Identifying and Finding Forensic Evidence from Windows Application

Funminiyi Olajide, *School of Engineering, University of Portsmouth, United Kingdom*

Nick Savage, *School of Engineering, University of Portsmouth, United Kingdom*

Galyna Akmayeva, *Infonomic Society, Republic of Ireland*

Charles Shoniregun, *Infonomic Society, United Kingdom*

Abstract

This paper presents the method of identifying and finding forensic evidence from the volatile memory of Windows computer systems. This is a scenario–based investigation on what amount of user input can be recovered when application is opened and images are captured at set interval while Windows system is still actively running. This approach of digital investigation revealed the extracted evidence of user input stored and as dispersed on the application memory of Windows system. In this experiment, the result shows a coherent view of user input on some commonly used applications with over 39% of user input stored on MS Access and 44% was stored on Excel. The quantitative assessment of user input will be presented on the basis of the repeated number of user input recovered, the percentage of user input found and the length of evidence found in a continuous block of the application memory.

1. Introduction

This paper details an approach for extracting user input information from the memory dump of Windows applications. Extracting user input requires the capturing of the memory allocated to an application, such that user input found is forensically sound for evidence presentation at the court of law. Application of digital forensic on physical image acquisition and memory analysis is gaining good attention from the experts in the community. The extracted user input information can reveal user actions on the applications which may augment a digital forensic investigation of crime and computer related fraud.

The research study focused on how much relevant user input can be recovered from the volatile memory of Windows application. This information can be used to infer what the user was doing, what the user have been doing and what they have been using the application for. It is generally accepted that the acquisition of volatile memory can reveal facts about the current and past usage of the computer system [l]. However, the digital forensics community feels the need for the development of tools and techniques for capturing and analysing the volatile memory (RAM) content [2]. The need for digital investigation into RAM is motivated by the fact that the volatile memory contains information that cannot be found using the traditional hard disk forensic investigation. Although, in recent times, this need has been identified, but little has been done on the finding of forensically relevant evidence stored on Windows applications memory.

2. Related work

In today digital investigation, data acquisition and physical memory analysis of Windows computer systems plays an important role in digital forensics [3]. Forensic investigators find it helpful to seize and capture data from the physical memory content and perform post-incident analysis for an essential and potential evidence presentation. However, there is little research into the amount of user input information that can be recovered from only the computer system memory (RAM) while the application is still running. The research of Burdach found that memory acquisition tools are still not fully developed and emphasised that these tools will receive much attention in the future [4]. An example of a technique that has been developed is a runtime monitor program in Windows systems and this provided a physical acquisition to log read and write operations in memory-mapped files [5]. This approach provides prospective evidence regarding the lifetime of process information in the physical memory. This process helps to prepare systems before any occurrence of incidents. Another technique is among the few hardware-based memory acquisition methods that change memory contents as little as possible by using a PCI extension card to dump the memory content to an external device [6].

A range of software-based tools have been recently developed for memory acquisition and memory analysis. With regards to memory acquisition, the paper of [7] is a command line tool that reconstructs the virtual address space of the system process and other processes. A method of [8] is a tool that is capable of revealing hidden and terminated processes and threads. Also, Win32dd [9] and Nigilant32 [10] are tools that can capture and or, image the volatile memory of computer systems. In addition, MemParser [11] and the Volatility Framework [12] are examples of tools that can perform memory analysis. Of these two, the Volatility Framework is more extensive in memory dumping and memory analysis. This tool is capable of performing the analysis on a variety of memory image formats such as DD format, crash dump and Hibernate Dumps. Volatility is able to list OS kernel modules, drivers, open network socket, loaded DLL modules, heaps stacks and open files.

In recent times the seminar on [13] addresses the need for more sophisticated tools on physical memory acquisition and analysis. The workshop of [2] and [14], recently issued a memory analysis challenge to encourage research and tool development in this direction. This process of application memory analysis is based on data carving method which is a recovery approach that is frequently used during digital investigations. Moreover, it is essential that a new development tools should integrate different approaches.

