International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 11 Issue: 12 | Dec 2024 www.irjet.net p-ISSN: 2395-0072

RAM Forensics of Various Versions of Windows OS Systems

Dija S, Avila Rose Fernandez

Center for Development of Advanced Computing,
Thiruvanathapuram, India

Abstract - *In today's digital landscape, cybercrime continues to evolve, making RAM Forensics a critical tool for investigators. This paper focuses on RAM forensics across various versions of Windows operating systems, specifically Windows 10, and Windows 11. It highlights the significance of memory analysis in uncovering vital information such as running processes, dynamic link libraries (DLLs), and command usage. A key component of the research is examining the EPROCESS signature, with notable findings about its behavior in newer versions of Windows. The study presents insights into memory-related artifacts crucial for modern cybercrime investigations, emphasizing how these findings contribute to forensic practices in today's world.*

Key Words: Memory Forensics, RAM Forensics, EPROCESS Signature, Memory Artifacts

1.INTRODUCTION

In a time of rising cybercrime, digital forensics is essential for uncovering vital evidence in investigations. Among the various branches of digital forensics, RAM Forensics has emerged as one of the most important techniques, particularly for identifying volatile data that traditional storage media may not capture. Random Access Memory (RAM) holds transient information such as running processes, dynamic link libraries (DLLs), and executed commands, making it a key source for investigating malicious activities and system behavior.

As operating systems evolve, so do the methods for analyzing their memory structures. This paper focuses on RAM forensics across multiple versions of the Windows operating system, with an emphasis on Windows 10 and Windows 11. A core aspect of this research is the analysis of the EPROCESS signature, a memory structure essential for identifying active processes within the system. By investigating the memory-related artifacts of these newer versions of Windows, this study aims to highlight critical differences in forensic approaches and techniques.

This paper examines the evolution of RAM Forensics across different Windows versions to provide insights into how memory structures have adapted to changing OS architectures. Understanding these changes is essential for forensic investigators, enabling them to adapt their techniques to analyze memory structures accurately and recover critical evidence. This research highlights the significance of keeping pace with operating system advancements to ensure effective RAM Forensics across different Windows versions.

2. LIVE AND OFFLINE FORENSICS

In the realm of digital investigations, both Live and Offline Forensics [1] are integral techniques used to uncover and analyze evidence. Live Forensics is employed to capture volatile data from a running system, which is crucial when the system is found powered on at the scene of a crime. This approach allows investigators to gather information that would otherwise be lost when the system is shut down. By acquiring the contents of Random Access Memory (RAM), Live Forensics provides insights into active processes, open files, network connections, dynamic link libraries (DLLs), and command-line inputs. This volatile data offers a real-time snapshot of the system's activities, revealing evidence related to ongoing operations, such as malware execution or encrypted data in its raw form, along with potential encryption keys. As modern-day cybercrimes often leave little trace on storage media, Live Forensics becomes indispensable for capturing transient data that can link to criminal activities.

In contrast, Offline Forensics involves the traditional approach of creating a bit-stream copy of storage media for later analysis. This method focuses on recovering files, including deleted and overwritten data, and analyzing unallocated space in storage. Through file system reconstruction and techniques such as keyword searches and timeline analysis, Offline Forensics helps in retrieving long-term stored data, which may provide crucial evidence. However, it is limited in accessing volatile information that only exists in active memory. By combining the strengths of both Live and Offline Forensics, investigators are equipped to capture a more complete picture of a suspect's digital footprint. While Live Forensics excels in preserving volatile data from active systems, Offline Forensics ensures thorough analysis of persistent storage. Together, they form a complementary approach, offering a holistic view of digital evidence in cybercrime investigations.

3. RAM FORENSICS

RAM Forensics is a critical component of Live Forensics, focusing on the acquisition and analysis of Random Access Memory (RAM) to gather essential forensic evidence. RAM contains volatile data such as running processes, network connections, encryption keys, open files,

and user credentials, all of which are lost once the system is powered down. This transient information is often crucial to cybercrime investigations, as it provides a real-time snapshot of a system's activities that cannot be found on non-volatile media like hard disks.

