I.J. Modern Education and Computer Science, **2016, 6, 50-56**
Published Online June 2016 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijmecs.2016.06.07

Digital Forensics through Application Behavior Analysis

Shuaibur Rahman
Shaheed Zulfikar Ali Bhutto Institute of Science and Technology, Islamabad, Pakistan
Email: akhunzada.shoaib@gmail.com

M. N. A. Khan
Shaheed Zulfikar Ali Bhutto Institute of Science and Technology, Islamabad, Pakistan
Email: mnak2010@gmail.com

***Abstract*—**The field of digital forensic analysis has emerged in the past two decades to counter the digital crimes and investigate the modus operandi of the culprits to secure the computer systems. With the advances in technologies and pervasive nature of the computing devices, the digital forensic analysis is becoming a challenging task. Due to ease of digital equipment and popularity of Internet, criminals have been enticed to carry out digital crimes. Digital forensic is aimed to investigate the criminal activity and bring the culprits to justice. Traditionally the static analysis is used to investigate about an incident but due to a lot of issues related the accuracy and authenticity of the static analysis, the live digital forensic analysis shows an investigator a more complete picture of memory dump. In this paper, we introduce a module for profiling behavior of application programs. Profiling of application is helpful in forensic analysis as one can easily analyze the compromised system. Profiling is also helpful to the investigator in conducting malware analysis as well as debugging a system. The concept of our model is to trace the unique process name, loaded services and called modules of the target system and store it in a database for future forensic and malware analysis. We used VMware workstation version 9.0 on Windows 7 platform so that we can get the detailed and clean image of the current state of the system. The profile of the target application includes the process name, modules and services which are specific to an application program.

***Index Terms*—**Digital Forensic Analysis, Digital Crime Investigation, Live Forensic Analysis, Memory-based analysis, Exculpatory Evidence.

I. Introduction

Computer is an important part of our professional life now, with continuously increasing of unauthorized criminal activities digital forensic caught attention. But alongside advance technologies have prolonged criminal base which is regularly desire for discovering new way to commit crimes in more and more complicated ways. Since computer is fundamental part of our professional life now therefore, digital evidence is officially conventional in civil or criminal matters. Digital field is usually clear approximately the identification, securing, documentation and analysis of evidence for final presentation in a law of court.

Digital evidences can be in the figure of a fragment which is recovered from diverse storage devices such as history of browsing, email, and application document. The digital evidence might be also found in the shape of deleted files and different techniques are applied to recover the deleted data. Also, digital evidences can be recovered from different Storage and other medium installed in digital equipment such as cameras computers hardest, cell phones. Therefore, forensic investigation should be complete watchfully by documenting to exactly exhibit the chain of custody to put together the evidence adequate in court of law.

Traditionally the static analysis been used to examine equipment, where static analysis only used to analyze non volatile data store in permanently storage devices. What if the non volatile data or memory is so large it may be time consuming? On other hand the live analysis show more clean picture of the attack system and examiner are analyses the volatile memory and the running process, memory allocation and open network port of the attack system.

Some authorized issues about live forensics consist is inconsistency of end result when compared to diverse analysis techniques the indomitable state of rare data is inconsistent. If one investigator produces end result after analysis, the other examined end result is not similar to the first one. Another type issue is how do we auto organize a system to identify the attached devices [1].

In this paper we present a module for profiling the behavior of program application and trace down the process name what services it load and what modules it call when a specific application program launch. Before to creating the profile of specific application we must have data about that application program.

The computer crimes through Internet are increasing nowadays and digital forensics plays an important role to reduce their rampant and unchecked usage. Forensic investigation is used to find out the digital proof or evidence using different tools and it is inherently a difficult and complex process. Digital investigations take

Copyright © 2016 MECS

place in three main phases. The first phase is acquisition. In this phase, the investigator takes snapshot or images of digital device and replicates these images from the target device to some other device for in-depth analysis. The second phase is called analysis. In this phase, the investigator identifies the digital evidence using different types of techniques such as recovering the deleted files, acquiring information of user accounts, identifying information about the attached devices like USB, CD/DVD drives, external hard disks etc. The third phase is called reporting in which the investigator reconstructs the actual scenario based on the sequence of activities happened on the target system. Digital forensic analysis is divided into two main categories. The first one is dead or the static forensic analysis. During this analysis, all the target devices that are required in the analysis are shutdown. The second category is live analysis. During this type of analysis, the system stays in the boot mode and is kept alive to acquire pertinent information from the physical memory content.

