Memory Forensics and Cyber Threat Analysis: A Review

Madhav Singh Solanki

SOEIT, Sanskriti University, Mathura, Uttar Pradesh, India

Email Id- madhavsolanki.cse@sanskriti.edu.in

ABSTRACT: Computers are becoming more widely used in today's culture. Assaults against military data centres, power grids, and proprietary information by commercial or governmental organizations are becoming more common. These kinds of assaults, as well as a variety of other computer-related events, need cyber security specialists to be able to recognize, react to, and report them. If real evidence has never been transferred to storage media, there seems to be no way to rely on disk forensics. On the other hand, even if destructive code from an infection is never transmitted to storage medium such as a hard drive, memory has a high chance of carrying it, partially or completely. Memory forensics can often recover passwords and encryption keys, as well as plain-text information from documents before they are encrypted, providing information that may be used to assess the scope of an attack. OS memory forensics includes Linux memory forensics as a major emphasis area. Businesses and government organizations are among the most passionate Linux users. Linux contains many security vulnerabilities. Memory forensics on Linux systems and advanced data analysis using machine learning should be the focus of research, since both will be very helpful to the Linux cyberspace society. As a result, research is needed to create tools and theories to enhance the security of the operating system, resulting in user cyber protection.

KEYWORDS: Cyber Threat Analysis, Forensic Process, Linux Forensic, Memory Forensic, Windows Forensic.

1. INTRODUCTION

The use of computers is becoming more common in today's society. Digital defence methods like as encryption, gateways, and parametric or pattern scanning, among others, are being used by government organizations and commercial businesses to defend themselves against cyberattacks [1]. Furthermore, the incidence of assaults against defence data centres, power grids, and proprietary information from either commercial or governmental entities is on the rise. Cyber security experts must be able to identify, respond to, and report these types of attacks, and several other computer-related events. Memory Analysis process is depicted in Figure 1.

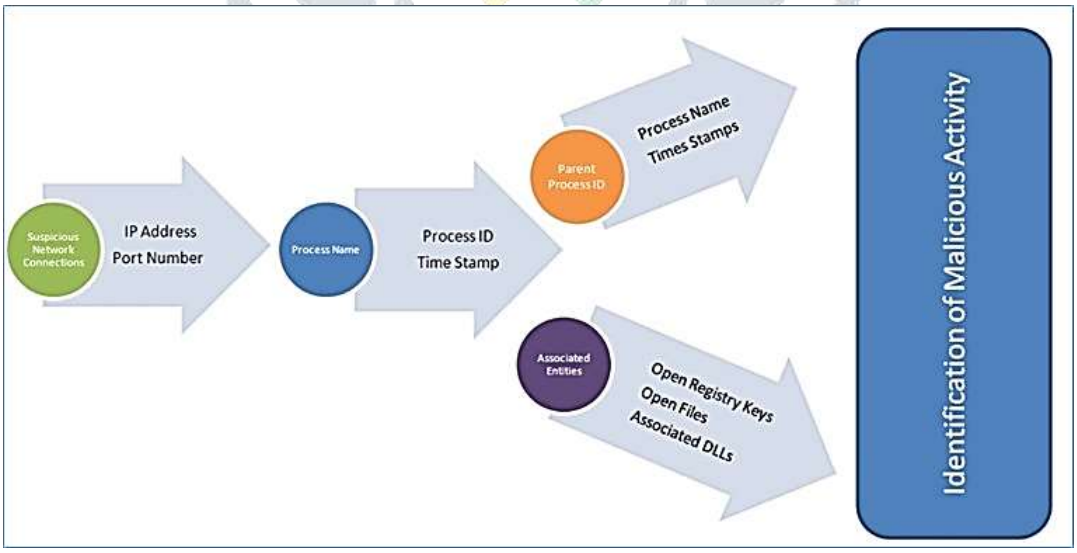

Figure 1: Representing the Depicts the process of Memory Analysis [2]**.**

Defensive players must find a way to prevail as these assaults get more sophisticated and the adversaries become more sophisticated [3]. There seems to be no way to depend on disk forensics if actual evidence has never been transmitted to storage media. But at the other side, even if harmful coding from an infection is never transferred to storage media like hard disk, memory does indeed have a high possibility to carry it, partly or fully. Since this destructive script is put into storage and then executed, that's the case. The evidence of the computer resources accessed by malicious programs is also stored in the affected computer's primary memory.

