Methodology for an Open Digital Forensics Model based on CAINE

Casimer DeCusatis
School of Computer Science and Mathematics
Marist College
3399 North Road
Poughkeepsie, NY USA
casimer.decusatis@marist.edu,

Aparicio Carranza, Alassane Ngaide,
Sundas Zafar, and Nestor Landaez
Computer Engineering Technology Department
CUNY New York City College of Technology
300 Jay Street, Brooklyn, NY USA
aparicio.arrranza@cuny.edu

***Abstract* — With the widespread adoption of public and private cloud computing environments, in addition to traditional enterprise-class data centers, cybersecurity has become increasingly important. In particular, forensic analysis of digital evidence has received increased attention. We investigate a relatively new suite of cyber-forensic tools in the open source CAINE Linux distribution. We propose how tools such as Guymager, Autopsy, Fred, and Photorec can be applied as part of a four tier forensic architecture. Experimental results are provided which demonstrate the application of these tools in a practical investigation.**

Keywords— CAINE, cybersecurity, forensics

I. Introduction

In the modern information age, data centers provide the hardware and software infrastructure which is critical to both Fortune 500 companies and startups. It is difficult to overestimate the importance of digital infrastructure and its impact on markets including business, finance, transportation, health care, education, entertainment, and government. With the widespread adoption of cloud computing, the majority of valuable data is migrating from private enterprise data centers into a public or hybrid cloud model. This has unfortunately only increased the risk of cybercrime, and reports of significant data breaches are almost a regular occurrence [1]. This had led to the development of advanced digital forensic techniques to collect admissible evidence following cybercrimes, and assist in identifying and prosecuting those responsible. Several frameworks have been proposed for structured forensic analysis in an effort to make this vital activity more efficient. However, there has been comparatively little discussion around how to implement these frameworks in practice. The potential use of open source tools is a particularly attractive approach to this problem, since open source enables rapid development of security forensics tools and places these tools at the disposal of the security community for little or no cost. Within the past several months, a new suite of open source forensics tools has become available as part of the CAINE specialized distribution of Linux [2]. Unfortunately, there is no overarching documentation for CAINE describing its embedded open source forensic tools in detail, so the overall capabilities need to be discovered mostly through trial and error. In this paper, we investigate how the forensics tools integrated with CAINE may be applied to a structured cybersecurity evidence gathering procedure. First, we review efforts to develop a broadly accepted architecture for cyber-forensics. Then we demonstrate use cases for the main tools within CAINE suitable for disk imaging, file recovery, and hive management, such as Autopsy, Guymager, Fred, and Photorec. We will then outline how these tools can be applied in succession to implement the key features of a cyber-forensics architecture. The relative strengths and weaknesses of this architecture will be discussed, based on experimental use cases which we have implemented using CAINE.

II. Forensic Architecture Development

Digital evidence collection is based on concepts taken from traditional (physical) forensic investigation methods, as practiced by law enforcement agencies such as the FBI. This includes fundamental principles which must be preserved in cyber-forensics in order for the results to be admissible as evidence during criminal prosecution. Original evidence must be preserved during the investigation, in the same way that physical evidence should be immune to tampering after it is collected. Digital evidence is considered "hidden" evidence, in the same category as fingerprint evidence; in its natural state, digital evidence cannot be known by the content in the physical object which holds such evidence. Thus investigative reports are required to explain the examination process and its limitations. Unfortunately, digital evidence is very fragile, and can easily be altered, damaged, or destroyed by improper examination. It is not easily kept in its original state [3], so the first step in an investigation usually involves creating an image of the original computer disk. This image is then examined while the original evidence is preserved.

There have been many efforts to develop a consistent, set of guidelines for digital forensic investigation. The definition of digital evidence, and digital forensics, comprises a wide range of computer activity and encompasses many types of media,

including hard drives, USB memory sticks, cell phones, and network traffic [4]. This includes evidence that may not be entered into a court of law, but still has investigative value [5]. The U.S. Department of Justice published the first electronic crime scene investigation model for first responders in 2001 [6]. This approach was enhanced to create the first academically driven models developed at the inaugural digital forensics research workshop [7]. This inspired subsequent work such as the Abstract Digital Forensics Model [8], the Integrated Digital Investigation Process model (IDIP) [3], and various other digital investigation methodologies [9-11].

