Required Logs for Mimikatz models:

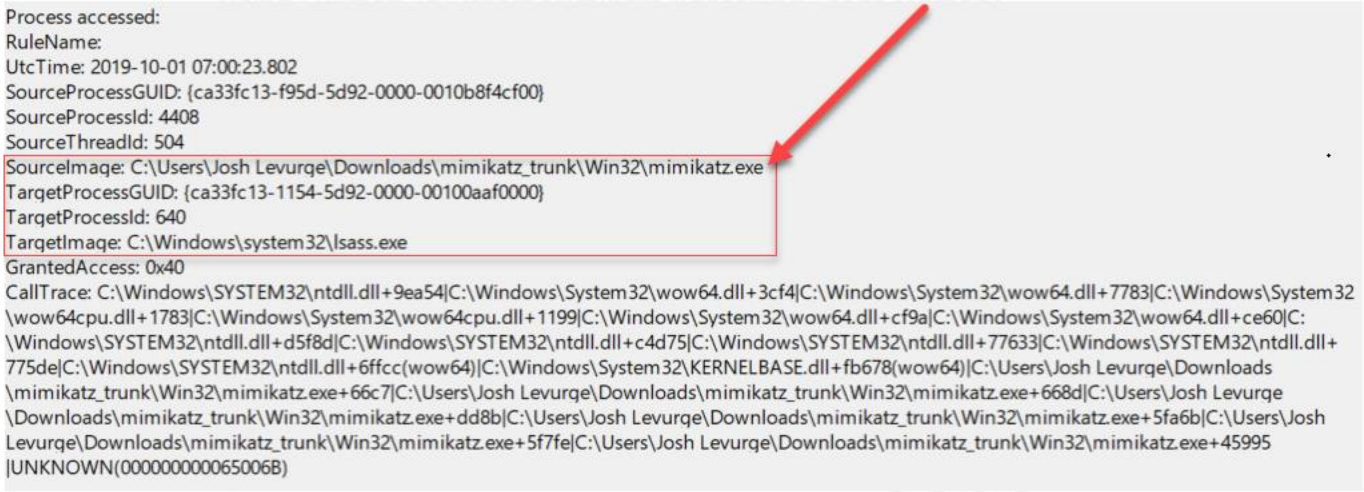
Resource	Event ID	Fields
Sysmon	7	ImageLoaded
	10	SourceImage, TargetImage, GrantedAccess
	11	TargetFilename
Windows Security	4103, 4104	Commanddetails
	4673	Privileges
	4688	newprocessname
	4703	Privileges, Process Name
	4656	Process Name, Accesses, Access Mask
	4663	Process Name, Access Mask
	4690	Source Process ID, Target Process ID

Mimikatz Detection:

1. Mimikatz detection based on Lsass Access

Resources: Sysmon, Event ID: 10, Fields: SourceImage, TargetImage

Sysmon is required as it provides both the parent process [SourceImage] and child process [TargetImage]. Monitor for any process running from a weird location [not System32 nor Syswow64] accessing the lsass.exe.



```
Process accessed:
RuleName:
UtcTime: 2019-10-01 07:00:23.802
SourceProcessGUID: {ca33fc13-f95d-5d92-0000-0010b8f4cf00}
SourceProcessId: 4408
SourceThreadId: 504
SourceImage: C:\Users\Josh Levurge\Downloads\mimikatz_trunk\Win32\mimikatz.exe
TargetProcessGUID: {ca33fc13-1154-5d92-0000-00100aaf0000}
TargetProcessId: 640
TargetImage: C:\Windows\system32\lsass.exe
GrantedAccess: 0x40
CallTrace: C:\Windows\SYSTEM32\ntdll.dll+9ea54|C:\Windows\System32\wow64.dll+3cf4|C:\Windows\System32\wow64.dll+7783|C:\Windows\System32
\wow64cpu.dll+1783|C:\Windows\System32\wow64cpu.dll+1199|C:\Windows\System32\wow64.dll+cf9a|C:\Windows\System32\wow64.dll+ce60|C:
\Windows\SYSTEM32\ntdll.dll+d5f8d|C:\Windows\SYSTEM32\ntdll.dll+c4d75|C:\Windows\SYSTEM32\ntdll.dll+77633|C:\Windows\SYSTEM32\ntdll.dll+
775de|C:\Windows\SYSTEM32\ntdll.dll+6ffcc(wow64)|C:\Windows\System32\KERNELBASE.dll+fb678(wow64)|C:\Users\Josh Levurge\Downloads
\mimikatz_trunk\Win32\mimikatz.exe+66c7|C:\Users\Josh Levurge\Downloads\mimikatz_trunk\Win32\mimikatz.exe+668d|C:\Users\Josh Levurge
\Downloads\mimikatz_trunk\Win32\mimikatz.exe+dd8b|C:\Users\Josh Levurge\Downloads\mimikatz_trunk\Win32\mimikatz.exe+5fa6b|C:\Users\Josh
Levurge\Downloads\mimikatz_trunk\Win32\mimikatz.exe+5f7fe|C:\Users\Josh Levurge\Downloads\mimikatz_trunk\Win32\mimikatz.exe+45995
|UNKNOWN(000000000065006B)
```

