

Lab 05 - Cybersecurity 3

Analyse Forensic Android

1. Mise en place

Après lecture du Lab05, je fais moi-même un **Dockerfile** pour construire et adapter les besoins de l'outil aux dernières versions. Pas de Volatility 2, mais 3. Je dois juste rajouter une compatibilité avec les Service Pack des versions antérieures.

Ensuite, je réfléchis à la correspondance des commandes d'une version à l'autre :

Action	Volatility 2	Volatility 3
Informations image	<code>volatility -f zeus.vmem imageinfo</code>	<code>vol -f zeus.vmem windows.info</code>
Liste des processus	<code>volatility -f zeus.vmem --profile=... pslist</code>	<code>vol -f zeus.vmem windows.pslist</code>
Arborescence processus	<code>volatility -f zeus.vmem --profile=... pstree</code>	<code>vol -f zeus.vmem windows.pstree</code>
Détection masquage	<code>volatility -f zeus.vmem --profile=... psscan</code>	<code>vol -f zeus.vmem windows.psscan</code>
Analyse réseau	<code>volatility -f zeus.vmem --profile=... connections</code>	<code>vol -f zeus.vmem windows.netscan</code>
Détection d'injection	<code>volatility -f zeus.vmem --profile=... malfind</code>	<code>vol -f zeus.vmem windows.malfind</code>
Analyse DLL unlinked	<code>volatility -f zeus.vmem --profile=... ldrmodules</code>	<code>vol -f zeus.vmem windows.ldrmodules</code>
Dump mémoire processus	<code>volatility -fzeus.vmem --profile=... memdump -p <PID></code>	<code>vol -f zeus.vmem windows.memdump --pid <PID></code>

Après le **Dockerfile**, le **docker-compose.yml** et les commandes corrigées, on peut lancer la stack.

```
D:\Users\GaWin\LiveCampus [main +4 ~0 -0 !]> docker ps
CONTAINER ID   IMAGE                COMMAND             CREATED        STATUS        PORTS        NAMES
44f3da040e31   volatility-volatility "bash"             4 hours ago   Up 34 minutes                volatility_lab
```

```
D:\Users\GaWin\LiveCampus\Semaine_12\Jour_05\volatility [main +4 ~0 -0 !]> docker exec -it volatility_lab bash
root@a7ba642c530d:/data#
```

PS : après le lab, j'ai passé la commande `for f in ./dump/*.{img,dat,dmp}; do [ -f "$f" ] && truncate -s 0 "$f"; done` pour "vider" les fichiers dans le dossier **dump** afin de montrer leurs noms et variété, mais sans garder les 181Mo de stockage qui allaient avec.

2. Analyses

A. Android avec **zeus.vmem**

a. Identification de l'image

Le processus de démarrage est assez long.

```

root@44f3da040e31:/data# vol -f zeus.vmem windows.info
Volatility 3 Framework 2.28.0
Progress: 16.93 Updating caches for 3124 files...

root@44f3da040e31:/data# vol -f zeus.vmem windows.info
Volatility 3 Framework 2.28.0
WARNING volatility3.framework.layers.vmware: No metadata file found alongside VMEM file. A VMSS or VMSN file may be required to correctly process a VMEM file. The
se should be placed in the same directory with the same file name, e.g. zeus.vmem and zeus.vmss.
Progress: 100.00 PDB scanning finished
Variable Value

Kernel Base 0x804d7000
DTB 0x39000
Symbols jar:file:/usr/local/lib/python3.10/site-packages/volatility3/symbols/windows.zip!windows/ntkrnlpa.pdb/BD8F451F3E754ED8A34B50560CEB08E3-1.json.xz
Is64Bit False
IsPAE False
layer_name 0 WindowsIntel
memory_layer 1 FileLayer
KdDebuggerDataBlock 0x80544ce0
NTBuildLab 2600.xpsp_sp2_rtm.040803-2158
CSDVersion 2
KdVersionBlock 0x80544cb8
Major/Minor 15.2600
MachineType 332
KeNumberProcessors 1
SystemTime 2010-08-15 19:17:56+00:00
NtSystemRoot C:\WINDOWS
NtProductType NtProductWinNt
NtMajorVersion 5
NtMinorVersion 1
PE MajorOperatingSystemVersion 5
PE MinorOperatingSystemVersion 1
PE Machine 332
PE TimeDateStamp Wed Aug 4 05:58:36 2004
root@44f3da040e31:/data#

```

CONCLUSION

Je me rends compte avec des sorties vides de **pslist** et **psvolumes** que ce DUMP de mémoire est taillé pour la version 2 de Volatility. Refusant de travailler avec du matériel obsolète, je préfère passer sur une version récente, quitte à la retélécharger. Je tente l'aventure avec l'image Windows.

