

A Novel Cloud Forensic Readiness Service Model

by

Victor Rigworo Kebande

Submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy (Ph.D.)

In the subject of Computer Science

at

Faculty of Engineering, Built-Environment and Information Technology at the Department of
Computer Science, University of Pretoria, Republic of South Africa



University of Pretoria

September, 2017

Supervisor: Prof Hein S. Venter

© Copyright by Victor Rigworo Kebande, 2017

UNIVERSITY OF PRETORIA

Department of Computer Science

The undersigned hereby certify that they have read and recommend to the Faculty Engineering, Built Environment and IT for acceptance a thesis entitled “**A Novel Cloud Forensic Readiness Service Model**” by **Victor Rigworo KEBANDE** in fulfilment of the requirements for the degree of **Doctor of Philosophy in Computer Science**.

Dated: September 2017

Supervisor: Prof Hein S. Venter

.....

Readers

.....

UNIVERSITY OF PRETORIA

DATE: **April 2018**

AUTHOR: **Victor Rigworo KEBANDE**

TITLE: **A Novel Cloud Forensic Readiness Service Model**

DEPARTMENT: **Computer Science**

DEGREE: **Doctor of Philosophy (Ph.D.)** CONVOCATION: **April** YEAR: **2018**

Permission is herewith granted to **University of Pretoria** to circulate and to have copied for non-commercial purposes, at its discretion, the above title upon the request of individuals or institutions. I understand that my thesis will be electronically available to the public.

The author reserves other publication rights and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

The author attests that permission has been obtained for the use of any copyrighted material appearing in the thesis [other than the brief excerpts requiring only proper acknowledgement in scholarly writing], and that all such use is clearly acknowledged.



Signature of Author

DEDICATION

To all the special people in my life.

TABLE OF CONTENTS

LIST OF TABLES	xiii
LIST OF FIGURES	xiv
LIST OF EQUATIONS.....	xvi
ABSTRACT.....	xviii
LIST OF ABBREVIATIONS USED	xx
ACKNOWLEDGEMENTS.....	xxiv
Chapter 1: Introduction	1
1.1 Introduction.....	1
1.2 Problem Statement.....	3
1.3 Motivation	6
1.4 Research Objectives	7
1.5 Methodology	8
1.6 Thesis Layout	10
1.6.1 Part One: Introduction	10
1.6.2 Part Two: Background.....	10
1.6.3 Part Three: Model.....	11
1.6.4 Part Four: Prototype	11
1.6.5 Part Five: Conclusion	11
1.7 Conclusion	12
Chapter 2: Digital Forensics	14
2.1 Introduction.....	14
2.2 Forensics as a Science	15
2.3 Definition of Digital Forensics.....	15
2.4 Digital Evidence	17
2.5 Legal Requirements for Admissibility of Digital Evidence.....	18
2.5.1 Legal Governance and Consideration to Evidence Collection and Monitoring	19
2.5.2 Requirements for Admissibility of Digital Evidence	21
2.5.2.1 Case Law Review: Daubert v.Merrell Dow Pharmaceuticals	22
2.5.2.2 Case Law Review: United States v.Mosley.....	23
2.6 Digital Forensic Investigations.....	23
2.7 Digital Forensic Investigation Process	24
2.7.1 Initialisation Process Class.....	26
2.7.2 Acquisitive Process Class	26

2.7.3 Investigative Process Class	26
2.7.4 Concurrent Process Class	27
2.8 Digital Forensic Investigation Process Models	27
2.9 Digital Forensic Readiness	29
2.9.1 Defining Digital Forensic Readiness.....	30
2.9.2 Goals of Digital Forensic Readiness.....	31
2.10 Digital Forensic Readiness Process Class	32
2.11 Digital Forensic Readiness Process Group	32
2.12 Cost Benefits of DFR in an Organisation	33
2.13 Conclusion	34
Chapter 3: Concepts of Cloud Computing	35
3.1 Introduction.....	35
3.2 Defining Cloud Computing	36
3.3 Cloud Computing Architecture.....	37
3.4 Cloud Computing Service Models.....	40
3.4.1 Infrastructure as a Service (IaaS)	41
3.4.2 Platform as a Service (PaaS).....	41
3.4.3 Software as a Service (SaaS).....	41
3.5 Cloud Deployable Models.....	42
3.5.1 Public Cloud.....	42
3.5.2 Private Cloud	42
3.5.3 Community Cloud.....	43
3.5.4 Hybrid Cloud.....	43
3.6 Role of Cloud Service Provider	44
3.7 Virtualisation and Cloud Computing	45
3.8 Adoption of Digital Forensics Readiness in the Cloud	46
3.9 Digital Forensic Evidence Collection from the Cloud	48
3.10 Conclusion	48
Chapter 4: Botnets	50
4.1 Introduction.....	50
4.2 Definition of a Botnet.....	51
4.3 Life-Cycle of a Botnet.....	52
4.4 Anatomy of Botnets	53
4.4.1 Centralised C&C Architecture	54
4.4.2 Decentralised C&C Architecture	55
4.5 Botnet Control and Administration	56

4.5.1	Internet Relay Chat (IRC).....	56
4.5.2	HTTP	57
4.5.3	P2P.....	58
4.6	Usage of Botnets	58
4.6.1	Cyber-Extortion.....	58
4.6.2	Traffic Sniffing	59
4.6.3	Key-Logging	59
4.6.4	Spam.....	59
4.6.5	Distributed Denial of Service(DDoS)	60
4.7	Botnet as Cloud Attack Vector.....	60
4.8	Conclusion.....	61
Chapter 5: Requirements of a Cloud Forensic Readiness Service Model		63
5.1	Introduction.....	63
5.2	Need for Model Requirements	64
5.3	Model Requirements for Achieving DFR in the Cloud.....	65
5.3.1	General Requirements.....	65
5.3.1.1	Organisation requirement	67
5.3.1.2	Digital forensic governance.....	67
5.3.1.3	Forensic Logging Capability and Management.....	68
5.3.1.4	Digital Preservation.....	69
5.3.1.5	Timestamping	70
5.3.1.6	Digital Evidence Characterisation	71
5.3.1.7	Non-Modification of Existing Cloud Architecture	71
5.3.1.8	Security Implementation	72
5.3.1.9	Obfuscation	72
5.3.1.10	Event Reconstruction	75
5.3.1.11	Constitutional, Legal and Statutory Provisions	76
5.3.1.12	Forensic Readiness Reporting	77
5.3.2	Architectural Requirements.....	77
5.3.2.1	Functional Requirements (FRs).....	78
5.3.2.1.1	Standard Implementation of DFR in the Cloud	78
5.3.2.1.2	Collaborative with Competent Legal Bodies	79
5.3.2.1.3	Incident Response Procedures.....	79
5.3.2.1.4	Effectiveness and Ease of Use	79
5.3.2.2	Non-Functional Requirements (NFR)	80
5.3.2.2.1	Scalability.....	80

