Finding File Fragments in the Cloud

Dirk Ras, Martin Olivier

Chapter 12

FINDING FILE FRAGMENTS IN THE CLOUD

Dirk Ras and Martin Olivier

Abstract As the use – and abuse – of cloud computing increases, it becomes necessary to conduct forensic analyses of cloud computing systems. This paper evaluates the feasibility of performing a digital forensic investigation on a cloud computing system. Specifically, experiments were conducted on the Nimbula on-site cloud operating system to determine if meaningful information can be extracted from a cloud system. The experiments involved planting known, unique files in a cloud computing infrastructure, and subsequently performing forensic captures of the virtual machine image that executes in the cloud. The results demonstrate that it is possible to extract key information about a cloud system and, in certain cases, even re-start a virtual machine.

Keywords: Cloud forensics, evidence recovery, file fragments

1. Introduction

With the rapid and near universal penetration of computers into society, the incidence of computer crime has grown accordingly [25, 28]. Computers have evolved from room-sized mainframes and desktop machines with limited storage capacity to devices such as cellular telephones and tablet devices with high capacity flash memory storage [14, 24]. Meanwhile, cloud computing is becoming extremely popular because it provides outsourced storage and computing solutions at a low cost. Indeed, infrastructure as a service (IaaS) is now the fastest growing paradigm of cloud computing [32].

Cloud computing allows for the rapid provisioning of computer infrastructure – servers, applications, storage and networking. This is accomplished by creating a pool of resources from which a user can provision the desired system using virtualization technology; the resources

can just as easily be released back into the pool for other users to provision. Cloud computing thus allows multiple users to engage the same underlying resources while keeping their information separate [9, 15, 16, 29].

Cloud computing poses several problems with regard to digital forensic investigations. As with non-cloud systems, data is not immediately erased when a resource is released, but is instead marked for overwriting. However, because a cloud is structured by clustering computers and abstracting the cluster using a single operating system, it is difficult to identify and forensically analyze the specific machine that potentially contains the data. Additionally, because numerous users potentially use cloud resources simultaneously, it may be not feasible to take down even portions of the cloud to conduct forensic investigations. Indeed, digital forensics of cloud computing systems is a topic that has yet to be explored by researchers [15, 25, 29].

This paper evaluates the feasibility of performing a digital forensic investigation on a cloud system when the location of the drive that contains the information of interest is known. Experiments are performed on Nimbula, a popular on-site cloud operating system. The experimental results demonstrate that it is possible to extract key information about a cloud system and, in certain cases, even re-start a virtual machine.

2. Digital Forensics

Digital forensics is a recognized scientific and forensic process used in investigations involving electronic evidence [3, 31]. This paper focuses on the analysis phase of the forensic process, where it is assumed that the preceding phases of locating and acquiring digital evidence have already been completed [8, 12].

The digital forensic process is the process of collecting, identifying, extracting, documenting and interpreting computer data [17]. A key difference between digital forensics and traditional forensics such as forensic pathology is that digital forensics often requires more flexibility when encountering something unusual. Nevertheless, this does not mean that digital forensics should be treated any differently from forensic pathology because both have well-defined processes that must be followed. Since the details of the digital forensic phases change often, they must be documented thoroughly. The specific digital forensic phases vary according to the author (see, e.g., [12, 17]). In this paper, we consider three simplified phases:

- **Acquisition:** An exact image of the digital evidence is made from a storage device using live or dead forensic techniques [17]. Note

that the acquisition phase has a direct impact on the captured data and how it can be analyzed.

- **Analysis:** The captured image is analyzed to find data relevant to the forensic investigation [17].

- **Reporting:** The digital evidence and the results of the analysis are presented in a formal report, potentially for use in legal proceedings [8, 17].