While all of these models differ in the number and details of their process steps, they can conceptually be reduced to a basic iterative four tier approach [12]. The first tier which is the preparation or collection phase involves the search, recognition, collection, and documentation of electronic evidence. To preserve the original evidence, this tier often required creating an image of the original digital media without disturbing its contents. The second tier is the examination phase, which helps to make digital evidence visible and explains its origins and significance. This includes revealing hidden or obscured information. The third tier is the analysis phase, which involves studying the product of the examination tier for its relative importance to the case under investigation. Finally, the fourth or reporting tier includes documenting the results of the examination process and limitations of the investigation. Ideally, this phase will be enabled throughout the course of the investigation by features built into the digital forensic tools. These tiers may each contain additional rules and procedures specifying best practices for data handling, which we will not address in this paper for the sake of brevity. In the following sections, we describe a set of tools available in CAINE which address this four tier approach. We then demonstrate experimentally how the tools may be applied in a practical use case.

III. Experimental Results using CAINE

Since there are many different tools designed to address specific areas of computer forensics, it would be advantageous to collect several tools optimized for specific tasks into a single distribution. There are also benefits associated with using open source tools, including rapid innovation driven by a global development community and free or inexpensive access to a broad cross-section of security professionals. Such tools would also facilitate training and education efforts to address the lack of security practitioners and service industry professionals [13].

In October 2014, a new, highly specialized open source Linux distribution became available to digital forensics practitioners. The project manager is Nanni Bassetti and the distro is known as CAINE (short for Computer Aided Investigative Environment, but also named after the popular character Horatio Caine on the television series CSI Miami). The latest edition of CAINE is based on Ubuntu 12.04 and Linux kernel 3.2, but with the GNONE 2 fork known as MATE providing the desktop environment. Just as other specialty distros such as Kali Linux focus on penetration testing, CAINE has been created to focus on digital forensics by combining several useful tools into one package. CAINE differs from many other specialized distros because it also provides a suite of general purpose desktop tools which allow it to be used as a classic Ubuntu system while still making advanced forensic tools available. This means that it's not necessary to install a more general purpose distro for common desktop applications. CAINE eliminates the need to switch back and forth between a general purpose environment and the advanced forensic tool features. CAINE can also run in a virtual partition along with VMWare or other hypervisors. Normally, a general purpose distro would not be suitable for forensic purposes, because it automatically mounts all available drives as read/write. This poses several problems, including changing the "last mounted" times and potentially erasing data (including hidden data) when writing to the drive. To avoid these issues, CAINE never automatically mounts any device. Mounting is only possible through an applet called Mounter, accessible to the user through the system tray or command line (via the "mount" command). This applet also allows users to toggle the system policy for all future mounts from read/write to read-only and back again.

CAINE includes many specialized tools, including memory forensics, databases, mobile devices (including iPhone and Android), and networking. In this paper, we will focus on four tools which enable the processing of a basic forensics report, and which can be mapped to the four tier forensics architecture discussed previously. While some of these tools are documented as stand-alone open source appliances, and there is some documentation in the CAINE user community, there is currently no over-arching documentation for how and when to apply each tool as part of a forensic investigation. By experimenting with CAINE, we hope to address this gap.

A. GUYMAGER

Guymager is an open sourced forensic disk imaging tool included with CAINE 3.0. It is a Qt-based forensic imager which is capable of producing image files in multiple formats. Its structure is based on various threads for reading, MD5 calculation, and writing [14]. Guymager also includes a compression engine, and uses multi-processor and hyper-threading machines for improved performance. Guymager is used to compact a disk image into one file for subsequent forensic analysis or hidden data recovery, without the actual image being harmed. This supports the principle of preserving evidence which will be admissible in a court of law, and provides the first step in our forensic framework.

