Harvesting digital evidence from an operational cloud environment for digital forensic readiness purposes

A Mini-dissertation by

Makura Sheunesu M

(13090012)

Submitted in partial fulfilment of the requirements for the degree

Masters in Information Technology (MIT)
(Stream A- Coursework)

In the

School of Information Technology

Of the

The Faculty of Engineering, Built Environment and Information Technology

University of Pretoria

April 2019

South Africa

Supervisor: Professor H.S Venter

Table of Contents

Declaration

I declare that this dissertation is the original work of the researcher and that it has not been submitted and will not be submitted to any other university for a similar or any other degree award. Works of others used in this dissertation have been duly acknowledged in the text and included in the reference list.

Makura Sheunesu M.

Signature:

Date: 31/04/2019

Acknowledgement

Firstly, I would like to thank the Almighty God for guiding me throughout this research study. All honour and glory belongs to Him. Secondly, I thank my supervisor Prof. H.S Venter for his mentorship, support and guidance during this research study. The weekly sessions we had were insightful and he was always supportive in the sessions. I also thank Dr. Victor Kebande and Mr. Hermann Ntsamo for their assistance and expertise in the field of digital forensics. Their research work inspired this research study. Dr. Victor Kebande also contributed immensely by proof reading the write-up of this research study, and for this I am most grateful.

Furthermore, my family played a significant role during study and dissertation writing. I thank my wife for her love, patience, support and encouragement as I was doing this research. I also thank my parents for their financial assistance and encouragement towards the completion of this study. May God bless them all.

Lastly, I thank the Information and Computer Security Architectures (ICSA) research group, now known as Digital Forensic Science (DigiForS) for providing the lab, necessary material and equipment that enable me to perform the various experiments for this research study.

Abstract

An increase in organisations' use of cloud computing technologies has led to cybercriminals targeting cloud environments in order to orchestrate malicious attacks. This led to the need for proactive approaches through the use of digital forensic readiness (DFR). A prototype developed by Kebande et al. (2016) sought to provide a means to attain DFR in a cloud environment without altering the existing cloud functionality. The prototype is presented as a forensic agent that uses modified botnet functionalities in order to amass digital information in a non-malicious operation. The prototype, which was implemented in a simulated environment, is able to harvest digital data like CPU and RAM usage, and keystrokes which are then hashed and stored as information in a database. However, the prototype was never tested on an operational cloud environment, hence this research study, which sought to implement a modified version of the prototype in an operational cloud environment for the purposes of achieving DFR in the cloud. OpenStack is used to provide the operational cloud environment. The prototype is deployed and executed in cloud instances hosted on OpenStack. The experiments performed in this research study show that it is viable to attain DFR in an operational cloud platform through the use of the prototype. Further observations show that the prototype is capable of harvesting digital data from cloud instances and store digital data in a database. The prototype also prepares the operational cloud environment to be forensically prepared for digital forensic investigations to be performed without alternating the functionality of the OpenStack cloud architecture.

Abbreviations

CFR Cloud Forensic Readiness

CFRaaS Cloud Forensic Readiness as a Service

CPU Central Processing Unit

CSPs Cloud Service Providers

DDoS Distributed Denial of Service

DFRWS Digital Forensic Research Workshop

DF Digital Forensics

DFR Digital Forensic Readiness

DFI Digital Forensic Investigations

DoS Denial of Service

ECT Electronic Communications and Transactions Act

IaaS Infrastructure as a Service

ISO International Organisation for Standardisation

IEC International Electro Technical Commission

IT Information Technology

JSON JavaScript Object Notation

MYSQL “My” Structured Query Language

NIST National Institute of Standards and Technology

NMB Non-Malicious Botnet

OS Operating System

PaaS Platform as a Service

PC Personal Computer

PDE Potential Digital Evidence

PHP Hypertext Preprocessor

PoPI Protection of Personal Information Act

RICA Regulation of Communications and Provision of Communication

SaaS Software as a Service

SLAs Service Level Agreements

SQL Structured Query Language

URL Uniform Resource Locator

VM Virtual Machine

Chapter 1: Introduction

1.1 Introduction

Information Technology (IT) has recently transformed the way organisations operate by providing an effective means of executing their tasks. IT has enabled the automation of tasks across organisations and this has led to increased productivity. This use of IT across organisations has resulted in the utilisation of the following advantages: faster communication, remote access and the storage of data in digital systems. Nevertheless, it is important to note that, nearly every organisation in this modern era makes use of IT for their operations in various ways. In addition, IT systems are assisting organisations in decision making, storage of organisational records, automating organisational processes and in increasing throughput.

Such advancements of IT have led to development of cloud computing technologies. Various organisations have adopted the cloud paradigm as a model for running their business solutions. Organisations are able to obtain access to cloud applications that enhance their business operations at minimal costs. The availability of cloud-based applications at the global level has led to the reduction of costs and easy access of the same applications regardless of the geological location. This means that there is no need for the organisation to avail the same cloud-based applications per each organisational site. Thus, the continued advancement of cloud computing technology over the years has afforded organisations the opportunity to utilise cloud-based applications and services. Cloud based infrastructure usage has indeed grown and in particular public cloud infrastructure expenses have had a yearly growth rate of 17.7% (more than 200 billion dollars) in the years 2010-2015 (De Marco et al. 2014). Examples of prominent organisations that host cloud computing services include Amazon, Google and Microsoft (Marston et al. 2011).

The use of cloud computing services has unwrapped many opportunities for organisations, however, these opportunities also bring with them formidable security and privacy challenges. One of these challenges arises from large volumes of data (big data) that Cloud Service Providers (CSPs) store. As big data gets uploaded onto the cloud, questions arise with regards to the security and seclusion of the data. Further questions ask about who owns the data, who has access to the data and whether or not the data is encrypted (Popović & Hocenski, 2015).

Some organisations are wary of adopting cloud computing because they are afraid that the cloud infrastructures can be hacked, which might lead to organisational data loss, the disruption of IT systems, and a reduction in performance and availability (Sen, 2015). Organisations are more concerned with keeping the data secure whether when it is in storage media or when it is in transit. As a result, it becomes necessary that the data stored in cloud infrastructures is protected at all times. The protection ensures the confidentiality, availability and integrity of data. In addition, should there be a compromise in the confidentiality, availability and integrity of data, then an investigation should follow in order to determine the causes of that incident. This investigation can be done through digital forensics.

Digital forensics (DF) is defined as the process of using scientifically demonstrated techniques in the, “collection, preservation, analysis and presentation of digital evidence” obtained from electronic devices in order to reconstruct events that appear to be criminal in nature (DFRWS 2001, p. 16). It makes use of scientifically proven methods in conducting any type of digital investigation (Tan, 2001). DF can be used to answer a variety of questions about what would have caused the incident, when it happened and about how the incident unfolded. These questions usually arise after an incident has occurred. An incident is a threat or violation of computer security policies (Bromiley, 2016). Examples of incidents include organisational data loss or a malware intrusion. Finally, DF can provide means to prevent security problems within cloud infrastructures by identifying potential security threats and assist in the creation of solutions to the security problems.

In order to protect data in the cloud, there is need for proactive approaches. This approach entails consistently and continually monitoring the movement and storage of information within the cloud. The approach also prepares organisations to be prepared before potential security incidents happen. In the case where an incident has already happened, there will be need to investigate and conduct an analysis of evidence in order to uncover what happened or the root cause of the problem. This fact-finding mission can be done through digital forensic readiness (DFR).

Tan (2001) outlines digital forensic readiness (DFR) as the capability of a digital forensic investigation agency in boosting the usage of collected digital evidence whilst reducing the expense of a digital forensic investigation to responding to an incident. Digital forensic investigations can be a challenge for organisations due to the costs that may be involved in modifying a cloud infrastructure since reprogramming the cloud is costly and time-consuming (Kebande & Venter, 2015). Potential digital evidence (PDE) is defined as any collected digital data that might be relevant to a digital forensic investigation. This research study seeks to review on the use of DF as a proactive tool in a cloud environment to attain DFR.

This remainder of this chapter is structured as follows. The reader has been provided with the necessary introduction in Section 1.1. Section 1.2 describes the motivation to the research study, while Section 1.3 outlines the study's problem statement. Section 1.4 presents the research objectives. Section 1.5 lays out the mini-dissertation structure, and Section 1.6 concludes the chapter.

1.2 Motivation

The study is motivated by the lack of a formal structure for conducting DFR in the cloud. Kebande and Venter (2015) argue that there lacks a structured approach for conducting DFR without the need to alter the cloud infrastructure. The authors proposed a software prototype called the, "Digital Forensic Evidence Collecting System (DFECS)", which they managed to implement in a simulated environment (Kebande et al. 2016). DFECS was meant as a proof of concept that DFR can be attained in a simulated environment. This research study seeks to implement a modified version of the DFECS in an operational cloud environment. A typical cloud environment setup would be one that can be provided by a cloud operating system.

1.3 Problem Statement

The presence of security threats and attacks in the cloud infrastructures necessitates the need for a proactive approach, which ensures that a cloud environment is ready for digital forensic investigations. This approach's advantage is that it boosts the value of the PDE that can be utilised in a digital forensic investigation. Hence, it becomes necessary for organisations to employ DFR as a proactive approach to maximise the worthiness of PDE and minimize the expense of a digital forensic investigation. Hence, the main problem and sub-problems are discussed in the next two sub-sections in order layout the problem that is addressed in this mini-dissertation.

1.3.1 Main research problem

The main problem that this research study seeks to look into is the absence of a novel approach to attaining DFR in a cloud environment. Kebande et al. (2016), developed the DFECS and implemented it in a simulated environment, in order to address this problem. DFECS was however never tested in an operational cloud environment. Therefore, this research study aims to implement the DFECS in an operational cloud environment offered by a cloud operating system and thereby prove the attainability of DFR in a cloud environment.

1.3.2. Sub-problems

Various sub-problems emerge from the above-outlined main research problem. These sub-problems are explained below.

(i) How can DFECS be implemented in an operational cloud environment?

The DFECS software prototype was never tested in an operational cloud environment such as an environment provided by a cloud operating system. This research question seeks to modify and test the DFECS prototype in an operational cloud environment. The aim is to analyse if the prototype can be implemented forensically in order to harvest digital data and forensically store digital data in a database. OpenStack, an open source cloud operating system is used to provide an operational cloud environment.

(ii) What is the impact and usefulness of the collected digital data?

The research question seeks to study if the collected digital data is useful in solving security incidents that occur in the cloud environment. Scenarios are performed to simulate typical security incidences that might happen in a cloud environment.

1.4 Research Objectives

The study’s research objectives are to:

- **Review literature and the current state of DFR**- The literature review focuses on the current state of DFR and unpacks the challenges encountered by digital forensic investigators in attaining DFR.
- **Implement the prototype in an operational cloud environment-** This objective focuses on the implementation and determines whether the proposed software prototype is in a position to amass PDE in a cloud environment, specifically OpenStack, and send the data to a forensic database.
- **Analyse the potential usefulness of data collected by a modified version of DFECS**- The analysis seeks to evaluate if the collected data can be used to solve security incidents that occur within a cloud environment.

1.5 Mini-dissertation layout

The mini-dissertation consists of four parts. Part 1 consists of Chapter 1, which presents the introduction and background of the research study. It also elaborates on the problem statement and research questions addressed in this study.

Part 2 consists of Chapters 2 and 3. Here, Chapter 2 describes the background to botnets and cloud computing. It describes what botnets are and how they propagate. The chapter also explains the concept cloud computing and the cloud deployment models.

Chapter 3 focuses on digital forensics. The chapter presents an outline of the history and current state of digital forensics. It also discusses the legal aspects and challenges faced in the digital forensics fraternity.

Part 3 of the mini-dissertation focuses on the prototype and consists of Chapters 4 and Chapter 5. Chapter 4 presents an overview of the prototype. It also outlines how the prototype functions and collects digital evidence from an operational cloud environment. Chapter 5 explains how the prototype was implemented in an operational cloud environment and expounds on the various scenarios that were performed.

Part 4 is the final part of the mini-dissertation and it consists of Chapters 6 and 7. Chapter 6 presents a critical evaluation of the research study.

Chapter 7 discusses the extent to which the problem statement and research questions have been addressed in the research study. It also concludes and outlines future work for this research study.

Figure 1.1 on page 15 presents a diagrammatic layout of the mini- dissertation.

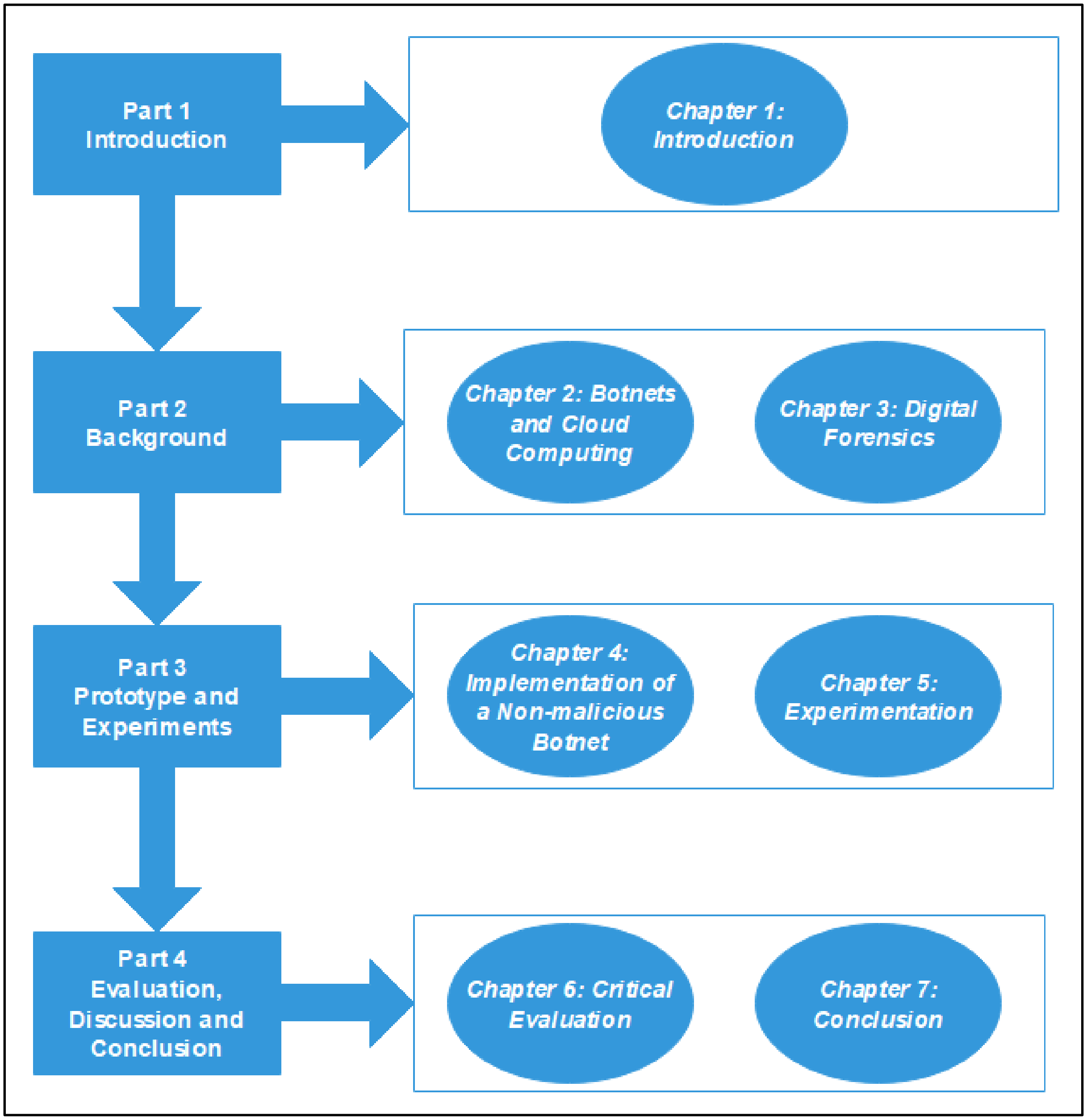

Part 1
Introduction
Chapter 1: Introduction
Part 2
Background
Chapter 2: Botnets and Cloud Computing
Chapter 3: Digital Forensics
Part 3
Prototype and Experiments
Chapter 4: Implementation of a Non-malicious Botnet
Chapter 5: Experimentation
Part 4
Evaluation, Discussion and Conclusion
Chapter 6: Critical Evaluation
Chapter 7: Conclusion

Figure 1.1. Dissertation layout

1.6 Conclusion

This chapter introduced the research study. It presented a brief overview of the IT fraternity and digital forensics. The chapter also explained motivation for this research study and the problem statement addressed in this study. The chapter further explained the study's main research and sub problems and the research objectives that the research study sought to attain. Finally, the chapter concluded with a layout of the dissertation.

The next chapter focuses on the background information on botnets and cloud computing.

Chapter 2: Botnets and Cloud Computing

2.1 Introduction

Chapter 2 presents the background on botnets and cloud computing. A botnet is a piece of software that can be used to infect a device and automate tasks over the Internet (Xie et al. 2008). Some of the tasks that botnets perform include capturing keyboard keys entered on the keyboard, executing denial of service attacks and infecting other machines with malware. Botnets can also infect devices used in cloud computing since they are connected to the Internet. Hence, the research study explores botnets as they are capable of amassing digital data in a cloud environment.

Cloud computing remains an essential topic globally. Many organisations are using cloud computing technologies due to the various advantages, such as a reduction in IT costs, scalability and business continuity, which cloud computing offers (Avram, 2014). In addition, Meyer and Stander (2015) note that access to cloud applications from cloud service providers (CSPs) enhance organisations' business operations at minimal costs. Thus, this chapter introduces cloud computing and considers some of its advantages.

