

Securité des SI

Volatility

Auteur: HARAKI Youness

Volatility

Est un outil open source d'extraction mémoire, pour la recherche et l'analyse de malware(memory forensics) afin de répondre aux incidents des sécurité. Volatility est programmé en Python et prend en charge Microsoft Windows, Mac OS X et Linux



Lien : <https://www.volatilityfoundation.org/releases>

Release Downloads

Volatility releases are the result of significant in-depth research into OS internals, applications, and user activities. Releases represent a milestone in not only our team's progress, but also in the development and forensics capabilities as a whole. While releases may seem few and far between, we strive to release our new features before calling it stable.

Volatility 2

Volatility 3

Volatility 2.6 (*Windows 10 / Server 2016*)

This release improves support for Windows 10 and adds support for Windows Server 2016, Mac OS Sierra 10.12, and Linux with KASLR kernels. A lot of bug fixes went into this release as well as performance enhancements (especially related to page table parsing and virtual address space scanning). See below for a more detailed list of the changes in this version.

Released: December 2016

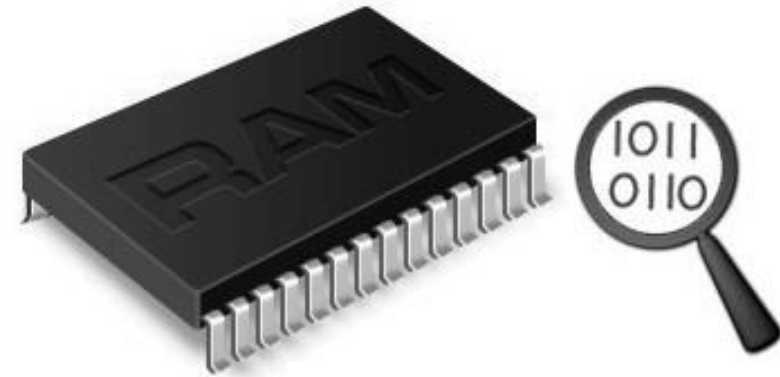
- [Download the Volatility 2.6 Windows Standalone Executable \(x64\)](#)
- [Download the Volatility 2.6 Mac OS X Standalone Executables \(x64\)](#)
- [Download the Volatility 2.6 Linux Standalone Executables \(x64\)](#)
- [Download the Volatility 2.6 Source Code \(.zip\)](#)
- [Download the Integrity Hashes](#)
- [View the README](#)
- [View the CREDITS](#)

[READ MORE >](#)

Auteur: HARAKI YOUNESS

En utilisant l'outil **dumpit** (par exemple), on génère un vidage de mémoire physique (memory dump) des machines Windows.

Lien: <https://github.com/thimbleweed/All-In-USB/tree/master/utilities/Dumplt>



Volatilityfoundation fournit différents échantillons de mémoire publiquement disponibles à des fins de test.

Lien: <https://github.com/volatilityfoundation/volatility/wiki/Memory-Samples>

This is a list of publicly available memory samples for testing purposes.

Description	OS
Art of Memory Forensics Images	Assorted Windows, Linux, and Mac
Mac OSX 10.8.3 x64	Mac Mountain Lion 10.8.3 x64
Jackcr's forensic challenge	Windows XP x86 and Windows 2003 SP0 x86 (4 images)
GrrCon forensic challenge ISO (also see PDF questions)	Windows XP x86
Malware Cookbook DVD	Black Energy, CoreFlood, Laqma, Prolaco, Sality, Silent Banker, Tigger, Zeus, etc
Malware - Cridex	Windows XP SP2 x86
Malware - Shylock	Windows XP SP3 x86
Malware - R2D2 (pw: infected)	Windows XP SP2 x86
Windows 7 x64	Windows 7 SP1 x64
NIST (5 samples)	Windows XP SP2, 2003 SP0, and Vista Beta 2 (all x86)

Pages 31

Home

Getting Started

- [FAQ](#)
- [Installation](#)
- [Linux](#)
- [Mac](#)
- [Android](#)
- [Basic Usage](#)
- [2.6 Win Profiles](#)
- [Encrypted KDBG](#)
- [Pyinstaller Builds](#)
- [Unified Output](#)

Command References

- [Windows Core](#)
- [Windows GUI](#)
- [Windows Malware](#)
- [Linux](#)
- [Mac OSX](#)

Development

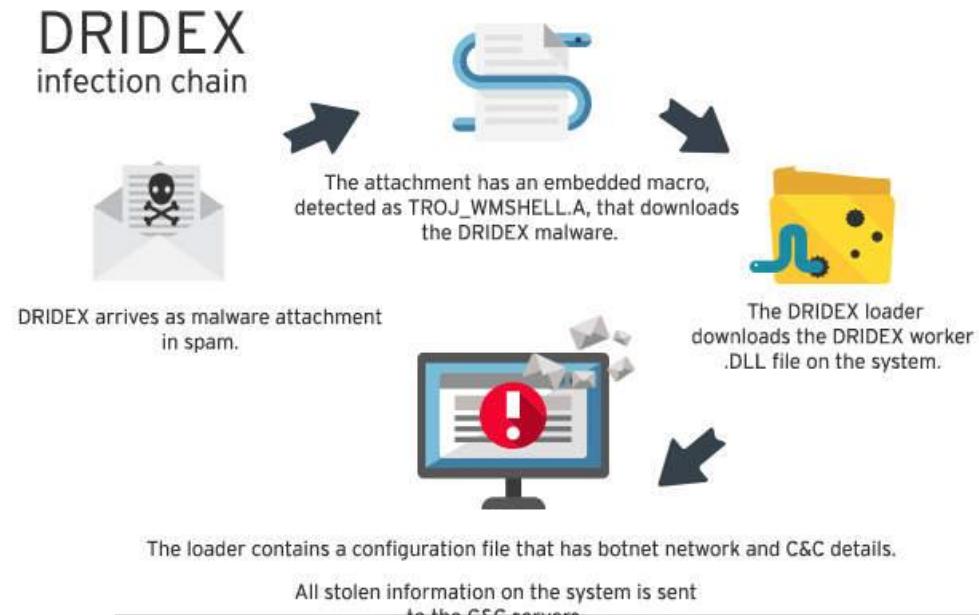
- [Windows Registry](#)
- [Address Spaces](#)
- [Style Guide](#)

Miscellaneous

Etude de cas 1









Dridex

Également connu sous le nom de Bugat, est une forme de malware spécialisé dans le vol d'informations d'identification bancaires via un système utilisant des macros de Microsoft Word.



Auteur: HARAKI YOUNESS

Nous devrions avoir **Volatility**, **Dumpit** et **cridex** memory image sous la même répertoire.

