Forensic Extraction of User Information in Continuous Block of Evidence

Funminiyi Olajide
Department of Electronic and Computer Engineering,
University of Portsmouth, UK
funminiyi.olajide@port.ac.uk

Nick Savage
Department of Electronic and Computer Engineering,
University of Portsmouth, UK
nick.savage@port.ac.uk

Abstract-**Extraction of user information in the physical memory of Windows application is vital in today's digital investigation. Digital forensic community feels the urge for the development of tools and techniques in volatile memory analysis. However, there have been few investigations into the amount of relevant information that can be recovered from the application memory. In this research, we present the quantitative and qualitative results of experiments carried out on Windows applications. In conducting this research; we have identified the most commonly used applications on Windows systems, designed a methodology to capture data and processed that data. This research report the amount of evidence that was stored over time and recovered in continuous block of evidence in the physical memory.**

Keyword - User, application, physical, Windows

I. INTRODUCTION

The recovery of user information from the physical memory of applications can be used to determine the amount of relevant data found in continuous block. This information was stored over time as dispersal evidence. In recent times, there have been outstanding results into the research of application level evidence from the extracted memory dump processes of Windows applications based on user activities [1]. But, little has been done on the analysis of the acquired evidence extracted from these applications. In this paper, the approach was focused on how much information can be recovered when the system is still running and images were captured but, user is not interacting with the systems. In this approach, evidence in continuous block will be fully explored. In addition, memory that has been allocated to applications was extracted to infer what the user has been using the applications for; the information which may not be visible when using traditional hard disk forensic investigation tools and techniques. By reconstructing the original user input information and pattern matching of the information with the extracted memory dump string processes, we can found the average length of character strings resided in the memory of the applications. This approach can visibly locate how the evidence is stored in the memory. The investigation of the information contained within the memory of the application may become an essential tool for information assurance as well as solving crime and tracing fraud against people, where evidence may reside on a computer. For example, bank fraud, identity fraud, and money laundering, child exploitation by paedophile using Facebook and all other twitted applications as a means of communication to commit various crimes. In progress to this research, forensic investigation techniques experiments have been designed to investigate the quantity and quality of information that can be recovered from the physical memory of commonly used applications. This approach uncovers what user has been doing on the application and provides additional relevant evidence when forensically investigating the volatile memory of Windows computer systems. In this research, the most commonly used applications have been identified and 100 sample data of the information retrievable from those applications had been conducted. The result of this investigation will facilitate the pattern matching techniques on physical memory of an application based on various user input. This will help in further analysis of the data found in the physical memory allocated to the applications.

II. RELATED WORK

The current tools and techniques for analyzing forensic data in the physical memory are limited. A recent workshop on digital forensic highlighted the need of the forensic community for the development of tools and techniques for capturing data and analyzing the physical memory of computer systems [2]. The research idea is motivated on the facts that physical memory content of applications contains information that cannot be found on traditional hard disk forensic investigations. Olajide and Savage, identified seven most commonly used applications in windows system and, design a methodology approach to the extraction processes of forensically relevant evidence from physical memory [3]. This research facilitated the investigation and the analysis of data found in the physical memory content of the applications. The extended version of this research paper was presented on [4], which discusses on the identification of information extracted from the Windows physical memory. Although, Burdach describes that the memory extraction tools are not fully developed, [5] some tools do currently exist. Some of these tools required specialist hardware to be added to the computer system in order to extract the memory image. An example of this type of tool has been

developed by Garcia, [6]. Garcia's tool is among the few hardware-based memory acquisition methods that use a PCI extension card to dump the memory content to an external device. Hardware based acquisition tools have advantages and is not requiring any additional software to be executed on the computer system when acquiring evidence. However, they do need to have the tool installed on every machine which may require investigation prior to that investigation. For this reason there has been much focus on software-based tools. Msuiche has developed a command line tool that can be used on Windows to extract the physical memory of the computer system, [7]. It also reconstructs the virtual address space of the system process and other processes.

