Digital Forensic Readiness in the Cloud

Philip M Trenwith
Department of Computer Science
University of Pretoria
Pretoria, South Africa
Email: ptrenwith@gmail.com

Hein S Venter
Department of Computer Science
University of Pretoria
Pretoria, South Africa
Email: hventer@cs.up.ac.za

***Abstract*—The traditional digital forensic investigation process has always had a post-event driven focus. This process is perhaps too long for the cloud. This paper investigates how digital forensic readiness can be used to quicken and update the traditional digital forensic investigation process to better suit cloud computing environments. John Tans states that centralized logging is the key to efficient forensic strategies. The author proposes a model that considers centralised logging of all activities of all the participants within the cloud in preparation of an investigation. This approach will quicken the acquisition of evidential data when an investigation is required, allowing the investigator to start the analysis and examination almost immediately.**

***Index Terms*—Digital Forensics, Digital Forensic Investigation Process, Digital Forensic Readiness, Cloud Computing, Remote and Centralized Logging, Windows Event Logs, Network Time Protocol, Diffie-Hellman, AES, RSA, Cryptographic Hash Functions.**

I. Introduction

The primary objectives of this research can be summarised as follows: Shortening the DFI process by quickening the acquisition of data through the use of a remote and central log server. The secondary objectives include the synchronization of timestamps and the filtering and indexing of log evidence on the server to quickly identify where the logs came from.

Digital Forensic Science is defined by Palmer[1], as "the use of scientifically derived and proven methods toward the preservation, collection, validation, identification, analysis, interpretation, documentation and presentation of digital evidence derived from digital sources for the purpose of facilitating or furthering the reconstruction of events found to be criminal, or helping to anticipate unauthorized actions shown to be disruptive to planned operations."

Normally a DFI follows a search and seizure approach. This entails seizing suspected devices for acquisition and analysis of data. Acquisitions is normally done by making a bit-by-bit copy of the storage medium of a device. However in a cloud environment this is not a simple task to perform. The cloud holds some challenges for digital forensic investigators that is discussed by Barbara[2]. One such challenge discussed by Birk[3] includes gaining access to the physical hardware running the cloud instance as the physical location of devices is often unknown, making search and isolation of devices difficult if not impossible. The time taken to complete a Digital Forensic Investigation(DFI) has the potential to be one of the biggest challenges of the investigation due to time-line constraints and deadlines, and could determine the success of the investigation. This poses the challenge to the investigators to attempt to complete the investigation or at least part of it as fast as possible. The cloud service providers also has full control over the sources of evidence and also the companies assets, making investigations by corporate security teams difficult if not impossible. Barbara concludes that the challenge for digital forensic examiners, with regard to cloud computing environments, is to determine the who, what, when, where, how, and why of cloud-based criminal activity[2].

The cloud holds further challenges such as data and process provenance. The history of digital objects, such as who accessed a specific object and when it was accessed are stored using meta-data that is referred to as data and process provenance. This can be crucial information in a DFI, and can be a question that remains unanswered if no supporting logs are made available.

The traditional model for a DFI was not designed with the cloud in mind and therefore it is worth investigating whether this traditional model can be optimized or improved to better suit cloud computing environments. Shortening the evidence acquisition phase already improves on the problem of time which is a major concern as Tan has shown[4]. Hence the research question that this paper addresses is as follows: How can we use digital forensic readiness to help to shorten the digital forensic investigation phase in cloud computing environments?

The remainder of this paper is structured as follows. Section 2 discusses the history and background leading up to this research being undertaken. Section 3 discusses the requirements and implementation of a prototype solution. Section 4 is a discussion section and section 5 concludes with the results obtained from this work.

II. Background

The background section discuss the history of both digital forensics and cloud computing and how it came into existence. As the Internet grew over the years, the number of crimes committed using digital devices grew as well. In response to this growing number of digital crimes, digital forensics also referred to as computer forensics emerged[1].