The remainder of Chapter 2 is structured as follows: Section 2.2 presents a synopsis on botnets. Section 2.3 introduces cloud computing, explains the cloud deployment models and the cloud operating systems. The chapter concludes with Section 2.4, which outlines a summary of the chapter.

2.2 Botnets

The name botnet, is derived from the term "bot" or simply "robot", which is a piece of software that can be used to infect a device and automate tasks over the Internet. A cluster of these bots forms what is known as a network of bots or botnets, which are a group of interconnected devices. However, botnets are usually controlled by an attacker remotely (Xie et al. 2008). Furthermore, botnets are capable of sending a huge amount of spam mails in a limited space of time (Xie et al. 2008). As a result, they have been used by cybercriminals to orchestrate criminal activities such as sending spam emails, performing distributed denial of service attacks (DDoS), providing an attacker with full access to an infected system and keystroke logging.

Figure 2.1 below shows how a botnet works.

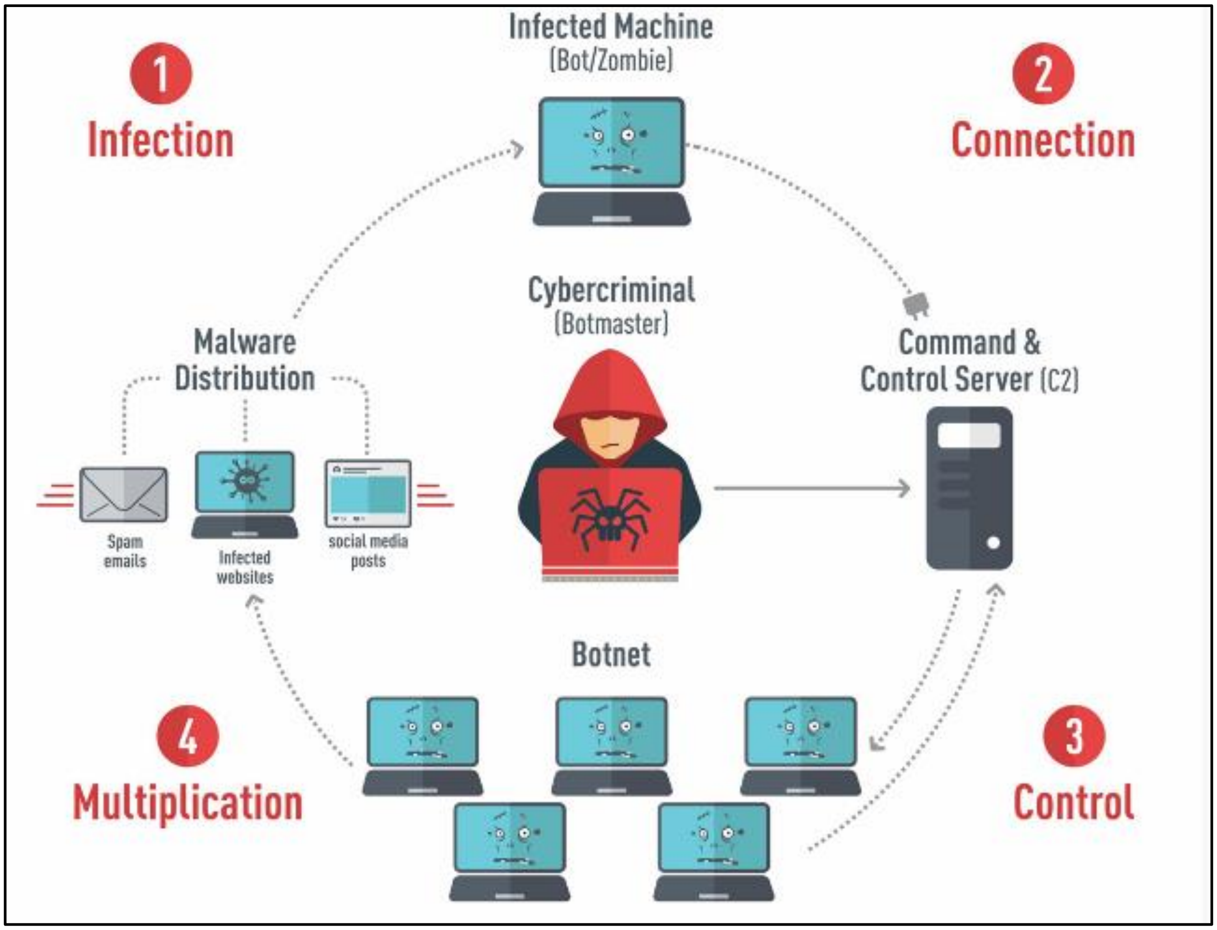


Figure 2.1. Structure of a botnet (Haylee, 2017)

The process begins at phase 1, in which the botnets infect a machine connected to the Internet through methods including email or drive by downloads. These methods are described in detail in sub section 2.2.1. Once the botnet executes on the infected machine, it connects to the command and control server and thus constituting phase 2 of the process. Here, the cybercriminal or “Botmaster” gains control of the botnet remotely and can start passing instructions to the command and control server through this remote control. The botmaster can then use these botnets to perform a variety of malicious attacks such as infecting other computers thereby increasing the number of botnets (phase 3 & 4). The botmaster can also use the botnets to execute distributed denial or service (DDoS) attacks, distribute spam, or steal confidential data such as credit card details and passwords. The group of botnets or "zombies" all link back to a command and control server where they receive instructions from the attacker. Thus, botnets' major function is to infect computers.

The following subsections describes some of the methods of infection and how botnets can be used to solve a digital investigation.

2.2.1 Method of Infection

Botnets infect computers through two means, which are (i) drive-by downloads and (ii) email (Fisher, 2013).

Firstly, drive-by downloads make use of a vulnerability that exists in a popular web browser such as Mozilla Firefox or Google Chrome. An attacker injects his/her own malicious code such that when a user clicks a particular webpage link, the user is re-directed to the attackers' website to download the malicious software. Pop-up ads are one of the tactics attackers use to attract the attention of the user to click (Fisher, 2013). Pop-up ads contain a message which might inform the user about how to optimise the PC to make it faster. A typical pop-up ad is shown in Figure 2.2 below.

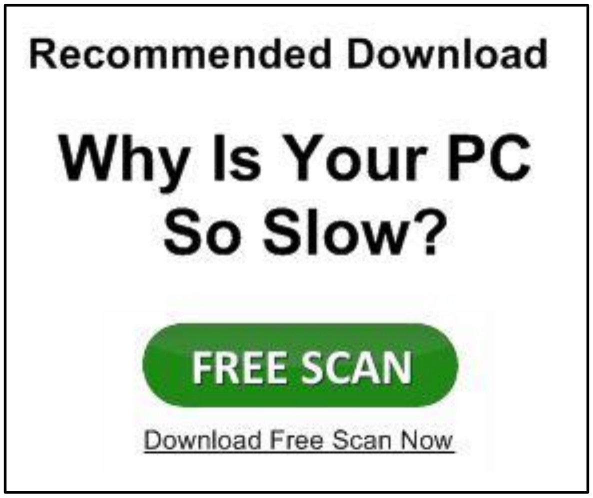


Figure 2.2. A typical pop-up ad used to redirect a user to a malicious site (PC Pitstop n.d)

Once the user clicks the ad, the user gets redirected to the attackers' website, where the user unknowingly downloads the software with the malicious code embedded in it. Once installed, the attacker can then access the computer remotely.

Secondly, the other method of infection is through email. Here, the user receives an email containing an attachment of a word document or a pdf that contains an embedded malicious code from the attacker. Once the user opens the attachment, the malicious code gets executed as well, and the computer makes contact with the attacker. The attacker can then issue commands remotely to the infected system and install more malicious software to allow him/her full access to the machine (Fisher, 2013).

2.2.2 Botnets for the good

Although botnets are used mainly to commit illegal activities such as DDoS attacks, sending of spam and phishing mails, their nature of operation can be viewed as a way of collecting digital evidence on a cloud infrastructure, as proposed by Kebande and Venter (2014). Their research work looked at how botnets can be used to harvest digital evidence in a non-malicious fashion in a simulated environment with the intent to attain digital forensic readiness in the cloud (Kebande et al. 2016). This will be discussed in further detail in Chapter 4. However, it is vital to comprehend what cloud computing entails in order to understand how botnets can be used in a cloud environment. This is elaborated in the next section.

2.3 Cloud Computing

The National Institute of Standards and Technology (NIST) defines cloud computing as, "a model for enabling ubiquitous, 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" (Mell & Grance, 2011, p. 2). This indicates that cloud computing entails the use of a virtual platform to host software services. This virtual platform can be used on several different workstations connected via a network. A virtual platform is one that can be provided by a cloud operating system. The cloud operating system is discussed in detail in section 2.3.3. A cloud model described in the cloud computing definition stated above consists of two types of models namely (i) the deployable models and (ii) the service models. The following sections explain both models.

2.3.1 Cloud deployment models

Since a cloud environment is elastic, it is possible to deploy the cloud into different cloud models. Deployment is the process of making software available for use. In this case of the cloud, the use will be where the software would be running, which will be a cloud model. A cloud model is a setting of a cloud environment that takes into account specifications such as the storage capacity, ownership and accessibility (Sam Solutions, 2017). Cloud models are based on their organisational deployment and storage structure of held information (Krutz & Vines, 2010). Finally, the four popular cloud deployment models are namely the public cloud, private cloud, community cloud and hybrid cloud. The cloud models are briefly discussed in the sub-sections below.

2.3.1.1 Public Cloud

In a public cloud model, the cloud service provider (CSP) is responsible for the upgrade and maintenance of the cloud infrastructure across all data centres (Krutz & Vines, 2010). Public clouds contain more than one user however, the CSP holds the administrative privileges of the cloud. In a typical public cloud setup, the CSP leases out cloud resources and virtual storage to the user. Examples of public cloud service services include Amazon’s Elastic Compute Cloud (EC2), Google’s AppEngine and Microsoft’s Azure Services platform (Zhang et al. 2010).

2.3.1.2 Private Cloud

A private cloud model is characterised by cloud infrastructure exclusively used by an individual organisation to provide virtual storage and computing resources for the particular organisation (Mell & Grance, 2011). The cloud infrastructure is owned by the organisation which controls organisational data and use. Examples of CSPs that deploy private clouds include Rackspace and VMware. Rackspace also contributed to the development of the OpenStack cloud operating system (Chen, et.al, 2017) which is mentioned in section 2.3.3.

2.3.1.3 Community Cloud

The cloud infrastructure in a community cloud model is shared among a community of several organisations (Goyal, 2014). The organisations share a common organisational goal or intent to share a set of IT resources (Krutz & Vines, 2010). In addition, the community cloud infrastructure will be administered by the community. Examples of community cloud services include G Suite for government and Microsoft 365 Government Community Cloud (Techno-Pulse, 2011).

2.3.1.4 Hybrid Cloud

In a hybrid cloud deployment model, the cloud infrastructure consists of two or more cloud deployment models, public, private and community, as discussed above. The main purpose of a hybrid cloud is to provide load balancing across multiple clouds (Krutz & Vines, 2010). This is best exemplified in the case of an organisation that uses a private cloud where they can obtain additional resources on lease from a public cloud in the event of having run out of cloud resources such as storage. Examples of hybrid cloud services are Microsoft Azure and VMware Cloud (Techno-Pulse, 2011).

Therefore, organisations might want to use one or more of the cloud deployment models described above to perform various organisational services. It also becomes necessary for the organisation to choose a cloud service model suitable for their organisation. The following section elaborates on the cloud service models.

2.3.2 Cloud Computing Service Models

A cloud service model refers to the types of cloud services that can be provided to customers. Cloud computing comes with three service models and these are Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) (Mell & Grance, 2011). A diagrammatical representation of these cloud service models is shown in Figure 2.3 on page 23.

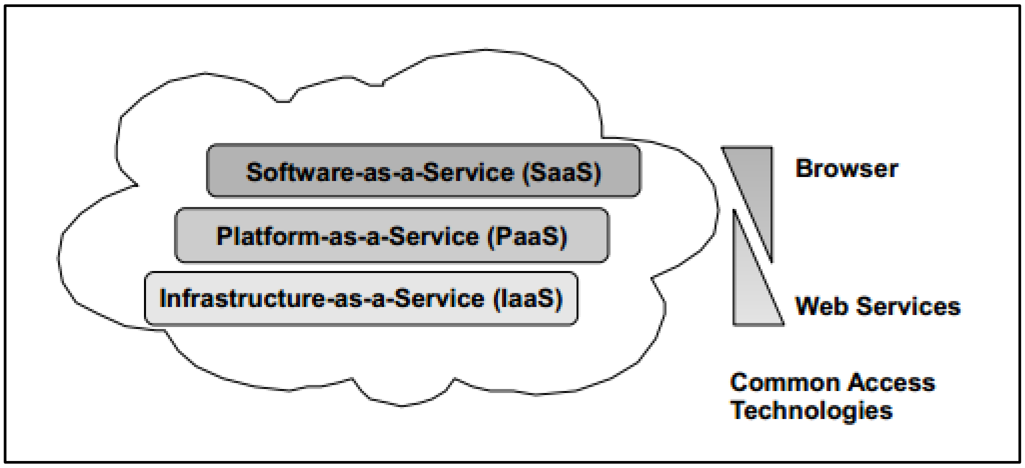

Software-as-a-Service (SaaS)
Platform-as-a-Service (PaaS)
Infrastructure-as-a-Service (IaaS)
Browser
Web Services
Common Access Technologies

Figure 2.3. Cloud computing service models (Jensen et al. 2009, p. 109)

The above-mentioned service models in Figure 2.3 are described briefly as follows:

2.3.2.1 Software as a Service (SaaS)

The Software as a Service (SaaS) service model furnishes the cloud user with the capability of executing software applications on a cloud infrastructure (Mell & Grance, 2011). The software applications provided on SaaS service model can be accessed using a variety of devices that include computers, laptops and tablets. The applications can run as an independent application or through an interface for example a web browser (Subashini & Kavitha, 2010).

2.3.2.2 Platform as a Service (PaaS)

The Platform as a Service (PaaS) service model enables the cloud user to set up user created applications in the cloud infrastructure using cloud components supported by the CSP that include services, libraries, and programming language (Mell & Grance, 2011). However, PaaS does not provide the cloud user with access to manipulate the underpinning cloud architecture for instance the network, storage and operating systems in use.

2.3.2.3 Infrastructure as a Service (IaaS)

The Infrastructure as a Service (IaaS) service model provides the user with fundamental computing resources, including storage and networking, to execute software such as operating systems and networking applications (Mell & Grance, 2011). Cloud infrastructure can be accessed through the use of virtual machines (Khajeh-Hosseini et al. 2010). OpenStack (mentioned in section 2.3.3 following) is deployed as IaaS and provides users with computing resources such as networking and storage (OpenStack, 2018). OpenStack is a typical example of a cloud operating system. The following section elaborates more on cloud operating systems.

2.3.3 Cloud Operating Systems

A cloud operating system is a software platform that is used in the management and monitoring of cloud resources to ensure that they are used effectively and efficiently (Chen, et.al, 2017). Cloud operating systems are similar to the traditional operating systems for instance Windows and Linux in that they also manage the hardware and software resources of a cloud infrastructure. The majority of cloud operating systems are built on Linux operating systems. In addition, these operating systems provide a virtualisation environment to run virtual servers and infrastructure (Chen, et.al, 2017).

The use of cloud operating systems depend on a number of factors. The factors include the available resources, virtual setup and cloud services that will be used. Some of the service uses include storage, streaming, office suite, music and videos. These services are enabled by cloud operating systems such as OpenStack, OpenNebula, EUCALYPTUS and Nimbus (Kurup et al. 2015).

This research study makes use of OpenStack operating system to provide an operational cloud computing environment to deploy and test the software prototype. OpenStack is an open source cloud operating system that contains a collection of software tools responsible for managing and configuring cloud computing environments (OpenStack, 2018). The following section concludes the chapter.

2.4 Conclusion

This chapter introduced botnets and cloud computing. Section 2.2 described how botnets infect machines and propagate. The DFECS proposed by Kebande et al. (2016) uses some of the botnet propagation techniques though it harvests digital information in a non-malicious fashion. Section 2.3 focused on the two cloud models namely the deployable and service models. In addition, an overview of the cloud operating systems was presented in which OpenStack, an open source cloud operating system used as an operational cloud environment to deploy the software prototype, is identified as the main focus of this study.

The next chapter focuses on digital forensics.

Chapter 3: Digital Forensics

3.1 Introduction

The chapter presents the background on digital forensics and digital forensic readiness. It also highlights the challenges faced by digital forensic investigators (DFI) in performing digital forensic investigations in the cloud. The chapter also describes the ISO/IEC 27043 standard that can be used to conduct digital forensic investigations in a cloud environment. Finally, the chapter also describes the legal issues encountered by DFI in performing digital forensic investigations in the cloud.

Every digital forensic investigation entails the collection of digital evidence. The digital evidence can be any digital data that might prove useful in solving the investigation. This evidence can also be used later in legal proceedings. This highlights the need to ensure that all the legal requirements are followed in order to make the digital evidence admissible in a court of law. Therefore, digital forensic investigators need to ensure that digital evidence is collected in a forensic manner that ensures and maintains the confidentiality, integrity and availability of the collected evidence.

The rest of the chapter is structured as follows: Section 3.2 outlines the background to digital forensics and the digital forensic process models. Section 3.3 discusses on digital forensic readiness and the digital forensic challenges in the cloud. Section 3.4 focuses on the ISO/IEC 27043 international standard while Section 3.5 considers some of the legal issues faced by digital forensic investigators when conducting digital investigations. Finally, Section 3.6 concludes the chapter.