2. Mimikatz detection based on Access rights

Resource: Sysmon, Event ID 10, Fields: TargetImage, GrantedAccess

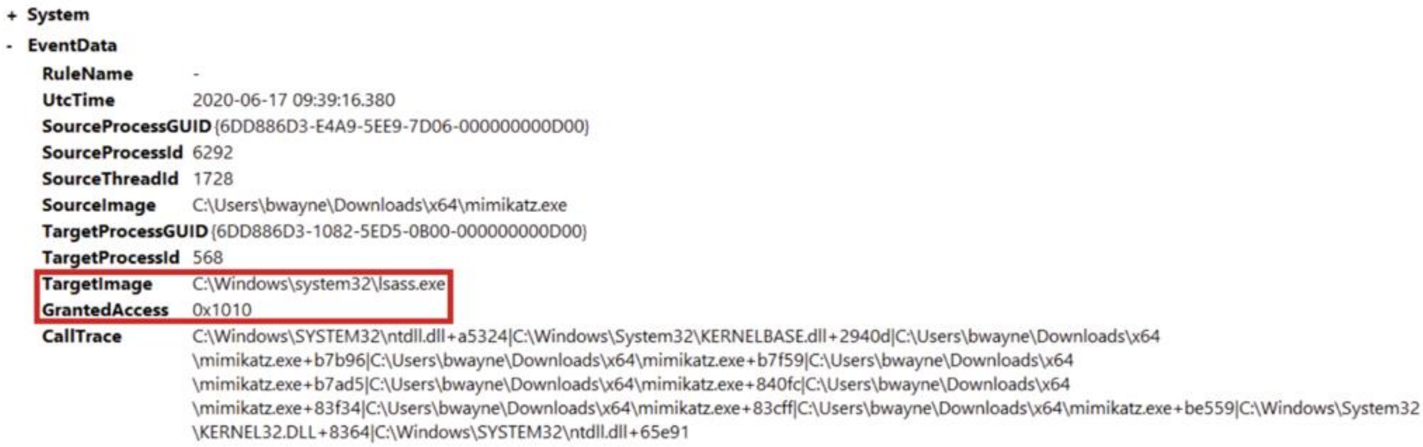
In this detection, we look for the process 'lsass.exe' is being accessed, AND has the following Access rights:

Primarily: '0x1010'

Mimikatz requires specific process access rights to initiate cross process injection via the Kernel32 OpenProcess function: PROCESS_VM_READ 0x0010 and PROCESS_QUERY_LIMITED_INFORMATION 0x1000.

These permissions, collectively observed via the bitmask 0x1010, are relatively rare for lsass.exe under normal conditions.

Other Access rights can be ‘0x1438’, ‘0x143a’, ‘0x1410’, ‘0x1FFFFF’, ‘0x1438a’, ‘0x40’