In an attempts to find the most applicable tool to use, Memory dd developed by ManTech International Corporation was tested [8]. This tool is capable of revealing hidden and terminated processes and threads. Also, Windows Internal 5, covering Window server by using Win32dd [9], developed by Solomon and Russinovich was also tested during our investigation. However, Nigilant32 was very useful when tested. This tool allows an investigator to preview a memory image and take a snapshot of it. It has a small footprint, using less than 1 MB in memory when loaded and with a minimal impact during acquisition [10]. This tool was used to capture and or image the physical memory of computer systems.

There are few tools that can perform memory analysis. Some example are MemParser [11] and the Volatility Framework [12]. Of these two, the Volatility Framework is more extensive. 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. Recently, a seminar addressed the need for more sophisticated tools on physical memory acquisition and analysis [2]. This is because external and internal intrusions will continue even in the robust security infrastructures of the best government and industry systems. But the key to successfully preventing and responding to any digital fraud investigations is the sound identification, collection, preservation and analysis of computer evidence. The paper of [13], issued a memory analysis challenge to encourage research and tool development in this direction. Therefore, a method of [14] laid emphasis on the importance of forensic live response and event reconstruction methods. The extension of this work relies on the research of application level evidence from physical memory [1]. This research identified the important aspects of memory analysis and proposed an approach for application level evidence. A paper [15], identified the dispersal of time aspect of information stored on physical memory. This approach provides prospective information on how evidence is dispersed in the memory of some commonly used application during memory analysis. Another paper [16], discuses the methodology approach of extracting relevant information from the physical memory of Windows system. This approach demonstrates the application process of user information on Windows application. As a result of various research investigations, it has become imperative to investigate into the research work as focussed into the extraction of user information by pattern matching techniques. This research uncovers the extraction of user information in continuous block of evidence.

III. METHODOLOGY

The important aspect of this research is the identification of most commonly used applications on Window Systems of which to perform the experiments on. To ensure that our results are as applicable as possible, we attempted to identify which applications are commonly used by businesses. This is by contacting different institutions like banks, commercial retailer, telecommunication and public services to ask their technical support team which application were most commonly used on Windows systems. The most commonly used applications identified by this enquiry were Word 2007, Excel 2007, PowerPoint 2007, Outlook 2007, MS Access 2007, Internet Explorer 7.0 and Adobe Reader 8.0. In order to make our results as applicable as possible we tried to replicate a normal working environment while capturing memory images. In this, the computer would be turned on at the start of the day and then turned off at the end of the business day. When the computer is first turned on the application will be opened. During the day user will interact with the applications recording exactly what was done and not using the computer for any other purpose. Example of 30 minutes blocks of user input information is shown in Table I.

The physical memory of the computer is captured every 30 minutes. As earlier mentioned, there are seven applications studied in the overall research but, in this paper, we are presenting only the results of three of these; Word 2007, Excel 2007 and PowerPoint. As describe in Table I, is a typical flow of the evidence-based procedures that was formulised for the purpose of this research. 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 to ensure that the memory allocated to each application had not previously been used to store unrelated data. Moreover, user inputs were made in each of the application at set interval of 30 minutes. As we can see from the Table I, user actions on the application vary at interval. In some cases, no input users were made on applications but, images were captured at every 30 minutes. We ran series of test for days and this was continued 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, we make copies of images captured on each application for data storage and preservation purposes. In this experiment, our opinion was to identify what information can be recovered from only the RAM when the application was still running on Windows system and user was not interacting with the systems.

TABLE I. METHODOLOGY APPROACH

Sample Application	User Action at every 30 minutes
Word 2007	Write paragraph of text, with commas, semi-colon, full stop, or do nothing on the document, save document or do not save document. Input alphanumeric, character 0-9, brackets closed or open and or, not. Long sentence or short.
Excel 2007	Lists set of numbers, draw a graph of the numbers or do nothing, save document or do not save. Input may contain alphanumeric, character 0-9, brackets close or open and or, not. Long sentence or short.
PowerPoint 2007	Write a slide, slides of texts or do nothing, save document or do not save. User input may contain or type alphanumeric, character 0-9, brackets close or open and or, not. Long sentence or short.