The process of RAM Forensics involves two key stages: Memory Acquisition [2] and Memory Analysis. Memory Acquisition captures the entire contents of a system's RAM and stores it in a memory dump file. This is performed while the system is still running to ensure that volatile data is preserved. Once the memory dump is obtained, Memory Analysis is conducted to extract forensic artifacts from the dump file. This analysis can reveal running processes, network activity, open files, and other volatile information that provides insights into the suspect's system behavior.

Given the volatile nature of RAM, RAM Forensics is indispensable in cybercrime investigations. It allows forensic investigators to recover key evidence that would otherwise be lost when the system is powered off. As modern cyberattacks often leave minimal traces on storage media, RAM Forensics ensures that essential evidence related to these attacks is preserved, making it a vital tool for uncovering malicious activities in real-time.

3.1 Structural Analysis

The _EPROCESS structure [3] is a cornerstone of RAM Forensics, residing within the kernel of the Windows operating system and containing critical components essential for understanding process management. Among its key elements are the Directory Table Base (DTB), Process Environment Block (PEB), and Affinity structure. The _EPROCESS structure has over 100 members, many of which are pointers to other structures that provide a comprehensive view of a running process. The DTB is vital for memory management as it aids in translating virtual addresses to physical addresses, crucial during memory analysis. Additionally, the _EPROCESS gives us the Process ID (PID) and Parent Process ID (PPID) of a given process, along with the creation and exit times, allowing forensic investigators to establish timelines of system activities.

The PEB, nested within the _EPROCESS structure, holds invaluable information about the loaded modules, command-line parameters, and various attributes associated with a process. Specifically, the PEB_LDR_DATA structure provides a list of Dynamic Link Libraries (DLLs) used by the process, while the RTL_USER_PROCESS_PARAMETERS structure contains command-line information. This data is instrumental in identifying malicious behaviors, as it reveals the libraries being utilized and the commands executed by the process. Moreover, the HANDLE_TABLE structure within the _EPROCESS gives details of all files opened by the process, further enriching the investigative context.

An important aspect of the _EPROCESS structure is that it contains a pointer to the PEB, facilitating a deeper analysis of the process's characteristics. The DTB value can be located at offset 0x28 in the _KPROCESS struct, which is found at the beginning of the _EPROCESS; up to the latest build version, 0x28 is the position of the DTB. This value is crucial for address translation, aiding investigators in transitioning from virtual to physical addresses. Furthermore, the PEB structure holds an ImageBaseAddress value that points to the first page of the process's content. By analyzing the section table of the Portable Executable (PE) structure on this first page, investigators can identify the locations of other pages, allowing for a comprehensive reconstruction of the program by combining all the pages specified in the section table.This comprehensive analysis of the _EPROCESS structure not only aids in revealing the behaviors of active processes but also highlights potential vulnerabilities and attack vectors present in the system.

```
struct _EPROCESS
{
   struct _KPROCESS Pcb;                                  //0x0
   struct _EX_PUSH_LOCK ProcessLock;                      //0x438
   VOID* UniqueProcessId;                                 //0x440
   struct _LIST_ENTRY ActiveProcessLinks;                 //0x448
   struct _EX_RUNDOWN_REF RundownProtect;                 //0x458
   union
   {
      ULONG Flags2;                                       //0x460
      struct
      {
         ULONG JobNotReallyActive:1;                      //0x460
         ULONG AccountingFolded:1;                        //0x460
         ULONG NewProcessReported:1;                      //0x460
         ULONG ExitProcessReported:1;                     //0x460
         ULONG ReportCommitChanges:1;                     //0x460
         ULONG LastReportMemory:1;                        //0x460
         ULONG ForceWakeCharge:1;                         //0x460
         ULONG CrossSessionCreate:1;                      //0x460
         ULONG NeedsHandleRundown:1;                      //0x460
         ULONG RefTraceEnabled:1;                         //0x460
         ULONG PicoCreated:1;                             //0x460
         ULONG EmptyJobEvaluated:1;                       //0x460
         ULONG DefaultPagePriority:3;                     //0x460
         ULONG PrimaryTokenFrozen:1;                      //0x460
         ULONG ProcessVerifierTarget:1;                   //0x460
```

