Journal of Applied Informatics and Computing (JAIC)
Vol.x, No.x, Juli 201x, pp. xx~xx
e-ISSN: 2548-6861

Memory Dump Analysis in Capture The Flag: Using Volatility 3 to Extract Hidden Files

Isnaeni Hari Yusriyah [1]*, Joy Gilbert Arapenta [2]*, Jean Tirstan Tambunan [3]*, Hamdani Arif,S.Pd., M.Sc[4]**
* Cyber Security Engineering Study Program, Politeknik Negeri Batam
** Informatics Engineering Department
isnaenihariyusriyah@gmail.com [1], joygilberta@gmail.com [2], jeant2212@gmail.com [3], hamdaniarif@polibatam.ac.id[4]

Article Info

Article history:

Received …
Revised …
Accepted …

Keyword:

Forensic, CTF, Analisis Memory Volatile, Memory Dump Analysis.

ABSTRACT

The field of study that examines how to uncover, collect, analyze, and present digital evidence from electronic devices is called computer forensics. This research focuses on the analysis of memory dumps in the Capture The Flag (CTF) cybersecurity competition with the aim of uncovering hidden files that may be concealed in memory by an attacker. Conducting analysis on memory dumps is an important technique in digital forensics and security incident investigation to uncover suspicious activities and hidden evidence that is not available on storage media. The Volatility Framework is utilized as the main framework for analyzing memory dumps. The analysis process adopts the general stages of the computer forensics investigation model, including acquisition, analysis, and extraction. Various Volatility plugins and modules, such as imageinfo, pslist, cmdline, filescan, grep, and dumpfiles, are optimized to identify suspicious processes, locations of hidden files, and passwords required to open encrypted files. This research shows that the Volatility Framework is an effective memory forensics tool for extracting important information from memory dumps, including hidden files, which is highly useful in the context of cybersecurity competitions such as Capture The Flag (CTF).

I. Introduction

Memory forensics is an important field in the recovery and analysis of digital evidence from device memory. As cybercrime increases, memory forensics is becoming an increasingly useful tool. Initially, computer forensics investigations focused on analyzing storage media from systems involved in criminal activities. This process involved first shutting down the system, then creating a digital copy (clone) of the storage media, which could then be used as evidence for further investigation [1]. However, the approach of investigators has shifted from previously directly shutting down the evidence computer to prioritizing taking snapshots of volatile data such as memory first before shutting down the system, as it is considered more effective for capturing important temporary information.

Volatile memory analysis plays a very important role in the cybersecurity defense world and Capture the Flag (CTF) competitions to uncover illegal activities and collect digital evidence. Memory forensics is a part of digital forensics that focuses on analyzing and extracting data from computer memory (RAM) for investigative purposes. When an operating system or application is running, they leave specific traces in the memory that can persist for a long time. Forensic analysis on memory can reveal important information such as login credentials, web sessions, deleted files or logs, and processes that were running at the time [2].

One of the challenges often faced in the investigation process is the existence of hidden files concealed by attackers after carrying out their malicious actions. To address this challenge, Volatility can be utilized as a memory forensics analysis framework, requiring an in-depth understanding of the command line to optimize its use. Volatility has the ability to analyze various types of memory dumps on Windows, Linux, and MacOS operating systems and has an extensive library to detect evidence such as malware, open ports, accessed files, and file encryption [3].

e-ISSN: 2548-6861

Volatility 2 was built using Python 2, which is now being phased out [5]. For example, Volatility 2.x provides Windows profiles based on the service version, such as profiles for Windows 7 Service Pack 1 or Windows XP Service Pack 2. This approach is now inadequate because significant changes to data structures occur in operating systems. This causes Volatility developers to have to create new profiles for updates that change the data structure [4]. Meanwhile, Volatility 3 is a complete rewrite of the framework using Python 3 and will be the future replacement [5].

In this research, the latest version of Volatility 3 is used, which now utilizes Python 3. Volatility 3 has become more modern, fast, and efficient, and is able to take advantage of the latest features and libraries. Volatility 3 adopts a modular architecture that facilitates the development and integration of new plugins for specific functions, such as extracting process, network, or file system information from memory dumps. Volatility 3 supports multi-platform, including Windows, Linux, and MacOS, and has an updatable profile system to support various operating systems and the latest kernel versions [6]. Volatility 3 also has more accurate physical and virtual memory mapping capabilities. With good documentation and active usage, Volatility 3 has become a strong and flexible choice for memory forensics analysis.