Live analysis aims at gathering evidence from systems using different operations and techniques related to primary memory content. Live forensic is the most challenging kind of digital forensic investigations. To perform the live forensics, it is vital to understand the basic techniques and tools used in digital forensics. The investigator needs to acquire the complete image of a computer usage history as well as the current state through live forensic analysis tools. Though static analysis is kind of a developed part of digital forensics, but other techniques related to live analysis need to be developed to mitigate its weaknesses.

Due to rapid increase in memory size, the forensic investigators strongly recommend the live response approach for acquisition of volatile evidence. Through this technique the investigator can collect not only the information about live processes but also about the terminated and cache processes. Volatile memory analysis becomes an important piece of investigation because the physical memory could have potential evidence which investigator cannot find on the disk storage. To get hold of the incident, the volatile data acquisition is the initial step in digital investigation. Usually the investigator gathers the volatile data through live response, while the attacker might use different libraries to make the system calls to connect to the kernel and alter the volatile data.

This paper consists of six sections. Related work along with our motivation to conduct this research is provided in section II. Our proposed model is discussed in the section III, followed by describing the implementation of the proposed methodology through experimentation and experimental results in section IV. The discussion is provided in Section V. Conclusion along with future dimensions to this research are described in the last section.

II. Related Work

In [1], the authors described a few live analysis techniques such as Standard user interface technique, command shell or a Telnet connection. In [2], the authors explain over all process of digital forensic investigation, Forensic analysis is a process which is used to identify the digital evidence through different forensics tools. In [3], the authors alert on the countermeasure tools to deal with rootkits, so many countermeasure tools available, for application level rootkits the examiner use CD one of trusted tool to counter the change. In [4], the authors paying attention on taking out memory page file, because the page file have data which directly related to the RAM dump. In [5], the authors propose a model for digital analysis by contravention it into different stages such as collecting evidence, examining it, and then generate a report. The present live forensic approaches suffer from some issues such as credibility, fidelity and integrity which are difficult to verify. On the other hand, anti-forensic techniques might change the static data when acquired by the investigators using different tools. In live forensic analysis, both the evidence gathering process and the analysis itself take place at the same time, so it might be difficult to recognize whether the acquired data values are legal or otherwise [5]. Through this technique the investigator can collect not only the information about live processes but also about the terminated and cache processes. Volatile memory analysis becomes an important piece of investigation because the physical memory could have potential evidence which investigator cannot find on the disk storage. To get hold of the incident, the volatile data acquisition is the initial step in digital investigation. Usually the investigator gathers the volatile data through live response, while the attacker might use different libraries to make the system calls to connect to the kernel and alter the volatile data. In [6], the authors state the volatile memory techniques, due to quick increase in memory size, the forensic examiner stoutly advise the live forensic approach for acquire of volatile data. Through this approach the examiner collect information not only the live process but also about the executed and cache process.

In [7], the authors alert on vitalization forensic approach, virtualization approach are also used in digital forensics which after acquiring memory image and copy of hard desk boot that to some virtual machine and collect the useful evidence which examiner may not entrée in the real environment. In [8], the authors conducted live forensic analysis on two virtual machine installed two diverse operating system window XP and windows 7. In [9], the authors paying attention on virtualization forensic technique, standard computer forensic is accepted on a target system's RAM dump. In [10] the authors propose a technique how to identify encryption keys from the memory dump, in Linux using TruCrypt software of AES

Copyright © 2016 MECS

encryption algorithms with XTS mode. In [11], the authors through an attempt to identify the important information from memory dump about the latest browser sessions of logged in users. In [12], the authors focused on how to acquire live files from the operating system that are linked with the virtual machine. References [16-46] reviewed different techniques in different domains and reported their critical evaluations.

