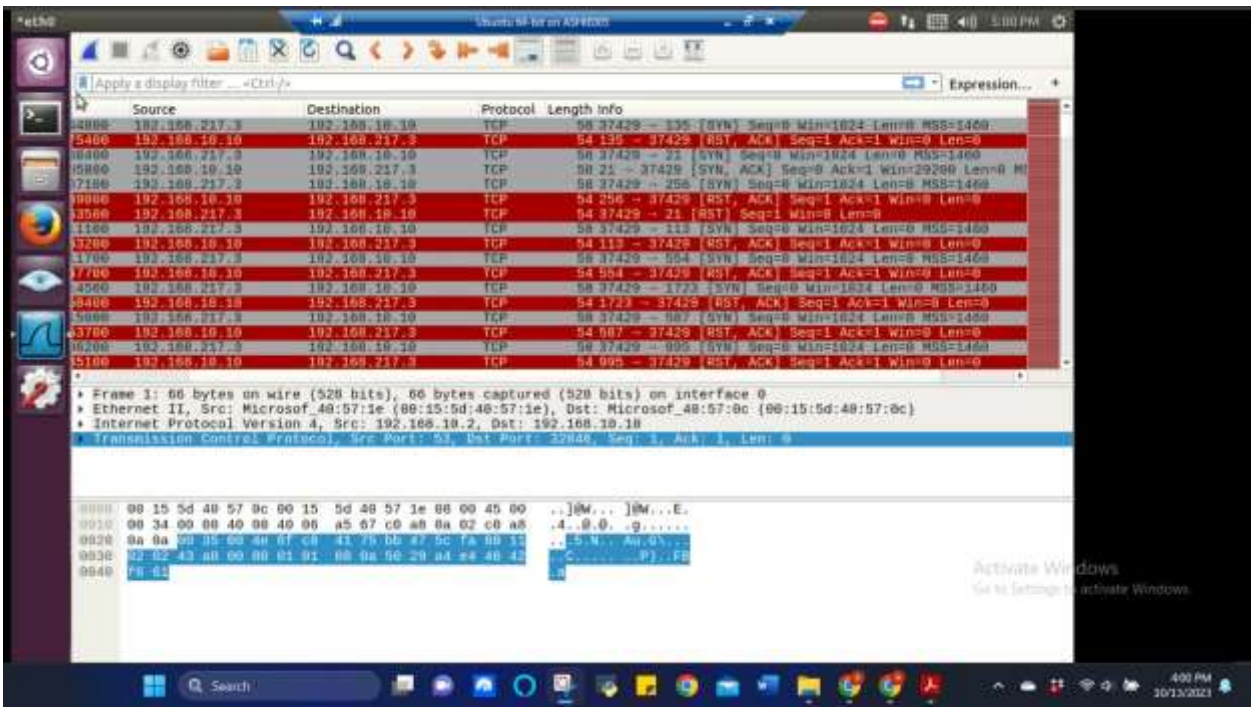

Figure 8 Continued screenshot of wireshark results ran on 192.168.10.10 (Ubuntu) while 192.168.217.3 (Attacker Linux) scanned the network for Task A.2

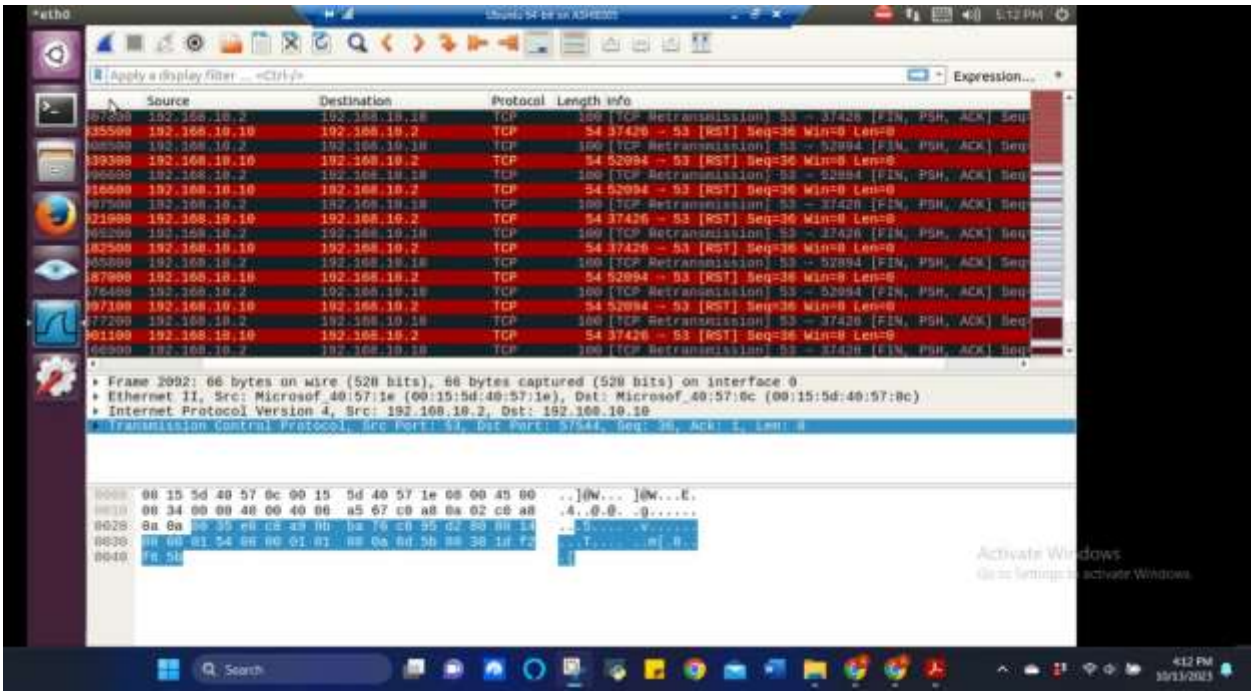

Figure 9 Continued screenshot of wireshark results ran on 192.168.10.10 (Ubuntu) while 192.168.217.3 (Attacker Linux) scanned the network for Task A.2

