



Quick Write Up: CryptoLocker v1


FCSC 2020 - Yxène

Énoncé.txt

Description

Un de nos admins nous a appelé en urgence suite à un CryptoLocker qui s'est lancé sur un serveur ultra-sensible, juste après avoir appliqué une mise à jour fournie par notre prestataire informatique. Ce *malware* vise spécifiquement un fichier pouvant faire perdre des millions d'euros à notre entreprise : il est très important de le retrouver ! L'administrateur nous a dit que pour éviter que le logiciel ne se propage, il a mis en pause le serveur virtualisé et a récupéré sa mémoire vive dès qu'il a détecté l'attaque. Vous êtes notre seul espoir.

Fichiers

 cryptolocker-
v1.tar.xz
121.43 MiB – 81207e9...

Auteur



haxom



Mémoire Vive / RAM (*Random Access Memory*)

La mémoire vive (RAM) est une mémoire **volatile** utilisée par un ordinateur pour stocker **temporairement** les données et instructions nécessaires à l'exécution des programmes en cours.

On y retrouve des informations sur :

- les connexions réseaux
- les clés de registre
- les mots de passe
- les processus en cours d'exécution
- ...



Volatility

- Volatility est le principal outil d'analyse de dump de mémoire vive
- Fonctionnement par plugin (avec possibilité de créer les siens)
- 2 versions existent





Challenge

```
# Analyse du type de fichier
```

```
file cryptolocker-v1.dmp
```

```
cryptolocker-v1.dmp: MS Windows 32bit crash dump, PAE, full dump,  
262030 pages
```

Informations sur le Dump

```
python3 volatility3/vol.py -f cryptolocker-v1.dmp windows.info
```

```
Is64Bit      False
IsPAE        True
layer_name    0 WindowsIntelPAE
memory_layer  1 WindowsCrashDump32Layer
base_layer    2 FileLayer
KdDebuggerDataBlock 0x8273ac78
NTBuildLab    7601.23915.x86fre.win7sp1_ldr.17
CSDVersion    1
KdVersionBlock 0x8273ac50
Major/Minor   15.7601
MachineType   332
KeNumberProcessors 1
SystemTime    2020-04-13 18:39:35
NtSystemRoot  C:\Windows
NtProductType NtProductWinNt
NtMajorVersion 6
NtMinorVersion 1
...
```

Analyse des processus

```
python volatility3/vol.py -f cryptolocker-v1.dmp windows.pslist
```

PID	PPID	ImageFileName	Offset(V)	Threads	Handles	SessionId	Wow64	CreateTime	ExitTimeFile	output
4	0	System	0x83d38ad0 89	526	N/A	False	2020-04-13 18:36:59.000000	N/A	Disabled	
248	4	smss.exe	0x844f8d28	2	29	N/A	False	2020-04-13 18:36:59.000000	N/A	Disabled
340	332	csrss.exe	0x84bc64d0	8	562	0	False	2020-04-13 18:37:04.000000	N/A	Disabled
380	372	csrss.exe	0x84bfa030	10	219	1	False	2020-04-13 18:37:05.000000	N/A	Disabled
412	332	wininit.exe	0x84bfe318	3	77	0	False	2020-04-13 18:37:05.000000	N/A	Disabled
424	372	winlogon.exe	0x84f0bd28	4	109	1	False	2020-04-13 18:37:05.000000	N/A	Disabled
484	412	services.exe	0x84f37030	9	232	0	False	2020-04-13 18:37:06.000000	N/A	Disabled
492	412	lsass.exe	0x84f3ba30	7	620	0	False	2020-04-13 18:37:06.000000	N/A	Disabled
500	412	lsm.exe	0x84f3da70 10	198	0	False	2020-04-13 18:37:06.000000	N/A	Disabled	
592	484	svchost.exe	0x84f6bbb8	11	363	0	False	2020-04-13 18:37:09.000000	N/A	Disabled
664	484	svchost.exe	0x84f76d28	7	284	0	False	2020-04-13 18:37:09.000000	N/A	Disabled
764	484	svchost.exe	0x84fa6030	21	438	0	False	2020-04-13 18:37:10.000000	N/A	Disabled

Conseil: Sauvegardez les résultats de chaque plugin dans un fichier

Analyse des processus

```
mkdir update_v0.5
```

```
python volatility3/vol.py -f cryptolocker-v1.dmp -o update_v0.5/  
windows.dumpfiles --pid 3388
```

