

OLD DOMINION UNIVERSITY

CYSE 301 CYBERSECURITY TECHNIQUES AND OPERATIONS

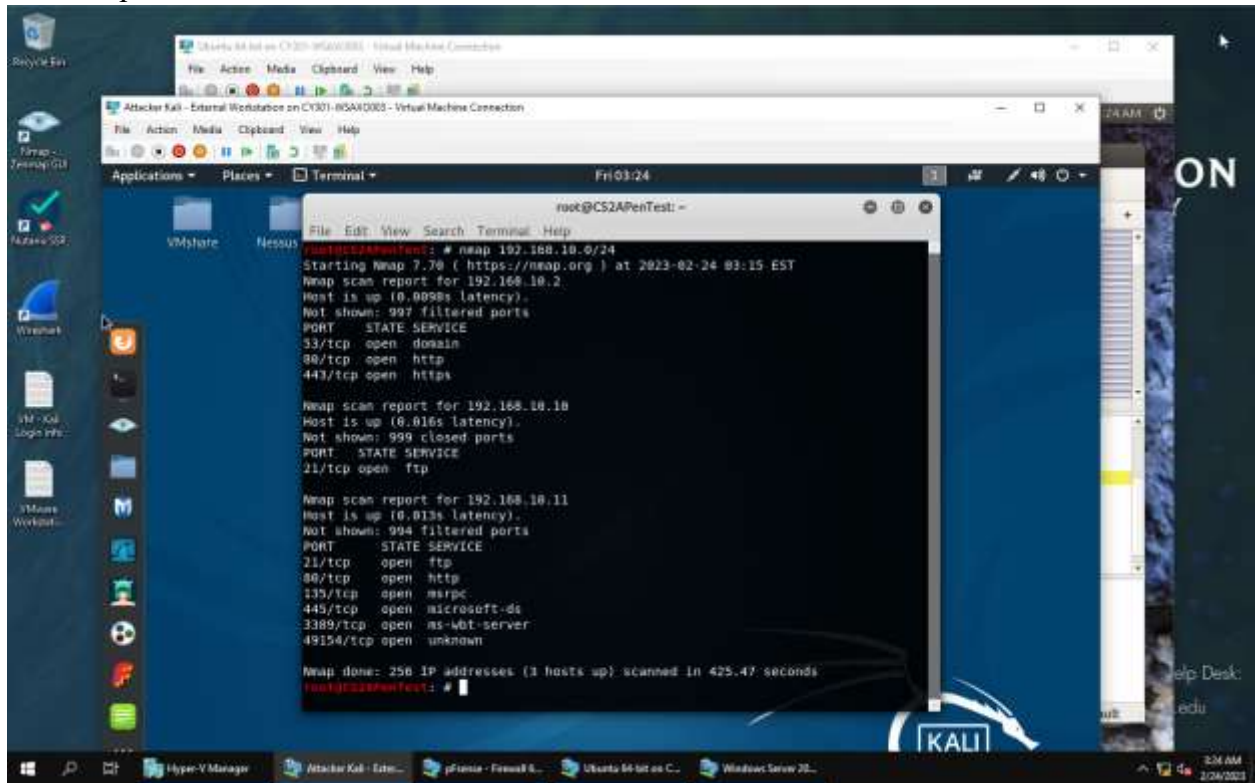
Assignment #1 Sword vs. Shield

William Saxon

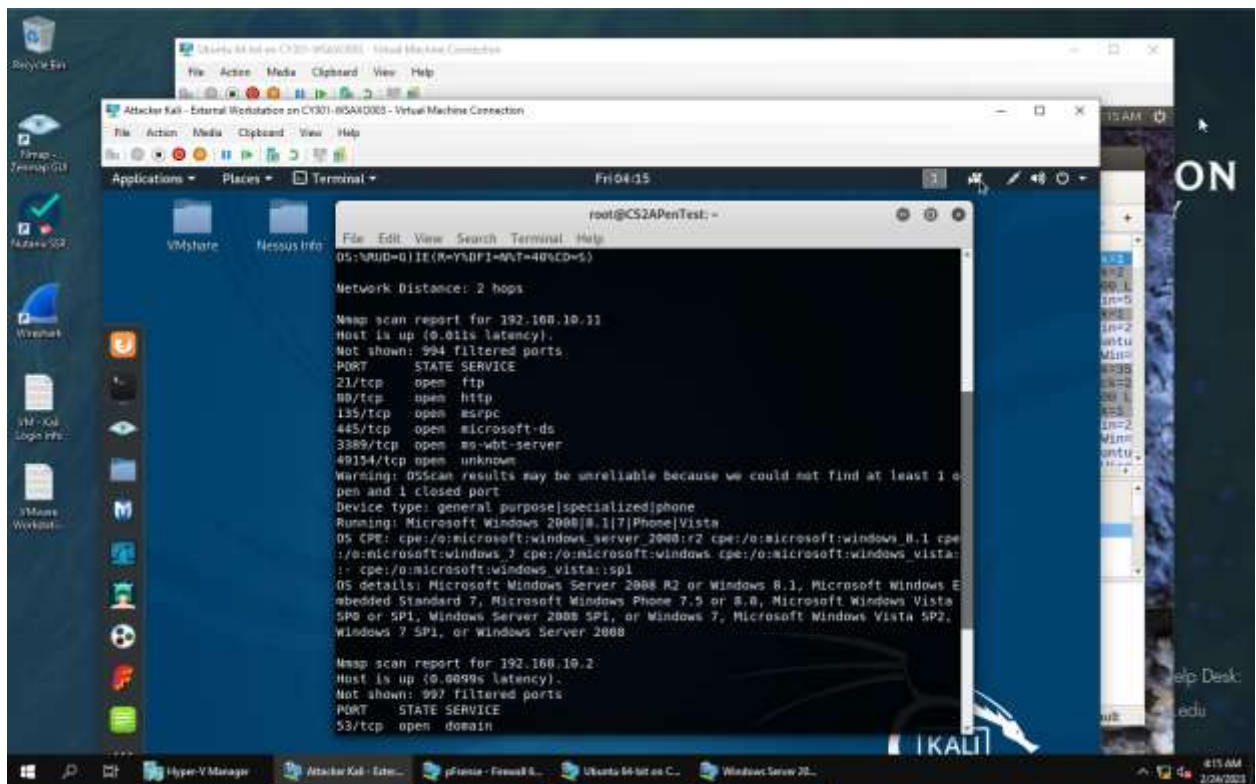
01216157

TASK A

1. Use nmap on subnet and show service and backend software



I scanned the subnet by using the netmask from the Lap 2 PDF



So I used nmap -O to check the OSs of the three systems and 192.168.10.10 and 192.168.10.2 had “No OS matches for host”. On 192.168.10.11 a bunch of Windows OSs showed up as possibilities.

2. 200 word essay

When the nmap subnet scan started there was a lot of ARP requests.

It looked like it was giving an ARP request to every address in the subnet range. It was doing a sweep to see what was on the network. I think it was trying to get the addresses of all the systems on the network.

Then I saw a lot of TCP requests. It looks like it was checking every port on 192.168.10.10 because 192.168.217.3 and 192.168.10.10 were going back and forth on a lot of ports. I noticed that all the SYN requests was from 192.168.217.3 on port 38122 and a lot ACK responses was replying.

