Thirdly, *the mean evidence in a continuous block* is the length of evidence found in a continuous block in the physical memory of an application. The length of the character is 10 in a row. This character length was determined by the process of matching the original user input information with the extracted memory dump strings data. This process counts the lengths of identical character numbers of evidence found in the memory, where evidence was allocated and then, matched the evidence found with the original user inputs information. It was discovered that the percentage amount of evidence found in PowerPoint is 25% being relevant evidence that was found on repetitions in the volatile memory. This information is stored in continuous block. The original user input contains slides of texts with commas, semi-colon, and full stop. Outlook Email 2007 recorded 53% of relevant data and as repeated in the memory. This information is stored in continuous block of the allocated memory with a paragraph of text with commas, semi-colon, full stop and special character. Moreover, we have large amount of evidence stored over time in Internet Explorer 7.0 applications with a remarkable amount of 71% of evidence recorded as dispersed in the memory. Memdump strings of user input were extracted from Internet Explorer 7.0 application. This information can be termed as application level information. The forensically relevant data was assessed to quantify what user is typing on the application, what user has been doing and what user is using the application for.

5. Future Work

In the future, we will investigate the quality of user input that can be recovered from Windows application memory using some commonly used English words.

6. Conclusion

In this research, we have presented the extracted user input found in the physical memory of Windows applications. This approach was based on how much data can be recovered when images were captured at 30 minutes, when the application was closed, but user was interacting with the system. We assumed user may be doing something else. Specifically, we have laid emphasis on the percentage amount of relevant evidence found, the repeated evidence and user input found in continuous block of evidence. This approach describes the process of securing digital evidence that is stored in the physical memory. This experiment involves memory dumping, data conversion of evidence into strings processes, and extraction of relevant user input. This information may be used as evidence in the court of law.

7. References

[1] DFRWS.(2007) 'Digital Forensic Research Workshop', http://www.dfrws.org/2007/challenge/index.shtml. (26 March 2010).

[2] EnCase. (2011) 'EnCase Guidance Forensic Software'. http://www.guidancesoftware.com/forensic.htm. (11 April 2010)

[3] FTK®. (2011) 'AccessData Digital Forensic Toolkit Software', http://accessdata.com/products/computer-forensics. (07 February 2010)

[4] Huebner Ewa, Bem Derek, Szezynska Magdalena, and Solomon Jason, (2007) 'User Data Persistence in Physical Memory', *Journal of Digital Investigation*, vol. 4, no. 2, pp. 68-72.

[5] Andrew Case. Andrew Cristina. Lodavico Marziale. Golden G. Richard. Vassil Roussev, (2008) 'FACE: Automated digital evidence discovery and correlation', *Journal of digital investigation,* , vol. 5, no. 8, pp. 65-78.

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

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

[8] Brian Carrier., (2005) '*File System Forensic Analysis'*, 1st Ed. Donnelley Mark M. Pollit, Ed. Crawfordsville, United States: Addison Wesley Professional.

[9] 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.

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

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

[12] Spafford Eugene and Carrier Brian, (2006) 'Categories of Digital Investigation Analysis Techniques based on the Computer History Model', *Digital Forensic Research Workshop (DFRWS)*, vol. 1, no. 2, pp. 1-28.

[13] Petron Nick and Walters AAron, (2007) 'Volatools: Integrating Volatile Memory into Digital Investigation Process', *Journal of Digital Investigation*, vol. II, no. 2, pp. 26-35.

[14] Sutherland Iain, Evans Jon, Tryfonas Theodore, and Blyth Andrew, (2008) 'Acquiring Volatile Operating System Data Tools and Techniques', *ACM SIGOPS Operating Systems Review - The ACM Digital Library published by Association for Computing Machinery*, vol. 42, no. 3, pp. 65-73.

[15] Garcia G.L., (2007) 'Forensic Physical Memory Analysis: An Overview of Tools and Techniques', *TKK T-110.5290 Seminar on Network Security*, Helsinki, Finland, pp. 305-320.

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

[17] Msuiche. (2008) 'Msuiche.net Capture memory under win2k3/vista/Windows7 with win32dd/win64dd' win32dd.msuiche.net/ (22 November 2010)

[18] Nigilant32. (2006) 'Agile Risk Management, Nigilant32 - Windows Incident Response Tool'. http://www.agilerm.net/publications_4_.html (16 April 2010).

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

[20] Walters AAron, Fraser Timothy, Petroni Nick, and Arbaugh William, (2007) 'FATKit: A Framework for the Extraction and Analysis of Digital Forensic Data from Volatile System Memory', *Journal of Digital Investigation*, vol. 3, no. 4, pp. 197-210.