B. AUTOPSY

Autopsy is supplied with CAINE< although it was previously available as the GUI interface of The Sleuth Kit (TSK) project. Autopsy provides a library and collection of command line tools that are designed to investigate and analyze images. It can be used to recover lost, deleted, or hidden (steganographic) data. Autopsy contains modules that help it perform actions such as time line analysis, hash filtering, keyword search, extract web artifacts and much more. As open source code, Autopsy allows users to input their own customized modules. It is also known for its speed because it uses multiple cores to run background tasks in parallel. These functions are useful in the second and thir d tiers of our framework.

C. FRED

Computers contain various files that log/store data regarding actions that were performed on the system. Different forensic tools are designed to analyze specific formats of files. One important class of files is the Linux or Windows Registry, a hierarchal database that stores the settings and options on the operating system for applications as well as users. The Windows registry, for example, consists of keys (folders) and key values (files), as well as data stored in disk files called hives. When a user boots a Windows system, that information is recorded on the hive branch of HKLM/HARDWARE or when a user logs into the system the information is stored under the branch HKEY_USERS [15].

For the analysis of registry hives, CAINE includes the Forensic Registry Editor (FRED), a cross-platform registry hive editor. The FRED project was created from the desire for an efficient Linux and Windows registry hive viewer. However, FRED differs from other registry viewers since it contains additional functionality, including a hex viewer with a data interpreter and reporting function with extension templates [15]. Registry Editors can be used in creating and modifying registry files, importing and exporting data, finding specific strings and key names, etc. FRED uses four different hive files: NTUSER.dat, SAM, SOFTWARE, and SYSTEM to generate a report of the interpreted data. It also shows the node tree, key list, and hex viewer. These reports are used for the second and third tiers of the forensic analysis framework, because they show the last actions performed on the system. Analysis of actions can help locate any suspicious activity. FRED also support the basic forensics principle of not leaving any trace on the system being analyzed which could cause results to be altered. Normal forensic procedure would require imaging these files to avoid altering them, however hive files running on a system cannot be copied. Running FRED as a stand-alone tool on the operating system would also run the risk of affecting data and damaging files. Both of these concerns are alleviated by running FRED under the CAINE operating system. .

D. PHOTOREC

Photorec is an open source data carver, used to recover deleted, lost, or damaged files. It is designed to avoid any attempt to write to the lost or damaged media that the user is trying to recover. Instead, the files are processed within the directory where Photorec is located.

In practically all storage media, files are stored in sectors called data clusters, which work in sequential order. When a file is saved it is broken down into many small portions, each of which is given an address which leads to the next portion. This is done in order to decrease the seek time of mechanical media such as hard drives, although similar approaches have benefits for solid state media as well. When a file is deleted the meta-information is lost but not completely forgotten. The files are still in their place but the blocks have been given permission to rewrite any new information over them. Photorec can scan the file system, identify the type of media formatting (including volume boot records such as FAT and NTFS or Superblocks), and determine block size based on a continuously growing open source database of file signatures. Photorec compares file checksums against this database to determine when a file has been completely recovered. This approach is successful at finding even small or fragmented files. By extracting files in this manner, Photorec is useful in the second through fourth tiers of our forensic architecture.

IV. EXPERIMENTAL RESULTS

We experimentally validated the use of these four tools in forensic analysis of a suspect disk drive. Tier 1 data recovery was done using Guymager to image the disk, and Tier 2-3 analysis was performed using Autopsy, Fred, and Photorec. Built-in features for all these tools (particularly Photorec) facilitate tier 4 documentation throughout the forensic process.

In order to image a disk, the user has to first click on the Menu within CAINE and go to Forensic Tools > Guymager. The first window that opens is the list of mounted disks. Information about the disks such as the size, serial number, device, and state are shown. The next step is to right click on the image and select “Acquire Image”. From there the proper image directory location has to be chosen by clicking on the “…” button. Insert case number, examiner, description, image filename and info filename in the corresponding boxes. The remaining settings should match the screenshot in figure 1. Once the proper information and settings have been made within the “Acquired Image” window, the user can then click the Start button. Clicking Start will begin

the process of creating the disk image within the specific directory that was assigned. This can be seen because the status of the process will be shown as “Running”. When the disk image has been created, the status will change to “Finished”.