A new model of [15], point towards the graphics extraction that is contained in a memory dump. This is a phase approach to forensic processes, classification of relevant data types and classes of forensic methods. A paper of [16] presented on the identification of user input stored on application memory of Windows systems. It describes the pathway of information dispersed in the memory. However, this paper focused on the finding and extraction of the user input evidence based on Scenario 1 of the research of a study of application level information and volatile memory analysis of Windows computer systems [17]. The Scenario 1 of the research indicated that user opened applications on Windows system and interacted with the applications and while the application is still opened, and the Windows system is actively running, volatile imaging was carried out.

3. Methodology

The aim of the research is to identify and to find out what amount of user input recovered from only the RAM of application when the application is still opened and Windows system is actively running while the user is still interacting with the systems. For this scenario, the applications are opened at the beginning of the day, the user uses the applications as if they were working on a normal day and the volatile memory images are captured at set interval of 30 minutes. An investigator finds that the computer is still turned on, and the applications that the user was using are still opened. The user has recently interacted with the applications and then, investigators capture the image of Windows.

To ensure that our research results are as applicable as possible, commonly used applications in business organisations were first identified. This was achieved by making contacts to different institutions like banks, commercial retailer, telecommunication and public services to ask their technical support team, which application were the most commonly used on Windows system. The research paper of [18], [19] and [20] discussed and presented on some commonly used applications.

In this paper, digital investigation into the amount of user input recovered from MS Access 2007 and Excel 2007 will be presented based on Scenario 1. Before we commenced the experiments using Nigilant32, Windows machine was shut down and rebooted to ensure that the system was as clean as possible. This is important to ensure that the memory allocated to each application had not previously been used to store unrelated data.

In order to make our results as applicable as possible we tried to replicate a normal working environment while capturing memory images. Moreover, user inputs were made in each of the application at set interval. During the day, a user interacted with the applications and volatile memory was captured at interval of 30 minutes. Example of the user input is shown in Table 1.

As shown in Table 1, user input on each application varies from one period to the other. In some cases, no inputs were made on the applications but, images were captured at set interval. Series of tests was achieved for days until 100 images were captured on each application.

As the physical memory in the computer was 2 Gigabytes (GB), this resulted in 200 GB of images being captured. After the collection of data using Nigilant32, we make copies of images captured on each application for data preservation purposes.

Table 1. Methodology approach

Application	User Input
MS Access 2007	Write text and numbers into a database or do nothing. Save the database or do not save
Excel 2007	List a set of numbers, draw a graph of the numbers or do nothing. Save the document or do not save the document. Input may contain alphanumeric characters.

The memory allocated to the application under investigation is extracted for every image captured. Text information is extracted from this memory dump using “strings”. After this, a pattern matching process is run to identify instances of user activity through the memory allocated to the application. This pattern matching techniques takes the original user input information and matches it with the extracted memory dump of the application.

The fragments of the user input being the text of information was extracted from the memory dump. The aim is to uncover the user input stored on the applications and what user input can be recovered from the application memory based on the Scenario 1 of the research of a study of application level information and volatile memory analysis of Windows computer systems. This investigation focused on answering the question “can all information related to how a user is using that application be recovered if the memory is captured while that application is still running?”.

The investigation was focused on user input to provide further pieces of information that may be useful for forensic investigation. This investigation includes how much user input can be recovered from the memory and how the evidence was dispersed over time in the volatile memory of Windows application. This approach will aid a forensic investigator when they are answering the questions shown in Table 2.

The approach of matching the text information that was extracted from the memory dump results in a file that contains fragments of data that may be considered as evidence. This method of extracting relevant information from the physical memory of Windows applications might be applicable to other Windows computer systems such as Windows 7.

