

OLD DOMINION UNIVERSITY

CYSE 301 CYBERSECURITY TECHNIQUES AND OPERATIONS

Assignment #5: Password Cracking (Part A) & Wi-Fi
Password Cracking (Part B)

Antonio Shields

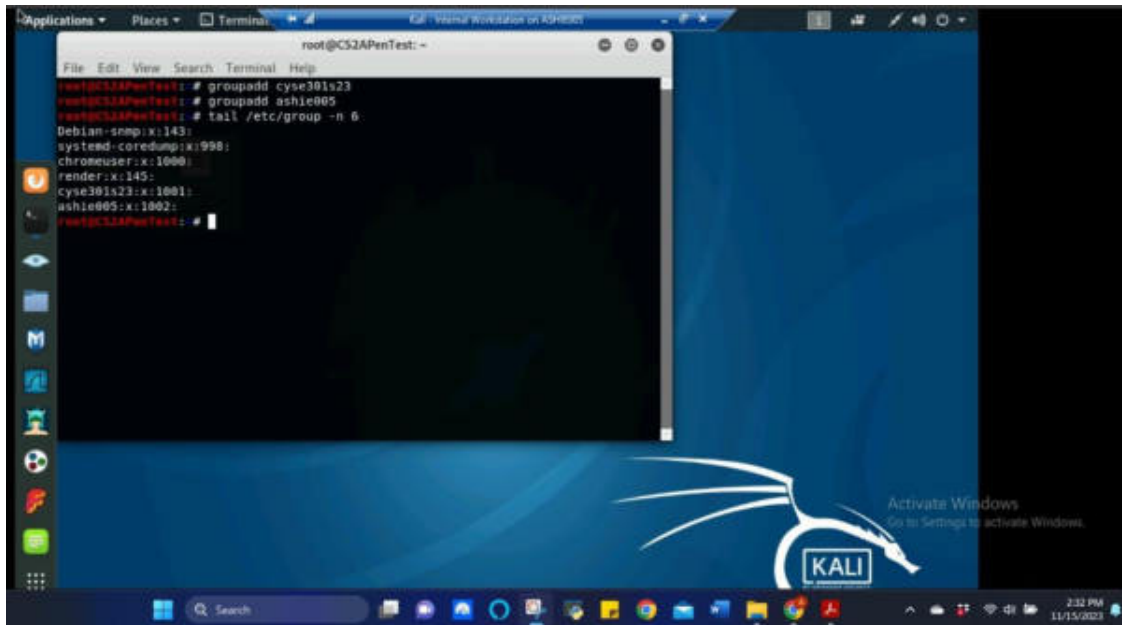
01240495

At the end of this module, each student must submit a report indicating the completion of the following tasks. Make sure you take screenshots as proof.

You need to use

TASK A: LINUX PASSWORD CRACKING (25 POINTS)

1. **5 points.** Create two groups, one is cyse301s23, and the other is your ODU Midas ID (for example, pjiang). Then display the corresponding group IDs.



```
root@CS2APenTest: ~  
root@CS2APenTest: # groupadd cyse301s23  
root@CS2APenTest: # groupadd ashie005  
root@CS2APenTest: # tail /etc/group -n 6  
Debian-scm:x:143:  
systemd-coredump:x:998:  
chromuser:x:1000:  
render:x:145:  
cyse301s23:x:1001:  
ashie005:x:1002:  
root@CS2APenTest: #
```

Figure 1 Screenshot of two groups being created and displaying of corresponding group IDs for Task A.1

The above screenshot shows the two groups cyse301s23 and ashie005 being made and the respective group IDs 1001 and 1002 that were assigned are displayed.

2. **5 points.** Create and assign three users to each group. Display related UID and GID information of each user.

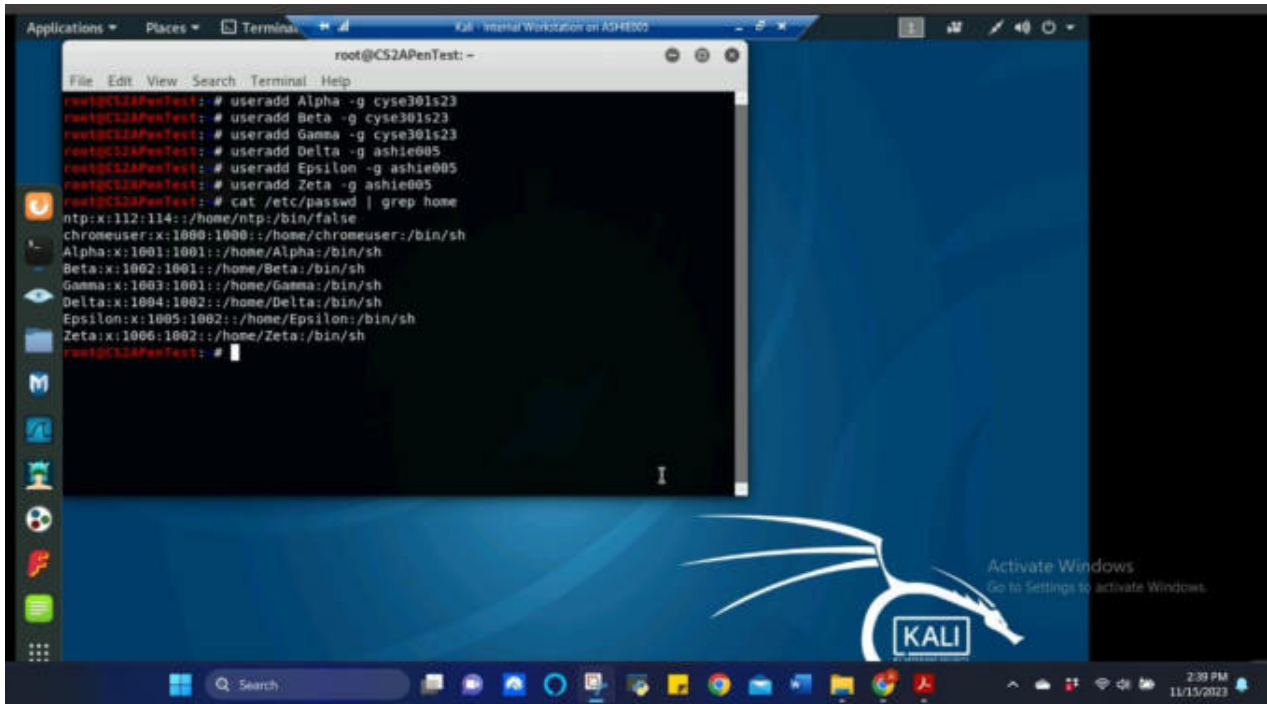


Figure 2 Screenshot of six users being created and assigned to one of the previously created groups for Task A.2

The above screenshot shows the six users, Alpha, Beta, Gamma, Delta, Epsilon, Zeta being created and the first three users were assigned to the cyse301s23 group and last three users were assigned to the ashie005 group.

3. **5 points.** 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.

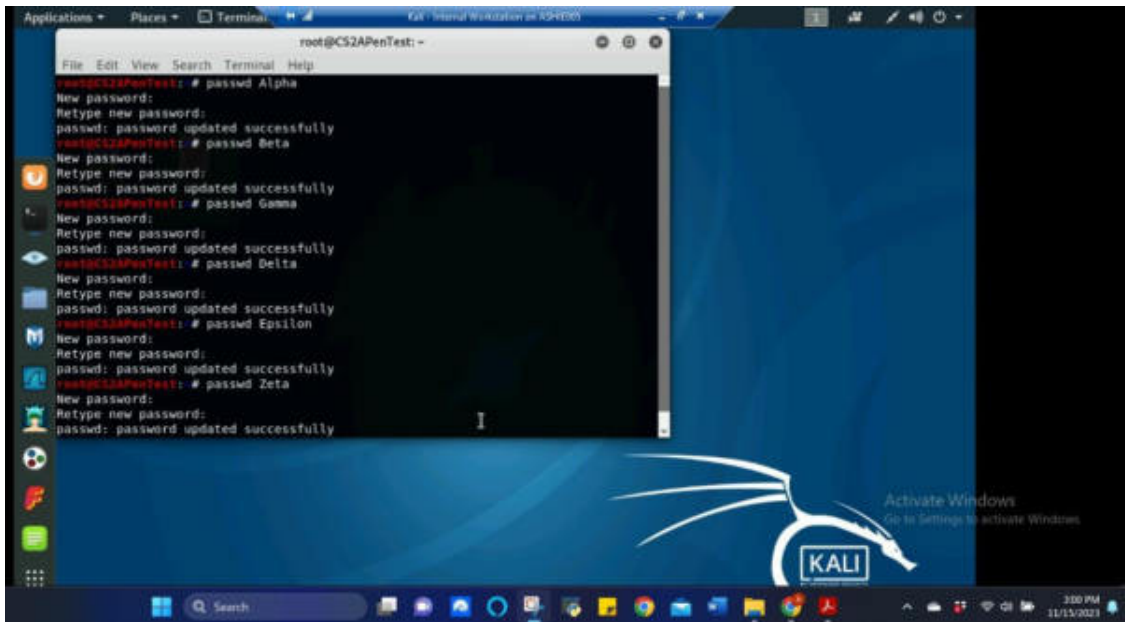


Figure 3 Screenshot of the six users created being assigned separate passwords for Task A.3

The above screenshot shows the six users, Alpha, Beta, Gamma, Delta, Epsilon, Zeta being assigned separate passwords. The passwords for each user are: Alpha-happy5678, Beta-sad1234, Gamma-GLad2468!, Delta-M@d789o?, Epsilon-T1h2r3e45tr!k3s, and Zeta-Gr8@bc!z32?V

4. **5 points.** 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.

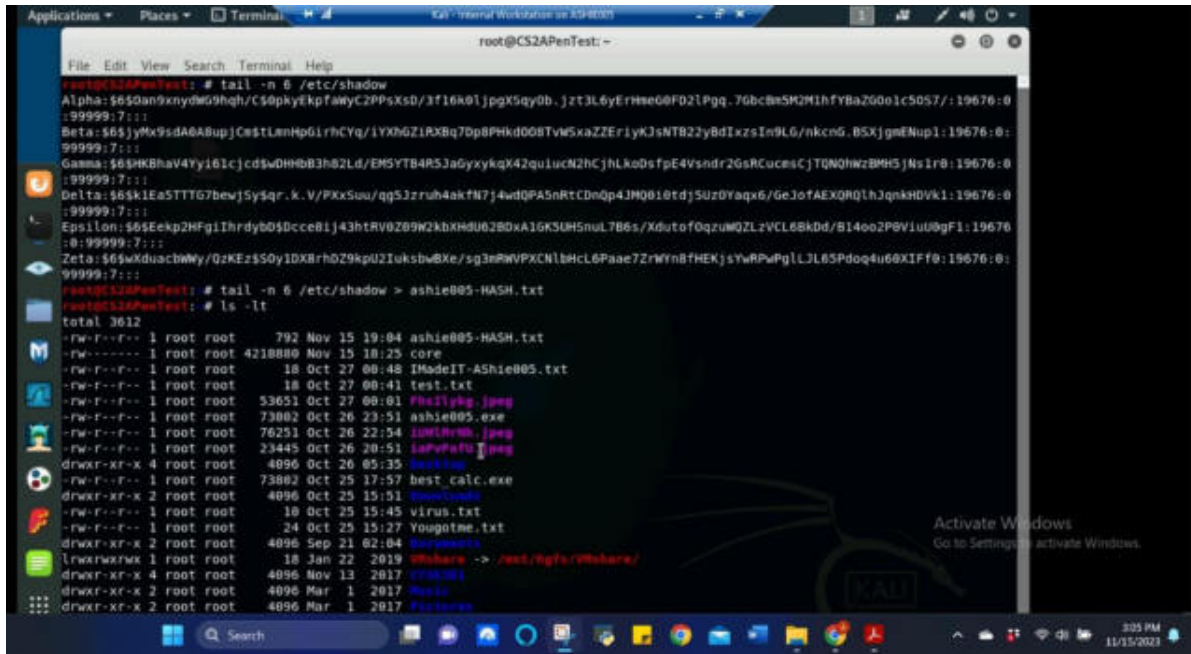


Figure 4 Screenshot of the six user's hashes being retrieved and exported to ashie005-HASH.txt for Task A.4

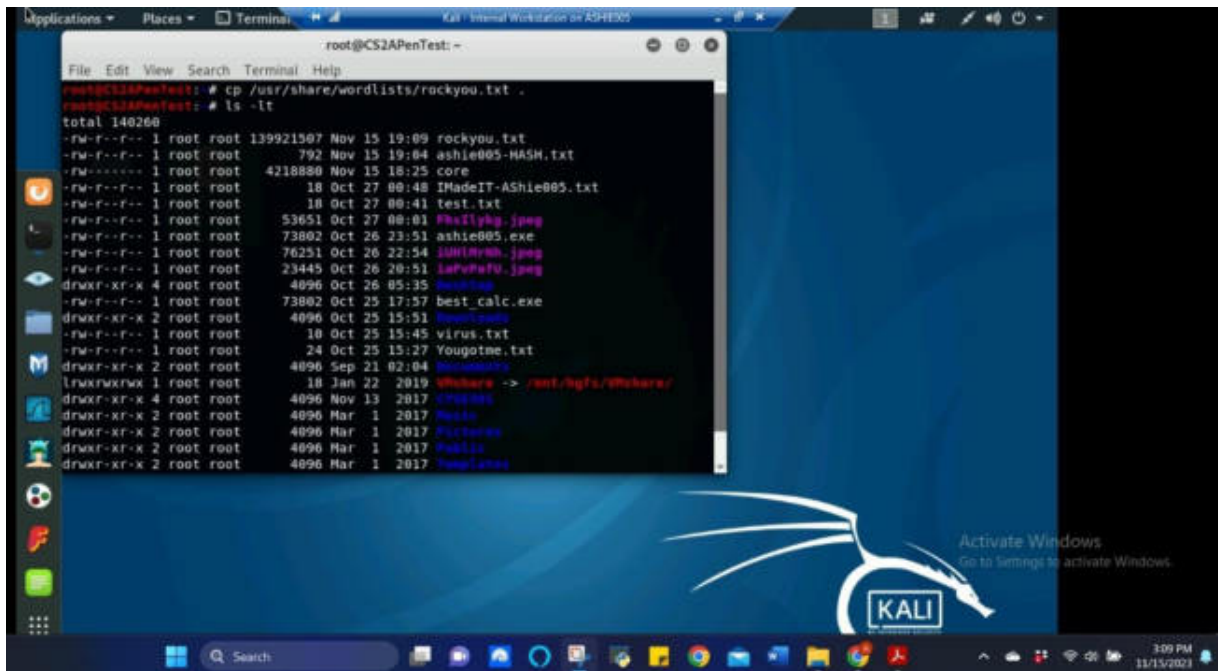


Figure 5 Screenshot of the rockyou.txt file being copied into directory prior to the dictionary attack for Task A.4