Fig -1: Format of _EPROCESS Structure

```
struct _KPROCESS
{
   struct _DISPATCHER_HEADER Header;                      //0x0
   struct _LIST_ENTRY ProfileListHead;                    //0x18
   ULONGLONG DirectoryTableBase;                          //0x28
   struct _LIST_ENTRY ThreadListHead;                     //0x30
   ULONG ProcessLock;                                     //0x40
   ULONG ProcessTimerDelay;                               //0x44
   ULONGLONG DeepFreezeStartTime;                         //0x48
   struct _KAFFINITY_EX Affinity;                         //0x50
   ULONGLONG AffinityPadding[12];                         //0xf8
   struct _LIST_ENTRY ReadyListHead;                      //0x158
   struct _SINGLE_LIST_ENTRY SwapListEntry;               //0x168
   volatile struct _KAFFINITY_EX ActiveProcessors;        //0x170
   ULONGLONG ActiveProcessorsPadding[12];                 //0x218
```

Fig -2: Format of _KPROCESS Structure

3.2 The new Methodology

In the field of RAM Forensics, the ability to accurately identify critical structures within memory dumps is paramount for successful investigations. One key focus of this paper is the _EPROCESS signature [4], which varies not only across different operating system versions but also among various builds. Understanding the unique characteristics of the _EPROCESS signature for each version and build of Windows 10 [5] is essential to ensure the precise identification of valid _EPROCESS instances within memory dumps.

Table -1: Eprocess Signature of Various Versions of Computer

Operating System	Major Version	Minor Version	OS Build Version	_EPROCESS Signature
Windows 10	10	0	17134	0x0300B600
Windows 10	10	0	18362	0x0300B800
Windows 10	10	0	18363	0x0300B800
Windows 10	10	0	19041	0x03000000
Windows 11	10	0	22621	0x03000000

Variations in the layout and structure of _EPROCESS can shift between versions due to updates in the Windows kernel. Therefore, forensic tools must be regularly updated to accommodate these changes and ensure that _EPROCESS instances are accurately identified across all relevant Windows builds. This adaptability is key to maintaining the integrity of investigations and ensuring the reliability of evidence extracted from memory dumps.

Fig -3: Screenshot of EPROCESS in memory dump of Windows 10 (19041 OS build).

The proposed methodology begins by systematically scanning the blocks of the memory dump file to locate a valid _EPROCESS signature. Once this signature is identified, investigators can leverage it to efficiently collect a wealth of evidence, including information about running processes, loaded DLLs, and command-line arguments used by those processes. This signature-based approach not only allows forensic analysts to quickly pinpoint relevant processes within the memory dump but also streamlines the overall analysis process. By enabling a focused examination of pertinent data, this methodology enhances the accuracy and effectiveness of RAM Forensics, empowering investigators to construct a detailed picture of system activities and identify potential security threats. Through this structured approach, the methodology significantly contributes to the timely identification and mitigation of cyber threats.

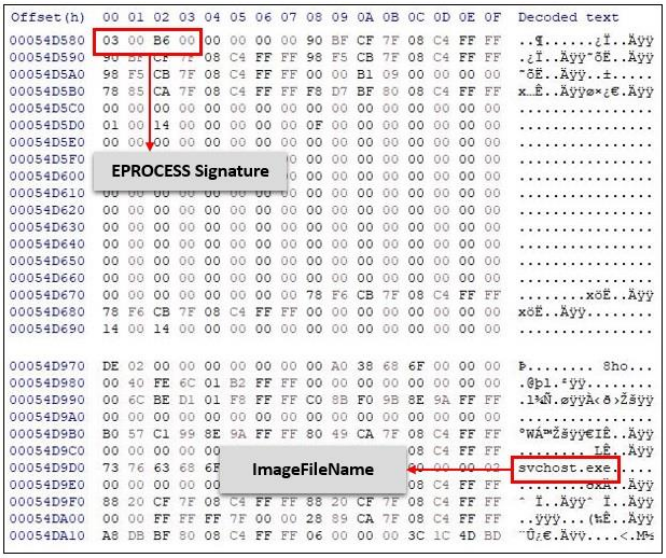

Fig -4: Screenshot of EPROCESS in memory dump of Windows 10 (17134 OS build).