Cache	FileObject	FileName	Result
DataSectionObject		0x84f13898	key.txt file.0x84f13898.0x854fbc98.DataSectionObject.key.txt.dat
SharedCacheMap	0x84f13898	key.txt	file.0x84f13898.0x84f13700.SharedCacheMap.key.txt.vacb
ImageSectionObject		0x84f39da0	sspicli.dll file.0x84f39da0.0x84f3e2d8.ImageSectionObject.sspicli.dll.img
DataSectionObject		0x84f66b60	update_v0.5.exe file.0x84f66b60.0x85279860.DataSectionObject.update_v0.5.exe.dat
ImageSectionObject		0x84f66b60	update_v0.5.exe file.0x84f66b60.0x8548c008.ImageSectionObject.update_v0.5.exe.img
ImageSectionObject		0x8548e038	sfc.dll file.0x8548e038.0x853c1e20.ImageSectionObject.sfc.dll.img
ImageSectionObject		0x844bfff80	msacm32.dll file.0x844bfff80.0x84f10940.ImageSectionObject.msacm32.dll.img

Analyse des processus

```
mkdir update_v0.5
```

```
python volatility3/vol.py -f cryptolocker-v1.dmp -o update_v0.5/  
windows.dumpfiles --pid 3388
```

Cache	FileObject	FileName	Result
DataSectionObject	0x84f13898	key.txt	file.0x84f13898.0x854fbc98.DataSectionObject.key.txt.dat
...			
DataSectionObject	0x84f66b60	update_v0.5.exe	file.0x84f66b60.0x85279860.DataSectionObject.update_v0.5.exe.dat
...			



Analyse des processus

```
cat file.0x84f13898.0x854fbc98.DataSectionObject.key.txt.dat
```

```
0ba883a22afb84506c8d8fd9e42a5ce4e8eb1cc87c315a28dd
```

```
file file.0x84f66b60.0x85279860.DataSectionObject.update_v0.5.exe.dat
```

```
PE32 executable (console) Intel 80386, for MS Windows
```

Démo sur Ghidra !



À la recherche du flag chiffré

```
python volatility3/vol.py -f cryptolocker-v1.dmp windows.filescan
```

0x3e948	\Windows\System32\LogFiles\Scm\90805a89-edee-47cc-baee-591d011d90d2	128
0x3eada00	\Windows\System32\pcasvc.dll	128
0x3e9adb8	\\$PrepareToShrinkFile	128
0x3e9adb70	\Windows\System32\	
0x3e9add18	\Windows\System32\	
0x3e9a9dec8	\Windows\System32\	
0x3e9a9df80	\Windows\System32\	
0x3e9a9f3b8	\Windows\Registration\R000000000000006.clb	128
0x3e9a9f08c0	\Windows\System32\security.dll	128
0x3e9a9f1520	\Windows\System32\drivers\intelppm.sys	128
0x3e9a9f25a8	\Windows\System32\drivers\monitor.sys	128

4490 lignes 🙄



À la recherche du flag chiffré

```
python volatility3/vol.py -f cryptolocker-v1.dmp windows.filescan
```

Offset	Name	Size
...		
0x3ed139f0	\Users\IEUser\Desktop\flag.txt.enc	128
...		

À la recherche du flag chiffré

```
mkdir flagCipher
```

```
python volatility3/vol.py -f cryptolocker-v1.dmp -o flagCipher/  
windows.dumpfiles --physaddr 0x3ed139f0
```

Cache	FileObject	FileName	Result
DataSectionObject	0x3ed139f0	flag.txt.encfile.0x3ed139f0.0x855651e0.DataSectionObject	flag.txt.enc.dat

```
xxd file.0x3ed139f0.0x855651e0.DataSectionObject.flag.txt.enc.dat
```

```
00000000: 277b 6b70 1a01 0055 0507 5d0c 5355 0555  '{kp...U..].SU.U  
00000010: 095d 595e 065c 0402 0654 0751 0055 015e  .]Y^.\...T.Q.U.^  
00000020: 5557 525b 575c 5154 5007 5107 0b5e 5551  UWR[W\QTP.Q..^UQ  
00000030: 5556 0259 5a07 0502 5751 5201 0f03 5702  UV.YZ...WQR...W.  
00000040: 0601 5a50 0f1b 6e00 0000 0000 0000 0000  ..ZP..n.....
```



Déchiffrement

```
with open("key.txt", "rb") as f:
    key = f.read().strip(b'\x00')

print(f"Clé: {key}\n")

with open("flag.txt.enc", "rb") as f:
    cipher = f.read().strip(b'\x00')

print(f"Flag chiffré: {cipher}\n")

flag = b""
for i in range(len(cipher)):
    flag += (cipher[i] ^ key[(i+2)%len(key)]).to_bytes(1, 'big')

print(f"Flag déchiffré: {flag}")
```


Questions