Similarly, if the information leaked from the business is encrypted throughout the network, it will be impossible to identify which important files were stolen using conventional packet capture methods. Memory forensics may frequently recover passwords and encryption keys, as well as the document's plain-text information prior they would be encrypted, giving information to determine the extent of an assault. Figure 2 depicts flow of memory storage.

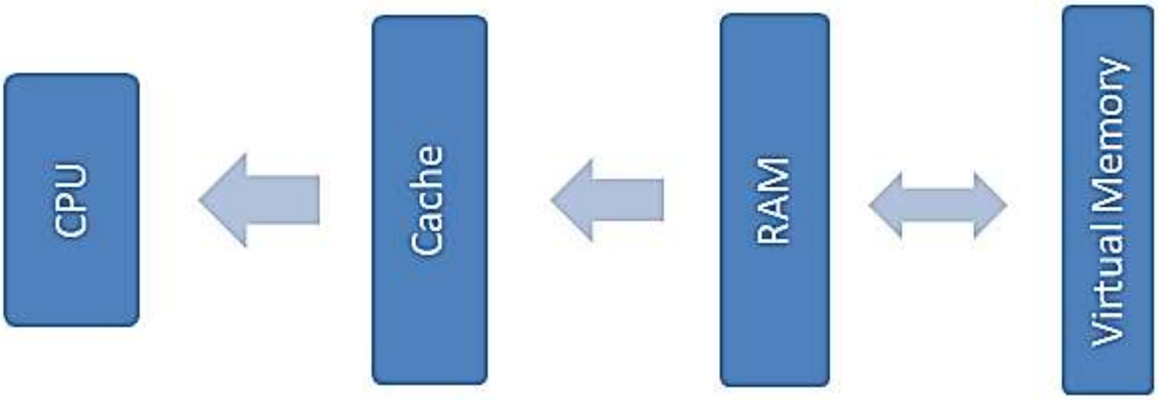

Figure 2: Representing the Flow of Memory Storage from Virtual memory to CPU.

2. LITRATURE REVIEW

Zhang et al. begin by reviewing the history and development of memory forensics research, and then introduce the fundamental operating system memory management mechanism [4]. They next go through the various techniques for acquiring and analyzing memory data, as well as a summary of the most recent memory forensics technologies. The article closes with a review of present memory forensics issues, as well as a perspective on memory forensics trends and future research paths.

Shaaban et al.'s goals is to evaluate the qualitative outputs of various analysis methods, such as memory, super timeline, and live analysis, on the same event in order to determine which methodology is better suitable in different situations [5].

Case et al. review the state-of-the-art in memory forensics, offer a critical critique of current-generation methods, explain significant changes in operating system architecture that affect memory forensics, and outline key research topics [6].

3. DISCUSSION

3.1. Memory Forensics:

Memory forensics is probably the most intriguing and productive area of computer forensics [7]. It is the study of files held in physical memory during the execution of an operating system. Its main use is in the analysis of sophisticated computer assaults that are subtle enough to stop sending data on the hard drive. As a result, forensic information must be extracted from the memory (RAM). Every function that a program or system software performs causes a unique kind of change in the system memory. These modifications typically persist for a long period after the procedure is completed, which is important for preserving them. Memory forensics also gives you unparalleled insight into the system's runtime state, such as how many programs were executing, which network interfaces were open, and which instructions were just performed. People may extract such artifacts in a way that is completely independent of the equipment under investigation. It also lowers the risk of rootkits or viruses obstructing the inquiry. Unencrypted e-mail communications, disk password protection, non-cacheable web browsing history, off-the-record chat conversations, and memory-resident inserted code snippets are examples of data that may only live in memory.

Memory forensics involves recording both the profile and the contents of the RAM, and it may be a useful resource for incident response, virus analysis, and information security. And although network packet captures and hard disks can provide compelling proof, it is often the contents of the memory storage that allows a full restoration of events, enabling an user to evaluate what's already occurred, what's really currently occurring, and what would occur if further infection by spyware or an encroachment by sophisticated threat actors were to occur. For instance, a type of proof discovered in RAM may aid in the identification of common forensic artifacts that seem to be unrelated, allowing for an integration that would otherwise go undetected.