5.3.2.2.2	Security	80
5.3.2.2.3	Usability	80
5.3.2.2.4	Flexibility.....	81
5.3.2.2.5	Auditability	81
5.4	Conclusion.....	81
Chapter 6: Hypothetical Case Scenarios		83
6.1	Introduction.....	83
6.2	Hypothetical Investigative Case Scenarios	84
6.2.1	Case Scenario I: Information Security Breach and Identity Theft.....	84
6.2.2	Case Scenario II: Intrusion, Information Theft, Information Tampering and Framing	86
6.2.3	Case Scenario III: Sexual harassment, Child Pornography and Framing	88
6.3	Conclusion.....	90
Chapter 7: Cloud Forensic Readiness as a Service (CFRaaS) Model		92
7.1	Introduction.....	92
7.2	Formalisation of the Cloud Model	93
7.3	Formalisation of the Cloud Architecture	94
7.4	High-Level Overview of the CFRaaS Model	97
7.5	Detailed CFRaaS Model	99
7.5.1	Provider Layer	99
7.5.2	Virtualisation Layer.....	100
7.5.3	Digital Forensic Readiness Layer.....	101
7.5.3.1	Forensic Readiness Policy	103
7.5.3.2	CFRaaS Approach Strategy	104
7.5.3.2.1	Planning and Preparation.....	104
7.5.3.2.2	Scenario Identification	106
7.5.3.2.3	NMB Deployment.....	108
7.5.3.3	Digital Evidence Collection	109
7.5.3.3.1	Digital Evidence Collection Requirements (1)	110
7.5.3.3.2	Bot Client Infection and Digital Evidence Capture (2)	110
7.5.3.3.3	Digital Preservation and Forensic Database (3&4).....	111
7.5.3.4	Discussion based on hypothetical case scenarios	111
7.5.3.5	Pre-Incident Analysis	115
7.5.3.6	Incident Detection	118
7.5.3.7	Event Reconstruction	122
7.5.3.7.1	Retrieval of Relevant PDE	122

7.5.3.7.2	Location of Relevant PDE in Fields	123
7.5.3.7.3	Searching PSEs from Fields.....	125
7.5.3.7.3.1	Checking Event Similarity Measure (ESM)	125
7.5.3.7.3.2	Testing ESM Using Non-Negative Attributes	127
7.5.3.7.4	Evaluation of ESM and Finding	128
7.5.3.8	Forensic Readiness Report	130
7.5.4	Incident Response Procedure Layer	131
7.5.5	Concurrent Processes.....	132
7.6	Comparing the CFRaaS Model with Existing Readiness Models.....	133
7.7	Conclusion.....	139
Chapter 8: CFRaaS Prototype Design		141
8.1	Introduction.....	141
8.2	CFRaaS Prototype Overview.....	142
8.3	CFRaaS Prototype Main Components.....	142
8.4	Design of the CFRaaS Prototype	145
8.4.1	Technical Goals	146
8.4.2	Deployment and Virtual Environment.....	146
8.4.3	Technical Specifications.....	148
8.4.3.1	Programming language	148
8.4.3.2	Platform	148
8.4.3.3	Database.....	149
8.4.3.4	Integrity Checking Tool	149
8.4.4	CFRaaS Prototype Operation	150
8.4.4.1	High-Level NMB Process	150
8.4.4.2	Detailed NMB Process.....	151
8.4.4.2.1	Main Thread (A)	153
8.4.4.2.2	Connection Listener Thread (B)	156
8.4.4.2.3	Installer Thread (C)	157
8.4.4.2.4	Ping Server Thread(D)	159
8.4.4.2.5	Evidence Capture Thread (E)	161
8.4.4.2.6	Update Thread (F)	163
8.4.4.2.7	Destroy Thread (G).....	165
8.4.4.2.8	Send Data Thread (H)	166
8.5	Conclusion	169
Chapter 9: CFRaaS Prototype Implementation		170
9.1	Introduction.....	170

9.2	CFRaaS Prototype Set-up	171
9.3	Experiment to Identify Intrusion, Theft of Personal Information and Framing among organisations.....	172
9.3.1	Motivation	172
9.3.2	Experiment Purpose & Hypothetical Scenarios Used	173
9.3.3	Execution of the Experiment	174
9.3.4	Approach Strategy adopted in the Experiment	174
9.3.4.1	Main and Connection Listener Thread	176
9.3.4.1.1	Integrity Verification	177
9.3.4.1.1.1	Motivation for Verifying Integrity	177
9.3.4.1.1.2	Hash Values Generation Technique	177
9.3.4.1.1.3	Integrity Verification Technique	178
9.3.4.1.1.4	Integrity Verification Results	179
9.3.4.1.2	Pre-Incident Analysis	182
9.3.4.1.2.1	RAM usage analysis	183
9.3.4.1.2.2	CPU usage analysis.....	184
9.3.4.2	Installer and ping Server Thread	185
9.3.4.3	Evidence Capture Thread.....	186
9.3.4.3.1	Hashing.....	187
9.3.4.3.2	MD5 Encoding	187
9.3.4.4	Send Data Thread.....	188
9.3.4.5	Forensic Readiness Reporting	190
9.4	Phases Not Implemented.....	190
9.5	Implementation Challenges	192
9.5.1	NMB obstruction.....	192
9.5.2	NMB implementation.....	192
9.5.3	Live evidence acquisition	192
9.5.4	Legal authority.....	193
9.6	Conclusion	193
	Chapter 10: Critical Evaluation of Research Study	195
10.1	Introduction.....	195
10.2	Critical Evaluation of the Proposed CFRaaS Model.....	196
10.3	Critical Evaluation of the Proposed CFRaaS Prototype	199
10.4	Evaluation of Research Objectives	201
10.5	Evaluation of Research Questions	204
10.5.1	Main Research Question	205

10.5.2	Research Sub-Questions.....	206
10.6	CFRaaS Implementation Challenges.....	209
10.6.1	Overview.....	209
10.6.2	Motivation for Exploring Challenges.....	211
10.6.3	Related Work on Challenges.....	212
10.6.4	Challenges and Proposed High-Level Solutions.....	214
10.6.4.1	General Challenges	215
10.6.4.1.1	NMB Disinfection.....	215
10.6.4.1.2	NMB Implementation.....	216
10.6.4.1.3	Increased Number of Devices.....	216
10.6.4.1.4	Technological Changes.....	217
10.6.4.1.5	Trust of Forensic Data in the Cloud.....	217
10.6.4.1.6	Large-Scale Data Management.....	217
10.6.4.1.7	Forensic Evidence Monitoring.....	218
10.6.4.2	Technical Challenges	219
10.6.4.2.1	Live Evidence Acquisition.....	219
10.6.4.2.2	Virtualisation.....	220
10.6.4.2.3	Data Volatility.....	220
10.6.4.2.4	Data Integrity.....	220
10.6.4.2.5	Anti-Forensics.....	221
10.6.4.2.6	PDE Handling.....	221
10.6.4.2.7	Malicious Activity.....	221
10.6.4.2.8	Privacy Issues.....	222
10.6.4.2.9	Multi-tenancy.....	223
10.6.4.2.10	Big Data.....	223
10.6.4.2.11	Encrypted Data.....	224
10.6.4.3	Operational Challenges	227
10.6.4.3.1	Legal Authority.....	227
10.6.4.3.2	Colossal Forensic Evidence Analysis in the Cloud.....	228
10.6.4.3.3	Contractual Obligations.....	228
10.6.4.3.4	Standard Operating Procedures (SOPs).....	228
10.6.5	Discussion.....	229
10.7	Conclusion.....	231
Chapter 11: Conclusion		232
11.1	Introduction.....	232
11.2	Discussion on Novel Contributions.....	232

11.3 Future Work	234
11.4 Final Conclusion	236
BIBLIOGRAPHY	237
APPENDIX A. PSEUDO-EQUATIONS AND ILLUSTRATIONS.....	256
APPENDIX B. Professional Publications	267
APPENDIX B.1 List of Papers Published in Peer Reviewed Journals and International Conferences.....	269
B.1.1 Peer Reviewed Scientific Journals.....	269
B.1.2 Peer Reviewed International Conferences	269

