Study of Memory Forensics: Memory Analysis Technique

Dimple Likhar[1], Prof. Monali Rajput[2]

[1]*Student, Department of MCA, Vivekanand Education Society's Institute of Technology,*
[2]*Assistant Professor, Department of MCA, Vivekanand Education Society's Institute of Technology*

---***---

Abstract: *The emerging world of technology is also becoming a platform for the increase of cybercrimes. The tools like antivirus and anti-malware serve no purpose in detecting malware, which is directly written into a computer's physical memory, i.e., RAM. Analysis of a computer's memory is known as memory forensics. It's the investigation of advanced attacks on the computer system of various organizations or home computers. This paper reviews how critical role of memory forensics is in investigation of any digital crime.it also determines how important the metadata is which helps in gaining information on the activity of opening and loading of files in a computer. Thus, the paper aims towards providing algorithms to retrieve metadata from the computer helping in the investigation process.*

Keywords: **Memory forensics, VAD, Address Translation, FAT, Volatile memory**

1. Introduction to Memory Forensics

Volatile data is any temporarily stored data on any device while it is in a running state and would be lost if it shuts down due to any reason. Temporary system files, cache and RAM have Volatile data stored in them.[1] Volatile data may also contain the last actions performed on any data, if it is unsaved. If any threat occurs on this data, it can't be detected by traditional security applications. It is reported that more than 80% of organizations were affected by cybercrime in the year 2019.[2]

Memory-resident malware also referred to as file-less malware, maybe a sort of malicious software that writes itself directly onto a computer's system memory. This behaviour leaves only a few signs of infection, making it difficult for traditional tools and non-experts to spot. Here the concept of Memory Forensics inherits the image. In such cases, security teams need to depend upon memory forensic tools to guard their valuable information against stealthy attacks like DoS and file-less. The breaches, cyberattacks, vulnerabilities can be investigated using various tools.[1]

To detect malware which are memory resident, traditional antivirus applications should be supplemented by technologies that help in capturing the Volatile Memory and monitor continuous behaviour.[7] Organisations should look to Network and Host-based IDS, also as Endpoint Analytics, to assist identify indicators of compromise.[7] When any memory-resident malware is detected, further analysis is needed to boost the response efforts and help in future to configure security systems to pinpoint similar attacks. An estimated $7.5 billion is cost due to various attacks on computers.[8]

2. Memory Analysis Algorithms

The main focus of the analysis is for the retrieval of the memory which is in running state and to find of the object of the memory in the computer. All the efforts are made using the algorithm which is Address Translation Algorithm, this algorithm helps in finding the information from the object which is linked to the memory in the computer which in turn helps in finding actual physical location of that memory in the computer.[5]

There are various other approaches for finding the path of the object in computer memory. One of it is by tracking Virtual Address Descriptor which helps in accessing the root data of the object from the process block of the computer.[5] By this the file can be obtained which in turn can be used to retrieve the filename and data within it. VAD is kind of like a tree which is made by the memory of the computer to keep a track of all the data which is getting used. [4] Because of the tree formation that VAD uses it is easy to find the data in depth rather than finding just the location where the file is kept rather it also helps in finding the name, created time and many other details of the file which is been on search, this single benefit makes the VAD more powerful than the Address translation the main difference between the Address translation and Virtual address descriptor.

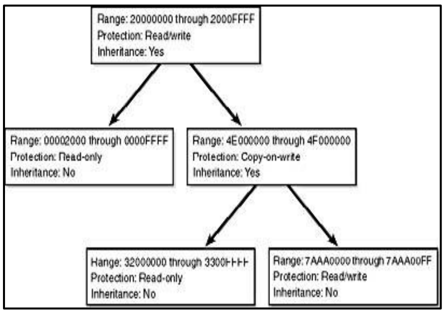
Range: 20000000 through 2000FFFF
Protection: Read/write
Inheritance: Yes
Range: 00002000 through 0000FFFF
Protection: Read-only
Inheritance: No
Range: 4E000000 through 4F000000
Protection: Copy-on-write
Inheritance: Yes
Range: 32000000 through 3300FFFF
Protection: Read-only
Inheritance: No
Range: 7AAA0000 through 7AAA00FF
Protection: Read/write
Inheritance: No

Figure 1 VAD Tree [9]

Although these techniques work efficiently only when the process is in running state in the computer if the process is stopped or terminated then this methods of retrieving the data will not be useful. They also do not help in finding the location of the file physical hard disk. As the file object is located in the page format which is impossible to detect on the computer because the data available on the computer is very less as maximum storage of the file is in the hard disk. As a result, both the methods need to be combined to find of the information on the file from both computer as well as hard disk.

The problem with not being able to find the information when the process is closed is the major drawback of the methods but it also helps in other way as it helps in finding the malware in the open file whenever a malware attack is been made on the computer. When dealing with a case where a computer is involved in orchestrating a crime, the evidence which are looked by the digital forensics' investigator will look only for the evidence or information which can be found in the hard drive. And a computer memory is loaded with various types of file extensions except .exe and .dll. There is need of a mechanism to track all the files in the computer memory to find the information and evidence on the particular crime happened on the computer, but these mechanism does not provide methods with such scalability.

3. File Allocation Table

To overcome such limitation from the previous mechanism the paper presents a modern way to obtain the file information present on the computer memory. This new approach focuses on directory entries of the File Allocation Table file system from the computer memory. The metadata collected from these directories can be collected on the basis of modification, access, create, time, size and location on the hard disk through the cluster structure of the file system on the hard disk on the computer memory. The new algorithms for the new technologies file system records the index record and file record from the computer memory as a result of using the directories of the computer system the file which will be found from the computer memory will be stored in the format of the directory settings with the help of index record found in the computer.

