Live Memory Forensics Investigations: A Comparative Analysis

Irfan Syamsuddin [1],* and Dedy Syamsuar [2]

[1] CAIR Center for Applied ICT Research, Department of Computer and Network Engineering, State Polytechnic of Ujung Pandang, Makassar, Indonesia

[2] Information Systems Department, School of Information Systems, Bina Nusantara University, Jakarta, Indonesia;
Email: dedy.syamsuar@binus.ac.id (D.S.)
*Correspondence: irfans@poliupg.ac.id (I.S.)

***Abstract*—The escalating dependence on information technology for daily activities ensures that cybercrime cases continue unabated. Consequently, the role of cyber forensics investigators is becoming increasingly crucial in addressing the surge of cybercrime incidents. Live forensics investigation, a challenging facet of digital evidence investigation, confronts several limitations. This study focuses on the complexities associated with retrieving digital evidence from volatile memory during live forensics investigations, explicitly comparing the efficacy of extracting digital evidence from DDR2 and DDR3 Random Access Memory (RAM). This study aims to analyze and compare potential variations in evidence acquisition outcomes between the two RAM types by applying three distinct scenarios: identifying registry and network activities, catching malicious codes, and obtaining login passwords on Social Media. The results demonstrate that DDR2 RAM exhibits a lower propensity for concealing digital evidence during live forensics investigations compared to DDR3 RAM. The implications of these findings are discussed, along with suggestions for potential ramifications and avenues for future research.**

***Keywords*—computer forensics, random access memory, DDR2, DDR3, digital evidence, live forensics investigation**

I. Introduction

Two distinct types of cyber forensics investigation are employed for digital evidence extraction: static forensics and live forensics. A static forensics investigation is a conventional approach wherein the forensic analysis is conducted on systems at rest, typically referring to shutdown systems. Conversely, live forensics focuses on investigating volatile memory in a running system, as valuable evidence would be lost once the system is powered off [1, 2], and this evidence is stored temporarily in Random Access Memory (RAM). RAM is a volatile memory that temporarily stores data while the computer runs. As soon as the power is turned off, the data in RAM is lost. In digital forensics, volatile memory is a vital source of information, as it can capture critical data about system activity [2] that might be lost after the system is powered off. Consequently, the transient data residing in RAM holds substantial relevance for forensics, as it captures a comprehensive record of all system activities [3, 4].

Recovering digital evidence from volatile memory poses a significant challenge in cyber forensics investigations due to the ephemeral nature of the evidence upon system shutdown [1, 3, 4]. Live forensics methodologies are designed to extract potential crime-related evidence and other pertinent data residing in memory [3, 5]. Unlike traditional forensic techniques that exclusively operate on offline systems, live forensics enables the acquisition of hidden digital evidence while the system remains operational, including memory activities, network processes, swap files, running system processes, and system information [3–5]. In certain instances, critical evidence pertaining to cyber-attacks is solely found within system memory, such as network connections and account credentials, necessitating live forensics approaches. The use of RAM in digital forensics investigations is rapidly growing due to the need to capture and analyze volatile data. Nonetheless, the rapid development of computer technology has increased the complexity of RAM architecture, making acquiring and analyzing data from RAM more challenging. Therefore, it is essential to continue research in digital forensics to develop new techniques and tools to extract and analyze volatile data from RAM effectively.

The efficacy of live forensics heavily relies on the active state of the computer, as it relies on the data stored in Random Access Memory (RAM). RAM data, also known as volatile or temporary data, is accessible only when the computer is powered on, and any loss of power results in the immediate loss of this data. This volatile data encompasses crucial information such as usernames, passwords, accessed files, modified files, utilized applications, and search keywords [1]. In live forensics analysis, both evidence gathering and analysis occur concurrently. Consequently, when multiple computers are involved in an attack, and investigators seek to discern the state of each system, live forensics emerges as the most suitable option [6], as demonstrated by Rahman and Khan’s proposed direct analysis method [7].

Manuscript received February 3, 2023; revised May 11, 2023; accepted May 24 2023; published September 22, 2023.

doi: 10.12720/jait.14.5.950-959

Recent studies have examined live forensics on the latest RAM types [8–11], while limited attention has been given to earlier generations, such as DDR2 and DDR3. One notable exception is the work by Lindenlauf *et al.* [12], who conducted cold boot attacks on both memory types. Given the continued prevalence and usage of DDR2 and DDR3, a research gap exists regarding the comparative analysis of live memory forensics on these memory types within the current literature. Therefore, the primary objective of this study is to investigate whether additional digital evidence can be obtained from DDR2 and DDR3 through live forensics analysis.

The following are novelties proposed through the study:

(1) Live memory forensics analysis is conducted in virtualization mode using open-source and free forensics applications.

(2) Three scenarios are designed to assess further potential volatile digital evidence from both RAM types.

(3) The results would have practical implications on any potential digital evidence losses in RAM.

This paper is organized as follows. Section II describes the materials and methods used in the study. Section III introduces several scenarios for the investigation, followed by analysis and discussion in Section IV. Finally, we conclude the study in the last section.

II. Research Methodology

The research methodology for conducting the live forensics analysis comparison between DDR2 RAM and DDR3 RAM is illustrated in Fig. 1. The flowchart is used to enhance understanding and provide a clear overview of the stages involved. The following sections explain each stage of the research process.

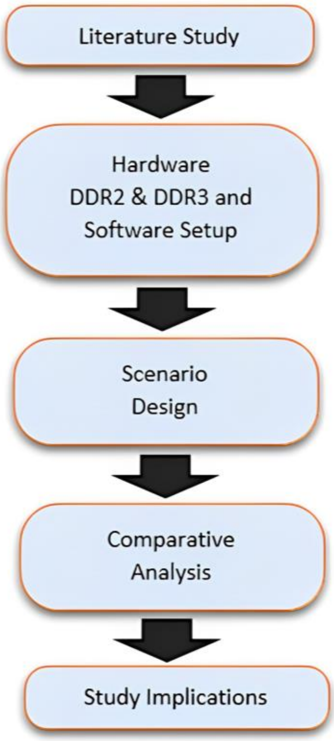


Figure 1. Research method.

A. Literature Study

The initial stage of this research involves conducting an in-depth study of prior research in the field of memory forensics. By examining existing literature, a solid theoretical foundation will be established, and any gaps left by previous studies will be identified, providing the motivation for this current research.

Memory forensics has gained significant attention from both academics and professionals since the 2005 Digital Forensics Workshop (DFRWS) forensics challenge [13]. With the increased memory size and the abundance of volatile digital evidence, forensic examiners face mounting challenges in extracting valid and accurate evidence [14]. Therefore, there is a growing interest in understanding and advancing the field of memory forensics. For this purpose, virtualization techniques are commonly employed in digital forensics studies [15]. This approach involves creating a duplicate copy of the target hard drive on a virtual machine after obtaining the memory image, boot disk, or other relevant system components. By utilizing virtualization, researchers can safely conduct experiments and investigations.

While previous studies in memory forensics have focused on various applications or operating systems, there is a notable gap in the literature regarding comparing different memory types. For instance, Salamh *et al.* [16] conducted a forensic simulation to recover deleted evidence from chat applications on the Android system, but their analysis did not include a comparative study. Similarly, Kazim *et al.* [17] examined digital evidence extracted from a memory dump of Google Hangout in a Windows operating system environment without comparing it to other applications or different memory types.

Although there have been studies that compared instant messaging applications or different operating systems [18], a comparative analysis of forensics analysis was introduced by Thantilage *et al.* [19]. They examined several instant messaging applications to identify potential digital evidence stored in memory. Their comprehensive framework effectively extracted volatile digital evidence and presented it in a legally admissible report. Another study focused on a Linux-based memory forensics examination compared Discord and Slack (Linux-based instant messaging) as a case study [20]. The researchers successfully extracted and presented digital evidence from memory.

Recent studies also focused on new technologies. Holmes and Buchanan [21] conducted live forensic techniques and dictionary attacks to ensure cryptocurrency wallet security. While video streaming is gaining popularity, Murias *et al.* [22] evaluate some tools to retrieve evidence in Android OS. Azzam *et al.* [23] empirically investigated the forensic readiness of Industrial Control Systems (ICS) toward cyberattacks.