LIST OF TABLES

Table 2.1 Existing Digital Forensic Investigation Process Models	28
Table 5.1 Summary of General Model Requirements	76
Table 5.2 Functional and Non-functional Requirements	77
Table 7.1 W_{1TP} and W_{2TP} Events With Attributes for ESM Testing	127
Table 7.2 W_{1TP} and W_{2TP} Events With Attributes and Values when $p=1, 2 > \text{to } \infty$	128
Table 7.3 Comparing the Proposed CFRaaS Model with Existing Forensic Readiness Models ..	136
Table 8.1 CFRaaS Design Processes and Sub-Process	145
Table 8.2 System Specifications for Development and Testing	149
Table 8.3 Representation of Communication Links	152
Table 8.4 Communication Links of the Main Threads	155
Table 8.5 Connection Listener Thread Communication Links	157
Table 8.6 Installer Thread Communication Link	159
Table 8.7 Ping Server Thread Communication Links	161
Table 8.8 Evidence Capture Thread Communication Link	163
Table 8.9 Update Thread Communication Link	164
Table 8.10 Update Thread Communication Link	166
Table 8.11 Evidence Capture Thread Communication Link	167
Table 9.1 A Comparison of MD5 Hash Values Generated by CFRaaS Prototype and MD5 & SHA-1 Checksum Utility	182
Table 10.1 A Summary of Achieved Research Objectives	203
Table 10.2 Issues and challenges faced in implementing an s	225

LIST OF FIGURES

Figure 1.1 Thesis Layout	9
Figure 2.1 Classes of Digital Investigation Process	25
Figure 3.1 A Visual Model of NIST Cloud Computing Definition	38
Figure 3.2 Roles of Cloud Service Provider.....	44
Figure 3.3 Example of Virtualisation	46
Figure 4.1 Life Cycle of a Botnet.....	53
Figure 4.2 Architecture of a Centralised C&C Architecture.....	54
Figure 4.3 Architecture of a Decentralised C&C Architecture	56
Figure 5.1 High-level View of the Process of Obfuscating a Forensic Agent.....	74
Figure 7.1 High-Level Overview of the CFRaaS Model.....	97
Figure 7.2 CFRaaS Provider Layer	99
Figure 7.3 CFRaaS Virtualisation Layer.....	100
Figure 7.6 CFRaaS Approach Strategy.....	104
Figure 7.7 Digital Evidence Collection	109
Figure 7.8 Pre-Incident Analysis.....	115
Figure 7.9 Incident Detection.....	119
Figure 7.10 Event Reconstruction	122
Figure 7.11 Event Search Function	125
Figure 7.12 ESM of Events for $p=1$, $p=2$ and $p=2$ to ∞	129
Figure 7.13 Forensic Readiness Report.....	130
Figure 7.15 Block Diagram of the Detailed CFRaaS Model.....	135
Figure 7.16 Comparisons of Proposed CFRaaS and Existing Forensic Readiness Models	138
Figure 8.1 CFRaaS Prototype Main Components	143
Figure 8.2 High-level view of the NMB process.....	150
Figure 8.3 Processes Initiation by the C&C Server and the Bot Client	151
Figure 8.4 The Main Thread	154
Figure 8.5 Connection Listener Thread	156
Figure 8.6 Installer Thread	158
Figure 8.7 The Ping Server Thread	160
Figure 8.8 The Evidence Capture Thread	162
Figure 8.9 The Update Thread.....	164
Figure 8.10 The Destroy Thread	165
Figure 8.11 The Send Data Thread.	166
Figure 8.12 Detailed Flow of the NMB Process.....	168
Figure 9.1 CFRaaS Experimental Set-up.....	171
Figure 9.2 Overview of the NMB Process Threads	175
Figure 9.3 Forensic Log Extraction Control Panel	176
Figure 9.4 Actions of Creating a Hash	178
Figure 9.5 Hash of Captured Forensic Logs	178
Figure 9.6 MD5 & SHA-1 Checksum Utility	179
Figure 9.7 Matched MD5 Hash for Log File 1.....	180
Figure 9.8 Matched MD5 Hash for Log File 2.....	180
Figure 9.9 Matched MD5 Hash for Log File 3	181
Figure 9.10 Unmatched MD5 Hash for Altered Log File 3	181
Figure 9.11 Collected CPU and RAM Usage, Timestamp Information.....	183
Figure 9.12 Pre-Analysis of RAM Usage Graph.....	183
Figure 9.13 Analysis of CPU Usage Graph	184
Figure 9.14 Analysis of CPU Usage Graph.....	185
Figure 9.15 Installed and Running Bot Client	186

Figure 9.16 A block of Captured Potential Digital Evidence.....	186
Figure 9.17 Stored Cryptographic Hash	187
Figure 9.18 POST/Send Data Action.....	188
Figure 9.19 Forensic Log, Hash, Timestamp, Machine IP Address, Machine ID	189
Figure 9.20 Forensically Captured Keystrokes in a Readiness Approach	189
Figure 10.1 A Hierarchical Classification of CFRaaS Implementation Issues and Challenges	215

LIST OF EQUATIONS

Equation 5.1: Forensic Logging	68
Equation 5.2: Digital Preservation	69
Equation 5.3: Hashing Equation	69
Equation 5.4: Overall Digital Preservation Equation	69
Equation 5.5: Timestamping Equation.....	70
Equation 5.6: Digital Evidence Characterisation	71
Equation 5.7: Vector Obfuscation	73
Equation 5.8: Jacobian Equation	73
Equation 5.9: Vector Obfuscation Matrix(I).....	73
Equation 5.10: Vector Obfuscation Matrix(II).....	73
Equation 5.11: Component-wise Decomposition of Obfuscation Matrix	73
Equation 5.12: Vector used to obfuscate an agent.....	74
Equation 7.1: Cloud Model Equation	93
Equation 7.2: CSP with Cloud Clients	93
Equation 7.3: Cloud Client Interdependence	93
Equation 7.4: CSP Interdependence	94
Equation 7.5: Client and CSP Intersection	94
Equation 7.6: Representation of CSP	95
Equation 7.7: CSP and Data Centre Formulation	95
Equation 7.8: CSP Deployable Equation	95
Equation 7.9: Data Centre Equation.....	96
Equation 7.10: Client and Data Centre Equation.....	96
Equation 7.11: Physical Server Equation	96
Equation 7.12: Virtual Server Equation	96
Equation 7.13: OS Equation.....	96
Equation 7.14: Overall Formalised Cloud Architecture Equation.....	96
Equation 7.15: Risk Equation.....	107
Equation 7.16: Cloud Client Activity Equation	108
Equation 7.17: Forensic Log Identifier.....	113
Equation 7.18: Timestamp Activity	113
Equation 7.19: Log Event Equation.....	114
Equation 7.20: Event Attribute Equation	114
Equation 7.21: Formalised CFRaaS Model	115
Equation 7.22: CFRaaS Model Digital Preservation Equation	117
Equation 7.23: Digital Preservation Properties	117
Equation 7.24: Set of Forensic Activities with Attributes	117
Equation 7.25: Overall Activity Representation.....	117
Equation 7.26: Incident Detection Rate.....	119
Equation 7.27: Incident Detection Rate Computation	120
Equation 7.28: Incident Growth Rate	120
Equation 7.29: Incident Response Mechanism.....	121
Equation 7.30: Event Occurrence	123
Equation 7.31: Minkowski Distance Metric	126

Equation 7.32: Attribute Absolute Value Distance127
Equation 7.33: Attribute Root Square Differences.....128
Equation 7.34: Attribute Absolute Magnitude Differences128

ABSTRACT

The ubiquity of the cloud has accelerated an abundance of modern Information and Communication Technology (ICT)-based technologies to be built based on the cloud infrastructures. This has increased the number of internet users, and has led to a substantial increase in the number of incidents related to information security in the recent past, in both the private and public sectors. This is mainly because criminals have increasingly used the cloud as an attack vector due to its prevalence, scalability and open nature. Such attacks have made it necessary to perform regular digital forensics analysis in cloud computing environments. Digital Forensics (DF) plays a significant role in information security by providing a scientific way of uncovering and interpreting evidence from digital sources that can be used in criminal, civil or corporate cases. It is mainly concerned with the investigation of crimes that are supported by digital evidence. Furthermore, DF is conducted for purposes of uncovering a potential security incident through Digital Forensic Investigations (DFIs).

