Memory Analysis Laboratory Exercise WK5
Cyber Crime Investigations and Forensics II CYB-356-Z1
Professor: Dennis Labossiere
Date: 08/05/2018
Examiner Name: Brian T. Carr

# **Table of** Contents

# List of Illustrative Materials

## Tables



## Figures

# Executive Summary

## Background

Senior Project Manager Alyssa Gleason has made it evident that she has suspicions that Schultz and Dooley Inc. employees are using the Internet for personal reasons during scheduled work hours. Alyssa Gleason had previously extracted a captured Random Access Memory (RAM) image from a suspected offenders' workstation.

## Request

On 8/1/2018 Senior Project Manager Alyssa Gleason requested that the Senior Forensic Examiner at Schultz and Dooley Inc. examine the memory image file previously extracted by Senior Project Manager Alyssa Gleason. Alyssa Gleason specifically requests that the Examiner locate and record any instances of browser activity. The Examiner was requested to analyze the data in order to enumerate evidence relating to the misuse of company resources.

## Summary of Findings

The Examiner was able to determine various important pieces of information from analyzing the memory.img file using the Volatility Framework. By implementing Volatility, the Examiner determined the possible profiles were: **Win2008R2SP0x64, Win7SP1x64, Win7SP0x64,** and **Win2008R2SP1x64**. The Examiner was able to determine that the Operating System was either **Windows 7 x64** or **Windows 2008 R2 x64**. Both possible Operating Systems were running 64-bit architectures, so the Examiner determined the memory image was taken from a computer running on a 64-bit architecture processor. It was evident to the Examiner that there were instances of Google Chrome and Internet Explorer active at the time of memory acquisition. The Examiner also determined the that the Process ID of Google Update was: **3300**, and the application running on Process ID 3780 was: **Adobe Acrobat Reader**. The Examiner was able to locate the history of commands, and the DLL list. The Examiner was able to successfully determine that AVG antivirus ran from: **C:\Program Files (x86)\AVG\Av\avgcsrva.exe**. The Examiner was able to successfully locate three instances of Google Chrome running with an established connection during the memory image capture. The three outbound Google Chrome connections were: **74.125.228.206:443**, **74.125.228.216:80**, and **184.168.27.206:80**.

## Evidence

Table 1 outlines the evidence items of this case.

| Description | Designation | Filename | MD5 Hash |
|---|---|---|---|
| Evidence Provided | Preservation Copy | memory.zip | 65155148944d1096cdff0fd678d85664c |
| Evidence Examined | Working Copy | WC_memory.zip | 65155148944d1096cdff0fd678d85664c |
| Evidence Created | Preservation Copy | PRES_memory.zip | 65155148944d1096cdff0fd678d85664c |
| Supplemental Files | Supplemental Files | LAB5typescript.tar.gz | 4163f9af0aed10d1ffe18321941654bd |

*Table 1: Case evidence items*

# Collection and Analysis

## Collection

On 8/5/2018, the Examiner utilized Wget v1.17.1 to download the file memory.zip from the URL httsp://s3-us-west.amazonaws.com/digitalforensicsworkbook/memory.zip. The Examiner can be seen downloading memory.img in *Figure 1* below. Once downloaded the Examiner began to conduct proper digital evidence preservation techniques. The Examiner utilized the Cp command to create WC_memory.zip and PRES_memory.zip. Once all three zip files were located in ~/Desktop/MemoryLab/, The Examiner ran Hashdeep v4.4 to retrieve each .zip files MD5 hash value. The MD5 hash values and the Examiner's implementation of Hashdeep can be seen in *Figure 2* below. The Examiner successfully obtained the MD5 hash values with Hashdeep v4.4, then compared them to the MD5 hash value provided on the webpage hosting the .zip file which can be seen in *Figure 3*.