Name	Date modified	Type	Size
 AUTHORS.txt	12/27/2016 4:44 PM	Text Document	1 KB
 CREDITS.txt	12/27/2016 4:52 PM	Text Document	4 KB
 LEGAL.txt	7/7/2016 4:16 AM	Text Document	1 KB
 LICENSE.txt	7/7/2016 4:16 AM	Text Document	15 KB
 README.txt	12/24/2016 3:13 PM	Text Document	32 KB
 volatility_2.6_win64_standalone.exe	12/27/2016 5:02 PM	Application	15,424 KB
 DumpIt.exe	1/8/2022 5:15 PM	Application	203 KB
 cridex.vmem	8/2/2012 5:23 AM	VMEM File	524,288 KB

volatility.exe –help: accès aux pages de documentation de volatility (Options & plugins supportés)

```
Options:
-h, --help            list all available options and their default values.
                        Default values may be set in the configuration file
                        (/etc/volatilityrc)
--conf-file=.volatilityrc
                        User based configuration file
-d, --debug            Debug volatility
--plugins=PLUGINS      Additional plugin directories to use (semi-colon
                        separated)
--info                Print information about all registered objects
--cache-directory=C:\Users\youne/.cache/volatility
                        Directory where cache files are stored
--cache                Use caching
--tz=TZ                Sets the (Olson) timezone for displaying timestamps
                        using pytz (if installed) or tzset
-f FILENAME, --filename=FILENAME
                        Filename to use when opening an image
--profile=WinXPSP2x86
                        Name of the profile to load (use --info to see a list
                        of supported profiles)
-l LOCATION, --location=LOCATION
                        A URN location from which to load an address space
-w, --write            Enable write support
--dtb=DTB              DTB Address
--output=text          Output in this format (support is module specific, see
                        the Module Output Options below)
--output-file=OUTPUT_FILE
                        Write output in this file
-v, --verbose          Verbose information
--shift=SHIFT          Mac KASLR shift address
-g KDBG, --kdbg=KDBG   Specify a KDBG virtual address (Note: for 64-bit
                        Windows 8 and above this is the address of
                        KdCopyDataBlock)
--force                Force utilization of suspect profile
--cookie=COOKIE        Specify the address of nt!ObHeaderCookie (valid for
                        Windows 10 only)
-k KPCR, --kpcr=KPCR   Specify a specific KPCR address

Supported Plugin Commands:
amcache                Print AmCache information
apihooks               Detect API hooks in process and kernel memory
atoms                 Print session and window station atom tables
atomscan               Pool scanner for atom tables
auditpol               Prints out the Audit Policies from HKLM\SECURITY\Policy\PolAdtEv
```


volatility.exe -f cridex.vmem imageinfo: Le profil d'image du dump est requis pour faire des analyses sur volatility.

1- Profil d'image : WinXPSP2x86

```
C:\Users\youn\OneDrive\Documents\cours m2 intense\Security_course\volatil
tandalone>volatility.exe -f cridex.vmem imageinfo
Volatility Foundation Volatility Framework 2.6
INFO      : volatility.debug      : Determining profile based on KDBG search..
          Suggested Profile(s) : WinXPSP2x86, WinXPSP3x86 (Instantiated wi

          AS Layer1 : IA32PagedMemoryPae (Kernel AS)
          AS Layer2 : FileAddressSpace (C:\Users\youn\OneDrive
s m2 intense\Security_course\volatility_2.6_win64_standalone\cridex.vmem)
          PAE type : PAE
          DTB : 0x2fe000L
          KDBG : 0x80545ae0L
          Number of Processors : 1
          Image Type (Service Pack) : 3
          KPCR for CPU 0 : 0xffdff000L
          KUSER_SHARED_DATA : 0xffdf0000L
          Image date and time : 2012-07-22 02:45:08 UTC+0000
          Image local date and time : 2012-07-21 22:45:08 -0400
```

volatility.exe -f cridex.vmem --profile=WinXPSP2x86 pslist / pstree: En utilisant les informations du profil d'image, nous pouvons trouver des informations sur le vidage de la mémoire. « Observer les processus qui ont été exécutés »

PID : est l'ID du processus. PPID, est le PID du processus parent (le processus qui a engendré le processus PID actuel).

Volatility Foundation Volatility Framework 2.6											
Offset(V)	Name	PID	PPID	Thds	Hnds	Sess	Wow64	Start			
0x823c89c8	System	4	0	53	240	-----	0				
0x822f1020	smss.exe	368	4	3	19	-----	0	2012-07-22	02:42:31	UTC+0000	
0x822a0598	csrss.exe	584	368	9	326	0	0	2012-07-22	02:42:32	UTC+0000	
0x82298700	winlogon.exe	608	368	23	519	0	0	2012-07-22	02:42:32	UTC+0000	
0x81e2ab28	services.exe	652	608	16	243	0	0	2012-07-22	02:42:32	UTC+0000	
0x81e2a3b8	lsass.exe	664	608	24	330	0	0	2012-07-22	02:42:32	UTC+0000	
0x82311360	svchost.exe	824	652	20	194	0	0	2012-07-22	02:42:33	UTC+0000	
0x81e29ab8	svchost.exe	908	652	9	226	0	0	2012-07-22	02:42:33	UTC+0000	
0x823001d0	svchost.exe	1004	652	64	1118	0	0	2012-07-22	02:42:33	UTC+0000	
0x821dfda0	svchost.exe	1056	652	5	60	0	0	2012-07-22	02:42:33	UTC+0000	
0x82295650	svchost.exe	1220	652	15	197	0	0	2012-07-22	02:42:35	UTC+0000	
0x821dea70	explorer.exe	1484	1464	17	415	0	0	2012-07-22	02:42:36	UTC+0000	
0x81eb17b8	spoolsv.exe	1512	652	14	113	0	0	2012-07-22	02:42:36	UTC+0000	
0x81e7bda0	reader_sl.exe	1640	1484	5	39	0	0	2012-07-22	02:42:36	UTC+0000	
0x820e8da0	alg.exe	788	652	7	104	0	0	2012-07-22	02:43:01	UTC+0000	
0x821fcda0	wuauclt.exe	1136	1004	8	173	0	0	2012-07-22	02:43:46	UTC+0000	
0x8205bda0	wuauclt.exe	1588	1004	5	132	0	0	2012-07-22	02:44:01	UTC+0000	

2- Processus suspects : Le processus reader_sl.exe crée par explorer.exe

Volatility Foundation Volatility Framework 2.6											
Offset(V)	Name	PID	PPID	Thds	Hnds	Sess	Wow64	Start			
0x823c89c8	System	4	0	53	240	-----	0				
0x822f1020	smss.exe	368	4	3	19	-----	0	2012-07-22	02:42:31	UTC+0000	
0x822a0598	csrss.exe	584	368	9	326	0	0	2012-07-22	02:42:32	UTC+0000	
0x82298700	winlogon.exe	608	368	23	519	0	0	2012-07-22	02:42:32	UTC+0000	
0x81e2ab28	services.exe	652	608	16	243	0	0	2012-07-22	02:42:32	UTC+0000	
0x81e2a3b8	lsass.exe	664	608	24	330	0	0	2012-07-22	02:42:32	UTC+0000	
0x82311360	svchost.exe	824	652	20	194	0	0	2012-07-22	02:42:33	UTC+0000	
0x81e29ab8	svchost.exe	908	652	9	226	0	0	2012-07-22	02:42:33	UTC+0000	
0x823001d0	svchost.exe	1004	652	64	1118	0	0	2012-07-22	02:42:33	UTC+0000	
0x821dfda0	svchost.exe	1056	652	5	60	0	0	2012-07-22	02:42:33	UTC+0000	
0x82295650	svchost.exe	1220	652	15	197	0	0	2012-07-22	02:42:35	UTC+0000	
0x821dea70	explorer.exe	1484	1464	17	415	0	0	2012-07-22	02:42:36	UTC+0000	
0x81eb17b8	spoolsv.exe	1512	652	14	113	0	0	2012-07-22	02:42:36	UTC+0000	
0x81e7bda0	reader_sl.exe	1640	1484	5	39	0	0	2012-07-22	02:42:36	UTC+0000	
0x820e8da0	alg.exe	788	652	7	104	0	0	2012-07-22	02:43:01	UTC+0000	
0x821fcd00	wuauclt.exe	1136	1004	8	173	0	0	2012-07-22	02:43:46	UTC+0000	
0x8205bda0	wuauclt.exe	1588	1004	5	132	0	0	2012-07-22	02:44:01	UTC+0000	

volatility.exe -f cridex.vmem --profile=WinXPSP2x86 psxview

Nous vérifions si aucun autre processus caché n'a été exécuté
(Comportement typique de certains malwares)

