Volatility: Part 2 – Malware in hiberfil.sys

Written by

Dan Doonan and Catherine Stamm

Researched by

Dan Doonan, Connor Hicks, David Leberfinger, and Catherine Stamm

The Senator Patrick Leahy Center for Digital Investigation

Champlain College

December 4, 2012

Disclaimer:

This document contains information based on research that has been gathered by employee(s) of The Senator Patrick Leahy Center for Digital Investigation (LCDI). The data contained in this project is submitted voluntarily and is unaudited. Every effort has been made by LCDI to assure the accuracy and reliability of the data contained in this report. However, LCDI nor any of our employees make no representation, warranty or guarantee in connection with this report and hereby expressly disclaims any liability or responsibility for loss or damage resulting from use of this data. Information in this report can be downloaded and redistributed by any person or persons. Any redistribution must maintain the LCDI logo and any references from this report must be properly annotated.

Contents

1 Introduction

1.1 Background

With an increase in the amount of computer systems being employed comes an increase in the potential threats and vulnerabilities they face. Malware has become very prominent over the years and has proved to be quite detrimental to systems. Having the capabilities to understand and recognize a threat is imperative when it comes to analyzing a device. For part two of our Volatility project, we have taken on malware analysis by searching the hiberfil.sys file. This file is created when a system is put into hibernation mode and allows the system to boot up quicker than normal by saving a system's contents to hiberfil.sys. Any malware that may be running in the background can be found within hiberfily.sys by converting it to a raw image and then using Volatility for analysis.

1.2 Research Questions

- Does Volatility convert hiberfil.sys on its own?
    - If not, what other software will?
- Will Volatility find traces of malware in hiberfil.sys?
- Besides malfind, what plugins would be used to analyze malware?

2 Methods and Findings

To set up our test environment, we created a fresh Windows XP SP3 virtual machine. We turned the VM into a sandbox so it would not infect our local machines and then infected the VM with the malware fujacks.exe. Fujacks was provided to use by a student in a malware class at Champlain. We then put the virtual machine into hibernation so we could later analyze the hiberfil.sys file using Volatility.

To extract the hiberfil.sys file, we used FTK Imager. The VMware virtual disk file (.vmdk) was added as an Image File. After navigating to the root of the C: drive we were able to extract the hiberfil.sys file with the export file option. For the hiberfil.sys file to be listed in Windows Explorer, hidden files must be turned on. This setting can be found in Control Panel -> Folder Options -> View.

2.1 Overarching Methodology

Next we needed to convert the hiberfil.sys file into a raw image so that Volatility could analyze it. We were able to do this with both Moonsols Windows Memory Toolkit and Volatility. Example commands for both tools are as follows:

- Moonsols: hibr2dmp.exe hiberfil.sys converted.raw
- Volatility: volatility-2.2.standalone.exe -f hiberfil.sys imagecopy -O converted.raw

The Windows Memory Toolkit can be downloaded here: http://www.moonsols.com/windows-memory-toolkit/

Volatility can be downloaded here: https://www.volatilesystems.com/default/volatility#platforms

The plugins used for this analysis were: pstree, printkey, sockscan, and dlllist.

2.2 Results

We began the analysis by running the pstree command to get a list of running processes on the system:

volatility-2.2.standalone.exe -f converted.raw --profile=WinXPSP3x86 pstree

```
Name                                                  Pid   PPid   Thds   Hnds Time
-------------------------------------------------- ------ ------ ------ ------ --------------------
 0x823c8830:System                                      4      0     58    171 1970-01-01 00:00:00
. 0x82227020:smss.exe                                 540      4      3     19 2012-11-26 19:46:05
.. 0x81e7b020:winlogon.exe                            636    540     27    557 2012-11-26 19:46:07
... 0x82020020:logonui.exe                           1128    636      4    140 2012-11-26 19:46:57
... 0x81fac6c8:lsass.exe                              700    636     24    355 2012-11-26 19:46:09
... 0x82047020:services.exe                           688    636     15    251 2012-11-26 19:46:09
.... 0x82248ae8:vmacthlp.exe                          848    688      1     25 2012-11-26 19:46:10
.... 0x82080b10:svchost.exe                          1156    688     12    177 2012-11-26 19:46:13
.... 0x81e708a0:svchost.exe                           944    688     10    222 2012-11-26 19:46:11
.... 0x81eec560:svchost.exe                           876    688     20    195 2012-11-26 19:46:11
.... 0x81ee72c8:svchost.exe                          1100    688      5     61 2012-11-26 19:46:12
.... 0x81ee3020:spoolsv.exe                          1268    688     13    142 2012-11-26 19:46:14
.... 0x81e68880:alg.exe                              1560    688      7     99 2012-11-26 19:46:18
.... 0x81fa3cb8:svchost.exe                           996    688     79   1187 2012-11-26 19:46:11
.. 0x81cec718:csrss.exe                               612    540     13    367 2012-11-26 19:46:07
 0x81e54970:explorer.exe                             1828   1724     19    350 2012-11-26 19:46:27
. 0x821af9e0:TXP1atform.exe                           380   1828     13    173 2012-11-26 19:46:43
.. 0x81e48780:IEXPLORE.EXE                            588    380     10    279 2012-11-26 19:46:55
```

- The TXP1atform.exe process ((highlighted above) immediately stood out due to its name. An internet search for TXP1atform found the following malware definition from Microsoft. This definition stated that this malware modified registry keys. Another search also validates that TXPlatform.exe is associated with the worm FUJACKS. Upon examining the registry keys specified in the definition using the printkey command we found the following:

volatility-2.2.standalone.exe -f converted.raw --profile=WinXPSP3x86 printkey -K "Software\Microsoft\Windows\CurrentVersion\Run"

```
REG_SZ        Explorer        : (S) C:\WINDOWS\system32\drivers\TXP1atform.exe4
----------------------------
Registry: \Device\HarddiskVolume1\WINDOWS\system32\config\default
Key name: Run (S)
Last updated: 2012-11-26 14:30:35
```

