An Overview of Cyber Threat Analysis using Memory Forensics

Pankaj Saraswat

SOEIT, Sanskriti University, Mathura, Uttar Pradesh, India

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

ABSTRACT: ***In today's society, computer systems are becoming more extensively utilized. Commercial or governmental entities are increasingly targeting military data centres, power grids, and proprietary information. Cyber security experts must be able to identify, respond to, and report these types of attacks, as well as, a number of other computer-related incidents. There is no way to depend on disk forensics if actual evidence was never copied to storage media. On the other hand, even if an infection's destructive code is never transferred to a storage media like a hard disk, memory has a high probability of holding it, partly or fully. Memory forensics can frequently recover credentials and encryption keys, and also plain-text data from files before they are encoded, allowing investigators to evaluate the extent of an assault. Linux memory forensics is a significant focus area in OS memory forensics. Businesses and government agencies are among the most ardent Linux supporters. There are many security flaws in Linux. The emphasis of study should be on memory forensics on Linux systems and sophisticated data analysis utilizing machine learning, since both will be very beneficial to the Linux cyberspace civilization. As a consequence, research is required to develop tools and theories to improve the operating system's security, culminating in user cyber defence.***

KEYWORDS: ***Cyber Threat Analysis, Digital Forensics, Forensic Process, Memory Forensic Tools, Volatile Memory.***

1. INTRODUCTION

As the cyber attacks grow more complex and the opponents become smarter, defence teams must find a method to win [1]. If real evidence was never transferred to storage media, there appears to be no way to rely on disk forensics. On the other hand, even if destructive code from a virus program is never transmitted to storage medium such as a hard drive, memory has a high chance of carrying it, partially or completely. This is the situation because the harmful script is stored and subsequently executed. The evidence of malware accessing system resources is also recorded in the main memory of the afflicted machine. Fig. 1 illustrates Memory Analysis Process.

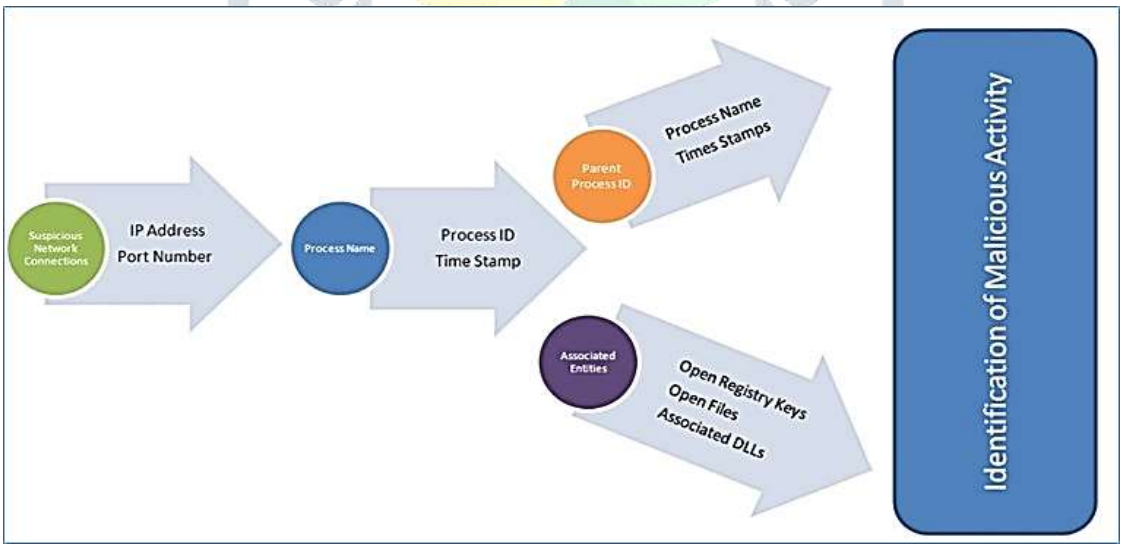

Fig. 1: Illustrates Memory Analysis Process [2]**.**

Computers are becoming more widely used in today's culture. Government agencies and commercial companies employ digital security techniques such as cryptography, firewalls, and parameterized or parametric scanning, and so many more, to protect themselves from hackers. Moreover, attacks on national defence datacentres, power grids, and trade secrets by corporate or regulatory entities 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.