The studies mentioned earlier clearly show that they mainly compared the memory forensics analysis by varying the applications, such as instant messaging, or running the same scenario on different operating systems

(Windows and Linux), the specific focus on memory types, such as DDR2 and DDR3, remains largely unexplored in the existing literature. To the best of the authors' knowledge, the only research comparing DDR2 and DDR3 memories was presented in [12]. Since the study only dealt with cold boot attacks, many open research questions might be raised. By investigating the potential variations in evidence extraction capabilities between DDR2 and DDR3 memories, this research intends to shed light on the importance of considering memory types in forensic investigations. The findings of this study will contribute to the broader understanding of memory forensics and inform the development of more effective and accurate forensic techniques.

A comprehensive methodology will be employed to accomplish this research objective, encompassing live forensics analysis of DDR2 and DDR3 memories. The research will systematically compare the evidence extraction results from these two memory types by conducting parallel investigations under similar scenarios. This rigorous analysis will provide empirical evidence to ascertain whether the choice of memory type significantly impacts forensic analysis outcomes. Through this comparative analysis, the research aims to fill the existing gap in the literature by highlighting the importance of considering memory types in memory forensics. The insights gained from this study will advance the field and contribute to developing more robust forensic techniques and methodologies. By bridging this research gap, this study seeks to enhance the effectiveness and accuracy of memory forensics investigations, ultimately assisting forensic examiners in pursuing digital evidence.

B. Hardware and Software Setup

The study requires specific hardware and software components to facilitate the comparative analysis of DDR2 and DDR3 RAM in live forensics analysis. The hardware requirements encompass two PCs equipped with AMD A6-6310 processors, each with 4 GB of DDR2 and DDR3 RAM, respectively. Both systems share a Seagate 500 GB hard drive and utilize the same internet connection. On the software front, Windows 7 Ultimate Operating System serves as the common platform for both PCs.

In order to conduct the memory forensics analysis, it is essential to utilize various software tools that enable the extraction and analysis of digital evidence from the volatile memory of the systems under investigation. The following software applications are indispensable: Volatility-2.4 Standalone, Belkasoft Live RAM Capturer v1.0 32bit, DumpIt v1.3.2, Volatility Framework v2.4, and VMware v10.0.7. These tools provide the necessary functionalities and capabilities to effectively examine and analyze the digital artifacts in the volatile memory of the systems being examined.

DDR2 RAM is a widely adopted memory technology known for its prevalence in computers utilizing Pentium 4 processors or later iterations. Introduced in 2005, DDR2 RAM operates at a voltage of 1.8 volts, thereby offering a balance between power efficiency and performance [24]. Its specifications encompass transfer frequencies ranging from 400MHz to 1066MHz, denoted by standard names such as DDR2-400, DDR2-533, DDR2-677, DDR2-800, and DDR2-1066.

Furthermore, DDR3 RAM was introduced by Intel in late 2007 and represented a further advancement in-memory technology. Operating at a reduced voltage of 1.5 volts, DDR3 RAM exhibits notable improvements in reading speeds compared to its predecessors [24]. Its specifications encompass transfer frequencies of up to 2133 MHz, rendering it more efficient than DDR2 RAM. However, it should be noted that DDR3 RAM tends to be relatively more expensive and boasts superior latency. Apart from differences in specifications in DDR2 RAM and DDR3 RAM, the hardware or shape is also different, as presented in Fig. 2. Moreover, it excels in transferring I/O data, exhibiting a remarkable eight-fold increase in data rate per memory cell.

By employing the hardware mentioned earlier and software configurations, this study explores the digital evidence extraction capabilities of DDR2 and DDR3 RAM through live forensics analysis. These RAM types have been deliberately chosen due to their widespread adoption in modern computing systems.

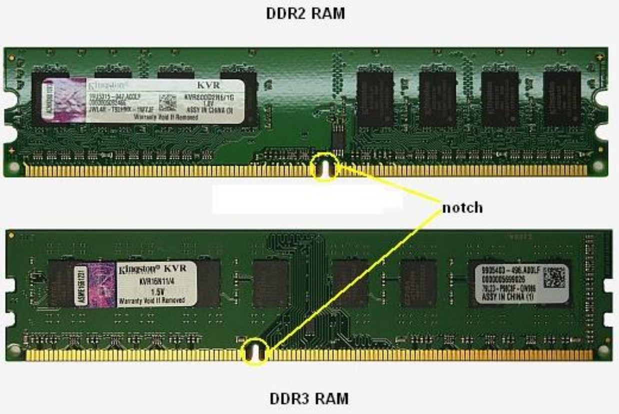


Figure 2. Physical look of DDR2 and DDR3 [21].

The physical differences between DDR2 and DDR3 RAM are essential in this study. As shown in Fig. 2, the two types of RAM differ in several ways [24]. The DDR2 notch is located slightly toward the right side, while the DDR3 notch is located slightly more toward the center of the memory module board. Additionally, the latest DDR3 notch is located slightly to the left, opposite the location of the first DDR notch. DDR2 and DDR3 have smaller, denser pins with 240 pins (120 pins on each side).

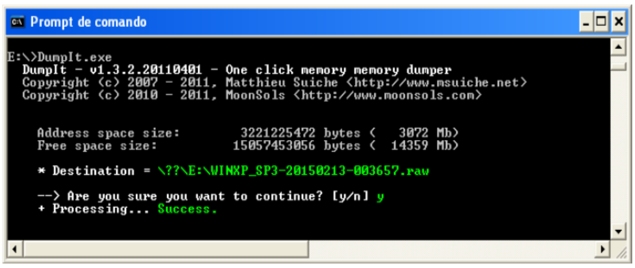
This study used two PCs with different mainboards: one supporting DDR2 and the other supporting DDR3. DDR3 RAM has several advantages over DDR2 RAM, including the ability to transfer I/O data at eight times the data rate contained in the memory cell. This benefit allows for a higher bus price and a higher price than previous peak memory technology. The DDR3 RAM standard also allows for a chip capacity of 512 megabits up to eight gigabits, which enables a maximum memory module size of 16 gigabytes.

Another advantage of DDR3 RAM is its reduced power consumption of more than 30% compared to

DDR2 and DDR1 RAM modules due to the RAM's supply voltage of less than 1.5 V. This contrasts with DDR2 and DDR1, which have supply voltages of 1.8 V or 2.5 V. This supply voltage works well with the 90-nanometer technology used in DDR3 RAM [24]. These differences between DDR2 and DDR3 RAM are essential in analyzing digital evidence from volatile memory.

In this study, open-source and freely available tools for forensic memory analysis are utilized. These tools have been widely employed in fundamental forensic analysis to acquire and analyze digital evidence [4, 7, 8]. Two commonly used memory forensics applications, DumpIt and BelkaSoft RamCapture, are employed in this research. DumpIt is a tool that combines the functionality of win32dd and win64dd into a single executable. By double-clicking the DumpIt executable, the tool is executed, and it captures a snapshot of the physical memory, storing it in the same folder as the DumpIt execution [25].

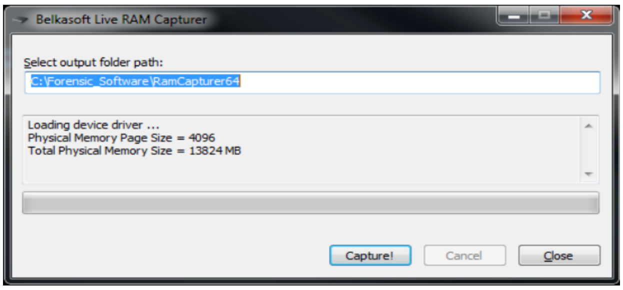
DumpIt provides a user-friendly approach to obtaining memory images from Windows systems, even in cases where the investigator is not physically present at the target computer. Its ease of use makes it accessible even to non-technical users. However, it may not be suitable for all scenarios, but it simplifies memory retrieval in many situations [25]. Fig. 3 illustrates an example of the DumpIt display.



```
E:\>DumpIt.exe
  DumpIt - v1.3.2.20110401 - One click memory memory dumper
  Copyright (c) 2007 - 2011, Matthieu Suiche <http://www.msuiche.net>
  Copyright (c) 2010 - 2011, MoonSols <http://www.moonsols.com>


    Address space size:         3221225472 bytes (   3072 Mb)
    Free space size:           15057453056 bytes (  14359 Mb)

    * Destination = \??\E:\WINXP_SP3-20150213-003657.raw

    --> Are you sure you want to continue? [y/n] y
    + Processing... Success.
```

Figure 3. DumpIt.

Similarly, Belkasoft RamCapture has established itself as a powerful tool for acquiring memory in Windows operating systems since the era of Windows XP [26]. It offers many outstanding features for effectively managing the system's memory, including robust anti-debugging and anti-memory dumping capabilities. The 64-bit live RAM capturer is meticulously crafted by combining two essential files, namely RamCapture64.exe and RamCaptureDriver64.sys, to acquire and extract memory data [26] seamlessly as seen in Fig. 4.