→ Aucun autre processus n'est caché.

```
Volatility Foundation Volatility Framework 2.6
Offset(P)  Name                               PID  pslist  psscan  thrddproc  pspcid  csrss  session  deskthrd  ExitTime
-----
0x02498700 winlogon.exe                        608  True    True    True    True    True    True    True    True
0x02511360 svchost.exe                        824  True    True    True    True    True    True    True    True
0x022e8da0 alg.exe                          788  True    True    True    True    True    True    True    True
0x020b17b8 spoolsv.exe                       1512 True    True    True    True    True    True    True    True
0x0202ab28 services.exe                   652  True    True    True    True    True    True    True    True
0x02495650 svchost.exe                       1220 True    True    True    True    True    True    True    True
0x0207bda0 reader_sl.exe                 1640 True    True    True    True    True    True    True    True
0x025001d0 svchost.exe                      1004 True    True    True    True    True    True    True    True
0x02029ab8 svchost.exe                      908  True    True    True    True    True    True    True    True
0x023fcda0 wuauclt.exe                      1136 True    True    True    True    True    True    True    True
0x0225bda0 wuauclt.exe                      1588 True    True    True    True    True    True    True    True
0x0202a3b8 lsass.exe                       664  True    True    True    True    True    True    True    True
0x023dea70 explorer.exe                   1484 True    True    True    True    True    True    True    True
0x023dfda0 svchost.exe                      1056 True    True    True    True    True    True    True    True
0x024f1020 smss.exe                        368  True    True    True    True    False False False
0x025c89c8 System                          4    True    True    True    True    False False False
0x024a0598 csrss.exe                     584  True    True    True    True    False True    True
```

volatility.exe -f cridex.vmem --profile=WinXPSP2x86 connsnscan Nous vérifions toutes les connections qui ont été établies à partir de notre machine locale.

→ Il existe deux types d'activité réseau, créées par le PID de explorer.exe

3- Connections suspectées :

0x02087620	172.16.112.128:1038	41.168.5.140:8080	1484
0x023a8008	172.16.112.128:1037	125.19.103.198:8080	1484

```
Volatility Foundation Volatility Framework 2.6
Offset(P)  Local Address          Remote Address          Pid
-----
0x02087620 172.16.112.128:1038     41.168.5.140:8080      1484
0x023a8008 172.16.112.128:1037     125.19.103.198:8080    1484
```

volatility.exe -f cridex.vmem --profile=WinXPSP2x86 cmdline :

Une fois que nous avons récupéré les informations de connexion, les informations de commande. Nous pouvons enquêter sur les dernières lignes de commande qui ont été exécutées en mémoire.

Rappel

1- Profil d'image : WinXPSP2x86

2- Processus suspects : Le processus reader_sl.exe (PID: 1640) crée par explorer.exe (PID: 1684)

3- Connections suspects :

0x02087620	172.16.112.128:1038	41.168.5.140:8080	1484
0x023a8008	172.16.112.128:1037	125.19.103.198:8080	1484

L'explorer.exe est un processus normal créé par le système Windows.

Cependant, Adobe Reader est suspect, car il se connecte à un réseau extérieur.










4- Exécutable suspect :

"C:\Program Files\Adobe\Reader 9.0\Reader\Reader_sl.exe"

```
explorer.exe pid: 1484
Command line : C:\WINDOWS\Explorer.EXE
*****
spoolsv.exe pid: 1512
Command line : C:\WINDOWS\system32\spoolsv.exe
*****
reader_sl.exe pid: 1640
Command line : "C:\Program Files\Adobe\Reader 9.0\Reader\Reader_sl.exe"
*****
```

volatility.exe -f cridex.vmem --profile=WinXPSP2x86 procdump -p 1640 --dump-dir . :

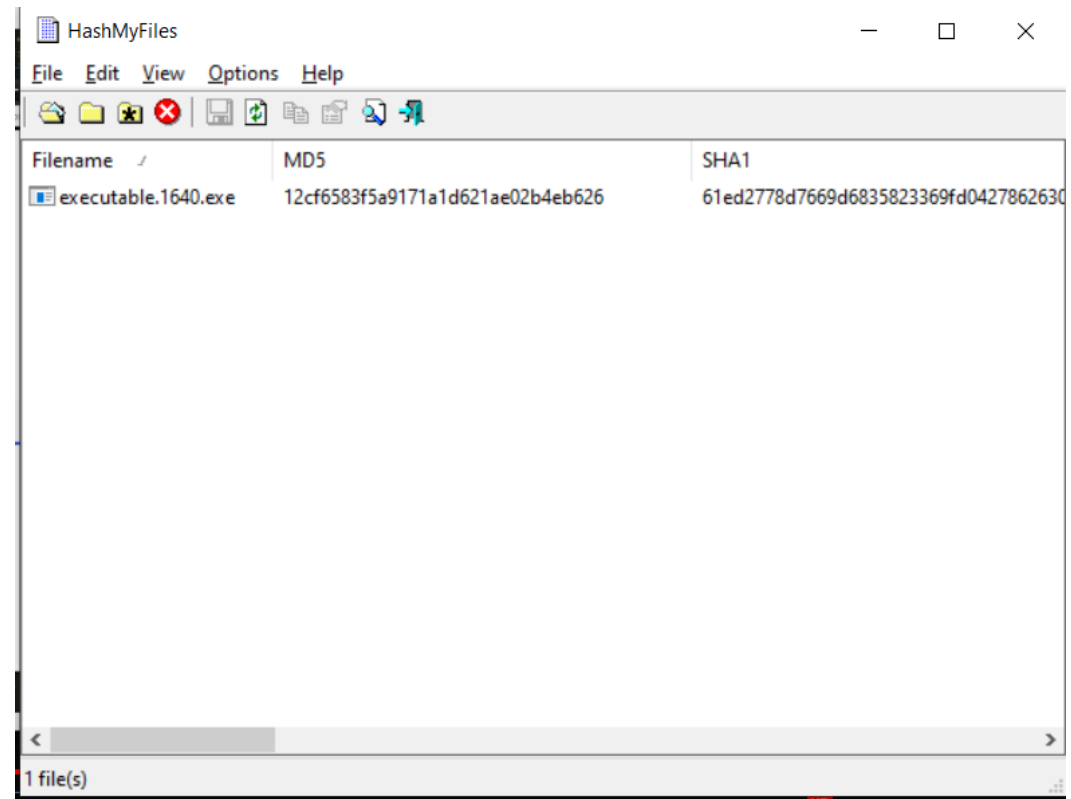
Nous pouvons créer une copie du fichier exécutable suspect sur notre machine locale (process dump).

Name	Date modified	Type	Size
 volatility.exe	12/27/2016 5:02 PM	Application	15,424 KB
 README.txt	12/24/2016 3:13 PM	Text Document	32 KB
 LICENSE.txt	7/7/2016 4:16 AM	Text Document	15 KB
 LEGAL.txt	7/7/2016 4:16 AM	Text Document	1 KB
 executable.1640.exe	1/8/2022 6:34 PM	Application	29 KB
 DumpIt.exe	1/8/2022 5:15 PM	Application	203 KB
 cridex.vmem	8/2/2012 5:23 AM	VMEM File	524,288 KB
 CREDITS.txt	12/27/2016 4:52 PM	Text Document	4 KB
 AUTHORS.txt	12/27/2016 4:44 PM	Text Document	1 KB

Génération du hash code de l'exécutable suspect, afin de le faire tester sur l'outil Virus-Total.