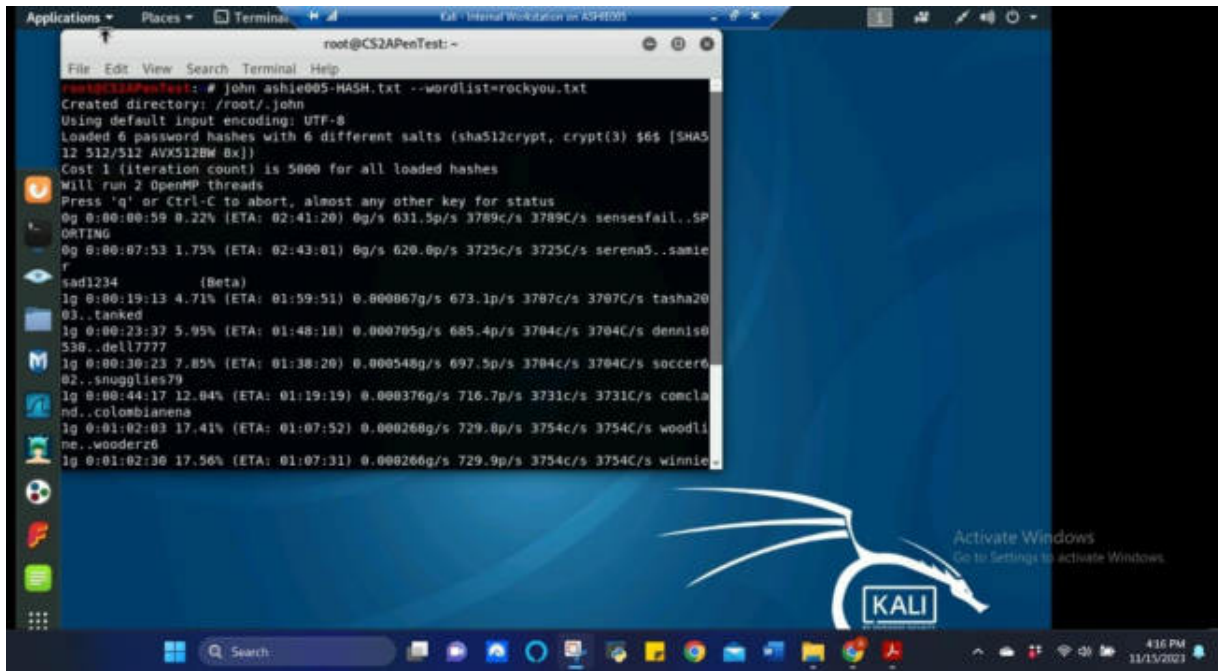


Figure 6 Screenshot of the dictionary attack being ran for an hour and one password being cracked for Task A.4

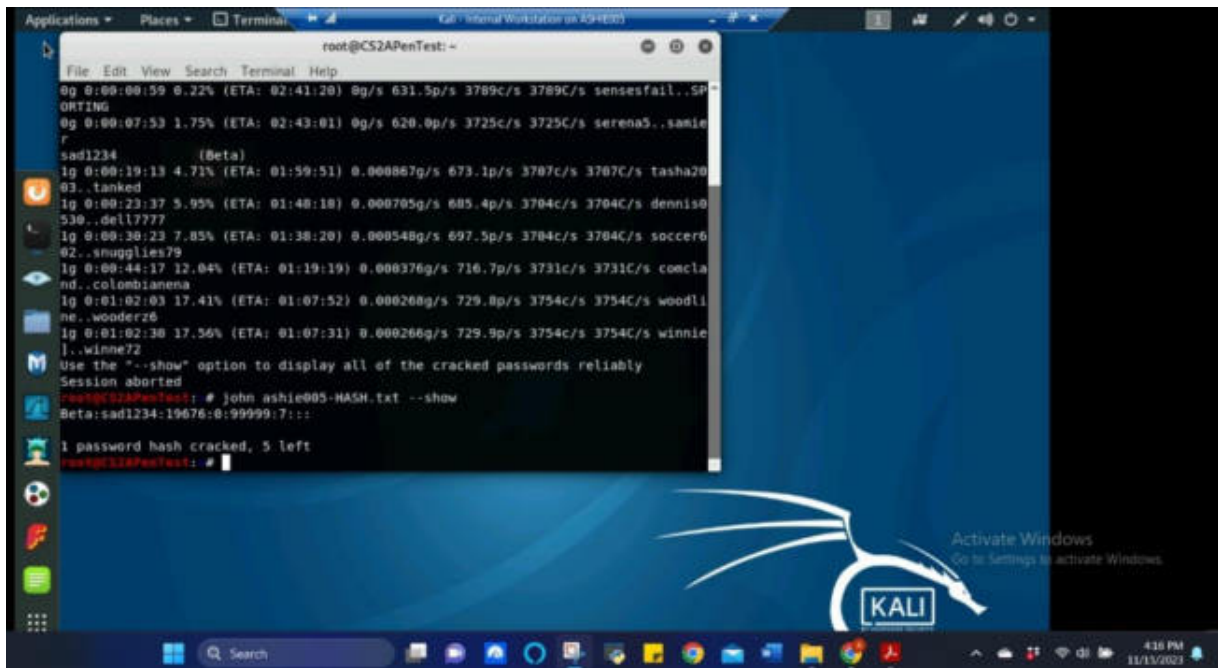


Figure 7 Screenshot of the one password cracked being shown for Task A.4

The above screenshots show the six user's hashes being retrieved and being exported to the ashie005-HASH.txt file as instructed. After exporting, the rockyou.txt wordlist file was cp to the current directory to use for the dictionary attack to see how many passwords could be cracked using John the ripper. Since the rockyou file was unzipped for a prior example, it did not require being unzipped again, so just needed to be copied to the current directory to be used. After running for an hour, only one password from the six users created was cracked, which was Beta

TASK B: WINDOWS PASSWORD CRACKING (25 POINTS)

Log on to Windows 7 VM and create a list of 3 users with different passwords. Then you need to establish a reverse shell connection with the admin privilege to the target Windows 7 VM.



Figure 8 Screenshot of the three users being created for Windows 7 for Task B

The above screenshot shows the three users that were created for Windows 7, AlphaWin7, BetaWin7, and GammaWin7 along with the other users that were previously created from past assignments, HackU and ASHie005. The passwords for the three created accounts are AlphaWin7-abcde12345, BetaWin7-GoTeamGo123!, and GammaWin7-AllSm!les757. The password for ASHie005 is abcd1234 and the password for HackU is password123, these accounts were created in previous assignments.

Now, complete the following tasks:

1. **5 points.** Display the password hashes by using the “hashdump” command in the meterpreter shell.

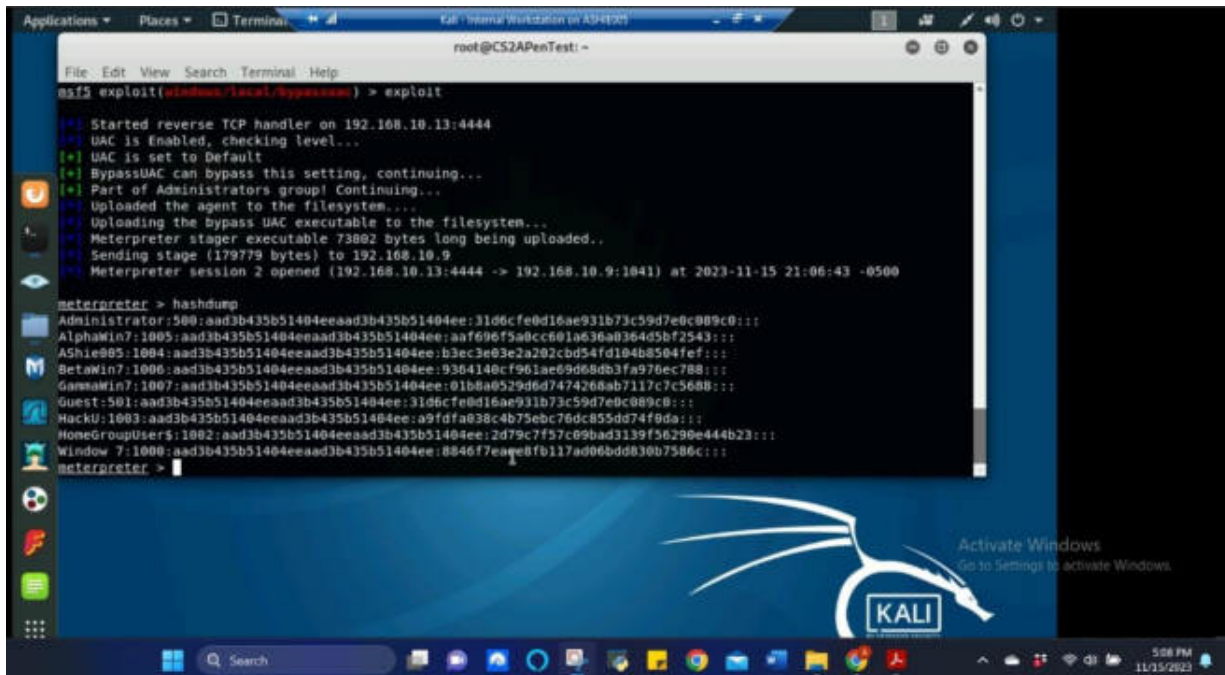


Figure 9 Screenshot of the reverse shell connection being established through metepreter and hashdump command being executed for Task B.1

The above screenshot shows the hashdump command being executed in the meterpreter shell after establishing a reverse shell connection to the Windows 7 VM from Internal Kali. The hashdump revealed all the hashes for every user account on the Windows 7 VM including Administrator, Guest, and HomeGroupUser\$ accounts that were not listed on the previous login screen.

2. **10 points.** 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).

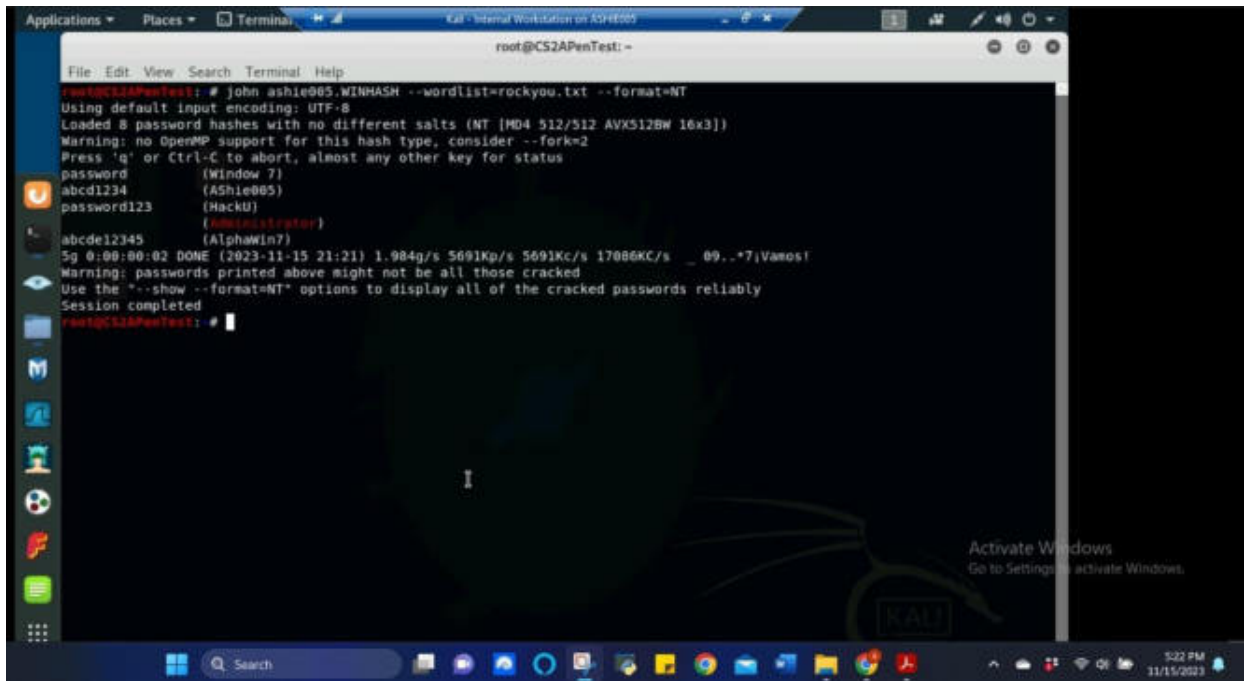


Figure 10 Screenshot of John the ripper being executed for Task B.2

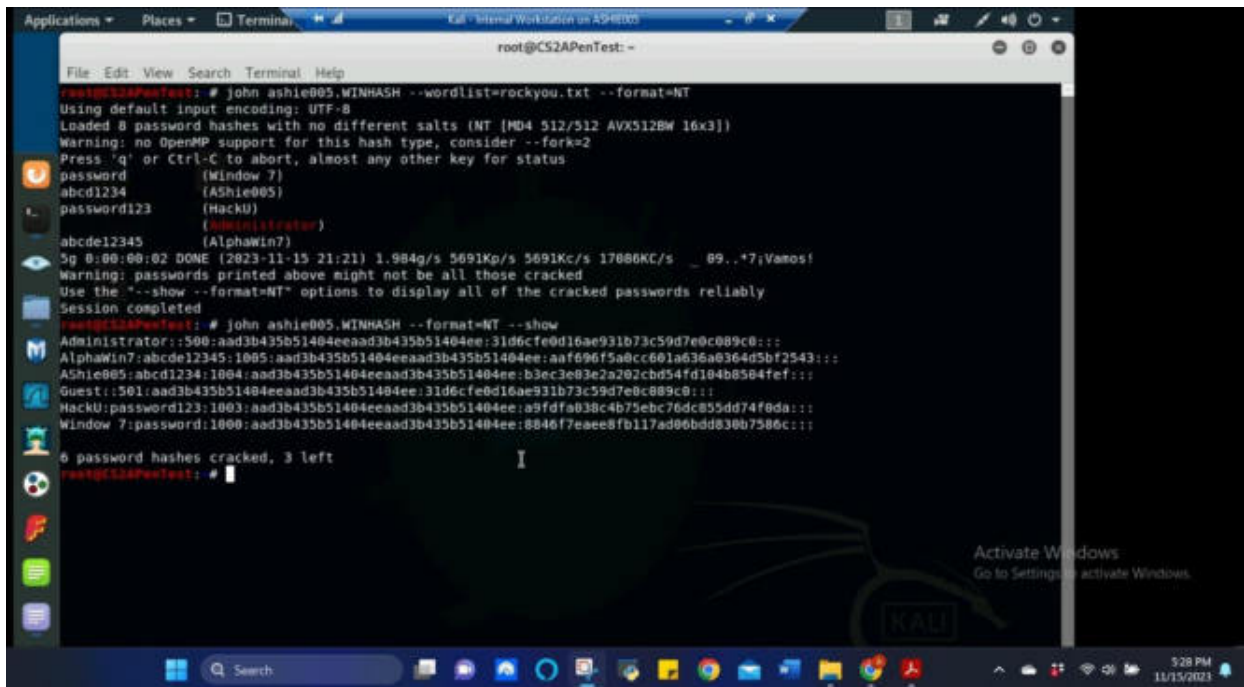


Figure 11 Screenshot of John the ripper results for Task B.2

The above screenshots show the WinHashes that retrieved in the previous step being ran through John the ripper and using the rockyou.txt file to perform the dictionary attack. We were instructed to let this run for at least 10 minutes, but it completed fairly quickly and only cracked 6 of the 9 hashes that were provided.