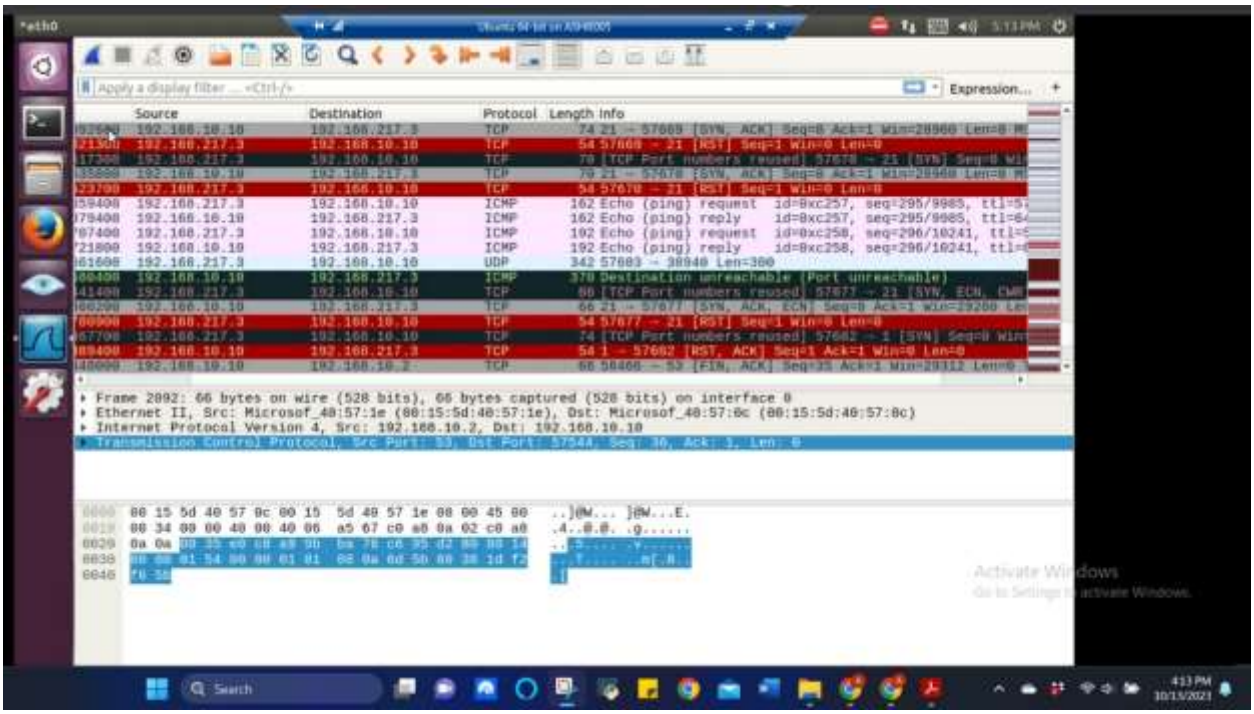

Figure 10 Continued screenshot of wireshark results ran on 192.168.10.10 (Ubuntu) while 192.168.217.3 (Attacker Linux) scanned the network for Task A.2

TASK B. SHIELD – PROTECT YOUR NETWORK WITH FIREWALL (10 + 10 + 20 + 20 = 60 POINTS)

In order to receive full credits, you need to fill the table (add more rows if needed), implement the firewall rule(s), show me the screenshot of your firewall table, and verify the results.

1. Configure the pfSense firewall rule to block the ICMP traffic from External Kali to Ubuntu VM.

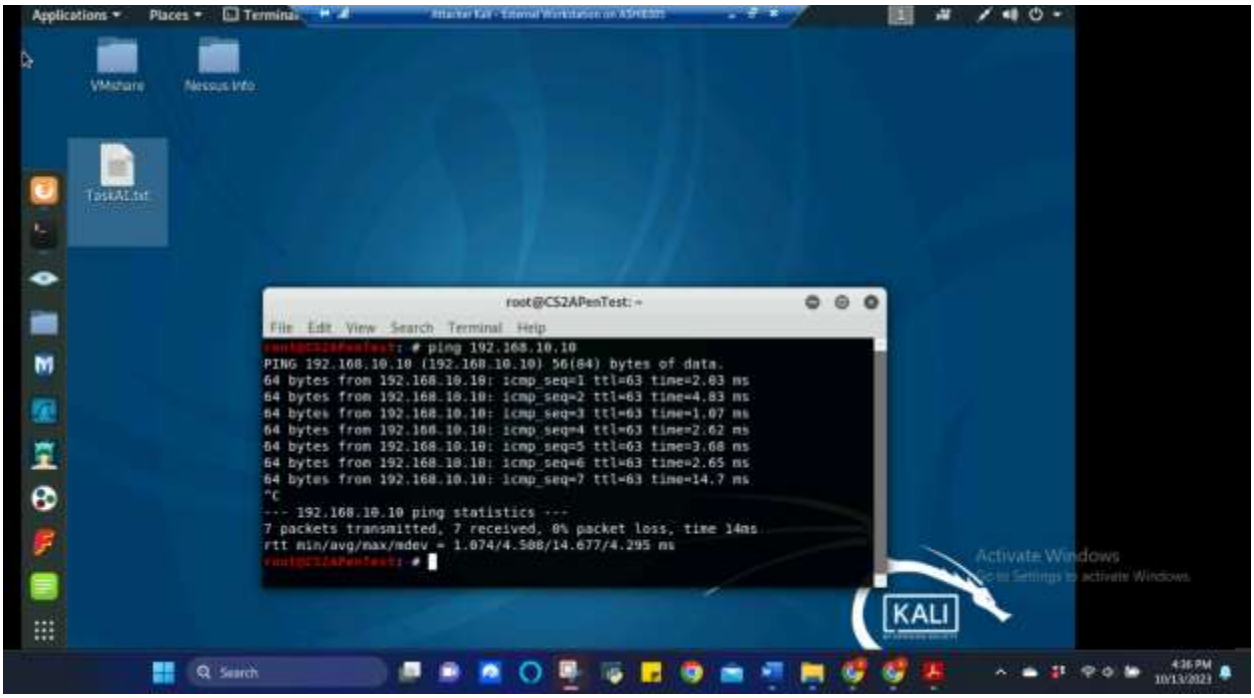

Figure 11 Screenshot of ping of 192.168.10.10 (Ubuntu) on 192.168.217.3 (Attacker Linux) prior to firewall rule set for Task B.1

The above screenshot shows the ping command results for 192.168.10.10 (Ubuntu) on 192.168.217.3 (Attacker Kali) prior to the firewall rule below being applied in pfSense. All 7 packets received.

Rule #	Interface	Action	Source IP	Destination IP	Protocol (port # if appliable)
1	WAN	Block	192.168.217.3	192.168.10.10	ICMP

Figure 12 Table 1 for Firewall Rule for Task B.1

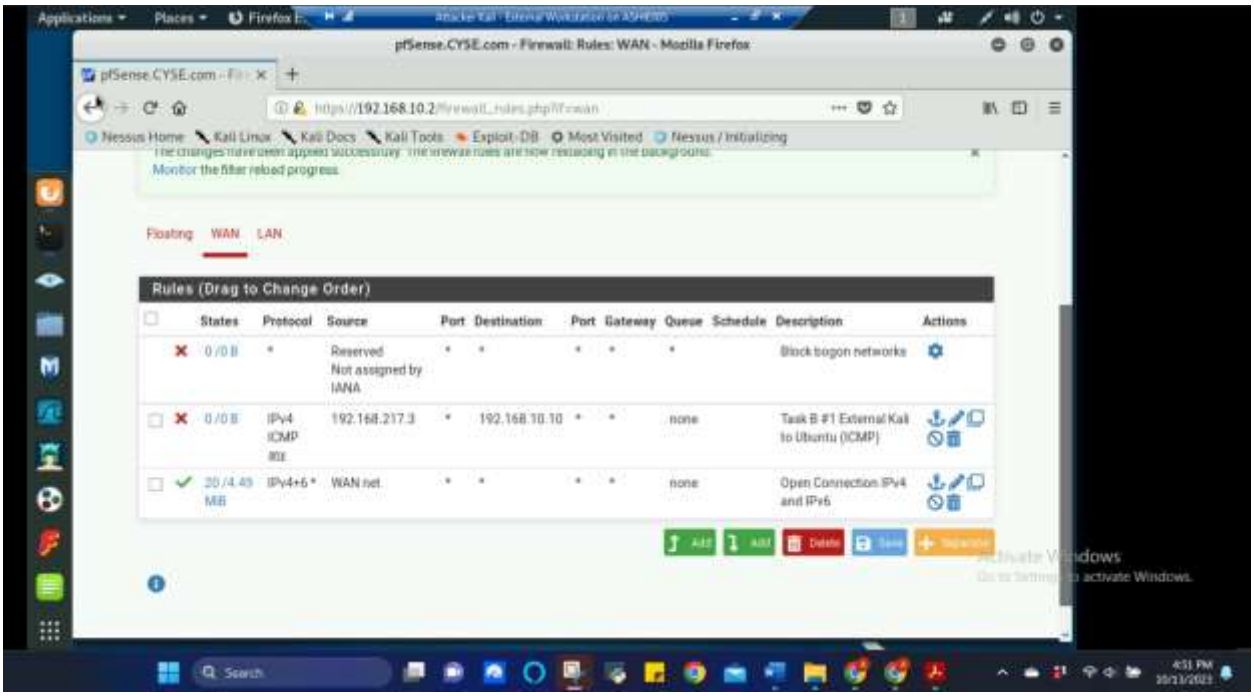

Figure 13 Screenshot of firewall rule from table 1 being created and set for Task B.1

The above screenshot shows the previous table being created and set in pfSense which will be tested below to ensure effectiveness.