Hash-code du fichier Reader_sl.exe :

MD5- 12cf6583f5a9171a1d621ae02b4eb626





29 security vendors flagged this file as malicious

5b136147911b041f0126ce82dfd24c4e2c79553b65d3240ecea2dcab4452dcb5

AcroSpeedLaunch.exe

direct-cpu-clock-access idle peexe

28.50 KB
Size

2021-12-25 10:43:57 UTC
14 days ago

EXE

DETECTION	DETAILS	RELATIONS	BEHAVIOR	COMMUNITY	4
Ad-Aware	⚠ Trojan.GenericKD.41512677		Alibaba	⚠ Trojan:Win32/Multiop.788dce0e	
ALYac	⚠ Trojan.GenericKD.41512677		Arcabit	⚠ Trojan.Generic.D2796EE5	
BitDefender	⚠ Trojan.GenericKD.41512677		Comodo	⚠ Malware@#b2ihr9eixviv	
Cybereason	⚠ Malicious.3f5a91		Cylance	⚠ Unsafe	
Emsisoft	⚠ Trojan.GenericKD.41512677 (B)		eScan	⚠ Trojan.GenericKD.41512677	
FireEye	⚠ Trojan.GenericKD.41512677		Fortinet	⚠ PossibleThreat	
GData	⚠ Trojan.GenericKD.41512677		Ikarus	⚠ Trojan.Win32.Patched	
K7AntiVirus	⚠ Riskware (0040eff71)		K7GW	⚠ Riskware (0040eff71)	
Lionic	⚠ Trojan.Win32.Generic.4!c		MAX	⚠ Malware (ai Score=99)	

Auteur: HARAKI YOUNESS

Historique et autres noms des fichiers avec de tel logiciel malveillant

History ⓘ	
Creation Time	2008-06-12 09:37:53
First Submission	2012-09-19 10:23:46
Last Submission	2021-11-26 19:33:29
Last Analysis	2021-12-25 10:43:57
Names ⓘ	
AcroSpeedLaunch.exe	
executable.1640.exe	
reader_sl.exe	
module.1640.207bda0.400000.dll	
1640.reader_sl.exe	
executable.reader_sl.exe_1640.exe	
MODULE.1640.207BDA0.400000.DLL	
executable.1640.exe.bin	
module.1640.207bda0.400000.exe	
5b136147911b041f0126ce82dfd24c4e2c79553b65d3240ecea2dcab4452dcb5.bin	
⌵	

Auteur: HARAKI YOUNESS

Indicator of compromise (IoC) est un artefact observé sur un réseau ou dans un système d'exploitation qui, avec un niveau de confiance élevé, indique une intrusion informatique.

IOC

File: Reader_sl.exe

Directory : "C:\Program Files\Adobe\Reader 9.0\Reader\Reader_sl.exe"

MD5: 12cf6583f5a9171a1d621ae02b4eb626











SHA1: 61ed2778d7669d6835823369fd04278626303362

Further investigation






volatility.exe -f cridex.vmem --profile=WinXPSP2x86 memdump -p 1640 --dump-dir . :



On peut vérifier le contenu de la mémoire du processus utilisé par le malware, on générant un Hex-dump des données.

Strings est ensuite utilisé ensuite pour observer le contenu du fichier Hex.

Name	Date modified	Type	Size
 volatility.exe	12/27/2016 5:02 PM	Application	15,424 KB
 README.txt	12/24/2016 3:13 PM	Text Document	32 KB
 LICENSE.txt	7/7/2016 4:16 AM	Text Document	15 KB
 LEGAL.txt	7/7/2016 4:16 AM	Text Document	1 KB
 HashMyFiles.exe	12/1/2021 7:32 PM	Application	59 KB
 Dumplt.exe	1/8/2022 5:15 PM	Application	203 KB
 cridex.vmem	8/2/2012 5:23 AM	VMEM File	524,288 KB
 CREDITS.txt	12/27/2016 4:52 PM	Text Document	4 KB
 AUTHORS.txt	12/27/2016 4:44 PM	Text Document	1 KB
 1640.dmp	1/8/2022 7:00 PM	Memory Dump File	75,396 KB

Strings v2.54

Article • 06/22/2021 • 2 minutes to read •      +1

Is this page helpful?  

By Mark Russinovich

Published: June 22, 2021



[Download Strings](#) (534 KB)

Introduction

Working on NT and Win2K means that executables and object files will many times have embedded UNICODE strings that you cannot easily see with a standard ASCII strings or grep programs. So we decided to roll our own. Strings just scans the file you pass it for UNICODE (or ASCII) strings of a default length of 3 or more UNICODE (or ASCII) characters. Note that it works under Windows 95 as well.

strings 1640.dmp | grep "41.168.5.140" -C 5 :

Nous vérifions la connexion extérieure qui a été établie par notre processus infecté.

```
C:\Users\youn\OneDrive\Documents\cours m2 intense\Security_course\volatility_2.6_win64_standalone>strings 1640.dmp | grep "41.168.5.140" -C 5
http://91.121.103.143:8080/zb/v_01_a/in/
http://213.17.171.186:8080/zb/v_01_a/in/
http://59.90.221.6:8080/zb/v_01_a/in/
http://188.40.0.138:8080/zb/v_01_a/in/
http://216.24.197.66:8080/zb/v_01_a/in/
http://41.168.5.140:8080/zb/v_01_a/in/
http://125.19.103.198:8080/zb/v_01_a/in/
http://190.81.107.70:8080/zb/v_01_a/in/
http://211.44.250.173:8080/zb/v_01_a/in/
http://210.56.23.100:8080/zb/v_01_a/in/
http://85.214.204.32:8080/zb/v_01_a/in/
--
)Gz
DpI8
POST /zb/v_01_a/in/ HTTP/1.1
Accept: */*
User-Agent: Mozilla/5.0 (Windows; U; MSIE 7.0; Windows NT 6.0; en-US)
Host: 41.168.5.140:8080
Content-Length: 229
Connection: Keep-Alive
Cache-Control: no-cache
)Gz
2GR
```

Nous notons que le processus essaie de se connecter à plusieurs adresses IP, puis il exécute un POST, ce qui indique que la connexion envoie des informations locales à un serveur extérieur.
L'hôte se connecte à un chemin particulier : /zb/v_01_a/in/

URL PATH IOC : /zb/v_01_a/in/

strings 1640.dmp | grep -i "/zb/v_01_a/in/ : L'hôte essaie de se connecter à ces adresses IP.

```
4_standalone>strings 1640.dmp | grep -i "/zb/v_01_a/in/"
http://188.40.0.138:8080/zb/v_01_a/in/cp.php
http://155.98.65.40:8080/zb/v_01_a/in/
http://184.106.189.124:8080/zb/v_01_a/in/
http://91.228.154.199:8080/zb/v_01_a/in/
http://110.234.150.163:8080/zb/v_01_a/in/
http://164.15.21.2:8080/zb/v_01_a/in/
http://91.121.103.143:8080/zb/v_01_a/in/
http://213.17.171.186:8080/zb/v_01_a/in/
http://59.90.221.6:8080/zb/v_01_a/in/
http://188.40.0.138:8080/zb/v_01_a/in/
http://216.24.197.66:8080/zb/v_01_a/in/
http://41.168.5.140:8080/zb/v_01_a/in/
http://125.19.103.198:8080/zb/v_01_a/in/
http://190.81.107.70:8080/zb/v_01_a/in/
http://211.44.250.173:8080/zb/v_01_a/in/
http://210.56.23.100:8080/zb/v_01_a/in/
http://85.214.204.32:8080/zb/v_01_a/in/
POST /zb/v_01_a/in/ HTTP/1.1
http://188.40.0.138:8080/zb/v_01_a/in/cp.php
http://188.40.0.138:8080/zb/v_01_a/in/cp.php
```

strings 1640.dmp | grep -i ".com : On commence à suspecter l'activité de ce malware, on vérifie si il essaie de se connecter à des sites web.