3.2 Digital Forensics

Investigations are a routine and integral process after a crime has been committed such as murder, rape or theft, in order to find the perpetrator. Law enforcement officials, who follow a predefined set of steps, normally carry out these investigations. For example, the investigators combing a murder scene make use of gloves when collecting evidence such as a murder weapon. This is done to avoid contaminating the evidence. There is need, in the case of a crime that involves the use of electronic devices for example computers and mobile phones, to make use of digital forensics to investigate the crime.

The following subsections provide background information such as the definition and history of digital forensics, digital forensic process models and cloud forensics in order to present what digital forensics entails.

3.2.1 Definition and History

Digital forensics (DF) is defined as a process that uses scientifically demonstrated techniques in the "preservation, collection, validation, identification, analysis, interpretation, documentation and presentation" of digital data retrieved from digital devices for the purposes of reconstructing events that show to be criminal in nature (Palmer, 2001, p.16). DF has, in the past decade proved to be a useful tool in combating crime (Taylor et al. 2011). It seeks to provide approaches to acquire digital data from electronic devices with the aim of distinguishing potential culprits of the crime.

The utilisation of computing appliances transformed drastically after the 1950s. Writings such as Donn Parker's "Crime by Computer" (Parker & Parker, 1976) depicts the earliest first ways that one can employ to perform an investigation using digital data on a crime that was perpetrated with the help of an electronic computing appliance. Another publication by Cliff Stoll, "The Cuckoo's Egg" (Stoll, 1990) describes how the DF profession started in the 1990s. Furthermore, the International Organisation on Computer Evidence (IOCE) was founded in 1995 and it looked into providing aid in crimes that involve digital evidence (Pollitt, 2010). As time progressed, DF standards were developed. One such standard is the ISO/IEC 27043, which seeks to provide digital forensic norms for collecting digital evidence, and to achieve its storage and preservation (ISO/IEC 27043, 2015). Nevertheless, DF is a science and thus, needs to follow a scientific process. These scientific processes, called digital forensic process models, are discussed in the following section.

3.2.2 Digital Forensic Process Models

Any scientific method follows a predefined set of processes, with DF, as a science, following a scientific process. A DF process model is a scientific method which follows a predefined set of forensic processes. Various digital forensic process models have been proposed in literature (DFRWS, 2001; Reith et.al, 2002; Carrier & Spafford, 2003; Beebe

& Clark, 2005; Agarwal et al, 2011; Cohen, 2012; Valjarevic & Venter, 2012; Kebande and Venter, 2014). However, this research study only focuses on one of the process models, the Harmonised Digital Forensic Investigation Process Model (HDFIPM). Hence, a brief description is provided below in order to present an insight of some of the processes covered in the process model.

Valjarevic and Venter (2012), proposed a process model they titled, the Harmonised Digital Forensic Investigation Process Model (HDFIPM). The process model consists of the following phases (in chronological order): “incident detection, first response, planning, preparation, collection, transportation, storage, analysis, presentation and conclusion”. It also consists of concurrent processes, which happen throughout the phases, and these are: “obtaining authorisation, documentation, information flow, preservation of chain of evidence, and interaction with physical investigation” (Valjarevic & Venter, 2012). This process model constitutes part of the ISO/IEC 27043 standard. The software prototype used in this research study follows some of the phases in the HDFIPM process model, which are the collection, transportation, storage phases. The following section describes on cloud forensics in order to explain how the digital forensic process models are implemented on the cloud.

3.2.3 Cloud Forensics

Meyer and Stander (2015, p. 286) define cloud forensics as, “the process to retrieve digital evidence from the cloud for investigative purposes.” Cloud forensics can generally be seen as a subsection of digital forensics. It encompasses the use of traditional digital forensics methodologies in order to acquire digital evidence on a cloud infrastructure for investigative purposes. Digital evidence may be acquired from a variety of sources on the cloud, for example in a cloud instance running on a public cloud infrastructure (Meyer and Stander, 2015). In addition, cloud forensics can be used to solve cloud security incidences.

Kaufman (2009) states that there is need for a proactive approach in order to ensure security of data within the cloud. A proactive approach to provide a solution to the security issues in cloud environments can be achieved through implementing readiness within the cloud. This would permit organisations to boost the use of the gathered potential digital evidence. The following section discusses on this readiness in the cloud.

3.3 Digital Forensic Readiness

Digital forensic readiness (DFR) is defined by Tan (2001) as the capability of a digital forensic investigation agency in boosting the usage of collected digital evidence whilst reducing the expense of a digital forensic investigation to responding to an incident. Rowlingson (2004) states that the main purpose of DFR in a digital forensic investigation is to make the most of the potential digital data whilst reducing the time and costs incurred in performing the forensic investigation.

DFR encompasses the gathering of digital data from computer components such as flash drives, hard drives and random access memory. With regards to logs stored on hard drives, DFR seeks to understand how the logging process happens, what processes logs the events, how the logs of the events are stored, the structure and type of data (De Marco et al. 2014). In addition, the digital forensic investigator (DFI) needs to follow proper forensic processes when performing a digital forensic investigation. DFR will make the investigation process easier for the DFI in that PDE is collected proactively, which means that the DFI can investigate how the incident took place. Potential digital evidence, which is any digital data collected that might be relevant to the digital forensic investigation, may be acquired from a variety of sources on the cloud. For example, a virtual machine used in a public cloud infrastructure may contain potential evidence in cases where an incident has occurred at that particular virtual machine. The underlying network infrastructure may also contain PDE (Meyer and Stander, 2015). Finally, DFR can also be employed in the cloud and this is discussed in the following section.

3.3.1 Digital Forensic Readiness in the cloud

In a typical digital forensic investigation, the DFI uses the traditional search and seizure method, in which the investigator seizes a particular electronic device such as a laptop and makes a bit by bit copy of the seized device (Casey, 2011). This procedure is easy when one has access to the physical device, but in a cloud environment this becomes a challenge because the data centres and cloud infrastructures maybe sitting in different areas (Barbara, 2009).

There exists no formal structure of conducting DFR in a cloud infrastructure (Kebande & Venter, 2014). Consequently, a number of international standards, such as the ISO/IEC 27043:2015, have been developed as international standard seeking to provide a formal method for conducting DFR. ISO/IEC 27043 consists of readiness processes that seek to maximise the potential worthiness of computer evidence in order to lower the costs involved in a typical digital forensic investigation.

Kebande and Venter (2014) argue that there lacks a structured approach for conducting DFR in the cloud without the need to adjust or change the existing cloud structure. This alteration of the existing cloud infrastructure is a huge challenge because of the costs incurred in performing DFR in the cloud (Kebande & Venter, 2014). ISO/IEC 27043 itself does not directly target the cloud environment but encompasses all DFR processes that can be conducted in any type of environment.

Nevertheless, De Marco et al. (2014) argues that DFR can be implemented through using a systematic and proactive methodology in the collection and storage of digital evidence. De Marco et al. (2014) note further that the DFR capability in the cloud can be attained through the employment of an information collecting system with capabilities to both collect sensitive data and warn the host system before an incident occurs.

A study by Van Staden and Venter (2012) focused on the usage of performance monitoring tools to attain DFR in the cloud. This study made use of a Learning Management System (LMS) as performance-monitoring tools in acquiring data from the LMS. Their results show that it is possible to acquire digital data while using the performance monitoring tool. Therefore, this data can be used by DFIs during forensic investigations.

Nonetheless, there is concern on the way digital forensic investigations are executed to combat the threats and attacks in a cloud platform and one of these concerns include predominantly the absence of DFR (Tan, 2001). The following section provides a description of some of the DF challenges in the cloud.

3.3.2 Digital Forensic Challenges in the Cloud

The organisations' increased use and reliance on technology and the rapid evolution of technology makes securing digital infrastructure a challenge (Hay et al. 2011). The observed increase in use of cloud computing technology leads to the increase in the risk of getting cyber-related attacks (Jang-Jaccard & Nepal, 2014). There is need to perform digital investigations on the cloud in order to look into these attacks. However, there are various challenges on conducting these investigations owing to the way the cloud infrastructure is distributed. Therefore, as noted by Hay et al. (2011), it is necessary for digital forensic investigators to equip themselves with the latest digital forensic tools so that they can investigate better the incidences which occur in cloud environments.

Digital forensic investigators (DFIs) face challenges when performing digital forensic investigations on the cloud. Taylor et al. (2011) discuss the challenges DFIs encounter when performing digital forensic investigations on the cloud. One such challenge is the acquisition of digital evidence, especially in a case, where a cloud infrastructure with virtual machines hosts a variety of software applications for use by cloud users. SaaS applications are constantly receiving updates and the updates have the potential to overwrite the old information that was on the previous version of the applications (Akervik, 2019). As a result, the retrieval of potential digital evidence (PDE) lying within the previous cloud application versions might prove difficult since the data might be overwritten.

DFIs are also faced with data gathering challenges. First, the typical solution of shutting down the entire network, in a network intrusion investigation, might not lead to the preservation of PDE. Some of the PDE may no longer be available or might prove difficult to collect and preserve in a forensically sound fashion because PDE might not be there due to the shutdown (Casey, 2011). Some of the PDE can be found in various logs and yet various organisations do not collect or retain the logs if they are a week or more old. These organisations might also not have a storage method that ensures the integrity of the data stored in the logs (Casey, 2011).

Cyber-attacks are a further challenge to DFR. Cyber-attacks can hit the cloud environment in the form a of denial of service attack (DoS) and DoS attacks major impact in preventing a user from accessing a service. Thus, a DoS attack on the cloud services may inhibit a DFI from performing an investigation because the investigator will not be able to access some cloud services, such as logs, which might contain vital evidence data that might help in the forensic investigation (Deshmukh & Devadkar, 2015). The challenge to DFR lies in the increase in time taken to conduct the investigation and associated increase in the costs as there might be need for data recovery tools.

Another challenge faced by DFIs in their attempts at acquiring digital evidence from the cloud is the absence of digital forensic tools that are capable of extracting digital evidence from the cloud (Casey, 2011). Traditional forensic acquisition tools are designed to acquire digital evidence from an electronic device, such as a personal laptop, which can be accessed physically. However, the conditions in a typical cloud setup are such that, organisational data can be stored on a cloud infrastructure that is located at different geographical locations. As a result, the acquisition of digital evidence with traditional forensic acquisition tools might prove difficult as the DFIs would need to physically travel to each site where the cloud infrastructure is located. Kebande and Venter, (2015) also point out that there is lack of proactive solutions in the cloud. This challenge extrapolates on the difficulties faced while trying to attain DFR in the cloud without the need to alter the existing cloud infrastructure. In addition, Dykstra and Sherman (2012) point out that there are no clear guidelines on how to acquire digital evidence on the cloud. Nevertheless, the standard developed so far, called ISO/IEC 27043, seeks to address this issue and is described in detail in section 3.4 below.

3.4 ISO/IEC 27043:2015

ISO/IEC 27043, is an international standard that entails, "information technology, security techniques and incident investigation principles and process". (ISO/IEC 27043 2015, p.1). In addition, “ISO/IEC 27043:2015 provides guidelines based on idealized models for common incident investigation processes across various incident investigation scenarios involving digital evidence" (ISO/IEC 27043 2015, p1).

These set of guidelines provided by the ISO/IEC 27043 can then be used by DFIs to attain DFR in the cloud. The ISO/IEC27043 consists of digital investigation processes divided into five classes and these are namely the, “Readiness Processes, Initialisation Processes, Acquisitive Processes, Investigative Processes and Concurrent Processes”

Figure 3.1 below diagrammatically describes the relationship between these processes.

Figure 3.1. Digital investigation process classes (ISO/IEC 27043 2015, p. 15)

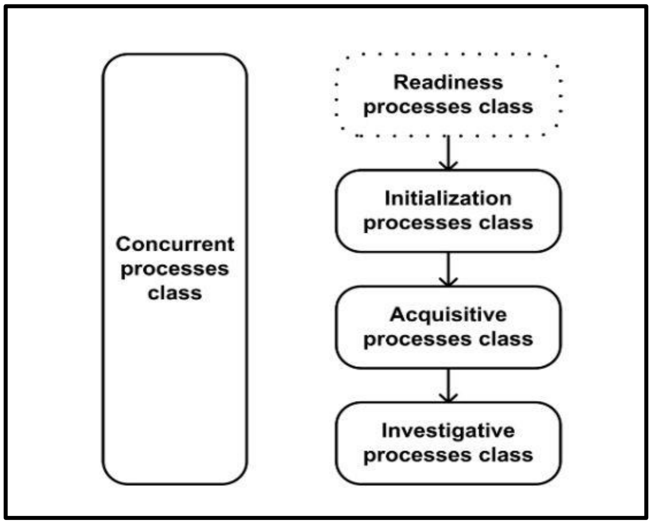

Concurrent processes class
Readiness processes class
Initialization processes class
Acquisitive processes class
Investigative processes class

Figure 3.1 shows that the digital investigation processes are multi-layered. They begin with the readiness process class and end with the investigative process class. The concurrent processes are unique in that they run throughout all the four process classes. According to the ISO/IEC 27043 standard, the concurrent class consists of the following processes, "managing information flow, documentation, obtaining authorisation, preserving chain of custody and preserving digital evidence".

These processes are important for DFIs to perform at each of the processes in order to preserve digital evidence. The readiness process classes at high level are examined further since this research only deals with DFR.

The readiness process class is a, "class of processes dealing with setting up an organization in such a way that, in the case that a digital investigation is required, such organization possesses the ability to maximize its potential to use digital evidence whilst minimizing the time and costs of an investigation." (ISO/IEC 27043 2015, p.7). Thus, the goal of any DFR process is "to maximise the potential use of digital evidence whilst minimizing the time and costs of conducting a digital forensic investigation" (Tan, 2001, p. 1).

Therefore, the ISO/IEC 27043 standard seeks to address the challenge of lack of proper guidelines in conducting digital forensic investigations in the cloud. However, it is worth noting that there exist legal implications for conducting such forensic investigations on the cloud. For example, if a public cloud is hosted across several different geographical regions, it might mean there will be different jurisdictions with regards to ‘who’ will be authorised access to ‘what’ data (Dykstra & Sherman, 2012). Some of these legal issues are discussed in the following section 3.5.

3.5 Legal Aspects

The use of IT technology in this modern age has led to the rise of cybercrimes. Digital forensics is seen as a means to gather digital evidence that assists in solving these cybercrimes. Once digital evidence has been found, analysed and preserved, it becomes necessary to present the digital evidence in court. The collection and presentation of digital evidence in a court of law can be faced with a variety of challenges. This research studies the harvesting of digital evidence from an operational cloud environment. As a result, it is necessary to understand the legal issues that might be involved in the harvesting of digital information. Some of these issues range from the different jurisdictions across different regions to a user’s privacy rights on personal information.

Cloud service providers (CSPs) might have structures where by their cloud infrastructures are situated in one region and the cloud servers situated in another region with different jurisdiction (Wilson, 2015). This raises the issue of where the actual forensic information is located, and also which jurisdiction applies since the data required may be scattered across multiple regions. This means that it will be necessary to identify the court that has the right to issue subpoenas and other authorisations needed for one to collect the data. This may also lead to more time being taken in gathering up of all these legal documents.

Another legal issue affecting digital forensic investigations, as pointed out by Brungs and Jamieson (2005), relates to the exhibition of digital evidence in a court of law. The differences in jurisdictional law might create difficulties in presenting the evidence in a court of law. Legislation differs from country to country and from region to region. As a result, the location of the court where the digital evidence will be presented becomes a point of consideration in order to ensure admissibility. This is because some of the digital evidence that might be collected can be accepted in some regions and yet denied in others (Brungs & Jamieson, 2005).

Finally, each region has specific requirements concerning the aquisition of digital evidence. Therefore, it is necessary that DFIs take note of these requirements and ensure that they comply with them. For example, South Africa demands that DFIs need to observe certain legal acts when performing digital forensic investigations. These include the Protection of Personal Information Act (PoPI) of 2013 and the Electronic Communications and Transactions Act (ECT) of 2002. The PoPI Act, is meant, “to promote the protection of personal information processed by public and private bodies.” (PoPI Act, 2013). The PoPI Act also provides regulations with regards to the acquisition, processing, storing and analysis of personal information. The ECT Act seeks, “to provide for the facilitation and regulations of electronic communications and transactions.” (ECT Act, 2002). The following section concludes the chapter.

3.6 Conclusion

This chapter presented background information to digital forensics and digital forensic readiness. Section 3.2 outlined the definition of DF and described the DF process models. A DFR overview was provided in Section 3.3 and it was established that this research investigates how DFR can be employed in the cloud. The chapter also discussed, in Section 3.3 some of the challenges encountered by digital forensic investigators in cloud forensics. Section 3.4 outlined the ISO/IEC 27043 standard, which is an international standard that seeks to provide a formal method for conducting DFR. The chapter concluded with Section 3.5, which explained some of the legal issues faced by DFIs in conducting digital forensic investigations in the cloud.

The next chapter introduces the prototype.

Chapter 4: Implementation of a Non-Malicious Botnet

4.1 Introduction

Chapters 2 and 3 focused on background information about botnets, cloud computing and digital forensics. This background information highlighted on how a botnet operates. This chapter introduces the prototype, which is presented as a forensic agent making use of some of the botnet characteristics in order to harvest digital information in an operational cloud environment.

In a traditional forensic investigation, the forensic image is created before the digital forensic investigation takes place. This is possible since the DFI has access to the device that needs to be imaged. This becomes a challenge in the case of a cloud environment because the data centres and cloud infrastructures may be sitting in different areas (Barbara, 2009). In addition, there are no clear guidelines for conducting DFR in the cloud (Dykstra & Sherman, 2012). Therefore, the lack of standardised guidelines for conducting DFR in the cloud necessitates the use of the prototype as a proof of concept on how a proactive DFR approach can be implemented in an operational cloud environment.

