OLD DOMINION UNIVERSITY

CYSE 301 CYBERSECURITY TECHNIQUES AND OPERATIONS

# Assignment #5 Password Cracking

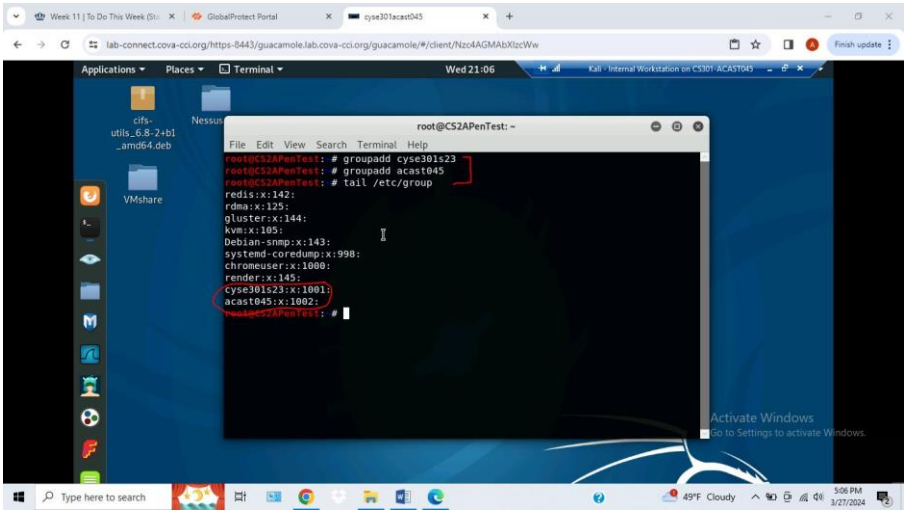
Angelica Grace Castro

01267060

# PART A

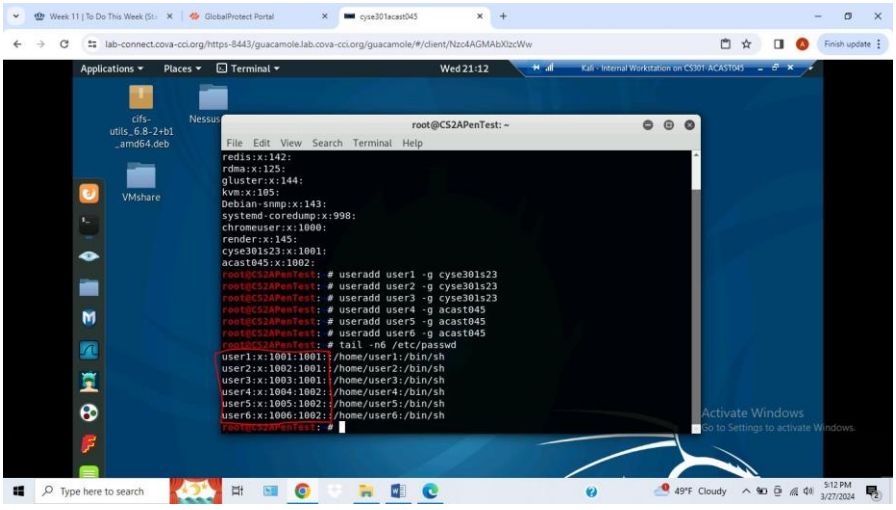
# TASK A

1. Create two groups, one is cyse301s23, and the other is your ODU Midas ID (for example, pjiang). Then display the corresponding group IDs.