As investigated, using the pattern matching techniques, the extracted user input contains both partial and whole fragments of evidence. The next stage of the investigation was to process the data and to identify those fragments of user input that was extracted from the application memory. The user input is contained in the memory and can be used as evidence. In this paper, the information extracted was analyzed to give answers to forensic question of Scenario 1 based on what the user was doing, what the user has been doing and what the user was using the application for.

Table 2. Specific forensic questions

S/n	Specific Forensic Questions
1.	What user was doing on the application, what they have been doing on the application and on what application was the user evidence was found or dispersed in the physical memory.
2.	How did the user input found was stored over time in the physical memory. This includes how user input on the application was found and stored over time e.g. line number allocated
3.	Where user input was stored and found in the memory. This includes how the evidence recovered was dispersed and as allocated in the physical memory.
4.	Is the evidence dispersed or in whole/partial fragment of evidence?
5.	Has the user input found been validated? Is it forensically sound and can it be presented as evidence in the court of law.

4. Result: Quantitative assessment

The three metrics calculated was to assess the quantity of information identified in the memory dump that was extracted; the mean number of times that evidence has been found, the mean percentage of evidence found and the mean length of evidence found in a continuous block. The mean number of times the evidence has been found is the average number of repeated fragments of evidence identified in the volatile memory of the application. The mean percentage of evidence found shows how much of the original user input has been identified in the memory dump that was extracted. The mean length of evidence found in a continuous block is the mean length of evidence identified by the pattern matching algorithm. This metric was deemed to be important as it could be used to make the pattern matching algorithm more efficient. During this investigation the pattern matching was performed on blocks of 10 characters. Table 3 show the results obtained by this investigation.

Table 3. Quantitative assessment

Sample Application	Mean Evidence Repetition	Mean % of Evidence found	Average Lengths of Evidence found in Continuous block
MS Access	453.00	39	17.22
Excel	51.65	44	21.33

In this investigation, it was discovered that not all the user input stored on the application was recoverable. The result on Table 3 described the quantity amount of information recovered on the applications. This relevant user input can be used as evidence. As investigated, there are little amount of information found in both MS Access and Excel applications. The percentage of evidence found on MS Access application was 39% while 44% of relevant user input was recovered on Excel application. Three different graphs were plotted to illustrate the quantitative assessment of user input stored and as recovered from volatile memory of Excel and MS Access applications. Based on this Scenario 1, user input contains a greater mixture of Latin characters and numerical data when compared to using other applications. There is large amount of in-built systems defined data in the memory of this application which makes it difficult to find data when using Excel. This is because of the existence of other numerical data in the memory image that was captured on this application. The Figures below describes three metrics of the quantitative assessment of user input on Windows memory.