III. Proposed Technique

We map different processes, services and modules to an individual application program which are specific to it. We identify how many processes, services and modules are loaded when we launch an application program. Then we list down the processes, services and modules which are unique to the application program by separating the processes, services and modules which are directly linked to the operating system. We then store the unique processes, services and modules which are indigenous to an application program into a database table for future reference. We call this activity as profiling the application program. The saved profile of an application program can be used later on to determine the footprint of the execution of the application program for digital analysis purpose.

Process: An instant of computer program that is executed is called process. A process is a running program and having a specific set of data associated with it just like a task. A process is started when a program is launched.

Service: A service is associated to the application program. The services usually run in the background and do not have any user interface. The services may be initiated manually or starts automatically when the system is booted or restarted. Each service provides some unique functionality to an application program, therefore, there could be several services linked to an application program.

Module: Module is a DLL (Dynamic Link Library) or executable file which has multiple procedures and code. DLL files are coded in such a way that multiple processes can load it simultaneously. A single process can load multiple modules.

The key steps involved in our proposed technique are described below:

A. System Preparation

In this phase, we prepare the system to acquire clean image of the current state of the system. For this purpose, we installed VMware Workstation version 11.0 on the machine running under Windows operating system and created a new user account on Microsoft Windows 7. The purpose of installing VMware Workstation and creating this account was to get clean state of the system. In this way, we only get the necessary operation system related running processes on the machine without the fear of third party application programs that are set to run in the auto mode. After running the VMware Workstation, we launch different application programs such as MS-Word, MS-Excel etc. This account will also have some permission to access some of applications which are installed on administrator account such as "What's Running" software and Microsoft Office.

B. Image Acquisition

There are several software applications which can be used for file system image acquisition such as FTK imager, "what's running". For experimentation in this study, we used the "what's running" software to acquire the image and analyze it. For acquisition of images of different application programs, we perform the following steps:

- Launch "what's running" application.
- Take image or snapshot of the current state of the system and save it on the local hard drive.
- Run the target application with and without opening a file in it.
- Take image or snapshot of the current state of the system and save it on the local hard drive.
- Close the application program.
- Take image or snapshot of the current state of the system and save it on the local hard drive.
- Close "what's running" application.
- Log out from the user account.

The dump of images is stored in XML file on a local drive. The dump of the images is obtained in such a way that it can be easily loaded into a database and compared with other images so that an investigator can easily analyze it.

C. Image/ Snapshot Comparison

In this phase, we compare the saved images with the current state of the system. After comparison we will get the process name, associated services and modules which are idiosyncratic to that application program. For example, if we are going to compare the image which we had taken when MS-Word application was running on the machine and compared it with the image or current state of a system on which the MS-Word was not running, then we can effectively identify the processes, services and modules eccentric to the MS-Word. We also used the "what's running" tool for comparing the running processes, services and modules on a system in either state i.e., by launching an application program and by not executing the application program at all.

We can also compare the two images which have been taken after and before the specific activity, by acquiring system snapshot after the running the specific application and before.

D. Report Generation

After comparison of the system activity state before and after execution of the specific application program, we extract all the unique services, process and module intrinsic to the application program. We repeat the same process for different application programs. We verify stored names of these unique processes, services and

modules in a database which are distinctive to different specific application programs respectively. The purpose of this is to use these application profiles in future forensic analysis of a system. By using the proposed profiling approach, the digital forensic investigator can easily identify and search specific process name, services and module in a database and determine which of the application programs were used to compromise the target system.

IV. Experiment and ResultS

We have performed a series of experiments to validate our proposed technique. We mainly used "what's running" tool which was launched on a virtual machine on Window 7 platform. The "what's running" tool provides us list of all the process, services and modules associated to each application as well takes snapshot of all the running processes and compares it with the other snapshots. Our repeated experiments observed the unique process name, services and module of Microsoft Word 2007, Excel 2007, PowerPoint 2007, Adobe Reader 10.13, internet explorer and google chrome we saved them in a database for future digital forensic analysis use. Following is detail of our experimental results.

To get unique processes, services and modules we have to compare first four experiments very meticulously because as we know that first four experimented applications used in this study are products of Microsoft Corporation and are part of Microsoft Office, so there would be some shared modules and services. In this way, we obtain trace of the processes, modules and services which specific to each application. The results for MS-Word, MS-Excel, MS-PowerPoint, Adobe Reader and Google Chrome profiles traced through our experiments is shown in Table 1, Table 2, Table 3, Table 4 and Table 5 respectively.