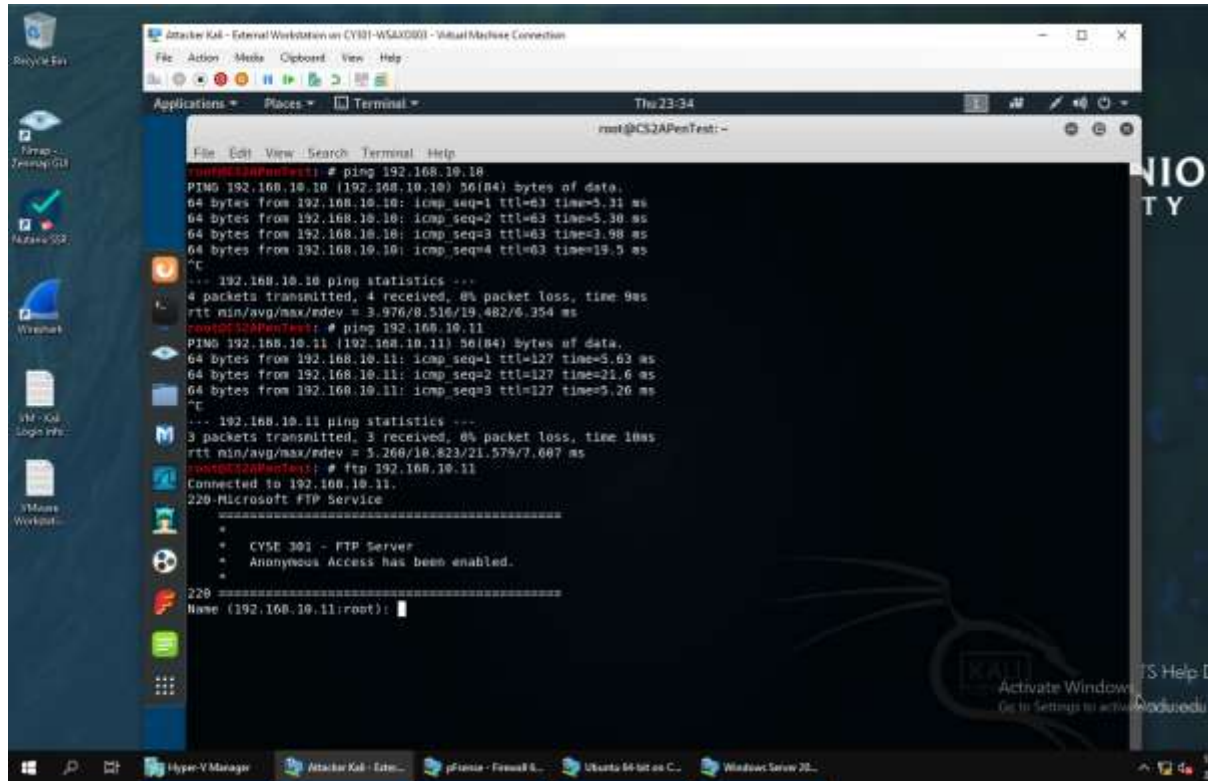
The next thing that stood out to me was a ton of SYN requests from 192.168.217.3:65069 to 192.168.10.11 on a lot very large number of ports. I think it was trying to start the TCP handshake with every port it could to see if there were any openings but there wasn’t any replies for a while. All I saw was SYN requests from 192.168.217.3:65069 and no ACK replies.

That was everything that stood out to me that Wireshark captured. When I used nmap -sV and nmap -O It looked about the same as the nmap subnet scan.

TASK B

NOTE: I saw on Discord that you told someone you wanted a before and after the rules picture. I did Task B a week before I saw that so I went back and added before the rules picture. That's why the time is off.