The acquisition of digital evidence employs live or dead forensic techniques. A live forensic technique is performed on a running system [1]. Interacting with a running system runs the risk of changing the data on the system. The principal advantage of a live forensic technique is that it allows the state of the system and its behavior to be captured. A live forensic technique can be used to capture evidence from a target drive in a cloud infrastructure.

A dead forensic technique creates an image of a target computer after it is shut down. The benefit is that the process of evidence capture can be tightly controlled to ensure that no data is lost or modified [8, 17]. The disadvantage is that the current state of the target computer is not captured, and data pertaining to executing processes may be lost. A dead forensic technique can be used to locate files in a cloud computing system.

3. Cloud Computing

According to the National Institute of Standards and Technology [22], cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. Cloud computing is the convergence of multiple computing technologies to form a new technology that removes the traditional limitations of a strict policy driven information technology infrastructure [20]. Two of the core computing technologies are virtualization and clustering [15].

3.1 Virtualization

Virtualization is a core technology underlying a cloud computing infrastructure [11, 20, 23]. It enables users to access scalable computer systems on demand [30]. Virtualization works by abstracting computing resources from their physical counterparts [15] into a resource pool

from which computing resources can be drawn by users. A virtual machine can be provisioned by a user as long as the resources required for the virtual machine do not exceed those of the physical host machine. This feature allows multiple virtual machines to run on a single host [4, 19].

3.2 Clustering

The clustering of resources occurs when two or more computing systems work together to perform some function [21]. Clustering provides a scalable solution with flexibility in terms of computing power, redundancy and availability. Cloud computing relies on large amounts of resources; clustering is an attractive proposition because it allows many redundant nodes to form a resource pool.

Clusters may be categorized as high availability, high performance and horizontal scaling [21]. Horizontal scaling is of special interest in this work. This type of cluster provides a set of resources that is accessible via a single interface, and these resources can grow or shrink as required over time [21].

3.3 Infrastructure as a Service

Infrastructure (hardware) as a service can be realized by combining virtualization and clustering. It provides the capability to deliver some form of hardware-based technology (e.g., storage, data center space or servers) as a service [16].

Typically, in a cloud setting, a user would provision a certain amount of resources in the form of a virtual machine. This virtual machine appears to the user as a normal computer because the underlying infrastructure is abstracted from the resource pool. The virtual machine can then be used for the task at hand. When the virtual machine is no longer required, it is simply deleted by the user and the resources returned to the resource pool from where they can be reallocated [16].

Clustering nodes to form a resource pool is an attractive option to service providers because it gives the benefits of horizontal scaling [16, 20]. A cluster of nodes is abstracted to a single resource pool from which users can provision virtual machines.

4. Forensic Investigation

Experiments were conducted to determine the feasibility of performing a digital forensic investigation on a cloud computing system. The experimental objectives were to: (i) find a reference file on a hard drive located within a cloud infrastructure; and (ii) find a virtual machine

image that executed in a cloud infrastructure and to re-instantiate it if possible.

4.1 Experimental Setup

This section describes the experimental setup, including the files, nodes, instances and configuration.

4.1.1 Files. Three reference files, each with a known string that was unlikely to occur on the drive, were created. The files were: (i) a small text file with a unique known string of characters with a size of a few bytes; (ii) a medium text file with a unique known string of characters repeated 50 times to yield a size of approximately 1 MB; and (iii) a large text file with a unique known string repeated 100,000 times to yield a size of approximately 1.2 GB.

The files served as "contrasting agents," a term taken from x-ray imaging, where a contrasting agent such as iodine or barium is used to enhance the visibility of structures in the body. MD5 digests were computed for the files and kept as a reference for comparison if the files were found.

4.1.2 Nodes. Three nodes were used in the experiments. The nodes were set up in a cluster configuration of desktop computers connected via a local area network using TCP/IP. The nodes were installed with the standard Nimbula operating system according to the user guide [26].