3.3 Extracted Forensic Artifacts

In the realm of RAM Forensics, the journey begins with the crucial task of locating relevant data structures within the memory dump. This initial step is vital for unveiling the intricate web of processes operating within a system. Central to this analysis is the _EPROCESS structure, which serves as a gateway to examining critical process information [6], such as the Process ID (PID), Parent Process ID (PPID), Thread Count, Image Name, and associated DLLs for each running process. This detailed analysis is instrumental in detecting anomalies and tracing the origins of malicious software.

Once the necessary data structures are identified, analysts can delve deeper into the _EPROCESS structure to retrieve essential information about running processes, loaded dynamic link libraries (DLLs) and command-line parameters used by the running processes. Examining the list of DLLs and command-line information allows forensic analysts to

identify potential malicious behaviors, as attackers often exploit specific DLLs or use unusual command-line arguments to execute their activities.

3.4 Forensic Relevance

As cybercrime [7] continues to rise, the relevance of RAM Forensics becomes increasingly significant. With the proliferation of malware and sophisticated attacks, spotting suspicious activity in real time is crucial. After acquiring a memory dump, investigators can extract vital information, such as running processes and loaded DLLs, which aids in identifying potential malware. This proactive approach to analyzing volatile memory allows forensic professionals to respond swiftly to emerging threats, making it an essential practice in modern cybercrime investigations.

By identifying the _EPROCESS signature in the latest Windows versions, forensic analysts can extract essential attributes that facilitate investigating suspicious activities and help establish timelines. This process enables investigators to map out the behavior of malicious entities, identify their origins, and understand their impact on the system.

Sl. No.	Process Name	Process ID	Parent Process ID	Create Time
85	svchost.exe	1992	804	2018/08/06 23:42:01
87	svchost.exe	8108	804	2018/08/06 23:41:55
88	svchost.exe	8708	804	2018/08/06 23:42:04
89	svchost.exe	7328	804	2018/08/02 0:59:59
94	svchost.exe	8560	4824	2018/08/02 1:43:10
100	svchost.exe	6000	804	2018/08/02 1:42:15
101	svchost.exe	2908	804	2018/08/02 0:50:35
103	svchost.exe	6744	804	2018/08/02 0:52:36
106	svchost.exe	6848	804	2018/08/06 23:42:01
111	svchost.exe	800	804	2018/08/02 0:50:29
112	svchost.exe	476	804	2018/08/02 0:50:29
114	svchost.exe	2128	804	2018/08/02 0:50:32

Fig -5: Screenshot of process list of memory dump of Windows 10 (17134 OS build)

The provided Figure 5 depicts a list of svchost.exe processes running on a Windows system. As mentioned, svchost.exe is a common system process that can be used by malware to evade detection. Analyzing the image, we see that most svchost.exe processes have a Parent Process ID (PPID) of 804, which corresponds to the services.exe process. This is expected behavior for legitimate svchost.exe instances. However, there is one process with a PPID of 4824, which is unusual. This discrepancy raises concerns about the legitimacy of this particular svchost.exe process.

Further investigation would involve examining the command line arguments, associated DLLs, and other attributes of the suspect process. If it is found to have unusual or suspicious characteristics, it could be indicative of malicious activity. For instance, the process might be loading unusual DLLs or executing commands that are not typical for a legitimate svchost.exe process. By carefully analyzing these details, forensic investigators can determine whether the process is a genuine system process or a malicious actor attempting to disguise itself as such.

From a Windows 10 memory dump, various key details can be extracted to assist in forensic analysis [8]. Investigators can retrieve information on running processes, DLL lists, and command line usage. Specifically, attributes like the ImageFileName, ProcessID, and ThreadCount provide a snapshot of active processes, while the Memory_ProcessDlls table can reveal critical DLL details, including size and base address. Furthermore, the Memory_CommandUsed section sheds light on the commands executed by processes. By leveraging the _EPROCESS signature, forensic analysts can reconstruct processes and establish a comprehensive understanding of the system's state at the time of the memory capture, facilitating the identification of potential vulnerabilities and attack vectors.