Select output folder path:
C:\Forensic_Software\RamCapturer64

```
Loading device driver ...
Physical Memory Page Size = 4096
Total Physical Memory Size = 13824 MB
```

Figure 4. Belkasoft RamCapture.

C. Scenario Design

At this stage, the step taken is to design a trial scenario designed in such a way as to resemble conditions that might occur in the field. The scenario is then presented to Random Access Memory for later data extraction examinations in these conditions. This study proposes three scenarios to perform live memory forensics on DDR2 and DDR3 memories.

1) Scenario 1: Registry and network activity check

Fig. 5 presents the structure of the first scenario. In this scenario, we used two computers with DDR2 and DDR3 RAM. We examined two pieces of evidence: the number of registry handles and network activities stored in both volatile memories. To analyze the first scenario, we followed the steps outlined below:

- The user boots each computer and executes the Windows operating system without launching any startup applications. Then, we acquire volatile data in RAM using the DumpIt and Belkasoft RamCapture tools.
- We run the first application, specifically the Winrar application, and proceed to acquire volatile data in RAM using the DumpIt and Belkasoft RamCapture tools.
- We execute the second application, resulting in the presence of two running applications: Winrar and Media Player Classic. We then acquire RAM data using the DumpIt and Belkasoft RamCapture tools.
- We launch the third application, which leads to three running applications: Winrar, Media Player Classic, and AIMP Player. We acquire RAM data using the DumpIt and Belkasoft RamCapture tools.
- The fourth application is run, resulting in four running applications: Winrar, Media Player Classic, AIMP Player, and Foxit Reader. We acquire RAM data using the DumpIt and Belkasoft RamCapture tools.
- We proceed to execute the fifth application, leading to five running applications: Winrar, Media Player Classic, AIMP Player, Foxit Reader, and Google Chrome. We acquire RAM data using the DumpIt and Belkasoft RamCapture tools.
- Lastly, we run the sixth application, resulting in six running applications: Winrar, Media Player Classic, AIMP Player, Foxit Reader, Google Chrome, and Mozilla Firefox. We acquire RAM data using the DumpIt and Belkasoft RamCapture tools.

Critical parameters in the first scenario are digital evidence regarding registry and network activities during the applications' running.

2) Scenario 2: Catching malicious codes

Fig. 6 shows the structure of the second scenario. There are two types of malicious codes applied in this scenario. They are Malware Explorer [27] and Zeus Malware [28]. Considering the high risk of the second scenario, all processes are carried out using VMware virtualization to prevent unwanted errors from occurring [15, 29].

After running the malware, the volatile data in DDR2 and DDDR3 RAM are acquired using DumpIt and Belkasoft RamCapture and saved as images. Furthermore, malware analysis is conducted on the image memory using the Volatility memory forensics tool [30]. Critical parameters in the second scenario are digital evidence identifying signatures of two malicious codes.

3) Scenario 3: Obtaining password login social media

In the last scenario (see Fig. 7), two popular browsers (Google Chrome and Mozilla Firefox) are used to perform login on several social media websites. The process concludes by logging out from both social media platforms. The aim is to find out digital evidence in memory related to user credentials in social media applications. Similarly, volatile data in RAM is acquired using DumpIt and Belkasoft tools.

After collecting all digital evidence from DDR2 memory for the three scenarios using the first computer, we applied the same procedures to the second computer with DDR3 memory. We then analyzed the data from both DDR2 and DDR3 in the next section.

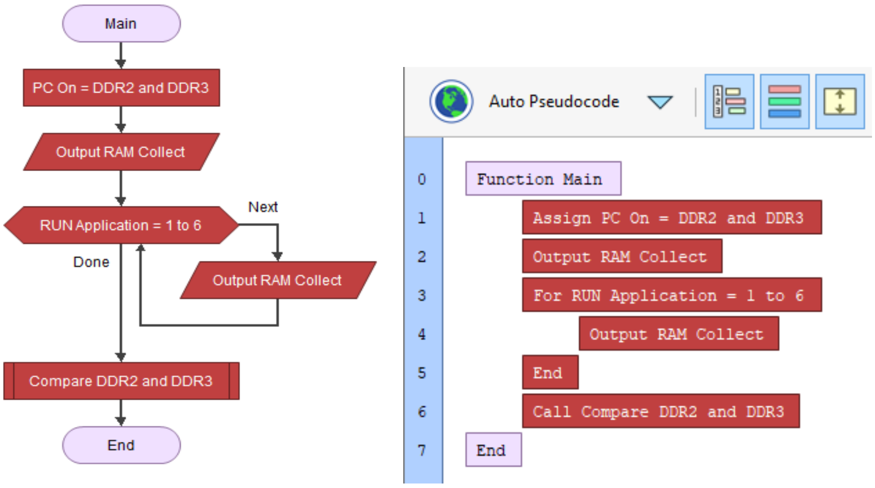
Critical parameters in the last scene are digital evidence that shows the username and password of both social media users.

D. Comparative Analysis

At this stage, we compare the volatile digital evidence obtained from DDR2 and DDR3 in the first scenario. We also conduct a comprehensive comparative analysis in the second and third scenarios, examining the differences and similarities between DDR2 and DDR3 memory.

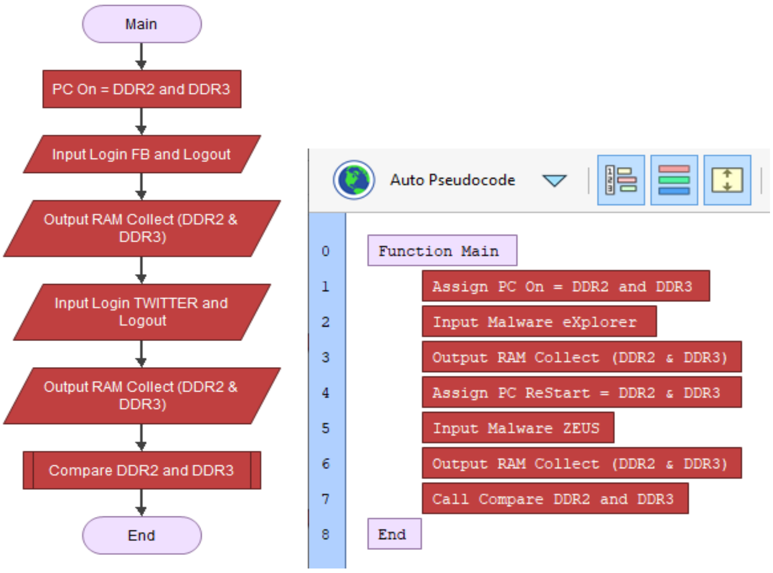
E. Study Implication

We present the study implications based on our previous stage analysis. These implications highlight the significance of our findings and provide insights into the practical implications, theoretical contributions, and potential future research directions in memory forensics.



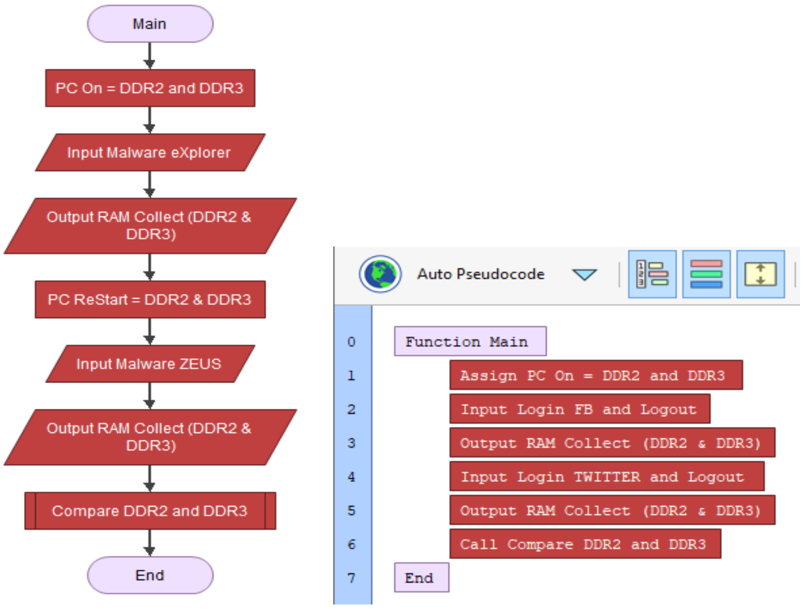
```
Function Main
    Assign PC On = DDR2 and DDR3
    Output RAM Collect
    For RUN Application = 1 to 6
        Output RAM Collect
    End
    Call Compare DDR2 and DDR3
End
```

Figure 5. Scenario 1 (flowchart and pseudocode).