Capture The Flag (CTF) is a cybersecurity competition that involves forensic challenges. One of the forensic challenges often encountered in Capture The Flag (CTF) is extracting hidden files from provided memory dumps. At the 4th National Polytechnic Informatics Student Competition (KMIPN), there was a challenge in the form of a memory dump, and participants had to analyze it to solve the given challenge. A memory dump is the result of collecting memory data from a device experiencing a system failure. The ability to parse and understand the contents of a memory dump is a skill in the cybersecurity world. This study discusses the Volatility 3 analysis of the Capture The Flag (CTF) challenge from the National Polytechnic Informatics Student Competition (KMIPN) in an effort to extract hidden files, so that the performance and capabilities of each version in handling similar cases can be known.

II. Methodology

The stages of this methodology are specifically designed for this research based on the researcher's experience and understanding. This methodology stage is the result of combining and adjusting from various sources that are aligned with the context and objectives of the research. Although there is no single source that covers all stages of this methodology, each step is based on best practices and proven approaches in the fields of memory forensics analysis and Capture The Flag (CTF). The methodology stages applied in this research are illustrated in Figure 1.

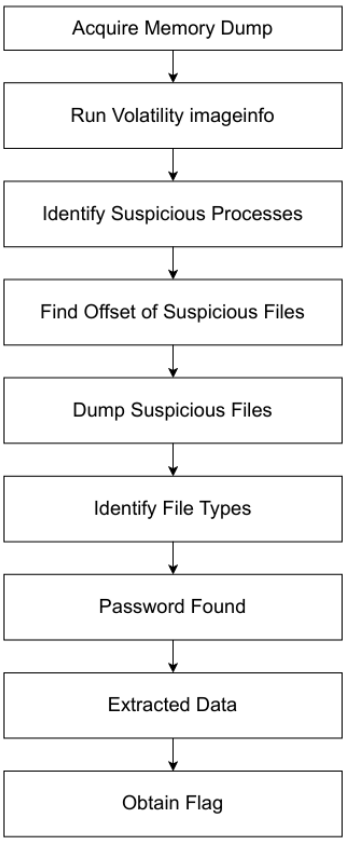

Figure 1. Methodology

A. Acquire Memory Dump Stage

At this stage, the acquisition of digital evidence in the form of a memory dump from the system is carried out while preserving its integrity so that it can be legally accepted [7]. A memory dump is an important source of digital evidence in forensic analysis. The acquisition process produces a .dmp file which is the memory dump from the system. Memory acquisition techniques are performed to obtain a memory dump from the system to be analyzed. The tools used can be special hardware such as FireWire or software tools like DumpIt.

B. Run Volatility imageinfo Stage

At this stage, the imageinfo plugin is used to obtain information about the analyzed memory dump, such as the operating system, service pack, hardware architecture (32-bit or 64-bit), time of the dump, and other relevant information [8]. This information is crucial to ensure the correct use of Volatility plugins and modules that match the characteristics of the analyzed memory dump.

C. Identify Suspicious Process Stage

At this stage, when the memory dump is taken, the pslist and cmdline plugins are used to identify suspicious processes. When the memory dump is taken, the pslist plugin displays a list of processes that were running on the system

at the time the memory dump was taken. This plugin can find ongoing activities in the system [9]. Ligh et al., (2014) state that the pslist plugin provides important information such as process name, PID, PPID, number of threads, number of handles, session ID, and create time [10]. With this information, ongoing activities on the system can be analyzed.

D. Find Offset of Suspicious Files Stage

Before extracting or dumping files from the memory dump, important information is needed such as the physical memory address or Offset where the data is stored. The memory offset is the location where certain data is stored in memory. In this research, to obtain the memory offset, the filescan option and grep command provided by Volatility are used. The filescan plugin is one of the plugins in the Volatility Framework that scans the memory dump to find files stored within it. The filescan plugin is useful for identifying hidden files or deleted files [11].