A. *Digital Forensics*

Murphy defines Digital Forensics as the application of science to the identification, collection, analysis and examination of digital evidence, whilst preserving the integrity of the information and maintaining a strict chain of custody for the evidence[5]. The goal of computer forensics according to Yasinsac, and Manzano[6], is to provide sufficient evidence to allow the prosecutor to successfully prosecute the perpetrator.

The DFI process involves five steps or phases according to Cohen[7]. These steps are identification, collection, transportation, storage, examination and presentation. Each of these steps are discussed next.

The goal of the identification process is, to identify relevant evidence so that it can be collected and processed. Only evidence deemed to be legal and useful in building a case should be collected for analysis. This is referred to as the proportionality rule. The proportionality rule also states that only upon good cause can there be discovery of computerized data[8]. This is to protect clients from unreasonable searches.

Next the acquisition of evidence is undertaken. The collected evidence has to be transported and stored in a secure location. The integrity of the evidence as well as that of the chain of custody must be maintained at all times in order for the evidence to be admissible and acceptable. The process of examination involves Analysis, Interpretation, Attribution and Reconstruction.

Evidence can be produced by many sequences of events. When trace evidence is associated with a specific program or event, this evidence can be analysed in a set. The analysis phase will focus on identifying trace evidence associated with sets of traces.

Evidence may suggest that a certain event occurred but in fact the evidence could have been produced by malicious code. Hence it is very important during the interpretation of evidence to establish, if the user in question is actually the one who committed the crime.

This is done by identifying anchor events during the attribution phase. Anchor events are events that can draw clear lines between the physical world and the digital world. Non-repudiation is the characteristic that draws the line between physical and digital world. When it can be proved that a certain event took place and it cannot be claimed that it did not take place.

Sometimes a higher level of certainty is required than what is given from analysis of logs and records. Then an examiner will perform reconstruction. The goal is to reconstruct the sequence of events to determine the state at which the system was at the time of the incident and to verify evidence.

At the conclusion of an investigation, the evidence has to be presented to a judicial body. This can be done in the form of a report or testimony to judges or juries. Such a report should be as simple as possible to ensure that the jury are informed and can make an informed decision. Technical jargon must be avoided at all cost.

Traditionally in computer forensics, an investigator pulls the plug on the machine, after which the disk or discs are imaged for analysis at a later stage[9]. Isolating a device is often necessary in order to perform an investigation as discussed by Delport[10], however isolation becomes a challenge in the cloud. It is not necessarily possible to pull the plug in a cloud computing environment. Even identifying suspected physical computers becomes much harder when considering the virtualization of resources in the cloud. computers suspected to be involved in a crime can not simply be seized for investigation therefore the post event driven model of investigating is inadequate in cloud computing environments. Hence, traditional computer forensics fails in cloud computing.

B. *Digital Forensics in the Cloud*

Digital forensics in the cloud requires a proactive approach; being prepared for expected litigations or disputes in advance. The proactive approach taken in this research is DFR. This research will attempt to apply digital forensic readiness to a cloud instance as part of the solution to improve on the traditional approach of a DFI and address the challenge posed by isolation in the cloud.

DFR is defined by Tan[4] as the ability of an organization to maximise its potential to use digital evidence while minimizing the cost of an investigation. An organization has the ability to actively collect data that could be required during a DFI such as log files, emails, network traffic records and other digital data that could be considered potential evidence should an incident occur. Implementing digital forensic readiness in a cloud environment can be achieved by setting a standard to which all of the organisations that use the specific cloud environment have to conform to. Some researchers have already started looking at DFR such as the work of Rowlingson[11]. Rowlingson proposes ten-steps an organization should implement to comply to digital forensic readiness. Three of these steps will be addressed in the proposed solution, Step 2 - Identify available sources and different types of potential evidence. Step 4 - Establish a capability for securely gathering legally admissible evidence to meet the requirement. Step 5 - Establish a policy for secure storage and handling of potential evidence. These steps are applicable and can be implemented in a cloud environment.