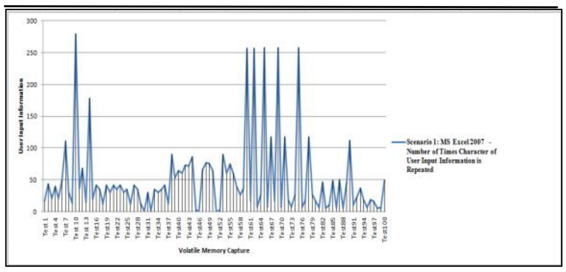

Figure 1 Excel: Number of times a character of user input is repeated

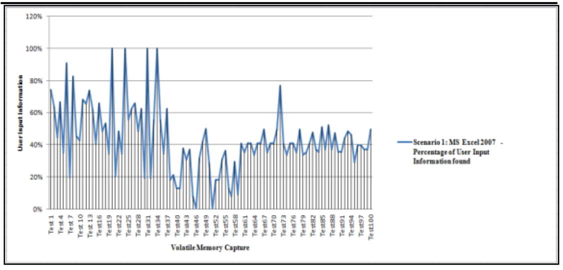

Figure 2 Excel: Percentage of user input found

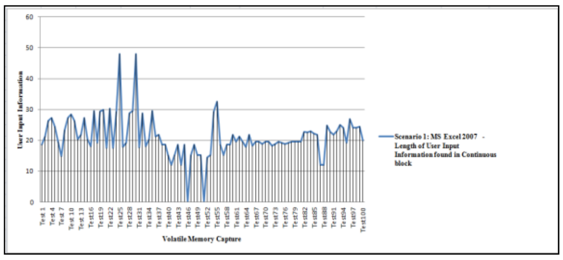

Figure 3 Excel: Length of user input found in continuous block of evidence

In Figure 1, test10, test59, test62, test65 and test74 reported the highest peak of the number of times a character of user input is repeated. Figure 2, reported the highest peak of the percentage of user input found in test19, test24, test31 and test34. It can be said that the user input contain more of textual characters than the numerical characters.

In Figure 3, similar result was reported in test24 and test31. This shows that the length of user input found in continuous block is at highest peak. This means more of textual data was initially entered on the application. The variation exhibited in the graphs is due to the relative amount of user input recovered from the memory which was Latin characters (easier to find) to numerical characters (difficult to find). However, 44% percentage of evidence was found on Excel application.

As required in Scenario 1 of the research of application level information and volatile memory analysis of Windows computer systems, the quantitative assessment of user input on MS Access are presented with three metrics as shown in figures below. The recovering of data is very difficult on this application. It is difficult to identify the user input against the in-built systems defined data of the application. This is because there are large amount of textual in-built system-defined data that resides in the application memory.

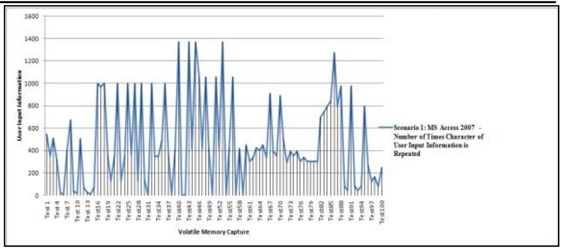

Figure 4 MS Access: Number of times a character of user input is repeated

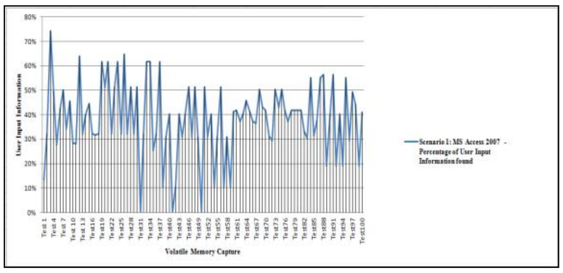

Figure 5 MS Access: Percentage of user input found

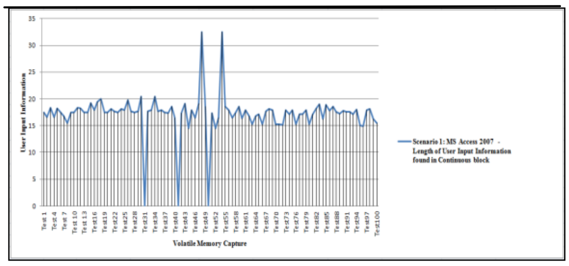

Figure 6 MS Access: Length of user input found in continuous block of evidence

As shown in Figure 4, MS Access recorded the number of repeated character of user input however; test40, test43, test45 and test53 reported the highest number of repeated characters of the user input found on this application. The highest percentage of user input found in Figure 5 was stored in test3, however, the mean percentage was at the lowest when compared with other application In Figure 6; the highest information found in continuous block was in test48 and test53. When search pattern was used, 39% of related user input was found in the memory allocated to MS Access application.

5. Conclusion

This model of Scenario 1 has been used to describe the process of securing what may be termed as forensically relevant data from the memory content of Windows applications. The approach taken has become part of forensic memory analysis in digital investigation. Extracted memory evidence of MS Access 2007 and Excel 2007 respond to the key questions of forensic investigators being the quantity amount of user input recovered. This investigation was based on what information can be recovered from the application memory.