Next we employ Autopsy to analyze the disk image. Autopsy can be accessed through CAINE via Menu > Autopsy. Clicking Autopsy will automatically open a terminal emulator with command line interface. It will prompt the user to enter the URL provided into a browser to open up the GUI of Autopsy. From here the user can click on a button which opens an existing case, creates a new one or ask for help. When the user creates a new case, they have to provide a name for the case along with names of the investigators of the case. There is also an option to describe the case. This prompt is important because it automatically keeps track and creates a report as the investigation proceeds.

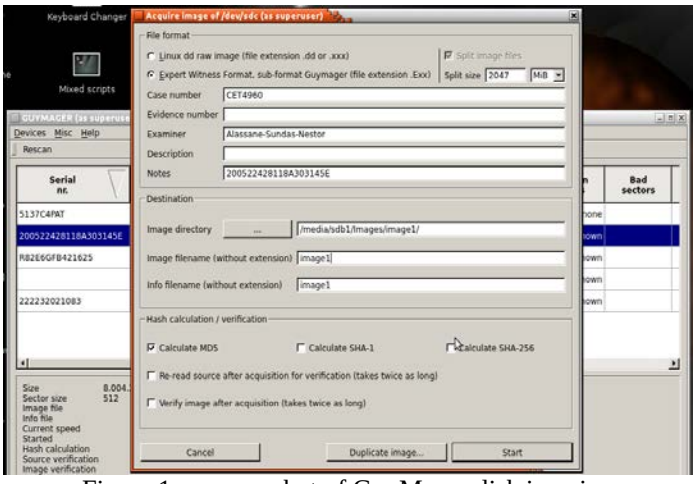

Figure 1 – screen shot of GuyMeyer disk imaging

`

This illustrates how the tier 4 requirements are enabled throughout the investigative process. In order to analyze an image of data, the user needs to create a new case for each image as shown in figure 2 We found this to be a bit cumbersome for investigations containing multiple images, since the Autopsy GUI can only organize this information based on input provided by the user. Care must be taken to use consistent filenames in order to keep track of the results.

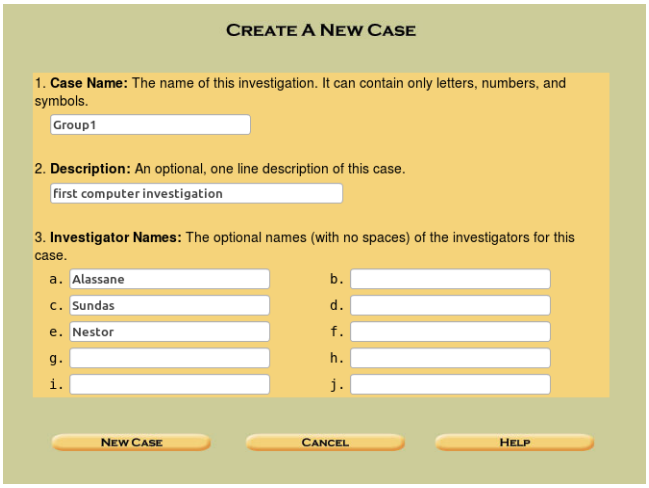

Figure 2 - Creating a New Case

After creating the case, the user is required to specify a host as shown in figure 3. Other information is optional, however we found it helpful to include the paths for the Alert and Ignore data bases. These paths can help filter the results, and files that have been tampered with can be identified in this manner. The main part of the analysis occurs after adding the host, when the user is prompted to add an image file. The proper file type and format must be selected in order to Autopsy to analyze the image. The Hash databases option provides additional information, and buttons for the file activity time line, notes and event sequencer allow the user to create time lines and notes for the investigation.

Figure 3 - Creating a New Host

Once the host and case are created, the user will be prompted to add the host to the case. In our experiment the new host "Alassane" was selected. The next step consists of choosing the image to analyze. The location chosen should be the same as where the file was created using Guymager. Select the type of image file as disk and the import method as Symlink. The next window will show a summary for the image that is being added, such as the mount point or C drive and the file system type (fat32). Afterwards, the user has to choose which image or volume they want to analyze. In figure 4 the mounted images are shown. The second image was selected and Autopsy allowed options of File Analysis, Keyword Search, File Type, Image Details, Meta Data, and Data Unit.