Le malware essaie effectivement de se connecter à des sites bancaires. C'est un « red flag », car il montre que le malware essaie d'obtenir des informations bancaires personnelles de l'utilisateur.

```
*bankonline.sboff.com*
*bankofbermuda.com*
*tdcommercialbanking*
*bxs.com*
*solutions-corporate.com*
*cbbusinessonline.com*
*corporate.epfc.com*
*global-ebanking.com*
*mcb-home.com/online*
*metrobankdirect.com*
*ncms-inc.com*
*online.1stnb.com*
*westfield.accounts-in-view.com*
*secure.1stfedbank.com*
*securebanking.cbtk.com*
*secure.dalhartfederal.com*
*vectrabank.com/busi_bank_00.jsp*
*springbankconnect.com/views/login/*
*statebanktx.com/cgi-bin/prosperity.asp*
*treasurylinkweb.com*
*web.accessor.com*
*wtdirect.com*
*business.macu.com*
*cencorpcu.com*
*webinfocus.mandtbank.com*
*commercialservices.mandtbank.com*
*commercialservices*
*ifxmanager.bankofny.com*
*commercebusinessdirect.com*
*comerica.com/businessconnect/*
*firstbanks.com/olb*
*ebill.highmark.com*
*businessonline.huntington.com*
*businessmanager.com*
*cib.bankofthewest.com*
*secure2.bank.com*
```


volatility.exe -f cridex.vmem --profile=WinXPSP2x86 printkey -K

"Software\Microsoft\Windows\CurrentVersion\Run : Le répertoire du fichier KB00207877.exe est très suspect car il contient une entrée dans les clés de registre **run** pour exécuter un programme lorsqu'un utilisateur se connecte. Ainsi que les données d'application se trouvent sous un nom de répertoire d'une personne et sous Documents et paramètres.

```
Subkeys:
Values:
-----
Registry: \Device\HarddiskVolume1\Documents and Settings\Robert\NTUSER.DAT
Key name: Run (S)
Last updated: 2012-07-22 02:31:51 UTC+0000
Subkeys:
Values:
-----
REG_SZ KB00207877.exe : (S) "C:\Documents and Settings\Robert\Application Data\KB00207877.exe"
-----
Registry: \Device\HarddiskVolume1\WINDOWS\system32\config\default
Key name: Run (S)
Last updated: 2011-04-12 20:31:49 UTC+0000
Subkeys:
Values:
-----
Registry: \Device\HarddiskVolume1\Documents and Settings\LocalService\Local Settings\Application Data\Microsoft\Windows\UsrClass.dat
Key name: Run (S)
Last updated: 2011-04-13 00:55:13 UTC+0000
Subkeys:
Values:
-----
Registry: \Device\HarddiskVolume1\Documents and Settings\NetworkService\NTUSER.DAT
Key name: Run (S)
Last updated: 2011-04-13 00:49:16 UTC+0000
Subkeys:
Values:
```

strings 1640.dmp | grep -i "KB00207877.exe" : Effectivement, le fichier infecté par le malware est à l'origine de cette entrée dans la liste des **run**.
On trouve "KB00207877.exe" dans les commandes du dump de la mémoire du processus infecté.

```
KB00207877.exe  
KB00207877.EXE-040404D7.pf  
KB00207877.exe  
KB00207877.exe  
KB00207877.EXE-040404D7.pf  
^C
```

Etude de cas 2









Stuxnet

Stuxnet, cible spécifiquement les contrôleurs logiques programmables (PLC), qui permettent l'automatisation de processus électromécaniques tels que ceux utilisés pour contrôler les machines et les processus industriels, y compris les centrifugeuses à gaz pour la séparation des matières nucléaires.



Auteur: HAKKI YOUNESS

Nous devrions avoir **Volatility**, **Dumpit** et **cridex** memory image sous la même répertoire.

Name	Date modified	Type	Size
 AUTHORS.txt	12/27/2016 4:44 PM	Text Document	1 KB
 CREDITS.txt	12/27/2016 4:52 PM	Text Document	4 KB
 LEGAL.txt	7/7/2016 4:16 AM	Text Document	1 KB
 LICENSE.txt	7/7/2016 4:16 AM	Text Document	15 KB
 README.txt	12/24/2016 3:13 PM	Text Document	32 KB
 volatility_2.6_win64_standalone.exe	12/27/2016 5:02 PM	Application	15,424 KB
 DumpIt.exe	1/8/2022 5:15 PM	Application	203 KB
 cridex.vmem	8/2/2012 5:23 AM	VMEM File	524,288 KB

volatility.exe -f cridex.vmem imageinfo: Le profil d'image du dump de la mémoire est requis pour exécuter des analyses sur volatility.

1- Profil d'image : WinXPSP3x86

```
C:\Users\youne\OneDrive\Documents\cours m2 intense\Security_course\volat
ility.exe -f stuxnet.vmem imageinfo
Volatility Foundation Volatility Framework 2.6
INFO      : volatility.debug      : Determining profile based on KDBG search...
           Suggested Profile(s) : WinXPSP2x86, WinXPSP3x86 (Instantiated with WinXPSP2x86)
           AS Layer1 : IA32PagedMemoryPae (Kernel AS)
           AS Layer2 : FileAddressSpace (C:\Users\youne\OneDrive\Documents\cours m2 intense\Se
curity_course\volatility_2.6_win64_standalone\stuxnet.vmem)
           PAE type : PAE
           DTB : 0x319000L
           KDBG : 0x80545ae0L
           Number of Processors : 1
           Image Type (Service Pack) : 3
           KPCR for CPU 0 : 0xffdff000L
           KUSER_SHARED_DATA : 0xffdff000L
           Image date and time : 2011-06-03 04:31:36 UTC+0000
           Image local date and time : 2011-06-03 00:31:36 -0400
```

volatility.exe -f stuxnet.vmem --profile=WinXPSP3x86 pslist / pstree: En utilisant les informations du profil d'image, nous pouvons trouver des informations sur le vidage de la mémoire. « observer les processus qui ont été exécutés »