5. CHALLENGES AND FUTURE WORK

The variability of EPROCESS signatures across different operating system versions poses a significant challenge. When a new version is introduced, it is crucial to identify the corresponding EPROCESS signature by analyzing the memory dump. Additionally, if an update is made to an existing system, its internal structure may change, resulting in alterations to the EPROCESS signature. In some cases, the signature value may not appear unique, necessitating further examination of additional EPROCESS data to ensure accurate identification.

Another challenge arises from the variations in dump file extensions and their corresponding offset values, which can change between different dumps. Some dumps may contain extra data, leading to changes in offset and potentially producing different results.

Furthermore, the presence of terminated processes alongside running processes adds complexity to the analysis. The current implementation displays both types of processes and aim to differentiate between them in future versions to provide clearer insights into the system's state..

5. CONCLUSIONS

Cyber Forensics investigators must prioritize the acquisition and analysis of volatile data to secure critical evidence in cybercrime investigations. By focusing on the analysis of volatile data within Random Access Memory (RAM), investigators can gain valuable insights into active processes, network connections, and potential malicious activities. The EPROCESS structure, along with its signature, serves as a vital component in this analysis, enabling the

identification of running processes and facilitating a deeper understanding of system behavior.

However, the challenges posed by differing EPROCESS signatures across operating system versions, variations in memory dump extensions, and the presence of both running and terminated processes necessitate ongoing research and adaptation of forensic methodologies. As the field of RAM Forensics continues to advance, it is essential for investigators to refine their tools and techniques to ensure effective analysis and reliable results. By addressing these challenges, forensic professionals can enhance their capabilities in combating cybercrime and protecting digital assets in a rapidly changing technological landscape.

REFERENCES

[1] S. Dija, J. Ajana, V. Indu, and M. Sabarinath, "Web browser forensics for retrieving searched keywords on the Internet," in 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), 2021, Greater Noida, India, pp. 1664–1668,2021. https://doi.org/10.1109/ICAC3N53548.2021.9725457.

[2] R. Shree, A. K. Shukla, R. P. Pandey, V. Shukla, and D. Bajpai, "Memory forensic: Acquisition and analysis mechanism for operating systems," in Proceedings of the Conference on Materials Today: Proceedings, 2021, Springer, Heidelberg, pp. 1–13, 2021. https://doi.org/10.1016/j.matpr.2021.05.270.

[3] J. Jaina, G. S. Suma, S. Dija, and K. L. Thomas, "Extracting network connections from Windows 7 64-bit physical memory," in 2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), 2015, Madurai, India, pp. 1–4, 2015. https://doi.org/10.1109/ICCIC.2015.7435745.

[4] G. S. Suma, S. Dija, and K. L. Thomas, "A novel methodology for Windows 7 × 64 RAM Forensics," in 2014 IEEE International Conference on Computational Intelligence and Computing Research, 2014, Coimbatore, India, pp. 1–6, 2014. https://doi.org/10.1109/ICCIC.2014.7238400.

[5] D. Kim and T. Shon, "Future of kernel object-based RAM Forensics," in 2023 International Conference on Platform Technology and Service (PlatCon), 2023, Busan, Republic of Korea, pp. 64–66, IEEE, 2023. https://doi.org/10.1109/PlatCon60102.2023.10255186

[6] S. Dija, G. S. Suma, D. D. Gonsalvez, and A. T. Pillai, "Forensic reconstruction of executables from Windows 7 physical memory," in 2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), 2016, Chennai, India, pp. 1–5, IEEE, 2016. https://doi.org/10.1109/ICCIC.2016.7919641.

[7] D. B. Oh, D. Kim, D. Kim, and H. K. Kim, "volGPT: Evaluation on triaging ransomware process in RAM Forensics with Large Language Model," in Forensic Science I nternational: Digital Investigation, 2024, vol. 49, Supplement, pp. 301756, Elsevier, 2024. https://doi.org/10.1016/j.fsidi.2024.301756.

[8] N. Patil and B. B. Meshram, "Extraction of forensic evidences from Windows volatile memory," in Proceedings of the 2nd International Conference for Convergence in Technology (I2CT), 2017, Mumbai, India, pp. 421–425, IEEE, 2017. https://doi.org/10.1109/I2CT.2017.8226164