**3906ICT/7906ICT Digital Forensics**

**Tutorial 3.2 – Memory Forensics**

The aim of this tutorial is to give you some experience with identifying and examining Windows memory based forensic artefacts. We will cover some of the topics raised in the Lecture, but there are many more Windows memory based forensic artefacts that you are open to explore.

Please note that this tutorial is not a step by step guide. The expectations are that if you are not sure of how to do something, you should find out via internet search or by asking your tutor.

# Preliminaries

There are two options for doing the practical component of this tutorial. You can do this tutorial by logging into the Griffith Cyber Range which is an Internet isolated set of virtual machines that has been set up on the Griffith network. The other is to download and install the software on your local PC.

## Set Up Option 1 – Griffith Cyber Range

If you are not on a Griffith University campus need to VPN into the Griffith Network. Details of how to VPN into the Griffith Network can be found here: https://intranet.secure.griffith.edu.au/computing/remote-access/virtual-private-network. Go to the bottom of the page and find the instructions for your device.

Once you have set up your VPN to the Griffith network, you can use your browser to go to the following page:

https://cyber.ict.griffith.edu.au/

The credentials for the Griffith Cyber Range Server are:

**Username: sXXXXXXX**

**Password: changeme**

sXXXXXXX is your Griffith username. When you log in for the first time change your password (which you will need to remember). To do this go to your username menu on the top right corner of the web page and select the Settings item. The Settings page will allow you to reset your password. Once you have reset the password, use your new password for subsequent logins. For this tutorial we will be using the SIFT workstation. Click on the SIFT link and you will be connected to a virtual machine running the SANS SIFT workstation Linux distribution.

When you have finished your tutorial simply close the browser tab with the connection to the virtual machine. Or press Shift-Ctrl-Alt to access the web menu and disconnect from the Griffith Cyber Range.

## Set Up Option 2 – Install on your local PC

The other option is to install the SIFT workstation on your local PC. Links to the virtual machine OVA file for download are found on the Learning@Griffith web site. **Note:** The SIFT workstation is a 15GB download. You will need to install VirtualBox and select File->Import Appliance to install the SIFT workstation Virtual machine. Start the Virtual Machine and log in.

The login credentials for the SIFT workstation are:

**Username: sansforensics Password: forensics**

# Memory Forensics Scenario

The following is a well-known memory digital forensics example of a capture the flag competition.

The scenario is that your friend Rick who is a bit of an eccentric and alcoholic mad scientist has had some malware infect his PC. You have been asked to investigate while Rick entertains his grandson Morty. Luckily Rick has been able to grab a memory dump of his PC.

# Evidence

Assuming that the correct evidence has been identified and collected, the next step is to download the disk image onto the SIFT workstation virtual machine.

Check Disk space “df -h”

<https://github.com/volatilityfoundation/volatility/wiki/Command-Reference>

1. If you are using the Griffith Cyber Range, your virtual machines are isolated from the Internet but you can download the evidence for this tutorial from [http://forensic-tutorials.griffith.internal](http://forensic-tutorials.griffith.internal/) in the *tutorial3.2* directory. If you are using your own local SIFT workstation you can download the evidence from the link provided in the Learning@Griffith page for this tutorial. It is a 700 Mb download.

**Answer:**

unzip OtterCTF.vmem

md5sum OtterCTF.vmem

1. Create a directory in the /cases directory called OtterCTF and unzip the file here.

**Answer:**

unzip OtterCTF.vmem

md5sum OtterCTF.vmem

1. Check the md5sum of the vmem file is ad51f4ada4151eab76f2dce8dea69868.

**Answer:**

unzip OtterCTF.vmem

md5sum OtterCTF.vmem

ad51f4ada4151eab76f2dce8dea69868 OtterCTF.vmem

1. Familiarize yourself with the volatility framework. What command is required to list the available plugins on the SIFT workstation?

**Answer:**

vol.py -h

-f FILENAME, --filename=FILENAME

--profile=WinXPSP2x86

--plugins=PLUGINS

imageinfo Identify information for the image

1. Let’s examine the vmem file. What is the image profile of the memory dump?

**Answer:**

$ vol.py -f OtterCTF.vmem imageinfo

Volatility Foundation Volatility Framework 2.6.1

INFO : volatility.debug : Determining profile based on KDBG search...