Table 1. MS-Word Profile

Sr #	Process Name	Services	Modules	Module Description
1	WINWORD.EXE	-	Wwlib.dll	Wwlib.dll is part of Microsoft office word it used to run the application stably and properly on PC.
2			Wwintl.dll	Wwintl.dll is part of Microsoft office word, it uses by window to load lot of setting including registry setting, color setting etc of application.
3			Msostyle.dll	Msostyle.dll is part of Microsoft office suit; it is providing the functionality to window platform and Microsoft application. As part of office suit it create IME (input method editor) a feature which is allow user to input symbol and character which is not on the ordinary keyboard, also encoding languages.
4			Msgr3en.dll	Msgr3en.dll is part of Microsoft office word, it responsible for Microsoft English Natural Language Server.
5			Nlsdata0009.dll	Nlsdata0013.dll is the part of Microsoft windows operating system having Microsoft English Natural Language Server Data and Code
6			Winword.exe	Winword.exe is part of Microsoft office WinWord application which loads the WinWord application on PC.

Table 2. MS-Excel Profile

Sr #	Process Name	Services	Modules	Module Description
1	EXCEL.EXE	-	Nlsdata0013.dll	Nlsdata0013.dll is the part of Microsoft windows operating system having Microsoft Neutral Natural Language Server Data and Code.
2			Photometadatahandler.dll	Photometadatahandler.dll is a part of Microsoft windows operating system it is responsible for photo thumbnail extraction.
3			Excel.exe	Excel.exe is part of Microsoft office Excel application which loads the Excel application on PC.
4			Pnpts.dll	Pnpts.dll is belong to Microsoft windows operating system and responsible for PlugPlay troubleshooting.
5			Rasadhlp.dll	Rasadhlp.dll is belong to Microsoft windows operating system and responsible for component Remote Access AutoDial Helper.

Table 3. MS-PowerPoint Profile

Sr #	Process Name	Services	Modules	Module Description
1	POWERPNT.EXE	-	Ppintl.dll	Ppintl.dll is belong to Microsoft office PowerPoint, it uses by window to load lot of setting including registry setting, color setting etc of application.
2			Ppcore.dll	Ppcore.dll is one of window common file which is applied to PowerPoint and responsible to process and load the application.
3			Powerpnt.exe	Powerpnt.exe is part of Microsoft office PowerPoint application which loads the PowerPoint application on PC.

Table 4. Adobe Reader Profile

Sr #	Process Name	Services	Modules	Module Description
1	AcroRd32.exe	-	Acrord32.dll	Acrord32.dll is part of Adobe Reader application used to setting and configration.
2			Acrord32.exe	Acrord32.exe is part of Adobe Reader application which loads the Adobe Reader application on PC.
3			Agm.dll	Agm.dll or Adobe Graphics Manager is responsible to repurposing the graphics for print etc.
4			Cooltype.dll	Cooltype Typography Engine
5			Bib.dll	Bib.dll is part of Bravo Binder Interface of Adobe Reader which is use to bind it with other Adobe Reader component to run PDF file properly and other task etc.
6			Ace.dll	Adobe Color Engine
7			Axe8sharedexpat.dll	AXE Shared EXPAT (export and publish PDF files.)
8			Ccme_base.dll	Ccme_base.dll is library from RSA - The Security Division of EMC installed with Adobe Reader.
9			Ia32.api	Ia32.api is belongs to Adobe Acrobat Internet Access.
10			Acroform.api	Acroform.dll is part of Adobe Reader and using interaction with PDF and server.
11			Axsle.dll	Axsle.dll is belongs to AXSLE created by Adobe Systems Incorporated. (Adobe XSLT Engine).
12			Ppklite.api	Ppklite.api is part of Adobe Reader which is responsible for setting and process of the application.

Table 5. Google Chrome Profile

Sr #	Process Name	Services	Modules	Module Description
1	CHROME.EXE	-	chrome.exe	Main executable of Google Chrome.
2			chrome_elf.dll	Handle errors messages.
3			chrome.dll	It is loaded within chrome.exe process to make the web efficient.
4			chrome_child.dll	Containing the source code and binary data of users.
5			ffmpegsumo.dll	Support video in browser.