In addition to using the filescan plugin, this research also utilizes the grep command to assist in the process of searching and identifying memory offsets. The grep command is a powerful text search tool widely used in Unix/Linux environments. In the context of memory forensics, grep is used to search for specific text patterns or strings within the memory dump contents, which can help identify the presence of files or data relevant to the searched pattern.

E. Dump Suspicious Files Stage

After the offset is found, the next stage is dumping files using the dumpfiles option. The dumpfiles plugin from the Volatility Framework is used to extract files stored in the analyzed memory dump. This plugin allows access and collection of important data that may not be available on other storage media. By extracting files that were previously only in memory, critical information such as documents, images, or even malware can be fully obtained for further investigation purposes. The dumpfiles plugin leverages the caching concept in operating systems, where accessed files are temporarily stored in memory to improve system performance [12]. The dumpfile plugin can retrieve files in their entirety, unlike file carving techniques that only extract data fragments without considering how files are mapped in memory.

F. Identify File Types Stage

The identification stage is a very important step in the forensic file examination process because it impacts the subsequent processes and the final investigation results [13]. After successfully extracting suspicious files from the memory dump, the next stage is to identify the type or format of each extracted file. This process is carried out using the file command available on many operating systems. The file command will analyze the header or signature of each file and provide information about its file type or format.

Accurately identifying the file type is crucial in forensic investigations as it will help determine the appropriate analysis steps and tools to use for each file. If a file is identified as an encrypted zip file, additional steps are needed to decrypt the file before further analysis can be performed. From the results of the file command, it will be known that one of the extracted files has a zip format that matches the information on the previously obtained command line (cmdline). This zip file is encrypted and requires a password to be opened and analyzed.

G. Password Found Stage

After identifying the existence of an encrypted zip file, the next step is to find the password to open that file. In some cases, the password may be present in a text file or other related document. Forensic researchers can search for these files in the memory dump or system being analyzed.

If the text file found does not directly contain the flag or information being sought, its contents may provide clues, keywords, or passwords to open other encrypted files. By using the discovered password, the encrypted file can be opened and further analyzed to find critical information.

H. Extracted Data Stage

The process of extracting encrypted files is an important step in forensic analysis to reveal the contents of the file and any artifacts that may be hidden within it [10]. At this stage, after the password has been found to open the encrypted zip file, this stage performs an extraction or unzipping of a file. The extraction process is performed using the unzip command, which includes a password option to enter the previously obtained password [14]. This command is executed in the terminal or command prompt so that the extraction process can run. During the extraction process, the system will verify the correctness of the entered password, and if it is correct, the zip file will be successfully extracted, and its contents can be accessed.

I. Obtain Flag

After going through a series of detailed forensic memory analysis processes, the researchers finally succeeded in finding the flag or important evidence they were looking for. This flag is usually a text string or special information hidden in a file, image, or other artifact in the analyzed memory or system. Finding the flag is the main objective of the entire forensic investigation process. Successfully obtaining the flag indicates that all stages and steps in the forensic investigation have been carried out properly and correctly.

III. Results and Analysis

A. Run Volatility Imageinfo

After obtaining the memory dump file to be analyzed, the first step is to use the imageinfo option in Volatility.

e-ISSN: 2548-6861

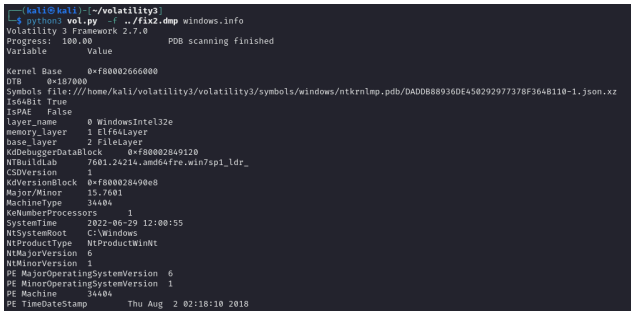

Figure 2. imageinfo options from volatility

B. Identify Suspicious Process

After determining the operating system being used, the second step is to use pslist to display the processes that were running at that time. From the results, suspicious processes were found such as 7zFM and notepad.