There are many three reasons why data from the physical memory should be gathered and analyzed. The physical memory, for example, stores real-time data about the software environment, including the presently mounted file system as well as the listing of running processes. Second, when cryptographic data is kept in memory space, it is usually deciphered. Third, this approach is well-suited to the needs of embedded systems. The data in the memory space of an embedded system is largely durable since it is seldom switched off. As a result, if examination of the physical memory is done properly, important information may be retrieved. Threads, Dynamic Link Libraries (DLL), program memory, picture identity, kernel memory and objects, networking, registry, spyware, and rootkits are all examples of data that may be retrieved from memory.

3.2. Memory Forensics Process:

Figure 3 shows a self-explanatory schematic of the memory forensics procedure at a higher level. The collection of target computers is the first step in the Memory Forensics procedure. These pictures may now be in any format, including:

- Crash Dump
- Page File
- Hibernation File
- Raw Format

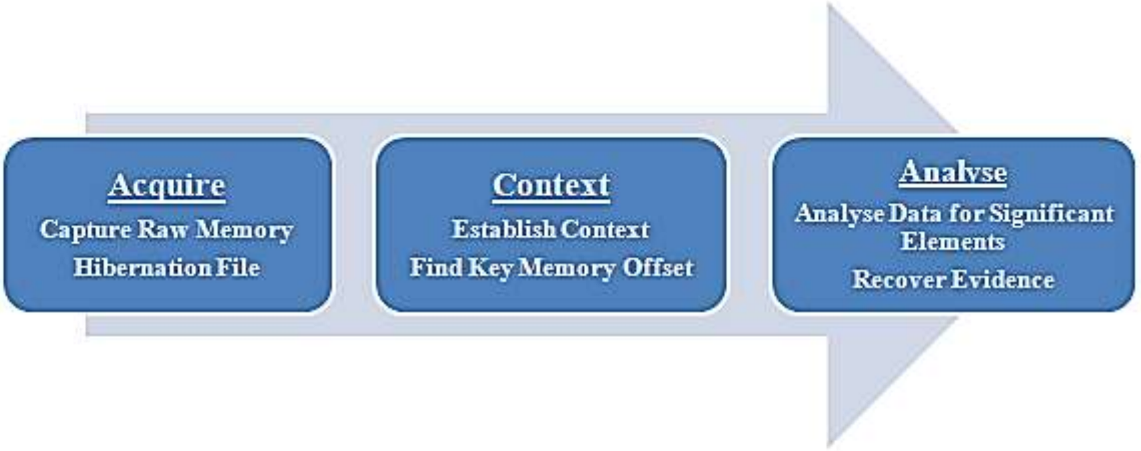

Figure 3: Representing the Process of Memory Forensics from Capturing Raw memory to Analysis of significant Elements.

3.3. Why Memory Forensics?

Almost everything within a operating system, which include process steps and threads, spyware and computer viruses, IP addresses, internet backbone sockets, web links, open files, passcodes, catches, clipboards, as well as other user-generated information, cryptographic keys, equipment and software setups, and registry entries keys and incident logs, passes through random access memory.

The kinds of artifacts discovered in memory dumps all come from the same place. They all begin as an assignment. In view of the information stored inside and surrounding the storage areas, the why, when, and how they were assigned distinguishes them. The analysis of these behaviours as component of memory forensics may be useful in making conclusions about content allocation, resulting in the ability to locate and classify particular kinds of material across a huge memory dump. Furthermore, knowing memory assignment and re-assignment methods may help in comprehending the changing external. For instance, which memory cell is free and which is reserved.

Memory forensics aids in the investigation of sophisticated malware. Since malicious products can be examined more extensively in memory, more valuable IoCs may be constructed. Memory forensics of well-known assaults like Stuxnet and black energy, for instances, disclosed some previously unknown artifacts concerning the attack. Memory forensics is a procedure that begins with locating a compromised system, collecting its memory, evaluating it, and, if necessary, dropping the harmful program for further investigation. Memory forensics allows us to examine a variety of OS artifacts, including active processes, live security mechanisms, installed drivers, API hooks, and objects like the shim cache, which exists only in memory and is only flushed to disk after a computer restart. It may also be used to investigate memory-resident malware, which does not write any data to disk and therefore goes undetected. Memory forensics may also be used to identify malware concealing activities by analyzing sophisticated kernel level assaults such as Direct Kernel Object Manipulation (DKOM). Memory forensics may be aided by a variety of tools like Volatility, Rekall, Redline, and others.