3. **10 points.** 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.)

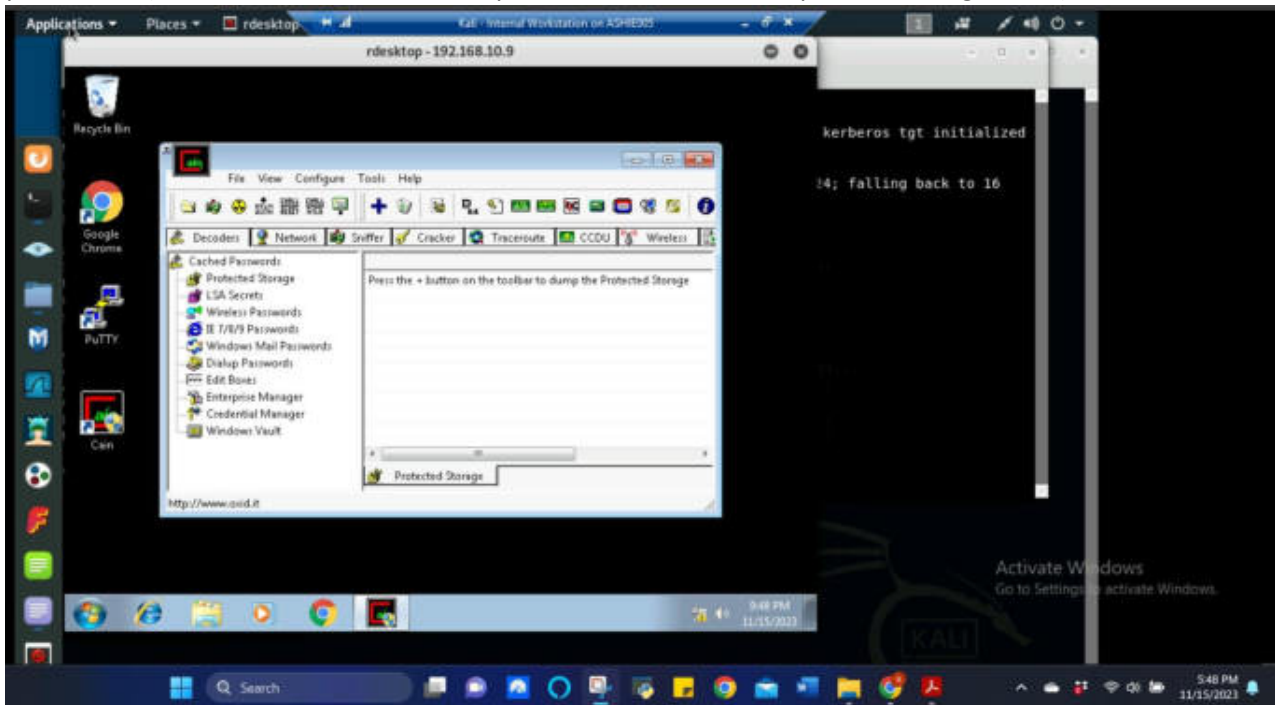


Figure 12 Screenshot of Cain and Abel being installed on Windows 7 VM through remote desktop on Internal Kali for Task B.3

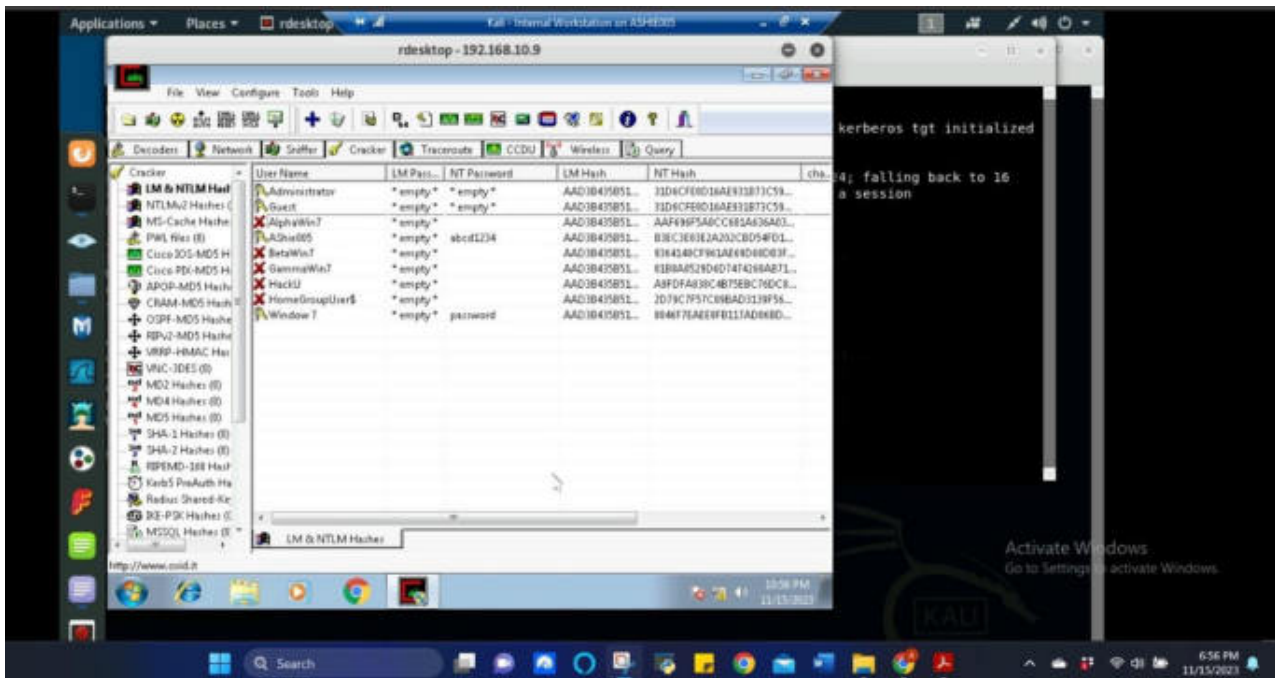


Figure 13 Screenshot of Cain and Abel being used on the WinHashes retrieved using both Brute Force and Dictionary attacks to crack the passwords for Task B.3

The above screenshots show the Cain and Abel being installed on Windows 7 using remote desktop through Internal Kali. Once Cain and Abel was installed, both brute force and dictionary attacks were performed on the hashes that were retrieved to attempt to crack the passwords. The Administrator, Guest passwords were determined to be blank, ASHie005 password was cracked abcd1234 and Window 7 was cracked password. The remaining users with red Xs were not cracked due to the amount of time would be needed to perform the attacks.

TASK C: EXTRA CREDIT (25 POINTS)

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
2. 63a9f0ea7bb98050796b649e85481845

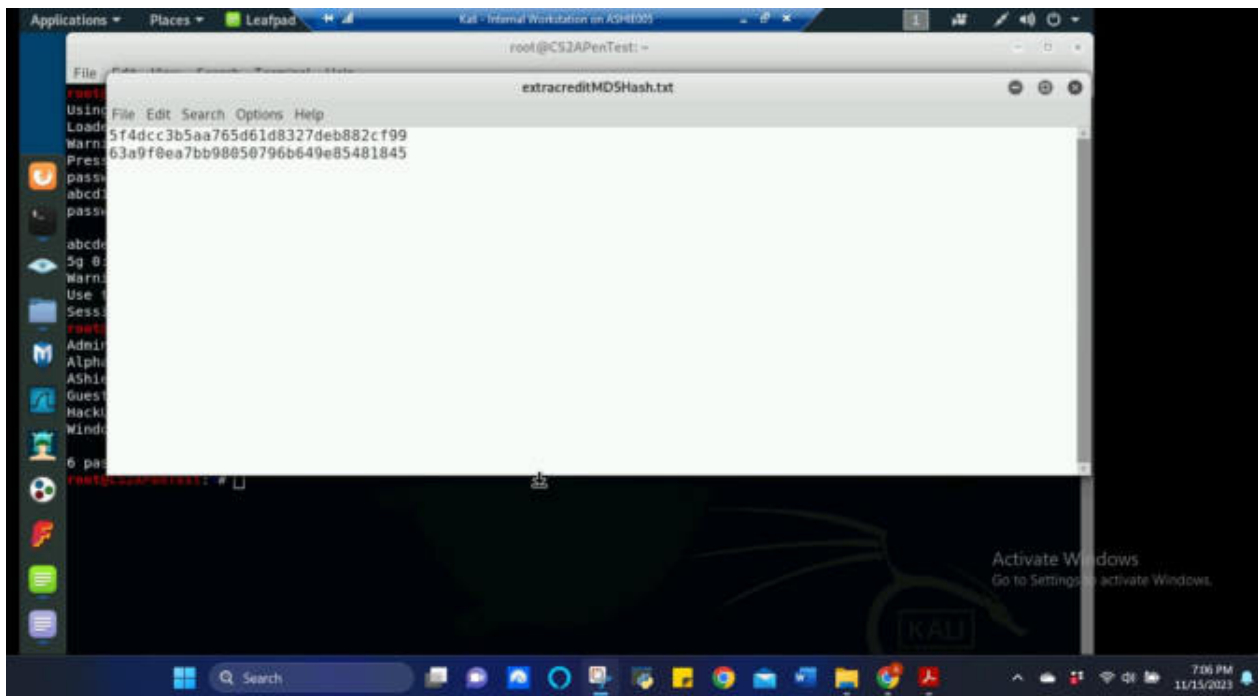


Figure 14 Screenshot of given MD5 hashes given placed in file called extracreditMD5Hash.txt for Task EC

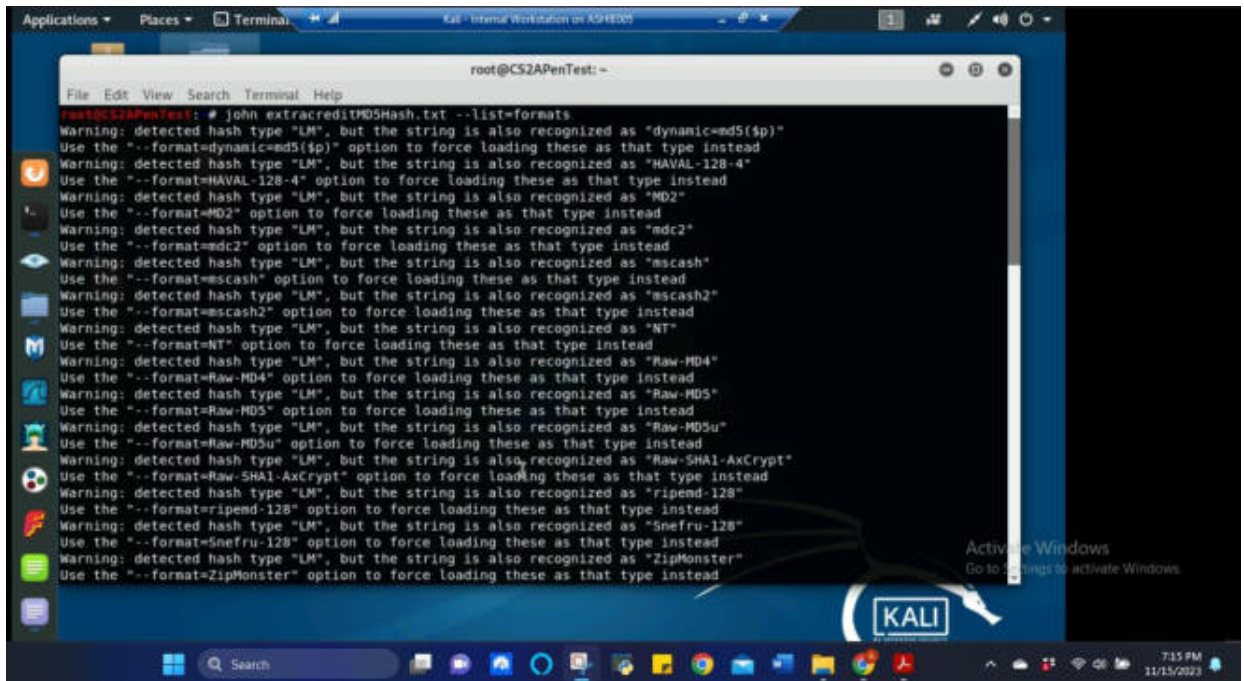


Figure 15 Screenshot of list of formats needed to crack the hashes for Task EC

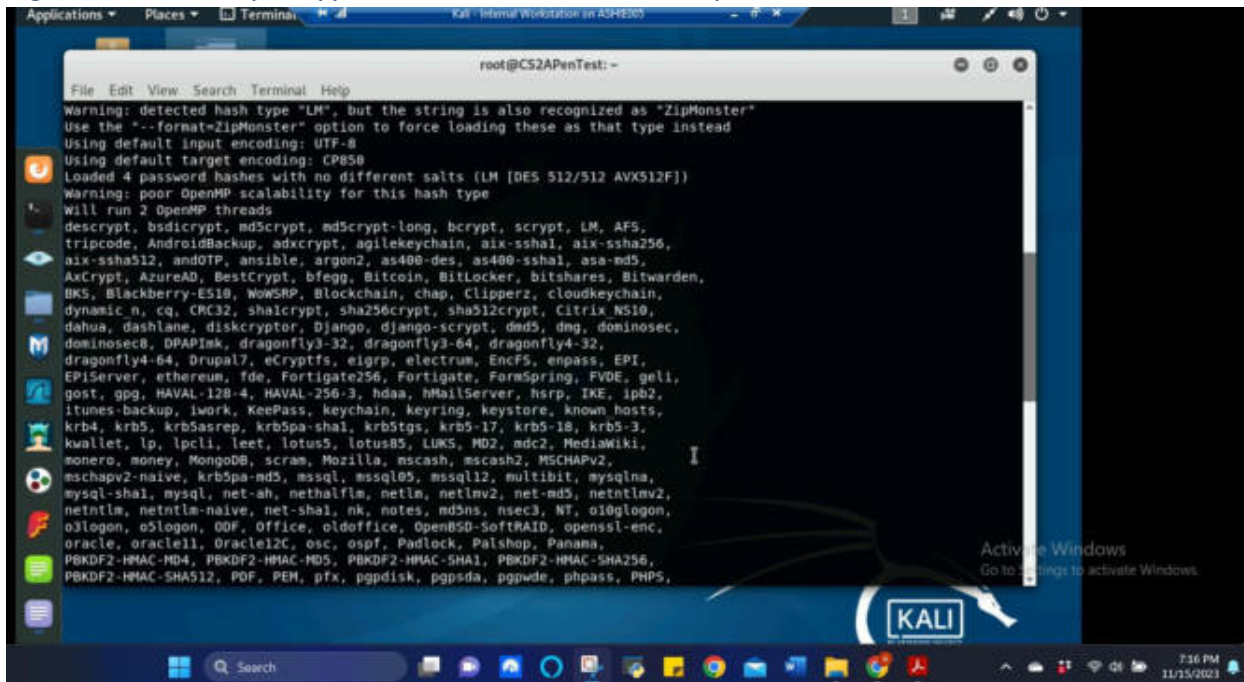


Figure 16 Screenshot of list of formats (continued) needed to crack the hashes for Task EC