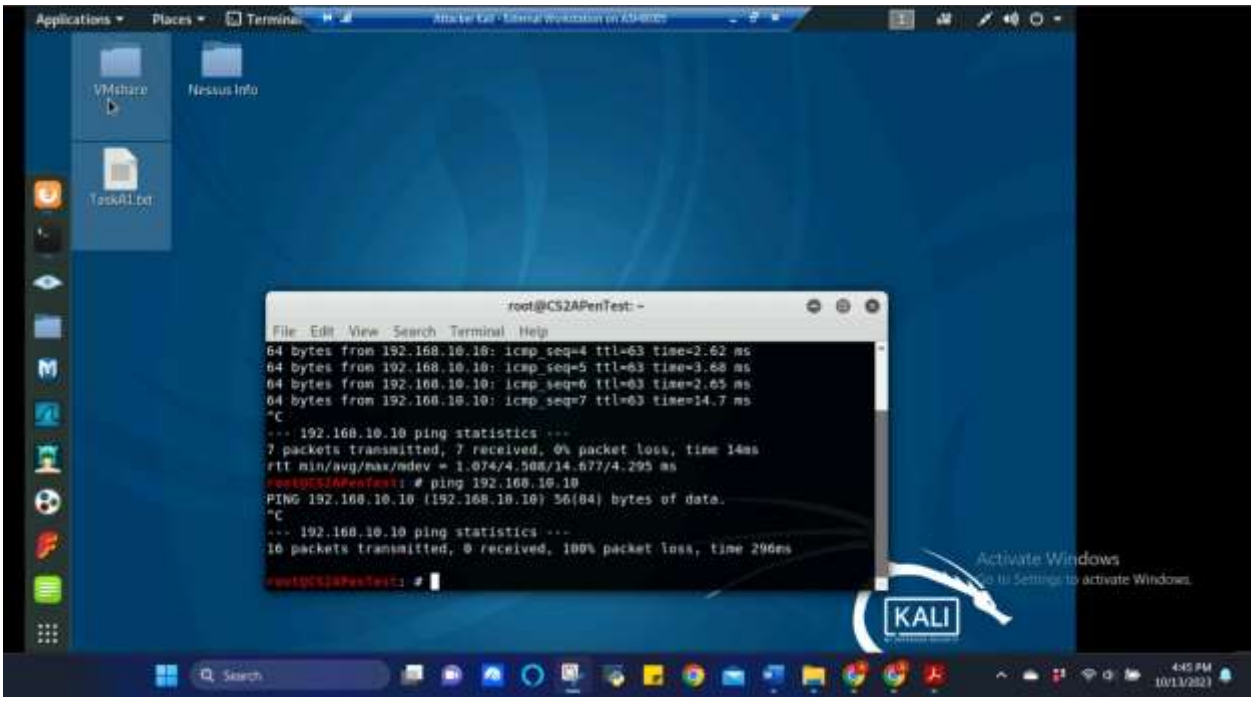

Figure 14 Screenshot of ping of 192.168.10.10 (Ubuntu) on 192.168.217.3 (Attacker Linux) after firewall rule from Table 1 was set and applied for Task B.1

The above screenshot shows the ping command results for 192.168.10.10 (Ubuntu) on 192.168.217.3 (Attacker Kali) after the firewall rule was applied in pfSense. All 16 packets were lost.

2. Clear the previous firewall policies and configure the pfSense firewall to block all ICMP traffic from External Kali to the LAN side.

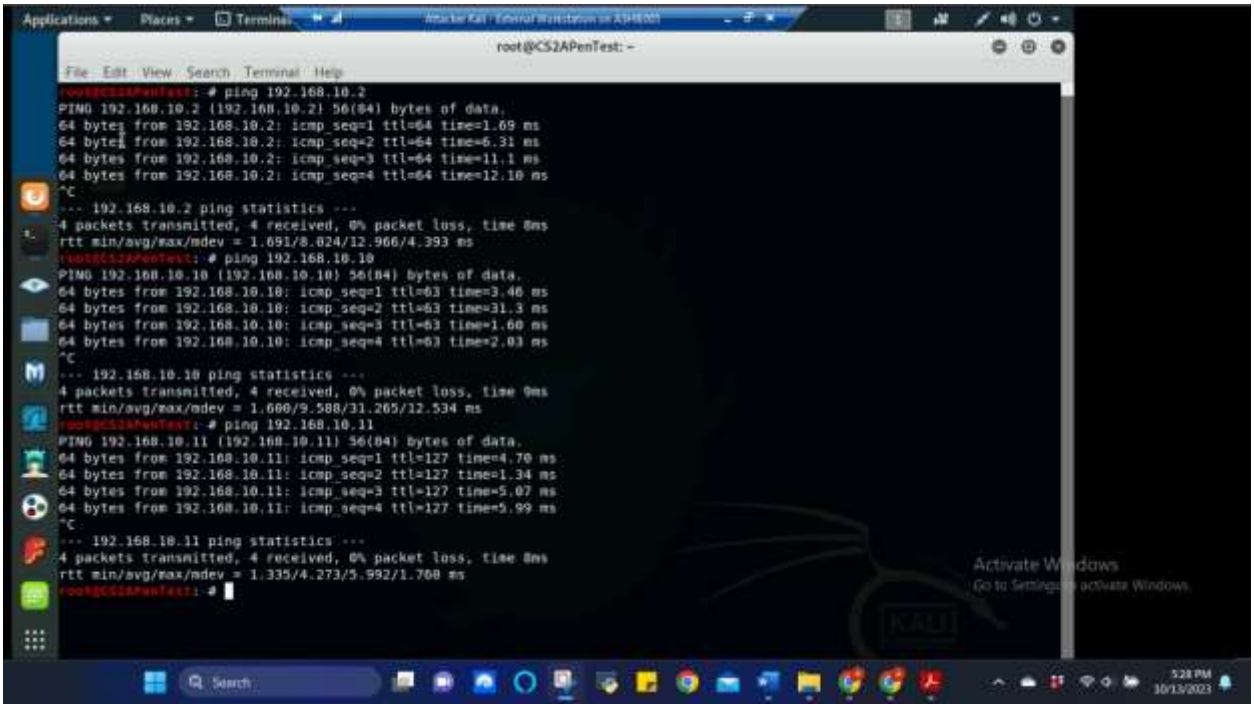

Figure 15 Screenshot of pinging of LAN Side, which consists of 192.168.10.2 (pfSense), 192.168.10.10 (Ubuntu), and 192.168.10.11 (Windows Server 2008) on 192.168.217.3 (Attacker Linux) prior to firewall rule set for Task B.2

The above screenshot shows the ping command results for the entire LAN side consisting of 192.168.10.2 (pfSense), 192.168.10.10 (Ubuntu), and 192.168.10.11 (Windows Server 2008) on 192.168.217.3 (Attacker Kali) prior to the firewall rule below being applied in pfSense. 4 packets on each LAN Side ip address were received.