There is always some degree of uncertainty when cyber-security incidents occur in an organisation. This is because the investigation of cyber-security incidents, as compared to the investigation of physical crimes, is generally still in its infancy. Unless there are proper post-incident response and investigating strategies in place, there will always be questions about the level of trust and the integrity of digital forensic evidence in the cloud environment. The impact of cyber-security incidents can be enormous. Much damage has already been experienced in many organisations and a disparity between cyber-security incidents and digital investigations lies at the origin of where an incident is detected. Organisations need to reach a state of Digital Forensic Readiness (DFR), which implies that digital forensic planning, preparation must be in place, and that organisations can implement proper post-incident response mechanisms.

However, research study on science and theories focused on the legal analysis of cloud computing has come under scrutiny because there are several constitutional and statutory provisions with regard to how digital forensic evidence can be acquired from Cloud Service Providers (CSPs). Nevertheless, for Digital Forensic Evidence (DFE) to satisfy admissibility conditions during legal proceedings in a court of law, acceptable DF processes should be systematically followed. Similarly, to enable digital forensic examination in cloud computing environments, it is paramount to understand the technology that is involved and the issues that relate to electronic discovery. At

the time when this research thesis was being written, no forensic readiness model existed yet that focused on the cloud environment and that could help cloud-computing environments to plan and prepare to deal with cyber-security-related incidents.

The aim of this research study is therefore to determine whether it is possible to achieve DFR in the cloud environment without necessarily having to modify the functionality and/or infrastructure of existing cloud architecture and without having to impose far-reaching architectural changes and incur high implementation costs. Considering the distributed and elastic nature of the cloud, there is a need for an easy way of conducting DFR by employing a novel software application as a prototype. In this research thesis, therefore, the researcher proposes a Cloud Forensic Readiness as a Service (CFRaaS) model and develops a CFRaaS software application prototype. The CFRaaS model employs the functionality of a malicious botnet, but its functionalities are modified to harvest digital information in the form of potential evidence from the cloud. The model digitally preserves such information and stores it in a digital forensic database for DFR purposes.

The experiments conducted in this research thesis showed promising results because both the integrity of collected digital information and the constitutional and statutory conditions for digital forensic evidence acquisition have been maintained. Nevertheless, the CFRaaS software application prototype is important because it maximises the use of digital evidence while reducing the time and the cost needed to perform a DFI. The guidelines that have been used while conducting this process comply with ISO/IEC 27043:2015, namely Information Technology - Security techniques - Incident investigation principles and processes. The ISO/IEC 27043 international standard was used in this context to set the guidelines for common incident investigation processes. Based on this premise, the researcher was able to prove that DFR can be achieved in the cloud environment using this novel model.

Nevertheless, the proposed CFRaaS concept prepares the cloud to be forensically ready for digital forensic investigations, without having to change the functionality and/or infrastructure of the existing cloud architecture. Several CFRaaS prototype implementation challenges have been discussed in this research thesis from a general, technical and operational point of view. Additionally, the researcher could relate the challenges to existing literature and eventually contributed by proposing possible high-level solutions for each associated challenge.

LIST OF ABBREVIATIONS USED

ADFM	Abstract Digital Forensic Model
APG	Assessment Process Group
APC	Acquisitive Process Class
API	Application programming interface
ACPO	Association of Chief Police Officers
BSD	Berkeley Software Distribution
CART	Computer Analysis Response Team
CERT	Computer Emergency Response Teams
CFR	Cloud Forensic Readiness
CFRaaS	Cloud Forensic Readiness as a Service
CFRaaS	Cloud Forensic Readiness as a Service Prototype
CFTT	Computer Forensic Tool Testing
CIP	Critical Infrastructure Protection
CIS	Critical Infrastructure Systems
CM	Cloud Model
CPC	Concurrent Process Class
CPU	Central Processing Unit
CSPs	Cloud Service Providers
CSA	Cloud Security Alliance
CSIRT	Computer Security Incident Response Team
C&C	Command and Control
CL	Client
CART	Computer Analysis and Response Team
DC	Datacentre
DDoS	Distributed Denial of Service
DDNS	Dynamic Domain Name System
DE	Digital Evidence
DEC	Digital Evidence Characterisation
DFRWS	Digital Forensic Research Workshop
DF	Digital Forensics
DFE	Digital Forensic Evidence
DFR	Digital Forensic Readiness
DFI	Digital Forensic Investigations
DFIP	Digital Forensic Investigation Process
DFMMIP	Digital Forensic Model based on Malaysian Investigation Process
DNA	Deoxyribonucleic acid
DP	Digital Preservation
DoS	Denial of Service
DNS	Domain Name System
ECPA	Electronic Communication Privacy Act
EDIP	Enhanced Digital Investigation Process
EC2	Elastic Compute Cloud

ECT	Electronic and Communication Transaction Act
ESM	Event Similarity Measure
ESF	Event Search Function
EMCI	Extended Model of Cybercrime Investigation
ENISA	European Network & Information Security Agency
FA	Forensic Agent
FRs	Functional Requirements
FTP	File Transfer Protocol
FBI	Federal Bureau of Investigations
FDI	Framework for a Digital Investigation
FCCS	Federal Cloud Computing Strategy
FRE	Federal Rules of Evidence
GIFTIR	Guide to Integrating Forensic Techniques to Incident Response
HDFIPM	Harmonised Digital Forensic Investigation Process Model.
HTTP	Hypertext Transfer Protocol
HTTPS	Secure HTTP
HB	Hybrid Cloud
IaaS	Infrastructure as a Service
IC	Incident Classification
ICR	Internet Crime Report
IBM	International Business Machines
IDIP	Integrated Digital Investigation Process
IDR	Incident Detection Rate
IEC	International Electro Technical Commission
IP	Internet Protocol
IPC	Initialisation Process Class
IPG	Implementation Process Group
ISO	International Organisation for Standardisation
IT	Information Technology
IRC	Internet Relay Chat
IRM	Incident Response Mechanism
ID	Identity
PI_DES	Pre-Incident Description
IDC	International Data Corporation
IDS	Intrusion Detection Systems
IDR	Incident Detection Rate
IGR	Incident Growth Rate
IRM	Incident Response Mechanism
IRP	Incident Response Procedures
IRC	Internet Relay Chat
IRT	Incident Response Team
LEA	Law Enforcement Agencies
MacOS	Macintosh Operating System
MD5	Message Digest 5
MIME	Multi-Purpose Internet Mail Extensions
MST	Main Server Thread

MYSQL	“My” Structured Query Language
NAB	Not-A-Bot
NFRs	Non-Functional Requirements
NIST	National Institute of Standards and Technology
NIST SP	National Institute of Standards and Technology Special Publication
NMB	Non-Malicious Botnet
NMBaaS	Non-Malicious Botnet as a Service
NIJ	National Institute of Justice
NR_PDE	Non-Relevant Potential Digital Evidence
OB	Obfuscation
OS	Operating System
Org	Organisation
PaaS	Platform as a Service
PB	Public Cloud
PDE	Potential Digital Evidence
PHP	Hypertext Preprocessor
Pi_A	Pre-Incident Analysis
Pi_P	Pre-Incident Planning
PL	Provider Layer
PoPI	Protection of Personal Information
POP3	Post Office Protocol 3
PPG	Planning Process Group
PR	Private Cloud
Ps	Physical Server
PSE	Possible Security Events
PSI	Potential Security Incident
P2P	Peer to Peer
QoS	Quality of Service
RFC	Request for Comments
RICA	Regulation of Communications and Provision of Communication
RPGs	Readiness Process Groups
RT	Receiver Thread
R_PDE	Relevant Potential Digital Evidence
SaaS	Software as a Service
SCA	Stored Communication Act
SDFIM	Systematic Digital Forensic Investigation Model
SHA-1	Secure Hash Algorithm 1
SIEM	Security Information and Event Management
SLAs	Service Level Agreements
SMEs	Small to Medium Enterprises
SOCKS	Socket Secure
SOPs	Standard Operating Procedures
SOA	Service Oriented Architectures
SQL	Structured Query Language
SYN	Synchronise
SunOS	Sun Microsystem Operating System

SWGDE	Scientific Working Group on Digital Evidence
TCL	Tool Control Language
TCP	Transmission Control Protocol
TP	Timestamp
UDP	User Datagram Protocol
URL	Uniform Resource Locator
VM	Virtual Machine
Vs	Virtual Server
WEP	Wireless Equivalent Privacy
XaaS	Everything as a Service

ACKNOWLEDGEMENTS

I would firstly like to thank the Almighty God for the gift of life. This is what has enabled me to pursue PhD research. Secondly, I sincerely would like to thank my academic advisor **Prof Hein S. Venter** for his insightful guidance, patience, support and untiring brainstorming sessions on cloud forensics aspects. **Prof Hein S. Venter** paved way for me to have a broad vision of learning, research and guidance that led to the successful completion of this thesis. He has been resourceful, always believed in me, offered continuous support and he was always a source of encouragement throughout this journey. I discussed with him about this thesis more than anyone else. Personally, I owe **Prof Hein S. Venter** an inexpressible debt of gratitude.