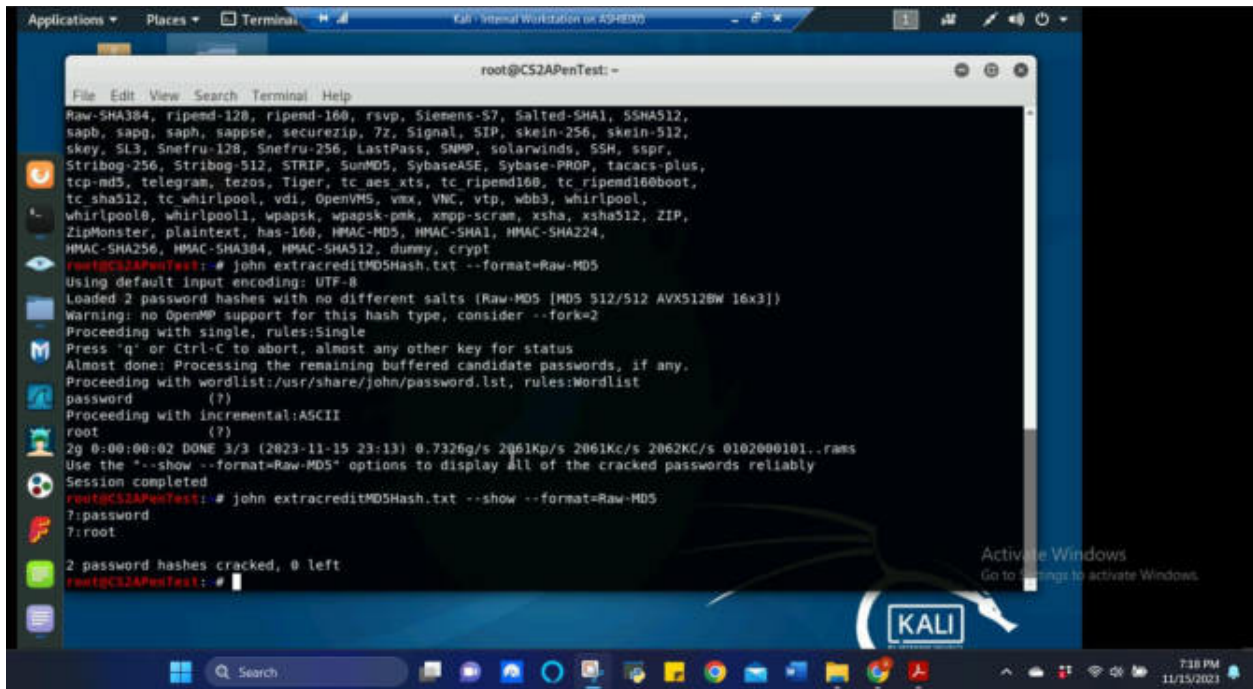


Figure 17 Screenshot of using Raw-MD5 format to crack the hashes and results for Task EC

The above screenshots show given MD5 hashes, the hashes were compiled into a text file named “extracreditMD5Hash.txt” and using John the ripper, the hashes were attempted to be cracked. Since these were MD5 hashes, a support format was needed to extract the password. The Raw-MD5 format was select to be used and the command was executed. The results that came back were that the first hash’s cracked password was password and the second hash’s cracked password was root.

WI-FI PASSWORD CRACKING

TASK A: (40 POINTS)

Follow the steps in the lab manual, and decrypt WEP and WPA/WPA2 protected traffic.

Requirements:

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

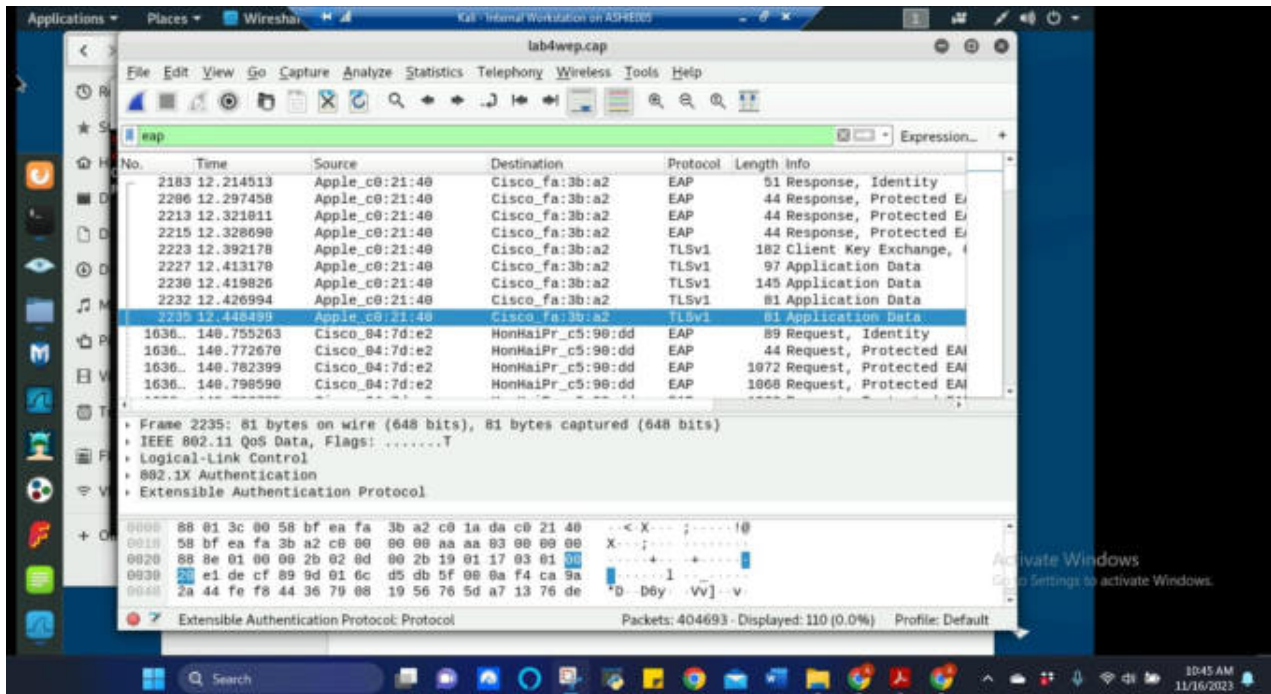


Figure 18 Screenshot of eap filter being used on lab4wep.cap encrypted capture in wireshark for Task A.1

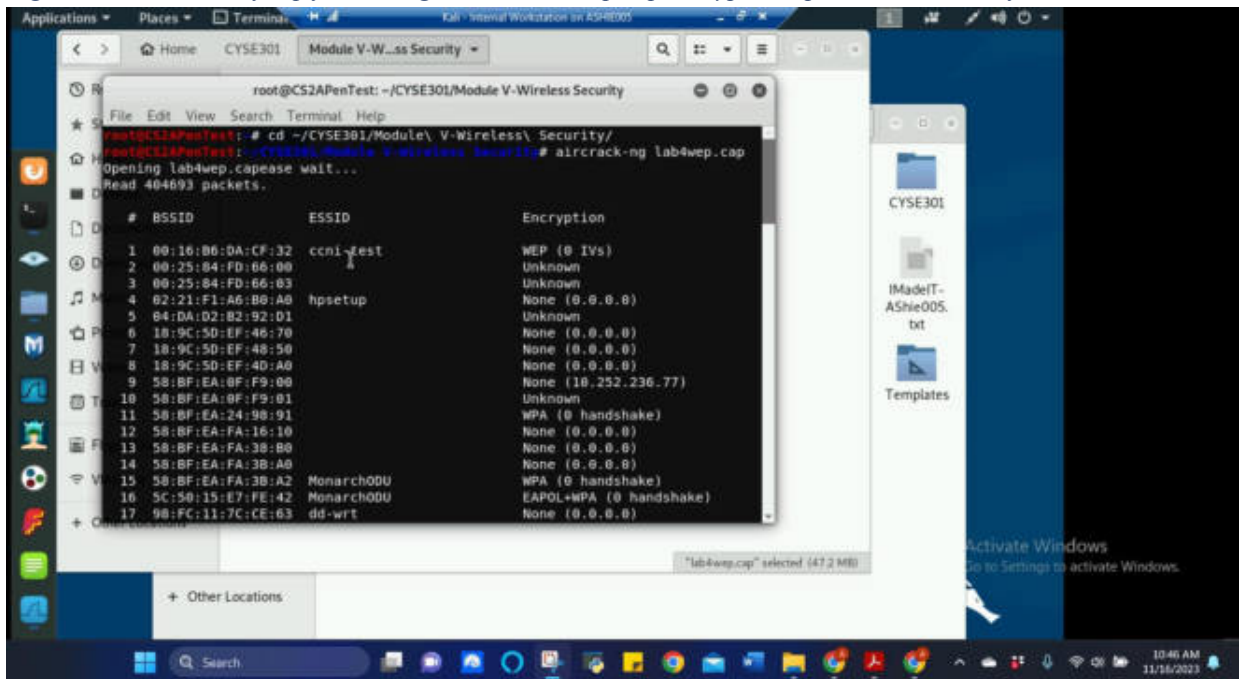


Figure 19 Screenshot of aircrack being used on lab4wep.cap to decrypt the file for Task A.1

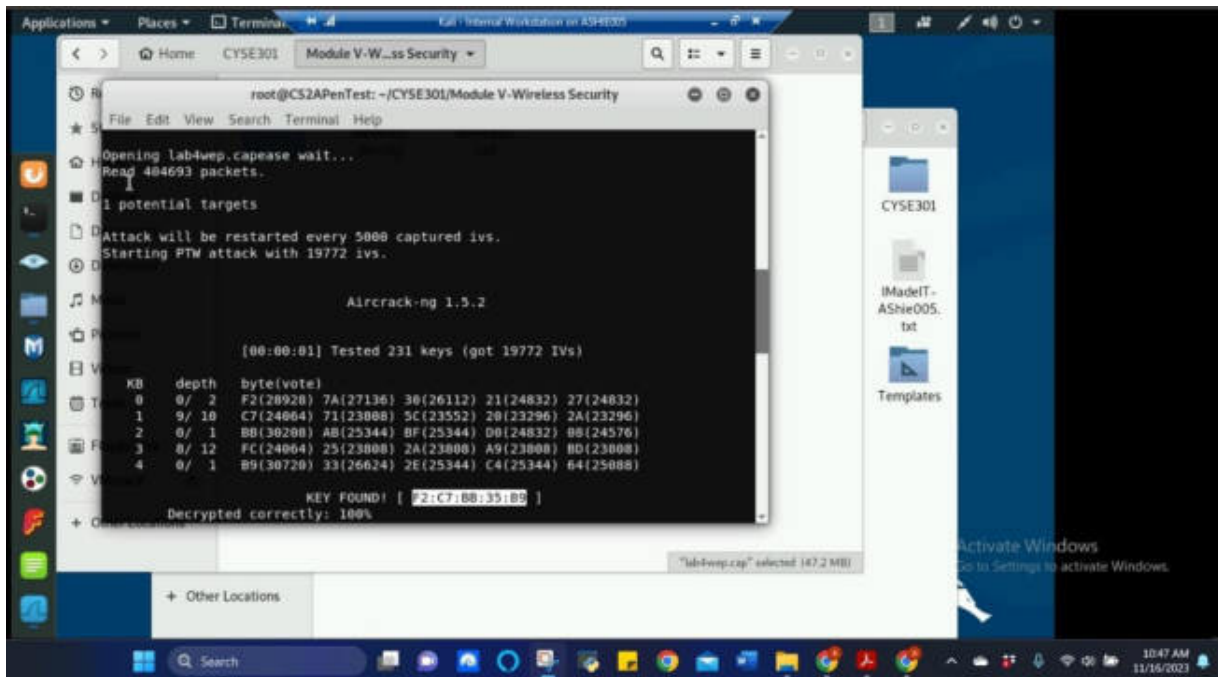


Figure 20 Screenshot of aircrack (continued) being used on lab4wep.cap to decrypt the file and key being found for Task A.1

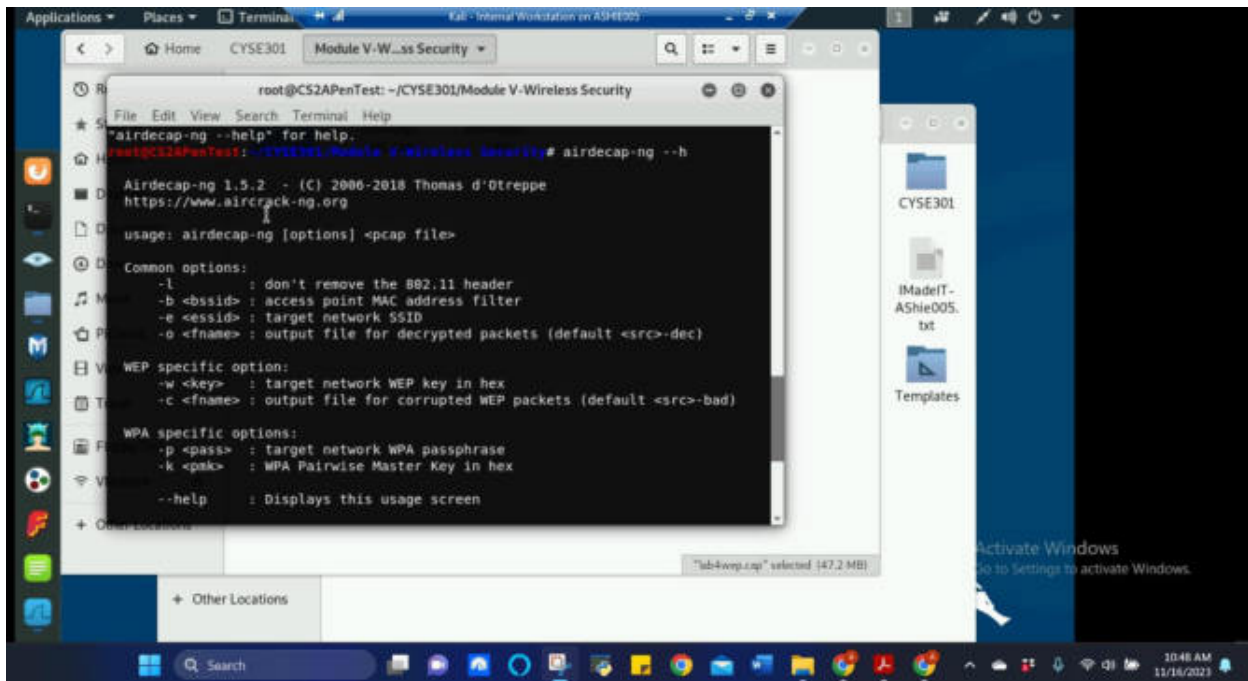


Figure 21 Screenshot of aircrack commands being accessed to determine which to use to decrypt the lab4wep.cap file for Task A.1