The rest of the chapter is structured as follows: an overview of the prototype presented in Section 4.2 providing a brief overview of how it operates; Section 4.3, which describes the DFECS prototype; Section 4.4 that outlines describes the processes followed by the DFECS prototype; and Section 4.4, which concludes the chapter.

4.2 Prototype overview

The prototype provides a proof of concept for the proactive gathering of digital evidence in an operational cloud environment. The chosen cloud environment is OpenStack, which is an open source cloud operating system. The prototype proactively collects digital information on cloud instances hosted on the cloud and stores the collected digital information in a forensic database. The prototype hashes the collected information and, in that way, maintain the integrity of the collected data. It operates in way similar to that of how a botnet operates, however, in this case the use is not for malicious intents. Botnets were chosen in this research due to the attributes they possess and in particular their stealthiness, resilience and capability of gathering data (Mónica & Ribeiro, 2013). It is worth underscoring here that the prototype in question is represented as a botnet possessing characteristics that enable collection of digital information in an operational cloud environment for DFR purposes. The following section provides and in-depth description of the DFECS prototype and the way it operates.

4.3 Digital Forensic Evidence Collecting System

Kebande et al. (2016), developed a software prototype that seeks to prove the attainability of DFR in a virtualised environment. The authors designed a software prototype called the “Digital Forensic Evidence Collecting System (DFECS)”, which can collect digital data from a simulated environment and store the collected data in a forensic database. DFECS in essence collects digital data in a non-malicious manner. DFECS monitors system activities such as RAM usage, keystrokes made by the user and CPU usage. DFR in the context of their research was achieved through the modification of a botnet in order for it to act as a cloud agent for collecting digital data in a virtualised platform (Kebande et al. 2016). The modified structure of a botnet is deployed as an agent-based solution (ABS) in a simulated environment to forensically capture PDE in order to attain DFR. Figure 4.1 below shows a diagram of the process followed by the DFECS in harvesting digital information.

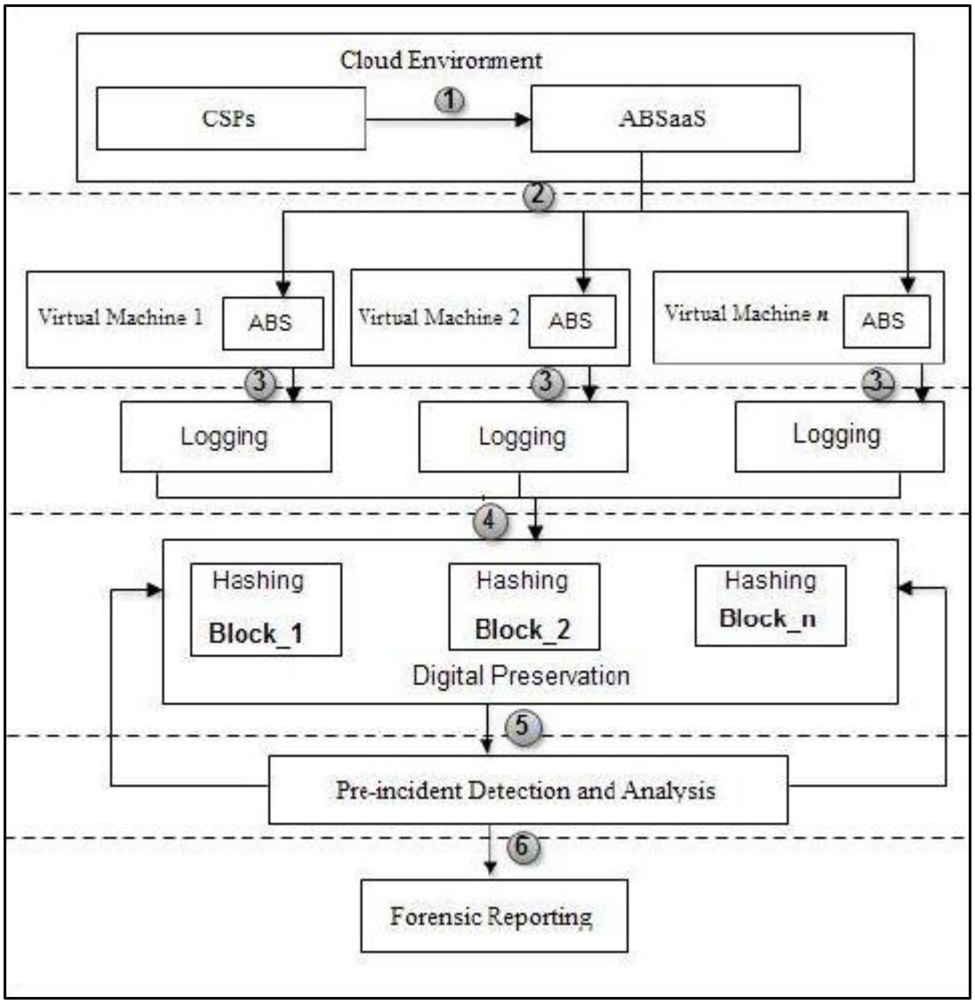


Figure 4.1. Digital information harvesting process using DFECS (Kebande et al. 2016, p.2)

Figure 4.1 shows the processes followed by the DFECS. Label 1 shows a cloud environment consisting of CSPs and ABSaaS. Agent Based-Solution as a Service (ABSaaS) denotes how the ABS is implemented in the cloud environment. In this case, it is implemented as SaaS but Kebande et al. (2016) term it ABSaaS. DFECS as an agent-based solution (ABS) is installed (label 2) as a cloud service in a cloud environment consisting of three virtual machines (label 3). As soon as the DFECS is operational, it starts capturing digital information in the virtual machines (label 3). The digital information is then hashed and forensically stored in a forensic database (label 4). Label 5 details the detection of incidences in the collected evidence, while a forensic report is finally produced in label 6 (Kebande et al. 2016).

DFECS follows the readiness processes stipulated in the ISO/IEC 27043 international standard in its gathering of PDE. DFECS collects both volatile and non-volatile digital data and stores the data in a forensic database. A forensic investigator can use this PDE later in a digital forensic investigation. This proactive approach has the advantage of making best use of PDE while reducing the costs and time taken when conducting a DFI in the cloud thereby providing a way of DFR in the cloud.

DFECS was never tested in an operational cloud environment. The major contribution provided by the research study in this mini-dissertation is to implement the DFECS in the OpenStack operational cloud environment to investigate if it can harvest PDE in an operational cloud environment and thereby prove the attainability of DFR in the cloud. The following section focuses on the digital information gathering process in order to explain how DFECS was implemented in OpenStack.

4.4 Prototype Processes

The prototype follows the following processes shown in figure 4.2. The processes are explained in detail in the following subsections.

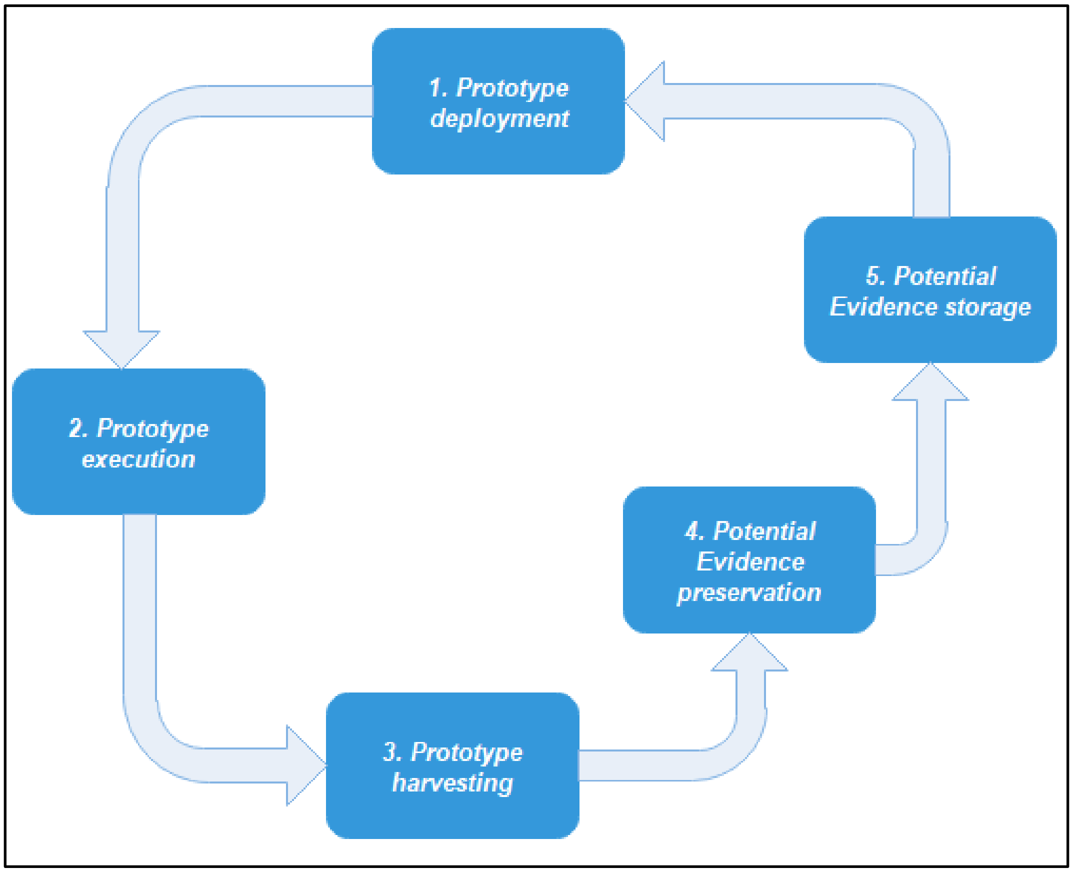


Figure 4.2. Prototype Processes

4.4.1 Prototype deployment

OpenStack deploys the prototype to the cloud instances through via file transfer protocol. The cloud instances are hosted in an operational cloud environment. Once the prototype has been deployed to the specific cloud instance, the command and control server can then be used to execute the prototype.

4.4.2 Prototype execution

This is the process that the prototype executes in the cloud instance once the deployment process is complete. Kebande et al. (2016), term this the “infection” process to signify the stage that the agent-based solution (ABS) executes for the purposes of collecting digital information. The command and control server is responsible for executing the prototype.

4.4.3 Prototype harvesting

This is the process where the prototype starts to acquire digital data from the cloud instance. The digital data collected by the prototype includes CPU usage, RAM usage and keystrokes on the keyboard.

4.4.4 Potential Evidence preservation

The prototype hashes the collected digital information in this process. Hashing is a preservation technique mention in the ISO/IEC27043 standard. Thus, the hashing here ensures the integrity of the collected information. Once integrity is maintained, the collected information can then be used in conducting digital forensic investigations.

4.4.5 Potential Evidence storage

The collected evidence is stored in a forensic database in this process. A MySQL database is used as it can store digital data. A DFI can access what was stored on the database later and use the information to perform digital forensic investigations. The following section concludes the chapter.

4.5 Conclusion

The chapter provided an insight on the prototype. An overview of the prototype and the development environment is presented in Section 4.2. Section 4.3 outlined the DFECS prototype operations while Section 4.4 presents a description on the prototype processes. The chapter noted that the research work by Kebande et al. (2016), underpins the processes used in the prototype to collect digital information in a virtual environment for DFR purposes.

The next chapter focuses on the implementation of the prototype in an operational cloud environment.

Chapter 5: Experimentation

5.1 Introduction

This chapter expounds on the experiments performed in this research study. The experiments sought to provide answers to the research questions posed in Chapter 1. In addition, the experiments show the implementation of the prototype in an operational cloud environment and then demonstrate the proof of concept in order to validate Kebande & Venter (2016) that the prototype can be implemented in an operational cloud environment.

The chapter also details some case scenarios on how the prototype can be used by DFIs in solving digital forensic investigations in the cloud. The prototype in question is represented here as a forensic agent, which possesses botnet characteristics that enable collection of digital information in a non-malicious manner and in an operational cloud environment for DFR purposes.

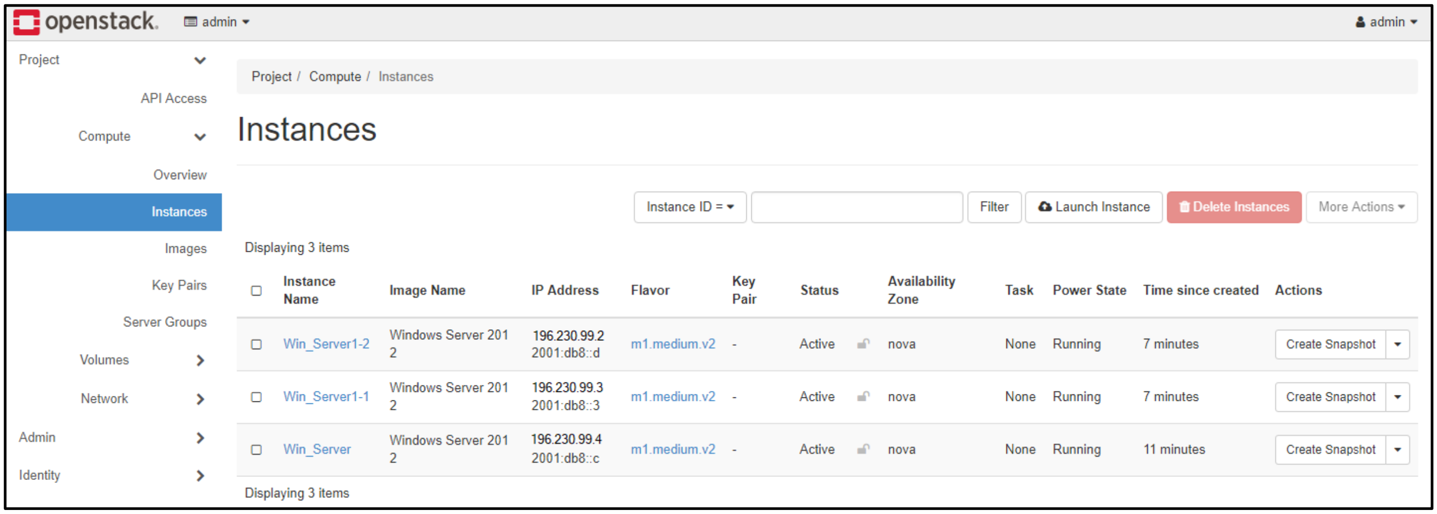
The chapter first presents an overview of the cloud operating system, OpenStack. This description is presented in Section 5.2, whilst Sections 5.3 and 5.4 summarise how the experiments were set up and describe the characteristics of the cloud instances setup in the OpenStack cloud environment. Section 5.5 describes the proof of concept and section 5.6 explains the scenarios performed in OpenStack. Finally, Section 5.7 concludes the chapter.

5.2 OpenStack Overview

OpenStack is an open source cloud operating system that creates and manages cloud infrastructures (OpenStack, 2018). The system, which is managed by the OpenStack Foundation, started in 2010 through a collaboration between NASA and Rackspace Hosting (Yadav, 2013). It provides a cloud computing environment where virtual servers and cloud resources are made available to the clients. OpenStack operates on both private and public clouds. Many of OpenStack’s cloud computing resources are deployed as Infrastructure as a Service (IaaS). In addition, OpenStack is designed in a way that offers cloud administrators a platform to deploy IaaS infrastructure and supply tools for creating and managing cloud instances on top of existing cloud infrastructure. This research study makes use of the OpenStack cloud platform to provide an operational cloud environment to test the prototype to harvest digital information in cloud instances hosted on OpenStack in order to attain DFR. The following section elaborates on how this implementation was carried out.

5.3 Prototype implementation in OpenStack

The cloud instances considered in this study are virtual machines that run on the OpenStack infrastructure. The cloud instances can be launched from the available OpenStack images. This research study made use of the Windows Server 2012 image to spawn cloud instances within OpenStack. The prototype was deployed to these cloud instances to test it in a cloud environment, in this case OpenStack. Figure 5.1 below shows the three (3) cloud instances created in OpenStack.



Instances

Displaying 3 items

□	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status		Availability Zone	Task	Power State	Time since created	Actions
□	Win_Server1-2	Windows Server 2012	196.230.99.2 2001:db8::d	m1.medium.v2	-	Active		nova	None	Running	7 minutes	Create Snapshot
□	Win_Server1-1	Windows Server 2012	196.230.99.3 2001:db8::3	m1.medium.v2	-	Active		nova	None	Running	7 minutes	Create Snapshot
□	Win_Server	Windows Server 2012	196.230.99.4 2001:db8::c	m1.medium.v2	-	Active		nova	None	Running	11 minutes	Create Snapshot

Displaying 3 items

Figure 5.1. Cloud instances in OpenStack

These instances were spawned from the Windows Server 2012 cloud image. The instance name column provides the name for each instance (*Win_Server, Win_Server1-1 & Win_Server1-2*). Each cloud instance is given a specific IP address by the OpenStack Network management component called Neutron. The cloud instances shown in Figure 5.1 above have a “running” state, which means that they are operating without any problems.

Various activities can be performed on the cloud instance. For example, you can access the Internet, install applications, copy and move folders and files. The completion of the setting-up of the cloud instances enables the performing of operations such as the deployment and testing of applications on the cloud. The next section discusses how the prototype was deployed into the cloud instances and the various tests performed.

5.4 Experimental Setup

The experimental set up to deploy the prototype to cloud instances followed the setup presented by Kebande and Venter (2016) in deploying DFECS to virtual machines. Figure 5.2 below shows the experimental set up used by both researchers.