4.1.3 Instance. An instance corresponds to a virtual machine with an installed operating system running on the cloud infrastructure [26]. The experiments used a single instance with Ubuntu Linux version 10.04 LTS installed. The instance served as a gateway to the cloud, allowing the files to be stored on the cloud. The instance ran on top of the cloud operating system as a virtual machine as shown in Figure 1.

4.1.4 Configuration. The infrastructure nodes, corresponing to the physical computers described in Table 1, were installed with the Nimbula cloud operating system. This was done via a network installation. After the nodes were installed, the cloud became active and could be accessed externally by connecting to the Nimbula Director using the specified IP address in a web browser [26].

For each of the file configuration scenarios, the reference files were deployed onto the cloud via the web interface from the external device.

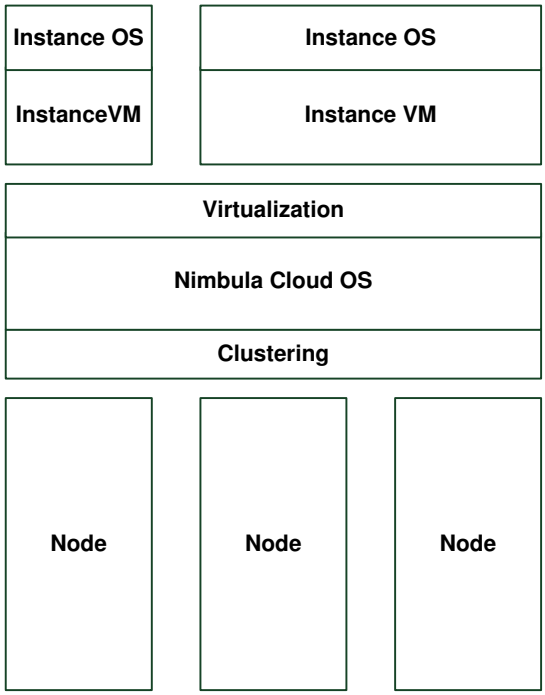

Figure 1. Cloud infrastructure.

The cloud was then taken offline (or kept running) using the following procedures [26]:

- **Controlled Shutdown:** The instance was stopped and the node was shut down manually.
- **Uncontrolled Shutdown:** The connection to the main power supply was severed.
- **No Shutdown:** The instance was not shut down and a live capture was made of the running image over a network.

After the cloud was taken offline, the hard disk drives were removed from the nodes in the configuration. A copy was made from a node drive to the external analysis drive via a write blocker. In the case of the network capture, the image was directly cast onto the external drive using `dd`. The procedure described above was then applied to the captured drive image.

4.2 Controlled Shutdown

The controlled shutdown procedure specified in the operating system user guide [26] was performed. First, the instance virtual machine was taken down by shutting down the instance operating system. Next, the node was shut down from the console using the Unix `halt` command. The machine went through the shutdown procedure and eventually powered down. After the node was shut down, the power plug connected to the wall socket was removed. The drive was then removed and the drive image captured. Finally, the drive image was analyzed.

4.3 Uncontrolled Shutdown

The uncontrolled shutdown procedure involved the more traditional method of severing the power connection to the target machine [17, 27, 31]. Since a standard personal computer was used as the node, the power cord was pulled out of the power supply unit of the machine. As expected, the machine immediately shut down. The hard drive was removed and imaged. Finally, the drive image was analyzed.

4.4 Network Capture

In this case, the node was not powered down and was allowed to continue to operate. A secure shell tunnel (`ssh`) [5] was opened from the external machine to the node. Next, `dd` was used to create an image of the node hard drive on the analysis drive, which was connected to the external machine used as the target for the output of `dd`. The capture was done over a standard TCP/IP local area network. A write blocker could not be used during the capture operation because it would have prevented the image from being made to the analysis drive.

4.5 Hardware

The hardware used in the experiments included standard desktop personal computers connected with a standard 100 Mbps TCP/IP network using a 100 Mbps switch with standard CAT5 cable. The cloud computing system was isolated from other networks to avoid unwanted distribution of the operating system by accidental network installations. Table 1 lists the hardware used in the cloud infrastructure.