Rule #	Interface	Action	Source IP	Destination IP	Protocol (port # if appliable)
1	WAN	Block	192.168.217.3	LAN Side	ICMP

Figure 16 Table 2 for Firewall Rule for Task B.2

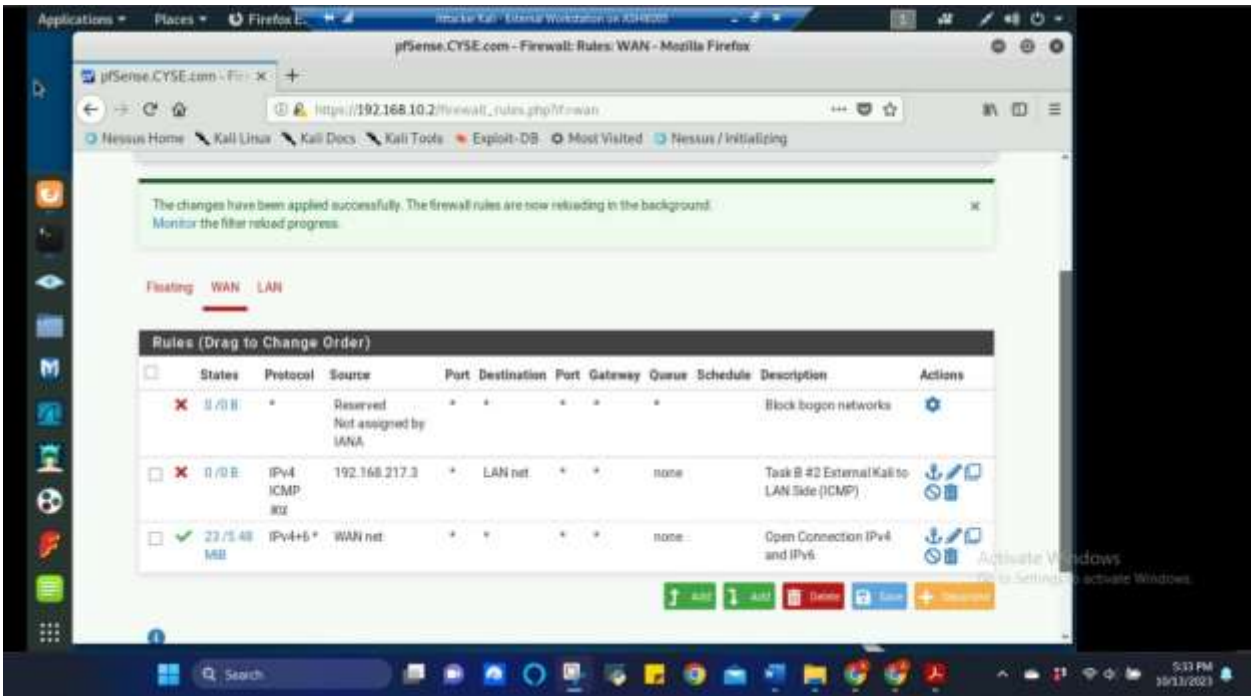

Figure 17 Screenshot of firewall rule from table 2 being created and set for Task B.2

The above screenshot shows the previous table being created and set in pfSense which will be tested below to ensure effectiveness.

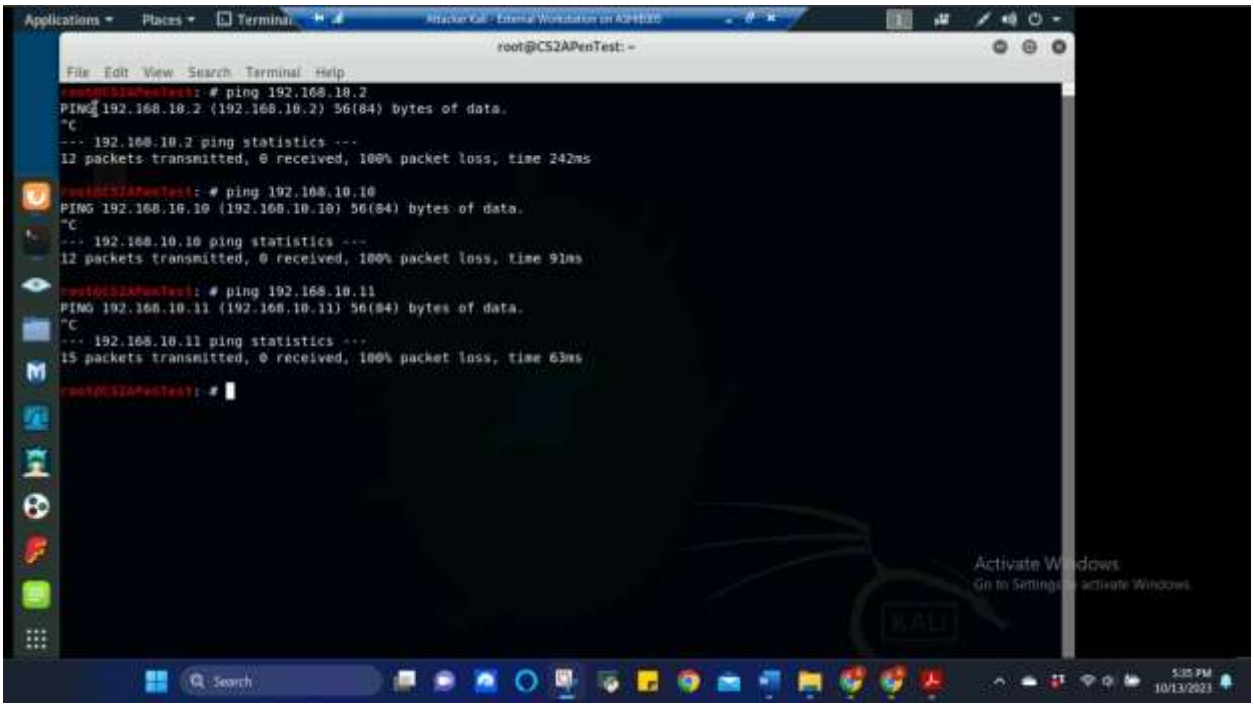

Figure 18 Screenshot of pinging of LAN Side, which consists of 192.168.10.2 (pfSense), 192.168.10.10 (Ubuntu), and 192.168.10.11 (Windows Server 2008) on 192.168.217.3 (Attacker Linux) after firewall rule from table 2 was set and applied for Task B.2

The above screenshot shows the ping command results for the entire LAN side consisting of 192.168.10.2 (pfSense), 192.168.10.10 (Ubuntu), and 192.168.10.11 (Windows Server 2008) on 192.168.217.3 (Attacker Kali) after the firewall rule was applied in pfSense. All packets sent to each LAN Side ip address were lost (12 packets to pfSense, 12 packets to Ubuntu, and 15 packets to Windows Server 2008).

3. Clear the previous firewall policies and configure the pfSense firewall to block ALL traffic from External Kali to the LAN side, except for the FTP protocol towards Windows Server 2008.