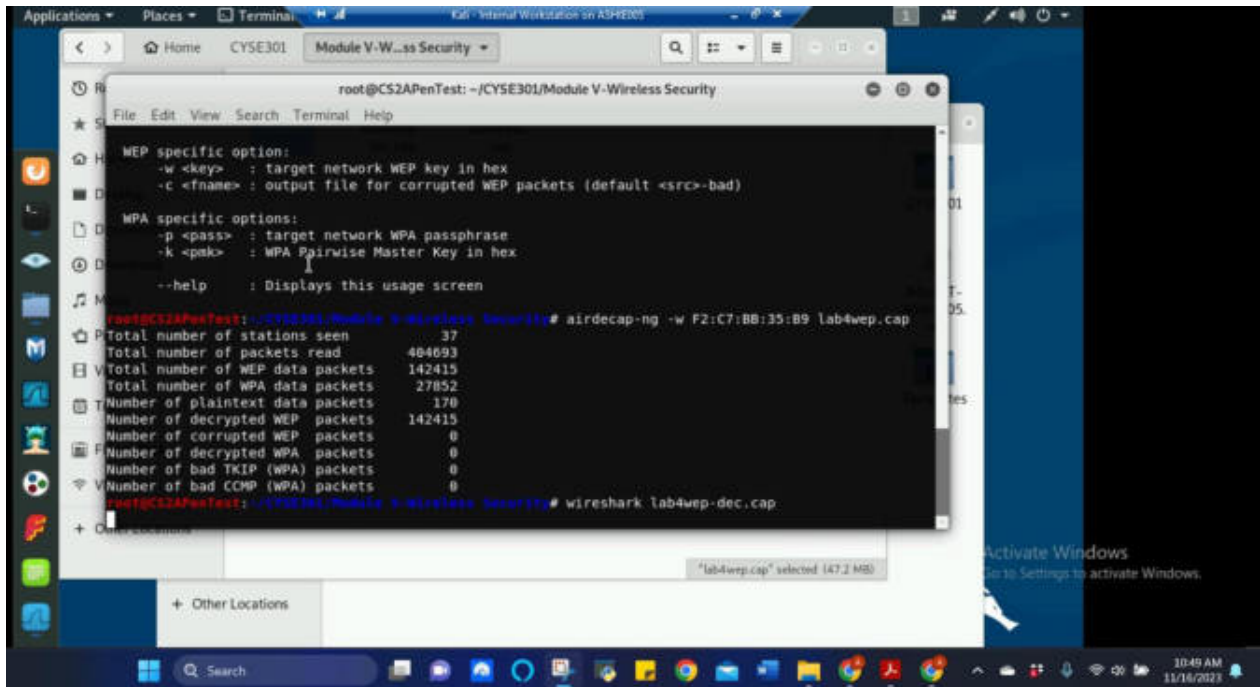


Figure 22 Screenshot of aircrack command with -w being used to decrypt the lab4wep.cap file for Task A.1

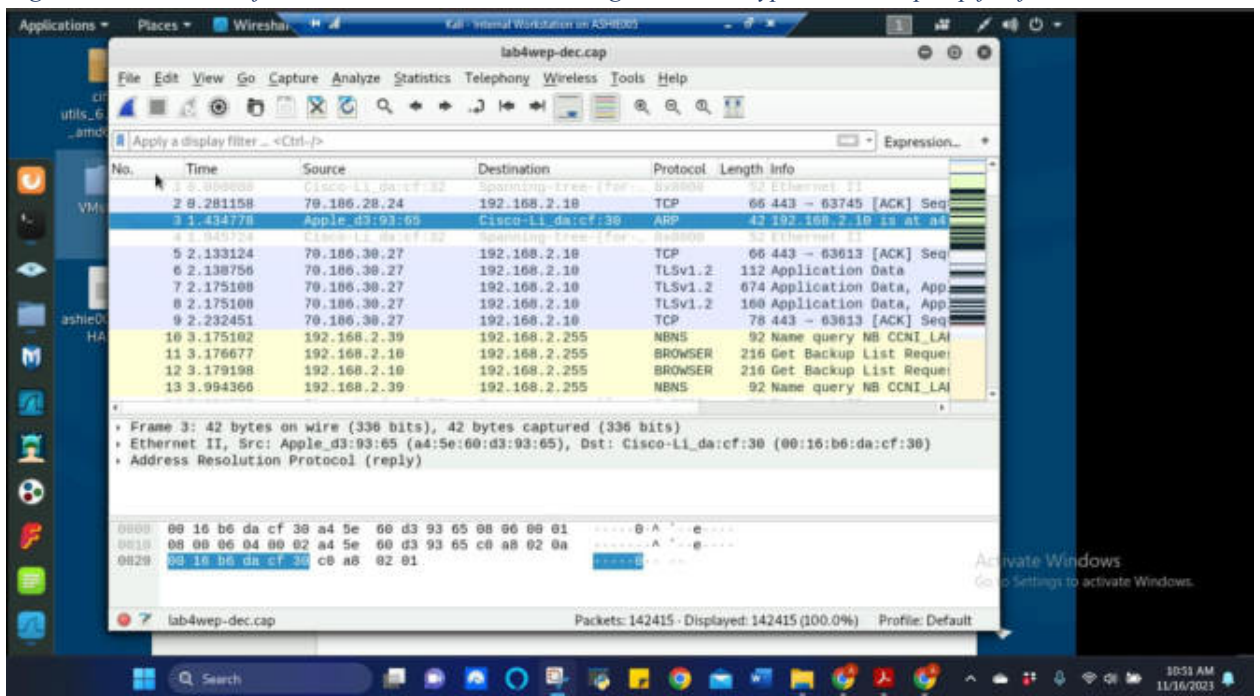


Figure 23 Screenshot of wireshark capture of the decrypted lab4wep-dec.cap file for Task A.1

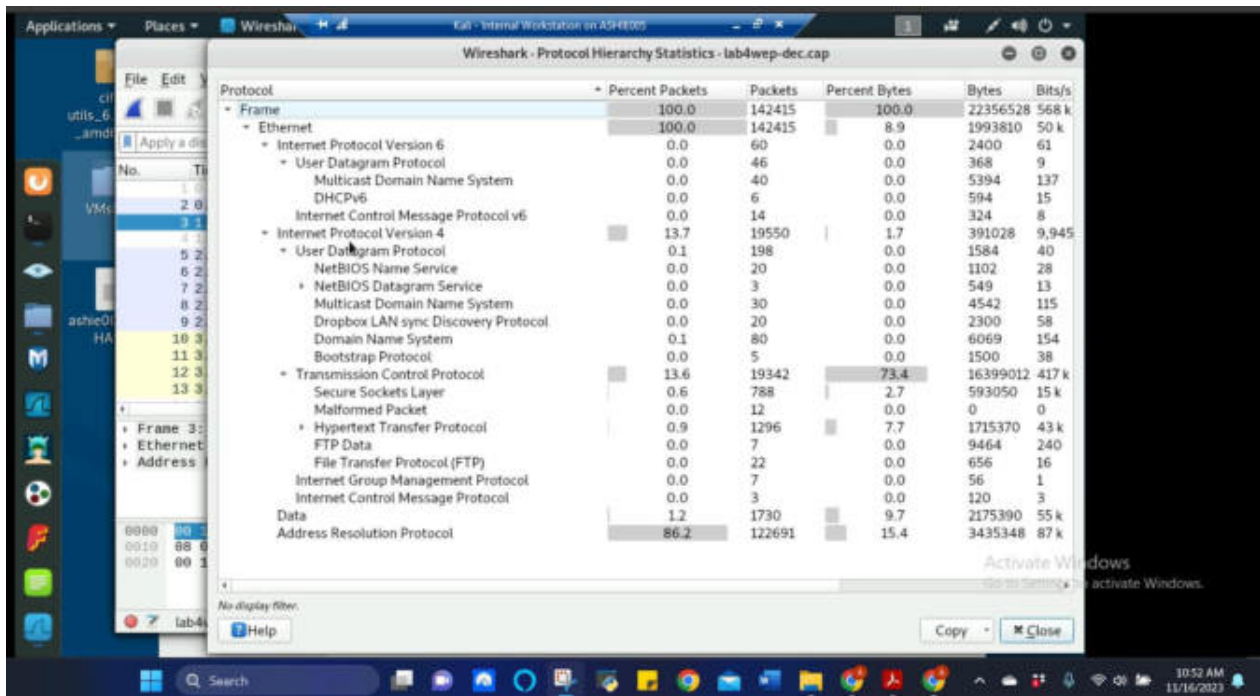


Figure 24 Screenshot of wireshark protocol hierarchy statistics of the decrypted lab4wep-dec.cap file for Task A.1

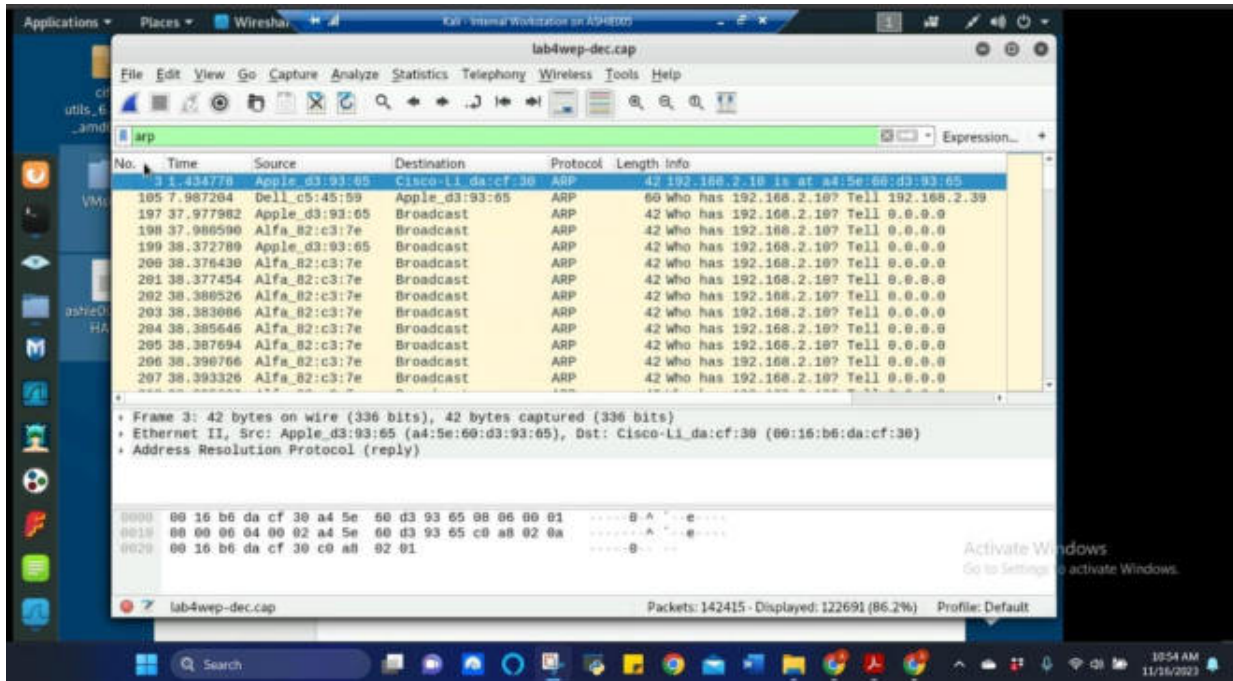


Figure 25 Screenshot of wireshark capture using arp filter of the decrypted lab4wep-dec.cap file for Task A.1

Looking at the decrypted packet traffic and the Protocol Hierarchy Statistics, you see out of the 142415 packets obtained, 19342 (13.6%) were TCP packets and 122691 (86.2%) were ARP packets. In looking deeper into why there were so many ARP requests sent the ARP request kept asking who has the IP address 192.168.2.10, which was an Apple device with the MAC address a4:5e:60:d3:93:65. One ARP came from a Dell device with the MAC address f8:b1:56:c5:45:59 and the IP address of 192.168.2.39. The remaining ARP requests repeatedly asked for the IP address 192.168.2.10 was a Broadcast and was primarily initiated by Alfa device with the MAC address 00:c0:ca:82:c3:7e. With the Alfa device being the source, this would indicate that an Alfa Wi-Fi dongle antenna is being utilized and it is injecting ARP packets into the AP to cause

IV reuse. In analyzing the http traffic collected, there were a few jpegs in there, but too small to view clearly, but there were many audio files collected, some in mp3 format and some in m4a. Since these files were collected in the http traffic, these files could be from streaming music from a streaming site. Some other observations from the resolved addresses obtained is that Rutgers.edu was visited, dropbox was used, icloud was used, a site www.alobbs.com was shown and going to the website, it shows Alvaro's Site made by Alvaro Lopez Ortega.