0. Tests before the Firewall Rules.



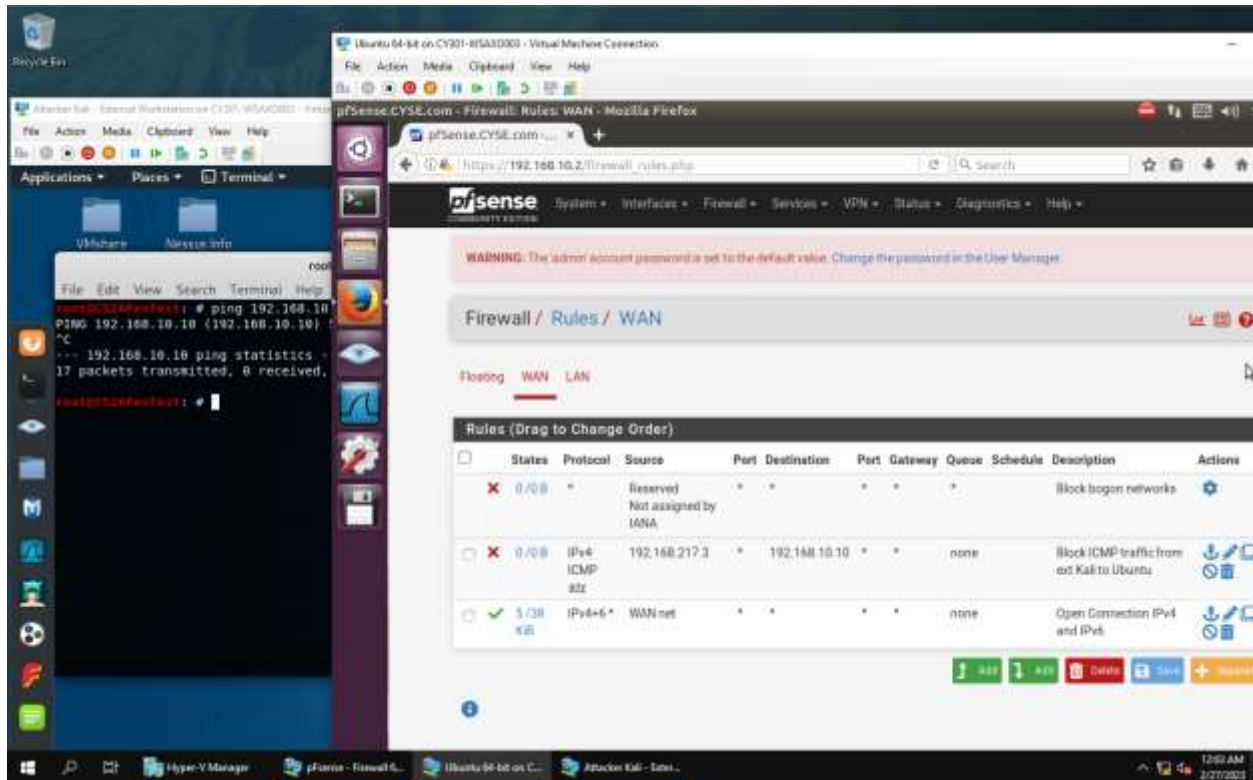
```
root@CS2APentest:~# ping 192.168.10.10
PING 192.168.10.10 (192.168.10.10) 56(84) bytes of data:
64 bytes from 192.168.10.10: icmp_seq=1 ttl=63 time=5.31 ms
64 bytes from 192.168.10.10: icmp_seq=2 ttl=63 time=5.30 ms
64 bytes from 192.168.10.10: icmp_seq=3 ttl=63 time=3.98 ms
64 bytes from 192.168.10.10: icmp_seq=4 ttl=63 time=19.5 ms
^C
--- 192.168.10.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 9ms
rtt min/avg/max/ndev = 3.976/8.516/19.482/6.354 ms
root@CS2APentest:~# ping 192.168.10.11
PING 192.168.10.11 (192.168.10.11) 56(84) bytes of data:
64 bytes from 192.168.10.11: icmp_seq=1 ttl=127 time=5.63 ms
64 bytes from 192.168.10.11: icmp_seq=2 ttl=127 time=21.6 ms
64 bytes from 192.168.10.11: icmp_seq=3 ttl=127 time=5.26 ms
^C
--- 192.168.10.11 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 10ms
rtt min/avg/max/ndev = 5.260/10.823/21.579/7.607 ms
root@CS2APentest:~# ftp 192.168.10.11
Connected to 192.168.10.11.
220 Microsoft FTP Service

Name (192.168.10.11:root):
```

I Pinged Win 2008 and Ubuntu to show that ICMP traffic from ext Kali worked on all systems on the LAN side. I also showed that FTP from ext Kali to Win 2008 also works before the Firewall rules.