*Figure 1: Wget Download of memory.zip*

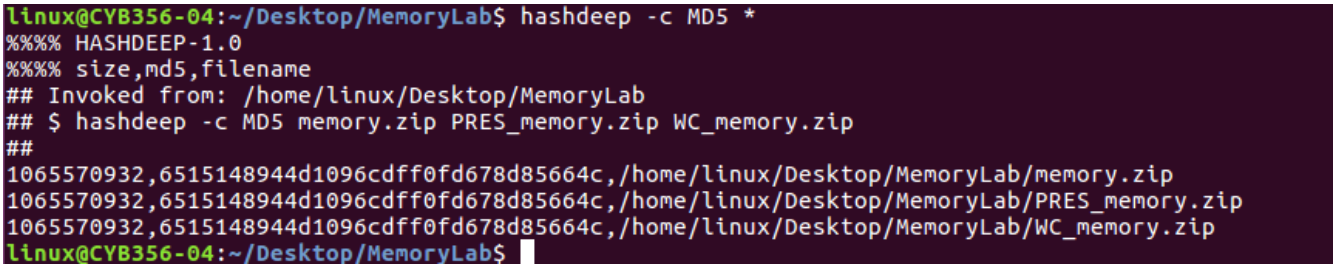


*Figure 2: Hashdeep Output of Evidence*


*Figure 3: Hash Values Provided on Webpage Hosting the .zip File*

At this point the Examiner had successfully downloaded the .zip file using Wget v1.17.1, and successfully created preservation and working copies. The Examiner then continued to unzip the file WC_memory.zip which produced memory.img. This Examiner can infer that all the .zip files contained identical memory.img files since their MD5 hash values were identical. The Examiner can be seen utilizing the unzip command and hashing the file memory.img in *Figure 4*.

```
linux@CYB356-04:~/Desktop/MemoryLab$ sudo unzip WC_memory.zip
Archive:  WC_memory.zip
  inflating: memory.img
linux@CYB356-04:~/Desktop/MemoryLab$ ls
memory.img  memory.zip  PRES_memory.zip  WC_memory.zip
linux@CYB356-04:~/Desktop/MemoryLab$ hashdeep -c MD5 memory.img
%%%% HASHDEEP-1.0
%%%% size,md5,filename
## Invoked from: /home/linux/Desktop/MemoryLab
## $ hashdeep -c MD5 memory.img
##
5368709120,349f6a9bc1efbdc9ca024be701823604,/home/linux/Desktop/MemoryLab/memory.img
linux@CYB356-04:~/Desktop/MemoryLab$
```

*Figure 4: Image File Extraction and Hashdeep v4.4 MD5*

## Analysis

The Examiner began the analysis of memory.img extracted from WC_memory.zip by running the Volatility command: $ **vol.py imageinfo -f memory.img**. This command returned the Examiner with not only the possible Operating Systems, but also the system architecture. The possible Operating Systems can be determined by which of the suggested profiles are available. It can be seen in *Figure 5* that the possible Operating Systems are Windows 7 and Windows 2008R2, and the processor is a 64-bit AMD. The Examiner was unable to indefinitely determine the correct profile through trial and error. For the rest of the analysis the Examiner implemented the –profile=Win7SP1x64 in his Volatility syntax.

```
linux@CYB356-04:~/Desktop/MemoryLab$ vol.py imageinfo -f ~/Desktop/MemoryLab/memory.img
Volatility Foundation Volatility Framework 2.6
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : Win7SP1x64, Win7SP0x64, Win2008R2SP0x64, Win2008R2SP1x64_24000, Win2008R2SP1x64_23418, Win
2008R2SP1x64, Win7SP1x64_24000, Win7SP1x64_23418
                     AS Layer1 : WindowsAMD64PagedMemory (Kernel AS)
                     AS Layer2 : FileAddressSpace (/home/linux/Desktop/MemoryLab/memory.img)
                      PAE type : No PAE
                           DTB : 0x187000L
                          KDBG : 0xf80002e020f0L
          Number of Processors : 1
     Image Type (Service Pack) : 1
                KPCR for CPU 0 : 0xfffff80002e03d00L
             KUSER_SHARED_DATA : 0xfffff78000000000L
           Image date and time : 2015-10-12 11:41:31 UTC+0000
     Image local date and time : 2015-10-12 07:41:31 -0400
linux@CYB356-04:~/Desktop/MemoryLab$
```