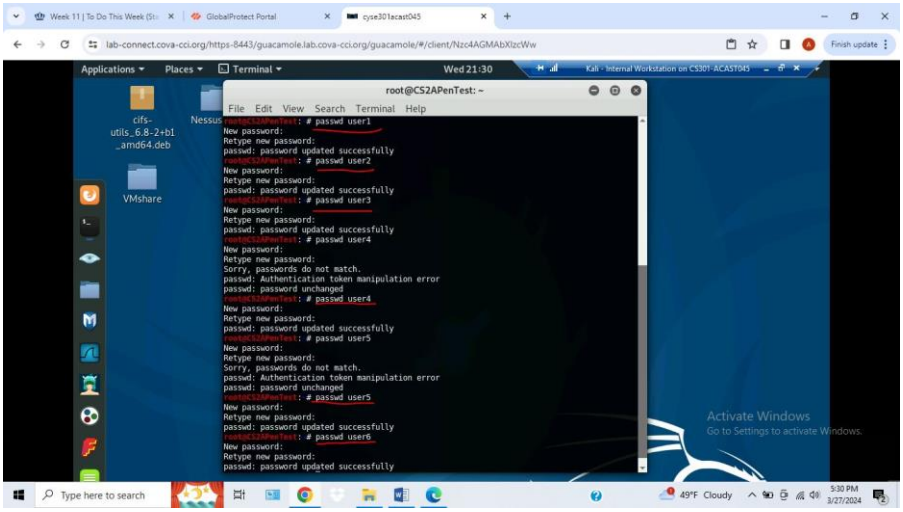

In the screenshot above, I created two groups by entering groupadd cyse301s23 and groupadd acast045 to the command line. I then entered tail /etc/group to display the groups and group ID. The group ID for cyse301s23 is 1001 and the group ID for acast045 is 1002.

2. Create and assign three users to each group. Display related UID and GID information of each user.



In the screenshot above, I created three users for each group by entering the command useradd username –g groupname. Each username is just user plus their respective number and the groupnames are those I created previously. I then entered tail –n6 /etc/passwd to the command line to display the usernames, their UID, and GID. User1 has a UID of 1001 and GID of 1001, User2 has a UID of 1001 and GID of 1001, User3 has a UID of 1003 and GID of 1001, User4 has a UID of 1004 and GID of 1002, User5 has a UID of 1005 and GID of 1002, and User6 has a UID of 1006 and GID of 1002.

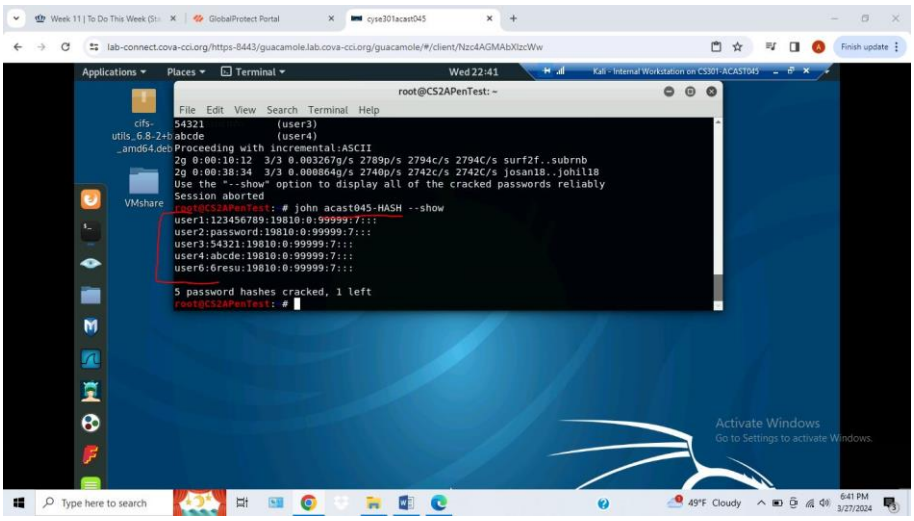
3. Choose six new passwords, from easy to hard, and assign them to the users you created. You need to show me the password you selected in your report, and DO NOT use your real-world passwords.




In the screenshot above, I entered passwd user1 into the command line to assign a password to user1. I repeated that step for the other five users to assign them passwords. The passwords I assigned are as follows:

- user1 – 123456789
- user2 – password
- user3 – 54321
- user4 – abcde
- user5 – five5
- user6 – 6resu