VI. RESULT: SAMPLE QUANTITATIVE ASSESSMENT

In this section, we present the quantitative assessment of forensically relevant data that was extracted from the physical memory content of Word 2007, Excel 2007 and PowerPoint. As shown in Table 2, this investigation focused on the identification of user information and the amount of data recovered and as dispersed in the physical memory. In Excel, the application level evidence was found dispersed and stored in continuous block of evidence. Excel application recorded the least amount of evidence information that can be recovered from the physical memory. We discovered that the percentage amount of evidence found on Excel was as a result of what user input on the application. Thus, user input may contain mixture of data texts, numeric data, graphs and tables.

TABLE II. QUANTITATIVE RESULT

	Word 2007	Excel 2007	PowerPoint 2007
Mean Evidence Repetition	194.70	62.30	110.00
Mean % of Evidence found	96	44	94
Average Length of Evidence found in Continuous block	48.65	21.33	51.89

Moreover, we have large amount of evidence stored over time in Word and PowerPoint applications. Word recorded a remarkable amount of 96% of evidence stored in continuous block in the memory. The original user input information on Word application contained only the data text of paragraphs, alphanumeric characters with bracket, semi-colon, full stop, question mark and currency. In PowerPoint application, user input also includes a paragraph of texts with commas, full stop, semi-colon, with subject headings. In PowerPoint application the percentage amount of evidence found was 94%. In Word and PowerPoint application, it was recorded that large percentage amount of evidence was found as stored over time in the memory while the applications were running and images were captured. This is because these applications can retain information in the memory for a longer period.

V. RESULT: SAMPLE QUALITATIVE ASSESSMENT

In this section, we describe the quality of information that can be recovered from only the computer system memory of Word, Excel and PowerPoint applications. This qualitative assessment reveals relevant user information found in continuous blocks of evidence.

Fig. 1, illustrated how evidence was dispersed and stored over time in continuous block in the memory of PowerPoint application. For example, the slide contains, user input information like: *"Hostile Voters ambush Cameron"*. This information was recovered in continuous block of the physical memory. The information was found repeated and appears consistently in three different locations. The whole body texts of user input were retrieved and contained both partial and whole fragment of evidence in continuous block.

For example, line numbers *1851, 14072, 14085 and 1531899 contained whole fragment of relevant user information.* This was visibly found in continuous block. Partial fragment of evidence that was recovered in the memory can be traced to line numbers *1852, 1853, 1854, 13264, 14082, 14083, 14087, 1531900 and 1531901.* Process of extraction of user information was achieved by pattern matching techniques and reveals user action in the allocated memory in continuous block.

In another example, we discovered that partial fragment of evidence was contained in line numbers *14086 and 14087.* This evidence found was in continuous block. The allocated evidence was matched with *line number 14085* to form a whole fragment of evidence information. Another example of this experiment was in *line number 1537900 and 1531901.* This evidence is partially allocated. This can be reconstructed and matched with *line number 1531899* to form a whole fragment of user information extracted from the memdump strings processes of the application. *To clarify more information on this experiment, line number 1852 and 1853* contained the partial fragment of evidence in continuous block. This evidence was allocated as stored in the physical memory of the application. This also was reconstructed and matched with the *line number 1851* to form a whole fragment of evidence of user information.