*Figure 5: Imageinfo Extracted With Volatility*

The Examiner's next step was to locate any evidence of active web browsers. The Examiner completed this function by implementing the command: **$ vol.py pslist -f ~/Desktop/MemoryLab/memory.img.** When the Examiner executed this syntax, the Examiner was presented with findings that both Google Chrome and Internet Explorer were active in this memory image file. The Examiner's findings can be seen in *Figure 6* below.

```
0xfffffa8006939b10 chrome.exe            3564   2056     31      765      1      1 2015-10-12 11:34:25 UTC+0000
0xfffffa8004639060 chrome.exe            4316   3564      5      166      1      1 2015-10-12 11:34:26 UTC+0000
0xfffffa80067e2060 chrome.exe            3120   3564      8      167      1      1 2015-10-12 11:34:38 UTC+0000
0xfffffa8006b26b10 iexplore.exe           768   2056     16      542      1      0 2015-10-12 11:34:42 UTC+0000
0xfffffa80068cb060 iexplore.exe          4352    768     53      879      1      1 2015-10-12 11:34:43 UTC+0000
0xfffffa8006c6e060 iexplore.exe          3684    768     31      711      1      1 2015-10-12 11:35:03 UTC+0000
0xfffffa8004632060 wordpad.exe           4740   2056      4      140      1      0 2015-10-12 11:35:41 UTC+0000
0xfffffa8006dee060 audiodg.exe           1204    652      8      131      0      0 2015-10-12 11:35:52 UTC+0000
0xfffffa800691eb10 cmd.exe               3320   2056      1       24      1      0 2015-10-12 11:36:03 UTC+0000
0xfffffa800646c060 conhost.exe           3504    696      2       51      1      0 2015-10-12 11:36:03 UTC+0000
0xfffffa8006fb5270 FTK Imager.exe        3460   2056     14      375      1      1 2015-10-12 11:41:01 UTC+0000
linux@CYB356-04:~/Desktop/MemoryLab$ grep cat.txt
```

*Figure 6: Web Browsers Located With Pslist.*

The Examiner continued on to output the results of : **$ vol.py pslist -f ~/Desktop/MemoryLab/memory.img** to a text file named Test.txt. Once Test.txt was created the Examiner then used GREP to locate the Process ID for Google Update, and to locate which process was running on PID 3780. The Examiner can be seen enumerating the desired information by utilizing GREP in *Figure 7*. The Process ID for GoogleUpdate.exe was: **3300**. The software application running on Process ID 3780 was AcroRd32.exe, which was Adobe Acrobat Reader.

```
linux@CYB356-04:~/Desktop/MemoryLab$ grep -i "GoogleUpdate" Test.txt
0xfffffa80055d3060 GoogleUpdate.e         3300   2452      5      134      0      1 2015-10-12 03:57:18 UTC+0000
linux@CYB356-04:~/Desktop/MemoryLab$ grep -i "3780" Test.txt
0xfffffa80042a1b10 AcroRd32.exe           3780   4368      7      318      1      1 2015-10-12 11:05:26 UTC+0000
linux@CYB356-04:~/Desktop/MemoryLab$
```

*Figure 7: GREP Commands Locating Google Update and PID 3780*

The Examiner was tasked with locating the software tool used by Senior Project Manager Alyssa Gleason to extract the memory image file. The Examiner determined that due to the nature of memory acquisitions it was inevitably the last program run. The last program run on the memory image was FTK Imager.exe which was run at 2015-10-12 11:41:01 UTC+0000. The Examiner's results can be located in *Figure 8* below.

```
0xfffffa8006fb5270 FTK Imager.exe         3460   2056     14      375      1      1 2015-10-12 11:41:01 UTC+0000
```

*Figure 8:Memory Extraction Software*

The Examiner proceeded to extract the command history from memory.img by implementing the cmdscan tool located in the Volatility Framework. The syntax used by the Examiner was: **$vol.py cmdscan -f ~/Desktop/MemoryLab/memory.img – profile=Win7SP1x64**. The Results of the cmdscan tool can be seen located in *Figure 9*.

```
linux@CYB356-04:~/Desktop/MemoryLab$ vol.py cmdscan -f ~/Desktop/MemoryLab/memory.img --profile=Win7SP1x64
Volatility Foundation Volatility Framework 2.6
**************************************************
CommandProcess: conhost.exe Pid: 3240
CommandHistory: 0xa2350 Application: TPAutoConnect.exe Flags: Allocated
CommandCount: 0 LastAdded: -1 LastDisplayed: -1
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x60
**************************************************
CommandProcess: conhost.exe Pid: 3504
CommandHistory: 0x2f21a0 Application: cmd.exe Flags: Allocated, Reset
CommandCount: 15 LastAdded: 14 LastDisplayed: 14
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x60
Cmd #0 @ 0x2f0cd0: cd\
Cmd #1 @ 0x2e7810: cd "Program Files (x86)"
Cmd #2 @ 0x2e57e0: cd mandiant
Cmd #3 @ 0x2f0b10: dir
Cmd #4 @ 0x2e5800: cd Memoryze
Cmd #5 @ 0x2f6540: dir
Cmd #6 @ 0x2f6590: f:
Cmd #7 @ 0x2f65a0: cls
Cmd #8 @ 0x2e5820: cd x64
Cmd #9 @ 0x2f65b0: dir
Cmd #10 @ 0x2e7850: memorydd.bat -output E:\
Cmd #11 @ 0x2f65c0: dir
Cmd #12 @ 0x2f1a00: memorydd.bat -o E:\
Cmd #13 @ 0x2e7890: memorydd.bat -output F:\
Cmd #14 @ 0x2f1a30: memorydd.bat -o F:\
```