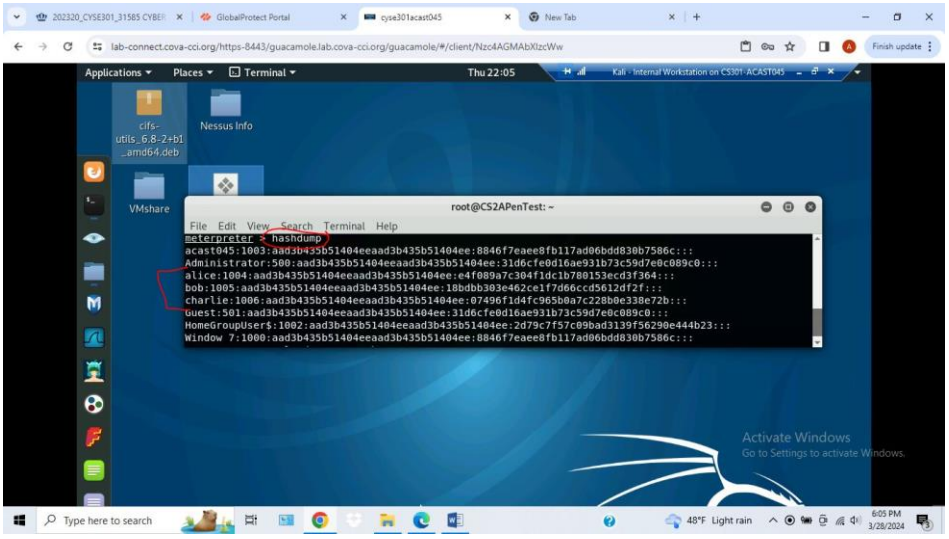
4. Export all six users' password hashes into a file named "YourMIDAS-HASH" (for example, pjiang-HASH). Then launch a dictionary attack to crack the passwords. You MUST crack at least one password in order to complete this assignment.




In the screenshot above, I exported the password hashes into a file by entering tail –n6 /etc/shadow > acast045-HASH. I then ran john the ripper by entering john acast045-HASH into the command line. I then entered john acast045-HASH --show in order to display the passwords that were cracked. Five out of six passwords were cracked with only user5's password of five5 not being cracked.

# TASK B

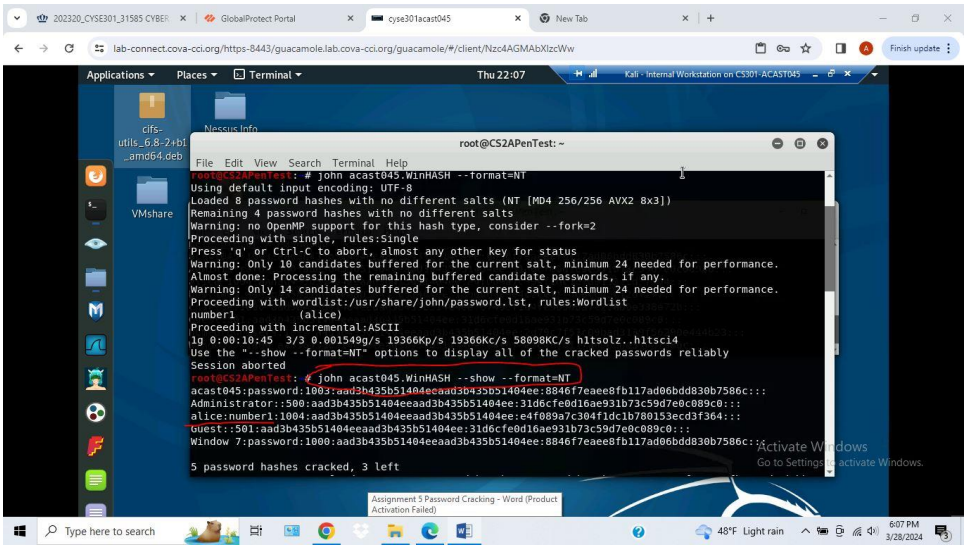
1. Display the password hashes by using the "hashdump" command in the meterpreter shell.




In the screenshot above, after I created the three users I used the payload made from the previous module to set up a reverse tcp connection. I then used the bypassuac exploit to escalate my