Figure 4 – Mounted images within Autopsy

The user can enter a keyword, string or expression that can be searched within the image. The type of data can also be selected, such as ASCII or Unicode. In this case "strings" was entered as the keyword to search. A set of predefined search options are also available. After clicking "Search", a new window shows that in our case, 92 occurrences of "strings" were found with the selected settings. Subsequent views expand on this screen showing all of the occurrences within that unit. These can be viewed in both HEX and ASCII.

The next option chosen was "File Analysis" of the image. In File Browsing mode the analyzer can view a list of files within the image. Input boxes allow the user to search for specific directories or files by name. The date the files were created, last date accessed, size of the files, and meta data are given. To view the deleted files on the image, select the "Deleted Files" button. All of the deleted files on the image are seen, along with information on when they were created or last used. Finally, the system highlights files which are in data recovery mode. These files can be opened to view their contents (which may still be encoded). It is possible to search for deleted files that have specific keywords or names, which cannot be done on a regular operating system. Reports can be generated in ASCII or HEX to facilitate tier 4 documentation.

To successfully analyze the hive files on this system without harming them, the user has to boot the CAINE operating system from a CD. The hard disk and any other USB storage connected to the host computer will be mounted. From there the user can double-click the mounted hard disk and explore the files of the system without any restrictions. It is not possible to tamper with

the hard disk since it is read only. The first registry of hive files we scanned was NTUSER.dat. This file was found by going to the mounted hard disk and then USERS \Sundas (or hostname). After copying the file onto the desktop as an additional precaution against tampering, the CAINE GUI was used to select Menu > Forensic Tools > Fred. The NTUSER file is opened by going to File > Open Registry Hive. When the file opens, the analyzer will see the nodes within the file, keys and hex viewer. While this is consistent with our tier 3 analysis objectives, we also want to get translated data by generating a report. If we go to Reports > Generate Report, Figure 5 shows the available reports that can be generated and the information each file provides. The NTUSER gives information on auto runs, launched applications, recent documents, typed URLS, Windows 7 searched keywords, Windows 7 typed paths and Windows live accounts.

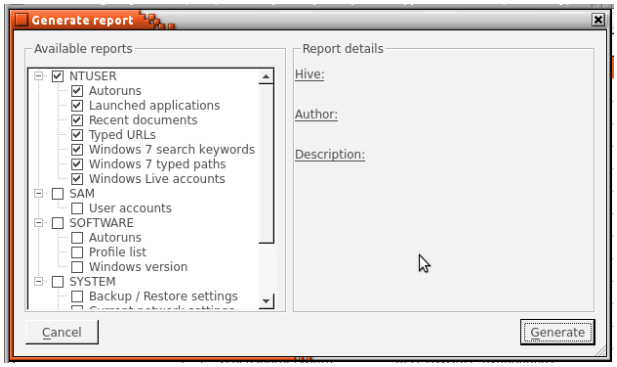

Figure 5 – report generation screen shot

The generate button will create reports from the NTUSER hive file. If a user even deletes their history, analyzers can use this scan to determine which applications were used and when they were last used. The NTUSER report shows all of the recent documents that have been opened along with typed URLs and typed keywords. If there are unknown URLs or files that are being accessed on the system unknowingly, then the NTUSER file will provide pertinent information.

The next file to be analyzed is a SAM file. SAM files give information related to user accounts on the system. Once the SAM file is opened, the nodes, keys, and hex viewer can be seen. To get translated information, the user should go to Reports > Generate. After clicking the generate button, the results in figure 6 will be shown. Other information is also given such as the last time the user has been logged in, account expiry, failed logins, and even password hints.