A laptop computer, which was external to the cloud infrastructure, was used to access the cloud via a web interface. This machine was also used for forensic analysis. Table 2 lists the hardware corresponding to this analysis machine.

Table 1. Cloud infrastructure hardware.

Device	Specification
CPU	Intel Core i5 750 2.6 GHz
RAM	2 GB DDR3 1,600 MHz
Hard Drive	250 GB
Motherboard	IBM
Network Adapter	Intel 82577LM

Table 2. External device hardware.

Device	Specification
CPU	Intel Core i5 750 2.6 GHz
RAM	10 GB DDR3 1,600 MHz
Hard Drive	1 TB
Motherboard	IBM
Network Adapter	Intel 82577LM
External Drive	2 TB
Write Blocker	

4.6 Software

This section provides details of the operating system and software systems used in the experiments.

4.6.1 Operating System. The Nimbula cloud operating system was used for cloud management. The operating system offers an on-site solution for a cloud infrastructure, i.e., a cloud system that is owned and operated by the user or user's organization. This on-site solution differs from solutions such as Windows Azure [10], where the cloud infrastructure is owned and operated by an external entity. Because it would have been impossible to obtain physical access to a commercial cloud infrastructure for the experiments, the only option was to use this smaller in-house configuration. The configuration also gave complete control over the parameters governing the operation of the cloud, namely the number and configuration of nodes, virtual instances running on the nodes, and distributions of the instances across the nodes.

The virtual machine instance used Ubuntu Linux 10.04 LTS primarily because it provides a utility that simplifies the creation of operating system images.

The forensic analysis machine was installed with the Squeeze version of the Debian Linux distribution. This version was the latest stable version of the operating system at the time of the experiments. The

system was installed via a minimal network install to ensure that only core packages were available. Note that all the operating systems used in the experiments were based on the Debian Linux distribution and, as such, had very similar file system structures. Also, 64-bit versions of the operating systems were used in all the installations.

4.6.2 Software. The key programs used in the experiments were: (i) `dd` for creating images from raw data by performing low-level copy operations; (ii) Sleuth Kit [7] for investigating volume and file system data, and performing string searches on the captured images; (iii) Autopsy [6] as a graphical front-end for Sleuth Kit; (iv) `gzip` for uncompressing archive files; (v) Universal File Unpacking Utility, a Perl script available under the Debian and Ubuntu repositories, for uncompressing and unpacking file archive types; (v) Kernel Based Virtual Machine [18] as a virtualization solution for Linux; and (vi) `md5sum`, a Linux package for computing 128-bit MD5 hash values.

5. Experimental Results

This section presents the results of the experiments involving drive analysis and drive structure identification.

5.1 Drive Analysis

Upon analyzing the drive, it was discovered that only the node operating system could be accessed. The drive was mounted via the Unix `mount` command using the standard ext3 file system. When observing the file system, a base install of the node operating system was detected. This appeared to be a standard small footprint network installation of the Debian operating system. However, the amount of data copied from the original drive to the image did not match – it was far greater than the data visible on the drive (the observed size of the drive was approximately 400 MB while the original instance image was approximately 6 GB). From this, it was clear that the drive was partitioned in some manner. After scanning the disk for additional partitions using the `fdisk -l` command, it was revealed that the drive contained a secondary physical partition. This partition was also too small to contain the instance image.

After mounting the new partition, analysis revealed the connection interface of the node operating system to the instance operating system with symbolic links between them. These symbolic links appeared to point to a logical partition. The `lvscan` command was then used to scan the drive for logical partitions that potentially contained the copied

data. The scan revealed a logical partition on the drive that was not mounted. After creating the mounting point for the logical partition, the logical partition was mounted to the created point. The mounted logical partition was then scanned for volume groups using the `vgscan` command. This operation revealed a volume group containing five logical volumes. Mounting points for the volumes were created and the volumes were mounted.