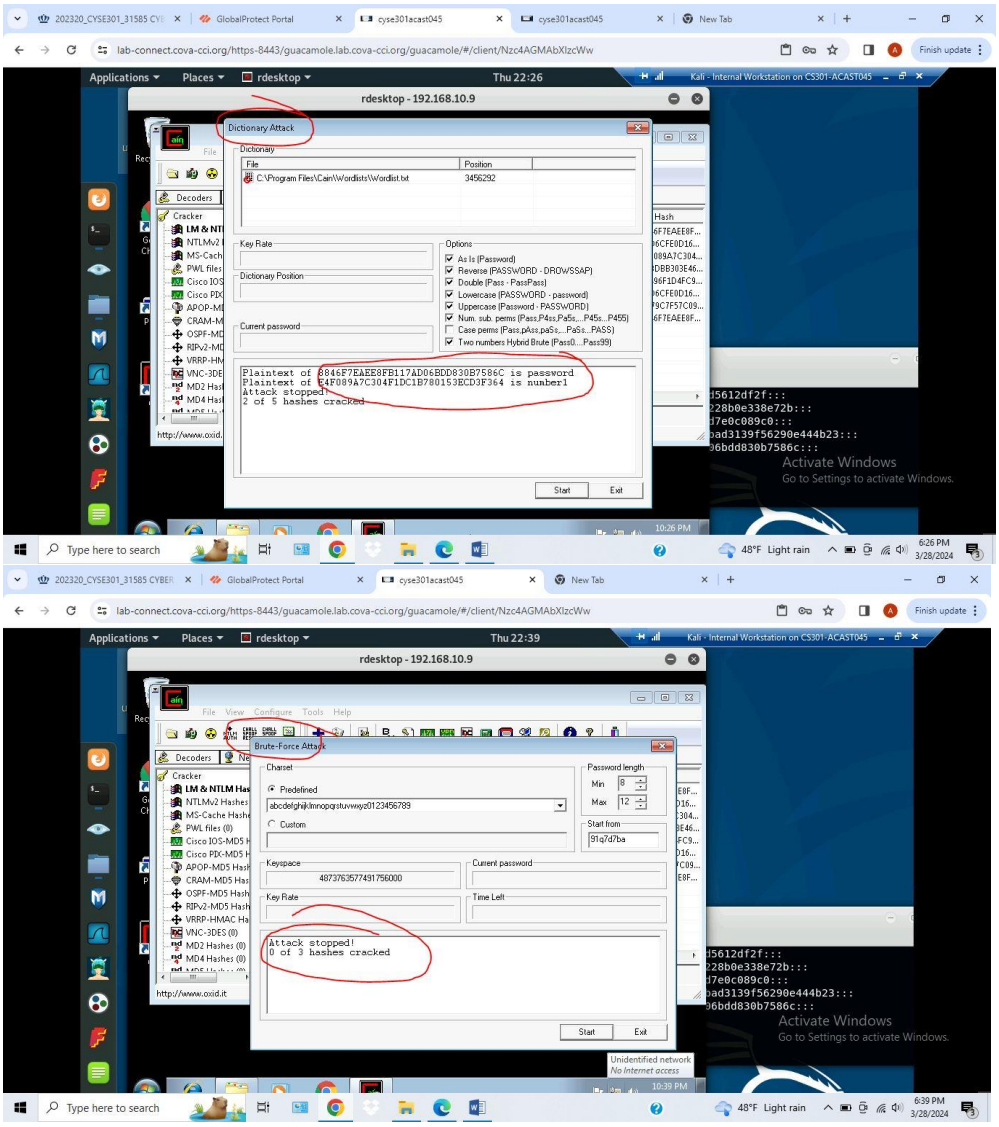
privileges. I used the command hashdump to display the password hashes. The image above shows the three users I created which are alice, bob, and charlie.

2. Save the password hashes into a file named “your_midas.WinHASH” in Kali Linux (you need to replace the “your_midas” with your university MIDAS ID). Then run John the ripper for 10 minutes to crack the passwords (You MUST crack at least one password in order to complete this assignment.).




In the screenshot above, I used the method showed in the lab manual to create a file. I entered gedit acast045.WinHASH to create my file and then copied and pasted the information displayed by the hashdump command. I then entered john acast045.WinHASH but it did not work because I had to enter john acast045.WinHASH --format=NT. After entering the right command and letting it run for 10 minutes, I stopped the process by holding ctrl + c. I then entered john acast045.WinHASH --show --format=NT to display the passwords cracked. Out of the three users I created only alice’s password which was number1 was cracked.

3. Upload the password cracking tool, Cain and Abel, to the remote Windows 7 VM, and install it via a remote desktop window. Then, implement BOTH brute force and dictionary attacks to crack the passwords. (You MUST crack at least one password in order to complete this assignment.).