Suggested Profile(s) : Win7SP1x64, Win7SP0x64, Win2008R2SP0x64, Win2008R2SP1x64\_24000, Win2008R2SP1x64\_23418, Win2008R2SP1x64, Win7SP1x64\_24000, Win7SP1x64\_23418

AS Layer1 : WindowsAMD64PagedMemory (Kernel AS)

AS Layer2 : FileAddressSpace (/home/sansforensics/Downloads/Tutorial 3.2/OtterCTF.vmem)

PAE type : No PAE

DTB : 0x187000L

KDBG : 0xf80002c430a0L

Number of Processors : 2

Image Type (Service Pack) : 1

KPCR for CPU 0 : 0xfffff80002c44d00L

KPCR for CPU 1 : 0xfffff880009ef000L

KUSER\_SHARED\_DATA : 0xfffff78000000000L

Image date and time : 2018-08-04 19:34:22 UTC+0000

Image local date and time : 2018-08-04 22:34:22 +0300

First guess: Win7SP1x64, Windows 7 Service Pack 1 64-bit

vol.py -f OtterCTF.vmem --profile=Win7SP1x64 pslist

For vmsn file vol.py sould be run in the same directory, nmsn holds all metadata

OR

Combine the vmem and vmsn file into a raw memory dump ( list of 1s and 0s )

1. List the available registry hives.

**Answer:**

vol.py -h

hivelist Print list of registry hives.

vol.py -f OtterCTF.vmem --profile=Win7SP1x64 hivelist

Volatility Foundation Volatility Framework 2.6.1

Virtual Physical Name

------------------ ------------------ ----

0xfffff8a00377d2d0 0x00000000624162d0 \??\C:\System Volume Information\Syscache.hve

0xfffff8a00000f010 0x000000002d4c1010 [no name]

0xfffff8a000024010 0x000000002d50c010 \REGISTRY\MACHINE\SYSTEM

0xfffff8a000053320 0x000000002d5bb320 \REGISTRY\MACHINE\HARDWARE

0xfffff8a000109410 0x0000000029cb4410 \SystemRoot\System32\Config\SECURITY

0xfffff8a00033d410 0x000000002a958410 \Device\HarddiskVolume1\Boot\BCD

0xfffff8a0005d5010 0x000000002a983010 \SystemRoot\System32\Config\SOFTWARE

0xfffff8a001495010 0x0000000024912010 \SystemRoot\System32\Config\DEFAULT

0xfffff8a0016d4010 0x00000000214e1010 \SystemRoot\System32\Config\SAM

0xfffff8a00175b010 0x00000000211eb010 \??\C:\Windows\ServiceProfiles\NetworkService\NTUSER.DAT

0xfffff8a00176e410 0x00000000206db410 \??\C:\Windows\ServiceProfiles\LocalService\NTUSER.DAT

0xfffff8a002090010 0x000000000b92b010 \??\C:\Users\Rick\ntuser.dat

0xfffff8a0020ad410 0x000000000db41410 \??\C:\Users\Rick\AppData\Local\Microsoft\Windows\UsrClass.dat

1. Dump the registry hives so you can analyse the registry with rip.pl later.

**Answer:**

$ mkdir RicksRegistry

vol.py -f OtterCTF.vmem --profile=Win7SP1x64 dumpregistry -D RicksRegistry

1. What is the PC name and IP address?

**Answer:**

$ vol.py -f OtterCTF.vmem --profile=Win7SP1x64 netscan

0x7fb9cec0 UDPv4 192.168.202.131:1900 \*:\* 164 svchost.exe 2018-08-04 19:28:42 UTC+0000

0x7fb9d430 UDPv4 127.0.0.1:58341 \*:\* 164 svchost.exe 2018-08-04 19:28:42 UTC+0000

rip.pl -h

rip.pl -l | less

rip.pl -l | grep -5 winver

36. winver v.20200525 [Software]

rip.pl -l | grep -5 comp

244. compname v.20090727 [System]

- Gets ComputerName and Hostname values from System hive

cd RicksRegistry/

ls

registry.0xfffff8a000024010.SYSTEM.reg

$ rip.pl -r registry.0xfffff8a000024010.SYSTEM.reg -p compname

Launching compname v.20090727

compname v.20090727

(System) Gets ComputerName and Hostname values from System hive

ComputerName = WIN-LO6FAF3DTFE

TCP/IP Hostname = WIN-LO6FAF3DTFE

Google: registry computername key

<https://www.technlg.net/windows/computername-registry-key/>

# $ vol.py -f OtterCTF.vmem --profile=Win7SP1x64 printkey -K "ControlSet001\Control\ComputerName\ComputerName"

# Volatility Foundation Volatility Framework 2.6.1

# Legend: (S) = Stable (V) = Volatile

# ----------------------------

# Registry: \REGISTRY\MACHINE\SYSTEM

# Key name: ComputerName (S)

# Last updated: 2018-06-02 19:23:00 UTC+0000

# Subkeys:

# Values:

# REG\_SZ : (S) mnmsrvc

# REG\_SZ ComputerName : (S) WIN-LO6FAF3DTFE

# Running processes

1. What processes are running on the PC?

**Answer:**

$ vol.py -f OtterCTF.vmem --profile=Win7SP1x64 pslist

1. Rick just loves to play some good old videogames. Can you tell which game is he playing? What’s the IP address of the server?