Offset(V)	Name	PID	PPID	Thds	Hnds	Sess	Wow64	Start	Exit
0x823c8830	System	4	0	59	403	-----	0		
0x820df020	smss.exe	376	4	3	19	-----	0	2010-10-29 17:08:53 UTC+0000	
0x821a2da0	csrss.exe	600	376	11	395	0	0	2010-10-29 17:08:54 UTC+0000	
0x81da5650	winlogon.exe	624	376	19	570	0	0	2010-10-29 17:08:54 UTC+0000	
0x82073020	services.exe	668	624	21	431	0	0	2010-10-29 17:08:54 UTC+0000	
0x81e70020	lsass.exe	680	624	19	342	0	0	2010-10-29 17:08:54 UTC+0000	
0x823315d8	vmacthlp.exe	844	668	1	25	0	0	2010-10-29 17:08:55 UTC+0000	
0x81db8da0	svchost.exe	856	668	17	193	0	0	2010-10-29 17:08:55 UTC+0000	
0x81e61da0	svchost.exe	940	668	13	312	0	0	2010-10-29 17:08:55 UTC+0000	
0x822843e8	svchost.exe	1032	668	61	1169	0	0	2010-10-29 17:08:55 UTC+0000	
0x81e18b28	svchost.exe	1080	668	5	80	0	0	2010-10-29 17:08:55 UTC+0000	
0x81ff7020	svchost.exe	1200	668	14	197	0	0	2010-10-29 17:08:55 UTC+0000	
0x81fee8b0	spoolsv.exe	1412	668	10	118	0	0	2010-10-29 17:08:56 UTC+0000	
0x81e0eda0	jqs.exe	1580	668	5	148	0	0	2010-10-29 17:09:05 UTC+0000	
0x81fe52d0	vmtoolsd.exe	1664	668	5	284	0	0	2010-10-29 17:09:05 UTC+0000	
0x821a0568	VMUpgradeHelper	1816	668	3	96	0	0	2010-10-29 17:09:08 UTC+0000	
0x8205ada0	alg.exe	188	668	6	107	0	0	2010-10-29 17:09:09 UTC+0000	
0x820ec7e8	explorer.exe	1196	1728	16	582	0	0	2010-10-29 17:11:49 UTC+0000	
0x820ecc10	wscntfy.exe	2040	1032	1	28	0	0	2010-10-29 17:11:49 UTC+0000	
0x81e86978	TSVNCache.exe	324	1196	7	54	0	0	2010-10-29 17:11:49 UTC+0000	
0x81fc5da0	VMwareTray.exe	1912	1196	1	50	0	0	2010-10-29 17:11:50 UTC+0000	
0x81e6b660	VMwareUser.exe	1356	1196	9	251	0	0	2010-10-29 17:11:50 UTC+0000	
0x8210d478	jusched.exe	1712	1196	1	26	0	0	2010-10-29 17:11:50 UTC+0000	
0x82279998	imapi.exe	756	668	4	116	0	0	2010-10-29 17:11:54 UTC+0000	
0x822b9a10	wuauclt.exe	976	1032	3	133	0	0	2010-10-29 17:12:03 UTC+0000	
0x81c543a0	Procmon.exe	660	1196	13	189	0	0	2011-06-03 04:25:56 UTC+0000	
0x81fa5390	wmiprvse.exe	1872	856	5	134	0	0	2011-06-03 04:25:58 UTC+0000	
0x81c498c8	lsass.exe	868	668	2	23	0	0	2011-06-03 04:26:55 UTC+0000	
0x81c47c00	lsass.exe	1928	668	4	65	0	0	2011-06-03 04:26:55 UTC+0000	
0x81c0cda0	cmd.exe	968	1664	0	-----	0	0	2011-06-03 04:31:35 UTC+0000	2011-06-03 04:31:36 UTC+0000
0x81f14938	ipconfig.exe	304	968	0	-----	0	0	2011-06-03 04:31:35 UTC+0000	2011-06-03 04:31:36 UTC+0000

Note 1:

LSASS, ou service de sous-système de l'autorité de sécurité locale, est un processus qui fonctionne dans le cadre du système d'exploitation Microsoft Windows. LSASS fait partie du processus de maintenance et d'application des protocoles de sécurité du système d'exploitation.

Note 2:

La relation parent-enfant normale du fichier lsass

winlogon.exe (624) démarre, DATE: 2010-10-29 17:08:54

a) services.exe (668), DATE: 2010-10-29 17:08:54

b) lsass.exe (680), 2010-10-29 17:08:54

La relation parent-enfant Stuxnet

services.exe(668) **n'est PAS censé le faire, mais démarre**

a) lsass.exe (1928), DATE : 03/06/2011 04:26:55

b) lsass.exe (868), DATE : 03/06/2011 04:26:55

Notez que ces deux processus lsass.exe ont été créés 216 jours après le démarrage de winlogin.exe

Offset(V)	Name	PID	PPID	Thds	Hnds	Sess	Wow64	Start
0x823c8830	System	4	0	59	403	-----	0	
0x820df020	smss.exe	376	4	3	19	-----	0	2010-10-29 17:08:53 UTC+0000
0x821a2da0	csrss.exe	600	376	11	395	0	0	2010-10-29 17:08:54 UTC+0000
0x81da5650	winlogon.exe	624	376	19	570	0	0	2010-10-29 17:08:54 UTC+0000
0x82073020	services.exe	668	624	21	431	0	0	2010-10-29 17:08:54 UTC+0000
0x81e70020	lsass.exe	680	624	19	342	0	0	2010-10-29 17:08:54 UTC+0000
0x823315d8	vmacthlp.exe	844	668	1	25	0	0	2010-10-29 17:08:55 UTC+0000
0x81db8da0	svchost.exe	856	668	17	193	0	0	2010-10-29 17:08:55 UTC+0000
0x81e61da0	svchost.exe	940	668	13	312	0	0	2010-10-29 17:08:55 UTC+0000
0x822843e8	svchost.exe	1032	668	61	1169	0	0	2010-10-29 17:08:55 UTC+0000
0x81e18b28	svchost.exe	1080	668	5	80	0	0	2010-10-29 17:08:55 UTC+0000
0x81ff7020	svchost.exe	1200	668	14	197	0	0	2010-10-29 17:08:55 UTC+0000
0x81fee8b0	spoolsv.exe	1412	668	10	118	0	0	2010-10-29 17:08:56 UTC+0000
0x81e0eda0	jqs.exe	1580	668	5	148	0	0	2010-10-29 17:09:05 UTC+0000
0x81fe52d0	vmtoolsd.exe	1664	668	5	284	0	0	2010-10-29 17:09:05 UTC+0000
0x821a0568	VMUpgradeHelper	1816	668	3	96	0	0	2010-10-29 17:09:08 UTC+0000
0x8205ada0	alg.exe	188	668	6	107	0	0	2010-10-29 17:09:09 UTC+0000
0x820ec7e8	explorer.exe	1196	1728	16	582	0	0	2010-10-29 17:11:49 UTC+0000
0x820ecc10	wscntfy.exe	2040	1032	1	28	0	0	2010-10-29 17:11:49 UTC+0000
0x81e86978	TSVNCache.exe	324	1196	7	54	0	0	2010-10-29 17:11:49 UTC+0000
0x81fc5da0	VMwareTray.exe	1912	1196	1	50	0	0	2010-10-29 17:11:50 UTC+0000
0x81e6b660	VMwareUser.exe	1356	1196	9	251	0	0	2010-10-29 17:11:50 UTC+0000
0x8210d478	jusched.exe	1712	1196	1	26	0	0	2010-10-29 17:11:50 UTC+0000
0x82279998	imapi.exe	756	668	4	116	0	0	2010-10-29 17:11:54 UTC+0000
0x822b9a10	wuauclt.exe	976	1032	3	133	0	0	2010-10-29 17:12:03 UTC+0000
0x81c543a0	Procmon.exe	660	1196	13	189	0	0	2011-06-03 04:25:56 UTC+0000
0x81fa5390	wmiprvse.exe	1872	856	5	134	0	0	2011-06-03 04:25:58 UTC+0000
0x81c498c8	lsass.exe	868	668	2	23	0	0	2011-06-03 04:26:55 UTC+0000
0x81c47c00	lsass.exe	1928	668	4	65	0	0	2011-06-03 04:26:55 UTC+0000
0x81c0cda0	cmd.exe	968	1664	0	-----	0	0	2011-06-03 04:31:35 UTC+0000
0x81f14938	ipconfig.exe	304	968	0	-----	0	0	2011-06-03 04:31:35 UTC+0000

2- Processus suspects : Les processus lsass.exe (PID = 1928)
& lsass.exe (PID = 868), créés par services.exe

volatility.exe -f stuxnet.vmem --profile=WinXPSP3x86 psxview

Nous vérifions si aucun autre processus caché n'a été exécuté
(Comportement typique de certains malwares)

→ Aucun processus soupçonné n'est caché.