1. Block ICMP traffic from ext Kali to Ubuntu

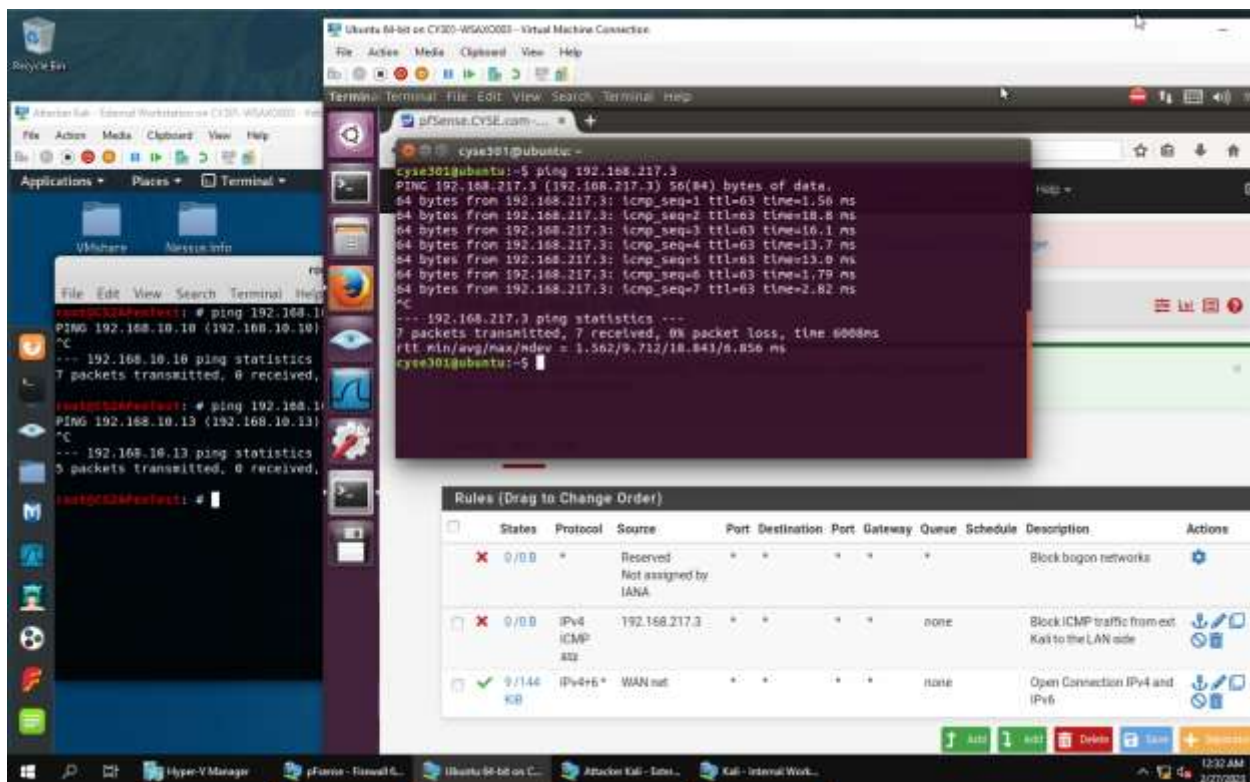
Rule #	Interface	Action	Source IP	Dest IP	Protocol/port
2	WAN	Block	192.168.217.3	192.168.10.10	ICMP



I blocked the ICMP (ping) traffic from ext Kali to Ubuntu. On the left you can see the failed ping from ext Kali to Ubuntu. I tested pinging ext Kali from Ubuntu and it worked. I tested pinging from ext kali to int kali and it worked.