Baggili[12] identifies a few challenges that digital forensic investigator are faced with in a cloud environment, we will address two of these challenges in our proposed solution: 1. Jurisdictional issues - this relates to who has the jurisdiction to investigate an international incident in a cloud environment 2. Decreased control over data and decreased access to forensic data from a client side.

Supporting logs is important in the cloud as data and process provenance could be crucial in a DFI[13]. This data provides the investigators with meta-data regarding the history of digital objects. Tan[4] states that centralized logging is the key to efficient forensic strategies. Logging data to a system other than itself better maintains the integrity of the data. This is exactly how we propose to do logging in our solution to preserve the integrity of the stored logs which could be used as potential evidence and quicken the acquisition phase by allowing easier access to the data.

The reviewed literature in this section has shown that conducting a DFI in the cloud can be a very complex task and comes with a host of challenges. The process can be simplified by the application of digital forensic readiness in the cloud. However to apply DFR successfully to cloud environments require the collection of best evidence in such a manner that the integrity of the evidence is maintained throughout the process of collecting, transporting and storing of evidence. Based on the information provided in this section the author can compile a list of requirements for the proposed model. This will be looked at in the next section.

III. Model for Digital Forensic Readiness in cloud computing environments

From statistics collected by StatOwl[14] the Windows Operating System is the most widely used operating system on the market at the time of this writing, accounting for 84.69%. Therefore the focus of the development of the prototype for centralized logging is solely targeted towards the Windows Operating System.

The subsections that follow is laid out as follows. Section 3.1 gives a brief overview of the requirements for the modal. Section 3.2 gives a overview of the environment in which the model will be implemented. Section 3.3 discuss the implementation.

A. *Model Requirements*

Based on the literature review we have identified the following basic requirements that a DFR on the cloud should have. Communication Channel, Encryption, Compression, Authentication of log data and proof of integrity, Authenticating the client and server, and Timestamping.

The communication channel is required to move data between the client side and the server side software. Encryption is used to secure the communication to ensure it is not tampered with during transfer. Compression is used to package all of the backup log files into a single file to make transfer easier while compressing the data. The authentication of log data and proof of integrity proves that the data has not been tampered with, authenticating the client and server is required to ensure the log data came from the correct machine and was sent to the correct server. Timestamping is done to sync the clocks of the machines to avoid confusion during the presentation phase.

The requirements of the two applications with regards to the DFI process is laid out below.

Identification - Determined pre-development
Collection - Client application
Transportation - Client and Server Application
Storage - Server Application
Examination - Independent task

B. *Proposed Model*

Figure 3.2.1 shows a brief layout of the cloud environment on which centralized logging will be applied.

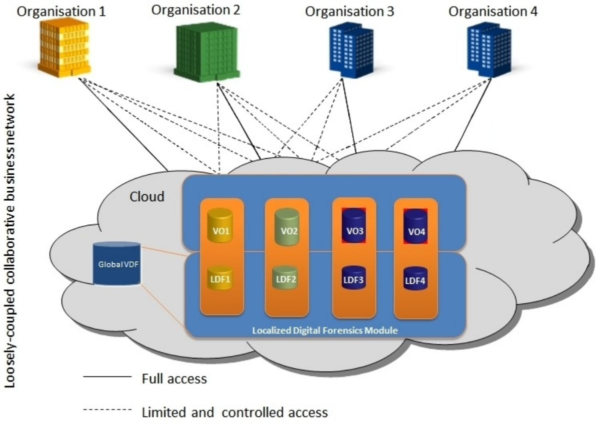


Figure 3.2.1 - Layout

The proposed solution implements remote and centralized logging, in an attempt to improve the integrity of stored evidence. It also overcomes the jurisdictional issues that investigating teams could be faced with. Each organisation has a virtual machine sharing data and application with other organisations. Each virtual machine contains a local digital forensic module, this is the client module responsible for forwarding log evidence to a central server. The Windows Event Logs of each virtual machine in the hybrid cloud is backed up using the Win32 API, compressed and forwarded to a remote and central log server, where it is indexed and stored in a digital forensic ready manner.