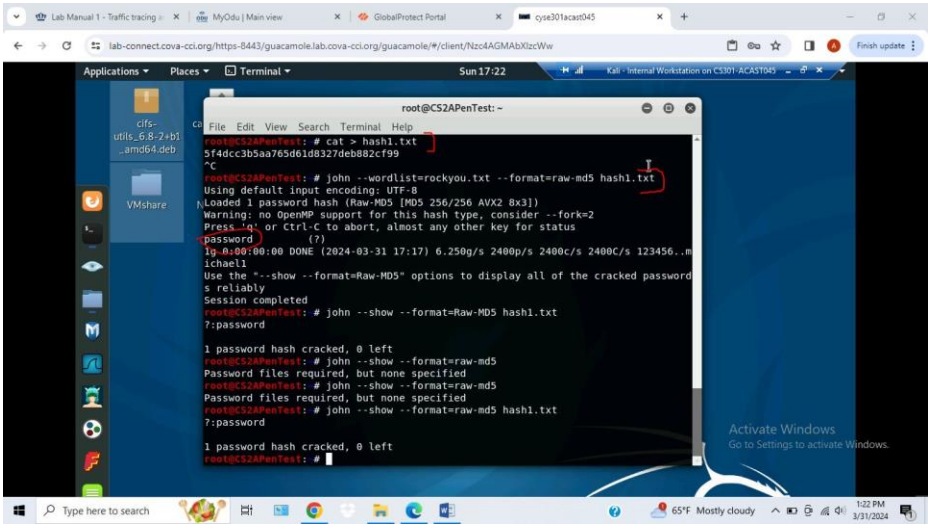

In the screenshots above, I copied the ca_setup.exe file to the desktop and uploaded it to the target VM by entering upload /root/Desktop/ca_setup.exe C:// on the meterpreter shell. I then opened a new terminal window and entered rdesktop 192.168.10.9 to open a remote desktop window. Once the remote desktop window is open I logged into the acast045 account that was created in the previous module and has administrator privileges. After installing Cain and Abel, I first implemented a dictionary attack and then a brute force attack. The dictionary attack was able to identify two passwords one belonging to the alice account which is number1 and the other to the acast045 account which is password. In the brute force attack, I set the password length to 8-12 which matches the password lengths of the passwords I created for the users. The total time to complete the attack fully was around 26000 years so after about 10 minutes I stopped the attack. The brute force attack was unable to crack any of the other passwords.

# TASK C

Search the proper format in John the Ripper to crack the following MD5 hashes (use the --list=formats

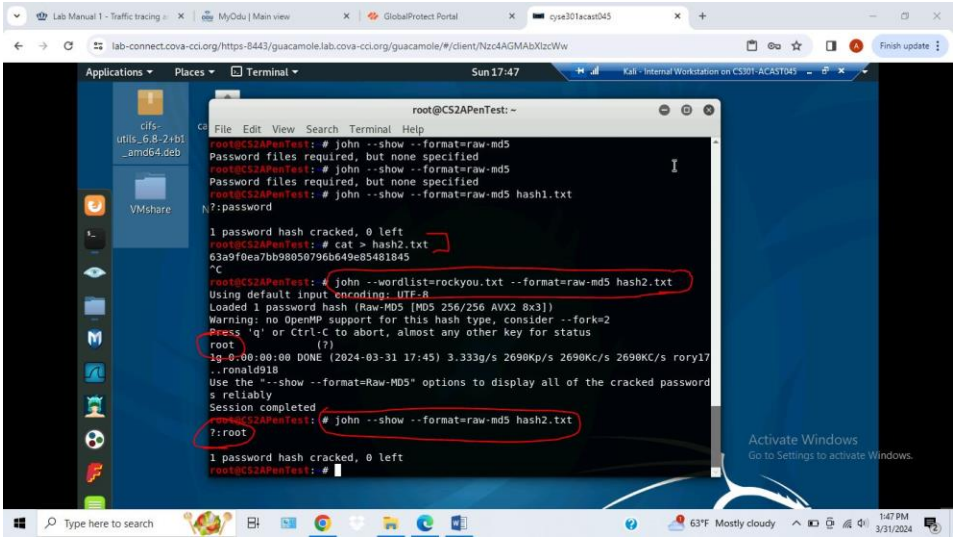
option to list all supported formats) . Show your steps and results.