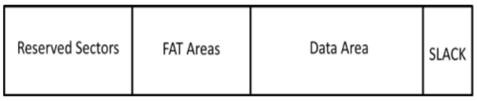

Figure 2 Directory Entry in FAT [3]

4. Process of Gathering Metadata

Now to collect the information from the directories we use the approach of the FAT system, the FAT system checks weather the directory is present in the computer memory or no or if it is been destroyed during the attack. [5] But to confirm the existence of the directory in computer system it is essential to know that the targeted computer contains the FAT system if the computer does not contain the FAT system then it's totally useless to check for the directories. The way to check that the computer has FAT system is to check the boot sector of the computer. The boot sector holds all the structure of the file system information located on the hard disk. By this we can gain information like the System ID, Volume Name etc. [3]

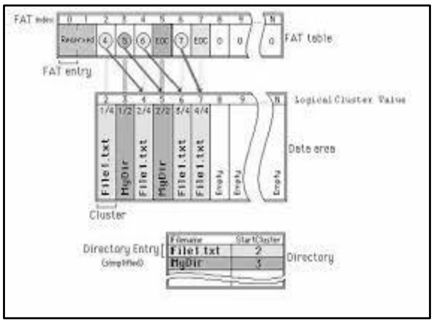

Figure 3 Directory Entry in FAT [10]

The System ID which is discovered on the hard disk will help in finding what type of file is been stored in the file system on hard disk. Whenever the computer is operational the boot sector always loads first in the computer memory before all the file on the hard disk starts its operation. Hence the analyst can know what type of files are been stored in the computer memory so that the investigation can be performed in that manner as it becomes easy and clear to investigate when all file types are known beforehand. As every computer has its own way to set up the file system in the computer memory, so to find the structure of this file system in computer memory has to be done by incorporating various unique value and signature.

As the directories in the file system has the same entry structure both on hard disk and computer memory so it is possible that the data structure of both the file system is same. The major difference will be that in computer memory the location of the files will be random rather than hard disk as in hard disk the location is fixed and known for each and every file. As there is not concrete way to find the directory entry in the FAT system in computer memory however it is easy to recognise, by using the ASCII value of every file it is easy to understand that it is the directory entry of the FAT system.[5]

To secure that the directory entry found is from FAT we can check the first three bytes of the directory entry. If the ASCII value is found of the first three bytes there is the possibility that the directory entry is from the FAT. To confirm that it is actually form the FAT we can test the Eight byte of the ASCII value of the directory entry. The result is than compared with the files extension database.

As seen above the algorithms have their own features and some disadvantages to find some points but at the end they do help in finding the missing data as seem the address translation algorithm helps in finding the physical location of the data whereas the virtual address descriptor helps in finding the root data, where it is stored and by that helps in finding the filename of that missing file.

5. Conclusion

The paper concludes that the algorithms used for retrieving the data from the hard disk are unique in their own self but lack some features when used alone but when combined these algorithms solves the problem which they are not able to solve when used single handed, the best option available is the FAT system which gives us all the necessary data required while performing the investigation hence this paper explains the use of the all three types of algorithms and file system which was invented to keep a track of data so that it can be retrieved when system failure occurs.

6. References

[1] EC Council. (2020, July 24). All you need to know about Memory Forensics – Identifying potential volatile data. EC-Council Official Blog. https://blog.eccouncil.org/all-you-need-to-know-about-memory-forensics-identifying-potential-volatile-data/#:%7E:text=What%20is%20volatile%20data%3F,files%2C%20RAM%20and%20system%20files.&text=Volatile%20data%20also%20contain%20the%20last%20unsaved%20actions%20performed%20in%20a%20document.

[2] CyerEdge Group, '2020 CyberThreath Defense Report", https://cyber-edge.com/wp-content/uploads/2020/03/CyberEdge-2020-CDR-Report-v1.0.pdf.

[3] Kelsey Laine Rusbarsky, "A Forensic Comparison of NTFS and FAT32 File Systems", FSC 630 Forensic Science Internship.

[4] Dolan-Gavitt, B. (2007). The VAD tree: A process-eye view of physical memory. Digital Investigation, 4. https://doi.org/10.1016/j.diin.2007.06.008

[5] Khairul Akram Zainol Ariffin, "Tracking File's Metadata from Computer Memory Analysis", Digitala Forensics Department CyberSecurity Malaysia.

[6] Kristine Amari, "Techniques and Tools for Recovering and Analysing Data from Volatile Memory", SANS Institue of Information Security Reading Room.

[7] The RedScan Team. (2020, March 25). How to Detect and Analyse Memory-Resident Malware. Redscan. https://www.redscan.com/news/memory-forensics-how-to-detect-and-analyse-memory-resident-malware/

[8] Wiggers, K. (2020, June 16). Kasada raises $10 million to fight content scraping and other cyberthreats. VentureBeat. https://venturebeat.com/2020/06/16/kasada-raises-10-million-to-fight-content-scraping-and-other-cyberthreats/

[9] Solomon, D. A., Russinovich, M. E., & Russinovich, M. (2000). Inside Microsoft Windows, Third Edition (Microsoft Programming Series). Microsoft Press.

[10] Chidanandan, A. (2004). An Overview of FAT. https://www.eit.lth.se/fileadmin/eit/courses/eitn50/Literature/fat12_description.pdf