Main
PC On = DDR2 and DDR3
Input Login FB and Logout
Output RAM Collect (DDR2 & DDR3)
Input Login TWITTER and Logout
Output RAM Collect (DDR2 & DDR3)
Compare DDR2 and DDR3
End
Auto Pseudocode

```
Function Main
    Assign PC On = DDR2 and DDR3
    Input Malware eXplorer
    Output RAM Collect (DDR2 & DDR3)
    Assign PC ReStart = DDR2 & DDR3
    Input Malware ZEUS
    Output RAM Collect (DDR2 & DDR3)
    Call Compare DDR2 and DDR3
End
```

Figure 6. Scenario 2 (flowchart and pseudocode).



```
Function Main
    Assign PC On = DDR2 and DDR3
    Input Login FB and Logout
    Output RAM Collect (DDR2 & DDR3)
    Input Login TWITTER and Logout
    Output RAM Collect (DDR2 & DDR3)
    Call Compare DDR2 and DDR3
End
```

Figure 7. Scenario 3 (flowchart and pseudocode).

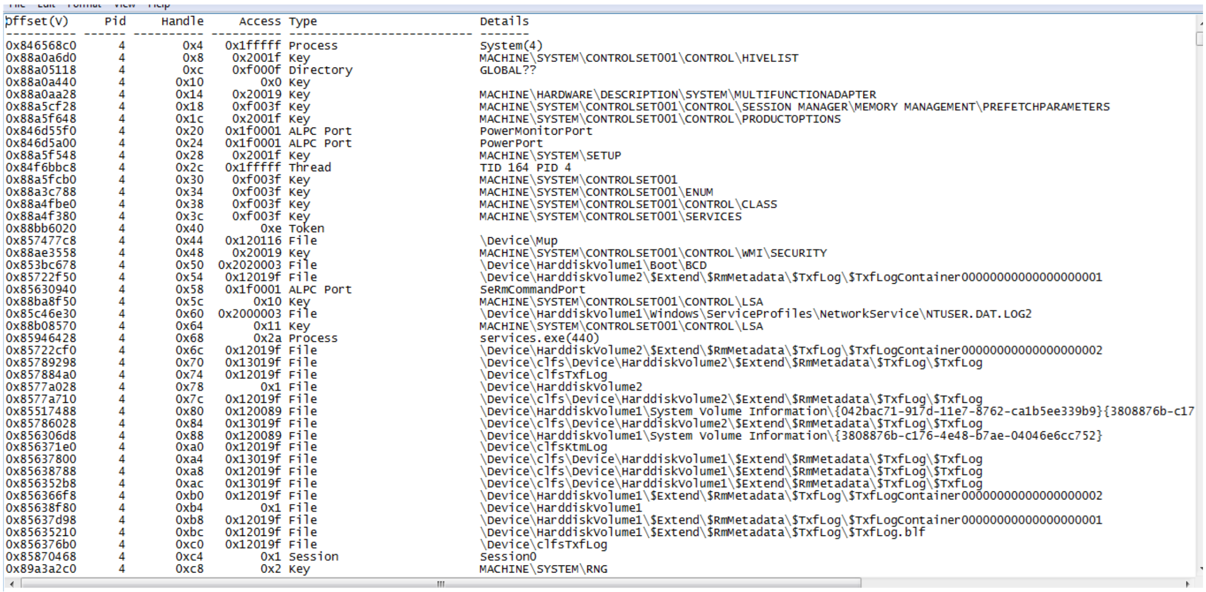
III. Results and Discussion

This section presents the results and analysis from all three scenarios conducted on DDR2 and DDR3 memories.

A. Registry and Network Activity Check

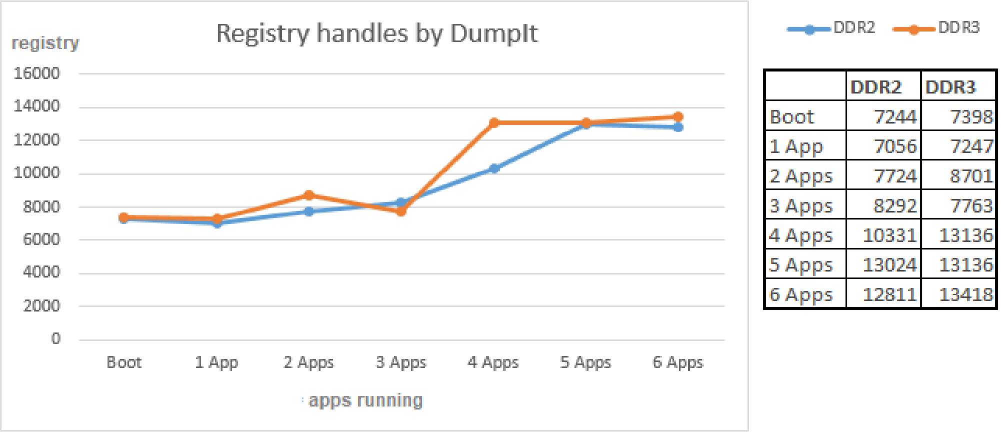
In this scenario, both computers equipped with DDR2 RAM and DDR3 RAM sequentially run multiple applications. DumpIt and Belkasoft RamCapture tools were deployed to capture and store the associated running process in the memory. The first scenario aims to capture two volatile primary pieces of evidence: the number of registry handles and network activities, which are analyzed from the perspectives of DumpIt and Belkasoft RamCapture.

Fig. 8 illustrates an example of registry handles captured in RAM, including offset, PID, handle, access type, and details. The results for the number of registry handles obtained from DumpIt and Belkasoft RamCapture are presented in Fig. 9 (DumpIt) and Fig. 10 (Belkasoft RamCapture). Generally, the number of registry handles increases with the growing number of running applications for DDR2 and DDR3. However, it is noteworthy that DDR3 consistently exhibits a higher number of registry handles compared to DDR2, as observed in both DumpIt and Belkasoft RamCapture results. For instance, when running five applications, DumpIt detected 13,024 registry handles on DDR2 and 13,136 on DDR3 (Fig. 9). Similarly, Belkasoft RamCapture identified 12,438 registry handles on DDR2 and 13,425 DDR3 (Fig. 10).