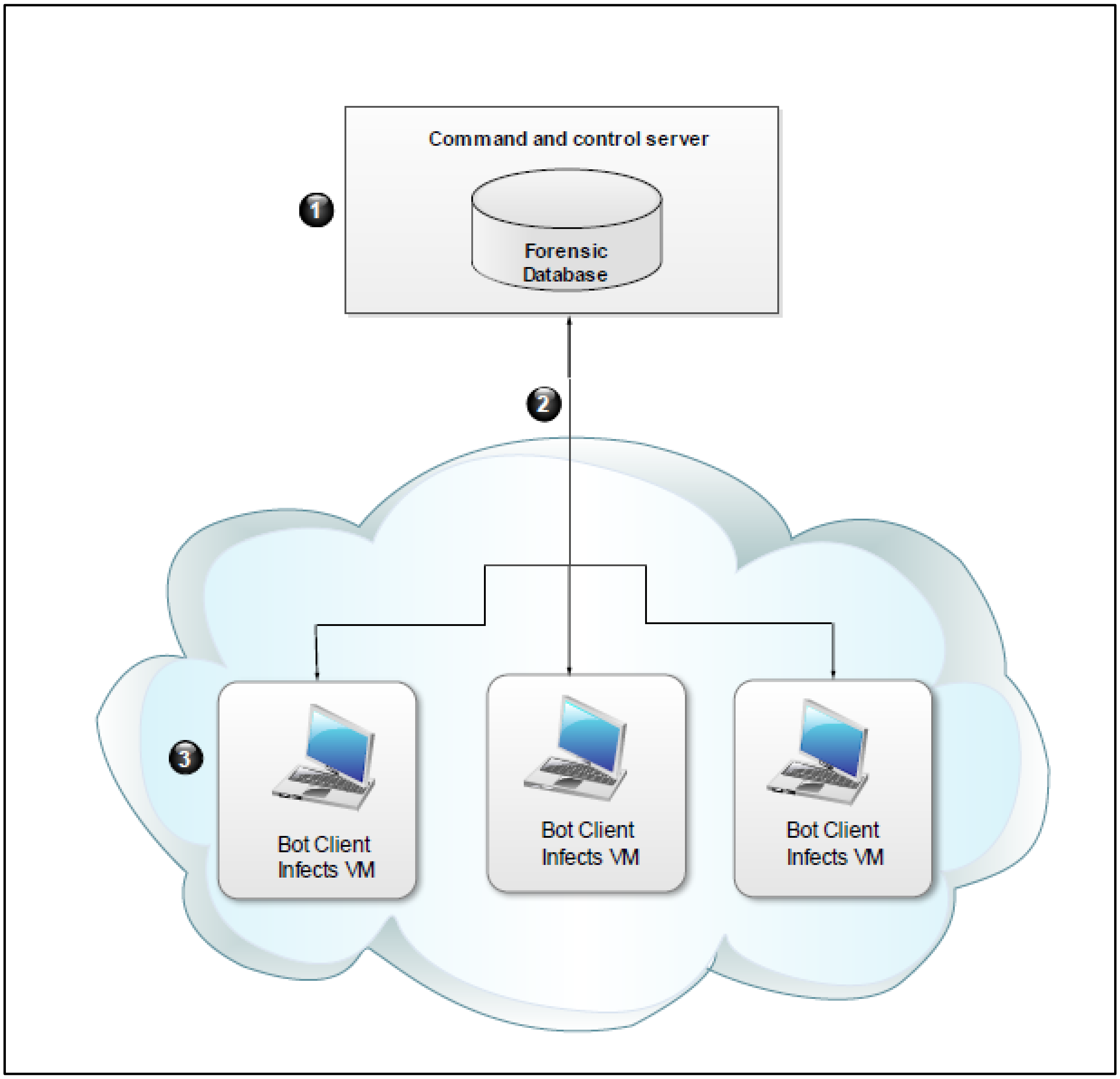


Figure 5.2. Experimental Setup of DFECS (Kebande & Venter. 2016)

The diagram shows how the DFECS was set up while making use of a virtual environment consisting of virtual machines. The same setup is used in this study, however, instead of making use of virtual machines, the prototype is deployed on cloud instances hosted in an operational cloud environment, in this case OpenStack. It is worth noting that the prototype is deployed to the cloud instances through via FTP. Once deployed, the command and control center can then execute the prototype.

Making use of the set up mentioned in Figure 5.2, the prototype is executed by the command and control server (label 1). Label 2 depicts the transfer of data between the command and control server and the cloud instances. The prototype collects PDE and dispatches it to the command and control server where it is kept in a forensic database.

The command and control server forms part of the prototype. The VMs (label 3) are the cloud instances setup in OpenStack. It is worth noting here that the use of OpenStack simulates an operational cloud environment to test the prototype. The following section describes the proof of concept of performing the above-mentioned experiment in an operational cloud environment.

5.5 Proof of concept

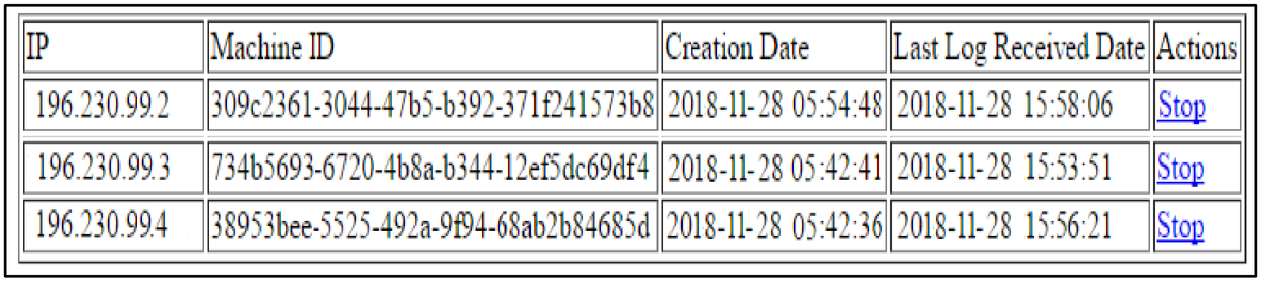
The proof of concept seeks to show the reader how the prototype operates in real time in an operational cloud environment. It is worth noting that this prototype is not new. It is an existing prototype that was developed and tested in a simulated environment Kebande et al. (2016), which they called DFECS. In this research study, the DFECS is extended and implemented in an operational cloud environment, which is OpenStack, to prove that DFR can be attained in an operational cloud environment and not in a simulated environment.

Modification was made on the way the DFECS posts digital information to the command and control server. Since the DFECS was running on OpenStack infrastructure, it was necessary to modify the way it posts data to ensure that the data arrives at the command and control server. The digital data had to pass through the OpenStack network management component, Neutron so modification was performed in order to make sure the that the data gets sent to the command and control server successfully.

The proof of concept follows the prototype processes described in Chapter 4.

5.5.1 Prototype command and control server

As noted previously, the command and control server is responsible for the deployment of the prototype to the cloud instances hosted in an operational cloud environment. Figure 5.3 below shows the command and control server used to start or stop the prototype:



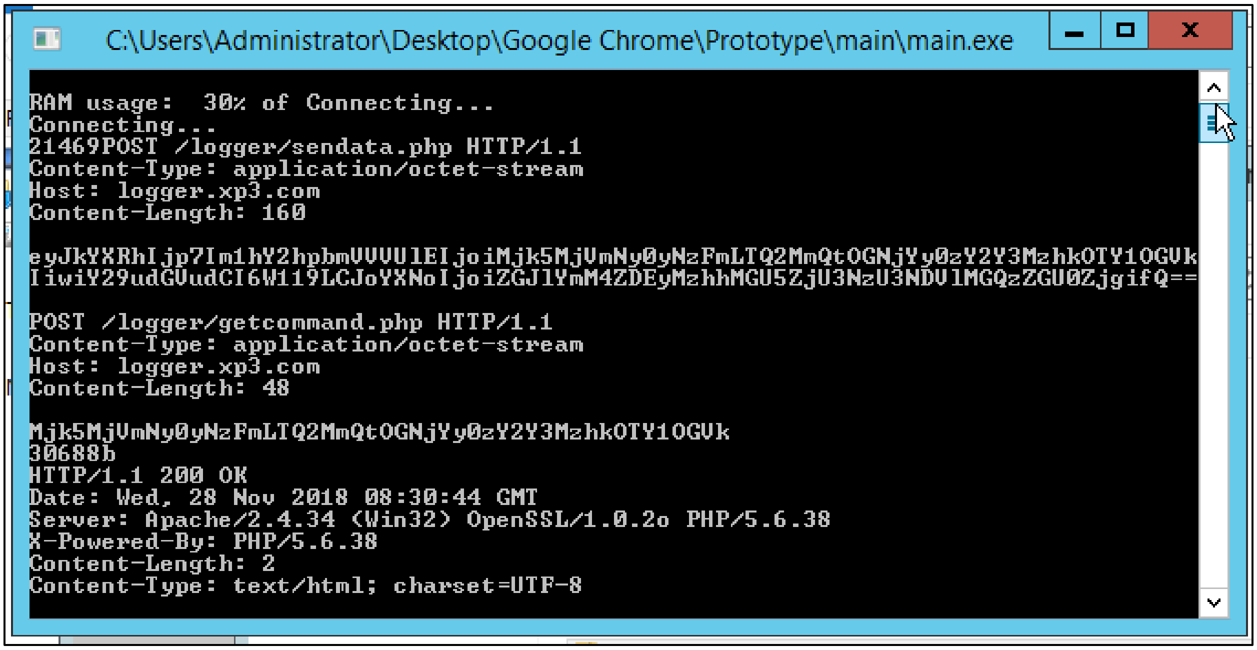
IP	Machine ID	Creation Date	Last Log Received Date	Actions
196.230.99.2	309c2361-3044-47b5-b392-371f241573b8	2018-11-28 05:54:48	2018-11-28 15:58:06	Stop
196.230.99.3	734b5693-6720-4b8a-b344-12ef5dc69df4	2018-11-28 05:42:41	2018-11-28 15:53:51	Stop
196.230.99.4	38953bee-5525-492a-9f94-68ab2b84685d	2018-11-28 05:42:36	2018-11-28 15:56:21	Stop

Figure 5.3. Command and Control server

The command and control server lists the IPs for the respective cloud instances (IP column). Each cloud instance has a specific Machine ID that can be used to identify the cloud instance (Machine ID column). Once prototype deployment is performed (via FTP) to a particular cloud instance, the prototype is initially executed manually so that it contacts the command and control server. Manual execution is only done initially to ensure contact with the command and control server. Once contact is made, there is no need to manually execute it again and this is done by the command and control server. The creation date and time (Creation Date columns) shows the time when the prototype in a respective cloud instance first contacted the command and control server. The "last log received date" shows the last log entry received from the particular cloud instance. The action state in the "actions" column depict the state of the prototype, in particular, whether it would be running or stopped. When start is clicked, the prototype executes in the cloud instance, and the clicking of stop halts the prototype from collecting digital information.

5.5.2 Prototype execution

Once "start" is clicked from the command and control server, the prototype executes in the cloud instance. Figure 5.4 below shows this process on when the execution takes place.

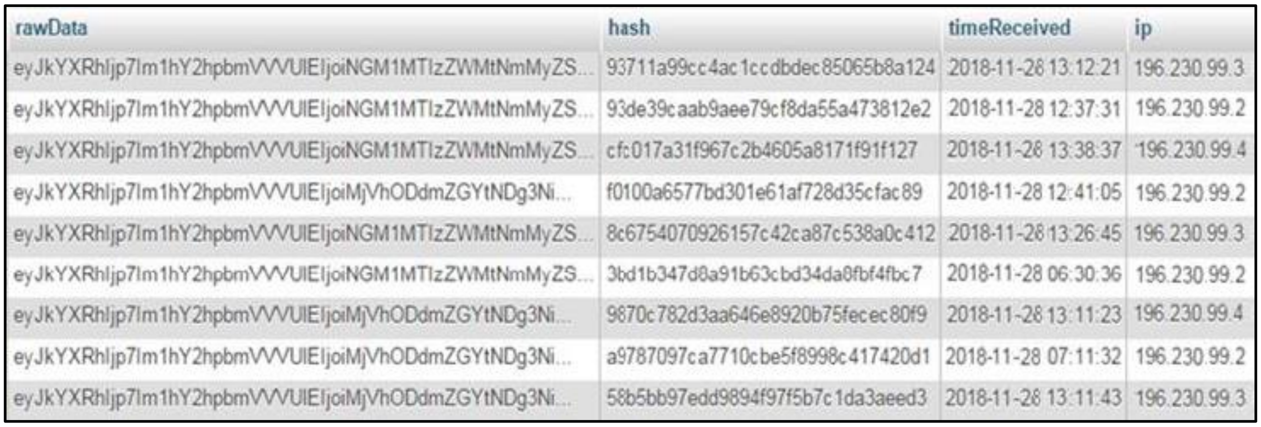


C:\Users\Administrator\Desktop\Google Chrome\Prototype\main\main.exe

```
RAM usage:  30% of Connecting...
Connecting...
21469POST /logger/sendata.php HTTP/1.1
Content-Type: application/octet-stream
Host: logger.xp3.com
Content-Length: 160

eyJkYXRhIjp7Im1hY2hpbmVVVU1EIjoiMjk5MjVmNy0yNzFmLTQ2MmQtOGNjYy0zY2Y3MzhkOTY1OGVk
IiwiY29udGVudCI6W119LCJoYXNoIjoiZGJ1YmM4ZDEyMzhhMGU5ZjU3NzU3NDV1MGQzZGU0ZjgifQ==

POST /logger/getcommand.php HTTP/1.1
Content-Type: application/octet-stream
Host: logger.xp3.com
Content-Length: 48

Mjk5MjVmNy0yNzFmLTQ2MmQtOGNjYy0zY2Y3MzhkOTY1OGVk
30688b
HTTP/1.1 200 OK
Date: Wed, 28 Nov 2018 08:30:44 GMT
Server: Apache/2.4.34 (Win32) OpenSSL/1.0.2o PHP/5.6.38
X-Powered-By: PHP/5.6.38
Content-Length: 2
Content-Type: text/html; charset=UTF-8
```

Figure 5.4. Prototype executed in cloud instance

The prototype operates in a stealth mode. This indicates that it operates behind the scenes such that even a cloud user making use of the cloud would not be disturbed. Figure 5.4 above shows the execution process when the stealth mode is disabled to show how the prototype operates. Once the prototype executes, it starts collecting digital information namely CPU usage, RAM usage and the keystrokes typed on the keyboard. These get captured at a 2-second interval. Once captured, the ‘chunk’ of captured raw data is hashed and is sent to the command and control server through a PHP POST method to http host *logger.xp3.com* (see Figure 5.4) where the command and control server is hosted. The captured hashes can be viewed from the command and control server as shown in figure 5.5 below:



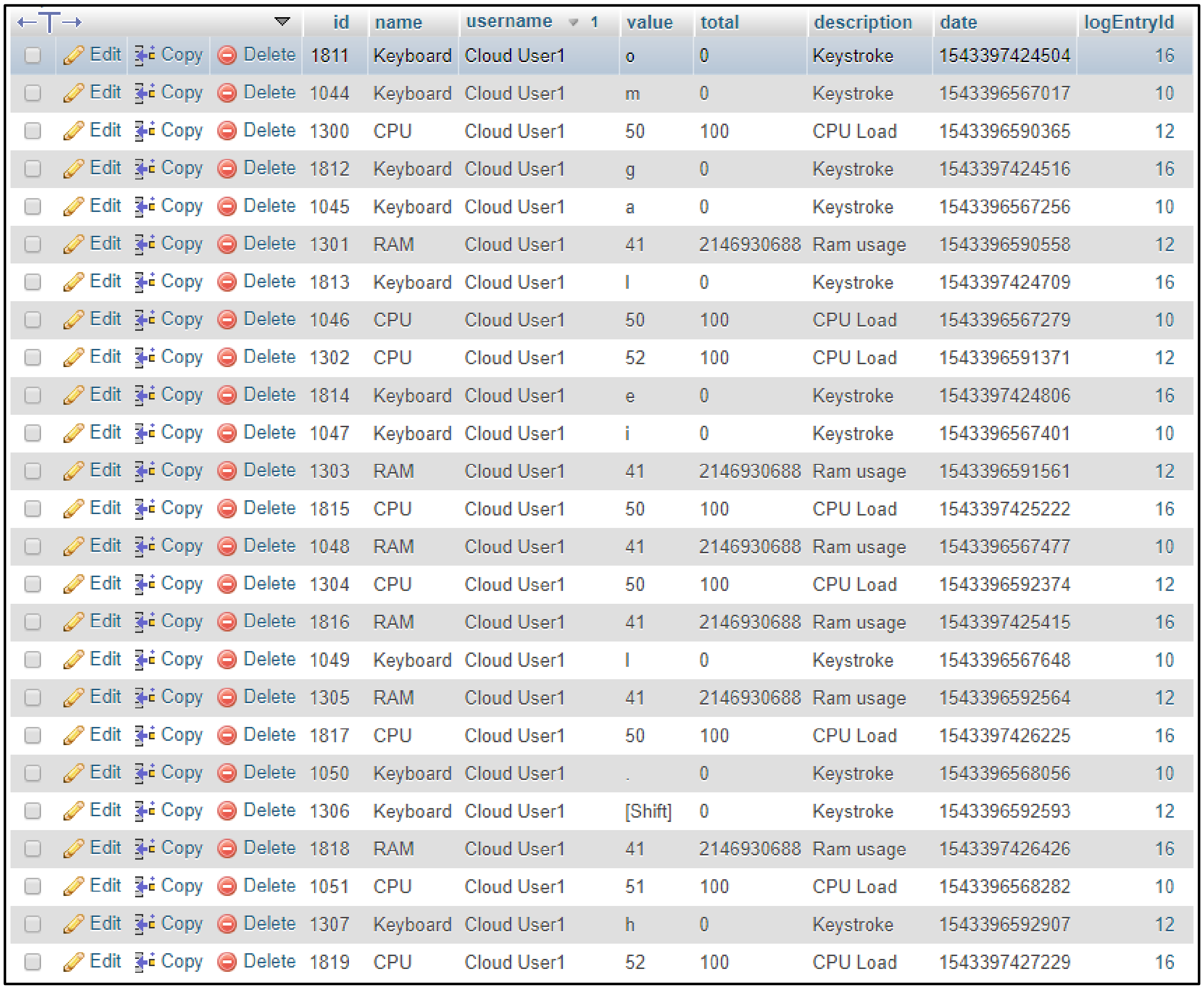
rawData	hash	timeReceived	ip
eyJkYXRhIjp7Im1hY2hpbmVVVUlEIjoiNGM1MTIzZWMtNmMyZS...	93711a99cc4ac1ccdbdec85065b8a124	2018-11-28 13:12:21	196.230.99.3
eyJkYXRhIjp7Im1hY2hpbmVVVUlEIjoiNGM1MTIzZWMtNmMyZS...	93de39caab9aee79cf8da55a473812e2	2018-11-28 12:37:31	196.230.99.2
eyJkYXRhIjp7Im1hY2hpbmVVVUlEIjoiNGM1MTIzZWMtNmMyZS...	cfc017a31f967c2b4605a8171f91f127	2018-11-28 13:38:37	196.230.99.4
eyJkYXRhIjp7Im1hY2hpbmVVVUlEIjoiMjVhODdmZGYtNDg3Ni...	f0100a6577bd301e61af728d35cfac89	2018-11-28 12:41:05	196.230.99.2
eyJkYXRhIjp7Im1hY2hpbmVVVUlEIjoiNGM1MTIzZWMtNmMyZS...	8c6754070926157c42ca87c538a0c412	2018-11-28 13:26:45	196.230.99.3
eyJkYXRhIjp7Im1hY2hpbmVVVUlEIjoiNGM1MTIzZWMtNmMyZS...	3bd1b347d8a91b63cbd34da8fbf4fbc7	2018-11-28 06:30:36	196.230.99.2
eyJkYXRhIjp7Im1hY2hpbmVVVUlEIjoiMjVhODdmZGYtNDg3Ni...	9870c782d3aa646e8920b75fecec80f9	2018-11-28 13:11:23	196.230.99.4
eyJkYXRhIjp7Im1hY2hpbmVVVUlEIjoiMjVhODdmZGYtNDg3Ni...	a9787097ca7710cbe5f8998c417420d1	2018-11-28 07:11:32	196.230.99.2
eyJkYXRhIjp7Im1hY2hpbmVVVUlEIjoiMjVhODdmZGYtNDg3Ni...	58b5bb97edd9894f97f5b7c1da3aeed3	2018-11-28 13:11:43	196.230.99.3