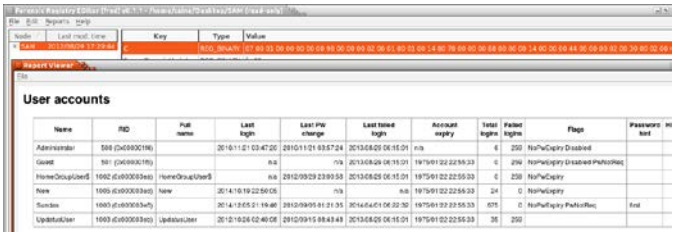

Figure 6 – Translated data screen for SAM file

The reports also show the last time the password has been changed. The information given by SAM can help computer forensic analyzers determine whether someone else is trying to physically log into user accounts. If an individual does not use their computer during a certain time, they can determine from using FRED that someone is logging onto their username at odd hours. From there, further research can be done to determine which files have been accessed. The next hive file used is SOFTWARE, which contains data for all of the different applications that are used on the operating system. Generating a report using SOFTWARE requires the same steps described previously. Figure 7 shows that a report of SOFTWARE would generate information about autoruns, profile list, and windows version (including install date, path, product ID, and name of the registered owner). The SYSTEM reports give information on backup/restore settings as well as current network settings. Aside from service and time log information, this report notes which USB storage devices have been mounted on the system. The report shows that each type of backup or restore setting has its own directories and files listed within a table. This is very beneficial in case a computer forensic analyzer wants to make sure that the backup data isn't being sent somewhere incorrect.

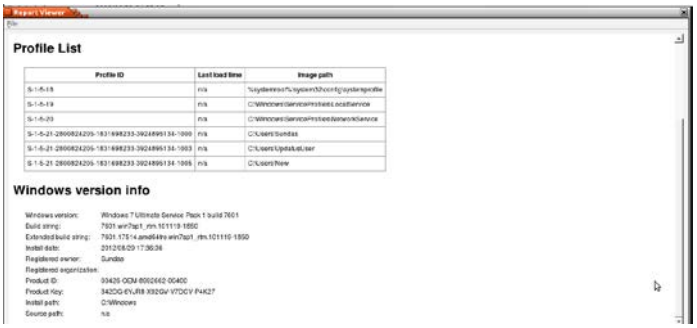

Figure 7 – Sample report screen

Autopsy will display the current network settings, showing information on each of the five adapters connected to the device, including IP address and subnet mask. The SYSTEM file shows every USB drive ever connected to the computer. The name of the storage device, vendor name, unique ID, class and mount point are given. This type of information is important because some users may be secretly trying to steal information by connecting a USB and copying files.

For the fourth tier of our forensic architecture, CAINE, the host operating system of Photorec, will be running behind the scenes and scanning a Windows host. First the CAINE ISO must be downloaded from its open source site (caine.org) and burned into a DVD or USB media (USB media must be made bootable). Since CAINE is being run live from a CD data cannot be written to it. Therefore, an external hard drive must be connected which is where the recovered files are stored. There are two alternatives; first, Guymager can be used to create an image (anything to be read and analyzed by Photorec). The other alternative is to scan the host and have it save all recovered files to the external drive. If using a virtual machine, the files can simply be stored on the host itself. Photorec will first ask to select the source, the file format of the host, and then the destination to save the recovered file. Photorec works by searching data clusters so the speed of this process may vary. The recovered files will most likely be compressed or zipped if they are large, but if they are small enough they will be placed in a folder.

V. CONCLUSIONS

Since CAINE is a fairly recent computer forensics resource, there is a limited amount of information regarding the tools provided. By experimenting with these tools, a documented process was developed which maps to the proposed theoretical four tier model of forensic analysis. One of our goals was to recommend a preferred order for applying the Guymager, Autopsy, Fred and Photorec tools. As a tier 1 tool, Guymager creates images of disks for forensic analysis, At tire 2, Autopsy analyzes the images and allows data recovery, At tier 3, FRED focuses on scanning and editing registry hives, while at tier 4 Photorec recovers lost or missing data. This combination affords some advantages. For example, unless the user already has an imaged disk, Autopsy as a stand-alone tool requires the extra step of creating an image before scanning for results. Guymager performs this function as an integral part of CAINE. While Autopsy provides useful results, it requires the overhead of creating a case and adding a host before attempting to recover the contents of a disk. By contrast, Photorec directly scans the disk for file recovery and may detect some files missed by Autopsy.