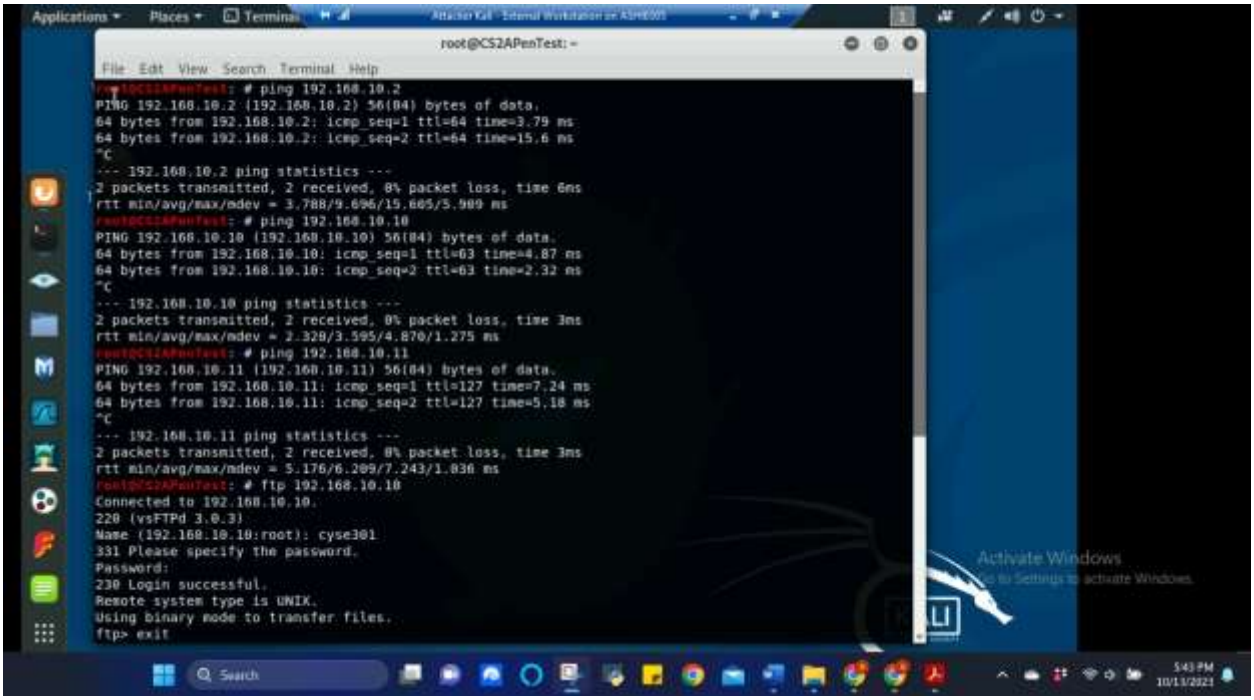

Figure 19 Screenshot of pinging of LAN Side, which consists of 192.168.10.2 (pfSense), 192.168.10.10 (Ubuntu), and 192.168.10.11 (Windows Server 2008) and ftp command on 192.168.10.10 (Ubuntu) on 192.168.217.3 (Attacker Linux) prior to firewall rule set for Task B.3

The above screenshot shows the ping command results for the entire LAN side consisting of 192.168.10.2 (pfSense), 192.168.10.10 (Ubuntu), and 192.168.10.11 (Windows Server 2008) and the ftp command was used on 192.168.10.10 (Ubuntu) on 192.168.217.3 (Attacker Kali) prior to the firewall rule below being applied in pfSense. 2 packets on each LAN Side ip address were received and the ftp was logged in successfully.

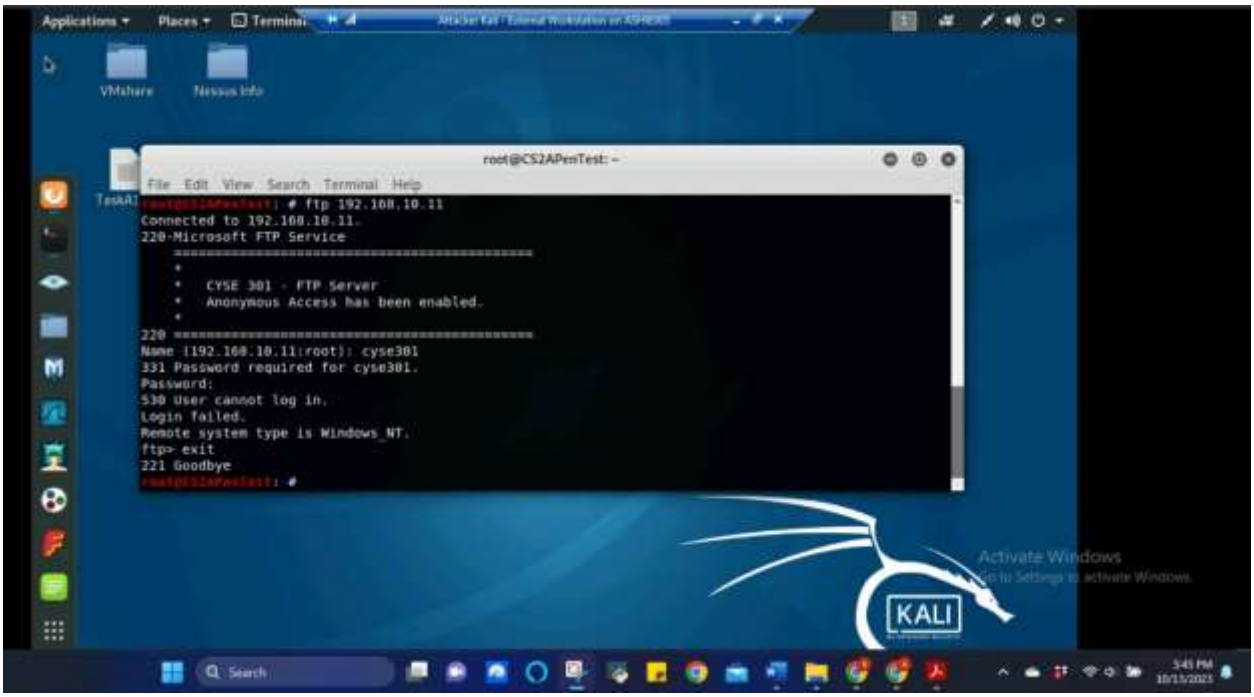

Figure 20 Screenshot of using the ftp command for 192.168.10.11 (Windows Server 2008) on 192.168.217.3 (Attacker Linux) prior to firewall rule set for Task B.3

The above screenshot shows the the ftp command results 192.168.10.11 (Windows Server 2008) on 192.168.217.3 (Attacker Kali) prior to the firewall rule below being applied in pfSense. This was unable to fit in previous screenshot. Connection was established with 192.168.10.11.