B. Windows avec **WinDump.mem**

Je vais donner des captures d'écran des sorties CLI, mais faire ensuite des exports dans un fichier contigu pour analyse à froid.

a. Identification de l'image

```

root@44f3da040e31:/data# ls
MacDump.dmp  WinDump.mem  zeus.vmem
root@44f3da040e31:/data# vol -f WinDump.mem windows.info
Volatility 3 Framework 2.28.0
Progress: 100.00          PDB scanning finished
Variable      Value

Kernel Base    0xf8021ec0f000
DTB            0x1aa000
Symbols        jar:file:/usr/local/lib/python3.10/site-packages/volatility3/symbols/windows.zip!windows/ntkrnlmp.pdb/0C9CC659210046EC84231597DADFDB3B-1.json.xz
Is64Bit        True
IsPAE          False
layer_name     0 WindowsIntel32e
memory_layer   1 FileLayer
KdVersionBlock 0xf8021ef510a0
Major/Minor    15.15063
MachineType    34404
KeNumberProcessors 4
SystemTime     2018-06-21 06:50:50+00:00
NtSystemRoot   C:\WINDOWS
NtProductType  NtProductWinNt
NtMajorVersion 10
NtMinorVersion 0
PE MajorOperatingSystemVersion 10
PE MinorOperatingSystemVersion 0
PE Machine     34404
PE TimeDateStamp Sat Apr 28 04:13:12 2018
root@44f3da040e31:/data# █

```

b. Inspection des processus

```

root@44f3da040e31:/data# vol -f WinDump.mem windows.pslist
Volatility 3 Framework 2.28.0
Progress: 100.00          PDB scanning finished

```

PID	PPID	ImageFileName	Offset(V)	Threads	Handles	SessionId	Wow64	CreateTime	ExitTime	File output
4	0	System	0xd98df30c24c0 178	-	N/A	False	2018-06-08 00:11:39.000000 UTC	N/A	Disabled	
356	4	smss.exe	0xd98df4e56600 3	-	N/A	False	2018-06-08 00:11:40.000000 UTC	N/A	Disabled	
468	460	csrss.exe	0xd98df60ef080 12	-	0	False	2018-06-08 00:12:23.000000 UTC	N/A	Disabled	
568	460	wininit.exe	0xd98df657c080 2	-	0	False	2018-06-08 00:12:24.000000 UTC	N/A	Disabled	
716	568	services.exe	0xd98df65e0080 9	-	0	False	2018-06-08 00:12:24.000000 UTC	N/A	Disabled	
744	568	lsass.exe	0xd98df65d7080 8	-	0	False	2018-06-08 00:12:25.000000 UTC	N/A	Disabled	
844	716	svchost.exe	0xd98df665a600 2	-	0	False	2018-06-08 00:12:25.000000 UTC	N/A	Disabled	
868	716	svchost.exe	0xd98df666c480 26	-	0	False	2018-06-08 00:12:25.000000 UTC	N/A	Disabled	
920	716	svchost.exe	0xd98df6594600 12	-	0	False	2018-06-08 00:12:25.000000 UTC	N/A	Disabled	
944	568	fontdrvhost.exe	0xd98df6590600 6	-	0	False	2018-06-08 00:12:25.000000 UTC	N/A	Disabled	
72	716	svchost.exe	0xd98df658c600 6	-	0	False	2018-06-08 00:12:26.000000 UTC	N/A	Disabled	
804	716	svchost.exe	0xd98df6729600 3	-	0	False	2018-06-08 00:12:26.000000 UTC	N/A	Disabled	
1056	716	svchost.exe	0xd98df674c600 4	-	0	False	2018-06-08 00:12:26.000000 UTC	N/A	Disabled	
1140	716	svchost.exe	0xd98df6774600 6	-	0	False	2018-06-08 00:12:26.000000 UTC	N/A	Disabled	
1260	716	svchost.exe	0xd98df6781600 8	-	0	False	2018-06-08 00:12:26.000000 UTC	N/A	Disabled	
1288	716	atiesrxx.exe	0xd98df67d5080 6	-	0	False	2018-06-08 00:12:27.000000 UTC	N/A	Disabled	
1316	716	svchost.exe	0xd98df67cc600 5	-	0	False	2018-06-08 00:12:27.000000 UTC	N/A	Disabled	
1368	716	svchost.exe	0xd98df67e0600 10	-	0	False	2018-06-08 00:12:27.000000 UTC	N/A	Disabled	
1412	716	svchost.exe	0xd98df682f600 4	-	0	False	2018-06-08 00:12:27.000000 UTC	N/A	Disabled	
1420	716	svchost.exe	0xd98df681b600 6	-	0	False	2018-06-08 00:12:27.000000 UTC	N/A	Disabled	
1436	716	svchost.exe	0xd98df6838600 7	-	0	False	2018-06-08 00:12:27.000000 UTC	N/A	Disabled	
1532	716	svchost.exe	0xd98df6867600 12	-	0	False	2018-06-08 00:12:27.000000 UTC	N/A	Disabled	
1548	716	svchost.exe	0xd98df6887600 4	-	0	False	2018-06-08 00:12:27.000000 UTC	N/A	Disabled	
1568	716	svchost.exe	0xd98df6894600 24	-	0	False	2018-06-08 00:12:27.000000 UTC	N/A	Disabled	
1644	716	svchost.exe	0xd98df68bb600 5	-	0	False	2018-06-08 00:12:27.000000 UTC	N/A	Disabled	
1652	716	svchost.exe	0xd98df68c9600 6	-	0	False	2018-06-08 00:12:27.000000 UTC	N/A	Disabled	
1724	716	svchost.exe	0xd98df686a600 14	-	0	False	2018-06-08 00:12:27.000000 UTC	N/A	Disabled	
1824	716	svchost.exe	0xd98df691a600 8	-	0	False	2018-06-08 00:12:28.000000 UTC	N/A	Disabled	
1872	716	svchost.exe	0xd98df6949600 6	-	0	False	2018-06-08 00:12:28.000000 UTC	N/A	Disabled	
1940	716	svchost.exe	0xd98df696a600 11	-	0	False	2018-06-08 00:12:29.000000 UTC	N/A	Disabled	
1948	716	svchost.exe	0xd98df698e600 6	-	0	False	2018-06-08 00:12:29.000000 UTC	N/A	Disabled	
2036	716	svchost.exe	0xd98df69b2600 5	-	0	False	2018-06-08 00:12:29.000000 UTC	N/A	Disabled	
2068	716	svchost.exe	0xd98df6a2c380 15	-	0	False	2018-06-08 00:12:29.000000 UTC	N/A	Disabled	
2260	716	svchost.exe	0xd98df6a94600 11	-	0	False	2018-06-08 00:12:31.000000 UTC	N/A	Disabled	
2304	716	svchost.exe	0xd98df6aa1600 7	-	0	False	2018-06-08 00:12:32.000000 UTC	N/A	Disabled	
2364	716	spoolsv.exe	0xd98df6abe600 14	-	0	False	2018-06-08 00:12:32.000000 UTC	N/A	Disabled	
2424	716	svchost.exe	0xd98df6b08600 14	-	0	False	2018-06-08 00:12:33.000000 UTC	N/A	Disabled	
2504	716	svchost.exe	0xd98df6b53600 6	-	0	False	2018-06-08 00:12:33.000000 UTC	N/A	Disabled	
2656	716	svchost.exe	0xd98df6b61600 5	-	0	False	2018-06-08 00:12:34.000000 UTC	N/A	Disabled	
2664	716	svchost.exe	0xd98df6b78600 11	-	0	False	2018-06-08 00:12:34.000000 UTC	N/A	Disabled	