Four of the volume groups contained only system data (required for running and managing the system), but not any user data from the instance. After performing a file search for all files larger than 3 GB, a compressed archive was found that was approximately 6 GB in size. This archive was decompressed using the `unp` package. The decompressed archive contained what appeared to be the original instance that was deployed to the cloud infrastructure. The recovered image file and the original deployed file were then successfully matched by their MD5 hash values. From here on, the methods for capturing the drive image became important because they impacted the success of the experiments.

5.1.1 Controlled Shutdown. Attempts to mount the image using the standard `mount` command failed. Attempts at finding the file system type also failed. After a brute force attempt to test every type of file system supported by `mount` also failed, it was decided that a new approach was needed.

Since the Nimbula operating system runs the deployed image files as virtual machines, it seemed reasonable that the image should be able to run as a virtual machine. As the Kernel Based Virtual Machine (KVM) [18] is often used as a virtual machine platform, it was employed in an attempt to run the image file. This operation was successful and the virtual machine booted. The virtual machine presented the standard Ubuntu Linux 10.04 LTS user interface and requested a password as it did when it was deployed to the cloud infrastructure. After the set password was entered, it appeared that the virtual machine deployed to the cloud infrastructure was re-instantiated.

With the virtual machine running, it was possible to search for the deployed reference files. The first attempt was made by inspecting the locations where the files were deployed. The files were not found in these locations. Next, a full search of the file system of the instance was conducted using the `find` command to search for the files by name; following this, the `grep` command was used to search for the unique identifying string in the reference files. In both cases, the reference files were not found. As the searches within the instance were unsuccessful, the virtual machine was shut down. After the shutdown was complete,

Sleuth Kit [7] was used to search the entire drive for the files. This was done by supplying Sleuth Kit with the identifying strings of the reference files. Some of the files were found, but they were highly fragmented. The fragmentation could be seen from the line numbers appended to the front of the reference string. From the line numbers, it is apparent that many of the reference strings were missing. Subsequent searches revealed no additional reference strings from the reference files.

5.1.2 Uncontrolled Shutdown. As in the case of the controlled shutdown experiment, attempts were made to mount the captured image using the `mount` command. Attempts to find the file system type also proved to be unsuccessful. The process of mounting the image by brute forcing all the file system types was attempted once again, but the mount procedure was unsuccessful as in the case of the controlled shutdown test.

KVM was used to successfully re-instantiate the image and subsequently boot up the virtual machine. The re-instantiated image was examined to find the reference files in their respective locations. However, the files were missing at all the locations where they were deployed. The `find` command was then used to search the entire file system for the files by name, but with no success. Next, the entire file system was searched with the `grep` command, this time with the reference string as the search parameter. However, this search was also unsuccessful.

The virtual machine was then shut down and Sleuth Kit was used to search the entire drive for the reference string. Multiple instances of the reference string were found, including the fully intact small and medium reference files. The MD5 hash values of the two files matched the hash values of the original files. The large reference file remained fragmented, but many of the fragments were found; in many cases, large blocks of consecutive line numbers were found.

5.1.3 Network Capture. As with the controlled and uncontrolled shutdown experiments, attempts were made to mount the captured image. Following the set procedure, attempts were made to mount the image using the `mount` command, but, as before, this operation failed. An attempt was again made to mount the image via a brute force method using all the supported file systems, but this was unsuccessful. An attempt to run the captured image with KVM was also unsuccessful.

With no method available to access the captured image, Sleuth Kit was used for further analysis. The entire drive was searched for the reference strings, and the three reference files were found in their entirety.