**Answer:**

$ vol.py -f OtterCTF.vmem --profile=Win7SP1x64 netscan

LunarMS.exe

$ vol.py -f OtterCTF.vmem --profile=Win7SP1x64 netscan | grep "LunarMS"

Volatility Foundation Volatility Framework 2.6.1

Offset(P) Proto Local Address Foreign Address State Pid Owner Created

0x7d6124d0 TCPv4 192.168.202.131:49530 77.102.199.102:7575 CLOSED 708 LunarMS.exe

0x7e413a40 TCPv4 -:0 -:0 CLOSED 708 LunarMS.exe

0x7e521b50 TCPv4 -:0 -:0 CLOSED 708 LunarMS.exe

1. We know that the account was logged in to a channel called Lunar-3. What is the account username? (Hint: You don’t need to use Volatility. Use the strings command to look in the memory file.)

**Answer:**

$ strings OtterCTF.vmem | grep username

$ strings OtterCTF.vmem | grep username | less

inside the less “/lunar”

<http://lunarms.zapto.org/username0tt3r8r33z3passwordhttp://lunarms.zapto.org/>

1. From a little research we found that the username of the logged-on character is always after this signature: 0x5a 0x0c 0x00 0x00 0x4d. What’s Rick’s character’s name?

**Answer:**

We need to extract the lunarMS program

Find process id of the lunar program

$ vol.py -f OtterCTF.vmem --profile=Win7SP1x64 pslist | grep -i lunar

Volatility Foundation Volatility Framework 2.6.1

0xfffffa801b5cb740 LunarMS.exe 708 2728 18 346 1 1 2018-08-04 19:27:39 UTC+0000

$ vol.py -f OtterCTF.vmem --profile=Win7SP1x64 memdump -p 708 -D RicksRegistry/memoryDump/

$ cd RicksRegistry/memoryDump/

Ls

708.dmp

\*open Bless\*

\*Open 708.dmp\*

\*Search: “0x5a 0x0c 0x00 0x00 0x4d“ \*

DID NOT WORK

man xxd

$ xxd -u 708.dmp | grep -6 "5A0C 0000 4D"

20b05fc0: B0E5 AF00 5A0C 0000 4D30 7274 794C 304C ....Z...M0rtyL0L

# Passwords

1. Silly Rick always forgets his email’s password, so he uses a Stored Password Services online to store his password. He always copies and pastes the password so he will not get it wrong. What’s Rick’s email password?

**Answer:**

copies and pastes 🡪 clip board

$ vol.py -f OtterCTF.vmem --profile=Win7SP1x64 clipboard

Volatility Foundation Volatility Framework 2.6.1

Session WindowStation Format Handle Object Data

---------- ------------- ------------------ ------------------ ------------------ --------------------------------------------------

1 WinSta0 CF\_UNICODETEXT 0x602e3 0xfffff900c1ad93f0 M@il\_Pr0vid0rs

1 WinSta0 CF\_TEXT 0x10 ------------------

1 WinSta0 0x150133L 0x200000000000 ------------------

1 WinSta0 CF\_TEXT 0x1 ------------------

1 ------------- ------------------ 0x150133 0xfffff900c1c1adc0

1. What is Rick’s PC password?

**Answer:**

$ vol.py -f OtterCTF.vmem --profile=Win7SP1x64 lsadump

Volatility Foundation Volatility Framework 2.6.1

DefaultPassword

0x00000000 28 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 (...............

0x00000010 4d 00 6f 00 72 00 74 00 79 00 49 00 73 00 52 00 M.o.r.t.y.I.s.R.

0x00000020 65 00 61 00 6c 00 6c 00 79 00 41 00 6e 00 4f 00 e.a.l.l.y.A.n.O.

0x00000030 74 00 74 00 65 00 72 00 00 00 00 00 00 00 00 00 t.t.e.r.........

DPAPI\_SYSTEM

0x00000000 2c 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ,...............

0x00000010 01 00 00 00 36 9b ba a9 55 e1 92 82 09 e0 63 4c ....6...U.....cL

0x00000020 20 74 63 14 9e d8 a0 4b 45 87 5a e4 bc f2 77 a5 .tc....KE.Z...w.

0x00000030 25 3f 47 12 0b e5 4d a5 c8 35 cf dc 00 00 00 00 %?G...M..5......