REFLEXION alors que je suis sur l'analyse Windows

Plutôt que m'embêter à faire les lignes à la main, je songe : pourquoi ne pas automatiser tout ça avec un simple script `.sh` ? Je le fais et je le mets à disposition de l'archive finale de rendu.

J'ai fais une double boucle "for: do" un peu sommaire, améliorée à l'IA ensuite pour résilience et "bonne bouille" là où mes talents en shell sont très austères :

```

root@44f3da040e31:/data# ls
MacDump.dmp WinDump.mem auto_scan.sh zeus.vmem
root@44f3da040e31:/data# chmod +x auto_scan.sh
root@44f3da040e31:/data# ./auto_scan.sh

=====
Automated Volatility 3 Analysis Engine
=====

[*] Analyse du dump : MacDump.dmp
Chemin : ./MacDump.dmp
-----
[+] Exécution de windows.info ... [ÉCHEC] -> Log: ./results/MacDump.dmp_windows.info.log
[+] Exécution de windows.pslist ... [ÉCHEC] -> Log: ./results/MacDump.dmp_windows.pslist.log
[+] Exécution de windows.pstree ... [ÉCHEC] -> Log: ./results/MacDump.dmp_windows.pstree.log
[+] Exécution de windows.psscan ... [ÉCHEC] -> Log: ./results/MacDump.dmp_windows.psscan.log
[+] Exécution de windows.netscan ... [ÉCHEC] -> Log: ./results/MacDump.dmp_windows.netscan.log
[+] Exécution de windows.malfind ... [ÉCHEC] -> Log: ./results/MacDump.dmp_windows.malfind.log
[+] Exécution de windows.ldrmodules ... [ÉCHEC] -> Log: ./results/MacDump.dmp_windows.ldrmodules.log

[*] Analyse du dump : WinDump.mem
Chemin : ./WinDump.mem
-----
[+] Exécution de windows.info ... [OK] -> Log: ./results/WinDump.mem_windows.info.log
[+] Exécution de windows.pslist ... [OK] -> Log: ./results/WinDump.mem_windows.pslist.log
[+] Exécution de windows.pstree ... [OK] -> Log: ./results/WinDump.mem_windows.pstree.log
[+] Exécution de windows.psscan ... [OK] -> Log: ./results/WinDump.mem_windows.psscan.log
[+] Exécution de windows.netscan ... [OK] -> Log: ./results/WinDump.mem_windows.netscan.log
[+] Exécution de windows.malfind ... [OK] -> Log: ./results/WinDump.mem_windows.malfind.log
[+] Exécution de windows.ldrmodules ... [OK] -> Log: ./results/WinDump.mem_windows.ldrmodules.log

[*] Analyse du dump : zeus.vmem
Chemin : ./zeus.vmem
-----
[+] Exécution de windows.info ... [OK] -> Log: ./results/zeus.vmem_windows.info.log
[+] Exécution de windows.pslist ... [OK] -> Log: ./results/zeus.vmem_windows.pslist.log
[+] Exécution de windows.pstree ... [OK] -> Log: ./results/zeus.vmem_windows.pstree.log
[+] Exécution de windows.psscan ... [OK] -> Log: ./results/zeus.vmem_windows.psscan.log
[+] Exécution de windows.netscan ... [ÉCHEC] -> Log: ./results/zeus.vmem_windows.netscan.log
[+] Exécution de windows.malfind ... [OK] -> Log: ./results/zeus.vmem_windows.malfind.log
[+] Exécution de windows.ldrmodules ... [OK] -> Log: ./results/zeus.vmem_windows.ldrmodules.log

=====
[V] Analyse terminée ! Logs enregistrés dans './results/'
=====
root@44f3da040e31:/data# █

```

b. Inspection des processus²

Incohérences parent/enfant

Les services "normaux" ont des parents "normaux". `svchost.exe` a toujours `services.exe`. `lsass.exe` et `services.exe` ont `wininit.exe`. Un `cmd.exe` ou `powershell.exe` ont une commande utilisateur à l'origine aussi.