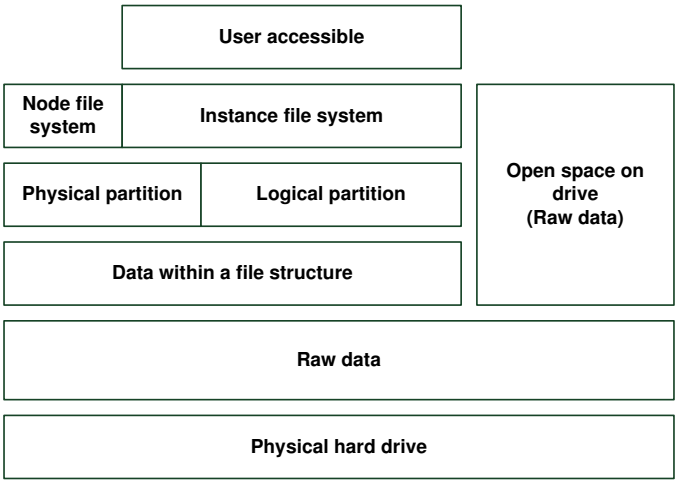

Figure 2. Data distribution on a node.

The files were in sequential blocks with no gaps or fragmentation. The MD5 hash values of the files matched those of the original files.

5.2 Drive Structure Identification

Analysis of the drive image revealed the structure shown in Figure 2. The data on the drive can be viewed as being in multiple layers. At the lowest level is the physical drive on which raw data is stored. The next level of abstraction is the raw data. This data is distributed according to the file system of the node operating system because it controls the physical drive. The raw data is divided into open or free space on the drive and the data that resides in a file structure. The open space spans levels 3 to 5 because this space is used as required by the node operating system or the cloud instance operating system.

The data in a file structure is split into a physical partition and a logical partition at level 4. The physical partition contains the file system of the physical node, which ensures the operation of the node. A small part of the partition is also allocated to the instance file system to enable it to interact with the node file system. Both the node and instance file systems are at level 6. The logical partition contains the majority of the instance file system, which enables the instance file system to grow dynamically as required. Finally, the user accessible area at level 6 corresponds to the instance file system. This is expected because the user can access the instance operating system via a network interface and thus have access to the entire instance file system; the user does

not have access to the node operating system or its file system. The search revealed some of the reference strings, but the files were highly fragmented.

5.3 Discussion

The experimental results demonstrate that it is possible to apply digital forensic methods to a cloud system. However, each of the methods used to obtain a forensic image has its own effect on the captured image.

As expected, the Nimbula cloud operating system implements the notion of a disposable instance. In cases where the instances could be re-instantiated, the files deployed to these instances were missing. The likely reasoning is that, in the event that an instance fails, it should be possible to immediately create a new instance from the original deployed image. Thus, the symbolic links to the files are lost and cannot be recovered upon restart. In terms of the original image itself, all the configurations made before deployment to the cloud remain intact. Thus, should programs be edited in a specific way or settings be performed (e.g., a database connection), the edits and settings would persist and would be available when the captured image is re-instantiated.

File fragmentation appears to be closely related to the method used for image capture. In the case of a controlled shutdown, the files are highly fragmented. This could be the result of the node operating system detecting the instance going down, at which point, it attempts to free the space in the logical partition. This is done during the period starting from when the instance was shut down to when the node itself was shut down. It also stands to reason that this clean up procedure might be part of the shutdown procedure itself. In both cases, the files deployed to the cloud via the instance were highly fragmented and incomplete.

In the case of an uncontrolled shutdown, the files found were found to be much less fragmented. Only the large file showed some fragmentation with large portions of the file containing consecutive blocks. The small and medium files were recovered in their entirety. This reinforces the belief that some type of clean up procedure occurs during shutdown.

No file fragmentation was observed in the case of the network capture, and the files were recovered in their entirety. Again, it appears that, since the system was live and the files had their symbolic links to the instance operating system intact, the files could be captured without fragmentation.

As mentioned above, the re-instantiation of an image could be useful to a forensic investigator. In particular, valuable information is available because all the configurations made to an image prior to deployment

Table 3. Summary of results.