*Figure 9: History of Commands Executed in memory.img*

Next the Examiner extracted the DLL files from the memory image using the syntax: **$vol.py dlllist -f ~/Desktop/MemoryLab/memory.img – profile=Win7SP1x64 | sudo tee dlllist.txt**. By outputting this file to a text file, the Examiner was able to perform GREP searches locating the desired information. The Examiner can be seen completing this function in *Figure 10* below. In this analysis the Examiner was attempting to locate the absolute path of AVG which was enumerated by implementing GREP as: **$ grep -i "AVG" dlllist.txt**. The Examiners implementation of GREP can be seen in *Figure 11* below. The absolute location of AVG antivirus was shown to be: **C:\ Program Files (x86)\AVG\Av\avgcsrva.exe** which can be seen in *Figure 11*.

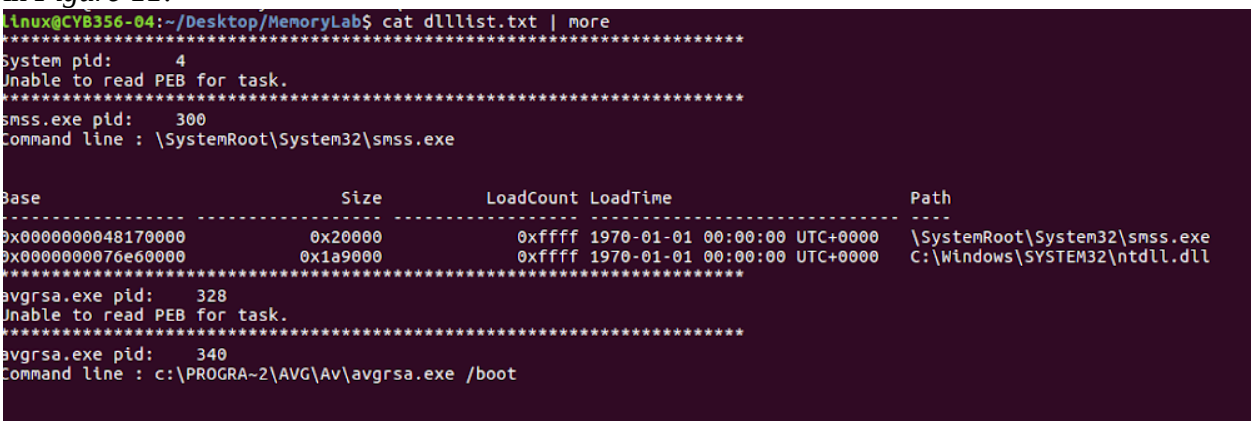


*Figure 10: Output of dlllist.txt Using Cat*

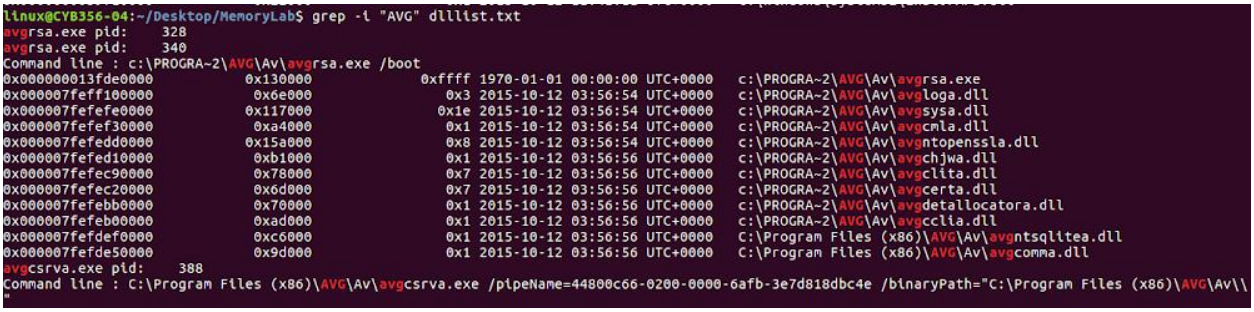


*Figure 11: dlllist.txt Instances of AVG*

The Examiner's final search was attempting to locate the external IP addresses connected to Google Chrome at the time of the memory acquisition. The Examiner began by first outputting the results from the Netscan tool within the Volatility Framework. The output of Netscan can be seen in *Figure 12* below. Next, the Examiner output the results of this Netscan command into a text file named NetTest.txt. After NetTest.txt was populated with the necessary data the Examiner used the GREP tool to extract all instances of the word "chrome" in NetTest.txt.