2. Block ICMP traffic from ext kali to LAN side

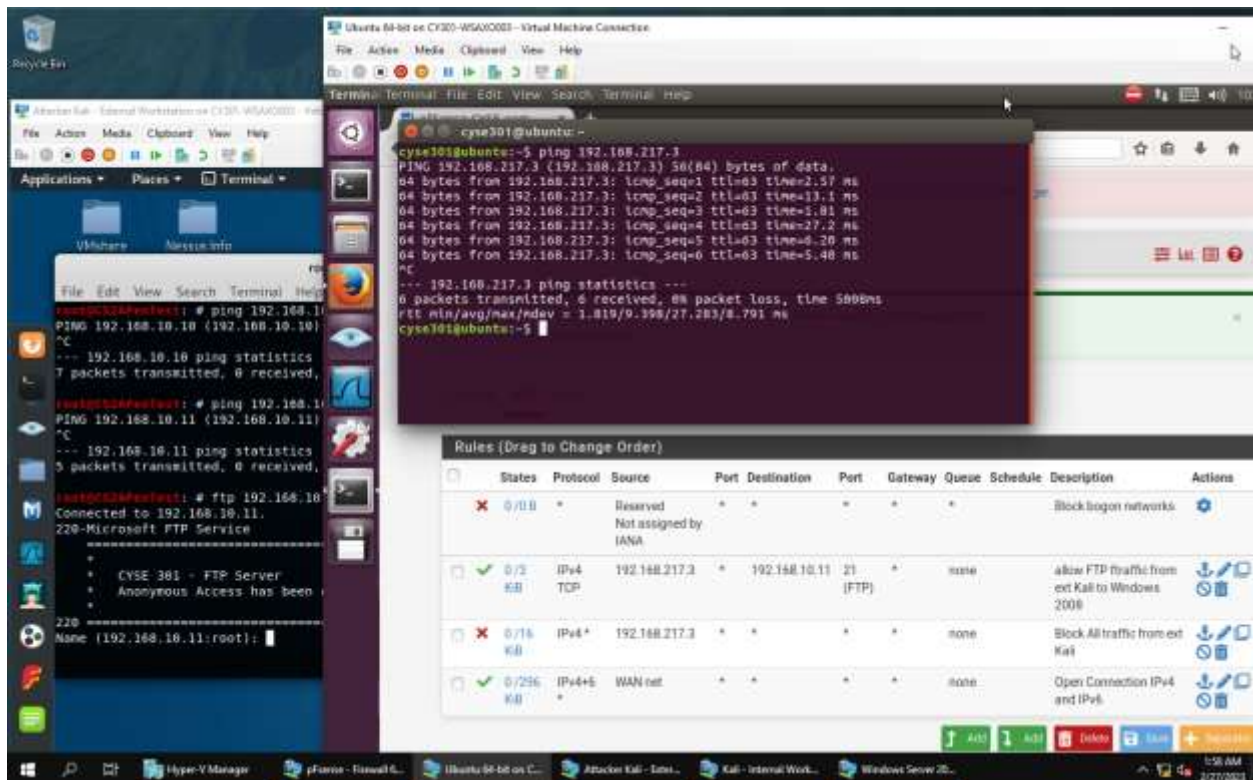
Rule #	Interface	Action	Source IP	Dest IP	Protocol/port
2	WAN	Block	192.168.217.3	any	ICMP



I set the destination to any so all IP addresses will be blocked from getting a ping from ext Kali. I tried to ping Ubuntu and int Kali from ext Kali as in the picture. The ping failed. On Ubuntu I tried to ping ext Kali and was able to (also in the picture).