1. 5f4dcc3b5aa765d61d8327deb882cf99




In the screenshot above, I entered --list=formats to look through the formats and found raw-md5 listed which is the format of the hash. I then entered cat > hash1.txt to create a file and typed the hash given into the file. I held ctrl + c to stop writing into the file. I then entered john --wordlist=rockyou.txt --format=raw-md5 hash1.txt to crack the hash which showed that the plaintext is password. I previously copied the rockyou.txt file to my home directory so I did not have to change directories or copy the file again. I also entered john --show --format=raw-md5 hash1.txt to display the plaintext of the cracked hash.

2. 63a9f0ea7bb98050796b649e85481845

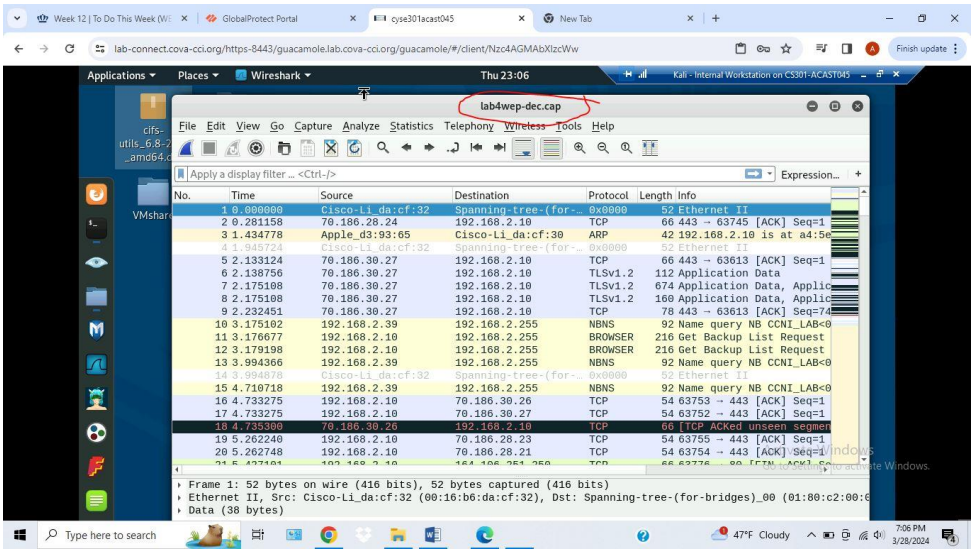


In the screenshot above, I used the same methods as the previous hash. I entered cat > hash2.txt to create a new hash file, entered the given hash, and then exited the process by holding ctrl + c. I then entered john --wordlist=rockyou.txt --format=raw-md5 hash2.txt to crack the hash. I entered john --show --format=raw-md5 hash2.txt to display the cracked hash which showed that the plaintext is root.

# PART B

# TASK C

1. Decrypt the lab4wep. cap file and perform a detailed traffic analysis.

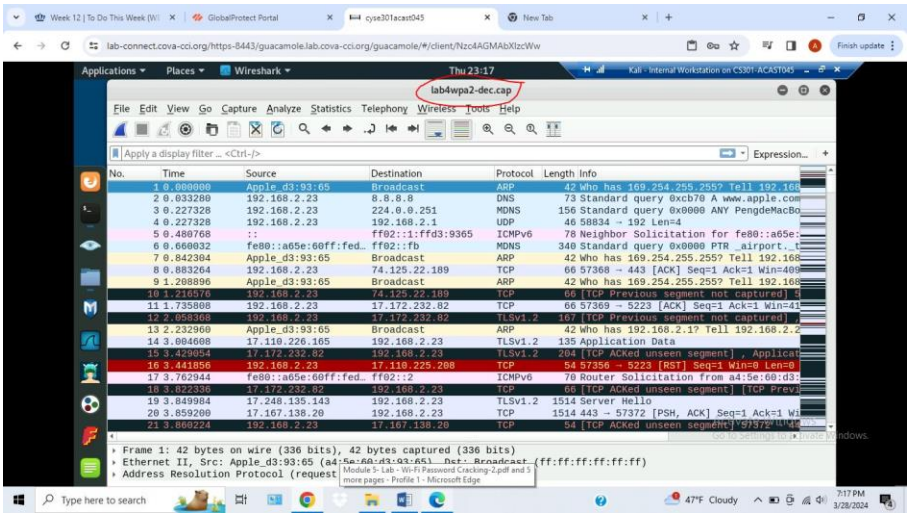



The screenshot above shows the decrypted traffic file. To decrypt lab4wep.cap file I changed directory to CYSE301 and then to Module V-Wireless Security. I then entered aircrack-ng lab4wep.cap to the command line and chose target network 1 which is ccni-test and was the only network that had only wep. I then entered airdecap-ng –w F2:C7:BB:35:B9 lab4wep.cap which created the decrypted file lab4wep-dec.cap that shows the decrypted traffic.