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

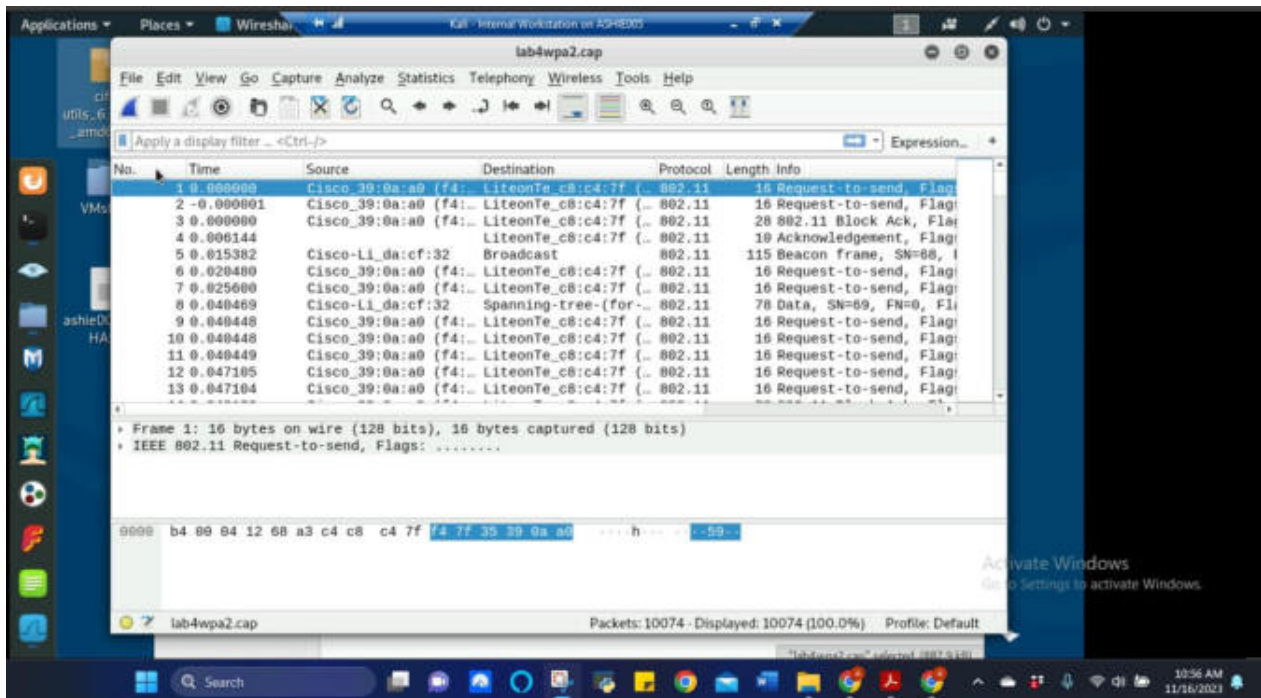


Figure 26 Screenshot of wireshark encrypted capture of lab4wpa2.cap for Task A.2

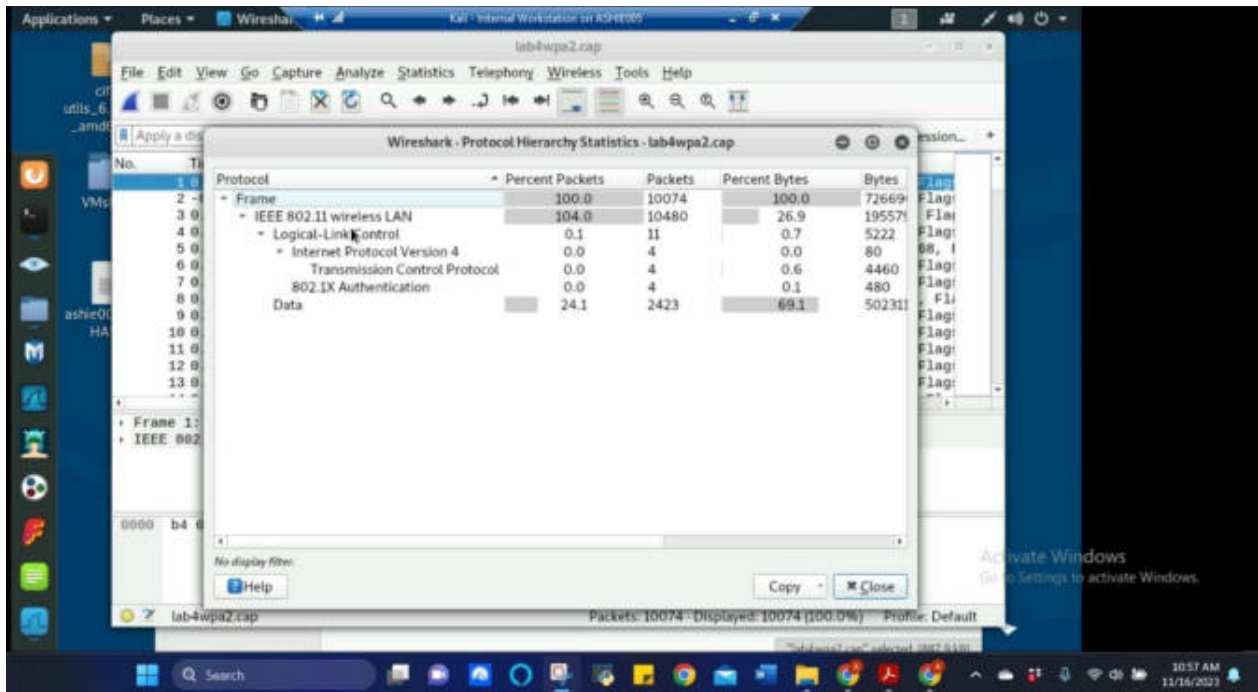


Figure 27 Screenshot of wireshark protocol hierarchy statistics for the encrypted capture of lab4wpa2.cap for Task A.2

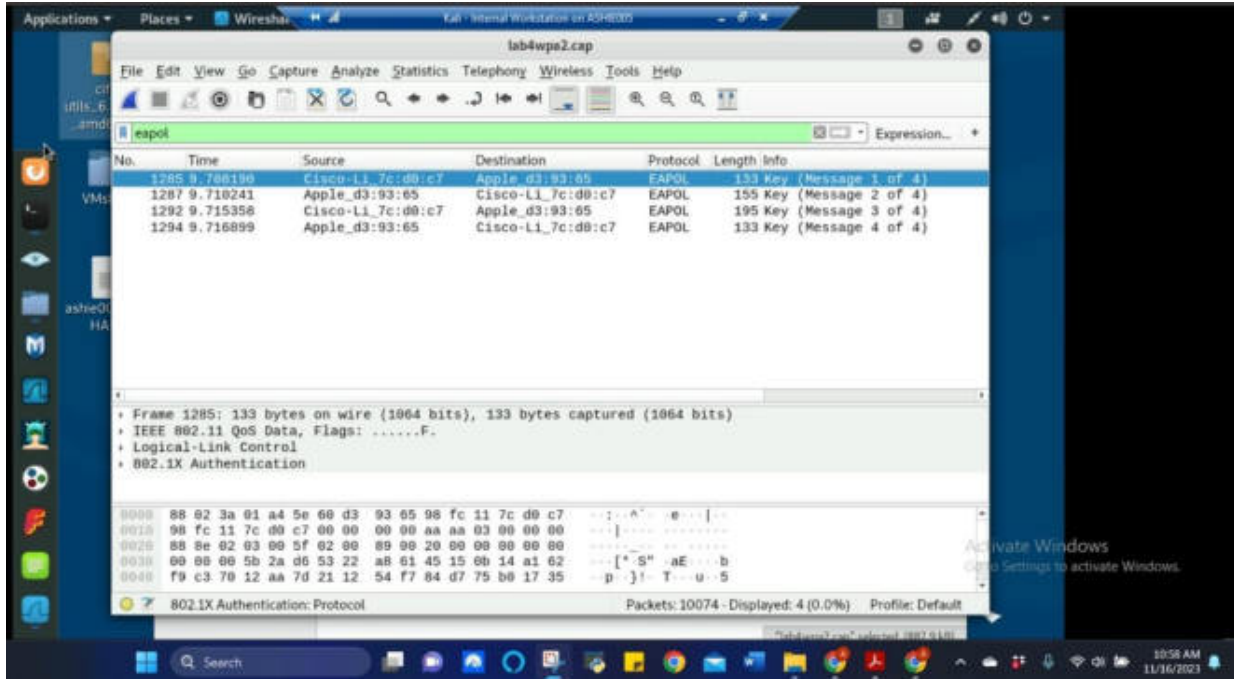


Figure 28 Screenshot of wireshark encrypted capture of lab4wpa2.cap with eapol filter used for Task A.2

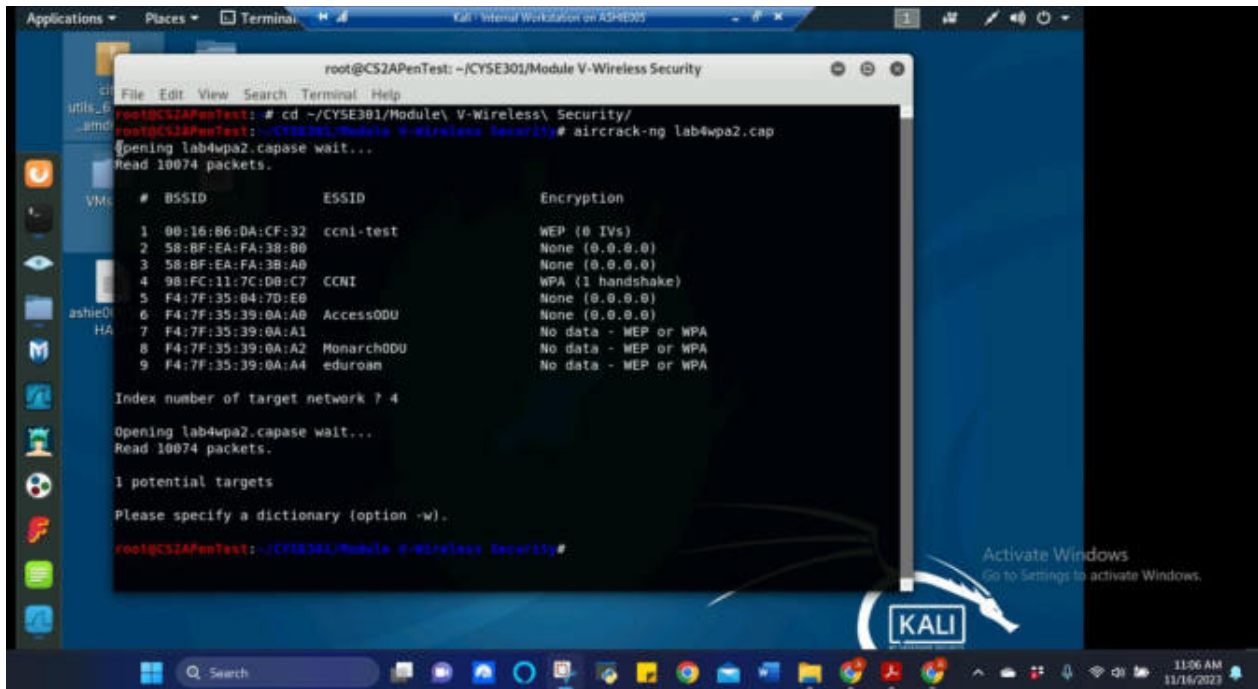


Figure 29 Screenshot of aircrack being used on lab4wpa2.cap to decrypt file for Task A.2

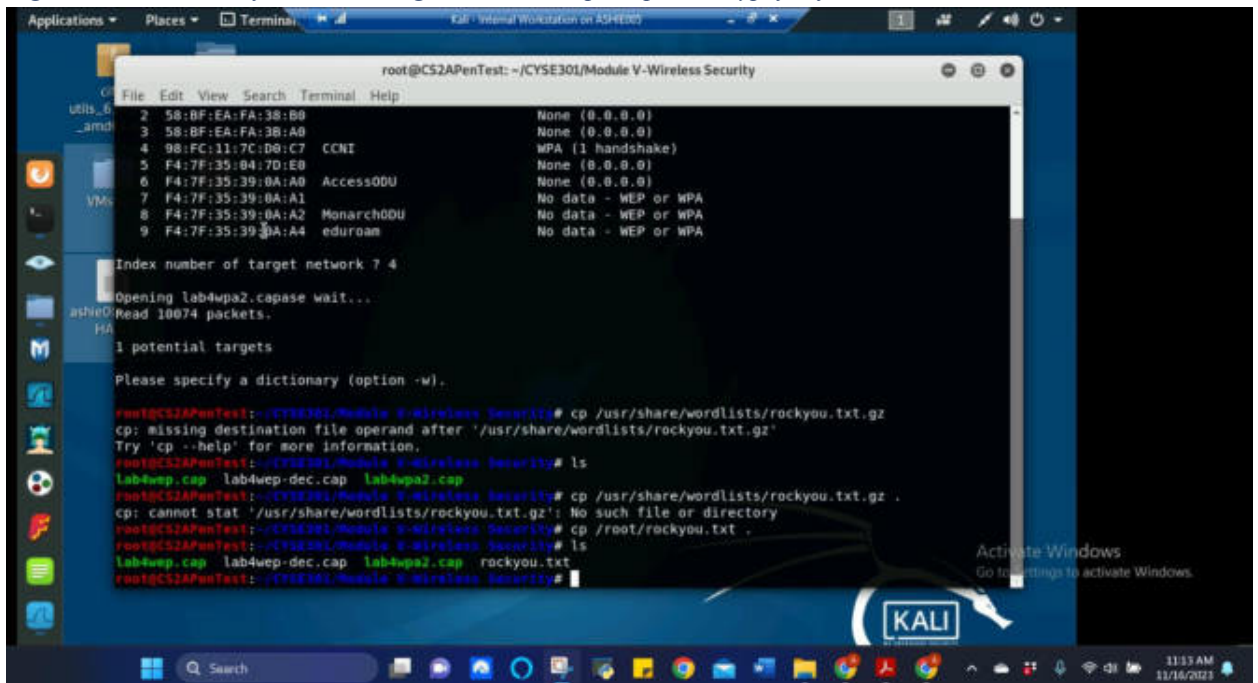


Figure 30 Screenshot of rockyou.txt being copied to directory for use to crack password on lab4wpa2.cap to decrypt file for Task A.2

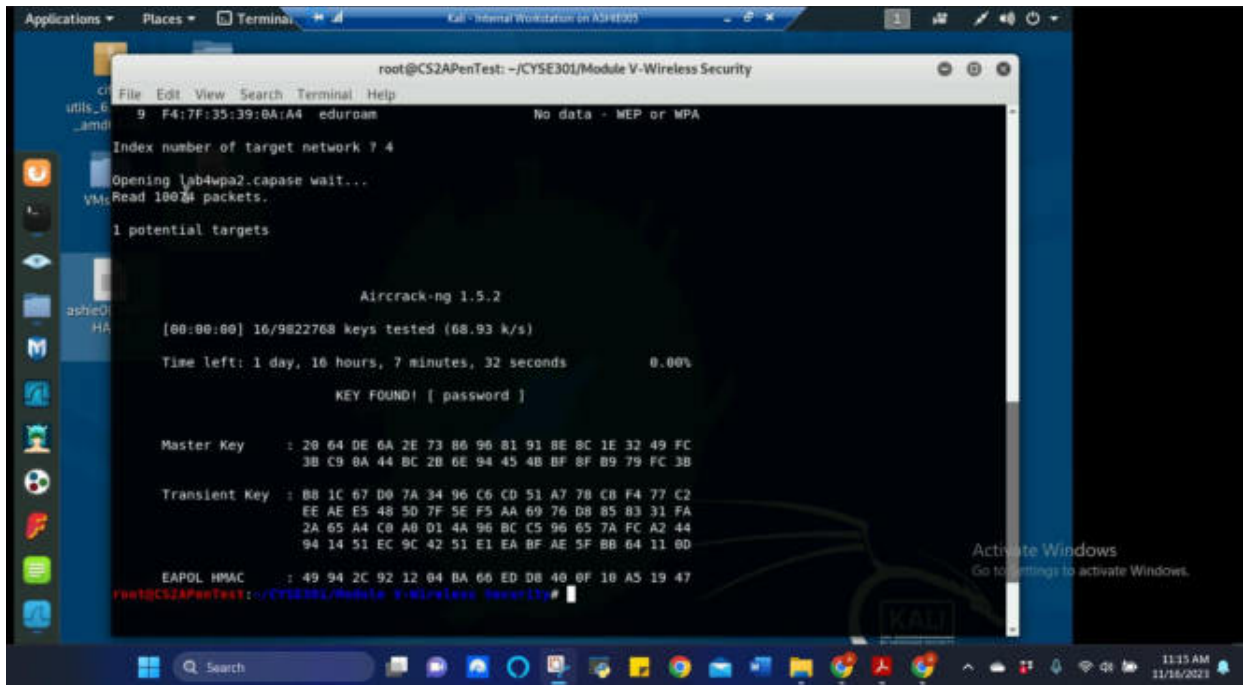


Figure 31 Screenshot of aircrack (continued) being used on lab4wpa.cap to decrypt file and password successfully cracked for Task A.2