Extracting Evidence from "PowerPoint Application"
Date of Data Captured: 27/04/2010 Memory Dump String: Yes

Original Text	Evidence Extracted From MEMDUMP STRINGS Partial /Whole Fragment with Line Numbers
Hostile voters ambush Cameron DAVID CAMERON was confronted by hostile voters as he hit the campaign trail on the south coast. The conservative leader was confronted at Southampton University by student Chloe Green over his education policies. The 19-year-old, from Dorset, accused him of doing nothing to help the working class and getting rid of vital funding for poorer students. But Mr Cameron hit back, claiming: 'We must always help people from lower-income backgrounds to go to university. That's why we keep bursaries and expand bursaries.'	1851 ile voters ambush Cameron DAVID CAMERON was confronted by hostile voters as he hit the campaign trail on the south coast. The conservative leader was confronted at Southampton University by student Chloe Green over his education policies. The 19-year-old, from Dorset, accused him of doing nothing to help the working class and getting rid of vital funding for poorer students. But Mr Cameron hit back, claiming: 1852 We must always help people from lower-income backgrounds to go to university. That 1853 s why we keep bursaries and expand bursaries. 1854 Hostile voters ambush Cameron 13264 Southampton 13304 Hostile voters ambush Cameron 14072 DAVID CAMERON was confronted by hostile voters as he hit the campaign trail on the south coast. The conservative leader was confronted at Southampton University by student Chloe Green over his education policies. The 19-year-old, from Dorset, accused him of doing nothing to help the working class and getting rid of vital funding for poorer students. But Mr Cameron hit back, claiming: 14082 We must always help people from lower-income backgrounds to go to university. That 14083 s why we keep bursaries and expand bursaries. 14085 DAVID CAMERON was confronted by hostile voters as he hit the campaign trail on the south coast. The conservative leader was confronted at Southampton University by student Chloe Green over his education policies. The 19-year-old, from Dorset, accused him of doing nothing to help the working class and getting rid of vital funding for poorer students. But Mr Cameron hit back, claiming: 14086 We must always help people from lower-income backgrounds to go to university. That 14087 s why we keep bursaries and expand bursaries. 1531899 DAVID CAMERON was confronted by hostile voters as he hit the campaign trail on the south coast. The conservative leader was confronted at Southampton University by student Chloe Green over his education policies. The 19-year-old, from Dorset, accused him of doing nothing to help the working class and getting rid of vital funding for poorer students. But Mr Cameron hit back, claiming: 1531900 We must always help people from lower-income backgrounds to go to university. That 1531901 s why we keep bursaries and expand bursaries.