Rule #	Interface	Action	Source IP	Destination IP	Protocol (port # if appliable)
1	WAN	Pass	192.168.217.3	192.168.10.11	FTP/21
2	WAN	Block	192.168.217.3	All	All

Figure 21 Table 3 for Firewall Rule for Task B.3

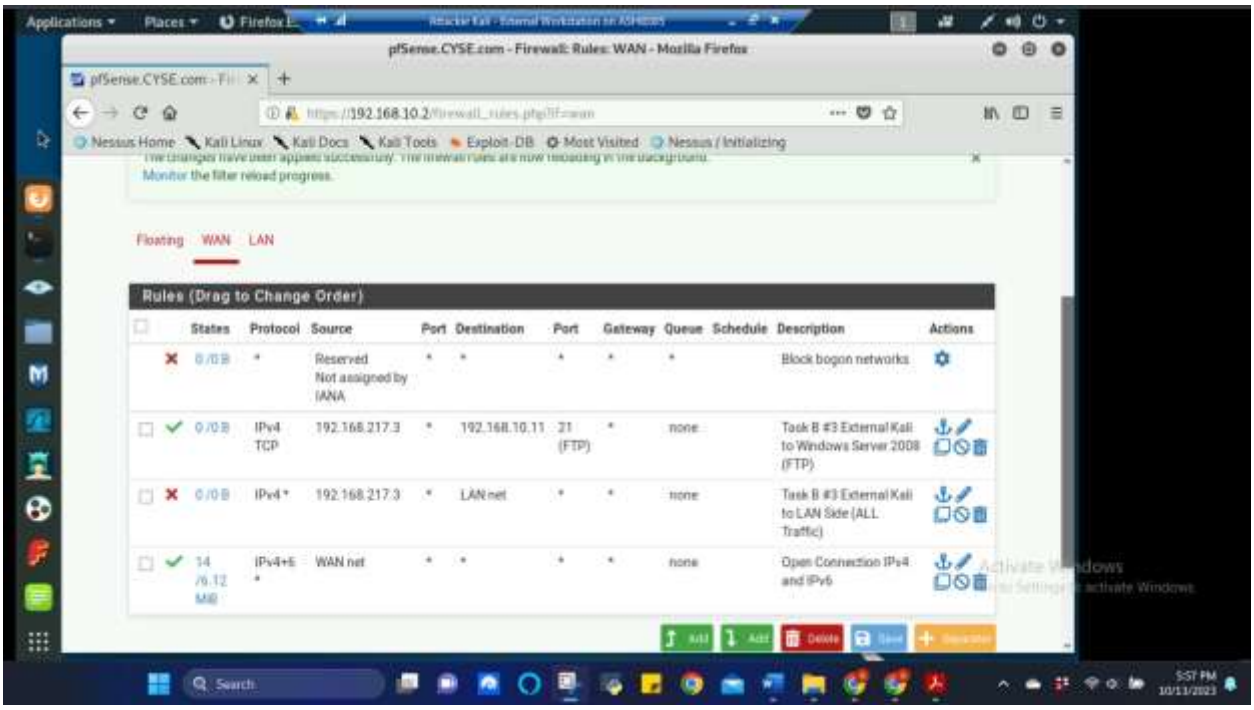

Figure 22 Screenshot of firewall rule from table 3 being created and set for Task B.3

The above screenshot shows the previous table being created and set in pfSense which will be tested below to ensure effectiveness.

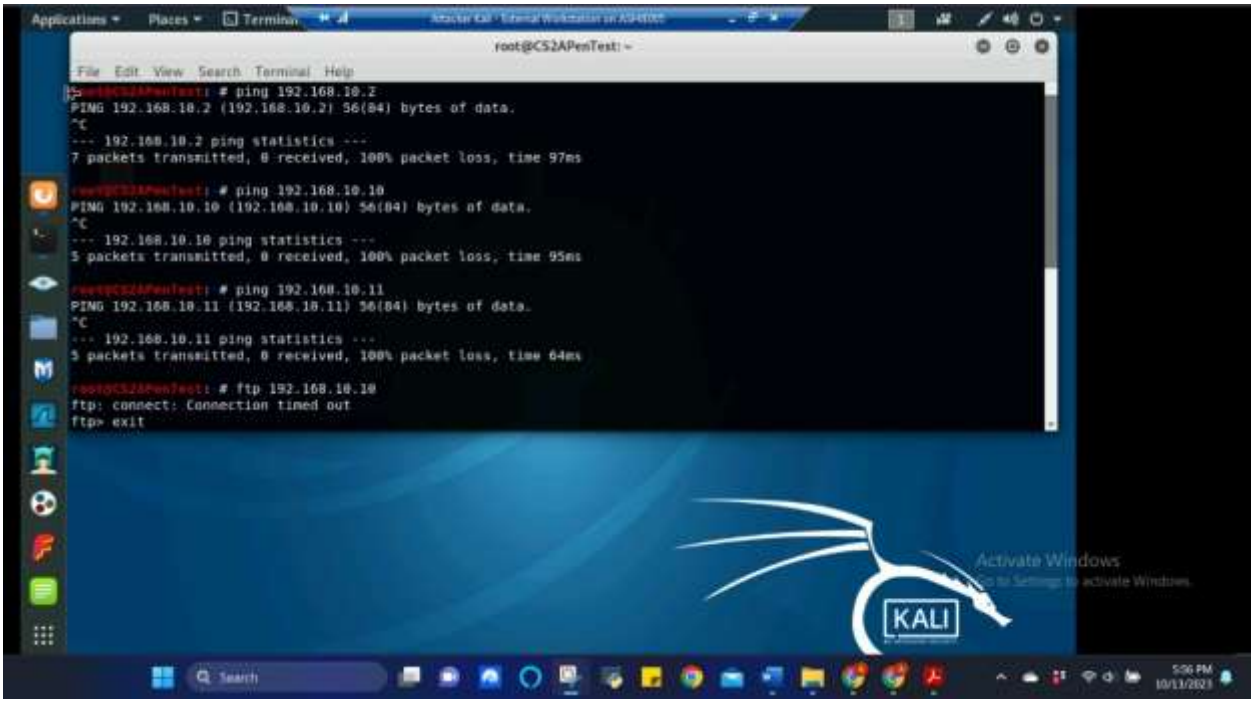

Figure 23 Screenshot of pinging of LAN Side, which consists of 192.168.10.2 (pfSense), 192.168.10.10 (Ubuntu), and 192.168.10.11 (Windows Server 2008) on 192.168.217.3 (Attacker Linux), also ftp command on 192.168.10.10 (Ubuntu) after firewall rule from table 3 was set and applied for Task B.3

The above screenshot shows the ping command results for the entire LAN side consisting of 192.168.10.2 (pfSense), 192.168.10.10 (Ubuntu), and 192.168.10.11 (Windows Server 2008) and the ftp command was used on 192.168.10.10 (Ubuntu) on 192.168.217.3 (Attacker Kali) after the firewall rule was applied in pfSense. All packets on each LAN Side ip address were lost (7 packets for pfSense, 5 packets for Ubuntu, and 5 packets for Windows Server 2008) and the ftp timed out connection.