# Identify Malware

1. The reason that we took Rick’s PC memory dump is because there was a malware infection. Please find the malware process name (including the extension).

**Answer:**

$ vol.py -f OtterCTF.vmem --profile=Win7SP1x64 pslist | less

0xfffffa801b486b30 Rick And Morty 3820 2728 4 185 1 1 2018-08-04 19:32:55 UTC+0000

$ vol.py -f OtterCTF.vmem --profile=Win7SP1x64 pstree | less

Parent and child

0xfffffa801b27e060:explorer.exe 2728 2696 33 854 2018-08-04 19:27:04 UTC+0000

. 0xfffffa801b486b30:Rick And Morty 3820 2728 4 185 2018-08-04 19:32:55 UTC+0000

.. 0xfffffa801a4c5b30:vmware-tray.ex 3720 3820 8 147 2018-08-04 19:33:02 UTC+0000

1. What is the message embedded in the associated malware files that shows the malware was specifically for Rick? **Answer:**

$ vol.py -f OtterCTF.vmem --profile=Win7SP1x64 filescan | less

$ vol.py -f OtterCTF.vmem --profile=Win7SP1x64 filescan | grep -i "Rick and Morty"

Volatility Foundation Volatility Framework 2.6.1

0x000000007d63dbc0 10 0 R--r-d \Device\HarddiskVolume1\Torrents\Rick And Morty season 1 download.exe

0x000000007d6b3a10 11 1 R--rw- \Device\HarddiskVolume1\Torrents\Rick and Morty - Season 3 (2017) [1080p]\Rick.and.Morty.S03E07.The.Ricklantis.Mixup.1080p.Amazon.WEB-DL.x264-Rapta.mkv

0x000000007d7adb50 17 1 R--rw- \Device\HarddiskVolume1\Torrents\Rick and Morty - Season 3 (2017) [1080p]\Rick.and.Morty.S03E06.Rest.and.Ricklaxation.1080p.Amazon.WEB-DL.x264-Rapta.mkv

0x000000007d8813c0 2 0 RW-rwd \Device\HarddiskVolume1\Users\Rick\Downloads\Rick And Morty season 1 download.exe.torrent

0x000000007da56240 2 0 RW-rwd \Device\HarddiskVolume1\Torrents\Rick And Morty season 1 download.exe

0x000000007dae9350 2 0 RWD--- \Device\HarddiskVolume1\Users\Rick\AppData\Roaming\BitTorrent\Rick And Morty season 1 download.exe.1.torrent

0x000000007dcbf6f0 2 0 RW-rwd \Device\HarddiskVolume1\Users\Rick\AppData\Roaming\BitTorrent\Rick And Morty season 1 download.exe.1.torrent

0x000000007e5f5d10 3 1 R--rw- \Device\HarddiskVolume1\Torrents\Rick and Morty Season 2 [WEBRIP] [1080p] [HEVC]\[pseudo] Rick and Morty S02E03 Auto Erotic Assimilation [1080p] [h.265].mkv

0x000000007e710070 8 0 R--rwd \Device\HarddiskVolume1\Torrents\Rick And Morty season 1 download.exe

0x000000007e7ae700 3 1 R--rw- \Device\HarddiskVolume1\Torrents\Rick and Morty Season 2 [WEBRIP] [1080p] [HEVC]\Sample\Screenshot 08.png

$ mkdir RicksRegistry/Q16

$ vol.py -f OtterCTF.vmem --profile=Win7SP1x64 dumpfiles -D RicksRegistry/Q16/ -Q 0x000000007dae9350

$ cd RicksRegistry/Q16

$ ls

file.None.0xfffffa801b42c9e0.dat

--------

$ vol.py -f OtterCTF.vmem --profile=Win7SP1x64 dumpfiles -D RicksRegistry/Q16/ -Q 0x000000007da56240

$ cd RicksRegistry/Q16

$ ls

$ strings file.None.0xfffffa801b42c9e0.dat

d8:announce44:udp://tracker.openbittorrent.com:80/announce13:announce-listll44:udp://tracker.openbittorrent.com:80/announceel42:udp://tracker.opentrackr.org:1337/announceee10:created by17:BitTorrent/7.10.313:creation datei1533150595e8:encoding5:UTF-84:infod6:lengthi456670e4:name36:Rick And Morty season 1 download.exe12:piece lengthi16384e6:pieces560:\I

!PC<^X

B.k\_Rk

0<;O87o

!4^"

3hq,

&iW1|

K68:o

w~Q~YT

$$o9p

bwF:u

e7:website19:M3an\_T0rren7\_4\_R!cke

# Tutorial Quiz

You have now completed the exercises for this tutorial. You can now attempt the quiz for this tutorial.