V. Discussion

Experiments were conducted on different applications such as Microsoft Word 2007, Microsoft Excel 2007, Microsoft PowerPoint 2007 Adobe Reader 10.13, internet explorer and Google Chrome. The aim of this study to find unique process name, services and modules related to running application, the configuration of system is we installed VMware Workstation version 11.0 on the machine running under Windows operating system and created a new user account on Microsoft Windows 7. The purpose of installing VMware Workstation and creating this account was to get clean state of the system. To trace down these specific process name, services and module we used "what's running" tool there are plenty of tool which used for this purpose but the "what's running" tool is used to acquire more clean and detail image of the running processes of the system as well as it provide the comparison functionality between two images. We did five to six experiments for each application because of running system there were other process relates to operating system so we get rid of that, we right down the common modules which mostly come in experiment. For confirmation of the module we also right down the description of specific modules. The first three experiments is on Microsoft application so we found some of module is overleaping so this will not be include in profile we only include those process name, services and modules which only specific to each application. The problem we faced in these experiments is we did not find the unique services of the target application. For the purpose of digital forensic and future use, we store each profile in database.

Internet Explorer executes two process when we run it for the first time one for the main application and the other is for child tab while Google chrome has Multi-process architecture it because why it open multi process when someone run this application, it puts the plug-ins and the web app in separate processes. Number of processes is about how much extension you have on browser and how much tab open.

VI. Conclusion and Future Work

In this paper we have present a module profiling the behavior of the running application that what specific process name have a target application, what services it load when launch and what module it call. We performed experiments using MS-Word, MS-Excel, PowerPoint and Adobe Reader, we take image of the system after and before launch of target application save it on local drive, after that we compare both the image to find the unique process name, services and module which related to target application. We used "what's running" tool for both image acquisition and comparing activity. The problem we faced in this experiment is we did not find the unique services yet.

The future direction of our work is, first we will conduct experiment on browsers and Skype application and our most focus in on services which uniquely loaded during execution of target application second how much memory allocated to a specific process and what is

impact on memory if we change the target document of specific application.

REFERENCES

[1] B. Hay, K. Nance, and M. Bishop, "Live Analysis: Progress and Challenges," *IEEE Security and Privacy* vol. **7**, no. 2, pp. 30–37 (Mar. 2009).

[2] S. Yadav, "Analysis of Digital Forensic and Investigation," VSRD-*IJCSIT*, vol. 1, no. 3, pp. 171-178 (2011).

[3] B. D. Carrier, "Risks of live digital forensic analysis," *Communications of the ACM*, vol. *49* no. 2, pp. 56-61 (2006)

[4] A.Savold., and p. Gubian, "Towards the virtual memory space reconstruction for windows live forensic purposes," In *IEEE Systematic Approaches to Digital Forensic Engineering, 2008. SADFE'08. Third International Workshop on, pp*. 15-22 (2008, May).

[5] L. Wang, R. Zhang., and S. Zhang, "A model of computer live forensics based on physical memory analysis," In IEEE *Information Science and Engineering (ICISE), 2009 1st International Conference on,* pp. 4647-4649 (2009, December).

[6] A. Aljaedi., D. Lindskog, P. Zavarsky, R. Ruhl, and F. Almari,, "Comparative Analysis of Volatile Memory Forensics: Live Response vs. Memory Imaging," In IEEE *Privacy, security, risk and trust (passat), 2011 ieee third international conference on and 2011 ieee third international conference on social computing (socialcom),* pp. 1253-1258 (2011, October).

[7] S. Mrdovic, A. Huseinovic, and E. Zajko, "Combining static and live digital forensic analysis in virtual environment," In IEEE *Information, Communication and Automation Technologies, 2009. ICAT 2009. XXII International Symposium on,* pp. 1-6 (2009, October).

[8] F. Gianni, and F. Solinas, "Live Digital Forensics: Windows XP vs Windows 7," In IEEE *Informatics and Applications (ICIA),2013 Second International Conference on,* pp. 1-6 (2013, September).