C. *Implementing the proposed model*

The proposed model is implemented in a proof of concept prototype. The implementation of this prototype is discussed in this section. Both the client and server applications consist of a windows service. The subsections is discussed in the order of execution that they are required in the implementation.

3.3.1 Collection of log data

Collection of event log data is accomplished by making use of the Advanced Windows 32 Base API system calls to OpenEventLog, BackupEventLog and CloseEventLog.

3.3.2 Compression

Once all the log data has been collected and the meta-data computed the data is compiled into a single archive file, this archive is then encrypted and transmitted to the server. The implementation of this model achieved a compression rate of up to 87% compressing the original logs from 27.6 Mb to 3.7 Mb.

3.3.3 Communication Channel

Because all communication occurring in the proposed model will be on a one-to-one bases between the client and the server the TCP protocol was selected as the communication channel. It is a connection orientated protocol which ensures better reliability than a connectionless connection such as UDP. This is an insecure connection therefore the data sent over the connection needs to be encrypted to maintain the confidentiality and integrity of the data.

3.3.4 Encryption

The primary purpose of encryption is to maintain the integrity and confidentiality of the data. It also ensure that the data being transmitted is not tampered with during the transmission phase. In the implementation of this prototype the use of the AES-256 encryption scheme was selected. However AES is a symmetric encryption scheme that requires the client to provide the server with the key to decrypt the datafile. To exchange this key the RSA asymmetric encryption scheme is used. In this research the AES and RSA algorithms was selected because it is the most popular schemes that guarantees confidentiality and authenticity of data over an unsecured connection[15]. The RSA algorithm has large computational overhead due to the size of its key. Therefore the large data files are encrypted using AES which is a faster algorithm[16], and only the aes-key is encrypted with the RSA algorithm. Each time a client sends data to the server, an AES key is encrypted with the servers public key and sent to the server.

3.3.5 Authentication of log data and proof of integrity

Authentication of the Windows Event Logs is accomplished by the computer-id field in the event log itself. It is thus also a requirement to ensure the integrity of the log data is maintained so that this identification can not be changed without being detected. A cryptographic hash function is to accomplish this goal as hash functions were designed specifically to protect the authenticity of information[17].

3.3.6 Authenticating the client and server

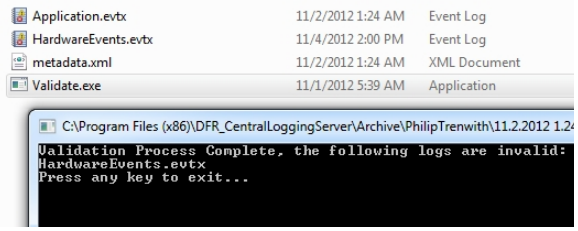
It is required that all evidence logged on the central log server be authenticated as original data. The Windows Event Logs may include a field indicating the computer name where the data came from, but what if an attacker clones a system before attempting an attack and thereafter replaces the log evidence indicating the events that took place during the attack with clean logs from the cloned system? Therefore it is required that any client sending data to the server be authenticated at the server before storing any log evidence.

To accomplish this the client and server communicates a diffie hellman shared secret. The client application signs the Diffie-Hellman shared secret using the RSA sign and verify functions. The signature is sent with the encrypted zip file to the server. The server verifies the signature against the hash of its own shared secret to authenticate the client.

3.3.7 Authentication of log data and proof of integrity

Each log file is hashed using the SHA-256 hash algorithm. The hash codes is saved in the meta-data.xml file with the names of each log. However this poses a problem, if the hash code is saved in plain-text anyone who changes the log-file can simply create a new hash of the file and overwrite the original in the meta data file. To overcome this problem the original hash of the log file is used as an encryption key to encrypt a salt value and the resulting cipher-text is then saved to the meta data file.