Method	Data Recoverable	Re-Instantiated
Controlled	Partial	Yes
Uncontrolled	Partial	Yes
Network	Yes	No

persist. Also, the fact that the connection settings to databases, network drives, servers, etc. remain intact means that is possible to reconnect to them when the image is re-instantiated. This can provide a wealth of information to the investigator about the operation and purpose of cloud instances. Conversely, programs installed and settings made to an image after deployment to the cloud system would be lost, just as files are lost because only the original deployed image is stored.

Table 3 summarizes the results of the experiments.

6. Conclusions

Cloud forensics is an important, but as yet unexplored, area of research. The experimental results with Nimbula, an on-site cloud operating system, demonstrate that it is possible to extract key information about a cloud system, especially when the cloud can be accessed directly. The fact that virtual machine instances can be relaunched after they were discovered on a captured drive is important because it means that the behavior of a cloud system can be monitored. Also, the recovery of planted files implies that forensic procedures such as file system reconstruction can be performed.

Further research is necessary to specify digital forensic procedures for the various types of cloud systems. Of course, industry cooperation would be needed to deal with massive systems such as Windows Azure [10], Google Cloud [13] and Amazon Cloud [2]. These vast, distributed clouds create major challenges to locating and isolating information of interest. Another challenge is to efficiently perform live forensics on cloud systems.

References

[1] F. Adelstein, Live forensics: Diagnosing your system without killing it first, *Communications of the ACM*, vol. 49(2), pp. 63–66, 2006.

[2] Amazon Web Services, Amazon Elastic Compute Cloud (Amazon EC2), Seattle, Washington (`aws.amazon.com/ec2`).

[3] M. Andrew, Defining a process model for forensic analysis of digital devices and storage media, *Proceedings of the Second IEEE International Workshop on Systematic Approaches to Digital Forensic Engineering*, pp. 16–30, 2007.

[4] D. Barrett, *Virtualization and Forensics: A Digital Forensic Investigator's Guide to Virtual Environments*, Syngress, Burlington, Massachusetts, 2010.

[5] D. Barrett, R. Silverman and R. Byrnes, *SSH, The Secure Shell: The Definitive Guide*, O'Reilly, Sebastopol, California, 2005.

[6] B. Carrier, Autopsy (`www.sleuthkit.org/autopsy`).

[7] B. Carrier, The Sleuth Kit (`www.sleuthkit.org/sleuthkit`).

[8] E. Casey (Ed.), *Handbook of Digital Forensics and Investigations*, Elsevier Academic Press, Burlington, Massachusetts, 2010.

[9] H. Cervone, An overview of virtual and cloud computing, *OCLC Systems and Services*, vol. 26(3), pp. 162–165, 2010.

[10] D. Chappell, Introducing the Windows Azure Platform, Technical Report, David Chappel and Associates, San Francisco, California, 2008.

[11] M. Christodorescu, R. Sailer, D. Schales, D. Sgandurra and D. Zamboni, Cloud security is not (just) virtualization security: A short paper, *Proceedings of the ACM Workshop on Cloud Computing Security*, pp. 97–102, 2009.

[12] F. Cohen, *Digital Forensic Evidence Examination*, ASP Press, Livermore, California, 2010.

[13] Google, Google Apps for Business, Mountain View, California (`www.google.com/apps/intl/en/business`).

[14] S. Gopisetty, S. Agarwala, E. Butler, D. Jadav, S. Jaquet, M. Korupolu, R. Routray, P. Sarkar, A. Singh, M. Sivan-Zimet, C. Tan, S. Uttamchandani, D. Merbach, S. Padbidri, A. Dieberger, E. Haber, E. Kandogan, C. Kieliszewski, D. Agrawal, M. Devarakonda, K. Lee, K. Magoutis, D. Verma and N. Vogl, Evolution of storage management: Transforming raw data into information, *IBM Journal of Research and Development*, vol. 52(4), pp. 341–352, 2008.