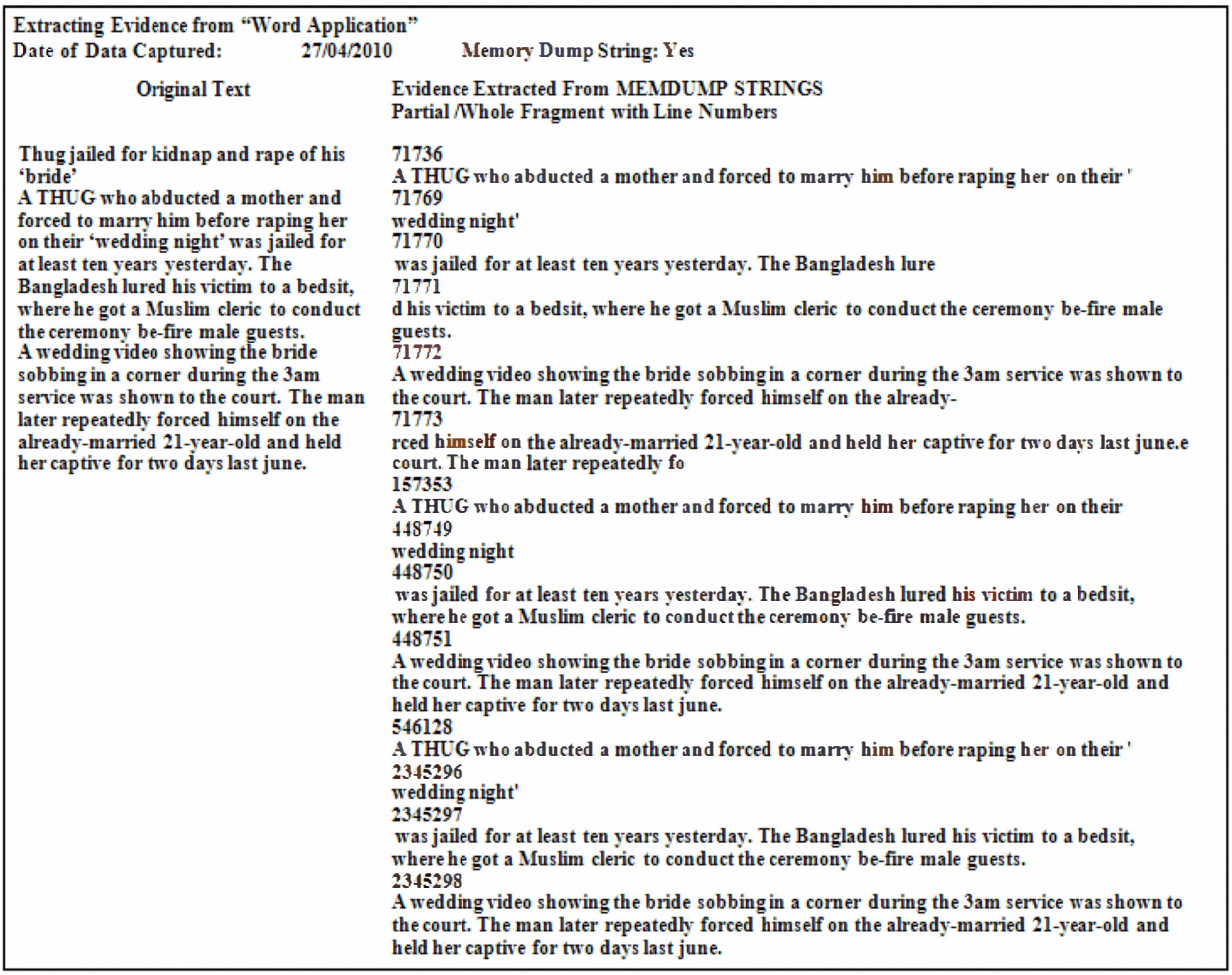
Figure 1. PowerPoint: Original User Input / Extracted memdump evidence

On Excel application, we investigated the qualitative assessment of user information. It was discovered that user input information contained both partial and whole fragment of evidence. This user information consists of alphanumeric characters. We discovered that only the data texts strings of user information can be recovered as evidence in continuous block whereas, numeric data entered by the user was irrecoverable. For example, user input information in Fig. 2, like: *Machinery Fatory in Hampshire*. In this, user input omitted *"c"* from the data texts of *"Fatory"*. We investigated to found out that this whole fragment of information was found repeated and appears consistently at four different locations in the extracted memdump strings process of the application. *In another example, line numbers 94413, 94469, 110300 and 117799* contained whole fragment of evidence found in continuous block whereas, *line number 94470* is a typical example of a partial fragment of user input information.

Extracting Evidence from "EXCEL Application"
Date of Data Captured: 27/04/2010 Memory Dump String: Yes

Original Text		Evidence Extracted From MEMDUMP STRINGS Partial /Whole Fragment with Line Numbers
Machinery Fatory in Hampshire		94413 Machinery Fcatory in Hampshire
Type	Amount Spent	94469
Electronic	£8,985,522.00	Machinery Fatory in Hampshire
		94470
Shoes	£411,225.00	y n Hampshire
Bags	£5,542,112.00	110300
Mouse	£122,220.00	Machinery Fatory in Hampshire
	£15,061,079.00	117799 Machinery Fcatory in Hampshire

Figure 2. Excel: Original text / Memdump evidence



Extracting Evidence from "Word Application"
Date of Data Captured: 27/04/2010 Memory Dump String: Yes