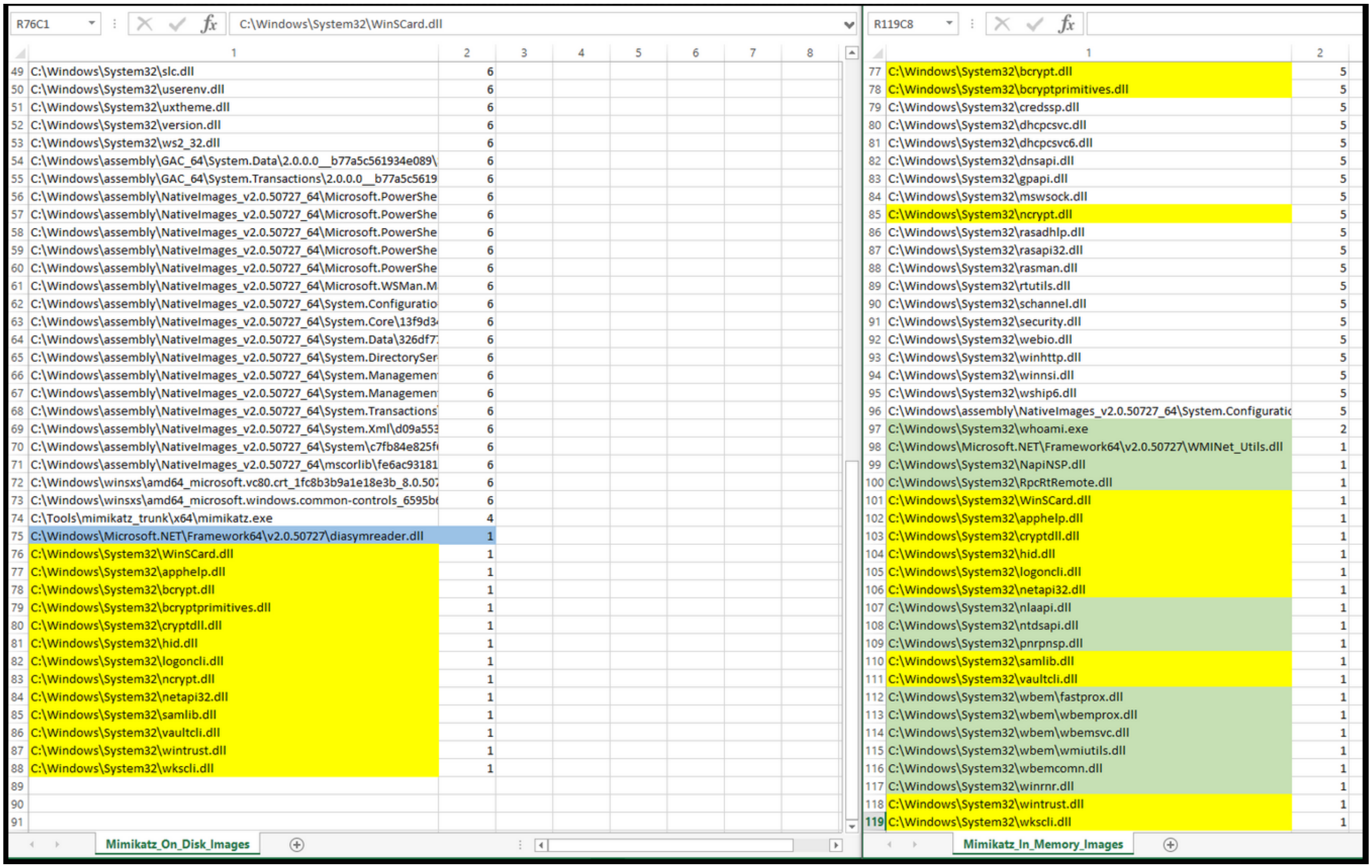


```
+ System
- EventData
  RuleName            -
  UtcTime             2020-06-17 09:39:16.380
  SourceProcessGUID   {6DD886D3-E4A9-5EE9-7D06-000000000D00}
  SourceProcessId     6292
  SourceThreadId      1728
  SourceImage         C:\Users\bwayne\Downloads\x64\mimikatz.exe
  TargetProcessGUID   {6DD886D3-1082-5ED5-0B00-000000000D00}
  TargetProcessId     568
  TargetImage         C:\Windows\system32\lsass.exe
  GrantedAccess       0x1010
  CallTrace           C:\Windows\SYSTEM32\ntdll.dll+a5324|C:\Windows\System32\KERNELBASE.dll+2940d|C:\Users\bwayne\Downloads\x64
                      \mimikatz.exe+b7b96|C:\Users\bwayne\Downloads\x64\mimikatz.exe+b7f59|C:\Users\bwayne\Downloads\x64
                      \mimikatz.exe+b7ad5|C:\Users\bwayne\Downloads\x64\mimikatz.exe+840fc|C:\Users\bwayne\Downloads\x64
                      \mimikatz.exe+83f34|C:\Users\bwayne\Downloads\x64\mimikatz.exe+83cff|C:\Users\bwayne\Downloads\x64\mimikatz.exe+be559|C:\Windows\System32
                      \KERNEL32.DLL+8364|C:\Windows\SYSTEM32\ntdll.dll+65e91
```

3. **Mimikatz detection via DLLs:**
   Resource: Sysmon, Event ID 7, Fields: ImageLoaded

For mimikatz to do its magic, there are certain DLLs [Dynamic Link Library] which are to be loaded into memory. Here, we try to look for few of the libraries/modules which are commonly associated with mimikatz.

