|  |  |
| --- | --- |
| **Digital Forensics**  Diploma in CSF/IT  Year 3 (2022/23) Semester 4/6 | Week 3-17 |
|  |
| **Write-up on Open Source Forensic Tools - Individual** | |

|  |  |  |
| --- | --- | --- |
| Student No. | Student Name | Tutorial Group |
| S10222894 | Dominic Lee | P02 |

*Text

Description automatically generated*

Note:

You are required to demonstrate the above tools during week 15. Change of tools will only be allowed if it is approved by your tutor.

1. **Name and description of open-source tools**
   1. **Volatility**

­­­Volatility is used even by security professionals to analyze the runtime state of a system using the data found in volatile memory dumps.

It is a cross-platform software meaning that it can be ran on different OS (Operating System) such as Windows and Linux, it is modular signifying simple use of commands and an extensible platform which allows you to extract information about network connections, running processes, registry hive and many more.

* + 1. **Functionalities**

**Link to commands and documentation**

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

**Installing**

* **git clone** <https://github.com/volatilityfoundation/volatility3.git>
* pip3 install -r requirements.txt
* python3 vol.py -h

**Commands**

Here we will look at some of the more notable commands in volatility that might be worth using during memory analysis of the RAM (Random Access Memory) dump. Note that the commands used here are meant for files on Windows. However, as the OS that we are using is Kali Linux, replace all “windows.command” with “linux.command.”

To learn how to use the different plugins, simply use -h after the command to see its different parameters. e.g., /vol.py -f ~/memdump.mem windows.pslist -h.

1. **pslist** à **command to list processes**

This command shows the processes that were running when the memory dump was captured. This is an effective way of identifying any malware or telling processes that may be running on the device.

Syntax:

* **python3 vol.py -f <filename> windows.pslist**
* **python3 vol.py -f <filename> windows.pslist | less**
* **python3 vol.py -f <filename> windows.pslist > output.txt**

**Second pointer will be less detailed and have lesser amount of data to be interpreted and third pointer will output all the results into a text file called “output.txt”.**

**Graphical user interface, text

Description automatically generated**

Above is a sample output of the command. Here, PID (Process ID) = Process ID number, PPID = Parent process ID number, ImageFileName = Name of the running process, Offset = Hex value representing the location in memory the process is running, CreateTime = Time process started, ExitTime = Time process ended.

This command is useful in identifying the different processes running on the computer as we can identify suspicious processes and better understand what the suspect was looking at and thinking.

1. **pstree à command to identify child processes of processes**

This command allows you to easily see what process spawned another process which helps in identifying suspicious process activity. Furthermore, it lets you spot malicious processes masquerading as legitimate processes.

**A picture containing text, plaque, screenshot

Description automatically generated**

The screenshot above shows that process “System” has spawned another process “smss.exe” which has spawned another “smss.exe” which has spawned “winlogon.exe”.

1. **netscan à Command to identify network connections**

**The network connections of the computer are also accounted for during a RAM dump. These connections can be useful as any malicious network connections can be identified such as the source port and the process that the network activity relates to.**

**Using this we can see if the suspect has had any connection with other computers that he wanted to hide.**

**Graphical user interface, text

Description automatically generated**

From the above image, offset = location in memory, proto = network protocol used by process, LocalAddr = Source address of network connection, LocalPort = Source port of network connection, ForeignAddr = Destination address of network connection, ForeignPort = Destination port of network connection, State = State of network connection (close, listening, established), PID = Process ID of associated process, Owner = Account associated with process, and Created = Time network connected was initiated.

1. **handles à Command to identify handles for processes**

This command allows us to see the different handles of processes in memory which we can use to identify suspicious processes.

**Graphical user interface, application

Description automatically generated with medium confidence**

1. **dumpfile à Command to dump a file**

This command allows you to dump files which allows you to better analyze the file or simply export important files that need to be exported.

**Graphical user interface, text, application

Description automatically generated**

In the above picture, -o represents the output directory, pid refers to the file with a PID of 1328 and virtual address of a file can be found using the handles command.

1. **hashdump à command to see password hashes**

This command allows you to see the password hashes for the system. By obtaining the password hashes, we can try and crack these hashes. Furthermore, if the suspect uses the same password for other services, this will exponentially increase the value of cracking the password hashes listed out in hashdump.

Graphical user interface, application

Description automatically generated

Above, we see the User John Doe with RID of 1001 and the lmhash and nthash value of his password.

1. **userassist à Command to show user activity in the system**

This command is useful in establishing user behavior to how many times a user has used a program and for how long. This gives us a better understanding of what the user was thinking, allowing us to make better inferences on what happened.

Graphical user interface, text, application

Description automatically generated

In the screenshot above, Last Write is the last time an action took place meaning if the Hive is a program, then it might be the last time the suspect ran the program, Count is the number of times the program was ran, Focus Count is the number of times the suspect focused on the program and Time Focused is the duration.