Original Text	Evidence Extracted From MEMDUMP STRINGS Partial /Whole Fragment with Line Numbers
Thug jailed for kidnap and rape of his 'bride'	71736 A THUG who abducted a mother and forced to marry him before raping her on their '
A THUG who abducted a mother and forced to marry him before raping her on their 'wedding night' was jailed for at least ten years yesterday. The Bangladesh lured his victim to a bedsit, where he got a Muslim cleric to conduct the ceremony be-fire male guests.	71769 wedding night' 71770 was jailed for at least ten years yesterday. The Bangladesh lure 71771 d his victim to a bedsit, where he got a Muslim cleric to conduct the ceremony be-fire male guests.
A wedding video showing the bride sobbing in a corner during the 3am service was shown to the court. The man later repeatedly forced himself on the already-married 21-year-old and held her captive for two days last june.	71772 A wedding video showing the bride sobbing in a corner during the 3am service was shown to the court. The man later repeatedly forced himself on the already- 71773 rced himself on the already-married 21-year-old and held her captive for two days last june.e court. The man later repeatedly fo 157353 A THUG who abducted a mother and forced to marry him before raping her on their 448749 wedding night 448750 was jailed for at least ten years yesterday. The Bangladesh lured his victim to a bedsit, where he got a Muslim cleric to conduct the ceremony be-fire male guests. 448751 A wedding video showing the bride sobbing in a corner during the 3am service was shown to the court. The man later repeatedly forced himself on the already-married 21-year-old and held her captive for two days last june. 546128 A THUG who abducted a mother and forced to marry him before raping her on their ' 2345296 wedding night' 2345297 was jailed for at least ten years yesterday. The Bangladesh lured his victim to a bedsit, where he got a Muslim cleric to conduct the ceremony be-fire male guests. 2345298 A wedding video showing the bride sobbing in a corner during the 3am service was shown to the court. The man later repeatedly forced himself on the already-married 21-year-old and held her captive for two days last june.

Figure 3. Word: Original User Input / Extracted memdump evidence

The third sample of our experiment was Word application. In this, we investigated the qualitative assessment of user information in continuous block of evidence. This information was stored over time in the physical memory. Fig. 3 illustrated the extraction of relevant user information in continuous block of evidence. The user input information was recovered both in partial and whole fragment of evidence. This evidence was dispersed over time in the physical memory of Word application. For example, the *line numbers 546128, 2345296, 2345297 and 2345298* can be reconstructed to form a whole fragment evidence of user information. The sample partial fragment of evidence found in continuous block of this application can be traced to the line numbers *71736, 71770, 71772 and 71773.* This allocated line numbers have answered the question of "*where"* the evidence was resided. The question of *"when"* can be answered as the evidence was dispersed over time in continuous block of physical memory. The question of *"what"* the user was doing can be answered as the original text of information entered on Word application. The question of *"how"* evidence can be reconstructed can be traced to the *line numbers 157353, 44749, 44750 and 44751.*

This information stored over time in continuous block and can be reconstructed to answer the question of "*how"*. This is relevant user information that was extracted from the memory of Windows application.

In another example, the *line numbers 546128, 2345296, 2345297 and 2345298* can be reconstructed to form a whole fragment evidence of user information. The partial fragment of evidence found in continuous block of this application can be traced to line numbers *71736, 71770, 71772 and 71773.* This allocated line numbers have answered the question of *"where"* the evidence was allocated and the question of *"when" can be* answered as the evidence was dispersed over time in continuous block of application memory. Also, the question of *"what"* the user was doing can be answered as the original text of information that was entered on Word application. The question of *"how"* evidence can be reconstructed can be traced to *line numbers 157353, 44749, 44750 and 44751.* This information was stored over time in continuous block and can be reconstructed to answer the question of *"how"*. This experiment demonstrated relevant user information that can be extracted from the physical memory of Windows

application while the applications are running and images were captured.