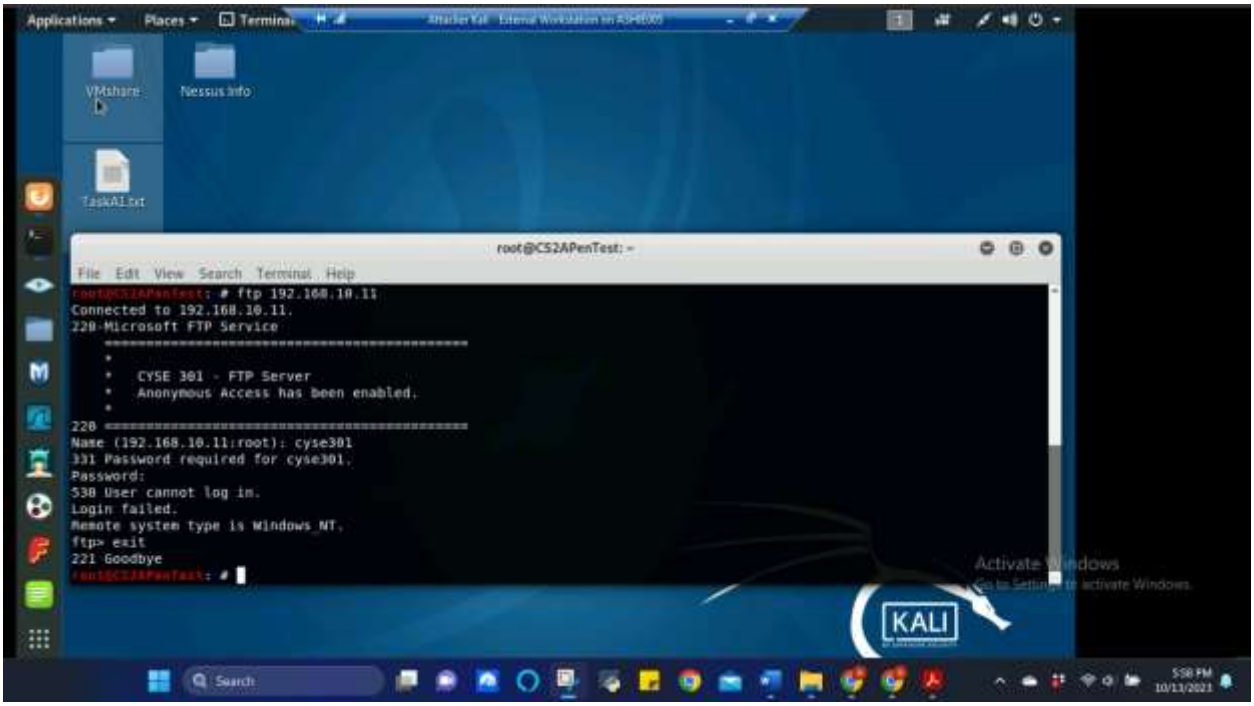

Figure 24 Screenshot of ftp command for 192.168.10.11 (Windows Server 2008) on 192.168.217.3 (Attacker Linux) after firewall rule from table 3 was set and applied for Task B.3

The above screenshot shows the the ftp command results 192.168.10.11 (Windows Server 2008) on 192.168.217.3 (Attacker Kali) after the firewall rule was applied in pfSense. The connection was still able to be established with 192.168.10.11.

4. Keep the firewall policies you created in Task B.3 and repeat Task A.1. What's the difference?

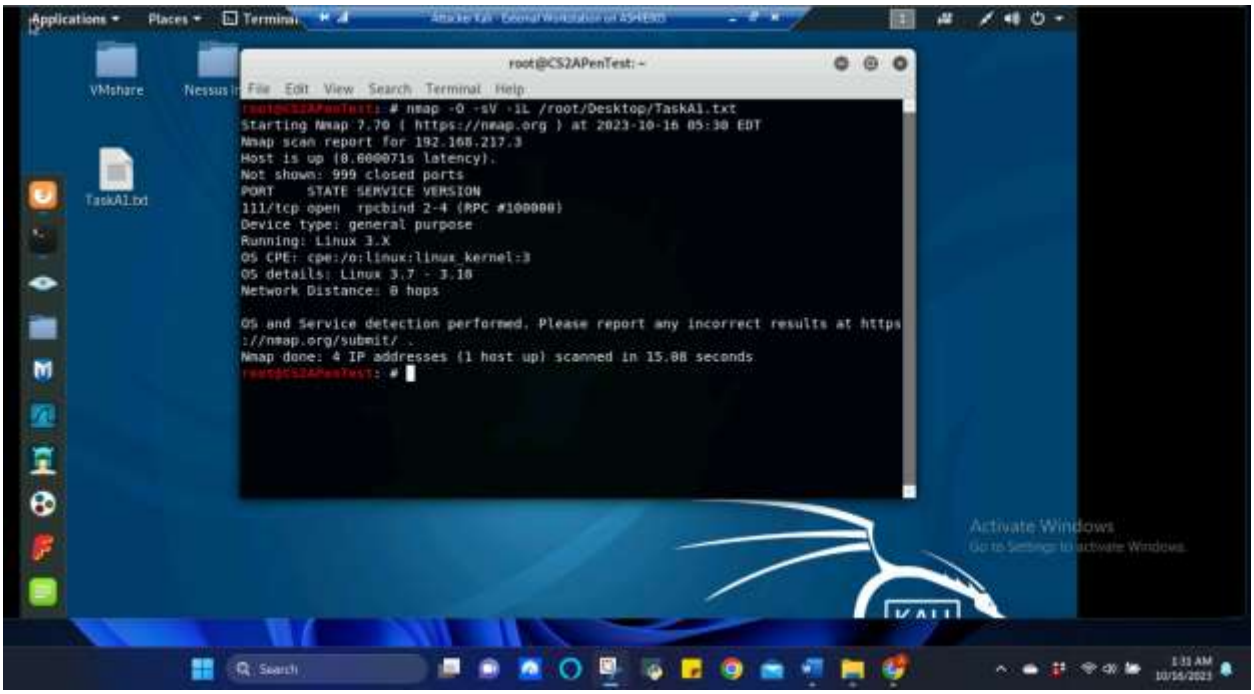

Figure 25 Screenshot of nmap command from Task A.1 being used with firewall rules from Task B.3 still in place and the results returned in Attacker Linux for Task B.4

The above screenshot shows the nmap command that was used in Task A.1 but with the firewall rules applied from Task B.3. The results only showed information for 192.168.217.3 (Attacker Kali), which is the machine that is currently running. It shows that 4 IP addresses were checked but only the host is up. With this lack of information, nmap was ran again, but using the LAN side ip addresses separately below.

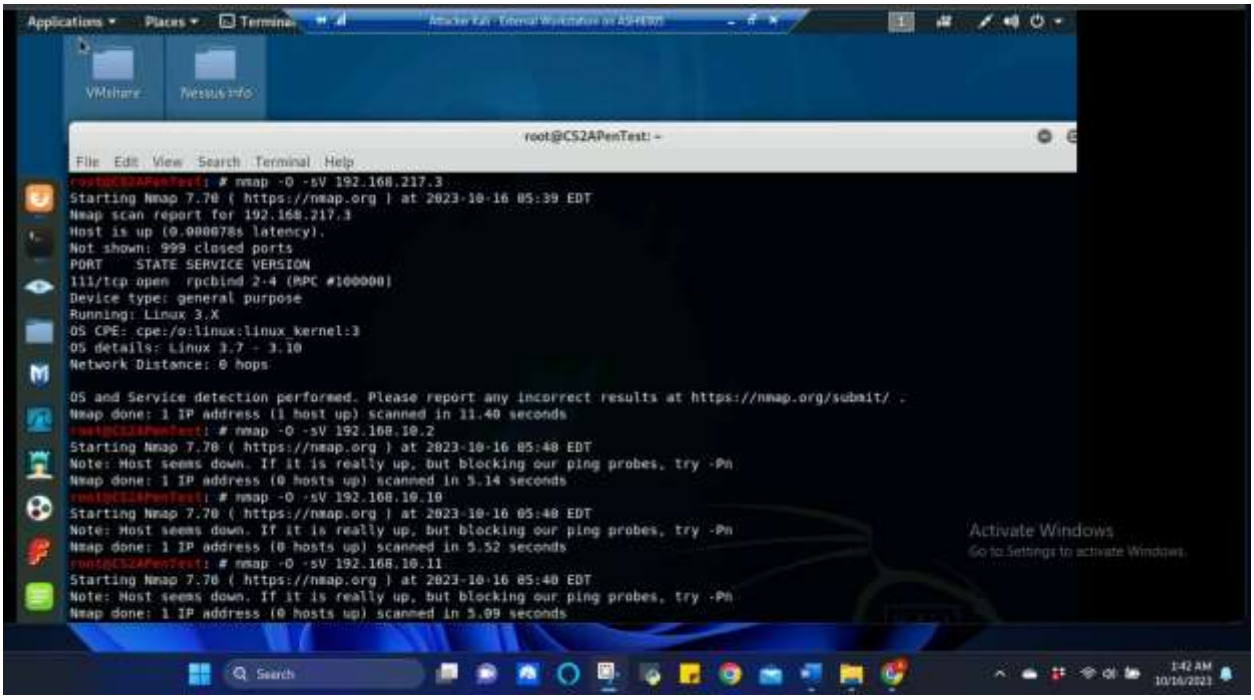

Figure 26 Screenshot of nmap command from Task A.1 separately applied to each ip address being used with firewall rules from Task B.3 still in place and the results returned in Attacker Linux for Task B.4

The above screenshot shows the individual results of each LAN side ip addresses and gave more details than the previous screenshot. This time when the nmap was ran, the LAN side responded “Host seems down. If it is really up, but blocking our ping probes, try -Pn”, so firewall rules blocked nmap from working on the LAN side ip addresses.