[15] K. Hess and A. Newman, *Practical Virtualization Solutions: Virtualization from the Trenches*, Prentice-Hall, Boston, Massachusetts, 2009.

[16] J. Hurwitz, R. Bloor, M. Kaufman and F. Halper, *Cloud Computing for Dummies*, Wiley, Hoboken, New Jersey, 2010.

[17] W. Kruse and J. Heiser, *Computer Forensics: Incident Response Essentials*, Addison-Wesley, Indianapolis, Indiana, 2002.

[18] KVM Admin, Kernel Based Virtual Machine (`www.linux-kvm.org/page/Main_Page`).

[19] H. Lagar-Cavilla, J. Whitney, R. Bryant, P. Patchin, M. Brudno, E. de Lara, S. Rumble, M. Satyanarayanan and A. Scannell, SnowFlock: Virtual machine cloning as a first-class cloud primitive, *ACM Transactions on Computer Systems*, vol. 29(1), pp. 2:1–2:45, 2011.

[20] T. Lillard, *Digital Forensics for Network, Internet and Cloud Computing: A Forensic Evidence Guide for Moving Targets and Data*, Syngress, Burlington, Massachusetts, 2010.

[21] E. Manoel, C. Carlane, L. Ferreira, S. Hill, D. Leitko and P. Zutenis, *Linux Clustering with CSM and GPFS*, IBM Redbooks, Armonk, New York, 2002.

[22] P. Mell and T. Grance, The NIST Definition of Cloud Computing, Recommendations of the National Institute of Standards and Technology, NIST Special Publication 800-145, National Institute of Standards and Technology, Gaithersburg, Maryland, 2011.

[23] R. Moreno-Vozmediano, R. Montero and I. Llorente, Elastic management of cluster-based services in the cloud, *Proceedings of the First Workshop on Automated Control for Datacenters and Clouds*, pp. 19–24, 2009.

[24] R. Morris and B. Truskowski, The evolution of storage systems, *IBM Systems Journal*, vol. 42(2), pp. 205–217, 2003.

[25] S. Naqvi, G. Dallons and C. Ponsard, Applying digital forensics in future Internet enterprise systems – European SME's perspective, *Proceedings of the Fifth IEEE International Workshop on Systematic Approaches to Digital Forensic Engineering*, pp. 89–93, 2010.

[26] Nimbula, Nimbula Director User Guide, Mountain View, California, 2010.

[27] M. Noblett, F. Church, M. Pollitt and L. Presley, Recovering and examining computer forensic evidence, *Forensic Science Communications*, vol. 2(4), p. 1–13, 2000.

[28] G. Pangalos, C. Ilioudis and I. Pagkalos, The importance of corporate forensic readiness in the information security framework, *Proceedings of the Nineteenth IEEE International Workshop on Enabling Technologies: Infrastructures for Collaborative Enterprises*, pp. 12–16, 2010.

[29] D. Reilly, C. Wren and T. Berry, Cloud computing: Forensic challenges for law enforcement enforcement, *Proceedings of the International Conference on Internet Technology and Secured Transactions*, pp. 1–7, 2010.

[30] B. Siddhisena, L. Warusawithana and M. Mendis, Next generation multi-tenant virtualization cloud computing platform, *Proceedings of the Thirteenth International Conference on Advanced Communication Technology*, pp. 405–410, 2011.

[31] Technical Working Group for Electronic Crime Scene Investigation, Electronic Crime Scene Investigation: A Guide for First Responders, NIJ Guide, NCJ 187736, U.S. Department of Justice, Washington, DC, 2001.

[32] M. Zhou, R. Zhang, D. Zeng and W. Qian, Services in the cloud computing era: A survey, *Proceedings of the Fourth International Universal Communication Symposium*, pp. 40–46, 2010.