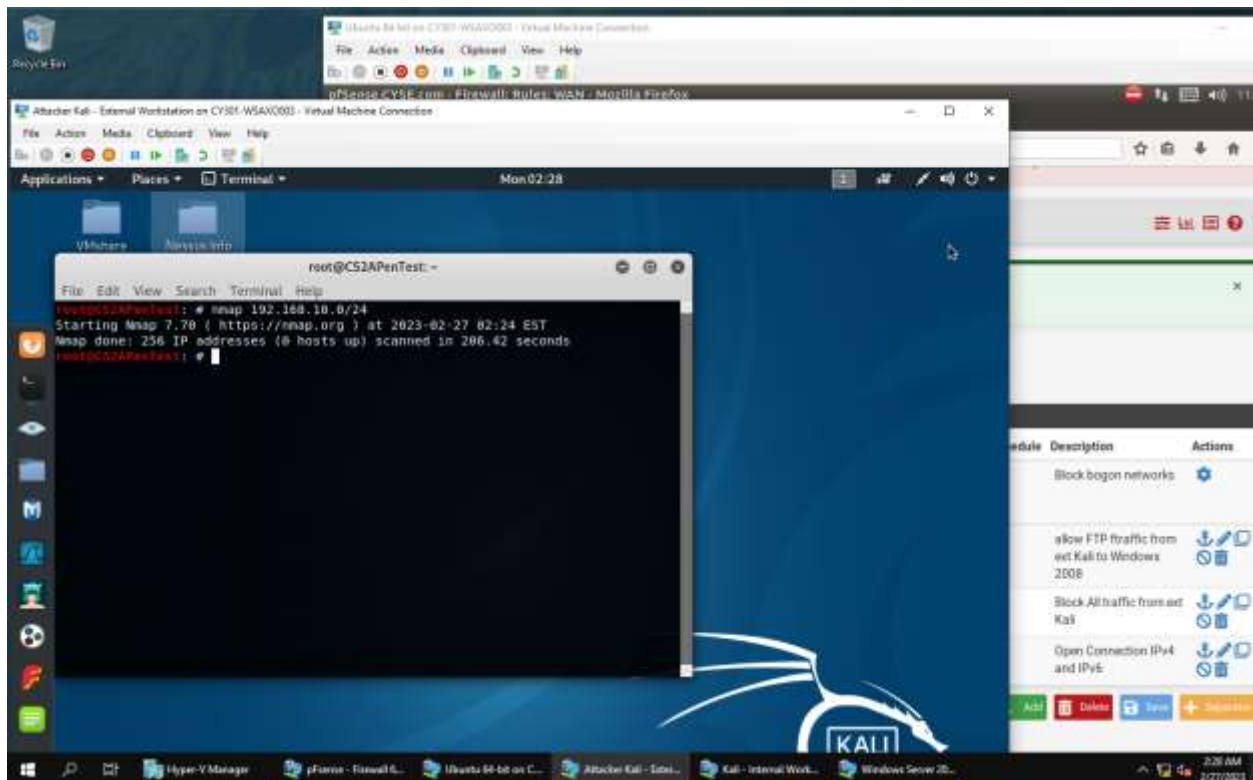
3. Block all traffic from ext Kali other than FTP to Windows 2008

Rule #	Interface	Action	Source IP	Dest IP	Protocol/port
2	WAN	Pass	192.168.217.3	192.168.10.11	TCP port 21 (FTP)
3	WAN	Block	192.168.217.3	any	any



The first thing I did was block all traffic from ext Kali to the LAN side. Then I made a rule that allowed FTP to Windows 2008 from ext Kali. I made sure the rule was on above the Block rule to make sure it worked. Then I tried to have the block on top to see if the FTP would fail and it did. On the left I tried to ping Ubuntu and Windows 2008 and both failed so the rule is working. I also tried to ping ext Kali from Ubuntu (on the right of the picture) and it worked.

4. Do A1 with the B3 rules



No hosts were found. It looks like nmap was blocked completely. The reason port 21 (FTP) wasn't found was because only FTP could go through. Base nmap without any modifications like nmap -sV uses ICMP (ping) to look for hosts then do other kinds of scans. Nmap couldn't see Windows 2008 because ICMP was blocked so it wasn't able to check the open ports.