[9] L. Zhang, D. Zhang, and L. Wang, "Live digital forensics in a virtual machine," In IEEE *Computer Application and System Modeling (ICCASM), 2010 International Conference on,* vol. 4, pp. V4-328 (2010, October).

[10] S. Balogh, and M. Pondelik, "Capturing encryption keys for digital analysis," In IEEE *Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), 2011 IEEE 6th International Conference on,* vol. 2, pp. 759-763 (2011, September).

[11] I. Mohanty, and R. L. Velusamy, "Information Retrieval From Internet Applications For Digital Forensic," *arXiv preprint arXiv:1209.3590* (2012).

[12] V. Meera, M. M. Isaac, and C. Balan, " Forensic acquisition and analysis of VMware virtual machine artifacts," In IEEE *Automation, Computing, Communication, Control and Compressed Sensing (iMac4s), 2013 International Multi-Conference on,* pp. 255-259 (2013, March).

[13] Y. Kim, S. Lee, and D. Hong, "Suspects' data hiding at remaining registry values of uninstalled programs," In ICST *Proceedings of the 1st international conference on Forensic applications and techniques in telecommunications, information, and multimedia and workshop on.* p. 32 (2008, January).

[14] C. H. Yang, and P. H. Yen, "Fast deployment of computer forensics with USBs," In IEEE *Broadband, Wireless Computing, Communication and Applications (BWCCA), 2010 International Conference on,* pp. 413-416 (2010, November).

[15] S. Mrdovic, and A. Huseinovic, "Forensic analysis of encrypted volumes using hibernation file," In IEEE *Telecommunications Forum (TELFOR), 2011 19th on*, pp. 1277-1280 (2011, November).

[16] Iqbal S., Khalid M., Khan, M N A. A Distinctive Suite of Performance Metrics for Software Design. International Journal of Software Engineering & Its Applications, 7(5), (2013).

[17] Iqbal S., Khan M.N.A., Yet another Set of Requirement Metrics for Software Projects. International Journal of Software Engineering & Its Applications, 6(1), (2012).

[18] Faizan M., Ulhaq S., Khan M N A., Defect Prevention and Process Improvement Methodology for Outsourced Software Projects. Middle-East Journal of Scientific Research, 19(5), 674-682, (2014).

[19] Faizan M., Khan M NA., Ulhaq S., Contemporary Trends in Defect Prevention: A Survey Report. International Journal of Modern Education & Computer Science, 4(3), (2012).

[20] Khan K., Khan A., Aamir M., Khan M N A., Quality Assurance Assessment in Global Software Development. World Applied Sciences Journal, 24(11), (2013).

[21] Amir M., Khan K., Khan A., Khan M N A., An Appraisal of Agile Software Development Process. International Journal of Advanced Science & Technology, 58, (2013).

[22] Khan, M., & Khan, M. N. A. Exploring Query Optimization Techniques in Relational Databases. International Journal of Database Theory & Application, 6(3). (2013).

[23] Khan, MNA., Khalid M., ulHaq S., Review of Requirements Management Issues in Software Development. International Journal of Modern Education & Computer Science, 5(1),(2013).

[24] Umar M., Khan, M N A., A Framework to Separate NonFunctional Requirements for System Maintainability. Kuwait Journal of Science & Engineering, 39(1 B), 211-231,(2012).

[25] Umar M., Khan, M. N. A, Analyzing Non-Functional Requirements (NFRs) for software development. In IEEE 2nd International Conference on Software Engineering and Service Science (ICSESS), 2011 pp. 675-678), (2011).

[26] Khan, M. N. A., Chatwin, C. R., & Young, R. C. (2007). A framework for post-event timeline reconstruction using neural networks. digital investigation, 4(3), 146-157.

[27] Khan, M. N. A., Chatwin, C. R., & Young, R. C. (2007). Extracting Evidence from Filesystem Activity using Bayesian Networks. International journal of Forensic computer science, 1, 50-63.

[28] Khan, M. N. A. (2012). Performance analysis of Bayesian networks and neural networks in classification of file system activities. Computers & Security, 31(4), 391-401.

[29] Rafique, M., & Khan, M. N. A. (2013). Exploring Static and Live Digital Forensics: Methods, Practices and Tools. International Journal of Scientific & Engineering Research 4(10): 1048-1056.