Equally, I would like to thank **Prof Martin S. Olivier**; the University of Pretoria, for offering expert opinions on digital forensics subject matter continuously as far as coming up with this research thesis was concerned. These suggestions encouraged me in many ways and they helped me overcome immeasurable difficulties that were involved at the time of this research.

I would like to thank Information and Computer security Architectures (ICSA) research group for the endless support for each and every day and night that we sat as a family to brainstorm on digital forensic parameters. I am indeed grateful for the exchange of ideas time and again. It is worth noting that these insights were indeed supportive. Also, I will at large would like to thank the Department of Computer Science, University of Pretoria, Hatfield Campus.

A million thanks go to my family who always believed and encouraged me in many ways even when times were hard. My Dad Robert O. KEBANDE, Mum Jane M. KEBANDE, Obed O. KEBANDE, Ericah N. KEBANDE, Elsie, Ethan, Romeo. Thanks for your patience and I still believe when times are hard and you cannot see the light, always walk by faith.

Last but not least, I wish to thank all the members who at some point stepped in my way educationally while pursuing this Ph.D. The list includes; Dr. Nickson Karie Menza (Semantic Disparities), I have no words the journey we started together is finally here, Herman Ntsamo and

Elsabe Ros for working closely with me while coming up with the prototype, Isabell Classen for professionally language editing this research thesis. Others include Dr. Ikusean, Ivan, Ben Magara Maake, Sayo, Makura, Stacey, Irene Venter, Masvosvere, Abboye, my friend Werner Hauger, Mokgadi Lindiwe, Pierre Procore, M.Makgatho, M.Mowaduba. The names are endless.

Finally, I would like to thank the funders, who made it possible for this PhD. University of Pretoria-UP Postgraduate Doctoral Research Award, UP Research Support, Special International Research Award. I acknowledge Egerton University for the support and National Council of science and Technology, NACOSTI (NACOSTI/RCD/ST&I 5TH CALL PhD/125).

*“Reading is going toward something that is about to be,
and no one yet knows what it will be.”*

– Italo Calvino, *If on a Winter's Night a Traveler*

Part One: Introduction

Part One consists of **Chapter 1** (the current chapter), which serves as an introduction to the research topic and lays the foundation for the rest of the research thesis. Specifically, this chapter provides the reader with a brief introduction to the research by setting the scene by means of a broad overview.

Next, Chapter 1 highlights the subject of the research thesis and identifies the main research problem (i.e. how DFR can be conducted in the cloud environment without changing functionalities and services of the existing cloud architectures). The chapter also includes the motivation for the study, research objectives and a conclusion.

Chapter 1: Introduction

1.1 Introduction

The advents of Information Technology (IT), the pervasiveness of the internet and numerous revolutionary innovations have had an influence on the way we operate in our daily lives. Furthermore, the impact of these phenomena has been felt widely in our societies because of the manner in which information is disseminated. Communications have been revolutionised through the creation of an interactive global network among organisations, governments, businesses, the military establishment, health institutions and sporting events, while vast investment opportunities have further enabled a seamless communication between Information Communication Technology (ICT) infrastructures. These significant technology-driven changes have also created new working patterns by transforming our day-to-day operations. New and ever-changing structures have influenced effective communication and created mechanisms for storing, manipulating and distributing information worldwide through the internet.

As a result of these advancements, modern computer networks are built on cloud infrastructures because cloud computing enables users to have an unprecedented ability in regard to how their data is being handled due to its vast resources. Moreover, the cloud has been preferred by many organisations because of its ability to operate in a virtual environment, provide Service-Oriented Architectures (SOA), support multi-tenancy architectures, reduce IT expenditures, reduce administration overhead costs and improve scalability (Kebande & Venter, 2015). Additionally, the development of cloud technology has facilitated numerous cloud-based innovations focusing on education without barriers, where access to data is not limited. Besides, cloud technology has seen the use of shared resources and steered major developments in sectors like banking, agriculture, science, engineering and healthcare. Due to the presence of shared resources and services in this environment, maintaining the security of vendors and consumers is of great importance (Ramgovind, Eloff & Smith, 2010).

At the same time that these technologies have become prevalent, the threat landscape has also evolved tremendously. Numerous significant concerns have been voiced regarding the increase in security-related incidents. This has resulted mainly from the fact that operations

and services provided over the cloud cannot be handled in conventional ways (Chambers, 2009). In some instances, security incidents go as unforeseen catastrophes, which mean that organisations can be compromised if the necessary measures are not in place to help in mitigating the effects of potential security risks. Such risks have unfortunately occurred because of the inability of organisations to prevent, detect and report security incidents. For example, the European Agency for Network and Information Technology (ENISA) published a cyber-security strategy that focuses mainly on how to prevent attacks that are channelled over networks and information systems, and how to prevent large-scale failures (ENISA, 2013). The main reason for publishing this strategy was to ensure compliance by businesses that provide critical services about how they should report security incidents. Therefore, since cyber-security incident detection is an important part of Critical Infrastructure Protection (CIP), it should be given priority by organisations to avert or prevent large-scale failures.

The main way to thwart a circumstance that has the potential to cause a security incident is by trying to expose its root cause so that the full impact of the incident can be scrutinised. Normally a Digital Forensic Investigation (DFI) is required to prove whether a security incident actually occurred at a particular time and place. Additionally, this requires analysis of specific aspects that might help to prove or disprove a given hypothesis in a court of law. The need for digital investigations has been preferred mainly because Digital Forensics (DF) has constantly adjusted to the growing and evolving computer technologies.

The field of DF provides a key solution to how computer- and cyber-related crimes can be solved. Better ways need to be found to gain an understanding of what a crime scene is, as well as to be able to unravel the suspect's identity and the actual motives of a suspect. Every organisation and every individual should know what DF involves, because more and more people, businesses and personal transactions nowadays use devices that have computing capabilities. An estimate conducted over the annual cost of cybercrime is estimated at \$113 Billion across 24 countries in the world (Lampe, 2015). Nevertheless, the Federal Bureau of Investigation (FBI) through its 2016 Internet Crime Report has highlighted that a total of 298,728 complaints/cyber-related crimes were recorded with a loss estimated at \$1.3 Billion (ICR, 2017). This is an indication that digital forensics should be enforced to urge organisations to make computer and information security a priority.

Digital forensics has therefore become an integral part of computer and information security. When the necessary DF technology is applied to cyber-related incidents, then many legal problems can be solved and security incidents can be managed far more effectively. Nevertheless, every organisation needs to enforce essential strategies, technologies, policies and procedures that comply with digital forensic aspects so as to ensure proper cyber-security incident management.