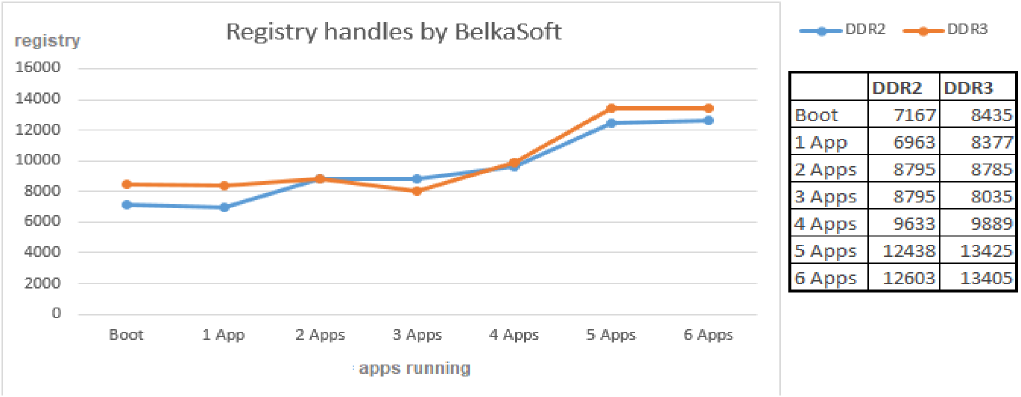
```
Offset(V)     Pid     Handle     Access Type                         Details
---------- ------ ---------- ---------- ---------------------------- -------
0x846568c0      4        0x4   0x1fffff Process                      System(4)
0x88a0a6d0      4        0x8    0x2001f Key                          MACHINE\SYSTEM\CONTROLSET001\CONTROL\HIVELIST
0x88a05118      4        0xc    0xf000f Directory                    GLOBAL??
0x88a0a440      4       0x10        0x0 Key
0x88a0aa28      4       0x14    0x20019 Key                          MACHINE\HARDWARE\DESCRIPTION\SYSTEM\MULTIFUNCTIONADAPTER
0x88a5cf28      4       0x18    0xf003f Key                          MACHINE\SYSTEM\CONTROLSET001\CONTROL\SESSION MANAGER\MEMORY MANAGEMENT\PREFETCHPARAMETERS
0x88a5f648      4       0x1c    0x2001f Key                          MACHINE\SYSTEM\CONTROLSET001\CONTROL\PRODUCTOPTIONS
0x846d55f0      4       0x20   0x1f0001 ALPC Port                    PowerMonitorPort
0x846d5a00      4       0x24   0x1f0001 ALPC Port                    PowerPort
0x88a5f548      4       0x28    0x2001f Key                          MACHINE\SYSTEM\SETUP
0x84f6bbc8      4       0x2c   0x1fffff Thread                       TID 164 PID 4
0x88a5fcb0      4       0x30    0xf003f Key                          MACHINE\SYSTEM\CONTROLSET001
0x88a3c788      4       0x34    0xf003f Key                          MACHINE\SYSTEM\CONTROLSET001\ENUM
0x88a4fbe0      4       0x38    0xf003f Key                          MACHINE\SYSTEM\CONTROLSET001\CONTROL\CLASS
0x88a4f380      4       0x3c    0xf003f Key                          MACHINE\SYSTEM\CONTROLSET001\SERVICES
0x88bb6020      4       0x40        0xe Token
0x857477c8      4       0x44   0x120116 File                         \Device\Mup
0x88ae3558      4       0x48    0x20019 Key                          MACHINE\SYSTEM\CONTROLSET001\CONTROL\WMI\SECURITY
0x853bc678      4       0x50  0x2020003 File                         \Device\HarddiskVolume1\Boot\BCD
0x85722f50      4       0x54   0x12019f File                         \Device\HarddiskVolume2\$Extend\$RmMetadata\$TxfLog\$TxfLogContainer00000000000000000001
0x85630940      4       0x58   0x1f0001 ALPC Port                    SeRmCommandPort
0x88ba8f50      4       0x5c       0x10 Key                          MACHINE\SYSTEM\CONTROLSET001\CONTROL\LSA
0x85c46e30      4       0x60  0x2000003 File                         \Device\HarddiskVolume1\Windows\ServiceProfiles\NetworkService\NTUSER.DAT.LOG2
0x88b08570      4       0x64       0x11 Key                          MACHINE\SYSTEM\CONTROLSET001\CONTROL\LSA
0x85946428      4       0x68       0x2a Process                      services.exe(440)
0x85722cf0      4       0x6c   0x12019f File                         \Device\HarddiskVolume2\$Extend\$RmMetadata\$TxfLog\$TxfLogContainer00000000000000000002
0x85789298      4       0x70   0x13019f File                         \Device\clfs\Device\HarddiskVolume2\$Extend\$RmMetadata\$TxfLog\$TxfLog
0x857884a0      4       0x74   0x12019f File                         \Device\clfsTxfLog
0x8577a028      4       0x78        0x1 File                         \Device\HarddiskVolume2
0x8577a710      4       0x7c   0x12019f File                         \Device\clfs\Device\HarddiskVolume2\$Extend\$RmMetadata\$TxfLog\$TxfLog
0x85517488      4       0x80   0x120089 File                         \Device\HarddiskVolume1\System Volume Information\{042bac71-917d-11e7-8762-ca1b5ee339b9}{3808876b-c17
0x85786028      4       0x84   0x13019f File                         \Device\clfs\Device\HarddiskVolume2\$Extend\$RmMetadata\$TxfLog\$TxfLog
0x856306d8      4       0x88   0x120089 File                         \Device\HarddiskVolume1\System Volume Information\{3808876b-c176-4e48-b7ae-04046e6cc752}
0x856371e0      4       0xa0   0x12019f File                         \Device\clfsKtmLog
0x85637800      4       0xa4   0x13019f File                         \Device\clfs\Device\HarddiskVolume1\$Extend\$RmMetadata\$TxfLog\$TxfLog
0x85638788      4       0xa8   0x12019f File                         \Device\clfs\Device\HarddiskVolume1\$Extend\$RmMetadata\$TxfLog\$TxfLog
0x856352b8      4       0xac   0x13019f File                         \Device\clfs\Device\HarddiskVolume1\$Extend\$RmMetadata\$TxfLog\$TxfLog
0x856366f8      4       0xb0   0x12019f File                         \Device\HarddiskVolume1\$Extend\$RmMetadata\$TxfLog\$TxfLogContainer00000000000000000002
0x85638f80      4       0xb4        0x1 File                         \Device\HarddiskVolume1
0x85637d98      4       0xb8   0x12019f File                         \Device\HarddiskVolume1\$Extend\$RmMetadata\$TxfLog\$TxfLogContainer00000000000000000001
0x85635210      4       0xbc   0x12019f File                         \Device\HarddiskVolume1\$Extend\$RmMetadata\$TxfLog\$TxfLog.blf
0x856376b0      4       0xc0   0x12019f File                         \Device\clfsTxfLog
0x85870468      4       0xc4        0x1 Session                      Session0
0x89a3a2c0      4       0xc8        0x2 Key                          MACHINE\SYSTEM\RNG
```

Figure 8. Registry handles.



	DDR2	DDR3
Boot	7244	7398
1 App	7056	7247
2 Apps	7724	8701
3 Apps	8292	7763
4 Apps	10331	13136
5 Apps	13024	13136
6 Apps	12811	13418

Figure 9. Registry handles by DumpIt.

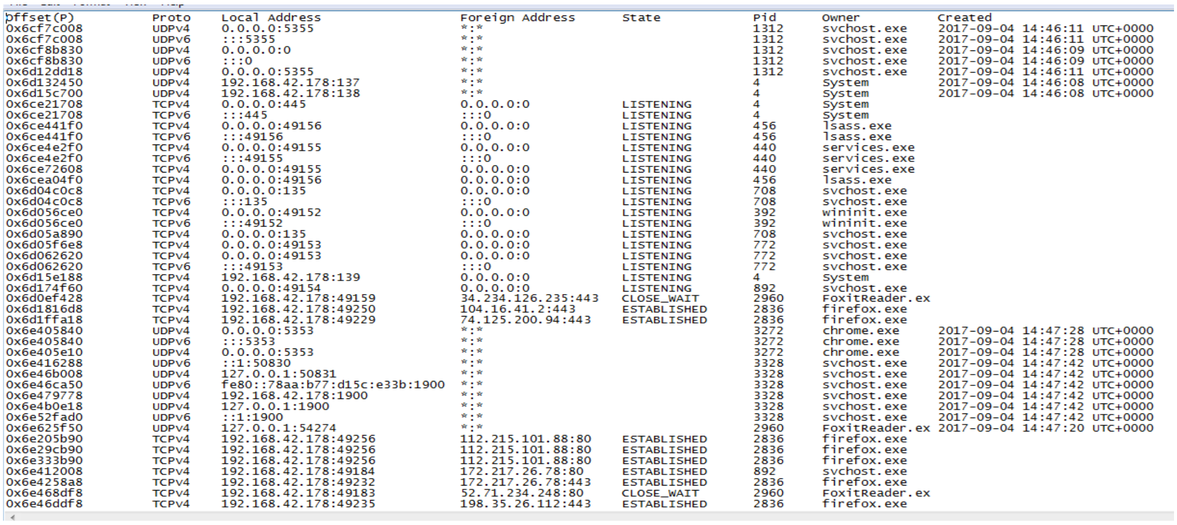


	DDR2	DDR3
Boot	7167	8435
1 App	6963	8377
2 Apps	8795	8785
3 Apps	8795	8035
4 Apps	9633	9889
5 Apps	12438	13425
6 Apps	12603	13405

Figure 10. Registry handles by BelkaSoft.

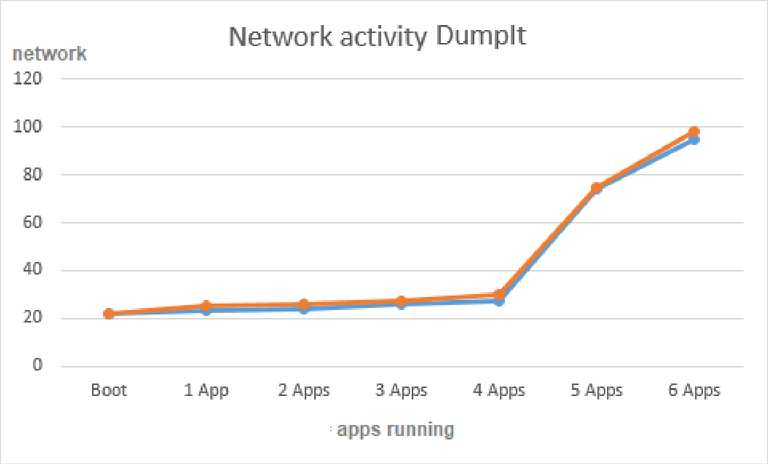
Additionally, network activity is another crucial data to be examined in the first scenario. An example of network activity captured in RAM is depicted in Fig. 11, providing information about established network connections between the PC and external networks. This feature includes offset, PID, handle, access type, and details.