On peut y aller à la comparaison regex :

```
SERVICES_PID=$(awk '$3 == "services.exe" {print $1}' results/*_windows.pslist.log)

awk -v spid="$SERVICES_PID" '$3 == "svchost.exe" && $2 != spid {print "SUSPECT (PPID anormale):", $0}' results/*_windows.pslist.log
```

```
root@44f3da040e31:/data# SERVICES_PID=$(awk '$3 == "services.exe" {print $1}' results/*_windows.pslist.log)
awk -v spid="$SERVICES_PID" '$3 == "svchost.exe" && $2 != spid {print "SUSPECT (PPID anormale):", $0}' results/*_windows.pslist.log
root@44f3da040e31:/data# echo $SERVICES_PID
716
root@44f3da040e31:/data# awk -v spid="$SERVICES_PID" '$3 == "svchost.exe" && $2 != spid {print "SUSPECT (PPID anormale):", $0}' results/*_windows.pslist.log
root@44f3da040e31:/data#
```

Hmmm... bah ça a l'air d'être bon pour **svchost.exe**.

```
awk '$3 ~ /^(lsass\.exe|services\.exe|wininit\.exe)$/ && !($2 ~ /^(4|356|568)$/) {print "ANOMALIE PARENT:", $0}' results/*_windows.pslist.log
```

```
root@44f3da040e31:/data# awk '$3 ~ /^(lsass\.exe|services\.exe|wininit\.exe)$/ && !($2 ~ /^(4|356|568)$/) {print "ANOMALIE PARENT:", $0}' results/*_windows.pslist.log
ANOMALIE PARENT: 568 460 wininit.exe 0xd98df657c080 2 - 0 False 2018-06-08 00:12:24.000000 UTC N/A Disabled
root@44f3da040e31:/data#
```

AH-AH !

Processus masqué

Ce qui est dans **psscan** mais pas dans **pslist** est un processus masqué. Un rootkit potentiel. Ou un anti-triche de FPS triple A.

On va donc extraire :

1. Les PID de **pslist**
2. Les PID de **psscan**
3. Comparer les deux

```
awk 'NR>3 && $1 ~ /^[0-9]+$/ {print $1}' results/*_windows.pslist.log | sort -u > /tmp/pids_pslist.txt

awk 'NR>3 && $1 ~ /^[0-9]+$/ {print $1}' results/*_windows.psscan.log | sort -u > /tmp/pids_psscan.txt

comm -23 /tmp/pids_psscan.txt /tmp/pids_pslist.txt
```

```

root@44f3da040e31:/data# awk 'NR>3 && $1 ~ /^[0-9]+$/ {print $1}' results/*_windows.pslist.log | sort -u > /tmp/pids_pslist.txt
root@44f3da040e31:/data# awk 'NR>3 && $1 ~ /^[0-9]+$/ {print $1}' results/*_windows.psscans.log | sort -u > /tmp/pids_psscans.txt
root@44f3da040e31:/data# comm -23 /tmp/pids_psscans.txt /tmp/pids_pslist.txt
10164
1028
1084
1088
1148
124
1432
1668
1732
1788
1944
1968
1980
216
2372
2628
2908
3828
432
4364
452
5180
5352
544
5440
5696
584
608
6120
6220
632
676
688
7268
8292
856
888
936
root@44f3da040e31:/data#

```

"Ca fait beaucoup, là, non ?"

Alors, certains sont des faux positifs : si un **ExitTime** a une date, le processus s'est fini avec le DUMP de la RAM. Genre :

```

root@44f3da040e31:/data# cat ./results/WinDump.mem_windows.psscans.log | grep "10164"
10164  9756  mscorsvw.exe  0xd98df8d99480  0  -  0  False  2018-06-21 06:48:41.000000 UTC  2018-06-21 06:48:42.000000 UTC  Disabled

```

Maintenant, si ce champ est vide, il est actif MAIS masqué. Et l'infection par Rootkit / DKOM se confirme :

```

root@44f3da040e31:/data# cat ./results/WinDump.mem_windows.psscans.log | grep "936"
1168  5380  SearchProtocol  0xd98dfa893600  6  -  0  False  2018-06-21 06:47:33.000000 UTC  N/A  Disabled

```

"SearchProtocol" pue donc le caca.

Faux noms

scvhost.exe, **expl0rer.exe** etc. C'est plus chaud à automatiser, parce qu'il faudrait un comparateur d'approximations basé sur un dictionnaire. Je suis sûr qu'il y a des bibliothèques Python qui pourraient aider, mais on peut se contenter d'un semi-manuel pour voir si un nom légitime est lancé depuis un endroit illogique :

```

# Lancement d'un processus Win32 hors de System32
grep -Ei "svchost\.exe|lsass\.exe|csrss\.exe" results/*_windows.pstree.log | grep

```

```
-v -i "System32"

# Fautes de frappe
grep -Ei "scvhost|svchost32|svchosts|lsasss|csrsss|explorerr|payloadd"
results/*_windows.pslist.log
```

```
root@44f3da040e31:/data# grep -Ei "svchost\.exe|lsass\.exe|csrss\.exe" results/*_windows.pstree.log | grep -v -i "System32"
root@44f3da040e31:/data# grep -Ei "scvhost|svchost32|svchosts|lsasss|csrsss|explorerr|payloadd" results/*_windows.pslist.log
root@44f3da040e31:/data#
```