V. ANALYSIS

This research uncovers the amount of data that can be stored in the memory based on the series of experiments carried out. There are three elements of experiments considered in this paper. These includes, the Mean evidence in repetitions, Mean percentage of evidence found and Average length of evidence found in continuous block. *Mean Evidence Repetition* is the average number of repeated extracted evidence found in the memory of an application. *The mean percentage of evidence found* is calculated based on the number of character of the original user input matched with the extracted memory dump strings of character evidence. *The mean evidence in continuous block* is determined by matching the original user information with the extracted memory dump strings data of user information entered on the application. This process counts the lengths of identical character numbers of evidence found in the memory allocated and then matched the evidence found with the original user inputs. The result consists of both quantitative and qualitative assessment of experiment carried out for days. The quantitative assessment was as shown in Table II. Three sample of qualitative assessment was shown in Fig 1, 2 and 3. The evidence of user input was found repeated and stored in continuous block in the memory. By reconstructing the evidence found, whole fragment of what the user was doing was recovered. In this paper, we introduce new methods for extracting forensically relevant information from commonly used application on Windows computer systems. The question of "*who*" answered to the user that input the original texts information on application. The question of *where*, can be answered as evidence of where user input was found as allocated in the memory. The question of "*what*" user was doing is related to the original user input on applications. Finally, the questions of "*how*" can be answered as how the extracted memdump processes of evidence can be reconstructed.

V. FUTURE WORK

In the future, we will investigate the amount of user information that can be recovered from Windows application using some natural language to search and match evidence in the physical memory.

V. CONCLUSION

In this research, we have presented the extracted user information found in continuous block of evidence in the physical memory. This approach was based on how much data can be recovered when images were captured and the application was still running but user was not interacting with the system. Specifically, we have laid emphasis on the amount of relevant evidence found, the repeated evidence and evidence in continuous block. This approach describes the process of securing digital evidence resident in the physical memory. This experiment involves memory dumping, conversion of evidence into strings processes, and extraction of relevant data from the physical memory.

REFERENCES

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

[2] Digital Forensic Research Workshop (DFRWS). http://www.dfrws.org/2007/challenge/index.shtml. (Access date: 2 July 2010).

[3] Olajide F. Savage N., "On The Extraction Of Forensically Relevant Information From Physical Memory," in *World Congress on Internet Security (WORLDCIS-2011), Technically Co-Sponsored by IEEE UK/RI Computer Chapter*, London, 2011, pp. 248-252.

[4] Olajide F. Savage N., "On The Identification Of Information Extracted From Windows Physical Memory," *International Journal for Information Security Research (IJISR) ISSN 2042-4639 (Online)*, vol. II, no. 2, March 2011.

[5] Burdach M., "Windows Memory Forensic Toolkit;," *Journal of Information and Computing Systems*, vol. (5), no. 2, pp. 45-75, March 2007.

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

[7] Msuiche. Msuiche.net at:, Capture memory under win2k3 or vista with win32dd. (Access date: 9 March 2008).

[8] ManTech Memory. at:, ManTech International Corporation. Memory dd.(Acees date: 4 March 2010).

[9] Russinovich M.E. Solomon D.A., *Microsoft Windows Internal Covering Windows Server 2008 and Windows Vista*, 5th ed. Washington, USA: Microsoft Press, 2009.

[10] Agile Nigilant32. Agile Risk Management, "Nigilant32". (Access date: 16 April 2010).

[11] Betz C., "Mempaser analysis tool. (Access date: 28 April 2005)," in *DFRWS 2005 Forensic Challenge: http://www.dfrws.org/2005/challenge/memparser.shtml*, MA, pp. 100-115.

[12] Volatile Systems. The Volatility framework: Volatlile Memory Artifact Extraction Utility Framework. (Access date: 12 April 2009).

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

[14] Olajide F. Savage N., "Forensic Live Response And Events Reconstruction Methods In Linux Systems," in *PGNET The Convergence of Telecommunications Networking and Broadcasting*, Liverpool, December 2009, pp. 141-147.

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

[16] Olajide F. Savage N., "Digital Forensic Research And Method Of Extracting Relevant Information From Physical Memory Of Windows Systems," in *Fourth International Conference on Internet Technologies and Applications (ITA 11)*, Wrexam, Wales, 2011.