3.4. Different Operating Systems with Memory Forensics:

3.4.1. Windows Forensics:

Locating and analysing executive items is part of memory forensics for Windows. Windows is built in C, and C structures are extensively used in data and attribute organization [8]. Several of these structures are referred to as executive objects. The Windows object manager creates, deletes, and protects them. The NT module implements the object manager, which is a kernel component.

For memory forensics, the most important executive objects in Windows are file, program, link, authentication, window station, process, workstation, mutant, types, etc. These executive objects may be found using the structure's name (for example, a file with the _FILE OBJECT structure). Several of the executive items mentioned above has minimum of one Volatility module that examines it.

3.4.2. Linux Forensics:

The basic technique in memory forensics of Linux is to start studying Linux memory dumps. On Linux, one has to be conscious of both classic and contemporary memory acquisition methods, as well as their advantages and disadvantages [9]. To conduct Linux memory forensics, Volatility needs to generate Linux profiles that are archives that include valuable information for finding and intercepting data in Linux memory dumps. Additionally, one should be mindful of the difficulties in deploying Memory forensics in an organizational setting, where important servers may lack C compilers and other libraries available on ordinary Linux workstations and desktops.

3.4.3. Mobile Operating Systems Forensics:

Windows 10, Android, iOS, Sailfish, Tizen, and Ubuntu touch are among the mobile operating systems presently in use. Android and iOS are the most commonly used mobile systems [10]. Because of the fast growth of Android as well as Mac OS systems across both home and business settings, both platforms have become targets for targeted assaults. As a result of these considerations, cybersecurity professionals have tried to create solutions for Android and iOS that will allow Windows and Linux systems to conduct comprehensive investigations. To conduct memory forensics on Android and Mac OS, build a Volatility configuration for Android and Mac platforms and utilize one of several solutions for memory collection. Aspects such as 64-bit addressing on 32-bit kernels, the usual userland and layout of kernel main memory, and the usage of microkernel elements are all important concerns.

3.5. Memory Forensics Tools:

While there are a variety of memory forensics tools available, the Volatility architecture and LiME i.e., Linux Memory Extractor are the most often used. One of the finest memory dump tools is LiME. ShmooCon published a Linux kernel module (LKM) that produces memory leaks for the Linux machine. It's the first program that can take complete memory dumps from Debian and Android phones. LiME is a useful system that can conduct memory dumps without any additional actions, such as changing kernel settings, by loading components immediately after building. After loading a pre-compiled package file into external memory and dumping a document straight into external memory, mainly in the presence of android, one may drop a file directly into memory card. LiME characteristics allow for complete memory acquisition as well as acquisition via a network interface while maintaining a small process footprint.

The Volatility framework is the second most popular memory forensic tool. It's a unified framework for analyzing RAM dumping from Linux, Windows, Mac OS X, and Android devices. Volatility's modular design makes it simple to handle new platforms and architectures when they become available. As a result, all gadgets are potential targets. Its forensic capabilities aren't limited to Windows PCs. It's also open source, built in Python, with an extendable and scriptable API that offers unrivalled feature sets and extensive file format coverage.

3.6. Advantages:

Memory forensics' advantages include, but are not restricted to:

- Memory forensics is a kind of investigation that involves both physical memory data (from RAM) and data from the Page File (or SWAP space).
- Memory analysis is the greatest location to look for harmful program activities.
- Memory forensics aids in the analysis and tracking of recent system actions in relation to the user's profile and attackers' activities.
- Malicious code in random access memory is not yet performing anti-forensics.
- Memory forensics is the sole method to gather evidence that can't be discovered elsewhere, such as chat threats, Internet activity, memory-only virus, and so on.
- RAM processes all data that is generated, read, or deleted. This covers all online surfing, document changes, pictures, network data sending and receiving, program execution, and basically everything that shows on the monitor.
- Because RAM functions as a "disk," it is essential to maintain and examine the memory as well as the hard drive.