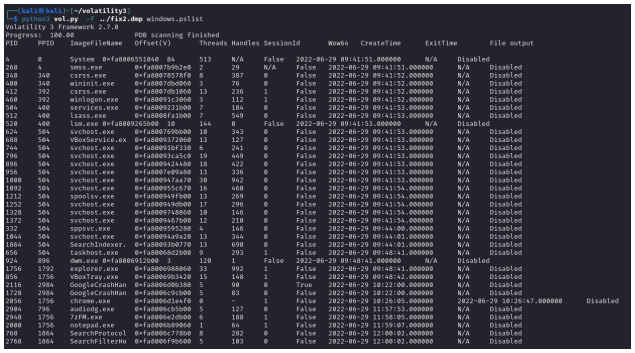

Figure 3. pslist options from volatility

Next, use the cmdline option to display details about how those processes were started and with what arguments they were run. From the results, it was found that there was an iniadalah.zip file created using 7z and a flag.txt.txt file using notepad.

Figure 4. cmdline options from volatility

C. Find Offset of Suspicious Files

After identifying the suspicious files or processes from the previous methods, the next step is to find the offsets of those two files using the filescan options and grep command to target the files you want to search for.

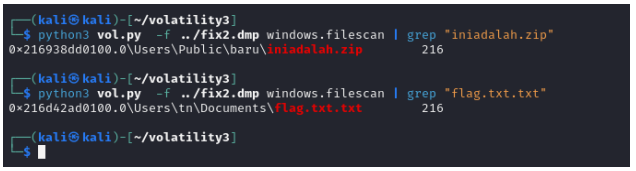

Figure 5. filescan options from volatility

Based on the output from that image, it was found that 0x216938dd0 is the offset of the iniadalah.zip file and 0x216d42ad0 is the offset of the flag.txt.txt file.

D. Dump Suspicious Files

After obtaining the offsets, the next step is to perform a dump using the dumpfiles option.

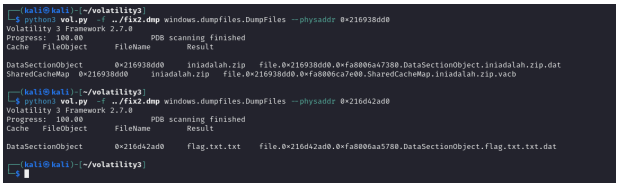

Figure 6. Dumpfiles options from volatility

E. Identify File Types

Then checking each file using the 'file' command. It is known that the file format matches the one seen in the previous cmdline. A password is required to open the iniadalah.zip file.

Figure 7. File command

F. Password Found

Next, when opening the flag.txt.txt file, there was some text inside it. From the file description, it was not the flag, but there was a string that we assumed to be the password to open the previous zip file.

Figure 8. Check file content

G. Extracted Data

After finding the password to open the encrypted zip file, the next step was to extract or unzip the file. The extraction process was done using the 'unzip' command equipped with the password option to enter the previously obtained

password. This command was executed in the terminal or command prompt so that the extraction process could run. During the extraction process, the system will verify the correctness of the entered password and if it is correct, the zip file will be successfully extracted and its contents can be accessed. From the result of extracting this zip file, an image file named word.png was found, which was previously hidden inside the encrypted zip file. This image file then became the object of further analysis in an effort to find the flag.

Figure 9. Extract zip file with password

H. Obtain Flag

After successfully extracting the word.png image file from the encrypted zip file, the next step was to open and analyze the contents of the image file. When the image file was opened using the appropriate image editing software, hidden text or information was found inside the image, which was the flag or important evidence being sought **Flag: KMIPN4{1ts_34sy}**.

This flag was obtained after going through a fairly complex and detailed series of memory forensic analysis processes. Successfully obtaining this flag indicates that the entire forensic investigation process was carried out properly and achieved its main objective. The fact that the flag was successfully obtained indicates that all stages and steps in the forensic investigation were carried out correctly.

Figure 10. flag

After all the analysis and flag search processes were completed, important findings and the steps taken during the investigation were documented in a written report or write-up.

This report serves as a record and evidence of the forensic investigation activities that have been carried out, and can be used as a reference or learning material in the future.