These DLLs are commonly seen during the start process AND its execution either on disk or in memory:
WinSCard.dll, cryptdll.dll, hid.dll, samlib.dll, vaultcli.dll, wlanapi.dll, apphelp.dll, logoncli.dll, netapi32.dll, wintrust.dll, wkscli.dll



R76C1

C:\Windows\System32\WinSCard.dll

	1	2	3	4	5	6	7	8
49	C:\Windows\System32\slc.dll	6						
50	C:\Windows\System32\userenv.dll	6						
51	C:\Windows\System32\uxtheme.dll	6						
52	C:\Windows\System32\version.dll	6						
53	C:\Windows\System32\ws2_32.dll	6						
54	C:\Windows\assembly\GAC_64\System.Data\2.0.0.0__b77a5c561934e089\	6						
55	C:\Windows\assembly\GAC_64\System.Transactions\2.0.0.0__b77a5c5619	6						
56	C:\Windows\assembly\NativeImages_v2.0.50727_64\Microsoft.PowerShe	6						
57	C:\Windows\assembly\NativeImages_v2.0.50727_64\Microsoft.PowerShe	6						
58	C:\Windows\assembly\NativeImages_v2.0.50727_64\Microsoft.PowerShe	6						
59	C:\Windows\assembly\NativeImages_v2.0.50727_64\Microsoft.PowerShe	6						
60	C:\Windows\assembly\NativeImages_v2.0.50727_64\Microsoft.PowerShe	6						
61	C:\Windows\assembly\NativeImages_v2.0.50727_64\Microsoft.WSMan.M	6						
62	C:\Windows\assembly\NativeImages_v2.0.50727_64\System.Configuratio	6						
63	C:\Windows\assembly\NativeImages_v2.0.50727_64\System.Core\13f9d3	6						
64	C:\Windows\assembly\NativeImages_v2.0.50727_64\System.Data\326df7	6						
65	C:\Windows\assembly\NativeImages_v2.0.50727_64\System.DirectorySer	6						
66	C:\Windows\assembly\NativeImages_v2.0.50727_64\System.Managemen	6						
67	C:\Windows\assembly\NativeImages_v2.0.50727_64\System.Managemen	6						
68	C:\Windows\assembly\NativeImages_v2.0.50727_64\System.Transactions	6						
69	C:\Windows\assembly\NativeImages_v2.0.50727_64\System.Xml\d09a553	6						
70	C:\Windows\assembly\NativeImages_v2.0.50727_64\System\c7fb84e825f	6						
71	C:\Windows\assembly\NativeImages_v2.0.50727_64\mscorlib\fe6ac93181	6						
72	C:\Windows\winsxs\amd64_microsoft.vc80.crt_1fc8b3b9a1e18e3b_8.0.507	6						
73	C:\Windows\winsxs\amd64_microsoft.windows.common-controls_6595b	6						
74	C:\Tools\mimikatz_trunk\x64\mimikatz.exe	4						
75	C:\Windows\Microsoft.NET\Framework64\v2.0.50727\diasymreader.dll	1						
76	C:\Windows\System32\WinSCard.dll	1						
77	C:\Windows\System32\apphelp.dll	1						
78	C:\Windows\System32\bcrypt.dll	1						
79	C:\Windows\System32\bcryptprimitives.dll	1						
80	C:\Windows\System32\cryptdll.dll	1						
81	C:\Windows\System32\hid.dll	1						
82	C:\Windows\System32\logoncli.dll	1						
83	C:\Windows\System32\ncrypt.dll	1						
84	C:\Windows\System32\netapi32.dll	1						
85	C:\Windows\System32\samlib.dll	1						
86	C:\Windows\System32\vaultcli.dll	1						
87	C:\Windows\System32\wintrust.dll	1						
88	C:\Windows\System32\wkscli.dll	1						
89								
90								
91								