[30] Bashir, M. S., & Khan, M. N. A. (2013). Triage in Live Digital Forensic Analysis. International journal of Forensic Computer Science 1, 35-44.

[31] Sarwar, A., & Khan, M. N. (2013). A Review of Trust Aspects in Cloud Computing Security. International Journal of Cloud Computing and Services Science (IJCLOSER), 2(2), 116-122.

[32] Gondal, A. H., & Khan, M. N. A. (2013). A review of fully automated techniques for brain tumor detection from MR images. International Journal of Modern Education

and Computer Science (IJMECS), 5(2), 55.
[33] Zia, A., & Khan, M. N. A. (2012). Identifying key challenges in performance issues in cloud computing. International Journal of Modern Education and Computer Science (IJMECS), 4(10), 59.
[34] Ur Rehman, K., & Khan, M. N. A. (2013). The Foremost Guidelines for Achieving Higher Ranking in Search Results through Search Engine Optimization. International Journal of Advanced Science and Technology, 52, 101-110.
[35] Khan, M., & Khan, M. N. A. (2013). Exploring query optimization techniques in relational databases. International Journal of Database Theory & Application,6(3).
[36] Shehzad, R., KHAN, M. N., & Naeem, M. (2013). Integrating knowledge management with business intelligence processes for enhanced organizational learning. International Journal of Software Engineering and Its Applications, 7(2), 83-91.
[37] Ul Haq, S., Raza, M., Zia, A., & Khan, M. N. A. (2011). Issues in global software development: A critical review. An Appraisal of Off-line Signature Verification Techniques 75 Copyright © 2015 MECS I.J. Modern Education and Computer Science, 2015, 4, 67-75 Journal of Software Engineering and Applications, 4(10), 590.
[38] Zia, A., & Khan, M. N. A. (2013). A Scheme to Reduce Response Time in Cloud Computing Environment. International Journal of Modern Education and Computer Science (IJMECS), 5(6), 56.
[39] Tariq, M. & Khan, M.N.A., (2011). The Context of Global Software Development: Challenges, Best Practices and Benefits. Information Management & Business Review, 3(4).
[40] Shahzad, A., Hussain, M., & Khan, M. N. A. (2013). Protecting from Zero-Day Malware Attacks. Middle-East Journal of Scientific Research, 17(4), 455-464. [38] Khan, A. A., & Khan, M. (2011). Internet content regulation framework. International Journal of U-& EService, Science & Technology, 4(3).
[41] Kaleem Ullah, K. U., & MNA Khan, M. K. (2014). Security and Privacy Issues in Cloud Computing Environment: A Survey Paper. International Journal of Grid and Distributed Computing, 7(2), 89-98.
[42] Abbasi, A. A., Khan, M. N. A., & Khan, S. A. (2013). A Critical Survey of Iris Based Recognition Systems. Middle-East Journal of Scientific Research, 15(5), 663-668.
[43] Khan, M. N. A., Qureshi, S. A., & Riaz, N. (2013). Gender classification with decision trees. Int. J. Signal Process. Image Process. Patt. Recog, 6, 165-176.
[44] Ali, S. S., & Khan, M. N. A. (2013). ICT Infrastructure Framework for Microfinance Institutions and Banks in Pakistan: An Optimized Approach. International Journal of Online Marketing (IJOM), 3(2), 75-86.
[45] Mahmood, A., Ibrahim, M., & Khan, M. N. A. (2013). Service Composition in the Context of Service Oriented Architecture. Middle East Journal of Scientific Research, 15(11).
[46] Masood, M. A., & Khan, M. N. A. (2015). Clustering Techniques in Bioinformatics. I.J. Modern Education and Computer Science, 2015, 1, 38-46.

Authors' Profiles

Shuaibur Rahman obtained MS degree in Computer Science from Shaheed Zulfikar Ali Bhutto Institute of Science and Technology, Islamabad, Pakistan. His research areas include artificial intelligence, digital forensics and data mining techniques.

M.N.A. Khan obtained D.Phil. degree from the University of Sussex, Brighton. His research interests are in the fields of software engineering, cyber administration, digital forensic analysis and machine learning techniques.