- This key ensures that TXP1atform.exe is started when the system is turned on.

volatility-2.2.standalone.exe -f converted.raw --profile=WinXPSP3x86 printkey -K "Microsoft\Windows\CurrentVersion\Explorer\Advanced\Folder\Hidden\SHOWALL"

```
----------------------------
Registry: \Device\HarddiskVolume1\WINDOWS\system32\config\software
Key name: SHOWALL (S)
Last updated: 2012-11-26 19:41:30

Subkeys:

Values:
REG_SZ        RegPath         : (S) Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced5
REG_SZ        Text            : (S) @shell32.dll,-305005
REG_SZ        Type            : (S) radio5
REG_DWORD     CheckedValue    : (S) 0
REG_SZ        ValueName       : (S) Hidden5
REG_DWORD     DefaultValue    : (S) 2
REG_DWORD     HKeyRoot        : (S) 2147483649
REG_SZ        HelpID          : (S) shell.hlp#511055
```

- This key ensures that hidden files are not shown in Windows Explorer.

volatility-2.2.standalone.exe -f converted.raw --profile=WinXPSP3x86 printkey -K "Software\Microsoft\Windows\CurrentVersion\Policies\Explorer"

```
----------------------------
Registry: \Device\HarddiskVolume1\Documents and Settings\Administrator\NTUSER.DAT
Key name: Explorer (S)
Last updated: 2012-11-26 19:45:02

Subkeys:

Values:
REG_DWORD     NoDriveTypeAutoRun : (S) 128
REG_DWORD     NoSimpleNetIDList : (S) 1
----------------------------
```

- This key ensures that AutoPlay is enabled.

- Going back to the original data presented by pstree, the IEXPLORE.EXE process was open under TXP1atform. This stood out as suspicious since Internet Explorer was never opened. This process was explained in the definition as being started in the background by the malware to access specific remote hosts. The definition also lists remote hosts that the malware will attempt to connect to. As networking is disabled on this VM no internet traffic was created to check this. However, using the sockscan command the following was found:

volatility-2.2.standalone.exe -f converted.raw --profile=WinXPSP3x86 sockscan

```
Offset(P)      PID   Port  Proto Protocol        Address         Create Time
---------- ------ ------ ------ -------------- --------------- -----------
0x01ed2ce8    700    500     17 UDP            0.0.0.0         2012-11-26 19:46:15
0x020cee98    996   1028     17 UDP            127.0.0.1       2012-11-26 19:46:30
0x02223398    588   1030     17 UDP            127.0.0.1       2012-11-26 19:46:56
0x0223fe98    944    135      6 TCP            0.0.0.0         2012-11-26 19:46:11
0x02278e08   1560   1025      6 TCP            127.0.0.1       2012-11-26 19:46:18
0x0227cbb8    700      0    255 Reserved       0.0.0.0         2012-11-26 19:46:15
0x023aee98   1156   1900     17 UDP            127.0.0.1       2012-11-26 19:46:30
0x02421e98      4    445     17 UDP            0.0.0.0         2012-11-26 19:46:05
0x024678e0      4   1029      6 TCP            0.0.0.0         2012-11-26 19:46:31
0x02481dd0    700   4500     17 UDP            0.0.0.0         2012-11-26 19:46:15
0x02482270    996    123     17 UDP            127.0.0.1       2012-11-26 19:46:16
0x0251e4a0      4    445      6 TCP            0.0.0.0         2012-11-26 19:46:05
```

PID 588 is IEXPLORE.EXE and is connected to the local host on UDP port 1030. It is used for BBN IAD.

Using the dlllist command we were able to see the all of the dlls in use by each process. The command that opened the IEXPLORE.EXE process including the website that it attempted to access was also shown. This address was listed in the definition as one of the web pages that the malware attempted to contact.

volatility-2.2.standalone.exe -f converted.raw --profile=WinXPSP3x86 dlllist -p 588

```
IEXPLORE.EXE pid:    588
Command line : "C:\Program Files\Internet Explorer\iexplore.exe" http://www.xinxinbaidu.com.cn/htm/1.htm
Service Pack 3
```

3 Conclusion

It is definitely possible, and fairly easy, to convert a hiberfil.sys file for analysis. This is incredibly useful for forensic examiners, as many users put their computers into hibernation mode at some point. Being able to extract this file and analyze it with such a powerful tool such as Volatility helps us put together a more detailed summary of what was occurring on a system. If malware is hiding and running in the background, it can be found and associated with certain processes using Volatility. Depending on the findings, an investigator will be able to search other parts of the infected systems for traces of origin, created files as a result of the malware, where the malware is hiding, and much more.

4 Further Work

A lot more can be done with this malware analysis. We are still researching all the different plugins within Volatility and are learning what exactly they can show us. Some plugins provided us with data that we did not understand or know what it meant and therefore did not include it in this report. Because we are still learning about the software, and there will always be more to learn, the analysis of fujacks.exe is far from complete. Hopefully as we dive further into this project we can develop more of an understanding for malware analysis and Volatility itself to add to this report.

5 References

https://www.microsoft.com/security/portal/Threat/Encyclopedia/Entry.aspx?Name=Virus%3aWin32%2fViking.gen!B

http://www.threatexpert.com/files/txplatform.exe.html