In conducting forensics research with the CAINE integrated toolkit, our main obstacle was gaining access to the right type of file for each tool to analyze, since all four tools are very particular about their input file type. Although the CAINE operating system runs within a VMware workstation, an error would occur when the image/file to scan was chosen, even if the file format was correct. Care must be taken in following the mounting directions. It is not recommended to use CAINE on the host operating system, which can result in the actual data being corrupted. Further, there were some issues with creating bootable media by writing the ISO to a USB stick. These issues and other tools within CAINE will be explored as part of our ongoing future research.

REFERENCES

[1] Cisco 2015 annual security report, http://www.cisco.com/web/offers/lp/2015-annual-security-report/index.html (last accessed June 25, 2015)

[2] N. Bassetti *CAINE Live USB/DVD. ForensicWiki* (last accessed Aug. 20, 2015)http://www.forensicswiki.org/wiki/CAINE_Live_CD

[3] B. Carrier, "Defining digital forensic examination and analysis tools using abstraction layers", Int. Journal of Digital Evidence, Vol. 1, no. 4 (Winter 2003) (last accessed Aug. 20, 2015) http://digital4nzics.com/Student%20Library/Defining%20Digital%20Forensic%20Examination%20and%20Analysis%20Tools%20Using%20Abstraction%20Layers.pdf

[4] B. Nikkel, "The role of digital forensic with a corporate organisation", Proc. IBSA Conf. Vienna, Austria (May 2006) http://digitalforensics.ch/nikkel06a.pdf (last accessed Aug. 20, 2015)

[5] S. Perumal “Digital forensic model based on Malaysian investigation process”, Vol. 9 no. 8 (2009) (last accessed Aug. 20, 2015): http://paper.ijcsns.org/07_book/200908/20080805.pdf

[6] J. Ashcroft (2001) Electronic Crime Scene Investigation: A guide for first responders: https://www.ncjrs.gov/pdffiles1/nij/187736.pdf (last access Aug. 20, 2015)

[7] G> Palmer, “A road map to digital forensic research”, 1st Digital Forensics Research Workshop Utica, NY (Aug. 7-8, 2001): http://www.dfrws.org/2001/dfrws-rm-final.pdf (last accessed Au 20, 2015)

[8] M. Reith, .C. Carr. And G. Gunsch, ”An examination of digital forensic model”. Int. Journal of Digital Evidence, Vol. 1, no. 3 (Fall 2002) http://digital4nzics.com/Student%20Library/An%20Examination%20of%20Digital%20Forensic%20Models.pdf .

[9] A. Agawal, M. Gupta, S. Gupta, and R. Gupta, “Systematic digital forensic investigation model” Int. Journal Computer Science and Security Vol. 5 no. 1 (2011)

[10] V. Baryamureeba and F. Tushabe, “The enhanced digital investigation process”, Proc. 4th annual Digital Forensics Research Workshop, Baltimore, MD (Aug. 11-13, 2004)

[11] S. Ciardhuain, “An extended model of cybercrime investigation” Int. Journal of Digital Evidence Vol. e no. 1 (Summer 2004) https://www.utica.edu/academic/institutes/ecii/publications/articles/A0B70121-FD6C-3DBA-0EA5C3E93CC575FA.pdf

[12] I. Ademu, C. Imafidon, and D. Preston, “A new approach to digital forensic models for digital investigation”, Int. Journal of Advanced Computer Science and Applications, vol 1, no 12, p. 1-4 (2011)

[13] P. Maglio, C. Kieliszewski, and J. Spohere, editors, Handbook of Service Science, Springer-Verlag (2010)

[14] G. Voncken, "Ubuntu Manuals." *Ubuntu Mainpage: Guymager*. http://manpages.ubuntu.com/manpages/precise/man1/guymager.1.html .

[15] "WindowsRegistry." *ForensicsWiki*. (last accessed Aug. 20, 2015) http://forensicswiki.org/wiki/Windows_Registry