IT Forensics – Week 8

Memory Forensics

Overview

- Memory Forensics
- Volatility Framework

Memory Forensics

- We have seen how to take a memory capture and the large amount of data stored could be considered to be a unorganized data blob.
- The memory capture does have a structure and this can be analysed using digital forensic tools.
- As previously discussed, the processes and files currently open on the computer are residing in memory.

Memory Analysis

- When working with a computer the data must be in its real form when it resides in memory.
- This may not always be true due to memory encryption techniques but the majority of items are in their true form in memory.
- Malware, encryption etc will be in it true unencrypted form. This offers the opportunity to analyse what we find to make snese of its operation and view its content.

Processes

- In the RAM capture we will be able to get a list of the processes currently running in memory.
- This is a list of the OS processes and the process of the applications opened by the user.
- We can get the PID and PPID of the processes and the process name.
- When looking at malware and intrusions we will be looking at processes to see what is currently happening. (Trying yo identify hijacked processes)
- Analysing processes is also a rich source of information to see what applications are open and what tasks they are performing etc

Forensic Techniques

- Looking at using digital forensic tools to find interesting things in memory.
- For a digital investigator they would need to be very familiar of the operating system files and structure of a clean machine. Why?
- By knowing what the common files are and their names / ports etc we can eliminate items from our investigation.
- **Whitelisting** can be used as a technique to use a set of hash values of files from a clean machine to see if they match the files on our target machine.

- Another approach to this is **Indicators of compromise (IoC)**, these are definitions of past incidences that are incorporated into a software tool that analyses the system to try match patterns etc.

Memory Tools - Volatility

- https://www.volatilityfoundation.org/
- “In 2007, the first version of The Volatility Framework was released publicly at Black Hat DC. The software was based on years of published academic research into advanced memory analysis and forensics. Up until that point, digital investigations had focused primarily on finding contraband within hard drive images. Volatility introduced people to the power of analyzing the runtime state of a system using the data found in volatile storage (RAM). It also provided a cross-platform, modular, and extensible platform to encourage further work into this exciting area of research. Another major goal of the project was to encourage the collaboration, innovation, and accessibility to knowledge that had been common within the offensive software communities.” - https://www.volatilityfoundation.org/about

Volatility

- “Volatility development is now supported by The Volatility Foundation, an independent 501(c) (3) non-profit organization. The foundation was established to promote the use of Volatility and memory analysis within the forensics community, to defend the project's intellectual property (trademarks, licenses, etc.) and longevity, and, finally, to help advance innovative memory analysis research.” - https://www.volatilityfoundation.org/about

Volatility – Basic Usage

- **Typical command components:**
  - vol.py -f [image] --profile=[profile] [plugin]
- **Display profiles, address spaces, plugins:**
  - vol.py –info
- **Display global command-line options:**
  - vol.py --help

Volatility – Basic Usage

- **Load plugins from an external directory:**
  - vol.py –f example.mem --plugins=[path] [plugin]
  - Where example.mem is the RAM capture. We will use this for the remainder of our examples

Volatility – Working with the Image

- **Identify the specific profile (operating system version)**
  - vol.py –f example.mem imageinfo

Volatility – Working with processes

- **Get the process list**
  - Vol.py –f example.mem --profile=WinXP pslist
- **Identify hidden processes**
  - Vol.py –f example.mem --profile=WinXP pscan
- **Cross reference processes with various lists:**
  - Vol.py –f example.mem --profile=WinXP psxview
- **Process tree**
  - Vol.py –f example.mem --profile=WinXP pstree
- Where WinXP is the profile for the RAM capture

Volatility – Working with processes

- **Show command line arguments:**
  - Vol.py –f example.mem --profile=WinXP cmdline
- **Dump all valid pages to a single file:**
  - Vol.py –f example.mem --profile=WinXP memdump –p 2001 --dump-dir=PATH
  - Where 2001 is the process id and PATH is the path on your machine
- **Dump a process:**
  - Vol.py –f example.mem --profile=WinXP procdump –p 2001 --dump-dir=PATH
  - Where 2001 is the process id and PATH is the path on your machine

Volatility – Logs Histories (Recover command history)

- **cmdscan and consoles**
  - Vol.py –f example.mem --profile=WinXP cmdscan
  - Vol.py –f example.mem --profile=WinXP cmdline

- **Recover IE cache/Internet history:**
  - Vol.py –f example.mem --profile=WinXP iehistory –p 2001
  - Where 2001 is the process id
- note there are other plugins we can get for other browsers

Volatility – Networking Information

- **Check TCP connections**
- Vol.py –f example.mem --profile=WinXP connscan

- **Check UDP and TCP connections**
- Vol.py –f example.mem --profile=WinXP sockets

- **Current versions of Windows**
  - Vol.py –f example.mem --profile=WinXP netscan

Volatility – Registry

- **Display cached hives:**
  - ./vol.py -f example.mem --profile=WinXP hivelist

- **Print a key's values and data:**
  - ./vol.py -f example.mem --profile=WinXP printkey -K "Software\Microsoft\Windows\CurrentVersion\Run" | more
  - Where "Software\Microsoft\Windows\CurrentVersion\Run" is a reg entry

Volatility – Strings

- We can use strings to parse a memory or process dump.
- Ie. we need the dump file first.

- Strings - We will use a program named strings to parse the memory process dump. This scans file for unicode or ascii content. This should(might) be preinstalled on Linux and can be downloaded for Windows (from Microsoft).
  - strings example.dmp
  - Where example.dmp is our mempry/process dump

Identifying Malware with Volatility

- We will demo this in our lecture session this week.

Questions