A priori ça va.

c. Analyse réseau

Lister les connexions réseau est plutôt facile :

```
grep -i "ESTABLISHED" results/*_windows.netscan.log
```

```
root@44f3da040e31:/data# grep -i "ESTABLISHED" results/*_windows.netscan.log
results/WinDump.mem_windows.netscan.log:0xd98df33c2cc0 TCPv4 192.168.2.169 50781 198.142.175.14 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df3982840 TCPv4 192.168.2.169 50800 198.142.175.14 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df3b324d0 TCPv4 192.168.2.169 50780 198.142.175.46 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df3b3d5e0 TCPv4 192.168.2.169 50873 104.95.127.140 443 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df3b8bcc0 TCPv4 192.168.2.169 50865 198.142.175.30 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df3cbaab0 TCPv4 192.168.2.169 50820 67.27.57.254 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df3cf8010 TCPv4 192.168.2.169 50429 67.27.63.254 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df3d8acc0 TCPv4 192.168.2.169 50815 67.27.59.254 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df3de7cc0 TCPv4 192.168.2.169 50812 198.142.175.14 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df3efb640 TCPv4 192.168.2.169 50777 198.142.175.14 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df4187cc0 TCPv4 192.168.2.169 50792 198.142.175.14 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df42bd180 TCPv4 192.168.2.169 50867 198.142.175.30 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df42d8660 TCPv4 192.168.2.169 50793 198.142.175.46 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df43e2010 TCPv4 192.168.2.169 50689 198.142.175.14 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df46472e0 TCPv4 192.168.2.169 50779 198.142.175.14 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df46ccb50 TCPv4 192.168.2.169 50802 67.27.59.254 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df4890400 TCPv4 192.168.2.169 50826 67.27.57.254 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df48f6820 TCPv4 192.168.2.169 50810 198.142.175.46 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df4ff4750 TCPv4 192.168.2.169 50684 198.142.175.14 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df5127a70 TCPv4 192.168.2.169 50654 67.27.59.254 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df5405cc0 TCPv4 192.168.2.169 50799 198.142.175.46 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df5588100 TCPv4 192.168.2.169 50817 198.142.175.46 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df55b2cc0 TCPv4 192.168.2.169 50804 198.142.175.46 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df5856560 TCPv4 192.168.2.169 50675 8.253.176.103 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df5d4a500 TCPv4 192.168.2.169 49923 52.230.7.59 443 ESTABLISHED 2720 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df5eb8830 TCPv4 192.168.2.169 50872 40.69.218.62 443 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df63a64f0 TCPv4 192.168.2.169 50796 198.142.175.14 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df67269d0 TCPv4 192.168.2.169 50798 67.27.59.254 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df71e37f0 TCPv4 192.168.2.169 50806 198.142.175.46 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df7431cc0 TCPv4 192.168.2.169 50827 67.27.59.254 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df74d41a0 TCPv4 192.168.2.169 50778 198.142.175.46 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df7579900 TCPv4 192.168.2.169 50824 198.142.175.14 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df75e7350 TCPv4 192.168.2.169 50795 198.142.175.46 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df790b870 TCPv4 192.168.2.169 50811 198.142.175.46 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df7918660 TCPv4 192.168.2.169 50797 198.142.175.46 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df7b35740 TCPv4 192.168.2.169 50816 198.142.175.46 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df7c36460 TCPv4 192.168.2.169 50822 198.142.175.46 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df7d38640 TCPv4 192.168.2.169 50825 67.27.59.254 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df7fc0010 TCPv4 192.168.2.169 50427 67.27.63.254 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df8294b30 TCPv4 192.168.2.169 50776 198.142.175.46 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df834d010 TCPv4 192.168.2.169 50819 67.27.59.254 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df84fbaf0 TCPv4 192.168.2.169 50774 198.142.175.46 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df8c21cc0 TCPv4 192.168.2.169 50447 8.253.176.103 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df8ed6cc0 TCPv4 192.168.2.169 50823 198.142.175.46 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df8f278f0 TCPv4 192.168.2.169 50662 67.27.59.254 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df9063630 TCPv4 192.168.2.169 50809 67.27.59.254 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df9091cc0 TCPv4 192.168.2.169 50821 67.27.59.254 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df933bcc0 TCPv4 192.168.2.169 50808 67.27.57.254 80 ESTABLISHED 5764 svchost.exe
results/WinDump.mem_windows.netscan.log:0xd98df93ac010 TCPv4 192.168.2.169 50834 192.168.2.2 445 ESTABLISHED 4 System N/A
```