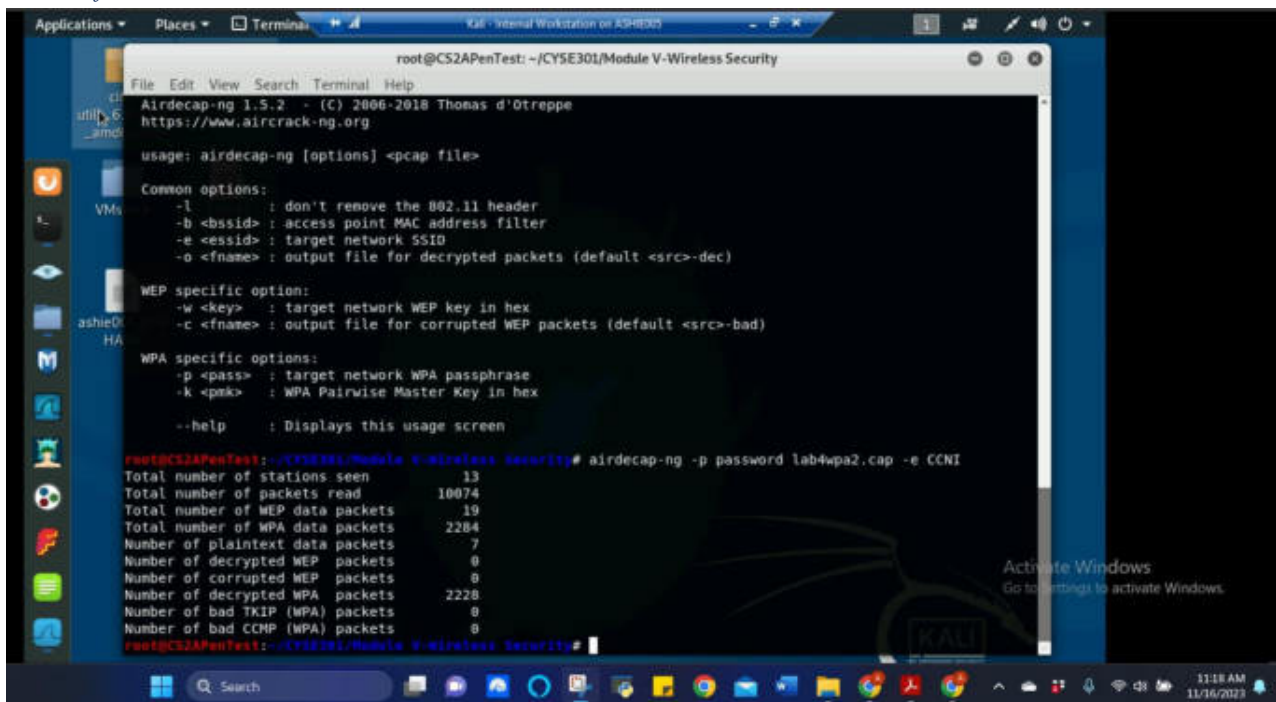


Figure 32 Screenshot of aircrack command with -p, password, -e and CCNI (network SSID) being used to decrypt the lab4wpa2.cap file for Task A.2

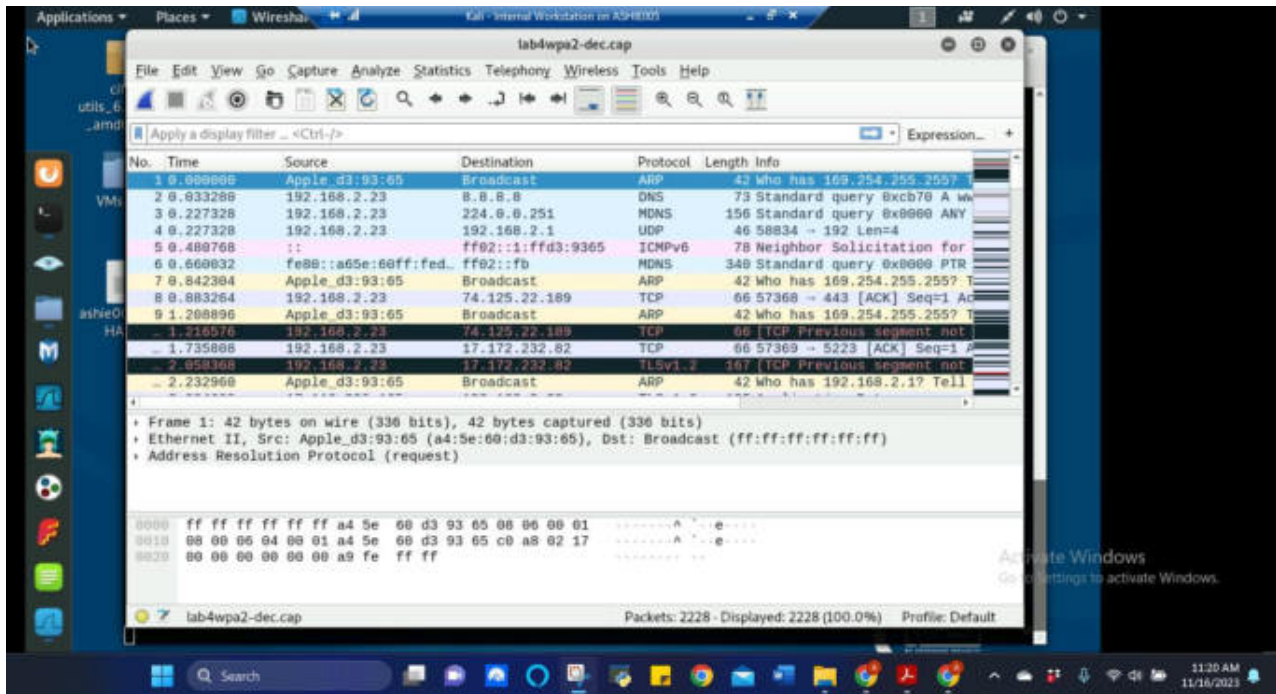


Figure 33 Screenshot of wireshark capture of the decrypted lab4wpa2-dec.cap file for Task A.2

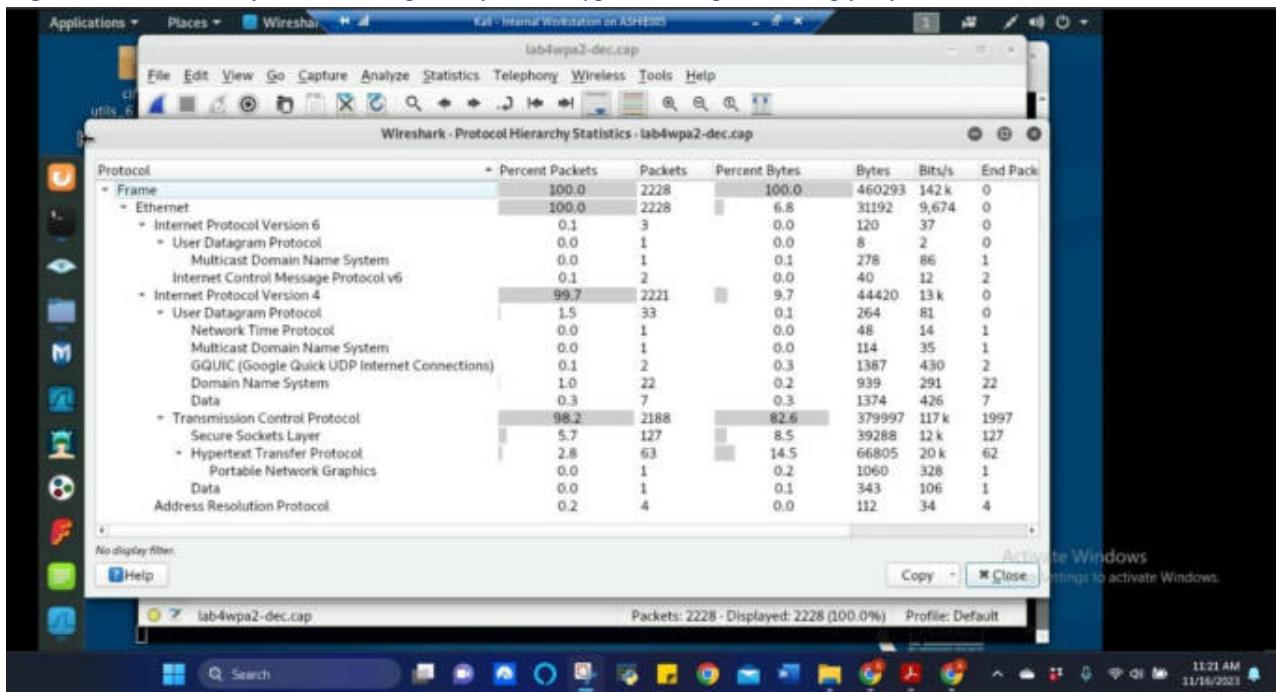


Figure 34 Screenshot of wireshark capture protocol hierarchy statistics of the decrypted lab4wpa2-dec.cap file for Task A.2

Looking at the decrypted packet traffic and Protocol Hierarchy Statistics, 2228 packets were captured and 2188 of those packets were TCP packets and far fewer ARP packets than were seen in the WEP capture. The TCP packets show multiple warnings throughout the handshake process with the message “TCP Previous segment not captured” or “TCP ACKed unseen segment”. There were also many TLSv1.2 protocol packets as well, which is the protocol used for encrypted messages. In observing the resolved addresses it showed addresses for plus.google.com, which is Google plus. Since this encrypted capture file was made in 2015, the packets were captured when Google plus was still operational. Google plus was shutdown on April 2,

2019, so unable to access the website anymore. In the http traffic, I can see the traffic occurs on ODU and can see in November of 2015 the news section was visited where it mentions of summer school and honoring veterans from IP address 128.82.112.29 to 192.168.2.23. The exported https information contained many object files and a couple of jpeg files that could not be opened due to errors.

WI-FI PASSWORD CRACKING

TASK B: (60 POINTS)

Each student will be assigned a new WPA2 traffic file for analysis. You need to refer to the table below and find the file assigned to you based on the LAST digit of the MD5 of your MIDAS ID. For example, the last digit of the hash for pjiang is e. Thus, I should pick up file "WPA2-P5-01.cap." MD5 of pjiang is 5a618cdc3edffd8b4c661e7e9b70ce1e

Your String	ashie005
MD5 Hash	973becb0703a7631fefbae9e4433c27d <input type="button" value="Copy"/>

Figure 35 Screenshot of MIDAS ID MD5 being generated to determine file the encrypted file needed for Task B

Then complete the following steps:

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

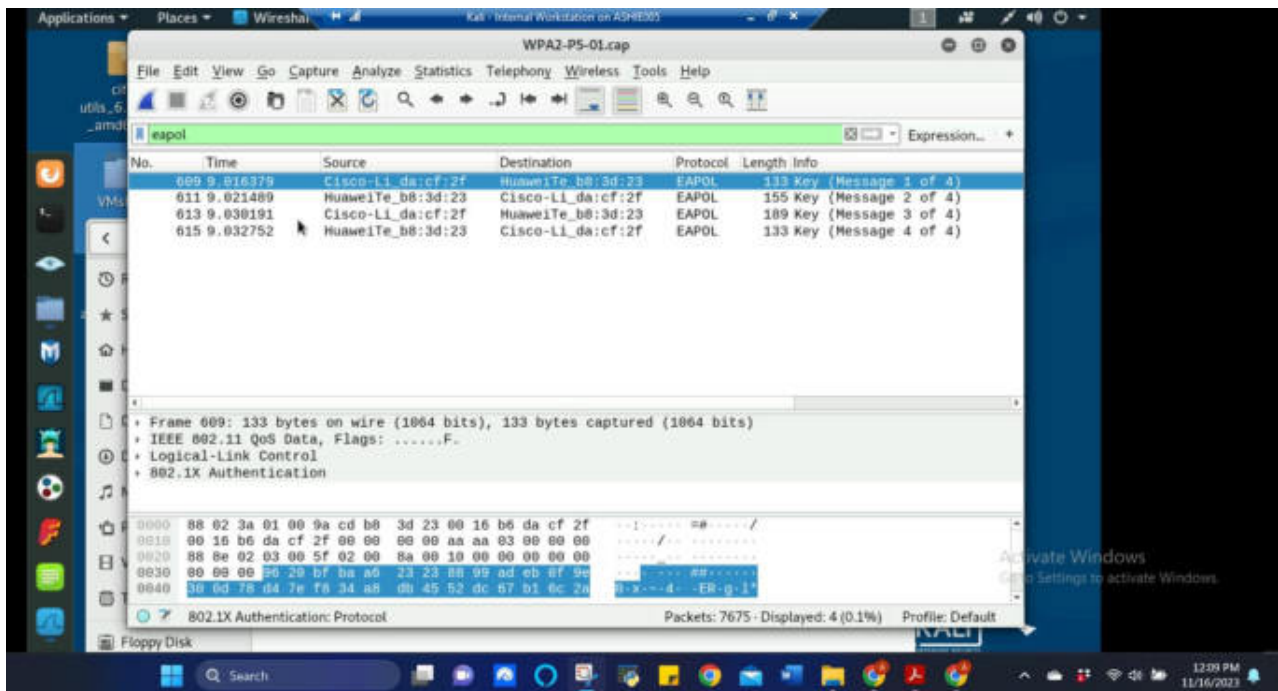


Figure 36 Screenshot of wireshark capture using the eapol filter to show the 4-way handshake on WPA2-P5-01.cap file for Task B.1

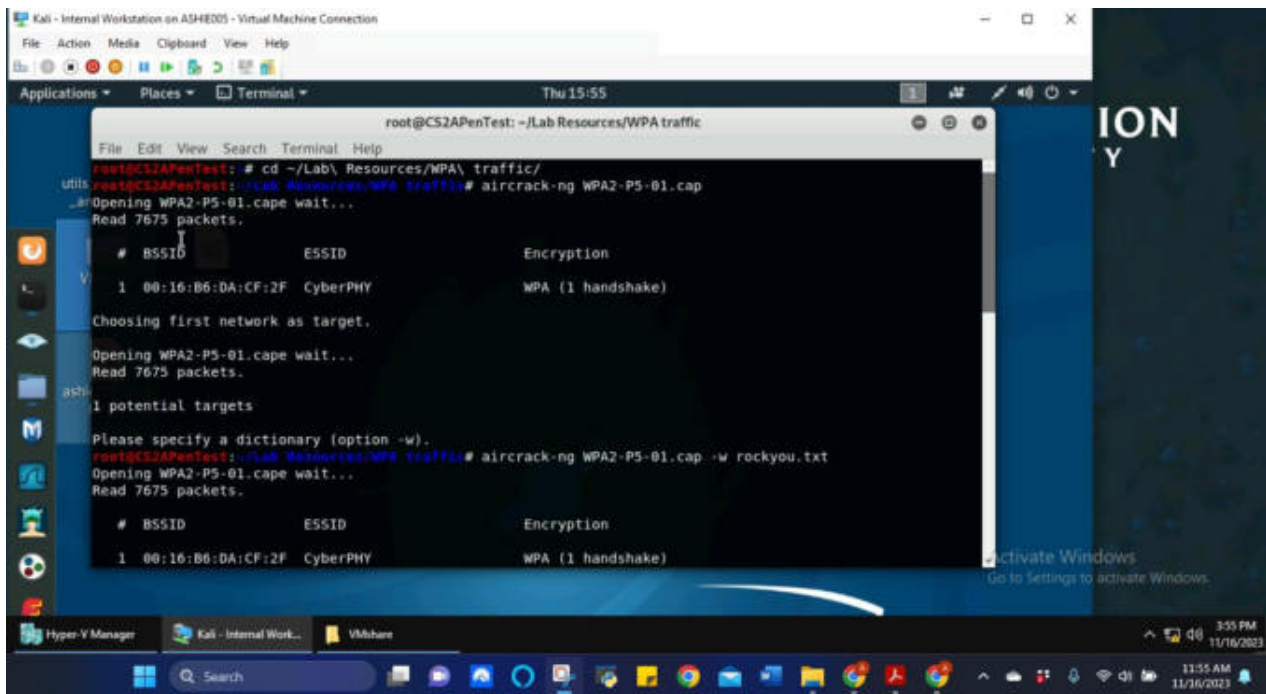


Figure 37 Screenshot of aircrack being used on WPA2-P5-01.cap and rockyou.txt to decrypt file for Task B.1