Figure 5.5. Raw data hashes

The ‘rawData’ column shows the captured digital information, at a particular time interval. Each of the chunks of digital information captured gets hashed and the hash is recorded as shown in the hash column.

5.5.3 Prototype harvest, preservation and storage

This process is where the prototype starts to acquire digital data from the cloud instance. The digital data collected by the prototype includes CPU usage, RAM usage and keystrokes on the keyboard. Figure 5.6 below shows the collected digital information as seen from the command and control server.



←T→			▼	id	name	username ▼ 1	value	total	description	date	logEntryId
□	Edit	Copy	Delete	1811	Keyboard	Cloud User1	o	0	Keystroke	1543397424504	16
□	Edit	Copy	Delete	1044	Keyboard	Cloud User1	m	0	Keystroke	1543396567017	10
□	Edit	Copy	Delete	1300	CPU	Cloud User1	50	100	CPU Load	1543396590365	12
□	Edit	Copy	Delete	1812	Keyboard	Cloud User1	g	0	Keystroke	1543397424516	16
□	Edit	Copy	Delete	1045	Keyboard	Cloud User1	a	0	Keystroke	1543396567256	10
□	Edit	Copy	Delete	1301	RAM	Cloud User1	41	2146930688	Ram usage	1543396590558	12
□	Edit	Copy	Delete	1813	Keyboard	Cloud User1	l	0	Keystroke	1543397424709	16
□	Edit	Copy	Delete	1046	CPU	Cloud User1	50	100	CPU Load	1543396567279	10
□	Edit	Copy	Delete	1302	CPU	Cloud User1	52	100	CPU Load	1543396591371	12
□	Edit	Copy	Delete	1814	Keyboard	Cloud User1	e	0	Keystroke	1543397424806	16
□	Edit	Copy	Delete	1047	Keyboard	Cloud User1	i	0	Keystroke	1543396567401	10
□	Edit	Copy	Delete	1303	RAM	Cloud User1	41	2146930688	Ram usage	1543396591561	12
□	Edit	Copy	Delete	1815	CPU	Cloud User1	50	100	CPU Load	1543397425222	16
□	Edit	Copy	Delete	1048	RAM	Cloud User1	41	2146930688	Ram usage	1543396567477	10
□	Edit	Copy	Delete	1304	CPU	Cloud User1	50	100	CPU Load	1543396592374	12
□	Edit	Copy	Delete	1816	RAM	Cloud User1	41	2146930688	Ram usage	1543397425415	16
□	Edit	Copy	Delete	1049	Keyboard	Cloud User1	l	0	Keystroke	1543396567648	10
□	Edit	Copy	Delete	1305	RAM	Cloud User1	41	2146930688	Ram usage	1543396592564	12
□	Edit	Copy	Delete	1817	CPU	Cloud User1	50	100	CPU Load	1543397426225	16
□	Edit	Copy	Delete	1050	Keyboard	Cloud User1	.	0	Keystroke	1543396568056	10
□	Edit	Copy	Delete	1306	Keyboard	Cloud User1	[Shift]	0	Keystroke	1543396592593	12
□	Edit	Copy	Delete	1818	RAM	Cloud User1	41	2146930688	Ram usage	1543397426426	16
□	Edit	Copy	Delete	1051	CPU	Cloud User1	51	100	CPU Load	1543396568282	10
□	Edit	Copy	Delete	1307	Keyboard	Cloud User1	h	0	Keystroke	1543396592907	12
□	Edit	Copy	Delete	1819	CPU	Cloud User1	52	100	CPU Load	1543397427229	16

Figure 5.6. Collected Digital information

The name column in Figure 5.6 gives a description of the type of digital information collected by the prototype. These are namely CPU, RAM and keystrokes. These get captured at a 2-second interval. Once captured, the ‘chunk’ of captured raw data is hashed and is sent to the command and control server through a PHP POST method to http host *logger.xp3.com* (see Figure 5.4) where the command and control server is hosted. A PHP function is responsible for translating the raw data (Figure 5.5) into the captured digital information. This information is stored in MySQL tables. The MySQL database also records the username of the particular cloud user logged in at the time the information was captured. The ‘total’ column shows the value captured.

The captured information can then be seen in a graphical diagram from the command and control server as shown below:

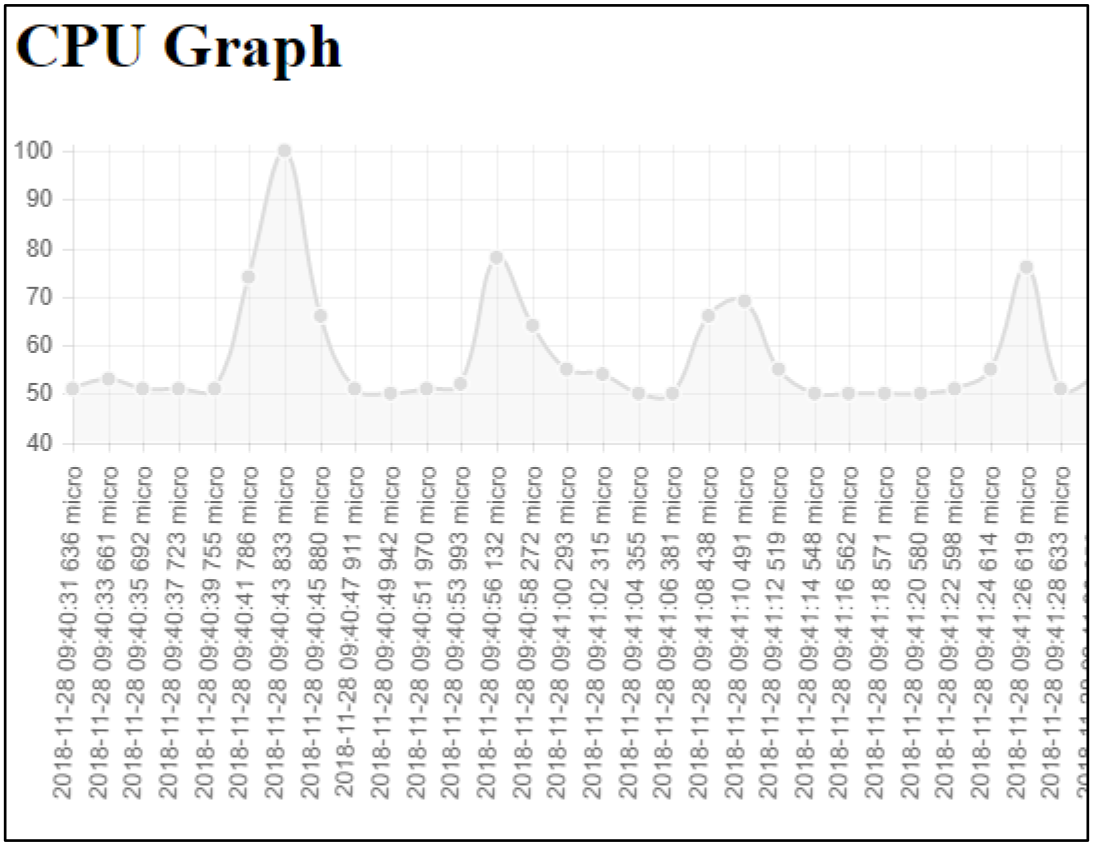


Figure 5.7. CPU graph showing the timestamp

The CPU Usage graph can be used to check where there was an increase in the CPU activity. The Y-axis shows the CPU percentage recorded and the X-axis shows the time stamp when the particular CPU percentage was recorded. Figure 5.8 below shows the RAM usage graph:

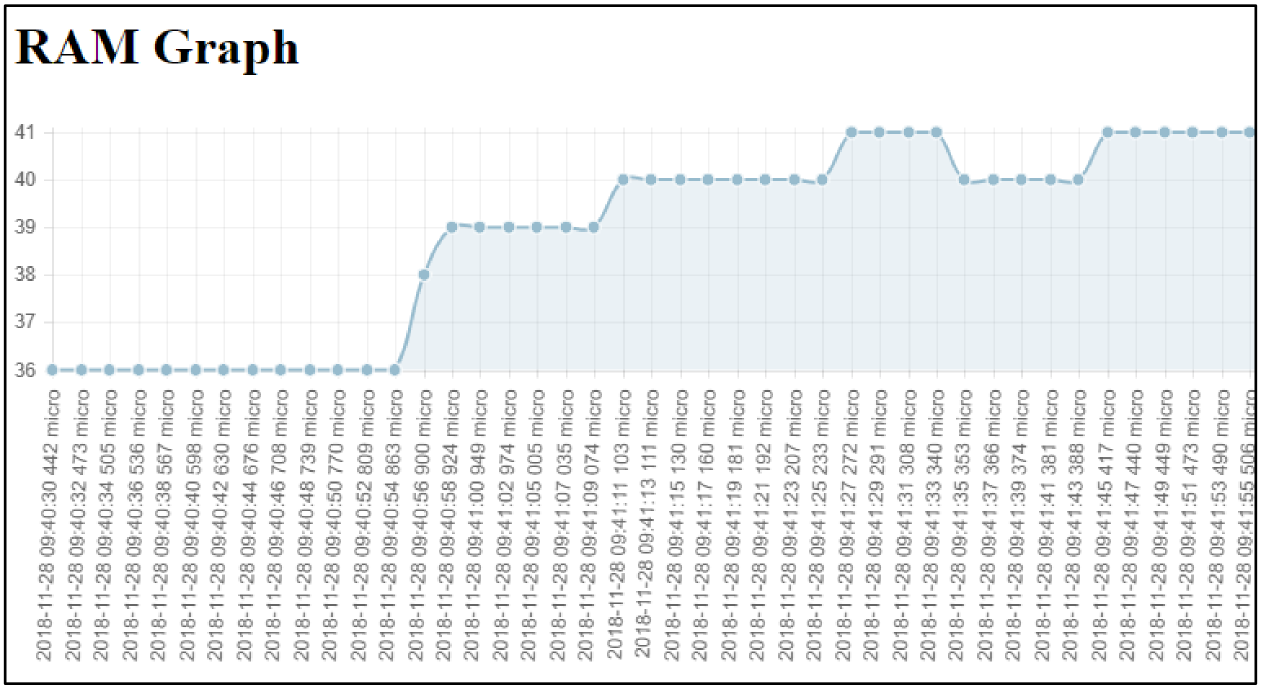


Figure 5.8. Ram Usage

The RAM usage graph can be used in forensic investigations. The usage graph can be used to trace when a malicious application was executed on a cloud instance. As with the CPU usage graph, the RAM usage graph shows the RAM usage percentage plotted against the timestamp consisting of the date and time. The range of time entries is captured at an interval of two seconds. Once the prototype has collected CPU usage, RAM usage and keystroke entries, it is stored and preserved in the MySQL database.

This presentation of the proof of concept leads to the evaluation of whether it will be possible to make use of the prototype in solving incidences that happen in the cloud environment. As a result, scenarios were performed in OpenStack cloud instances in order to experiment on this. These scenarios are described in the following section.

5.6 Scenarios

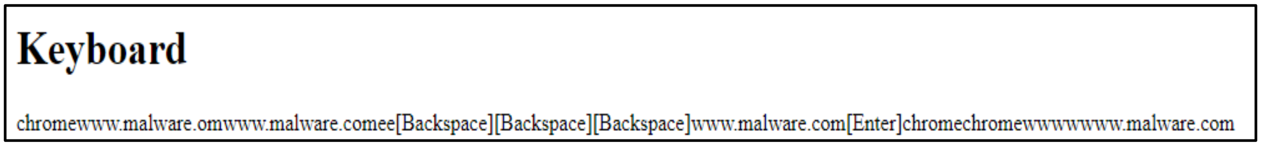
Scenarios of digital crimes that are linked to the cloud environment were performed. These scenarios were performed to determine the applicability of the prototype and how it can be used to identify and solve a digital crime. There were two scenarios investigated and both are described in the following sub-sections:

5.6.1 Case Scenario 1: Malware download and execution

In this scenario, a cloud user opens a web-browser on a cloud instance. The user browses a website and sees a pop-up ad which says, “the computer is infected by a virus” and that the user needs to download and install the antivirus software to remove the virus. The user clicks the pop-up ad and gets redirected to the website of the attacker. Once there, the user proceeds to download the software, which is actually a malware. The user installs the malware on the machine.

One might ask about how the prototype can be used to investigate this scenario. The answer to this question lies in the characteristics of the prototype and what it collects. To recall, the prototype collects namely, CPU and RAM usage, as well as the keystrokes the user enters on the keyboard. This information can prove critical in that it can tell us for instance, what the user entered on the keyboard.

This scenario was simulated in one of the cloud instances hosted on the OpenStack cloud environment. The results show that the CPU and RAM usage increases after the malware executed on the cloud instance. The keystrokes showed that the user typed in a malicious website and visited it. Figure 5.9 below shows the captured keystrokes the cloud user entered on the keyboard:

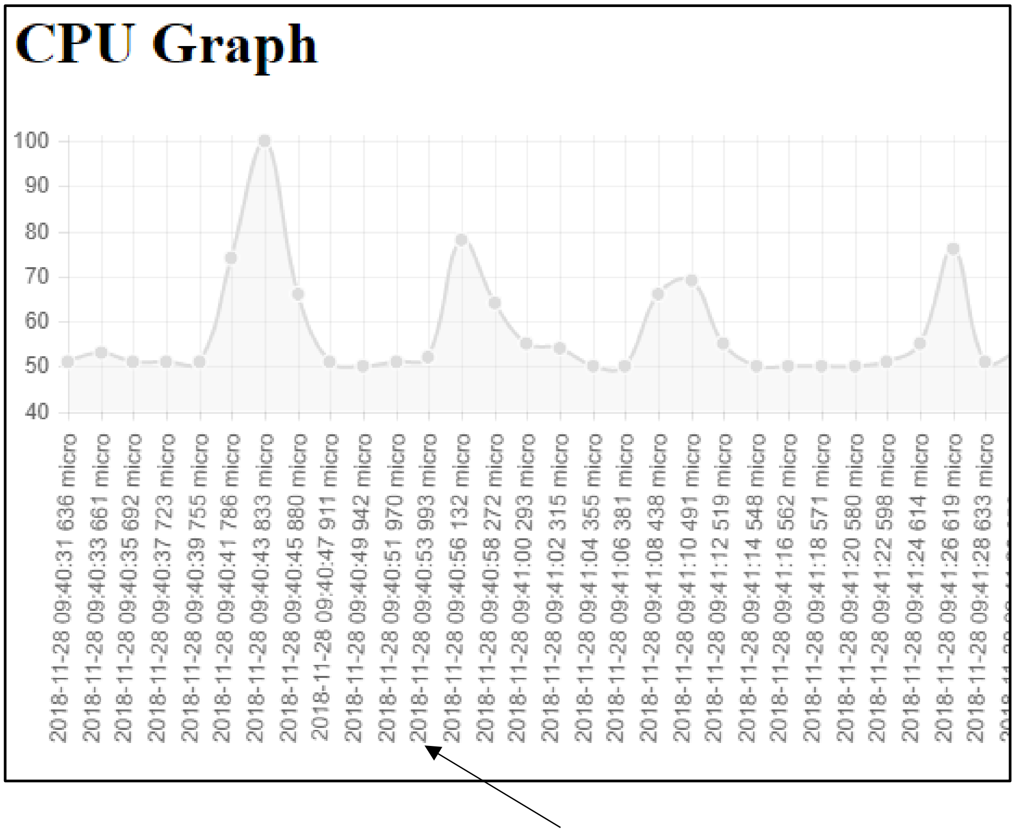


Keyboard

chromewww.malware.omwww.malware.comee[Backspace][Backspace][Backspace]www.malware.com[Enter]chromechromewwwwww.malware.com