The remainder of Chapter 1 is organised as follows: The next section 1.2 introduces the reader to the problem statement and research questions examined in this research study. After that follows the motivation for the study 1.3 and then a discussion of the research objectives in Section 1.4. Section 1.5 deals with the methodology applied in the research, before the chapter closes with an exposition of the thesis layout 1.6 and a brief conclusion 1.7.

1.2 Problem Statement

Cloud forensics, which is perceived to be the amalgamation of cloud computing and digital forensics, has grown enormously in a few years, and conventional digital forensics investigation techniques are not sufficient to deal with the challenges of the cloud environment. Corporate investigation teams and law enforcement agencies (LEAs) face a virtually impossible task when trying to prove whether an electronic event occurred in the cloud environment in a particular instance. On the same note, DF investigators have in various instances tried to prove the existence of digital evidence in the cloud infrastructure. Acquiring helpful digital information is a momentous challenge, because digital evidence is de-centralised across several and multiple servers and platforms. This problem is exacerbated by the fact that during the process of a DFI, an investigator may not have any control over the particular cloud environment.

The main problem addressed in this research thesis can therefore be summarised as follows: **Conventional DFI techniques are not suitable for use in the cloud environment and it is currently not possible to apply Digital Forensic Readiness (DFR) in the cloud environment without having to change the functionalities and architecture of existing cloud computing infrastructure. DFR implies that an organisation has forensic preparedness and planning in place in the event of security events. The ultimate goal of**

employing DFR in any organisation is to save time and money for the actual DFI process.

Thus the main question addressed in this research can be defined as follows:

Is it possible to proactively prepare and plan a digital forensic readiness process in the cloud environment?

In addition to overcome this drawback and to find more suitable solutions to the problem mentioned above, this thesis addresses the following research sub-questions in an attempt to address the main research problem.

- 1. What are the suitable techniques of conducting DFR in the cloud environment?** Everything in a cloud is geographically distributed around data centres and runs in a virtualised environment (Marturana et al., 2012). Seeing that essential artefacts are distributed too, an organisation needs to have DFR measures in place that can help with post-event response. According to Rowlingson (2004) an effective DFR approach requires incident preparedness to be a corporate goal. Thus, it is also important for an organisation to be acquainted with evidence collection requirements, which are discussed later in the thesis.
- 2. Is it possible to conduct DFR in the cloud environment without having to change the functionality and/or infrastructure of the existing cloud architectures?** It is essential to avoid modifying the functionality, infrastructure and services of the existing cloud architecture because this may save cost and time. A given organisation should know what kind of information should be collected, through which suitable sources or via which communication channels, when preparing for DFR. Nevertheless, a digital forensic investigator should be equipped with the best methods possible for handling the collected evidence in the cloud computing environment.
- 3. How can one digitally preserve Potential Digital Evidence (PDE) harvested from the cloud environment so that it can be used for DFR purposes? In other words, how can one preserve the integrity of collected PDE?** It is vital to know how to store collected information that may be used as potential evidence in a court of law. An organisation should be aware of

ways in which the integrity of Potential Digital Evidence (PDE) can be maintained before it is presented in a court of law. Horner (2002) highlights that “integrity of digital data becomes paramount for the accused, digital forensic investigators and the court prosecutors”.

4. **Can a software application that was originally applied for malicious purposes be used – without being detected – to capture PDE in a cloud environment? Can this be done in a non-malicious fashion for DFR purposes?** It is vital to know whether a software application that was originally considered malicious can be used in a virtual environment to collect potential evidence that can be used for DFR purposes. For example, a criminal may use a gun for illegal purposes while a police officer may use the same gun for law enforcement purposes. Furthermore, the researcher attempts to figure out whether such software applications could be detected in a cloud environment.

5. **What issues and challenges are encountered when conducting DFR by using a non-malicious software application in the cloud environment? What are the possible high-level solutions?** In the context of this research, it will be important to know the issues and challenges that may arise because of using a malicious software application to perform forensic monitoring in a non-malicious way in the cloud environment. This is because the existence of these challenges is a hindrance to forensic tools in general during acquisition and examination of digital evidence. Nevertheless, highlighting these challenges is a step towards the future development of stronger DFR tools.

The sub-questions provide necessary information to DF investigators when trying to check the systematic sequence of events. In the final part of this research, the researcher will identify potential technical, operational and legal challenges that have been encountered due to the implementation of this research study. This will then be followed by a proposal for possible high-level solutions.

1.3 Motivation

The field of digital forensics is still in its infancy and a number of methodologies on how DFR can be achieved have before been proposed in the ISO/IEC 27043: 2015 international standard. However, at the time of writing this research thesis, no standardised process that focuses on the cloud environment has yet been proposed. When a Digital Forensic Investigation Process (DFIP) with some relevance to a particular incident is conducted in a cloud environment, one should be able to extract PDE that can satisfy admissibility requirements in a court of law. Therefore, the primary motivations for this research thesis are as summarised below:

- **An exponential increase of security threats in the cloud environment.**
According to the Cloud Security Alliance (2008), widespread threats in the cloud have hindered the way the cloud operates, because clients' data and applications are moved to centralised data centres. This has led to commensurate concerns in the cloud environment regarding the risk of personal and private data, as well as the data of businesses that have moved a majority of their applications to the cloud. These prevailing cloud security threats include data loss; data breaches; insecure Application Programming Interfaces (APIs); traffic hijacking; Denial of Service (DoS); the existence of malicious insiders, and abuse of the services provided by the cloud.
- **Lack of standardised guidelines for conducting digital forensic readiness in the cloud environment.**
There is an ever-increasing need to standardise the DFR processes in the cloud environment. However, to date, no specified guidelines and standardised models or frameworks have been suggested on how to conduct DFR in the cloud environment. According to Mouhtaropoulos, Li and Grobler (2012), standardisation of the proactive process remains a struggle that is yet to gain worldwide acceptance. This is a daunting challenge to the corporate investigating teams and law enforcement agencies, because the cloud environment might not be suitable in allowing DF investigators to represent Digital Forensic Evidence (DFE) at a given time. However, while this research thesis was being finalised, a published umbrella standard of ISO/IEC 27043: 2015, which is the international standard for high-level concepts, confirmed the need to standardise and prioritise security incidents (ISO/IEC 27043, 2015).

- **The need to help digital forensic investigators and law enforcement agencies during incident response.**

Digital forensic investigators and LEAs need to be provided with an established process and accepted guidelines so that they can manage the incident response without having to fear the violation of statutory laws and regulations. Rowlingson (2004) highlights this instance as the ability of an organisation to access digital forensic evidence that will support the organisation in a legal process when there is an event. Moreover, one can only succeed with a legal process in the cloud if digital evidence is gathered actively and if it is available when needed by digital forensic investigators and LEAs.

The aforementioned bullets have shown the reader why there is a need to explore the problem that has been stated in Section 1.2 of this research thesis. Therefore, in the next section, the objectives that are aimed to be achieved in this thesis as a result re highlighted.

1.4 Research Objectives

The main objective of the research presented in this thesis is to determine how DFR can be achieved in cloud computing environments without having to change the functionalities and infrastructure of the prevailing cloud architecture. The following tasks have been addressed as research objectives in this research thesis (these objectives are met in the chapters throughout the remainder of the thesis):

- Conduct a comprehensive literature review on digital forensics, cloud computing and botnets.
- Propose the requirements and techniques used to attain DFR in cloud computing environments.
- Propose a novel forensic cloud model to perform DFR and propose systematic processes that can be used during PDE collection from the cloud environment.
- Contribute towards a prototype that acts as a proof of concept on how a proactive DFR process can be achieved in a cloud environment.
- Show the effectiveness of the proposed model in a virtualised environment.

The reader has now been introduced to the research objectives that are set to be achieved systematically in this research thesis. It is, therefore, important for the reader to know the

technique and methodology that is going to be employed to achieve the above-mentioned objectives. Therefore, in the next section, the reader is introduced to the methodology using in this research thesis.