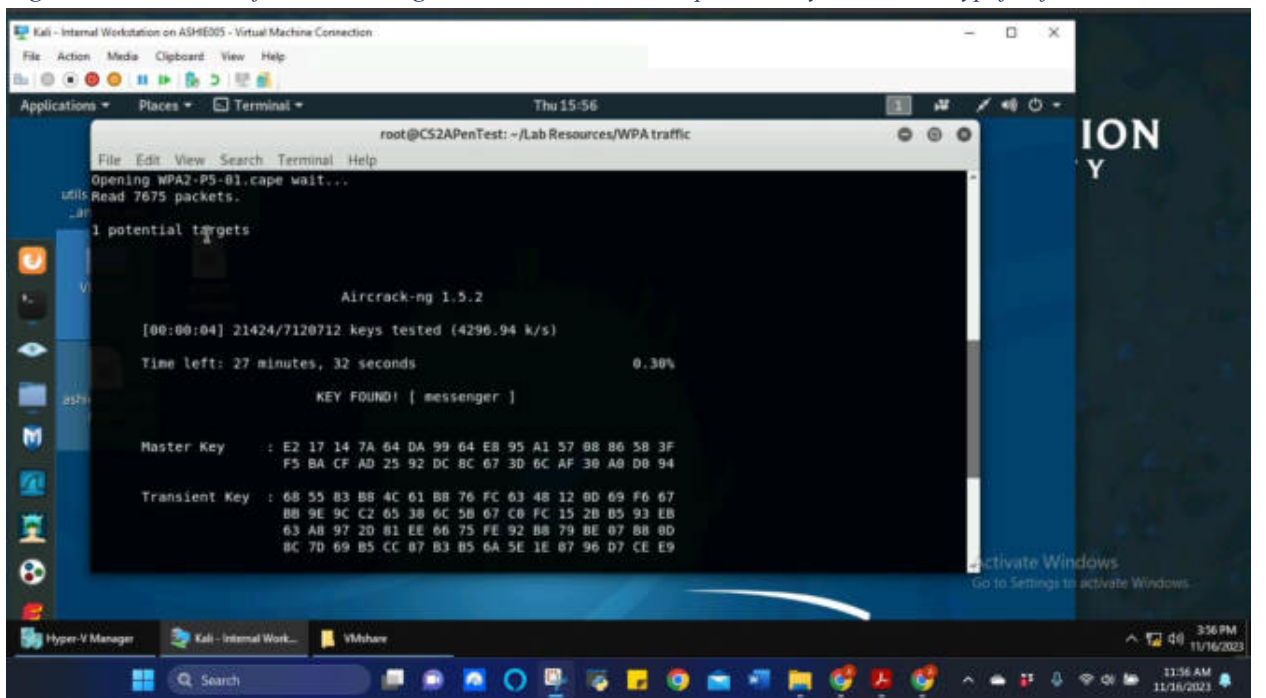


Figure 38 Screenshot of aircrack being used on WPA2-P5-01.cap to decrypt file and successfully cracking password for Task B.1

The Screenshots above show that after the MD5 hash was determined for my MIDAS ID, the last digit was a “d”, so that directed me to use the WPA2-P5-01.cap encrypted file given. Using the aircrack command and the rockyou.txt file, the password was successfully cracked and determined to be “messenger”. That password in the next section will be used to decrypt the WPA2-P5-01.cap file.

2. Decrypt the encrypted traffic and write a detailed summary to describe what you have explored from this encrypted traffic file. -30 points

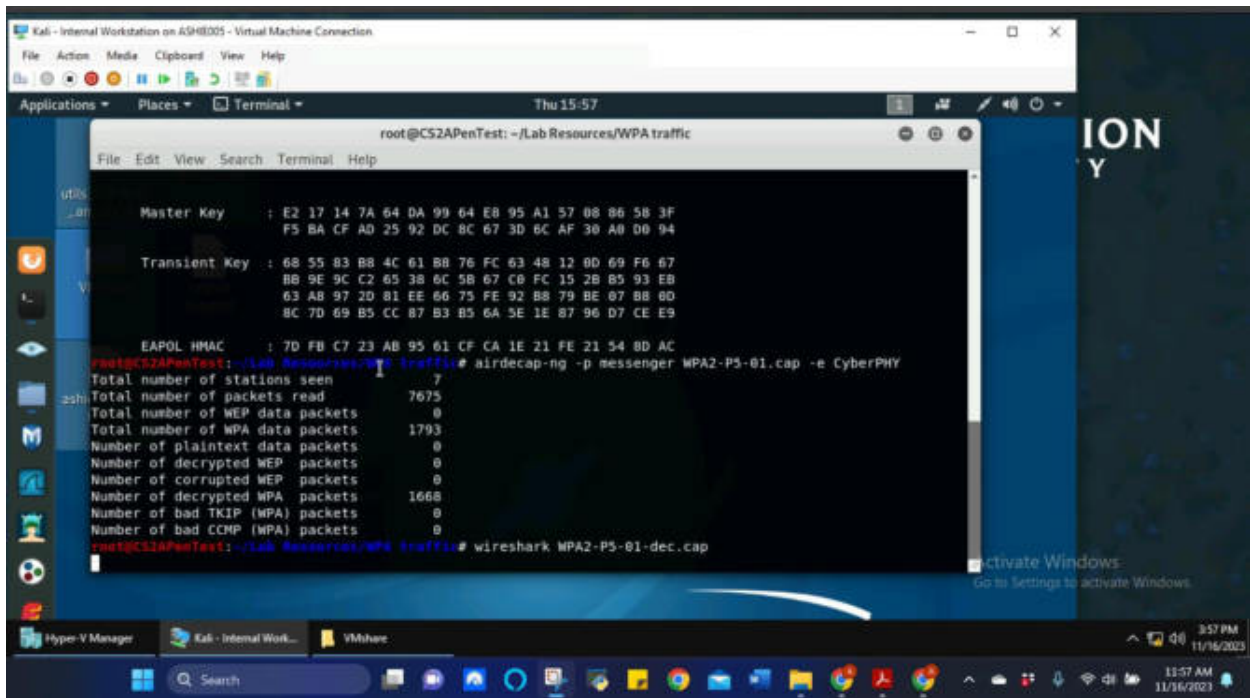


Figure 39 Screenshot of aircrack being used on WPA2-P5-01.cap with -p, messenger, -e and CyberPHY (network SSID) to decrypt file for Task B.2

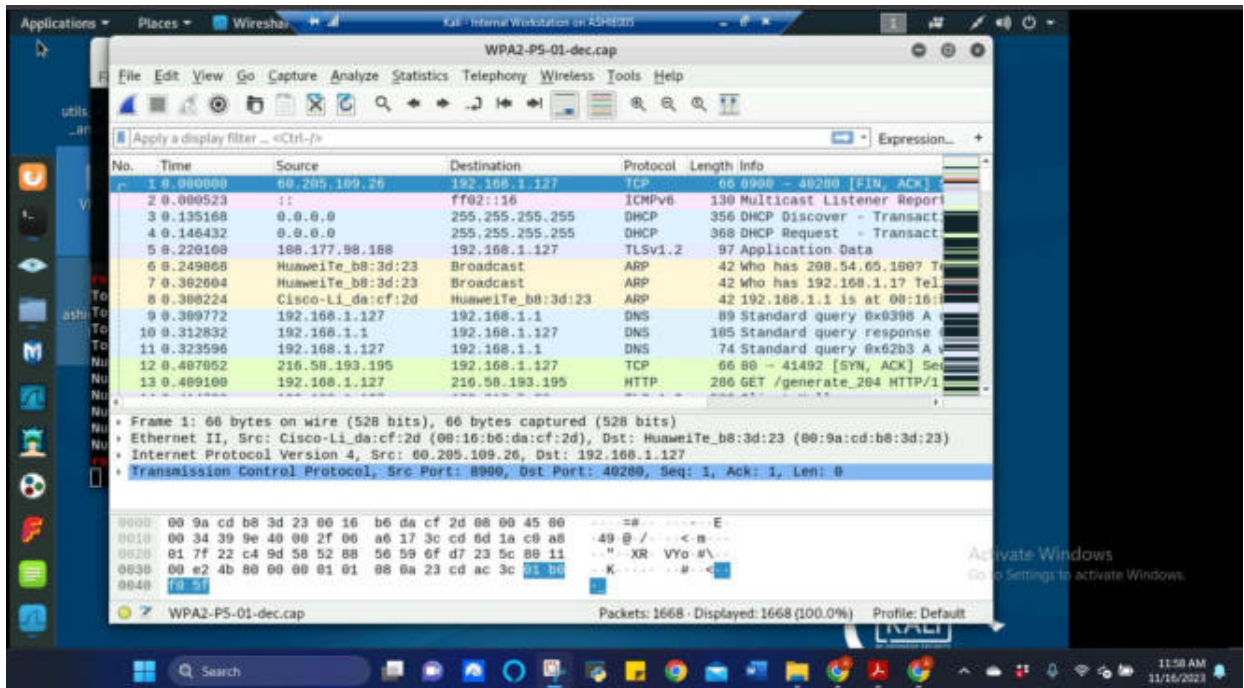


Figure 40 Screenshot of wireshark capture of WPA2-P5-01-dec.cap decrypted file for Task B.2

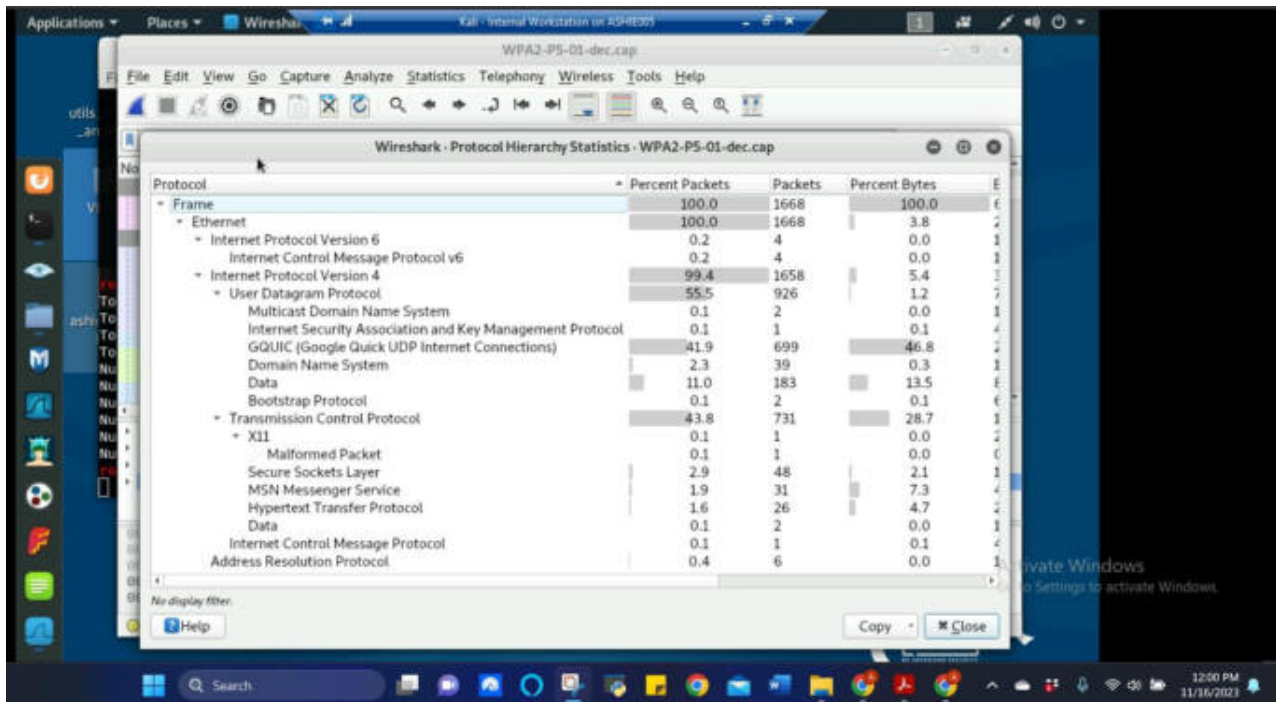


Figure 41 Screenshot of wireshark capture protocol hierarchy statistics of WPA2-P5-01-dec.cap decrypted file for Task B.2

Looking at the decrypted packet traffic, the first thing I noticed was there is a Huawei Technology device broadcasting and sending ARP request to find 208.54.65.100 and 192.168.1.1 and the Huawei device has the IP address of 192.168.1.127 and the MAC address of 00:9a:cd:b8:3d:23. Looking at the Protocol Hierarchy Statistics, the packets are primarily in IPV4 with the packets between UDP (55.5%) and TCP (43.8%). Some of the websites that were visited during this capture in UDP were www.google.com, www.taobao.com, www.youtube.com, manifest.googlevideo.com. Similar to the Lab4wpa2 decryption, in TCP, there were a few “TCP ACKed unseen segment” messages and “TCP Previous segment not captured” messages. There were also so Resets that took place during the handshake. A difference I did notice was although there are a few TLSv1.2 protocols used, I noticed that SSLv3 protocol was being used also. SSLv3 stands for Secure Sockets Layer and was replaced some time ago with the Transport Layer Security as far as the handshake is concerned. I also did notice that the Huawei device with the IP address of 192.168.1.127 utilized the GQUIC protocol in UDP and the payload shows as being encrypted. GQUIC, which stands for Google Quick UDP Internet Connections, is a new encrypted transport layer network protocol and is believed to be more secure, efficient, and faster. Since there were a total of 1793 packets and only 1668 decrypted, these GQUIC packets are still encrypted because of the enhanced protocol used.