Pour extraire les adresses publiques, simples : on élimine les adresses privées des "ForeignAddr", avec un regex `awk`.

```
awk '
$7 ~ /ESTABLISHED/ &&
$5 !~ /^(127\.0\.0\.1|0\.0\.0\.0|::1|::|\\*|-)$/ &&
$5 !~ /^(10\.|192\.168\.|172\.(1[6-9]|2[0-9]|3[0-1])\.)/ {
    print $0
}' results/*_windows.netscan.log
```

```
root@44f3da040e31:/data# awk '
$7 ~ /ESTABLISHED/ &&
$5 !~ /^(127\.0\.0\.1|0\.0\.0\.0|::1|::|\\*|-)$/ &&
$5 !~ /^(10\.|192\.168\.|172\.(1[6-9]|2[0-9]|3[0-1])\.)/ {
    print $0
}' results/*_windows.netscan.log
0xd98df33c2cc0 TCPv4 192.168.2.169 50781 198.142.175.14 80 ESTABLISHED 5764 svchost.exe N/A
0xd98df3982840 TCPv4 192.168.2.169 50800 198.142.175.14 80 ESTABLISHED 5764 svchost.exe N/A
0xd98df3b324d0 TCPv4 192.168.2.169 50780 198.142.175.46 80 ESTABLISHED 5764 svchost.exe N/A
0xd98df3b3d5e0 TCPv4 192.168.2.169 50873 104.95.127.140 443 ESTABLISHED 5764 svchost.exe N/A
0xd98df3b8bcc0 TCPv4 192.168.2.169 50865 198.142.175.30 80 ESTABLISHED 5764 svchost.exe N/A
0xd98df3cbaab0 TCPv4 192.168.2.169 50820 67.27.57.254 80 ESTABLISHED 5764 svchost.exe N/A
0xd98df3cf8010 TCPv4 192.168.2.169 50429 67.27.63.254 80 ESTABLISHED 5764 svchost.exe N/A
0xd98df3d8acc0 TCPv4 192.168.2.169 50815 67.27.59.254 80 ESTABLISHED 5764 svchost.exe N/A
0xd98df3de7cc0 TCPv4 192.168.2.169 50812 198.142.175.14 80 ESTABLISHED 5764 svchost.exe N/A
0xd98df3efb640 TCPv4 192.168.2.169 50777 198.142.175.14 80 ESTABLISHED 5764 svchost.exe N/A
0xd98df4187cc0 TCPv4 192.168.2.169 50792 198.142.175.14 80 ESTABLISHED 5764 svchost.exe N/A
0xd98df42bd180 TCPv4 192.168.2.169 50867 198.142.175.30 80 ESTABLISHED 5764 svchost.exe N/A
0xd98df42d8660 TCPv4 192.168.2.169 50793 198.142.175.46 80 ESTABLISHED 5764 svchost.exe N/A
0xd98df43e2010 TCPv4 192.168.2.169 50689 198.142.175.14 80 ESTABLISHED 5764 svchost.exe N/A
0xd98df46472e0 TCPv4 192.168.2.169 50779 198.142.175.14 80 ESTABLISHED 5764 svchost.exe N/A
0xd98df46ccb50 TCPv4 192.168.2.169 50802 67.27.59.254 80 ESTABLISHED 5764 svchost.exe N/A
0xd98df4890400 TCPv4 192.168.2.169 50826 67.27.57.254 80 ESTABLISHED 5764 svchost.exe N/A
0xd98df48f6820 TCPv4 192.168.2.169 50810 198.142.175.46 80 ESTABLISHED 5764 svchost.exe N/A
0xd98df4ff4750 TCPv4 192.168.2.169 50684 198.142.175.14 80 ESTABLISHED 5764 svchost.exe N/A
0xd98df5127a70 TCPv4 192.168.2.169 50654 67.27.59.254 80 ESTABLISHED 5764 svchost.exe N/A
0xd98df5405cc0 TCPv4 192.168.2.169 50799 198.142.175.46 80 ESTABLISHED 5764 svchost.exe N/A
0xd98df5588100 TCPv4 192.168.2.169 50817 198.142.175.46 80 ESTABLISHED 5764 svchost.exe N/A
0xd98df55b2cc0 TCPv4 192.168.2.169 50804 198.142.175.46 80 ESTABLISHED 5764 svchost.exe N/A
0xd98df5856560 TCPv4 192.168.2.169 50675 8.253.176.103 80 ESTABLISHED 5764 svchost.exe N/A
0xd98df5d4a500 TCPv4 192.168.2.169 49923 52.230.7.59 443 ESTABLISHED 2720 svchost.exe N/A
0xd98df5eb8830 TCPv4 192.168.2.169 50872 40.69.218.62 443 ESTABLISHED 5764 svchost.exe N/A
0xd98df63a64f0 TCPv4 192.168.2.169 50796 198.142.175.14 80 ESTABLISHED 5764 svchost.exe N/A
0xd98df67269d0 TCPv4 192.168.2.169 50798 67.27.59.254 80 ESTABLISHED 5764 svchost.exe N/A
0xd98df71e37f0 TCPv4 192.168.2.169 50806 198.142.175.46 80 ESTABLISHED 5764 svchost.exe N/A
0xd98df7431cc0 TCPv4 192.168.2.169 50827 67.27.59.254 80 ESTABLISHED 5764 svchost.exe N/A
0xd98df74d41a0 TCPv4 192.168.2.169 50778 198.142.175.46 80 ESTABLISHED 5764 svchost.exe N/A
0xd98df7579900 TCPv4 192.168.2.169 50824 198.142.175.14 80 ESTABLISHED 5764 svchost.exe N/A
0xd98df75e7350 TCPv4 192.168.2.169 50795 198.142.175.46 80 ESTABLISHED 5764 svchost.exe N/A
0xd98df790b870 TCPv4 192.168.2.169 50811 198.142.175.46 80 ESTABLISHED 5764 svchost.exe N/A
0xd98df7918660 TCPv4 192.168.2.169 50797 198.142.175.46 80 ESTABLISHED 5764 svchost.exe N/A
0xd98df7b35740 TCPv4 192.168.2.169 50816 198.142.175.46 80 ESTABLISHED 5764 svchost.exe N/A
0xd98df7c36460 TCPv4 192.168.2.169 50822 198.142.175.46 80 ESTABLISHED 5764 svchost.exe N/A
0xd98df7d38640 TCPv4 192.168.2.169 50825 67.27.59.254 80 ESTABLISHED 5764 svchost.exe N/A
0xd98df7fc0010 TCPv4 192.168.2.169 50427 67.27.63.254 80 ESTABLISHED 5764 svchost.exe N/A
0xd98df8294b30 TCPv4 192.168.2.169 50776 198.142.175.46 80 ESTABLISHED 5764 svchost.exe N/A
0xd98df834d010 TCPv4 192.168.2.169 50819 67.27.59.254 80 ESTABLISHED 5764 svchost.exe N/A
0xd98df84fbaf0 TCPv4 192.168.2.169 50774 198.142.175.46 80 ESTABLISHED 5764 svchost.exe N/A
0xd98df8c21cc0 TCPv4 192.168.2.169 50447 8.253.176.103 80 ESTABLISHED 5764 svchost.exe N/A
0xd98df8ed6cc0 TCPv4 192.168.2.169 50823 198.142.175.46 80 ESTABLISHED 5764 svchost.exe N/A
```