Figs. 12 and 13 show the patterns of network activity observed through DumpIt and Belkasoft RamCapture. No significant network activity was detected in DDR2 and DDR3 in up to four running applications. However, a substantial increase in network activity is observed when the fifth application is executed, followed by the sixth application. DDR3 exhibits a slightly higher number of network activities compared to DDR2, as reported by both DumpIt and Belkasoft RamCapture.



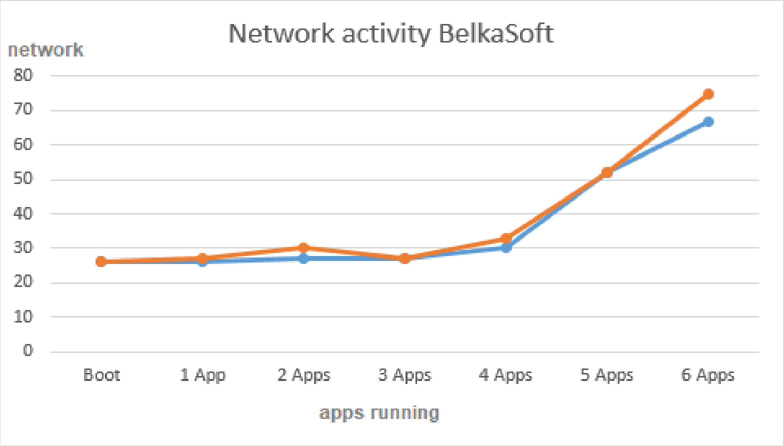
```
Offset(P)         Proto    Local Address                 Foreign Address     State            Pid      Owner          Created
0x6cf7c008        UDPv4    0.0.0.0:5355                  *:*                                  1312     svchost.exe    2017-09-04 14:46:11 UTC+0000
0x6cf7c008        UDPv6    :::5355                       *:*                                  1312     svchost.exe    2017-09-04 14:46:11 UTC+0000
0x6cf8b830        UDPv4    0.0.0.0:0                     *:*                                  1312     svchost.exe    2017-09-04 14:46:09 UTC+0000
0x6cf8b830        UDPv6    :::0                          *:*                                  1312     svchost.exe    2017-09-04 14:46:09 UTC+0000
0x6d12dd18        UDPv4    0.0.0.0:5355                  *:*                                  1312     svchost.exe    2017-09-04 14:46:11 UTC+0000
0x6d132450        UDPv4    192.168.42.178:137            *:*                                  4        System         2017-09-04 14:46:08 UTC+0000
0x6d15c700        UDPv4    192.168.42.178:138            *:*                                  4        System         2017-09-04 14:46:08 UTC+0000
0x6ce21708        TCPv4    0.0.0.0:445                   0.0.0.0:0           LISTENING        4        System
0x6ce21708        TCPv6    :::445                        :::0                LISTENING        4        System
0x6ce441f0        TCPv4    0.0.0.0:49156                 0.0.0.0:0           LISTENING        456      lsass.exe
0x6ce441f0        TCPv6    :::49156                      :::0                LISTENING        456      lsass.exe
0x6ce4e2f0        TCPv4    0.0.0.0:49155                 0.0.0.0:0           LISTENING        440      services.exe
0x6ce4e2f0        TCPv6    :::49155                      :::0                LISTENING        440      services.exe
0x6ce72608        TCPv4    0.0.0.0:49155                 0.0.0.0:0           LISTENING        440      services.exe
0x6cea04f0        TCPv4    0.0.0.0:49156                 0.0.0.0:0           LISTENING        456      lsass.exe
0x6d04c0c8        TCPv4    0.0.0.0:135                   0.0.0.0:0           LISTENING        708      svchost.exe
0x6d04c0c8        TCPv6    :::135                        :::0                LISTENING        708      svchost.exe
0x6d056ce0        TCPv4    0.0.0.0:49152                 0.0.0.0:0           LISTENING        392      wininit.exe
0x6d056ce0        TCPv6    :::49152                      :::0                LISTENING        392      wininit.exe
0x6d05a890        TCPv4    0.0.0.0:135                   0.0.0.0:0           LISTENING        708      svchost.exe
0x6d05f6e8        TCPv4    0.0.0.0:49153                 0.0.0.0:0           LISTENING        772      svchost.exe
0x6d062620        TCPv4    0.0.0.0:49153                 0.0.0.0:0           LISTENING        772      svchost.exe
0x6d062620        TCPv6    :::49153                      :::0                LISTENING        772      svchost.exe
0x6d15e188        TCPv4    192.168.42.178:139            0.0.0.0:0           LISTENING        4        System
0x6d174f60        TCPv4    0.0.0.0:49154                 0.0.0.0:0           LISTENING        892      svchost.exe
0x6d0ef428        TCPv4    192.168.42.178:49159          34.234.126.235:443  CLOSE_WAIT       2960     FoxitReader.ex
0x6d1816d8        TCPv4    192.168.42.178:49250          104.16.41.2:443     ESTABLISHED      2836     firefox.exe
0x6d1ffa18        TCPv4    192.168.42.178:49229          74.125.200.94:443   ESTABLISHED      2836     firefox.exe
0x6e405840        UDPv4    0.0.0.0:5353                  *:*                                  3272     chrome.exe     2017-09-04 14:47:28 UTC+0000
0x6e405840        UDPv6    :::5353                       *:*                                  3272     chrome.exe     2017-09-04 14:47:28 UTC+0000
0x6e405e10        UDPv4    0.0.0.0:5353                  *:*                                  3272     chrome.exe     2017-09-04 14:47:28 UTC+0000
0x6e416288        UDPv6    ::1:50830                     *:*                                  3328     svchost.exe    2017-09-04 14:47:42 UTC+0000
0x6e46b008        UDPv4    127.0.0.1:50831               *:*                                  3328     svchost.exe    2017-09-04 14:47:42 UTC+0000
0x6e46ca50        UDPv6    fe80::78aa:b77:d15c:e33b:1900 *:*                                  3328     svchost.exe    2017-09-04 14:47:42 UTC+0000
0x6e479778        UDPv4    192.168.42.178:1900           *:*                                  3328     svchost.exe    2017-09-04 14:47:42 UTC+0000
0x6e4b0e18        UDPv4    127.0.0.1:1900                *:*                                  3328     svchost.exe    2017-09-04 14:47:42 UTC+0000
0x6e52fad0        UDPv6    ::1:1900                      *:*                                  3328     svchost.exe    2017-09-04 14:47:42 UTC+0000
0x6e625f50        UDPv4    127.0.0.1:54274               *:*                                  2960     FoxitReader.ex 2017-09-04 14:47:20 UTC+0000
0x6e205b90        TCPv4    192.168.42.178:49256          112.215.101.88:80   ESTABLISHED      2836     firefox.exe
0x6e29cb90        TCPv4    192.168.42.178:49256          112.215.101.88:80   ESTABLISHED      2836     firefox.exe
0x6e333b90        TCPv4    192.168.42.178:49256          112.215.101.88:80   ESTABLISHED      2836     firefox.exe
0x6e412008        TCPv4    192.168.42.178:49184          172.217.26.78:80    ESTABLISHED      892      svchost.exe
0x6e4258a8        TCPv4    192.168.42.178:49232          172.217.26.78:443   ESTABLISHED      2836     firefox.exe
0x6e468df8        TCPv4    192.168.42.178:49183          52.71.234.248:80    CLOSE_WAIT       2960     FoxitReader.ex
0x6e46ddf8        TCPv4    192.168.42.178:49235          198.35.26.112:443   ESTABLISHED      2836     firefox.exe
```

Figure 11. Network activity example.



	DDR2	DDR3
Boot	22	22
1 App	23	25
2 Apps	24	26
3 Apps	26	27
4 Apps	27	30
5 Apps	74	75
6 Apps	95	98

Figure 12. Network activity by DumpIt.