Similarly, if the information stolen from the company is encoded across the network, traditional packet capture methods will be useless in determining which sensitive information have been stolen. Memory forensics can often recover credentials and private key, and also plain-text content from documents before they are encrypted, providing information that may be used to assess the scope of an attack. Memory storage flow is shown in Fig. 2 below.

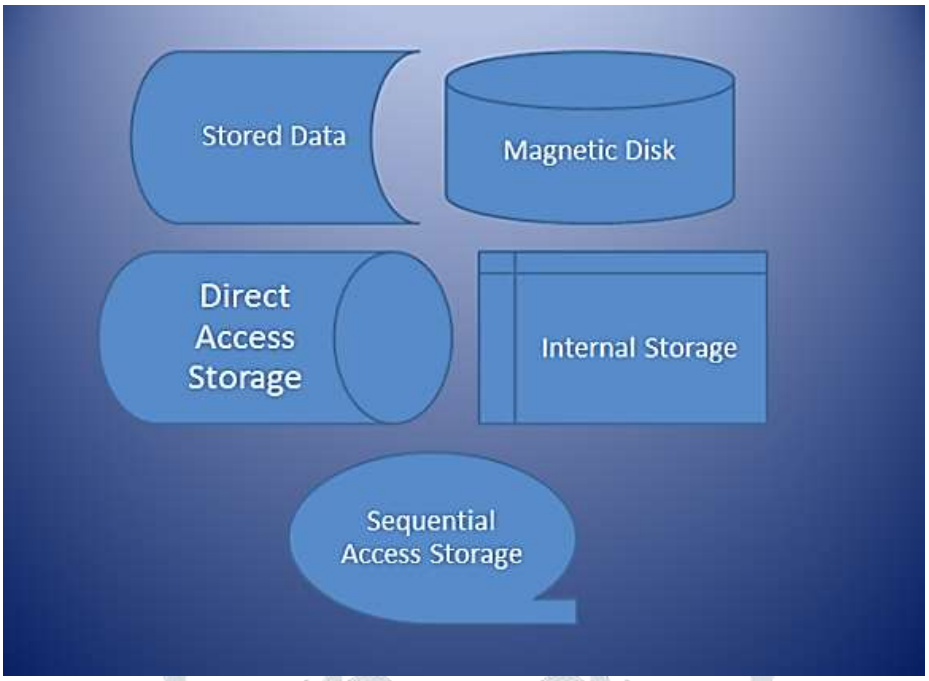

Fig. 2: Illustrates Memory Storage flow [3]**.**

2. LITRATURE REVIEW

Case et al. examine the present status of memory forensics, provide a critical assessment of current-generation techniques, explain how major changes in operating system design impact memory forensics, and identify essential research areas [4].

The aim of Shaaban et al. is to compare the qualitative outputs of several analysis techniques on the same event, such as memory, super timeline, and live analysis, in order to identify which approach is best suited to particular circumstances [5].

Zhang et al. begin by discussing the history and progression of memory forensics research, before moving on to the basic operating system memory management mechanism [6]. They next go through the many methods for collecting and analyzing memory data, as well as a rundown of the most current memory forensics technology. The essay concludes with a discussion of current memory forensics problems, as well as a look forward at memory forensics trends and research directions.