The decrypted file shows that 86.2 percent of the packets comes from address resolution protocol which suggests someone is broadcasting requests to find a certain IP address. When I entered arp to the display filter box to show only the ARP packets it shows that the majority of the packets displayed comes from a device that has the MAC address of Alfa_82:c3:7e with the destination broadcast and the request is finding the owner of the IP address 192.168.2.10. Occasionally, the device is looking for the device with the IP address of 169.254.255.255. When I opened a packet's details it shows that the sender has an IP address of 0.0.0.0 but a MAC address of a4:5e:60:d3:93:65 which is the same MAC address that is listed for the sender Apple_d3:93:65. 13.6 percent of the packets captured have the TCP protocol. Other protocols seen are NBNS, BROWSER, and TLSv1.2. There were a few TCP packets that was labeled black and has a

warning regarding a previous segment not being captured or the packet being suspected as a retransmission.

2. Decrypt the lab4wpa2. cap file and perform a detailed traffic analysis.

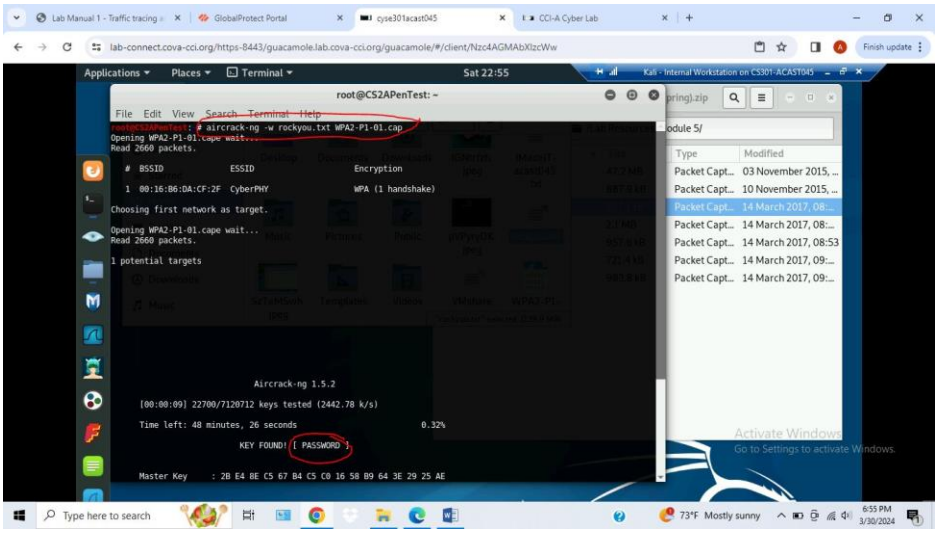



The screenshot above shows the decrypted traffic file lab4wpa2-dec.cap. Following the lab manual, I copied and unzipped the rockyou.txt file into the Module V-Wireless Security folder by entering cp /usr/share/wordlists/rockyou.txt.gz . and gunzip rockyou.txt.gz. To decrypt lab4wpa2.cap I entered aircrack-ng lab4wpa2.cap –w rockyou.txt then chose target network 4. I then entered airdecap-ng –e CCNI –p password lab4wpa2.cap to get the decrypted file lab4wpa2-dec.cap.

The decrypted file showed that 99.7 percent of the packets were ipv4 and 98.2 percent were using TCP with 5.7 percent being ssl and 2.8 percent being http packets. Similar to the lab4wep-dec.cap file, the lab4wpa2-dec.cap file has some TCP packets that has a black warning label that states a segment was not captured. There are also TCP packets that have a red warning that states the connection was reset. There are TLSv1.2 packets that are part of the ssl handshake protocol. Other packets captured include GQUIC, DNS, MDNS, HTTP, UDP, ARP, and ICMPv6. 91.2 percent of the generated traffic comes from a device with the IP address of 192.168.2.23.