The Validate.exe application verifies the integrity of the log by computing the hash of the log and encrypting the same salt value. The integrity of the log is then verified by matching the corresponding cipher-text against the cipher-text provided in the meta data file. If a single byte in a log file is changed it does not go undetected.



Application.evtx	11/2/2012 1:24 AM	Event Log
HardwareEvents.evtx	11/4/2012 2:00 PM	Event Log
metadata.xml	11/2/2012 1:24 AM	XML Document
Validate.exe	11/1/2012 5:39 AM	Application

C:\Program Files (x86)\DFR_CentralLoggingServer\Archive\PhilipTrenwith\11.2.2012 1.24

```
Validation Process Complete, the following logs are invalid:
HardwareEvents.evtx
Press any key to exit...
```

Figure 3.2.2 - Validation

Figure 3.2.2 above shows the process of validating the log evidence. In this figure the program has detected that the HardwareEvents log has been changed since the collection time.

3.3.8 Timestamping

Determining the time at which events took place is quite simple, the event logs contain a field that logs the timestamp at which an event took place. This value of the logged field however is determined by the date-time of the computer, set by the user. This presents a problem, the times on all the machines may not be synchronized. To overcome the problem of out-of-sync timestamps from different systems Tan[4] suggest the Network Time Protocol, RFC-5905[18], as the most efficient protocol to achieving time synchronization on IP based systems, and we will use the same protocol in this research. There exists a policy in the windows platform disallowing the users from changing the data and time on the machine. In order for the synchronization of timestamps to be successful the policy must be implemented on the client machines.

IV. Discussion

The primary objectives of this research can be summarised as follows: Shortening the DFI process by quickening the acquisition of data through the use of a remote and central log server. And in doing so improving the integrity of the log evidence[4]. The synchronization of timestamps and the filtering and indexing of log evidence on the server has the goal of quickly identify where the logs came from and is a secondary objective.

This model was designed after conducting a literature study to identify the gaps in existing solutions and based on this study the prototype was developed. The prototype captures log evidence and transport it to a remote and central log server. This better maintains the integrity of the log evidence. It provides the investigators with archived log records once every 2 hours. This time span can easily be adjusted to suit the needs of the specific organisation.

The prototype was developed for Windows platforms only, this is a limitation on the prototype. Further more Ibrahim[19] identified the Windows Registry, Slack Space, and the Windows Event Logs as the best locations for collecting evidence for a DFI. The proposed solution only acquisitions evidence data from the Windows Event Logs.

Other limitations that exist in cloud environments that is not addressed by this model includes security, such as access control on the central server. Or protecting central log server operating system or hypervisor. This model has the sole focus of addressing the acquisition of log evidence in a networked environment with the intention of shortening the DFI process to better suit a cloud environment.

Determining the time at which events took place is quite simple, the event logs contain a field that logs the time at which an event took place. This value of the logged field however is determined by the date-time of the computer, set by the user. This presents a problem, the times on all the machines may not be synchronized. To overcome the problem of out-of-sync timestamps from different systems Tan[4] suggest the Network Time Protocol, RFC-5905[25], as the most efficient protocol to achieving time synchronization on IP based systems, and we used the same protocol in this research. There exists a policy in the windows platform disallowing the users from changing the date and time on the machine. In order for the synchronization of timestamps to be successful the policy must be implemented on the client machines.

V. Conclusion

The research question that this paper addressed is as follows: How can digital forensic readiness help to shorten the digital forensic investigation process in cloud computing environments? To address this question we proposed the design of a model that incorporates digital forensic readiness into cloud computing environments. Specifically by periodically collecting log evidence in the form of Windows Event Log records from Windows platforms and transporting these logs to a remote and central log server where they are archived according to the machine name and time of acquisition. This model shortens the digital forensic investigation process by allowing for faster acquisition of evidence should an investigation be required. This model shortens the acquisition of log evidence, however the acquisition of other forms of evidence that may be required in a DFI in the cloud is not addressed by this model.