Mimikatz_On_Disk_Images

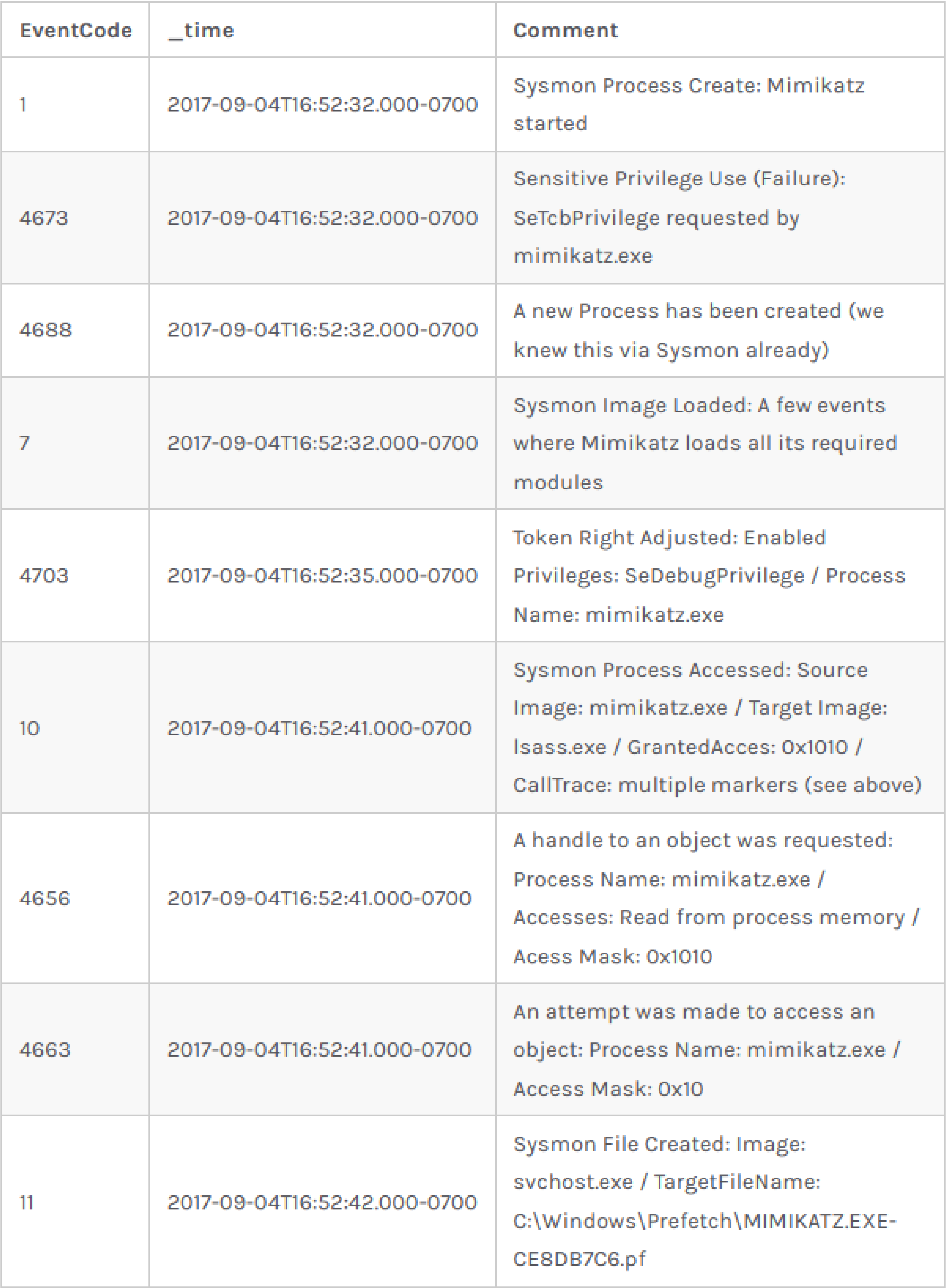
R119C8

	1	2
77	C:\Windows\System32\bcrypt.dll	5
78	C:\Windows\System32\bcryptprimitives.dll	5
79	C:\Windows\System32\credssp.dll	5
80	C:\Windows\System32\dhcpcsvc.dll	5
81	C:\Windows\System32\dhcpcsvc6.dll	5
82	C:\Windows\System32\dnsapi.dll	5
83	C:\Windows\System32\gpapi.dll	5
84	C:\Windows\System32\mswsock.dll	5
85	C:\Windows\System32\ncrypt.dll	5
86	C:\Windows\System32\rasadhlp.dll	5
87	C:\Windows\System32\rasapi32.dll	5
88	C:\Windows\System32\rasman.dll	5
89	C:\Windows\System32\rtutils.dll	5
90	C:\Windows\System32\schannel.dll	5
91	C:\Windows\System32\security.dll	5
92	C:\Windows\System32\webio.dll	5
93	C:\Windows\System32\winhttp.dll	5
94	C:\Windows\System32\winnsi.dll	5
95	C:\Windows\System32\wship6.dll	5
96	C:\Windows\assembly\NativeImages_v2.0.50727_64\System.Configuratic	5
97	C:\Windows\System32\whoami.exe	2
98	C:\Windows\Microsoft.NET\Framework64\v2.0.50727\WMINet_Utils.dll	1
99	C:\Windows\System32\NapiNSP.dll	1
100	C:\Windows\System32\RpcRtRemote.dll	1
101	C:\Windows\System32\WinSCard.dll	1
102	C:\Windows\System32\apphelp.dll	1
103	C:\Windows\System32\cryptdll.dll	1
104	C:\Windows\System32\hid.dll	1
105	C:\Windows\System32\logoncli.dll	1
106	C:\Windows\System32\netapi32.dll	1
107	C:\Windows\System32\nlaapi.dll	1
108	C:\Windows\System32\ntdsapi.dll	1
109	C:\Windows\System32\pnrpnsp.dll	1
110	C:\Windows\System32\samlib.dll	1
111	C:\Windows\System32\vaultcli.dll	1
112	C:\Windows\System32\wbem\fastprox.dll	1
113	C:\Windows\System32\wbem\wbemprox.dll	1
114	C:\Windows\System32\wbem\wbemsvc.dll	1
115	C:\Windows\System32\wbem\wmiutils.dll	1
116	C:\Windows\System32\wbemcomn.dll	1
117	C:\Windows\System32\winrnr.dll	1
118	C:\Windows\System32\wintrust.dll	1
119	C:\Windows\System32\wkscli.dll	1

Mimikatz_In_Memory_Images