3. DISCUSSION

3.1. What is Memory Forensics:

Memory forensics is probably the most intriguing and productive area of computer forensics. It is the study of information saved in physical storage during the execution of an operating system [7]. 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 evidence must be extracted from the memory (RAM). Every function that a program or system software performs causes a unique kind of change in the RAM. 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 real-time state, such as which programs were functioning, which networks were open, as well as which operations were just performed. Individuals may extract these objects 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 private key, non-cacheable web history records, chat conversations, and memory-resident implanted code snippets are examples of data that may only live in memory.

Memory forensics includes capturing the profile as well as the information of the RAM, and it may be helpful for incident response, virus investigation, and data security. And, while network packet acquires and storage devices can provide strong evidence, it's often the ingredients of data storage which enables a complete rebuild of events, allowing a user to assess what's already happened, what's really happening now, and what would happen if spyware or sophisticated threat actors intruded further. For example, a kind of evidence found in RAM may help identify seemingly unrelated forensic artifacts, permitting for an unification which could go unnoticed.

There seem to be three main factors why data from the physical memory should be gathered and analyzed. The physical memory, for example, stores real-time data about the software environment, such as the presently mounted file system as well as the list of running processes. Secondly, when encoded data is kept in physical memory, it is usually decrypted. Lastly, this technique is well-suited to enterprise network requirements. The data in an embedded system's memory area is relatively long-lasting since it is seldom turned off. As a consequence, significant information may be recovered if the memory location is examined correctly. Information that may be accessed from memory includes threads, Dynamic Link Libraries (DLL), program memory, image identification, registry, networking, and rootkits.

3.2. Process of Memory Forensics:

Fig. 3 shows a self-explanatory schematic of the memory forensics procedure at a higher level. The initial stage in the Memory Forensics process is to gather target PCs [8]. These images may now be saved in any format, such as:

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

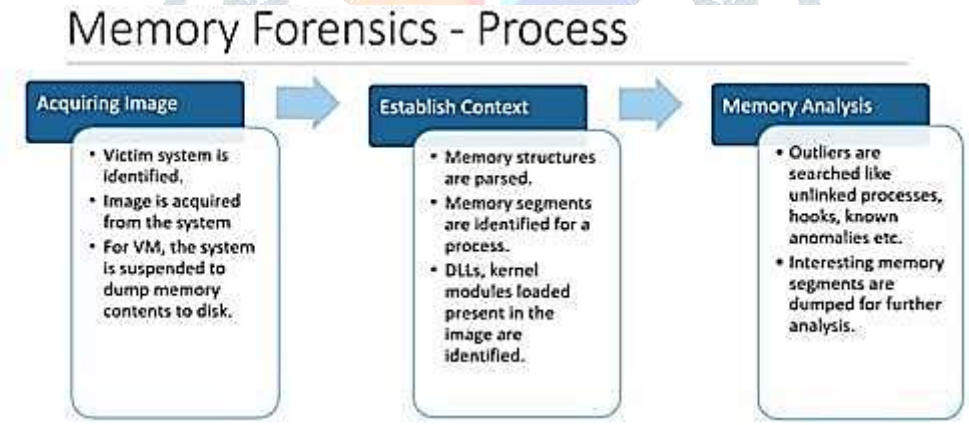

Fig.3: Illustrate general process involved in Memory Forensic [8]**.**

3.3. Why is Memory Forensics Important?

All things in an operating system, such as threads and operations, trojan and spyware, network connections, hyperlinks, open folders, login information, catches, clipboards, and many other device-generated subject matter, ciphertext keys, physical and logical settings, and windows event logs & registry keys, passes through flash memory.

Memory forensics is useful in the analysis of advanced malware. Because harmful goods may be evaluated in memory for longer periods of time, more useful IoCs can be built. For example, memory analysis of well-known attacks like Zika and dark energy revealed previously undisclosed artifacts about the attack. It is a process that starts with identifying a hacked machine, capturing its memory, analyzing it, and, if required, removing the malicious software for further research. Active applications, live security measures, loaded drivers, Application Program Interface, and entities like the shim caching that lives only in RAM and is only emptied to storage after a machine restart, may all be examined using memory forensics. It has also been used

to look into memory-resident viruses, which doesn't write anything to drive and thus remains unnoticed. By examining advanced kernel level attacks along with Direct Kernel Object Manipulation, memory forensics could even be utilized to detect malware hiding operations. A number of tools, such as Volatility, Rekall, Redline, and many others, may help in memory forensics.