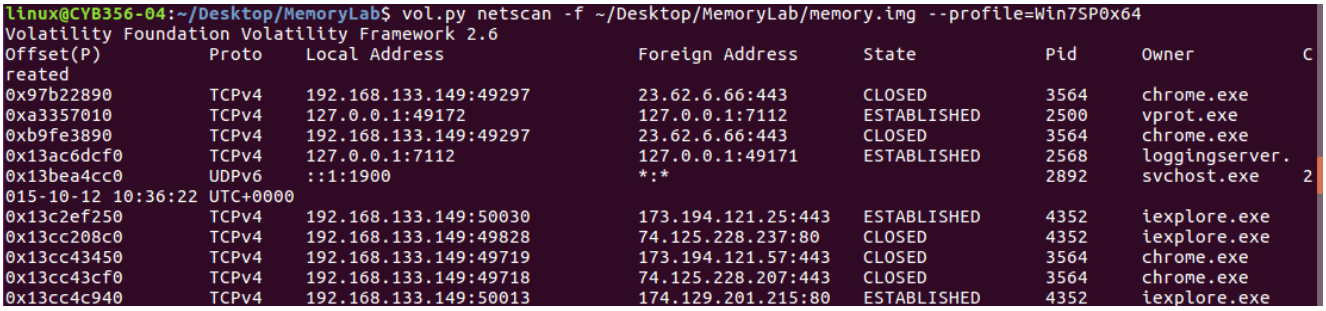


*Figure 12: Netscan Tool Within Volatility Framework*

```
linux@CYB356-04:~/Desktop/MemoryLab$ grep "chrome" NetTest.txt
0x97b22890         TCPv4    192.168.133.149:49297          23.62.6.66:443         CLOSED           3564     chrome.exe
0xb9fe3890         TCPv4    192.168.133.149:49297          23.62.6.66:443         CLOSED           3564     chrome.exe
0x13cc43450        TCPv4    192.168.133.149:49719          173.194.121.57:443     CLOSED           3564     chrome.exe
0x13cc43cf0        TCPv4    192.168.133.149:49718          74.125.228.207:443     CLOSED           3564     chrome.exe
0x13cf92be0        TCPv4    192.168.133.149:49915          74.125.228.206:443     ESTABLISHED      3564     chrome.exe
0x13d05caa0        TCPv4    192.168.133.149:49337          184.168.27.206:80      CLOSED           3564     chrome.exe
0x13d15eae0        TCPv4    192.168.133.149:49916          74.125.228.216:80      ESTABLISHED      3564     chrome.exe
0x13e0809d0        TCPv4    192.168.133.149:49335          184.168.27.206:80      CLOSED           3564     chrome.exe
0x13f1b1a80        TCPv4    192.168.133.149:49336          184.168.27.206:80      ESTABLISHED      3564     chrome.exe
linux@CYB356-04:~/Desktop/MemoryLab$
```

*Figure 13: External IP Addresses Connected to Chrome at Time of Capture.*

In *Figure 13* above it is evident that there were multiple connections to Google Chrome at the time of memory acquisition. Only three of the connections were 'ESTABLISHED' connections marking that they were actively running at the time of memory acquisition. The Examiner enumerated the active Google Chrome Connections at the time of memory acquisition to be **74.125.228.206:443**, **74.125.228.216:80**, and **184.168.27.206:80**.

# Conclusion

Senior Project Manager at Schultz and Dooley Inc. Alyssa Gleason, requests that the Senior Forensic Examiner analyses the previously collected memory image in order to determine if the employee was using a web browser during work hours.

The Senior Forensic Examiner at Schultz and Dooley Inc. has successfully determined that there were three instances of Google Chrome running with an established connection during the time the memory image was captured. The Senior Forensic Examiner is confident that there is enough inculpatory evidence to undoubtably say that the suspect was using a web browser during work hours.

# Appendix

## Appendix A: Examiner Workstation Specifications

- Computer Name: BrianCarrWorkstation
- Operating System (OS) Name: Windows 10
- OS Version: Student Edition
- System Make/Model: MSi GS63VR Stealth Pro
- System Serial Number: K1612N0043395
- Time Zone of Examiner Machine: Eastern Daylight Time (-4:00 GMT)
- System date/time is consistent with the time zone listed above, as verified by http://nist.time.gov/.

## Appendix B: Tools

- Hashdeep v4.4
- Wget v1.17.1
- Volatility Framework v2.6
- Grep (GNU grep) 2.25