{This alone cannot be used for detection, need to be combined with others to reduce false positives}

4. **Mimikatz detection with Windows Logs {Chain Model} [Running mimikatz on Disk]**
   Resource: Windows Security
   Events: 4673 (A privileged service was called), 4688 (A new process has been created), 4703 (Token Right Adjusted), 4656 (A handle to an Object was requested), 4663 (An attempt was made to access an object)



EventCode	_time	Comment
1	2017-09-04T16:52:32.000-0700	Sysmon Process Create: Mimikatz started
4673	2017-09-04T16:52:32.000-0700	Sensitive Privilege Use (Failure): SeTcbPrivilege requested by mimikatz.exe
4688	2017-09-04T16:52:32.000-0700	A new Process has been created (we knew this via Sysmon already)
7	2017-09-04T16:52:32.000-0700	Sysmon Image Loaded: A few events where Mimikatz loads all its required modules
4703	2017-09-04T16:52:35.000-0700	Token Right Adjusted: Enabled Privileges: SeDebugPrivilege / Process Name: mimikatz.exe
10	2017-09-04T16:52:41.000-0700	Sysmon Process Accessed: Source Image: mimikatz.exe / Target Image: lsass.exe / GrantedAcces: 0x1010 / CallTrace: multiple markers (see above)
4656	2017-09-04T16:52:41.000-0700	A handle to an object was requested: Process Name: mimikatz.exe / Accesses: Read from process memory / Acess Mask: 0x1010
4663	2017-09-04T16:52:41.000-0700	An attempt was made to access an object: Process Name: mimikatz.exe / Access Mask: 0x10
11	2017-09-04T16:52:42.000-0700	Sysmon File Created: Image: svchost.exe / TargetFileName: C:\Windows\Prefetch\MIMIKATZ.EXE-CE8DB7C6.pf

5. **Mimikatz detection with Windows Logs {Chain Model} [Running mimikatz from memory]**
   Resource: Windows Security
   Events: 4703 (Token Right Adjusted), 4656 (A handle to an Object was requested), 4663 (An attempt was made to access an object), 4673 (A privileged service was called), 4690 (An attempt was made to duplicate a handle to an object)