The kinds of artifacts discovered in memory dumps all come from the same place. They all begin with an allocation, which distinguishes them from one another based on why, when, and how the memory areas were assigned, as well as the data stored inside and surrounding them. The study of these behaviours as part of memory forensics may be useful in making conclusions about content allocation, culminating to the ability to locate and classify particular kinds of material across a huge memory dump. Furthermore, knowing memory allocation and de-allocation methods may help in comprehending the data context. For instance, which memory block is free and which is reserved.

3.4. Memory Forensics on Various Operating Systems:

3.4.1. Windows Forensics:

For memory forensics, the most important executive items in Windows are file, task, link, certificate, window platform, process, workstation, and keys. These executive objects may be found using the structure's title, for example, a file with the _FILE OBJECT structure [9]. Each of the executive objects mentioned above has atleast one Volatile component that examines it.

Memory forensics on Windows includes locating and analyzing executive objects. C structures are widely used in data and attribute management, and Windows is written in C. Executive objects are a term used to describe a number of these structures. They are created, deleted, and protected by the Windows object management. The object management system that really is a kernel element is implemented by the NT module.

3.4.2. Linux Forensics:

The basic technique in Linux memory forensics is to start by examining Linux memory dumps. On Linux, one must be aware of both conventional and contemporary memory acquisition methods, as well as the advantages and disadvantages of each [10]. To conduct Linux memory forensics, Volatility needs to generate Linux profiles, which are archives that include valuable information for finding and intercepting data in Linux memory dumps. Furthermore, one should be aware of the challenges of implementing Memory forensics in an organizational environment, where critical computers may lack C interpreters as well as other packages that are often found on Linux notebooks and workstations.

3.4.3. Forensics of Mobile Operating Systems:

Mobile operating systems such as Android, Windows 10, iOS, and Ubuntu touch are now in use. The most widely used mobile operating systems are Android and iOS. Because of the rapid use of Android and Mac OS systems in both home and corporate environments, both are becoming targets for targeted attacks. As a consequence of these concerns, cybersecurity experts have attempted to develop Android and iOS solutions that will enable Windows and Linux platforms to perform thorough investigations. Build a Volatility setup for Android and Mac platforms and use one of many memory collection methods to perform memory analysis on Mobile OS. Important considerations include 64-bit referencing on 32-bit processors, the typical userland and structure of kernel primary memory, and the utilization of microkernel components.

3.5. Resources for Memory Forensics:

Volatility is the second most often used memory investigative tool. It's a single tool for examining RAM dumps from Ubuntu, Microsoft, Mac OS X, and iOS devices. The modular nature of Volatility makes it easy to accept innovative products and architectures as they become available. As a consequence, any device might be a target. Its forensic skills aren't restricted to Windows-based computers. It's also free and open source, with a Python-based API that can be extended and scripted to provide unmatched functionality and file format support.

The Volatility framework and LiME (Linux Memory Extractor) are the most often used memory forensics tools. LiME is one of the best memory dump tools available. ShmooCon released a Linux kernel module (LKM) which causes the Linux system to leak memory. It's the first software to be able to capture full memory

dumps from Fedora and Android tablets. By launching items immediately after construction, LiME is a helpful system that may do memory dumps without requiring any further steps, such as altering kernel settings. One may transfer an entire data into usb drive after installing a pre-compiled program file into memory card and dropping a file strait into memory space, mostly in the existence of Android. LiME's features enable full memory collection as well as network integration while retaining a compact process footprint.