Offset(P)	Name	PID	pslist	psscan	thrdproc	pspcid	csrss	session	deskthrd	ExitTime
0x01e47c00	lsass.exe	1928	True	True	True	True	True	True	True	
0x021a5390	wmiprvse.exe	1872	True	True	True	True	True	True	True	
0x021c5da0	VMwareTray.exe	1912	True	True	True	True	True	True	True	
0x02479998	imapi.exe	756	True	True	True	True	True	True	True	
0x02273020	services.exe	668	True	True	True	True	True	True	True	
0x02018b28	svchost.exe	1080	True	True	True	True	True	True	True	
0x021ee8b0	spoolsv.exe	1412	True	True	True	True	True	True	True	
0x02061da0	svchost.exe	940	True	True	True	True	True	True	True	
0x024b9a10	wuauclt.exe	976	True	True	True	True	True	True	True	
0x0200eda0	jqs.exe	1580	True	True	True	True	True	True	True	
0x021f7020	svchost.exe	1200	True	True	True	True	True	True	True	
0x01e543a0	Procmon.exe	660	True	True	True	True	True	True	True	
0x022ecc10	wscntfy.exe	2040	True	True	True	True	True	True	True	
0x02070020	lsass.exe	680	True	True	True	True	True	True	True	
0x01e498c8	lsass.exe	868	True	True	True	True	True	True	True	
0x01fa5650	winlogon.exe	624	True	True	True	True	True	True	True	
0x0230d478	jusched.exe	1712	True	True	True	True	True	True	True	
0x025315d8	vmacthlp.exe	844	True	True	True	True	True	True	True	
0x0206b660	VMwareUser.exe	1356	True	True	True	True	True	True	True	
0x021e52d0	vmtoolsd.exe	1664	True	True	True	True	True	True	True	
0x01fb8da0	svchost.exe	856	True	True	True	True	True	True	True	
0x024843e8	svchost.exe	1032	True	True	True	True	True	True	True	
0x0225ada0	alg.exe	188	True	True	True	True	True	True	True	
0x023a0568	VMUpgradeHelper	1816	True	True	True	True	True	True	True	
0x022ec7e8	explorer.exe	1196	True	True	True	True	True	True	True	
0x02086978	TSVNCache.exe	324	True	True	True	True	True	True	True	
0x025c8830	System	4	True	True	True	True	False	False	False	
0x02114938	ipconfig.exe	304	True	True	False	True	False	False	False	2011-06-03 04:31:36 UTC+0000
0x023a2da0	csrss.exe	600	True	True	True	True	False	True	True	
0x022df020	smss.exe	376	True	True	True	True	False	False	False	
0x01e0cda0	cmd.exe	968	True	True	False	True	False	False	False	2011-06-03 04:31:36 UTC+0000

volatility.exe -f stuxnet.vmem --profile=WinXPSP3x86 cmdline :

Une fois que nous avons récupéré les informations de connexion, les informations de commande. Nous pouvons enquêter sur les dernières lignes de commande qui ont été exécutées en mémoire.

Rappel

1- Profil d'image : WinXPSP3x86

2- Processus suspects : Les processus lsass.exe (PID = 1928)
& lsass.exe (PID = 868), créés par services.exe

Un fichier suspect, qui a crée les deux processus 868 et 1928.












3- Exécutables suspect :

"C:\WINDOWS\system32\lsass.exe"

```
*****  
lsass.exe pid: 868  
Command line : "C:\WINDOWS\system32\lsass.exe"  
*****  
lsass.exe pid: 1928  
Command line : "C:\WINDOWS\system32\lsass.exe"  
*****
```

volatility.exe -f stuxnet.vmem --profile=WinXPSP3x86 procdump -p 1928 --dump-dir . :

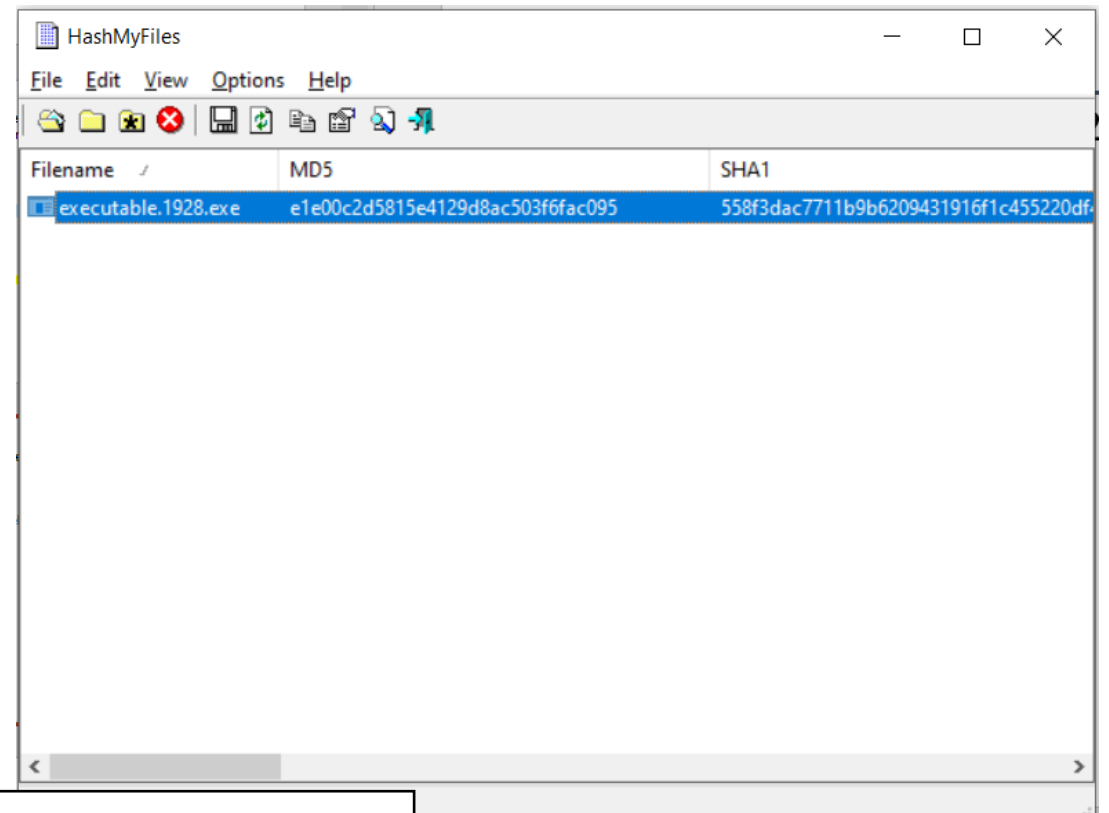
Nous pouvons créer une copie du fichier exécutable suspect sur notre machine locale (process dump).

 volatility.exe	12/27/2016 5:02 PM	Application	15,424 KB
 stuxnet.vmem	6/3/2011 9:06 AM	VMEM File	524,288 KB
 strings.exe	6/22/2021 2:58 PM	Application	362 KB
 README.txt	12/24/2016 3:13 PM	Text Document	32 KB
 LICENSE.txt	7/7/2016 4:16 AM	Text Document	15 KB
 LEGAL.txt	7/7/2016 4:16 AM	Text Document	1 KB
 HashMyFiles.exe	12/1/2021 7:32 PM	Application	59 KB
 executable.1928.exe	1/8/2022 9:18 PM	Application	9 KB
 cridex.vmem	8/2/2012 5:23 AM	VMEM File	524,288 KB
 CREDITS.txt	12/27/2016 4:52 PM	Text Document	4 KB
 AUTHORS.txt	12/27/2016 4:44 PM	Text Document	1 KB

Génération du hash code de l'exécutable suspect, afin de le faire tester sur l'outil Virus-Total.