Allons plus loin en extrayant les PID et les états :

```
awk '
NR > 2 && $5 !~ /^(127\.0\.0\.1|0\.0\.0\.0|::1|::|\\*|-|10\.|192\.168\.|172\.(1[6-9]|2[0-9]|3[0-1])\.)/ {
    printf "PID: %-6s | Processus: %-15s | IP Distant: %-15s | Port: %-5s | Etat:
```



```
%-5s | DLL: %s\n", $1, $2, $3, $4, $5, $6  
}' results/*_windows.ldrmodules.log
```

Il y en a BEAUCOUP (plus de 850). J'ai sorti un fichier "WinDump_injected_DLL.logs" pour l'avoir en entier. C'est même fou d'avoir autant d'infections.

A ce stade, on comprend le nombre d'éléments de **malfind**.

On va pouvoir générer un dump d'un programme suspect. Le PID 1168 me semble prometteur :

PID: 1168 | Processus: SearchProtocol | InLoad: 0x7ff9a14e0000 | InMem: False | InInit: False | DLL: False

```
mkdir -p dump  
  
vol -f WinDump.mem windows.memdump --pid 1168 --output-dir dump/
```

... et j'obtiens une erreur qui me prend pas mal la tête. J'interroge une IA pour parer au plus pressé, et elle me répond ceci :

windows.memdump n'existe plus dans Volatility 3. Le framework a divisé cette fonctionnalité en plusieurs plugins spécialisés selon ce que l'on souhaite extraire.

Elle me donne alors trois options que je vais tester une à une :

```
# Option A. Dumper toutes les pages mémoire du processus (Équivalent de memdump)  
vol -o dump/ -f WinDump.mem windows.vadinfo --dump --pid 1168  
  
# Option B. Extraire l'exécutable et les fichiers en mémoire (Recommandé)  
vol -o dump/ -f WinDump.mem windows.dumpfiles --pid 1168
```

En étudiant son fonctionnement :

- Il indexait du contenu pour le moteur de recherche Windows
- Il chargeait des bibliothèques DLL natives
- Il était appelé par **SearchIndexer.exe** au PID 5380

En bref... je me suis complètement gourré, il est bien légitime. L'échec est total, mon humiliation est totale, et j'en tire les conclusions nécessaires, à savoir me retirer de la vie politique.

e. Analyse manuelle

Je revise ma situation précédente avec le PID 6620 : **software_reporter_tool.exe** lancé depuis **C:\Users\passmark\AppData\Local** et qui contacte des adresses étranges comme **https://clients2.google.com/cr/report** explicitement.