Time	Comment
09/06/2017 11:55:33 PM	So we find that the only process that resembles the “CallTrace” parameter observed for the standalone Mimikatz is wininit.exe.
09/06/2017 11:55:33 PM	Pipe Created event where lsass.exe creates PipeName: \lsass
09/06/2017 11:55:33 PM	We have a “Pipe Connected” event where “C:\Windows\system32 \svchost.exe” uses “PipeName: \lsass”
09/06/2017 11:56:44 PM	When powershell is started to host the malicious script it needs to start as “admin” which creates an EventCode 4703 (Token Right Adjusted) with the “SeDebugPrivilege”. This can be used in a transactional search disregarding the name of the process and searching for the process ID instead across different events.
09/07/2017 12:00:25 AM	EventCode 4656 (A handle to an object was requested) - Process Name is “powershell”; Access Mask is 0x143A; Accesses are: “Create new thread in process; Perform virtual memory operation; Read from process memory; Write to process memory; Query process information”
09/07/2017 12:00:25 AM	EventCode 4663 (An attempt was made to access an object) - Process Name is “powershell”; Access Mask is 0x10; Object Name is “\Device \HarddiskVolume2\Windows\System32\lsass.exe”
09/07/2017 12:00:25 AM	EventCode 4673 (A privileged service was called) - Powershell fails to obtain SeTcbPrivilege; a behaviour we already observed with the standalone Mimikatz
09/07/2017 12:00:25 AM	EventCode 4690 (An attempt was made to duplicate a handle to an object) - Source Process ID matches that of Powershell and the Target Process ID is System (0x4)
09/07/2017 12:00:35 AM	EventCode 4673 (Sensitive Privilege Use) - lsass seems to invoke LsaRegisterLogonProcess() Service from the NT Local Security Authority Server. This happens 10s after Invoke-Mimikatz.

6. Invoke-mimikatz detection via PowerShell

Resource: Windows Security (PowerShell logs)

“System.Reflection.AssemblyName”
“System.Reflection.Emit.AssemblyBuilderAccess “
“System.Runtime.InteropServices.MarshalAsAttribute”
“TOKEN_PRIVILEGES”
“SE_PRIVILEGE_ENABLED“

Detecting other offensive PowerShell tools:

“GetDelegateForFunctionPointer”
“System.Reflection.AssemblyName“
“System.Reflection.Emit.AssemblyBuilderAccess“
“System.Management.Automation.WindowsErrorReporting”
“MiniDumpWriteDump”
“TOKEN_IMPERSONATE”
“TOKEN_DUPLICATE”
“TOKEN_ADJUST_PRIVILEGES”
“TOKEN_PRIVILEGES”

7. Mimikatz detection with command line parameters

Look for the following commands.

Resource: Windows Security (PowerShell logs)