6. Future work

In the future, we will investigate on the qualitative assessment of user input recovered while the application is opened and user is still interacting with the systems.

7. References

[1] W. Aaron, F. Timothy, A. William, and P. Nick, (2006) "FATKit: A Framwork for the extraction and analysis of digital forenisc from volatile sysytem memory," *Journal Digital Investigation*, vol. III, no. 4, pp. 197-210.

[2] DFRWS. (2007) ‘Digital Forensic Research. “http://www.dfrws.org/2007/challenge/results.sh” (26 March 2010).

[3] F. Olajide. N. Savage., (2009) "Forensic Live Response And Events Reconstruction Methods In Linux Systems," in *The 10th Annual Conference on the Convergence of Telecommunications, Networking & Broadcasting*, Liverpool, pp. 141-147.

[4] M. Burdach. (2005.) Windows Memory Forensic. "http://forensic.seccure.net/tools/wmft.tar.gz" (12 May 2010).

[5] C. Brian and G. Joe, (2004) "A hardware-based memory acquisition procedure for digital investigations," *Digital Investigation*, vol. I, no. 1, pp. 50-60.

[6] Garcia G.L., (2007) "Forensic Physical Memory Analysis: An Overview of Tools and Techniques," in *TKK T-110.5290 Seminar on Network Security*, Helsinki, Finland, pp. 305-320.

[7] Msuiche. (2008) ‘Msuiche.net Capture memory under win2k3/ vista/Windows7 with win32dd/wind64dd’. (9 March 2008).

[8] ManTech. (2008) 'ManTech Memory DD', http://www.mantech.com/msma/MDD.asp (8 March 2010)

[9] Solomon DA. Russinovich ME, (2009) '*Microsoft Windows internal Covering Windows Server 2008 and Windows Vista*', 5th ed. Washington, USA: Microsoft Press.

[10] Nigilant32. (2006) 'Agile Risk Management, Nigilant32 - Windows Incident Response Tool'. http://www.agilerm.net/publications_4_.html (16

April 2010).

[11] Betz C. (2005). "http://www.dfrws.org/2005/challenge/memparser.shtml"(26 August 2009).

[12] Volatile Systems. (2006) 'The Volatility framework: Volatlile Memory Artifact Extraction Utility Framework', https://www.volatilesystems.com/ (12 April 2009).

[13] Carvey H. Kleiman D. (2007), "Windows Forensic Analysis Incident Response and Cybercrime Investigation Secrets," *International Journal of Digital Investigation*, vol. II, no. 2, pp. 23-78.

[14] DFRWS. (2010) Digital Forensic Research, "http://www.dfrws.org/2010/challenge/results.sh" (26 March 2011).

[15] T. Hoppe, J. Dittmann, and S. Kiltz, (2009) "A New Forensic Model and ITS Application To The Collection, Extraction And Long Term of Screen Content OFF A Memory Dump," *Proceedings of the 16th international conference on Digital Signal Processing*, vol. 1, no. 1, pp. 1- 6.

[16] F.Olajide N.Savage., (2011) "On the Identification Of Information Extracted From Windows Physical Memory,*" International Journal of Information Security Research*, vol.1, Issue. 3, pp. 164-168.

[17] F.Olajide, (2011) "A Study of Application Level Information From The Volatile Memory of Windows Computer Systemns. PhD Thesis, University of Portsmouth.

[18] F. Olajide and N. Savage, (2011) "Forensic extraction of user information in continuous block of evidence," *Information Society (i-Society), 2011 International Conference*, London, pp. 476-481.

[19] F.Olajide N.Savage., (2009) "Application Level Evidence From Volatile Memory," *Journal of Computing in Systems and Engineering*, vol. II, no. 2, pp. 70-78.

[20] F.Olajide N.Savage., (2011) "Dispersal Of Time Aspect Of Information Stored On Physical Memory," in *Cyberforensic - International Conference on Cybercrime Security and Digital Forensics*, Glassgow.