1.5 Methodology

To meet the objectives that have been highlighted in Section 1.4, the researcher performed a comprehensive review of literature that is related to this study and conducted experiments that can help to achieve DFR. Consequently, to answer the research questions that have been posed in the problem statement, the researcher conducted scientific and descriptive research on the best way of conducting DFR in the cloud computing environments. The scientific part of this experiment involved conducting laboratory experiments, while the descriptive part involved literature that addressed cloud forensics. Nevertheless, mathematical approaches and an evaluative analysis have also been used as part of the proposed technique. In addition to that the model that has been proposed in this research thesis is largely based on mathematical constructs and set theory.

The researcher not only explored the current state of digital forensics, DFR, botnets and cloud computing, but also conducted a survey of the literature. In addition, the researcher expanded on different problems that have been identified by different researchers as related work. Based on the above literature review, a set of requirements to be fulfilled by the proposed model was generated.

After identification of the requirements, a model that complies with the deduced requirements was developed. An important objective of the created model was that it had to comply with the ISO/IEC 27043: 2015 to the extent that it would introduce new ideas. Moreover, the model had to be feasible, friendly and easy to implement. To prove that the model was viable, hypothetical case scenarios and an implementation are conducted with the help of a prototype that acted as a proof of concept. A critical evaluation of the prototype that acted as a proof of concept is presented in the final part of the study. Expected findings of this research study include the harvesting of digital forensic information that can be used as potential evidence for DFI purposes within the cloud environment.

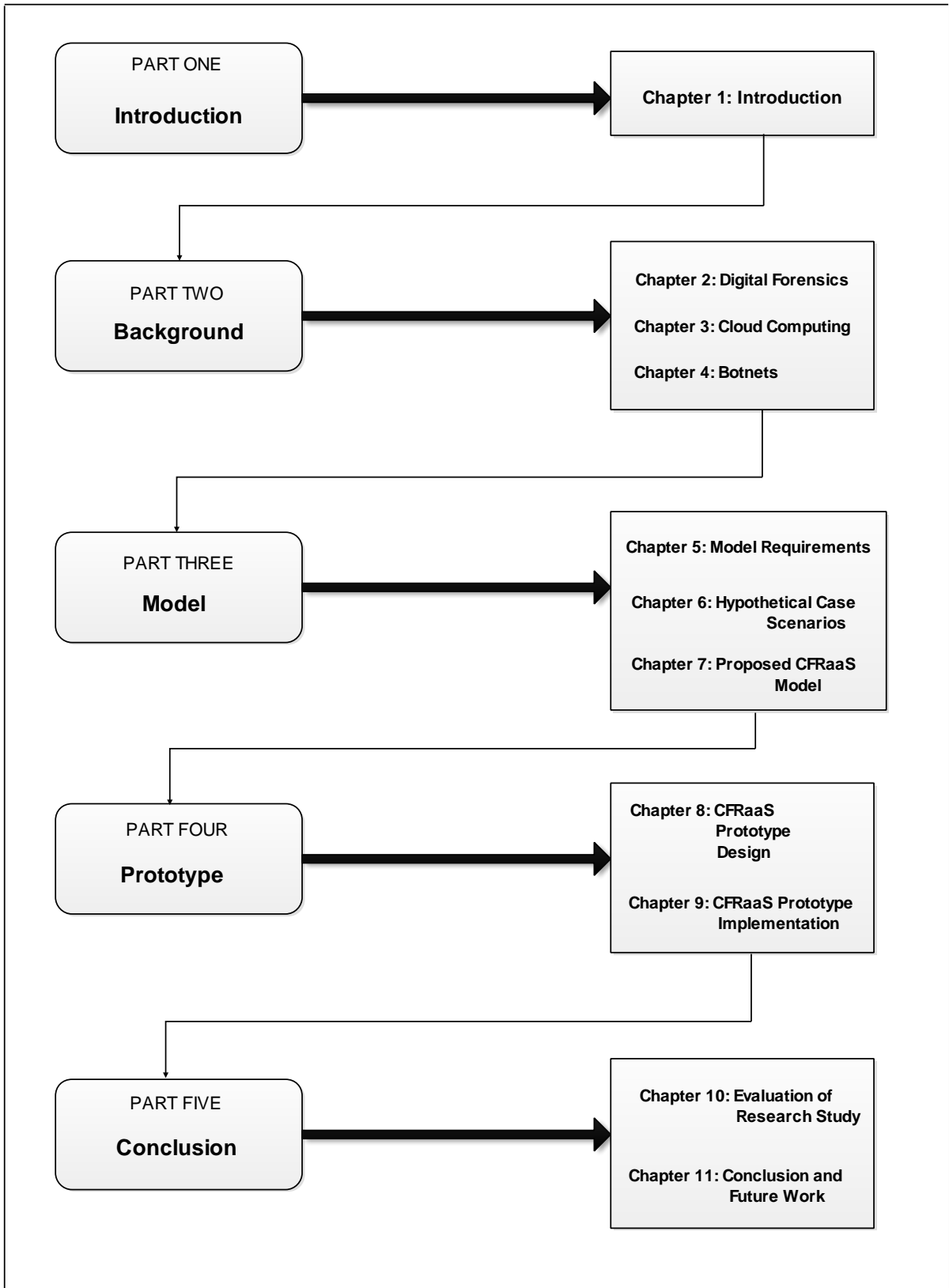


Figure 1.1 Thesis Layout

1.6 Thesis Layout

This section provides a layout of the remainder of the research thesis. It is structured in five parts and comprises eleven chapters. Figure 1.1 shows the relationships between the parts and the chapters. Each of the thesis parts is discussed briefly in the sections below. Additionally, works that are presented in this research thesis have already been published in scientific journals and international peer-reviewed conference proceedings. This has been shown in **Appendix B** of this thesis.

1.6.1 Part One: Introduction

Part One consists of **Chapter 1** (the current chapter), which serves as an introduction to the research topic and lays the foundation for the rest of the research thesis.

Specifically, this chapter provides the reader with a brief introduction to the research by setting the scene by means of a broad overview. Next, Chapter 1 highlights the subject of the research thesis and identifies the main research problem (i.e. how DFR can be conducted in the cloud environment without changing functionalities and services of the existing cloud architectures). The chapter also includes the motivation for the study, research objectives and a conclusion.

1.6.2 Part Two: Background

Part Two of this thesis provides the reader with some background to the research. It consists of three chapters, Chapters 2, 3 and 4. **Chapter 2** gives a comprehensive review of digital forensics, cloud computing and botnets, and it provides the background context of digital forensics and DFR. In addition, the chapter discusses digital evidence, the legal requirements for the admissibility of digital evidence, governance and considerations for digital forensic evidence monitoring. The chapter goes further to highlight the role that DFR should play in any organisation and discusses the background of the ISO/IEC 27043: 2015 which defines forensic readiness in different classes of digital investigations. **Chapter 3** gives a broad description of cloud computing and how the cloud can be made forensically ready, while **Chapter 4** provides a comprehensive review of the background of botnets. All the descriptions in this chapter are based on the definitions in the current and previous literature.

1.6.3 Part Three: Model

Part Three of this thesis specifically discusses the contribution of the research. It deals with the model proposed in this research and is further divided into three chapters – Chapters 5, 6 and 7. Following the reviews presented in **Chapter 2** on the need for DFR, **Chapter 5** discusses the model's requirements towards achieving DFR in the cloud. It explains the materials and methods used, as well as specific experiments that were conducted in the study. **Chapter 6** presents hypothetical case scenarios that were used to highlight the problem addressed in this thesis. These scenarios deal with fictitious scenes that provide a background on how DFR can help mitigate the effort and reduce the cost and time that will be needed by an organisation to conduct a DFI. The examples used while building the hypothetical case scenarios were used to introduce the prototype. The latter is discussed in **Chapter 8** and aims to show how DFR can be realised in the cloud environment. Finally, **Chapter 7** addresses the proposed cloud forensic readiness model proposed by the researcher.