Graphical user interface, text

Description automatically generated

Here we see that the software MS Edge was opened 3 times, focused on 16 times, and had a total time spent on it of 1 hour 50 minutes 56 seconds.

* + 1. **Similar Tools**

**BlackLight**

BlackLight provides details of user actions and reports of memory image analysis, it efficiently organizes different memory locations to find the traces of potentially important user activities. Furthermore, it can analyze several types of memory files including hibernation files, pagefile.sys, raw dumps and crash dumps. It is also able to perform bulk extraction content search for various items such as URLs and addresses. Lastly, it is much faster than other open-source tools and can analyze data from the 4 major platforms, (Android, Windows, iOS and MacOS).

**SANS SIFT**

Sans Sift is known for performing in-depth forensic or incident response investigation. It supports a variety of evidence formats including Advanced Forensic format (AFF), RAW (dd) evidence formats and Expert Witness Format for analysis. It has a user-friendly interface and includes greats tools such as Rifiuti for examining the recycle bin, Scalpel for data file carving to name a few.

* + 1. **Reasons**

All 3 tools listed are great tools to use for memory forensic analysis. I chose volatility and not the other 2 tools for the following reasons. Firstly, BlackList requires a subscription which immediately disqualifies it. Comparing volatility and SIFT, I chose volatility as it has one of the largest communities and is well documented though I believe SIFT and Volatility have similar analyzing capacities.

* 1. **The Sleuth Kit (TSK)**

The Sleuth Kit was originally a C library and collection command line file and volume system forensic analysis tools. It allowed you to examine file systems of a suspect computer in a non-intrusive fashion. Additionally, as the tools do not rely on the operating system to process the file systems, deleted and hidden content is shown. It can be run on Windows and Unix platforms.

The volume system (media management) tools allow you to examine the layout of disks and other media. The Sleuth Kit supports DOS partitions, BSD partitions (disk labels), Mac partitions, Sun slices (Volume Table of Contents), and GPT disks. Using these tools, you can identify where partitions are located and extract them so that they can be analysed with file system analysis tools.

* + 1. **Functionalities**

Below are all the commands available in TSK. We will take a greater look into the commands that might be useful.

Text

Description automatically generated

Text

Description automatically generated

* This is to view the help menu in TSK. To do it, type the command name without entering any other parameter. Through this, we are better able to learn the commands.

1. **img\_stat**

Graphical user interface, text

Description automatically generated

This command lets us determine the image type and size. This

is helpful at the start of the analysis to better understand the image we are working with (in this case, a raw disk image).

A Raw Disk Image contains an exact, sector-by-sector copy of a single partition or disk. A Raw Disk Image of a disk or a partition is a larger file than a Backup Disk Image of the same disk or partition and it takes a longer time to create.

1. **mmls**

**Text

Description automatically generated with medium confidence**

Here we can see the sector size of the disk (512 bytes), and the offset sector. We also see how the different portions of the disk, in this case, Primary Table is the partition table, unallocated is the unallocated space on the disk, and Win95 FAT32 which is a partition. This is useful in telling us how many partitions there are (in this case 1), the file systems that the partitions are using and lastly, the start point of the partitions that you want to analyze.

Using this command, we are also able to identify whether the disk image is physical or logical. If there is an error or only shows a partition, it is logical otherwise it is a physical disk image. As the computer would not be able to parse out the partition table, the image should be logical (part of a disk).

This allows us to be more organized in our analysis as we can map a detailed view of the organization of the disk we are analyzing.

1. **fsstat**

**Text

Description automatically generated**

This command gives us details about the Win95 FAT32 partition. By giving the -o 11264, we specify that we want to analyze the location at an offset of 11264, the start of the Win95 FAT32 sector. From here, we are given details of the sector such as sector size, cluster size, file system type etc.

With a better understanding of the partition or disk we are analysing, we are able to have greater analysis done on the disk as certain information found might require knowledge of the type of file system being analysed.

1. **fls**

**Text

Description automatically generated**

Here we use this command to list the files and directories of the Win95 FAT32 partition. Using this, we can view the different directories and files which might give us evidence and useful information.

Graphical user interface, text

Description automatically generated

Without specifying a number, fls will list out all files and directories in the root directory. By specifying 25 at the end, we will look in the directory called winMD5SumPortable. Through this, we can view the files in the directory. Note that we can recursively look through the directories to see all files in the system (-r).

* + 1. **Similar Tools**

**X-Ways Forensics**

X-Ways Forensics is a work environment for computer forensic examiners and runs under all versions of Windows OS. X-Ways is more efficient to use, not as resource hungry, and runs much faster than other tools available. Furthermore, it has simple hardware requirements and does not depend on setting up a complex database while also being fully portable, being able to run off a USB stick.