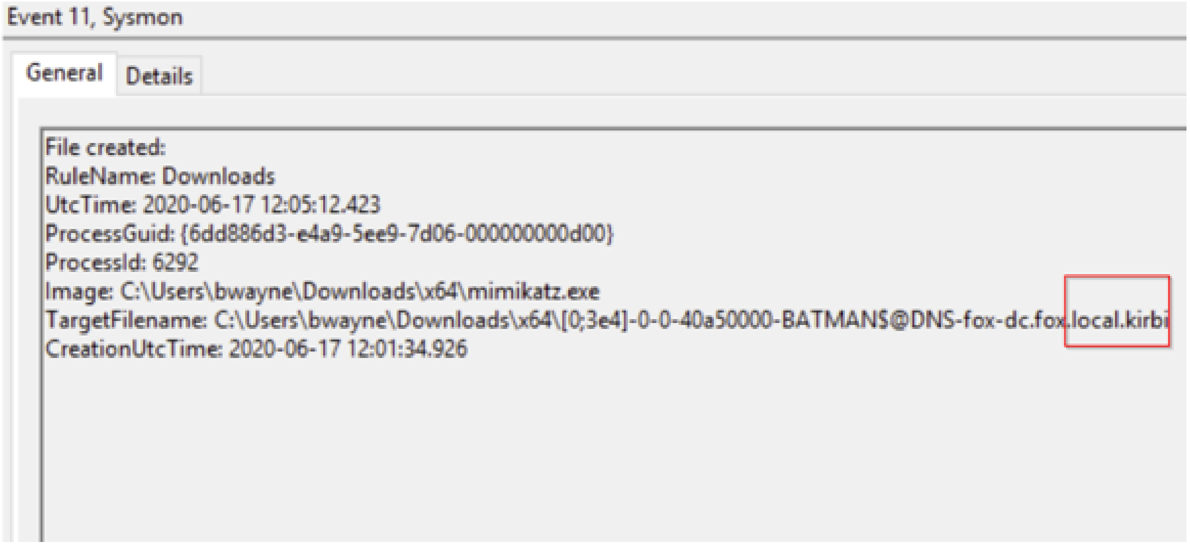
CRYPTO::Certificates – list/export certificates
KERBEROS::Golden – create golden/silver/trust tickets
KERBEROS::List – List all user tickets (TGT and TGS) in user memory. No special privileges required since it only displays the current user’s tickets.Similar to functionality of “klist.”
KERBEROS::PTT – pass the ticket. Typically used to inject a stolen or forged Kerberos ticket (golden/silver/trust).
LSADUMP::DCSync – ask a DC to synchronize an object (get password data for account). No need to run code on DC.
LSADUMP::LSA – Ask LSA Server to retrieve SAM/AD enterprise (normal, patch on the fly or inject). Use to dump all Active Directory domain credentials from a Domain Controller or lsass.dmp dump file. Also used to get specific account credential such as krbtgt with the parameter /name: “/name:krbtgt”
LSADUMP::SAM – get the SysKey to decrypt SAM entries (from registry or hive). The SAM option connects to the local Security Account Manager (SAM) database and dumps credentials for local accounts. This is used to dump all local credentials on a Windows computer.
LSADUMP::Trust – Ask LSA Server to retrieve Trust Auth Information (normal or patch on the fly). Dumps trust keys (passwords) for all associated trusts (domain/forest).
MISC::AddSid – Add to SIDHistory to user account. The first value is the target account, and the second value is the account/group name(s) (or SID). Moved to SID:modify as of May 6th, 2016.
MISC::MemSSP – Inject a malicious Windows SSP to log locally authenticated credentials.
MISC::Skeleton – Inject Skeleton Key into LSASS process on Domain Controller. This enables all user authentication to the Skeleton Key patched DC to use a “master password” (aka Skeleton Keys) as well as their usual password.
PRIVILEGE::Debug – get debug rights (this or Local System rights is required for many Mimikatz commands).

SEKURLSA::Ekeys – list Kerberos encryption keys
SEKURLSA::Kerberos – List Kerberos credentials for all authenticated users (including services and computer account)
SEKURLSA::Krbtgt – get Domain Kerberos service account (KRBTGT)password data
SEKURLSA::LogonPasswords – lists all available provider credentials. This usually shows recently logged on user and computer credentials.
SEKURLSA::Pth – Pass- theHash and Over-Pass-the-Hash
SEKURLSA::Tickets – Lists all available Kerberos tickets for all recently authenticated users, including services running under the context of a user account and the local computer's AD computer account. Unlike kerberos::list, sekurlsa uses memory reading and is not subject to key export restrictions. sekurlsa can access tickets of other sessions (users).
TOKEN::List – list all tokens of the system
TOKEN::Elevate – impersonate a token. Used to elevate permissions to SYSTEM (default) or find a domain admin token on the box
TOKEN::Elevate /domainadmin – impersonate a token with Domain Admin credentials.

8. Detecting stolen or forged Kerberos tickets

This can lead to Kerberoasting. "Kerberoasting," occurs when Kerberos tickets are extracted from memory and the password of an account is cracked, allowing the adversary to pivot within the environment via a newly hijacked account. Monitor for any files ending with ".kirbi"

Resource: Sysmon, Fields: TargetFilename



```
File created:
RuleName: Downloads
UtcTime: 2020-06-17 12:05:12.423
ProcessGuid: {6dd886d3-e4a9-5ee9-7d06-000000000d00}
ProcessId: 6292
Image: C:\Users\bwayne\Downloads\x64\mimikatz.exe
TargetFilename: C:\Users\bwayne\Downloads\x64\[0;3e4]-0-0-40a50000-BATMAN$@DNS-fox-dc.fox.local.kirbi
CreationUtcTime: 2020-06-17 12:01:34.926
```

9. The Following link contains YARA rules published by Benjamin DELPY: https://github.com/gentilkiwi/mimikatz/blob/master/kiwi_passwords.yar

Resources:

https://adsecurity.org/?page_id=1821
https://neil-fox.github.io/Mimikatz-usage-&-detection/
https://redcanary.com/threat-detection-report/threats/mimikatz/
https://www.eideon.com/2017-09-09-THL01-Mimikatz/#mimikatz-as-a-standalone-executable
https://cyberwardog.blogspot.com/2017/03/chronicles-of-threat-hunter-hunting-for.html