Extra credit (15 points): Use NESSUS to enumerate the security vulnerabilities of Microsoft Windows Server 2008 VM in the CCIA network.

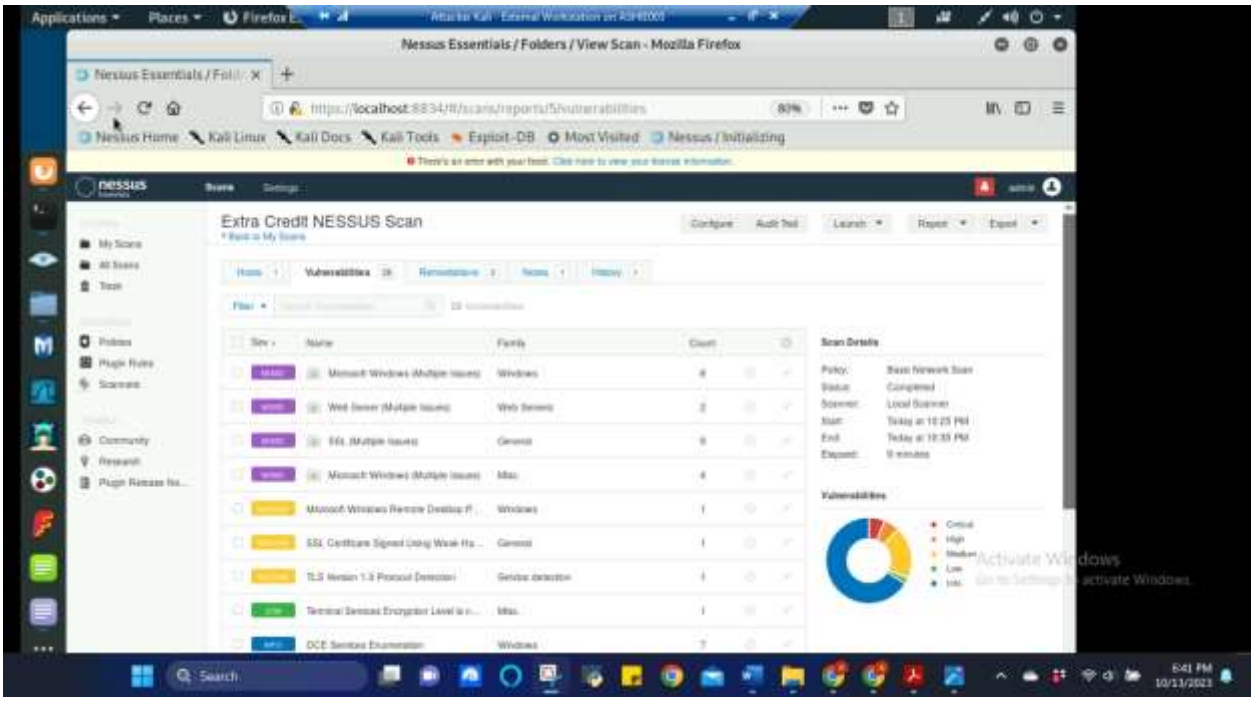

Figure 27 Screenshot from Nessus scanning of 192.168.10.11 (Windows Server 2008) to show vulnerabilities

The above screenshot shows the Nessus vulnerability scan on 192.168.10.11 (Windows Server 2008). The vulnerabilities came back 28 total and varies in the severity. The screenshot below shows one of the mixed serverities and how it is subdivided.

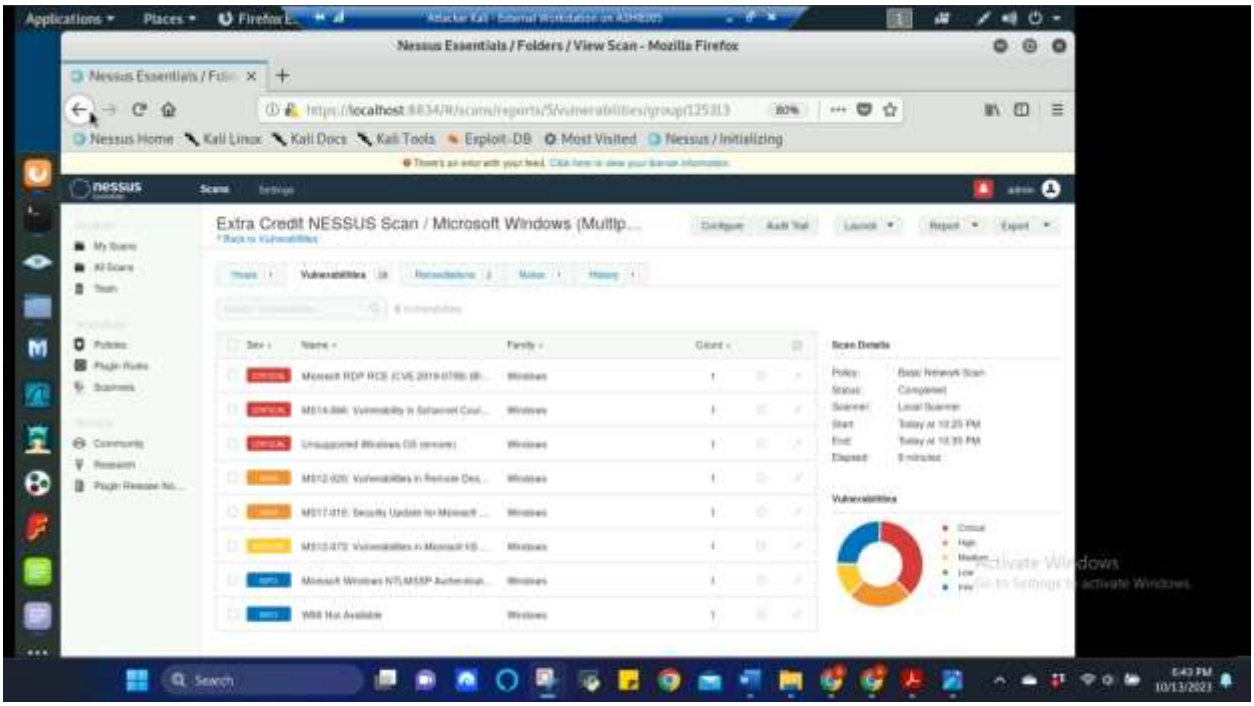

Figure 28 Screenshot of one of the mixed-severity vulnerabilities from the Nessus scanning of 192.168.10.11 (Windows Server 2008).

The above screenshot shows the first mixed vulnerability in the previous screenshot and how it is subdivided. This vulnerability shows 3 critical issues, 2 high issues, and 1 medium issue as an example.