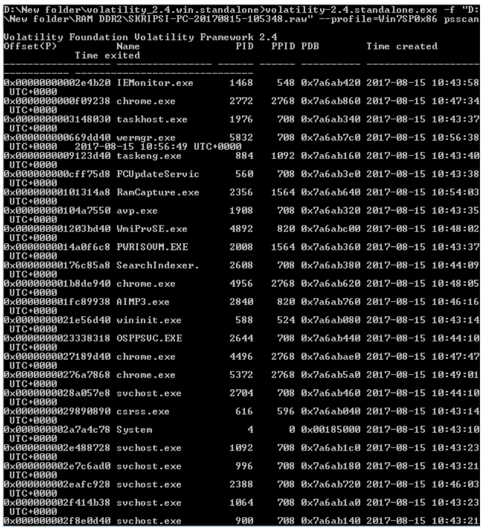


	DDR2	DDR3
Boot	26	26
1 App	26	27
2 Apps	27	30
3 Apps	27	27
4 Apps	30	33
5 Apps	52	52
6 Apps	67	75

Figure 13. Network activity by BelkaSoft.

B. Catching Malicious Codes

After acquiring data in the form of raw images of both DDR2 and DDR3 through VMware, the data is then analyzed with Volatility Framework, a specific forensics tool to identify the existence of malware artifacts (see Fig. 14).



```
D:\New folder\volatility_2.4.win.standalone>volatility-2.4.standalone.exe -f "D:
\New folder\RAM DDR2\SKRIPSI-PC-20170815-105348.raw" --profile=Win7SP0x86 psscan

Volatility Foundation Volatility Framework 2.4
Offset(P)          Name                PID   PPID PDB        Time created
            Time exited
------------------ ---------------- ------ ------ ---------- --------------------
---------- ------------------------------
0x000000000002e4b20 IEMonitor.exe      1468    548 0x7a6ab420 2017-08-15 10:43:58
 UTC+0000
0x0000000000f09238 chrome.exe          2772   2768 0x7a6ab860 2017-08-15 10:47:34
 UTC+0000
0x0000000003148030 taskhost.exe        1976    708 0x7a6ab340 2017-08-15 10:43:37
 UTC+0000
0x000000000669dd40 wermgr.exe          5832    708 0x7a6ab7c0 2017-08-15 10:56:38
 UTC+0000   2017-08-15 10:56:49 UTC+0000
0x0000000009123d40 taskeng.exe          884   1092 0x7a6ab160 2017-08-15 10:43:40
 UTC+0000
0x000000000cff75d8 FCUpdateServic       560    708 0x7a6ab3e0 2017-08-15 10:43:38
 UTC+0000
0x00000000101314a8 RamCapture.exe      2356   1564 0x7a6ab640 2017-08-15 10:54:03
 UTC+0000
0x00000000104a7550 avp.exe             1908    708 0x7a6ab320 2017-08-15 10:43:35
 UTC+0000
0x000000001203bd40 WmiPrvSE.exe        4892    820 0x7a6abc00 2017-08-15 10:48:02
 UTC+0000
0x0000000014a0f6c8 PWRISOVM.EXE        2008   1564 0x7a6ab360 2017-08-15 10:43:37
 UTC+0000
0x00000000176c85a8 SearchIndexer.      2608    708 0x7a6ab380 2017-08-15 10:44:09
 UTC+0000
0x000000001b8de940 chrome.exe          4956   2768 0x7a6ab620 2017-08-15 10:48:05
 UTC+0000
0x000000001fc89938 AIMP3.exe           2840    820 0x7a6ab760 2017-08-15 10:46:16
 UTC+0000
0x0000000021e56d40 wininit.exe          588    524 0x7a6ab080 2017-08-15 10:43:14
 UTC+0000
0x0000000023338318 OSPPSVC.EXE         2644    708 0x7a6ab440 2017-08-15 10:44:10
 UTC+0000
0x0000000027189d40 chrome.exe          4496   2768 0x7a6abae0 2017-08-15 10:47:47
 UTC+0000
0x00000000276a7868 chrome.exe          5372   2768 0x7a6ab5a0 2017-08-15 10:49:01
 UTC+0000
0x0000000028a057e8 svchost.exe         2704    708 0x7a6ab460 2017-08-15 10:44:10
 UTC+0000
0x0000000029890890 csrss.exe            616    596 0x7a6ab040 2017-08-15 10:43:14
 UTC+0000
0x000000002a7a4c78 System                 4      0 0x00185000 2017-08-15 10:43:10
 UTC+0000
0x000000002e488728 svchost.exe         1092    708 0x7a6ab1c0 2017-08-15 10:43:23
 UTC+0000
0x000000002e7c6ad0 svchost.exe          996    708 0x7a6ab180 2017-08-15 10:43:21
 UTC+0000
0x000000002eafc928 svchost.exe         2388    708 0x7a6ab720 2017-08-15 10:46:03
 UTC+0000
0x000000002f414b38 svchost.exe         1064    708 0x7a6ab1a0 2017-08-15 10:43:23
 UTC+0000
0x000000002f8e0d40 svchost.exe          900    708 0x7a6ab140 2017-08-15 10:43:21
```

Figure 14. Volatility framework forensics tool.

Based on the analysis, Explorer and Zeus malware could be correctly identified in DDR2 and DDR3 raw images previously obtained using DumpIt and BelkaSoft. Table I shows the result of the second scenario for both DDR2 and DDR3.

TABLE I. THE RESULTS OF THE SECOND SCENARIO

	Memory	Explorer Malware	Zeus Malware
DumpIt	DDR2 RAM	√	√
	DDR3 RAM	√	√
BelkaSoft	DDR2 RAM	√	√
	DDR3 RAM	√	√

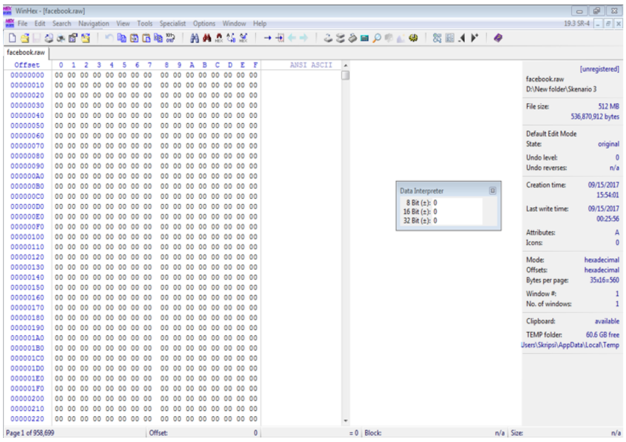
The test indicated that both malware was successfully identified in the volatile memory using DumpIt and BelkaSoft.

C. Showing Password Login Social Media

Table II shows the results of the last scenario. DumpIt and BelkaSoft acquired raw digital evidence related to social media platforms on both PCs. Later, WinHex is applied to search for passwords within the raw digital evidence and successfully obtain the passwords properly (see Fig. 15).

TABLE II. THE RESULTS OF THE THIRD SCENARIO

	Memory	Facebook Account	Twitter Account
DumpIt	DDR2 RAM	√	√
	DDR3 RAM	√	√
BelkaSoft	DDR2 RAM	√	√
	DDR3 RAM	√	√



```
Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F     ANSI ASCII
00000000  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000010  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
000000A0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
000000B0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
000000C0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
000000D0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
000000E0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
000000F0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000100  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000110  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000120  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000130  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000140  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000150  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000160  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000170  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000180  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000190  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
000001A0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
000001B0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
000001C0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
000001D0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
000001E0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
000001F0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000200  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000210  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000220  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
```

```
8 Bit (±): 0
16 Bit (±): 0
32 Bit (±): 0
```

```
[unregistered]
facebook.raw
D:\New folder\Skenario 3
File size:              512 MB
                536,870,912 bytes
Default Edit Mode
State:                original
Undo level:                  0
Undo reverses:             n/a
Creation time:      09/15/2017
                      15:54:01
Last write time:    09/15/2017
                      00:25:56
Attributes:                  A
Icons:                       0
Mode:              hexadecimal
Offsets:           hexadecimal
Bytes per page:      35x16=560
Window #:                    1
No. of windows:              1
Clipboard:           available
TEMP folder:       60.6 GB free
Users\Skripsi\AppData\Local\Temp
```

```
Page 1 of 958,699 | Offset: 0 | = 0 | Block: n/a | Size: n/a
```

Figure 15. Winhex tool.

Based on the findings obtained from the three scenarios, the following key observations can be made:

- DDR2 captures a smaller amount of digital evidence compared to DDR3 in terms of running applications.
- DDR2 and DDR3 exhibit similar patterns of evidence capture when dealing with existing malware.
- DDR2 and DDR3 demonstrate comparable evidence capture regarding password login on social media websites.