IV. Conclusion

This research proposes the use of the Volatility Framework, specifically the latest version of Volatility 3, to extract hidden files from the memory dump provided in the Capture The Flag (CTF) competition. Volatility 3 is a better choice as it is more modern, faster, efficient, and can take advantage of the latest features, making it effective in extracting important information from memory dumps as in the context of Capture The Flag (CTF). Memory dump analysis is an important technique in digital forensics and security incident investigation to uncover suspicious activities and hidden evidence that is not available on disk. The memory forensic analysis process in this research is based on the researchers' experience and understanding by combining and adapting from various sources aligned with the research context and objectives. Volatility is used as the main framework for analyzing the obtained memory dump, utilizing various plugins and modules such as imageinfo, pslist, cmdline, filescan, grep, and dumpfiles. Through a series of analysis processes using Volatility, suspicious processes, hidden file locations, and the password required to open the encrypted file were successfully identified. After successfully opening the encrypted file, an image file containing the sought flag or important evidence was found. This capability is very useful in the context of cybersecurity competitions such as Capture The Flag (CTF), which often involve memory dump analysis challenges.

References

[1] M. Nauval Rifkiansyah, R. Satria Wibowo, R. Priambudi, K. Ananda Putri, and H. Bayu Seta, Penerapan Memory Forensic Menggunakan Metode Live Forensic untuk Investigasi Random Access Memory. 2021.

[2] M. Parekh and S. Jani, “MEMORY FORENSIC: ACQUISITION AND ANALYSIS OF MEMORY AND ITS TOOLS COMPARISON,” International Journal of Engineering Technologies and

[3] Gregorius, “Implementasi Volatility dalam Mengalanalisa Malware pada Memory Dump,” Journal of Informatics and Advanced Computing (JIAC) , vol. 4, no. 1, pp. 36–43, 2023, Accessed: May 09, 2024. [Online]. Available: https://journal.univpancasila.ac.id/index.php/jiac/article/view/5491

[4] A. Case and G. G. Richard, “Memory forensics: The path forward,” Digital Investigation, vol. 20, pp. 23–33, Mar. 2017, doi: https://doi.org/10.1016/j.diin.2016.12.004.

[5] “Windows Memory Analysis with Volatility Analyst Reference Memory Analysis Volatility Analyst Reference.” Available: https://forwarddefense.com/media/attachments/2021/05/15/memory-analysis-with-volatlity-analyst-reference-20200131.pdf

[6] J. I. James, “Joshua I. James,” DFIRScience, Feb. 22, 2022. https://dfir.science/2022/02/Introduction-to-Memory-Forensics-with-Volatility-3

[7] Mualfah, Desti & Ramadhan, Rizdqi. (2020). Analisis Forensik Metadata Kamera CCTV Sebagai Alat Bukti

e-ISSN: 2548-6861

Digital. Digital Zone: Jurnal Teknologi Informasi dan Komunikasi. 11. 257-267. 10.31849/digitalzone.v11i2.5174.

[8] volatilityfoundation, "Command Reference," GitHub, May 07, 2020. https://github.com/volatilityfoundation/volatility/wiki/Command-Reference#imageinfo.

[9] "Command Reference," GitHub, May 07, 2020. https://github.com/volatilityfoundation/volatility/wiki/Command-Reference#pslist.

[10] Ligh, M. H., Case, A., Levy, J., & Walters, A. (2014). The art of memory forensics: detecting malware and threats in windows, linux, and Mac memory. John Wiley & Sons.

[11] "Command Reference," GitHub, May 07, 2020. https://github.com/volatilityfoundation/volatility/wiki/Command-Reference#filescan.

[12] "Command Reference," GitHub, May 07, 2020. https://github.com/volatilityfoundation/volatility/wiki/Command-Reference#dumpfiles.

[13] "Menjaga Dunia Digital Lewat Investigasi Forensik Digital," Horangi.com, 2024. https://id.horangi.com/blog/menjaga-dunia-digital-lewat-investigasi-forensik-digital/.

[14] Linuxize, "How to Unzip Files in Linux," Linuxize.com, Sep. 2018. https://linuxize.com/post/how-to-unzip-files-in-linux/