Research that remains to be done includes investigating the development of a model that targets other platforms apart from the Windows platform. The inclusion of other types of evidence for instance taking a automated snapshot of a virtual machine during the evidence collection phase, providing investigators with not only log evidence but with data in motion in memory at the time of the snapshot. This would allow investigators the opportunity to investigate attacks in the same way as live-forensics. Further more the acquisition of evidence data from the Windows Registry and Slack Space can be investigated.

VI. Bibliography

[1] G. Palmer, "A road map for digital forensic research," First Digital Forensic Research Workshop (DFRWS), Tech. Rep. DTR - T001-01 FINAL, August 2001.

[2] J. J. Barbara, "Cloud computing: Another digital forensic challenge," *Digital Forensic Investigator News*, October 2009.

[3] D. Birk and C. Wegener, "Technical issues of forensic investigations in cloud computing environments," in *Ruhr-University Bochum, Horst Goertz Institute for IT Security Bochum, Germany*, January 2011.

[4] J. Tan, "Forensic readiness," *Cambridge, MA:@ Stake*, 2001.

[5] J. J. Murphey, "Forensic readiness'," 2007. [Online]. Available: www.dexisive.com/wp-content/.../06/Forensic-Readiness.pdf

[6] A. Yasinsac and Y. Manzano, "Policies to enhance computer and network forensics," in *IEEE Workshop on Information Assurance and Security*, June 2001.

[7] F. Cohen Ph.D, *Digital Forensic Evidence Examination - 3rd Edition*. Fred Cohen and Associates out of Livermore, 2011.

[8] S. A. Moss, "Litigation discovery cannot be optimal but could be better: The economics of improving discovery timing in a digital age," 2009. [Online]. Available: http://legalworkshop.org/2009/05/24/litigation-discovery-cannot-be-optimal-but-could-be-better-the-economics-of-improving-discovery-timing-in-a-digital-age

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

[10] W. Delport, M. Kohn, and M. S. Olivier, "Isolating a cloud instance for a digital forensic investigation," in *Proceedings of the 2011 Information Security South Africa (ISSA 2011) Conference*, Johannesburg, South-Africa, August 2011.

[11] R. Rowlingson Ph.D, "A ten step process for forensic readiness," *International Journal of Digital Evidence*, vol. 2, 2004.

[12] R. K. Baggili, I. Carthy, and J. T. Kechadi, "Survey on cloud forensics and critical criteria for cloud forensic capability: A preliminary analysis," in *University College Dublin, Zayed University*, 2011.

[13] K. Muniswamy-Reddy, P. Macko, and M. Seltzer, "Provenance for the cloud," *Proceedings of the 8th USENIX conference on File and storage technologies*, pp. 15–14, 2010.

[14] StatOwl, "Operating systems market share," 2012. [Online]. Available: http://www.statowl.com/operating_system_market_share.php

[15] A. Al Hasib and A. Haque, "A comparative study of the performance and security issues of aes and rsa cryptography," *Convergence and Hybrid Information Technology, 2008. ICCIT'08. Third International Conference on*, vol. 2, pp. 505–510, 2008.

[16] C. P. Pfleeger and S. L. Pfleeger, *Security in Computing Fourth Edition*. Prentice Hall, 2006.

[17] B. Preneel, "Cryptographic hash functions," *European Transactions on Telecommunications*, vol. 5, no. 4, pp. 431–448, 1994. [Online]. Available: http://dx.doi.org/10.1002/ett.4460050406

[18] D. Mills, U. Delaware, J. E. Martin, J. Burbank, and K. Kasch, "Network time protocol version 4: Protocol and algorithms specification," June 2010. [Online]. Available: http://tools.ietf.org/html/rfc5905

[19] N. Ibrahim, A. Al-Nemrat, H. Jahankhani, and R. Bashroush, "Sufficiency of windows event log as evidence in digital forensics," *Global Security, Safety and Sustainability & e-Democracy*, pp. 253–262, 2012.