However, it is essential to acknowledge the limitations of this research. Firstly, the comparison is focused solely on volatile digital evidence between DDR2 and DDR3. Secondly, the analysis relies on DumpIt and BelkaSoft, open-source software tools, for conducting the three scenarios.

In future studies, it would be beneficial to explore the following avenues of research. Firstly, investigating other types of RAM to assess potential differences in the obtained digital evidence. Secondly, comparing the capabilities of open-source and proprietary software in extracting volatile digital evidence. Thirdly, exploring various attack scenarios such as email threats, ransomware, and other applications. Future studies can provide further insights and enhance the understanding of volatile memory forensics analysis by addressing these avenues.

IV. CONCLUSION

This paper shows live forensics investigation to examine whether DDR2 RAM and DDR3 RAM identify volatile digital evidence differently through three life forensics scenarios. The first scenario is RAM examination while the application is running, the second is malicious codes, and the third is password login social media. All scenarios are forensically examined using two open-source memory forensics, DumpIt, and BelkaSoft RamCapture tools.

While the results of scenarios two and three show that both DDR2 RAM and DDR3 RAM identify similar findings, both types of memories reveal slightly different results in the first scenario. In the first scenario, DDR2 RAM conceals lesser digital evidence related to registry handles and network activities during live forensics investigation than DDR3 RAM.

The findings in the first scenario indicate that using old version memory such as DDR2 may prevent computer forensics tools from revealing complete digital evidence, which is considered a severe issue by computer forensics examiners. Further investigations by incorporating different approaches and tools are required to assess the findings of this study.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

IS: conceptualization, methodology, validation, formal analysis, writing—original draft preparation. IS and DS: investigation, resources, writing—review and editing. DS: funding acquisition. Both authors had approved the final version.

FUNDING

This research is part of the grant PIB of the Year 2023, supported by Bina Nusantara University. This research grant contract number was 029/VRRTT/III/2023 on March 01, 2023.

ACKNOWLEDGMENT

The authors also thank the generous support from the CAIR Center for Applied ICT Research, Department of Computer and Networking Engineering, Politeknik Negeri Ujung Pandang.

REFERENCES

[1] A. Case and G. G. Richard III, "Memory forensics: The path forward," *Digit. Investig.*, vol. 20, pp. 23–33, 2017.

[2] I. Hamid, A. Alabdulhay, and M. M. H. Rahman, "A systematic literature review on volatility memory forensics. Computational vision and bio-inspired computing," in *Proc. ICCVBIC 2022*, 2022, pp. 589–600.

[3] F. Pagani, O. Fedorov, and D. Balzarotti, "Introducing the temporal dimension to memory forensics," *ACM Trans. Priv. Secur.*, vol. 22, no. 2, pp. 1–21, 2019.

[4] I. Syamsuddin and M. Musaruddin, "Daeng AMANG: A novel AIML based chatbot for information security training," in *Proc. the 9th Computer Science On-line Conference, CSOC 2023*, Springer International Publishing, 2023.

[5] A. S. Bozkir, E. Tahillioglu, M. Aydos, and I. Kara, "Catch them alive: A malware detection approach through memory forensics, manifold learning and computer vision," *Comput. Secur.*, vol. 103, 102166, 2021.

[6] N. Nasrullayev, T. O. Valijonovich, and D. M. Avlakulovich, "Static and live digital forensics, along with practical examples of tools used for each approach," *Texas Journal of Engineering and Technology*, vol. 19, pp. 21–27, 2023.

[7] S. Rahman and M. N. A. Khan, "Review of live forensic analysis techniques," *Int. J. Hybrid Inf. Technol.*, vol. 8, no. 2, pp. 379–388, 2015.

[8] D. Firoozjaei, A. Mahdi, and A. A. H. Lashkari, "Memory forensics tools: A comparative analysis," *Journal of Cyber Security Technology*, vol. 6, pp. 149–173, 2022.

[9] M. Hirano, T. Tsuzuki, S. Ikeda, N. Taka, K. Fujiwara, and R. Kobayashi, "Waybackvisor: Hypervisor-based scalable live forensic architecture for timeline analysis," in *Security, Privacy, and Anonymity in Computation, Communication, and Storage*, Cham: Springer International Publishing, 2017, pp. 219–230.

[10] M. I. Al-Saleh, Z. A. Al-Sharif, and L. Alawneh, "Network reconnaissance investigation: A memory forensics approach," in *Proc. 2019 10th International Conference on Information and Communication Systems (ICICS)*, 2019.

[11] K. K. Sunu and S. Sherly, "Extraction of memory forensic artifacts from windows 7 RAM image," in *Proc. 2013 IEEE Conference on Information & Communication Technologies*, IEEE, 2013.

[12] S. Lindenlauf, H. Hofken, and M. Schuba, "Cold boot attacks on DDR2 and DDR3 SDRAM," in *Proc. 2015 10th International Conference on Availability, Reliability and Security*, 2015.

[13] T. Thomas, M. Piscitelli, B. A. Nahar, and I. Baggili, "Duck hunt: Memory forensics of USB attack platforms," *Forensic Science International: Digital Investigation*, vol. 37, 301190, 2021.

[14] A. R. Javed, W. Ahmed, M. Alazab, Z. Jalil, K. Kifayat, and T. R. Gadekallu, "A comprehensive survey on computer forensics: State-of-the-art, tools, techniques, challenges, and future directions," *IEEE Access*, vol. 10, pp. 11065–11089, 2022.

[15] C.-W. Tien, J.-W. Liao, S.-C. Chang, and S.-Y. Kuo, "Memory forensics using virtual machine introspection for Malware analysis," in *Proc. 2017 IEEE Conference on Dependable and Secure Computing*, 2017.

[16] F. E. Salamh, U. Karabiyik, and M. K. Rogers, "Asynchronous forensic investigative approach to recover deleted data from instant messaging applications," in *Proc. 2020 International Symposium on Networks, Computers and Communications (ISNCC)*, 2020.

[17] A. Kazim, F. Almaeeni, S. A. Ali, F. Iqbal, and K. Al-Hussaeni, "Memory forensics: Recovering chat messages and encryption master key," in *Proc. 2019 10th International Conference on Information and Communication Systems (ICICS)*, 2019.

[18] J. Choi, J. Yu, S. Hyun, and H. Kim, "Digital forensic analysis of encrypted database files in instant messaging applications on Windows operating systems: Case study with KakaoTalk, NateOn and QQ messenger," *Digit. Investig.*, vol. 28, pp. S50–S59, 2019.

[19] R. D. Thantilage and N. A. Le Khac, "Framework for the retrieval of social media and instant messaging evidence from volatile memory," in *Proc. 2019 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/13th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE)*, 2019.

[20] M. Davis, B. McInnes, and I. Ahmed, "Forensic investigation of instant messaging services on linux OS: Discord and slack as case studies," *Forensic Science International: Digital Investigation*, vol. 42, 301401, 2022.

[21] A. Holmes and W. J. Buchanan, "A framework for live host-based Bitcoin wallet forensics and triage," *Forensic Science International: Digital Investigation*, vol. 44, 301486, 2023.

[22] J. G. Murias, D. Levick, and S. McKeown, "A forensic analysis of streaming platforms on Android OS," *Forensic Science International: Digital Investigation*, vol. 44, 301485, 2023.

[23] A. A. Khan, A. A. Shaikh, A. A. Laghari, and M. M. Rind, "Cloud forensics and digital ledger investigation: a new era of forensics investigation," *International Journal of Electronic Security and Digital Forensics*, vol. 15, pp. 1–23, 2023.

[24] B. Gervasi. *DRAM Module Market Overview*, SimpleTech, JEDEX Shanghai, 2005.

[25] R. Mcree, "Memory analysis with DumpIt and volatility," *ISSA Journal*, 2011.

[26] B. Popović, K. Kuk, and A. Kovačević, "Comprehensive forensic examination with Belkasoft evidence center," *Belgrade: Academy of Criminalistic and Police Studies*, vol. 2, pp. 419–433, 2018.

[27] H. Pomeranz, "Detecting malware with memory forensics," SANS Institute, 2015.

[28] A. Mohaisen and O. Alrawi, "Unveiling zeus: Automated classification of malware samples," in *Proc. the 22nd International Conference on World Wide Web*, 2013, pp. 829–832.

[29] A. Huseinovic and S. Ribic, "Virtual machine memory forensics," in *Proc. 21st Telecommunications Forum Telfor*, 2013.

[30] Volatility Workbench—A GUI for Volatility memory forensics. [Online]. Available: https://www.osforensics.com/tools/volatility-workbench.html