3.7. Challenges:

Memory forensics presents a number of difficulties, such as the following.

- Taking a look at the many memory harvesting tools that are available, each of which performs differently based on the operating system variant, installed devices, and settings.
- OS internal structures are quickly changing as a result of regular OS updates from OS manufacturers, yet memory forensics techniques are incompatible with such images. There are, for example, a number of picture acquisition programs that are incompatible with Windows 8.
- If the terminating character is missing. Consider the case where, when examining the logical address space of a computer that uses paged virtual memory, a string passes a page border to a page that is no more memory resident, necessitating additional processing or heuristics to ascertain the string's real size.
- The most crucial aspect is to ensure that the picture is correctly captured and that it retains its coherence throughout the analysis and inquiry. Without a clean picture capture, there would be very few, if any, artifacts inside the image to examine.
- The difficulties encountered in linked list evaluation also apply to memory tree analysis.
- Memory forensics knowledge land is vast, and filtering out expected products from abnormalities requires a comprehensive knowledge of underlying structures and anticipated process behaviour.
- Memory evidence may be discovered in a variety of forms and sizes on non-volatile media. As a cyberspace security specialist, you must understand the various formats and how to convert one type to another.
- The memory areas that are of interest and need additional examination must be emptied. Memory investigations must be coupled with Reverse Engineering to do an end-to-end analysis.

- The whole disk, individual partitions, and virtualized file-based containers are protected when the system is turned off. Even if investigators get access to the media, they face significant difficulties as a consequence of this protection.

4. CONCLUSION

Linux memory forensics is a significant focus area of OS memory forensics. Government agencies and businesses are among the most enthusiastic Linux users. The Linux has many security flaws. The study emphasis should be on memory forensics in Linux computers and sophisticated data analysis utilizing machine learning, both of which will be very beneficial to the Linux cyberspace society. Government agencies, businesses, and small businesses will be able to protect their system operations as a result of this. Antivirus software, that are some of the options, may prevent malware from accessing the system; but, what happens if an antivirus is deactivated due to rootkit admin access? As a consequence, research is required to develop tools and theories to improve the operating system's security, resulting in user cyber protection. One of the main factors may be research focusing on memory forensics utilizing machine learning, which is seldom addressed by anti-virus solutions on the market. In the cyber world, such study may be the answer to a lot of problems.

REFERENCES

[1] Y. Zhou, M. Han, L. Liu, J. S. He, and Y. Wang, “Deep learning approach for cyberattack detection,” 2018, doi: 10.1109/INFCOMW.2018.8407032.

[2] R. J. Mcdown, C. Varol, L. Carvajal, and L. Chen, “In-Depth Analysis of Computer Memory Acquisition Software for Forensic Purposes,” *J. Forensic Sci.*, 2016, doi: 10.1111/1556-4029.12979.

[3] G. Austin, “Restraint and Governance in Cyberspace,” in *Global Insecurity*, 2017.

[4] Y. Zhang, Q. Z. Liu, T. Li, L. H. Wu, and C. Shi, “Research and development of memory forensics,” *Ruan Jian Xue Bao/Journal of Software*. 2015, doi: 10.13328/j.cnki.jos.004821.

[5] A. Shaaban and N. Abdelbaki, “Comparison study of digital forensics analysis techniques,” 2018, doi: 10.1016/j.procs.2018.10.128.

[6] A. Case and G. G. Richard, “Memory forensics: The path forward,” *Digit. Investig.*, 2017, doi: 10.1016/j.diin.2016.12.004.

[7] Y. Cheng, X. Fu, X. Du, B. Luo, and M. Guizani, “A lightweight live memory forensic approach based on hardware virtualization,” *Inf. Sci. (Ny).*, 2017, doi: 10.1016/j.ins.2016.07.019.

[8] A. Barakat and A. Hadi, “Windows forensic investigations using powerforensics tool,” 2016, doi: 10.1109/CCC.2016.18.

[9] F. Block and A. Dewald, “Linux memory forensics: Dissecting the user space process heap,” 2017, doi: 10.1016/j.diin.2017.06.002.

[10] S. Hazra and P. Mateti, “Challenges in Android Forensics,” 2017, doi: 10.1007/978-981-10-6898-0_24.