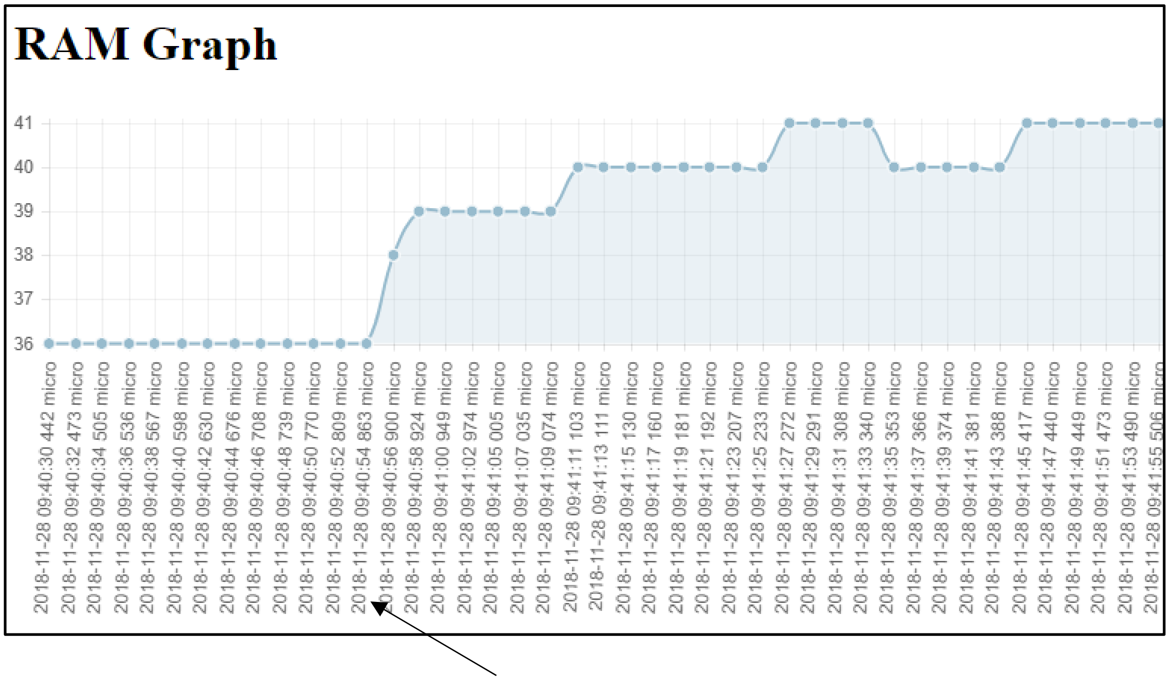
Figure 5.9. Captured keystrokes

The above Figure 5.9 shows that the user searched for the term Chrome, which was done in order to open the Google Chrome web browser. After opening the browser, the user, visited the site: *www.malware.com*. This is where the user sees the pop-up ad to download software to remove the virus on his/her PC, which in fact is malware from the attacker. The user unknowingly downloads and executes the malware. The execution of the malware on the cloud was followed by an increase in the CPU and RAM usage. These results are shown in Figure 5.10 and Figure 5.11 below.



Time stamp showing the point where CPU usage increased

Figure 5.10. CPU usage of user



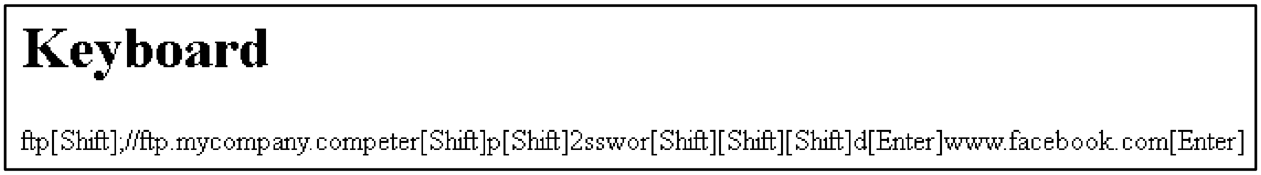
Time stamp showing the point where RAM usage increased

Figure 5.11. RAM usage graph

The observation from the RAM usage graph is that there was a sharp increase in the RAM usage at **09:40:54** time stamp (one with arrow). An analysis of the CPU usage graph confirms a corresponding increase in the CPU usage at **09:40:53** (one with arrow). This information can assist digital forensic investigators to narrow down the timeframe of malware execution within the cloud instance. They can then focus on the time preceding the increase in the CPU and RAM usage, and the time after the increase. The following sub-section discusses on the second scenario.

5.6.2 Case Scenario 2: Cloud user accessing FTP site

This second case scenario focuses on solving a case where an employee uses the credentials of the manager who has authorised access to retrieve confidential company information hosted on the company's FTP site. In this scenario, the company has rules and regulations, which forbids employees from accessing the FTP site. However, the company allows managers only to have access to the FTP site. The employees are also not allowed to access social media sites during the 8am to 4pm working hours. A cloud instance was setup in OpenStack and the prototype deployed on the cloud instance. Keyboard strokes entered by the user accessing the FTP site and keystrokes of the username and password employed by the user to gain unauthorised entry into the site were captured by the prototype. Figure 5.12 below shows the keystrokes recorded for the cloud user as observed from the command and control center:



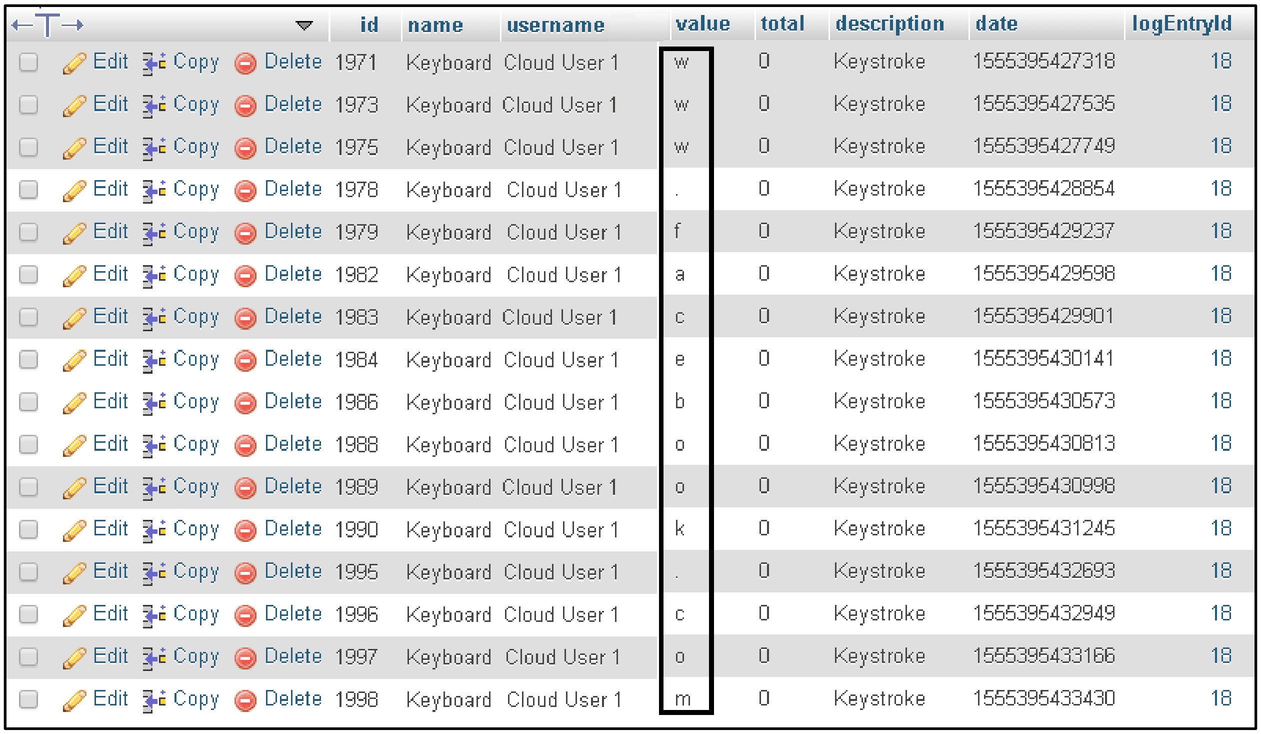
Keyboard

```
ftp[Shift];//ftp.mycompany.competer[Shift]p[Shift]2sswor[Shift][Shift][Shift]d[Enter]www.facebook.com[Enter]
```

Figure 5.12. Cloud User captured keystrokes

The observation from Figure 5.12 above, is that the user accessed the ftp site: ***ftp://ftp.mycompany.com***, entered the username: ***peter*** and password: ***P@ssworD***, which are the managers' credentials, and pressed enter to gain access to the site. After a few minutes, the user also accessed the social networking site ***facebook.com*** as shown in Figure 5.12. This information can prove to be vital in attempts at proving whether a cloud user accessed confidential company information without authorisation.

The fact that the cloud user used his/her manager's details to login onto the company's FTP site is proof that the user gained unauthorised access. The user also accessed Facebook, which flouts the company regulations that employees should not access social media sites during working hours (8am to 4pm). The time that the user accessed Facebook can be identified by checking the database from the command and control center to see the timestamps for the captured keystrokes. Figure 5.13 below shows the captured keystrokes as observed from the command and control center database.



←T→			▼	id	name	username	value	total	description	date	logEntryId
□	Edit	Copy	Delete	1971	Keyboard	Cloud User 1	w	0	Keystroke	1555395427318	18
□	Edit	Copy	Delete	1973	Keyboard	Cloud User 1	w	0	Keystroke	1555395427535	18
□	Edit	Copy	Delete	1975	Keyboard	Cloud User 1	w	0	Keystroke	1555395427749	18
□	Edit	Copy	Delete	1978	Keyboard	Cloud User 1	.	0	Keystroke	1555395428854	18
□	Edit	Copy	Delete	1979	Keyboard	Cloud User 1	f	0	Keystroke	1555395429237	18
□	Edit	Copy	Delete	1982	Keyboard	Cloud User 1	a	0	Keystroke	1555395429598	18
□	Edit	Copy	Delete	1983	Keyboard	Cloud User 1	c	0	Keystroke	1555395429901	18
□	Edit	Copy	Delete	1984	Keyboard	Cloud User 1	e	0	Keystroke	1555395430141	18
□	Edit	Copy	Delete	1986	Keyboard	Cloud User 1	b	0	Keystroke	1555395430573	18
□	Edit	Copy	Delete	1988	Keyboard	Cloud User 1	o	0	Keystroke	1555395430813	18
□	Edit	Copy	Delete	1989	Keyboard	Cloud User 1	o	0	Keystroke	1555395430998	18
□	Edit	Copy	Delete	1990	Keyboard	Cloud User 1	k	0	Keystroke	1555395431245	18
□	Edit	Copy	Delete	1995	Keyboard	Cloud User 1	.	0	Keystroke	1555395432693	18
□	Edit	Copy	Delete	1996	Keyboard	Cloud User 1	c	0	Keystroke	1555395432949	18
□	Edit	Copy	Delete	1997	Keyboard	Cloud User 1	o	0	Keystroke	1555395433166	18
□	Edit	Copy	Delete	1998	Keyboard	Cloud User 1	m	0	Keystroke	1555395433430	18

Figure 5.13. Cloud user 1 captured keystrokes as seen from the database

Figure 5.13 shows the results of the captured keystrokes from the cloud user 1. The name column describes the details of the item captured, which is the keyboard. The username column identifies the cloud user while the value column shows the keystrokes captured by the prototype. The observation from the figure is that the total value the user entered is ***www.facebook.com***. The date column shows the timestamp recorded for each keystroke captured. Timestamp is in milliseconds and its translation to a particular date and time. For example, the time when the "**f**" keystroke was captured with timestamp 1555395429237, would be: **Tuesday 16 April, 2019 08:17:09 GMT+02:00**

The time provides evidence that the user accessed the social media site during working hours.

This scenario shows how the prototype can be used to assist forensic investigators in finding out who accessed confidential company information. In addition, the prototype can be used by companies to monitor the sites visited by employees during office hours in order to guard against unproductivity and inactivity during office hours. The following section concludes the chapter.

5.7 Conclusion

The chapter highlighted the nature and characteristics of the experiments performed. The experiments sought to answer the research question posed in Chapter 1 of the research study. Section 5.2 provided an overview of the OpenStack operating system while Sections 5.3 and 5.4 provided an overview of how the experiments were set up. Further outlined, are the characteristics of the cloud instances setup in the OpenStack cloud environment. The chapter showed that the prototype was deployed and executed in these cloud instances.

The chapter's Section 5.5 described the proof of concept. The prototype retrieved digital information from the cloud instances set up in OpenStack and stored the PDE in a forensic database. Finally, Section 5.6 discussed the scenarios in an attempt to prove how the prototype can be used to solve incidences that happen in a cloud environment.

The following chapter focuses on the evaluation of the study research.

Chapter 6: Critical Evaluation

6.1 Introduction

The continued growth of cloud computing technologies witnessed over the years has led to cybercriminals making use of cloud computing as an environment to launch malicious attacks, and this necessitates the need for CSPs to implement proactive DFR processes to combat such security threats. These DFR processes seek to provide ways to collect digital information, in a cloud platform, which can be utilised in a digital forensic investigation. As a result, this research presents a way of attaining DFR in an operational cloud environment through the use of the DFECS, which is a modified structure of a botnet that is used as a forensic agent in a non-malicious format. The DFECS prototype, developed by Kebande et al. (2016), was tested in a simulated environment. Hence, this study tested a modified version of the DFECS prototype in an operational cloud environment in order to show that DFR can be attained in an operational cloud environment.

Chapter 5 of this research study focused on the implementation of the prototype in the OpenStack operational cloud environment. However, this chapter evaluates the prototype and in particular its usefulness in an operational cloud environment. The chapter also examines the extent to which the research objectives posed in chapter 1 were met.

The remainder of this chapter is structured as follows. Section 6.2 presents an evaluation of the prototype based on its deployment and use in OpenStack. In addition, Section 6.3 considers the research objectives identified in Chapter 1 while Section 6.4 concludes the chapter.

6.2 Prototype evaluation

The experiments performed in Chapter 5 sought to test the prototype developed by Kebande et al. (2016) in an operational cloud environment in order to prove attainability of DFR in an operational cloud environment. OpenStack provided an operational cloud environment to deploy the prototype. The conducted experiments successfully showed that the prototype can be implemented in an operational cloud environment thereby proving attainability of DFR in the cloud. The observation is that the prototype deployed to three cloud instances was capable of harvesting digital information in each of the cloud instances and forensically storing the digital data in a forensic database. The experiments showed that it is possible to deploy the prototype three cloud instances. Therefore, the prototype can be used by organisations that use cloud computing platforms to provide a DFR environment for their cloud computing platforms.

Kebande et al. (2016), note that their prototype complies with the digital investigative readiness processes stipulated in the ISO/IEC 27043 standard. The experiments sought to ensure that these processes were followed throughout the experimentation. Following these processes is essential to make sure that the acquired digital data is admissible in a court of law. In addition, the functionality of the cloud architecture was not changed and this had the advantage of reducing the cost.

The experiments also showed that a forensic investigator who notices a suspicious high usage of CPU and RAM than normal can isolate that particular cloud instance for further investigation. Depending on the size of organisational infrastructure you have, OpenStack can spawn more than 100 000 cloud instances given that enough resources are available (OpenStack, 2018). The prototype can be deployed to these cloud instances but it would mean increasing the size of the database to make it large enough to accommodate the amount of digital information that will be collected. The resources in this research study were limited, which is why only three cloud instances were set up. The prototype was successfully deployed to these three cloud instances and was able to collect digital information in all three cloud instances.

One of DFR's main advantage is that it decreases the cost of conducting a digital forensic investigation (Rowlingson, 2004). Now imagine if an organisation has about 100 cloud instances running on OpenStack without DFR put in place in that cloud environment. It would mean that the DFI would have to forensically image all of the 100 cloud instances and look at the images one by one. This will definitely increase the cost and the time to conduct the forensic investigation. However, a case where the organisation has DFR put in place would making use of such a prototype. The DFI will be able to isolate specific cloud instances where security incidences were identified and focus on those cloud instances alone. This reduces the time and the cost that will be incurred in conducting the forensic investigation (Rowlingson, 2004).

The following section provides an evaluation of the research objectives.