3.6. Benefits:

The benefits of memory forensics comprise, but are not constrained to:

- All data that is created, read, or destroyed is processed in RAM. This includes all internet browsing, document updates, photos, network data transmission and reception, program execution, and pretty much everything else that appears on the display.
- Memory forensics, on the other hand, is a kind of inquiry that includes both physical memory data (from RAM) and data from the Page File (or SWAP space).
- Because RAM acts as a disk, it's critical to keep both the memory and the hard drive in good working order.
- Memory forensics assists in the study and monitoring of recent system operations in connection to the user's profile and the activities of attackers.
- The best place to search for hazardous program activity is memory analysis.
- Malicious code in random access memory has not yet been found to conduct anti-forensics.
- Memory forensics is the only way to collect evidence that can't be found anywhere else, including chat threats, Internet activities, memory-only viruses, and so on.

3.7. Difficulties:

Memory forensics has a variety of challenges, including the below mentioned.

- Examining innumerable memory harvesting techniques present, each of which works differently depending on the operating system, connected devices, and settings.
- If the last character isn't present. Consider the situation when a string crosses a page boundary to a page that is no longer memory resident while analyzing the physical instruction set of a computer that utilizes paged virtual memory, requiring extra analysis or assumptions to determine the string's true size.
- As a consequence of frequent OS upgrades from OS vendors, OS internal architectures are rapidly changing, yet memory forensics methods are irreconcilable with such images. A variety of image capture applications, for example, are not compatible with Windows 8.
- The knowledge field of memory forensics is large, and filtering out expected results from anomalies requires a thorough understanding of fundamental mechanisms and typical key parameters.
- The most important element is to guarantee that the image is recorded properly and that it remains coherent throughout the study and examination. There'd be few, if any, abnormalities to analyze within the image without a clean image capture.
- Memory tree analysis faces the same challenges as linked list evaluation does.
- Once the system is switched off, the whole disk, specific volumes, and emulated file-based applications are all secured. Even if authorities get control, they will encounter considerable challenges as a result of the security.

- On non-volatile media, memory information may be found in a range of shapes and sizes. As a cybersecurity security expert, you must be familiar with the different file formats as well as how to convert between them.

- Empty the memory regions that are of relevance and need further investigation. To do an end-to-end examination, memory studies must be combined with reverse engineering.

4. CONCLUSION

Businesses and government organizations are one of the most passionate Linux users. Linux contains many security vulnerabilities. OS memory forensics includes Linux memory forensics as a major emphasis area. As a consequence of this, government organizations, corporations, and small enterprises will be able to safeguard their system operations. 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 Debian cyberspace culture. Some solutions, such as antivirus software, may prevent infection from gaining access to the system; however, what occurs if an antivirus is disabled owing to rootkit admin privilege? As a result, research is needed to create tools and concepts to enhance the security of the operating system, resulting in individual cyber defence. Research focused on memory forensics using deep learning, which is rarely addressed by anti-virus products on the market, may be one of the primary reasons. A research like this may be the solution to a lot of issues in the cyber world.

REFERENCES

[1] K. Geers, "The challenge of cyber attack deterrence," *Comput. Law Secur. Rev.*, 2010, doi: 10.1016/j.clsr.2010.03.003.

[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] "PowerPoint Flowchart – File And Info Storage Symbols." .

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

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

[6] 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.

[7] A. Case, A. K. Das, S. J. Park, J. Ramanujam, and G. G. Richard, "Gaslight: A comprehensive fuzzing architecture for memory forensics frameworks," 2017, doi: 10.1016/j.diin.2017.06.011.

[8] S. Ninja, "Memory Forensics Power: An Introduction," 2017. .

[9] N. Ruff, "Windows memory forensics," *J. Comput. Virol.*, 2008, doi: 10.1007/s11416-007-0070-0.

[10] C. H. Malin, E. Casey, and J. M. Aquilina, "Linux Memory Forensics," in *Malware Forensics Field Guide for Linux Systems*, 2014.