Hash-code du fichier Reader_sl.exe :

MD5- e1e00c2d5815e4129d8ac503f6fac095



Auteur: HARAKI YOUNESS



50 security vendors flagged this file as malicious



20a3c5f02b6b79bcac9adaef7ee138763054bbedc298fb2710b5adaf9b74a47d
executable.1928.exe

9.00 KB
Size

2021-10-03 00:10:58 UTC
3 months ago



detect-debug-environment long-sleeps peexe

DETECTION

DETAILS

RELATIONS

BEHAVIOR

COMMUNITY 12

Ad-Aware

Gen:Variant.Kazy.76811

AhnLab-V3

Trojan/Win32.Genome.R150575

Alibaba

Trojan:Win32/Stuxnet.99b5316e

ALYac

Gen:Variant.Kazy.76811

Antiy-AVL

Trojan/Generic.ASMalwFH.7976F9

Avast

Win32:Duqu-F [Rtk]

AVG

Win32:Duqu-F [Rtk]

Avira (no cloud)

TR/Crypt.XPACK.Gen

BitDefender

Gen:Variant.Kazy.76811

BitDefenderTheta

AI:Packer.C89F107B21

Bkav Pro

W32.AIDetect.malware1

Comodo

Malware@#2y7rz4fy5vpio

CrowdStrike Falcon

Win/malicious_confidence_100% (W)

Cybereason

Malicious.d5815e

Cylance

Unsafe

Cynet

Malicious (score: 100)

Auteur: HARAKI YOUNESS

Historique et autres noms de fichiers avec de tels logiciels malveillants

History ⓘ

Creation Time	2010-01-13 10:00:53
First Submission	2011-10-31 09:49:00
Last Submission	2021-10-03 00:10:58
Last Analysis	2021-10-03 00:10:58

Names ⓘ

executable.1928.exe

module.1928.1e47c00.1000000.dll

1928.lsass.exe

lsass_p1928.exe

20a3c5f02b6b79bcac9adaef7ee138763054bbedc298fb2710b5adaf9b74a47d.exe

20a3c5f02b6b79bcac9adaef7ee138763054bbedc298fb2710b5adaf9b74a47d.bin.exe

lsass.exe

check

executable.1928.exe.---

file-3103244_exe

Auteur: HARAKI YOUNESS

Indicator of compromise (IoC) est un artefact observé sur un réseau ou dans un système d'exploitation qui, avec un niveau de confiance élevé, indique une intrusion informatique.

IOC

File: lsass.exe

Directory : "C:\WINDOWS\system32\lsass.exe"

MD5: e1e00c2d5815e4129d8ac503f6fac095

SHA1: 558f3dac7711b9b6209431916f1c455220df40a7












Further investigation

volatility.exe -f stuxnet.vmem --profile=WinXPSP3x86

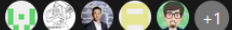
memdump -p 1928 --dump-dir . :



On peut vérifier le contenu de la mémoire du processus utilisé par le malware, on produisons un Hex-dump des données.

Strings est ensuite utilisé pour observer le contenu du fichier Hex.

Name	Date modified	Type	Size
 volatility.exe	12/27/2016 5:02 PM	Application	15,424 KB
 stuxnet.vmem	6/3/2011 9:06 AM	VMEM File	524,288 KB
 strings.exe	6/22/2021 2:58 PM	Application	362 KB
 README.txt	12/24/2016 3:13 PM	Text Document	32 KB
 LICENSE.txt	7/7/2016 4:16 AM	Text Document	15 KB
 LEGAL.txt	7/7/2016 4:16 AM	Text Document	1 KB
 HashMyFiles.exe	12/1/2021 7:32 PM	Application	59 KB
 crindex.vmem	8/2/2012 5:23 AM	VMEM File	524,288 KB
 CREDITS.txt	12/27/2016 4:52 PM	Text Document	4 KB
 AUTHORS.txt	12/27/2016 4:44 PM	Text Document	1 KB
 1928.dmp	1/8/2022 9:24 PM	Memory Dump File	133,736 KB


Strings v2.54

Article • 06/22/2021 • 2 minutes to read • 

Is this page helpful?  

By Mark Russinovich

Published: June 22, 2021

 [Download Strings](#) (534 KB)

Introduction

Working on NT and Win2K means that executables and object files will many times have embedded UNICODE strings that you cannot easily see with a standard ASCII strings or grep programs. So we decided to roll our own. Strings just scans the file you pass it for UNICODE (or ASCII) strings of a default length of 3 or more UNICODE (or ASCII) characters. Note that it works under Windows 95 as well.

Auteur: HARAKI YOUNESS

volatility.exe -f stuxnet.vmem --profile= WinXPSP3x86 printkey -K

"Software\Microsoft\Windows\CurrentVersion\Run" : Le répertoire du fichier **NTUSER.DAT** est très suspect car il contient une entrée dans la liste des Run, et les données d'application se trouvent sous Documents et paramètres.

```
Subkeys:
Values:
-----
Registry: \Device\HarddiskVolume1\Documents and Settings\Robert\NTUSER.DAT
Key name: Run (S)
Last updated: 2012-07-22 02:31:51 UTC+0000

Subkeys:
Values:
-----
REG_SZ KB00207877.exe : (S) "C:\Documents and Settings\Robert\Application Data\KB00207877.exe"
-----
Registry: \Device\HarddiskVolume1\WINDOWS\system32\config\default
Key name: Run (S)
Last updated: 2011-04-12 20:31:49 UTC+0000

Subkeys:
Values:
-----
Registry: \Device\HarddiskVolume1\Documents and Settings\LocalService\Local Settings\Application Data\Microsoft\Windows\UsrClass.dat
Key name: Run (S)
Last updated: 2011-04-13 00:55:13 UTC+0000

Subkeys:
Values:
-----
Registry: \Device\HarddiskVolume1\Documents and Settings\NetworkService\NTUSER.DAT
Key name: Run (S)
Last updated: 2011-04-13 00:49:16 UTC+0000

Subkeys:
Values:
```

strings 1928.dmp | grep -i "NTUSER.DAT" : Effectivement, le fichier infecté par le malware est à l'origine de cette entrée dans la liste des 'run'.
On trouve "KB00207877.exe" dans les commandes du dump de la mémoire du processus infecté.

```
C:\Users\youne\OneDrive\Documents\cours m2 intense\Security_course\volatility_2.6_win64_standalone>strings 1928.dmp | grep -i "NTUSER.DAT"
ntuser.dat.log
ntuser.dat
ntuser.dat.LOG
NTUSER.DAT
ntuser.dat.LOG
ntuser.dat.LOG
NTUSER.DAT
ntuser.dat.LOG
NTUSER.DAT
ntuser.dat.LOG
NTUSER.DAT
ntuser.dat.LOG
```