6.3 Evaluation of research objectives

Three research objectives were posed and these were to:

- **Explore literature review and the current state of DFR**
- **Implement the prototype in an operational cloud environment**
- **Analyse on the potential usefulness of data collected by the prototype**

The main objective of the research study was to implement the prototype in an operational cloud environment, namely OpenStack, and thereby prove attainability of DFR in the cloud. The prototype can then be used to make ready for security incidents that occur in the cloud. The research objectives mentioned above are critically evaluated below:

a) **Explore literature review and the current state of DFR –** This research objective sought to provide a background on digital forensics and DFR. Chapters 2 and 3 present the background on botnets, cloud computing, digital forensics and DFR. The background on botnets assisted in understanding matters related to the prototype implemented in Chapter 5, which consisted of botnet propagation techniques and the deployment of a botnet as a forensic agent to cloud instances hosted in OpenStack to collect digital information in an operational cloud environment. The prototype followed the Cloud Readiness as a Service (CFRaaS) model proposed by Kebande and Venter (2016). In addition, a background on cloud computing was necessary as it clarified on how best to deploy the prototype on an operational cloud environment. OpenStack was chosen to provide the operational cloud computing environment to test the prototype. The literature review on the current state of DFR in the cloud showed that there existed no formal methods for conducting DFR in the cloud without modifying the functionality of an existing cloud architecture (Kebande & Venter, 2014). The review revealed further that the ISO/IEC 27043 standard seeks to provide standards for conducting DFR though the standard does not focus on the cloud environment specifically. Finally, the review indicated that the prototype developed by Kebande et al. (2016), sought to provide a way of conducting DFR in a simulated environment. The prototype was however, never tested in an operational cloud environment, which became the main motivation for this research study.

b) **Implement the prototype in an operational cloud environment**

The DFECS proposed by Kebande et al. (2016), was implemented in a simulated environment. However, it was never tested on a real operational cloud environment. This research objective sought to test the prototype in an operational cloud environment. The prototype, whose architecture was discussed in Chapter 4, employs the functionality of botnet. However, the prototype functionalities are changed so as to acquire digital data from an operational cloud environment in a non-malicious manner. Chapter 5 of this research study showed how the prototype was implemented in the OpenStack cloud environment. The prototype was able to harvest digital information from all three cloud instances hosted on the OpenStack infrastructure. The collected information was hashed and kept in a forensic database.

c) **Analyse on the potential usefulness of data collected by the prototype**

This research objective sought to find out if the integrity of the collected digital data is maintained. The prototype is able to collect raw data and hash it. The hashes of the collected data can be viewed from the command and control server side. The maintenance of hashing is important because the evidence can only be admissible in a court of law if the hashes match. To explain, it is important that the hash of the digital information acquired during collection be the same with the hash of the data stored in the database. This research objective also sought to determine if the collected data could prove useful in investigating cloud incidents. Two case scenarios were outlined in Chapter 5. In the first case scenario, collected data on CPU usage, RAM usage and keyboard strokes was used to investigate cloud incidents after a cloud user had downloaded a malware from a malicious site. Keystrokes of the malicious site visited by the cloud user were captured. The observation was that the malware execution on the cloud instance, led to an increase in the CPU and RAM usage. Hence, digital forensic investigators can use this information to check the timeframe where the CPU and RAM usage spiked and focus their attention on that identified timeframe.

In the second case scenario, keystrokes from the keyboard of a cloud user were captured by the prototype in an attempt at solving the case where an employee accessed confidential company information hosted on the company's FTP site. Keystrokes of the user accessing the FTP site and keystrokes of the username and password the user used to gain unauthorised entry to the site were captured by the prototype. The scenario also showed that the prototype can be used to monitor employees of the sites they visit during office hours and thus guard against non-productivity and inactivity during office hours.

It is thus evident from the critically evaluated research objectives above that, they were met in full. The following section concludes the chapter.

6.4 Conclusion

This chapter focused on a critical evaluation of the research study. It initially paid attention on the prototype in Section 6.2 and then went on to evaluate the study's research objectives in Section 6.3. The chapter further evaluated the extent to which the research objectives were met and noted that all research objectives mentioned in Chapter 1 were met in full.

The following chapter outlines the final conclusion to the research work.

Chapter 7: Conclusion

7.1 Introduction

The increase in cybercrimes in cloud environments raised the need to develop a proactive approach to dealing with the incidences that happen in cloud environments. DFR processes can be used to prepare for these security incidences. As a result, clients of CSPs should understand the significance of having DFR processes for their cloud environments. The prototype implemented in this research study, was meant as a proof of concept that DFR can be attained in an operational cloud environment.

Chapter 1 presented the problem statement that the research study sought to address. This chapter provides the reader with the concluding remarks to the research study and also the extent to which the problem statement has been addressed by the research study. The chapter is structured as follows: Section 7.2 revisits the problem statement stated in Chapter 1 of this research study while Section 7.3 concludes the research study and suggests future work.

7.2 Revisiting of the problem statement

The study's main problem relates to the non-existence of a novel approach to attaining DFR in an operational cloud environment. Kebande et al. (2016), developed the DFECS, a software application possessing functionality of a modified version of a botnet, and implemented it in a simulated environment, in order to address this problem. The prototype is capable of collecting digital information from a simulated environment in a forensic manner by hashing the collected data and storing the data in a forensic database for DFR purposes (Kebande et al. 2016). DFECS follows the readiness processes stipulated in the ISO/IEC27043: 2015 standard. Nevertheless, DFECS was not tested in an operational cloud environment. Therefore, this research study aimed to implement DFECS in an operational cloud environment, namely OpenStack, and prove the attainability of DFR in an operational cloud environment.

The advantages of proving attainability of DFR in an operational cloud environment are that it maximises the use of the collected PDE and reduces the time and cost needed in performing a DFI (Rowlingson, 2004). The prototype was successfully implemented in OpenStack and collected digital data that can be used in a digital forensic investigation.

Experiments performed in Chapter 5 showed how the prototype can be used to solve security incidents that occur within an operational cloud environment. The acquired digital data also proved useful in that it can assist in the identification of security incidents. However, it is worth mentioning that the cloud architecture was not altered based on the execution of the prototype.

Two sub problems were presented and they are:

(i) **How can DFECS be implemented in an operational cloud environment**?

(ii) **What are the impacts and the usefulness of the collected digital data?**

The sub-problems sought ways to implement the DFECS prototype in an operational cloud environment. OpenStack was used to provide an operational cloud environment with three cloud instances setup in OpenStack and DFECS deployed into these cloud instances. The results showed that it was possible to implement the prototype in an operational cloud environment. DFECS was capable of harvesting digital data from the cloud instances, hash the collected data, and then forensically store it in a database.

The second sub problem sought to check if the collected digital data could be useful in solving security incidents that occur in the cloud environment. Two scenarios tested this with the results showing that the collected digital information can assist forensic investigators in investigating security incidents in an operational cloud environment.

The following section provides a conclusion to the research study and discusses and discusses future work that can be performed.

7.3 Conclusion and Future Work

The study presented a way in which the DFECS prototype developed by Kebande et al. (2016) can be implemented in an operational cloud environment. The DFECS is able to gather digital data in a proactive manner for DFR purposes (Kebande et al. 2016). However, Kebande et al. (2016) never tested the prototype in an operational cloud environment. The major contribution of this research study was therefore to implement the prototype in an operational cloud environment for DFR purposes. Implementation of the prototype was a success and the results of the implementation presented in Chapter 5. The study concluded that the utilisation of this prototype can help organisations that implement DFR processes in their cloud environments as it maximises the use of PDE whilst reducing the cost of conducting DFI in the cloud.

There has been an increase in the use of cloud computing technologies by organisations due to the fact that cloud computing carries a lot of advantages like reduction in IT costs, scalability and business continuity. This study's background chapters elaborated on this observation. Furthermore, the use of cloud computing technologies has also led to an increase of cyber-security incidents that target cloud computing environments. Hence, organisations that make use of cloud computing technologies need to have proper DFR standards in order to deal with these incidents. This research study noted that there were no DFR standards focusing on the cloud environment. Therefore, this researcher suggests that more research be performed in this area in the future work to develop DFR standards that focus specifically on cloud environments.

The prototype mentioned in this research study can harvest digital information in an operational cloud environment. The harvested digital information might be later used as evidence in a court of law. Therefore, the study noted the significance of the legal aspects that affect the gathering of digital information from cloud environments, as shown in Chapter 3 of this research study. The researcher is of the opinion that more future work is required in the legal field to ensure that collected digital information is admissible in a court of law. It is also necessary to ensure that digital forensic investigators as well as CSPs are aware of the applicable legal laws, such as the PoPI (PoPI, 2013) and the ECT (ECT, 2002) Acts, which govern the collection of digital information.

The researcher also notes that there is need to improve the prototype so that it can collect more digital information. The current DFECS is only limited to collecting keystrokes, CPU and RAM usage. Hence, the functionality of the prototype can be expanded to enable it to collect network logs, event logs and other important digital information that can prove useful in reducing the cost and time spent in conducting DFI in an operational cloud environment.

References

Agarwal, A., Gupta, M., Gupta, S., & Gupta, S. C. (2011). Systematic digital forensic investigation model. *International Journal of Computer Science and Security (IJCSS)*, *5(1),* 118-131.

Akervik, T. (2019). What are the changes of losing information in cloud storage?, accessed 20 February 2019, https://blog.marconet.com/blog/what-are-the-chances-of-losing-information-in-cloud-storage

Avram, M. G. (2014). Advantages and challenges of adopting cloud computing from an enterprise perspective. *Procedia Technology*, *12*(0), 529-534.

Barbara, J. J. (2009). Cloud computing: Another digital forensic challenge. *Digital Forensic Investigator News*.

Beebe, N. L., & Clark, J. G. (2005). A hierarchical, objectives-based framework for the digital investigations process. *Digital Investigation, 2(2), 147-167*.

Bromiley, M. (2016). Threat intelligence: What it is, and how to use it effectively. *SANS Institute InfoSec Reading Room, 15*.

Brungs, A., & Jamieson, R. (2005). Identification of legal issues for computer forensics. *Information Systems Management, 22(2)*, 57-66.

Carrier, B., & Spafford, E. H., (2004). An event-based digital forensic investigation framework. *In Digital forensic research workshop (pp. 11-13)*.

Casey, E. (2011). Digital evidence and computer crime: *Forensic science, computers, and the Internet*. Academic press.

Chen, Z. N., Chen, K., Jiang, J. L., Zhang, L. F., Wu, S., Qi, Z. W., ... & Sun, A. B. (2017). Evolution of cloud operating system: from technology to ecosystem. *Journal of Computer Science and Technology*, 32(2), 224-241.

Cohen, F. B. (2012). Digital forensic evidence examination. *Livermore: Fred Cohen & Associates.*

De Marco, L., Ferrucci, F., & Kechadi, T. (2014). Reference Architecture for a Cloud Forensic Readiness System. *EAI Endorsed Transactions on Security and Safety*, pp. 1-9

Deshmukh, R.V. & Devadkar, K.K., (2015). Understanding DDoS attack & its effect in cloud environment. *Procedia Computer Science*, *49*, pp.202-210.

DFRWS. (2001). A road map for digital forensics research-report from the first *Digital Forensics Research Workshop* (DFRWS). Utica, New York.

Dykstra, J., & Sherman, A. T. (2012). Acquiring forensic evidence from infrastructure-as-a-service cloud computing: Exploring and evaluating tools, trust, and techniques. *Digital Investigation, 9, S90-S98*.

Fisher, D. (2013). What is a Botnet?, accessed 11 October 2018, https://www.kaspersky.com/blog/botnet/1742/

Goyal, S. (2014). Public vs private vs hybrid vs community-cloud computing: a critical review. *International Journal of Computer Network and Information Security, 6(3), 20*.

Hay, B., Nance, K. & Bishop, M., (2011). Storm clouds rising: security challenges for IaaS cloud computing. In *System Sciences (HICSS), 2011 44th Hawaii International Conference on* (pp. 1-7). IEEE.

Haylee, (2017). Botnets: Dawn of the connected dead, accessed 10 December 2018, https://blog.emsisoft.com/en/27233/what-is-a-botnet/

ISO/IEC 27043, (2015)- Information technology -- Security techniques -- Incident investigation principles and processes, accessed 10 December 2018 https://www.iso.org/standard/44407.html

Jang-Jaccard, J., & Nepal, S. (2014). A survey of emerging threats in cybersecurity. *Journal of Computer and System Sciences, 80(5)*, 973-993.

Jensen, M., Schwenk, J., Gruschka, N., & Iacono, L. L. (2009). On technical security issues in cloud computing. In *2009 IEEE International Conference on Cloud Computing* (pp. 109-116) IEEE.

Kaufman, L. M. (2009). Data security in the world of cloud computing. *IEEE Security & Privacy, 7(4)*, 61-64.

Kebande V. R., Ntsamo, H. S., & Venter, H.S., (2016). Towards a Prototype for Achieving Digital Forensic Readiness in the Cloud using a Distributed NMB Solution, In *The European Conference of Cyber Warfare and Security*, Bundeswar University, Munich, Germany.

Kebande V. R., & Venter H.S., (2016). Architectural Design of a Cloud Forensic Readiness as a Service (CFRaaS) System Using an NMB Solution as a Forensic Agent, *International Journal of Information and Computer Security*. Inderscience

Kebande V.R., & Venter, H.S., (2014) A Cloud Forensic Readiness Model Using a Botnet as a Service, In *The International Conference Digital Forensic and Security*, Czech Republic, 2014.

Kebande V.R., & Venter, H.S., (2015). Obfuscating a Cloud-Based Botnet Towards Digital Forensic Readiness. In *Iccws 2015- The Proceedings of the 10th International Conference on Cyber Warfare and Security* (p. 434). Academic Conferences

Khajeh-Hosseini, A., Sommerville, I., & Sriram, I. (2010). Research challenges for enterprise cloud computing. *arXiv preprint arXiv:1001.3257*.

Krutz, R. R., & Vines, R. (2010). Cloud Security - A Comprehensive Guide to Secure Cloud Computing. New York City, NY: Wiley.

Kurup, L.D, Chandawalla, C, Parekh, Z & Sampat, K (2015). Comparative Study of Eucalyptus, OpenStack and Nimbus. I*nternational Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-4 Issue-6*, PP 23-27

Marston, S., Li, Z., Bandyopadhyay, S., Zhang, J. & Ghalsasi, A., (2011). Cloud computing—The business perspective. *Decision support systems*, 51(1), pp.176-189.

Mell, P., & Grance, T. (2011). The NIST definition of cloud computing.

Meyer, G., & Stander, A. (2015). Cloud computing: The digital forensics challenge. In *Proceedings of Informing Science & IT Education Conference (InSITE)* (p. 286).

Mónica, D., & Ribeiro, C. (2013). Leveraging honest users: Stealth command-and-control of botnets. In *The 7th {USENIX} Workshop on Offensive Technologies*.

OpenStack, (2018). OpenStack, accessed 10 December 2018, https://www.openstack.org/

Palmer, G. (2001). A road map for digital forensic research. In *First Digital Forensic Research Workshop, Utica, New York* (pp. 27-30).

Parker, D. B., & Parker, D. B. (1976). *Crime by computer (pp. xii-xii)*. New York: Scribner.

PC Pitstop, (n.d), accessed 10 October 2018, http://www.pcpitstop.com/images/affiliates/500755.gif

Pollitt, M. (2010). A history of digital forensics. In IFIP *International Conference on Digital Forensics* (pp. 3-15). Springer, Berlin, Heidelberg.

Popović, K., & Hocenski, Ž. (2015). Cloud computing security issues and challenges. In *The 33rd International Convention MIPRO* (pp. 344-349). IEEE.

Reith, M., Carr, C., & Gunsch, G. (2002). An examination of digital forensic models. *International Journal of Digital Evidence*, 1(3), 1-12.

Rowlingson, R. (2004). A ten-step process for forensic readiness. *International Journal of Digital Evidence*, *2*(3), 1-28.

Sam Solutions, (2017). Top 4 Cloud Deployment Models You Need to Know, accessed 12 October 2018, https://www.sam-solutions.com/blog/four-best-cloud-deployment-models-you-need-to-know/

Santos, T. & Serrão, C, (2016). Secure Javascript Object Notation (SecJSON) Enabling granular confidentiality and integrity of JSON documents. Barcelona, 2016 11th *International Conference for Internet Technology and Secured Transactions* (ICITST).

Sen, J. (2015). Security and privacy issues in cloud computing. In *Cloud Technology: Concepts, Methodologies, Tools, and Applications* (pp. 1585-1630). IGI Global.

Stoll, C. (1990). The Cuckoo's Egg: *Tracking a Spy Through the Maze of Computer Espionage*, New York, Pocket Books.

Subashini, S. & Kavitha, V, (2011). A survey on security issues in service delivery models of cloud computing. *Journal of network and computer applications*, *34*(1), pp.1-11.

Tan, J. (2001). Forensic readiness. *Cambridge, MA:@ Stake*, 1-23.

Taylor, M., Haggerty, J., Gresty, D. & Lamb, D., (2011). Forensic investigation of cloud computing systems. *Network Security*, *2011*(3), pp.4-10.

Techno-Pulse, (2011). Cloud Deployment Models – Private, Community, Public, Hybrid with Examples, accessed 12 October 2018, https://www.techno-pulse.com/2011/10/cloud-deployment-private-public-example.html

The Electronic Communications and Transactions (ECT) Act, Act 25 of 2002 (2002).

The Protection of Personal Information Act (POPI) (2013) Vol 581 No 4.

Valjarevic, A., & Venter, H. S. (2012). Harmonised digital forensic investigation process model. In *2012 Information Security for South Africa* (pp. 1-10). IEEE.

Van Staden, F., & Venter, H. (2012). Implementing Forensic Readiness Using Performance Monitoring Tools. In IFIP *International Conference on Digital Forensics (pp. 261-270)*. Springer, Berlin, Heidelberg.

Wilson, D (2015). Legal Issues with Cloud Forensics, accessed 12 October 2018, https://www.forensicmag.com/article/2015/05/legal-issues-cloud-forensics

Xie, Y., Yu, F., Achan, K., Panigrahy, R., Hulten, G., & Osipkov, I. (2008). Spamming botnets: signatures and characteristics. *ACM SIGCOMM Computer Communication Review, 38(4)*, 171-182.

Yadav, S. (2013) Comparative Study on Open Source Software for Cloud Computing Platform: Eucalyptus, OpenStack and OpenNebula. Research Inventy: *International Journal of Engineering And Science Vol.3, Issue 10 (October 2013)*, PP 51-54

Zhang, Q., Cheng, L., & Boutaba, R. (2010). Cloud computing: state-of-the-art and research challenges. *Journal of Internet services and applications*, *1*(1), 7-18.