* + 1. **Reasons**

X-Ways forensics requires payment and only works on Windows OS. Also, The Sleuth Kit has great documentation on its commands and has a wide array of features making it a good starting point for forensics investigation.

1. **Demonstration**
   1. **Case Description**

You are a forensics investigator for the police and have been contacted regarding an ongoing investigation.

Recently, there has been a murder of a girl named Alice Loh who was stabbed multiple times with bruises on her body. She was found in a home of a man named Steven Yeo who is currently missing. Police suspect that this to be a case of rape and murder.

Upon interviewing the relevant people for the case, you find from Alice’s mother that Alice has been extremely anti-social over the last week, being on her phone for long durations of time, refusing to share her life with her parents.

Suspecting that there is technology involved in the case, you seize the electronic devices on site which includes a laptop that is believed to belong to Steven Yeo. The laptop was found to be in screen saver on arrival, with no password. Hence you decided to take a memory dump of the state of the laptop seeing that processes such as Skype were running. Following this, you created a disk image of the laptop using a write-blocker. This was all done after determining that there were no attempts to destroy any of the files (evidence) in the laptop.

Following, you bagged and tagged all the relevant devices to start the chain of custody before sending them back to the lab for analysis. The evidence bound for analysis includes a memory dump of the state of the laptop found at the crime scene as well as a disk image of the laptop’s hard disk.

* 1. **Source of evidence files**

I intend to download the memory dump and evidence file (disk image) from the internet. If there are no suitable memory dumps and evidence files from the internet, then I will create the evidence myself. The suitability of the evidence is subjective to me, but I will determine whether the processes and files in the evidence files can be linked to the case scenario which I have described.

To simulate the forensic workstation, I will be using a Kali virtual machine to analyze the evidence file and memory dump of the suspect’s machine.

Memory Samples & Evidence files to look at:

* <https://github.com/volatilityfoundation/volatility/wiki/Memory-Samples>
* <http://www.digitalforensicsassociation.org/evidence-files/>
  1. **Source of evidence files**

Should I be unable to find any suitable evidence files, I will then create the disk image and memory dump by myself.

Regarding the memory dump, I will take a memory dump of my own computer system, I will make a note to include some processes or files which are suspicious (e.g., incriminating/malicious).

Like the memory dump, I will also modify my computer system to have incriminating evidence and take a forensic image of it using tools such as FTK image.

An example of incriminating evidence would be a text file that had a password locked on it in the system which was called Alice\_Location.txt.

* 1. **Physical setup (with a list of resources) for the demo**

**Simulated tools**

1. Evidence files
2. Forensic workstation

**Tools Used at the crime scene**

1. Write-blocker
2. Live response tools
3. Large capacity storage drive
4. Drive adapters
5. Digital camera
6. Notepads, pens
7. Anti-static wrist straps
8. Toolkits for disassembly
9. Evidence bags
10. Anti-static bags
11. Latex gloves

The write-blocker is used to safely create an image of the suspect’s disk by preventing the workstation from writing to the suspect’s disk which would corrupt the integrity of the evidence.

The digital camera, notepads and pens are used to document the initial state of the scene such as mouse position etc. It is also used to capture live data on scenes.

Anti-static wrist straps are used to prevent damage to electronic devices when handling them and toolkits for disassembly are when devices need to be disassembled e.g., workstations.

Lastly, latex gloves are used to prevent contamination of evidence, anti-static bags and evidence bags are used to store all evidence and allow for tagging which is when the chain of custody begins.

**Tools used in the lab**

1. Volatility
2. The Sleuth Kit (TSK)

The evidence collected consists of a memory dump of the live system at the scene and the disk image of the laptop seized also at the scene. As such, we will be analyzing both using different tools as there could be valuable information in them. The forensic software which will be used to analyze the evidence files are stated above

Volatility is a CLI (Command Line Interface) memory forensics tool that can be used on Deian-based Linux distributions. Using volatility, we can analyze memory dumps and retrieve useful information such as running processes, last file modified, user’s browser history and more, all of which are stored in the memory of the computer.

The Sleuth Kit is a CLI tool that uses a C library to analyze disk images and recover files from them. It can be used for many purposes though its main one is forensics. Using it we will be able to recover deleted image files, perform keyword searching, and understand what data is stored on a disk drive, even if the operating system has removed all meta data.

**Setup for demo**

To setup for the demonstration, only a laptop and virtual machine are needed but it must contain the downloaded evidence files, as well as the software tools used in the forensic lab.

* 1. **Diagrams**
* Below is the flowchart for how I will prepare to do the demonstration if there are suitable evidence files that can be downloaded online.
* Below is the flowchart for what happens if I create and use my own evidence files.

**END**