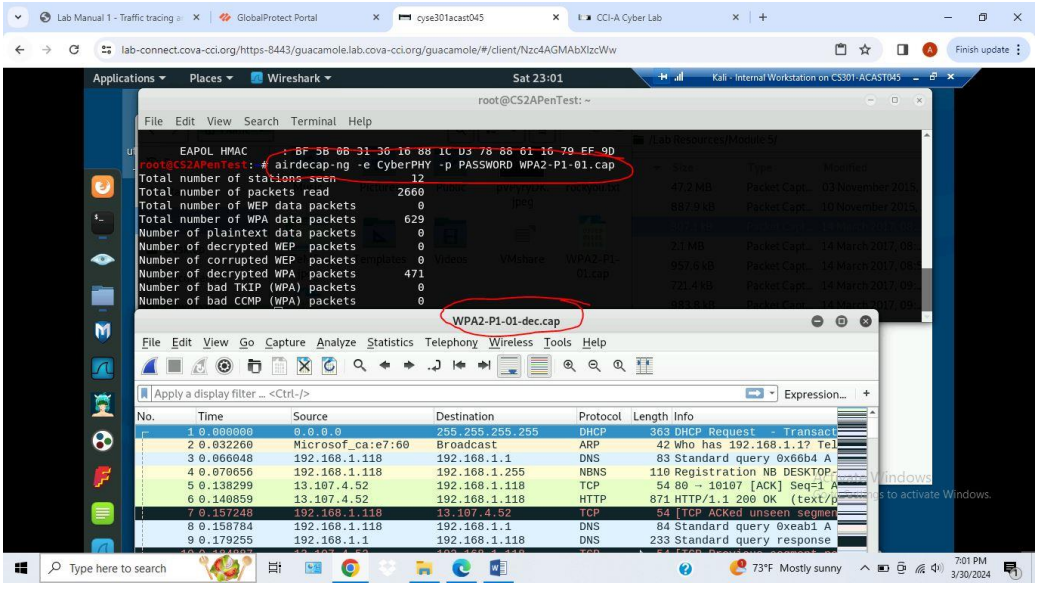
# TASK D

1. Implement a dictionary attack and decrypt the traffic.




Following the assignment instructions, I entered echo –n acast045 | md5sum in the terminal which gave me a hash ending in 0 indicating I should be decrypting WPA2-P1-01.cap. The file was already in the Internal Kali VM so I copied and pasted the file to the home folder to locate it easier and copied the rockyou.txt to the home folder as well. I changed my directory back to the home folder by entering cd ~ and then entered aircrack-ng –w rockyou.txt WPA2-P1-01.cap to implement a dictionary attack on the traffic file which gave me the key PASSWORD shown in the screenshot above.

2. Decrypt the encrypted traffic and write a detailed summary to describe what you have explored from this encrypted traffic file.




In the screenshot above, I entered airdecap-ng –e CyberPHY –p PASSWORD WPA2-P1-01.cap which decrypted the traffic file and created a decrypted traffic file in my home folder with the name WPA2-P1-01-dec.cap. I clicked on the statistics tab and selected protocol hierarchy. The

majority of the packets captured uses the IPv4 protocol with a percentage of 90.9. The majority of those packets use TCP with a percentage of 64.3 while UDP makes up 18.9 percent of the packets. Only 1.3 percent of the packets use ARP. Similar to the other two traffic files previously analyzed, there are TCP as well as TLSv1.2 packets that have a black warning label stating that the acknowledgement segment was not captured. There is also a TCP packet that has a red warning stating connection was reset. The captured traffic contains protocols not previously seen such as LLMNR, NBNS, SSDP, DHCP, and IGMPv3. I noticed many packets having the source IP address of 192.168.1.118 so I entered ip.src==192.168.1.118 into the display filter box to see how much of the traffic is coming from that device. I found that 216 out of 471 packets which is 45.9 percent were from the device with the IP address 192.168.1.118. The traffic generated by the device varied but included TCP handshakes, TLSv1.2 handshake protocols, ping requests, multiple name queries, standard queries, and client key exchanges. There is also an encrypted payload coming from 192.168.1.118 to 172.217.4.132 using the GQUIC protocol.