Je fais donc un extract des données mémoire :

```
strings dump/*6620* | grep -Ei "http://|https://" | sort -u
```

```
strings -e l dump/*6620* | grep -Ei "http://|https://|SwReporter|chrome" | sort -u
```

```
root@44f3da040e31:/data# strings dump/*6620* | grep -Ei "http://|https://" | sort -u
targetNamespace="http://schemas.microsoft.com/Search/2013/ApplicationContent"
xmlns:ac="http://schemas.microsoft.com/Search/2013/ApplicationContent"
<dpiAware xmlns="http://schemas.microsoft.com/SMI/2005/WindowsSettings">true</dpiAware>
style="display:none;"><cli><a href="http://www. type='text/javascript'>
"c:\users\passmark\appdata\local\google\chrome\user data\swreporter\25.141.202\software_reporter_tool.exe" --crash-handler "--database=c:\users\passmark\ap
pdata\local\Google\Software Reporter Tool" --url=https://clients2.google.com/cr/report --annotation=plat=Win32 --annotation=prod=ChromeFoil --annotation=ve
r=25.141.202 --initial-client-data=0x224,0x21c,0x228,0x220,0x22c,0x7ff7a0e19e10,0x7ff7a0e19e28,0x7ff7a0e19e40
--url=https://clients2.google.com/cr/report
<html xmlns="http://W3C//DTD HTML 4.01 TentsByTagName('script')input type="hidden" nam<script type="text/javas" style="display:none;">document.getElementBy
Id=(document.createElement(' type='text/javascript'input type="text" name="d.getElementsByTagName(snical" href="http://www.C//DTD HTML 4.01 Transit<style t
ype="text/css">
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
https://
https://D
https://clients2.google.com/cr/report
https://sb-ssl.google.com/safebrowsing/clientreport/chrome-sw-reporter?key=AIzaSyB0ti4mM-6x9WdnZIjIeyEU210pBXqWBgw
root@44f3da040e31:/data#
```

```
root@44f3da040e31:/data# strings -e l dump/*6620* | grep -Ei "http://|https://|SwReporter|chrome" | sort -u
"c:\users\passmark\appdata\local\google\chrome\user data\swreporter\25.141.202\software_reporter_tool.exe" --crash-handler "--database=c:\users\passmark\ap
pdata\local\Google\Software Reporter Tool" --url=https://clients2.google.com/cr/report --annotation=plat=Win32 --annotation=prod=ChromeFoil --annotation=ve
r=25.141.202 --initial-client-data=0x224,0x21c,0x228,0x220,0x22c,0x7ff7a0e19e10,0x7ff7a0e19e28,0x7ff7a0e19e40
--annotation=prod=ChromeFoil
--url=https://clients2.google.com/cr/report
<?xml version="1.0" encoding="utf-16"?> <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" %s elementFormDefault="qu
alified" > <xsd:element name="%s"> <xsd:complexType> <xsd:sequence> <xsd:any minOccurs="0" maxOccurs="u
nbounded" processContents="lax"/> </xsd:sequence> </xsd:complexType> </xsd:element> </xsd:schema>
C:\Program Files (x86)\Google\Chrome\Application\67.0.3396.87\
C:\Users\passmark\AppData\Local\Google\Chrome\User Data\SwReporter\25.141.202\
C:\Users\passmark\AppData\Local\Google\Chrome\User Data\SwReporter\25.141.202\software_reporter_tool.exe
CHROME_CRASHPAD_PIPE_NAME=\
CHROME_CRASHPAD_PIPE_NAME=\\.\pipe\crashpad_1596_YXFKEDKLBjYHYXNJ
CHROME_RE
CHROME_RESTART=Google Chrome|Whoa! Google Chrome has crashed. Relaunch now?|LEFT_TO_RIGHT
Chrome Cleanup Tool
Chrome Cleanup Tool Test
Chrome.lnk
Google Chrome.lnk
Path=C:\Program Files (x86)\Google\Chrome\Application;C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\WINDOWS\System32\WindowsPowerShell\v1.0\;C
:\Program Files (x86)\ATI Technologies\ATI.ACE\Core-Static;C:\Users\passmark\AppData\Local\Microsoft\WindowsApps;
Software\Google\Chrome Cleanup Tool Test\TempTestKeys
T=Google Chrome|Whoa
\Device\HarddiskVolume2\Users\passmark\AppData\Local\Google\Chrome\User Data\SwReporter\25.141.202\software_reporter_tool.exe
al\google\chrome\user data\swreporter\25.141.202\software_reporter_tool.exe
c:\users\passmark\appdata\local\google\chrome\user data\swreporter\25.141.202;C:\WINDOWS\SYSTEM32;C:\WINDOWS\system;C:\WINDOWS.;C:\Program Files (x86)\Goo
gle\Chrome\Application;C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\WINDOWS\System32\WindowsPowerShell\v1.0\;C:\Program Files (x86)\ATI Techn
ologies\ATI.ACE\Core-Static;C:\Users\passmark\AppData\Local\Microsoft\WindowsApps;
c:\users\passmark\appdata\local\google\chrome\user data\swreporter\25.141.202\WININET.dll
c:\users\passmark\appdata\local\google\chrome\user data\swreporter\25.141.202\WINMMBASE.dll
c:\users\passmark\appdata\local\google\chrome\user data\swreporter\25.141.202\software_reporter_tool.exe
data\local\google\chrome\user data\swre
ext-ms-oncore-appchromeapi-11-1-0
ext-ms-oncore-shellchromeapi-11-1-2
http://
http://ctldl.windowsupdate.com/msdownload/update/v3/static/trustedr/en
http://schemas.microsoft.com/SMI/2005/WindowsSettings
http://schemas.microsoft.com/SMI/2011/WindowsSettings
http://schemas.microsoft.com/SMI/2013/WindowsSettings
http://schemas.microsoft.com/SMI/2014/WindowsSettings
http://schemas.microsoft.com/SMI/2016/WindowsSettings
http://schemas.microsoft.com/SMI/2017/WindowsSettings
http://www.w3.org/XML/1998/namespace
https://
https://clients2.google.com/cr/report
ogle Chrome
ogle\chrome\user data\swreporter\25.141.202\CRYPTBASE.DLL
techsnab\chrome_launcher\tasks.dll
x8C:\Program Files (x86)\Google\Chrome\Application\67.0.3396.87\
root@44f3da040e31:/data#
```

Bon... encore une fois, je pense avoir fait chou blanc, ça semble légitimement utilisé par Google (enfin... si on suppose que Google fasse quoi que ce soit de légitime dans la vie...), même si la simulation de comportement et son origine semblent étranges.

C. macOS avec MacDump.dmp

Hélas, le dump n'est pas du tout compatible avec la version Volatility 3.