1.6.4 Part Four: Prototype

Part Four, which explains the practical steps needed to build a prototype, consists of **Chapter 8** and **Chapter 9**. Chapter 8 introduces the design of CFRaaS prototype while Chapter 9 introduces the prototype implementation as a proof of concept on the best way to conduct DFR in the cloud environment. Chapter 8 begins by highlighting an overview of the prototype with the prototype requirements. Thereafter, the chapter 9 shows the novel architecture of the prototype and discusses and describes the prototype that the researcher developed for achieving DFR. The prototype shows how PDE can be collected from the cloud environment using a botnet with modified functionalities that is able to work in a non-malicious fashion.

1.6.5 Part Five: Conclusion

Part Five – the final part of the thesis – acts as a driver for critical evaluations of the proposals offered in this research, and shows how these propositions were implemented from the researcher's point of view. It is the concluding part of this research study and consists of two chapters, Chapters 10 and 11. A critical evaluation of the research conducted in this study is presented in **Chapter 10**, followed by a detailed evaluation of the proposed cloud forensic model, the prototype and the research questions. **Chapter 11** is the concluding

chapter that contains the novel contributions, recommendations and suggested avenues for future work.

1.7 Conclusion

Chapter 1 presented an introduction to the thesis by providing an overview of the study and stating the problem and motivation of the study. It presented the research objectives, followed by the research methodology and finally the thesis layout. The next chapter provides the reader with some background with regard to digital forensics.

“The search for truth is in one way hard and in another way easy, for it is evident that no one can master it fully or miss it wholly. But each adds a little to our knowledge of nature, and from all the facts assembled there arises a certain grandeur.”

-Aristotle-

Part Two: Background

Part Two of this thesis provides the reader with some background to the research. It consists of three chapters, Chapters 2, 3 and 4. **Chapter 2** gives a comprehensive review of digital forensics (the main focus of this research), cloud computing and botnets, and it provides the background context of digital forensics and DFR. In addition, the chapter discusses digital evidence, the legal requirements for the admissibility of digital evidence, governance and considerations for forensic evidence monitoring. The chapter goes further to highlight the role that DFR should play in any organisation and discusses the background of the ISO/IEC 27043: 2015 which defines forensic readiness in different classes of digital investigations. **Chapter 3** gives a broad description of cloud computing and how the cloud can be made forensically ready, while **Chapter 4** provides a comprehensive review of the background of botnets. All the descriptions in this chapter are based on the existing literature.

Chapter 2: Digital Forensics

2.1 Introduction

The existence of Digital Forensics (DF) as a discipline can be traced to early 1984 when the Federal Bureau of Investigations (FBI) was tasked to form the Computer Analysis and Response Team (CART) to assist with forensic examinations and technical support during digital forensic investigations of the “Magnetic Media Program”. Since inception, the DF field has emerged as the fastest growing investigative field in computing and law (FBI, 1984). To date, the need to prove that digital evidence can be admissible in a court of law has led to the establishment of standardised DF processes that have gained wide acceptance in the eyes of the legal and forensic community.

From a normative perspective, DF is concerned with the process of discovering evidential fragments, as well as acquiring, examining and analysing digital evidence by means of scientifically proven methods (Jordan, 2013). Evidential fragments may be data, hardware or software that can be used to prove the occurrence of a security incident during a DFI process. Moreover, the need to perform DFIs has been necessitated by the continued increase in the use of anti-forensic tools, digital devices, computers and network devices – all of which has led to the rise of security incidents and scenarios involving the use of these devices.

In well-documented research by Beebe and Clark (2004), DF is presented as a field that consists of phases or processes that ascertain a confirmatory analysis with regard to the absence or presence of digital evidence. During criminal investigations, answers are sought to questions about “who, what, where, when, why and how”. The essence of DF processes is to gather Potential Digital Evidence (PDE) that can be used as admissible evidence in a court of law during civil and criminal cases. For example: the standard for admissibility of scientific evidence introduced by the verdict in the Daubert case in the US in 1993 requires that evidence to be used in prosecuting criminal and civil cases should meet specific requirements on admissibility (Daubert, 1993).

The purpose of this chapter is to introduce the reader to DF as an abstract field. The sections that follow provide some background on the following: Section 2.2 discusses forensics as a science and Section 2.3 proposes a definition of DF. Section 2.4 discusses digital evidence,

followed by a discussion of the legal requirements for the admissibility of digital evidence in Section 2.5. DFIs are dealt with in Section 2.6, the digital forensic investigation process in Section 2.7, DFI process model in Section 2.8, DFR in Section 2.9, classes of digital investigation processes in Section 2.10, DFR process group in Section 2.11 and cost benefits of DFR in the organisation in Section 2.12. The chapter conclusion appears in Section 2.13.

2.2 Forensics as a Science

Forensic science has been in existence for the last three centuries (American Academy of Forensic Sciences, 2016). The term forensic originates from the Latin *forēnsis* which, according to the Oxford Dictionary is a scientific process for collecting and examining information to be used as evidence in a court of law (Oxford, 2007). Forensic science represents a wide range of disciplines that have their own practices and a range of strategies focusing on techniques, methodologies and general acceptability (Committee of Forensic Science, 2009).

The advances in science, technology and forensic science continue to gain ground and have further been extended to computing devices and networks. Hence, Almirall and Furton (2003) argue that significant scientific developments happen to be the main reason for the revival of forensic science in the 20th century. The aim of this section is to give the reader an insight into forensics as a science, which forms the basis of a digital forensics definition.

2.3 Definition of Digital Forensics

This section presents a discussion on various definitions of digital forensics and subsequently coins a substantive umbrella definition from these definitions. Since DF is a convergence of different entities like law and technology, there are currently many definitions of which none stands out as a formal one. According to Pollitt (2004), there is no single answer to the question of what DF really comprises. On the same note, Pollitt (2004) has highlighted that DF is represented by tasks that are coupled with processes in investigation. From the ideologies put forward by Pollitt (2004), DF should be ready to adapt and incorporate other technologies to a significant extent.

In a technical report for a roadmap for DF research that emerged from the first Digital Forensic Research Workshop (DFRWS) in Utica, New York in 2001, Palmer (2001) defined

DF as a process that could employ acceptable, derived and proven methods that could help while preserving, collecting, validating, identifying, analysing and documenting the way digital evidence is presented. Furthermore, he argued that these factors happen for the main reason of reconstructing how events that are found to be criminal or that may help to expect actions that are unauthorized or that may be disruptive to planned operations (Palmer, 2001).

By revisiting the definition formulated at the DFRWS, it is evident that DF is focused on all digital devices with many tasks and processes (Pollitt, 2004). Additionally, with reference to the DFRWS definition, Carrier (2003) singles out an identification phase together with analysis and identifies the goal as “to identify digital evidence using scientifically derived and proven methods to facilitate reconstruction of events”. The intuition that is presented by Carrier (2003) tries to show that all the data to be presented as digital forensic evidence has to be analysed and identified through acceptable means.

Research by Beebe (2009) argues that DF as a subject can no longer be considered a shallow discipline. She convincingly presents DF as a mainstream discipline that is able to detect the digital footprints that are left behind whenever there is interaction with computers and networks.

On the other hand, according to Lillis, Becker, O'Sullivan and Scanlon (2016), with the ever-increasing prevalence in technology, there is likelihood that digital forensic investigation process faces challenges mostly in identification, acquisition, storage and from analysis perspective. This can be attributed to the inexistence of standard and consistent DF methodology. Instead, DF methodology comprises a set of methods and tools that are developed mainly on the basis of the expertise and experiences of LEAs, system administrators and hackers. Due to this, there is need for a standardised framework to guide digital forensic process (Kohn, Eloff and Olivier, 2013).

Furthermore, the Scientific Working Group on Digital Evidence (SWGDE, 2013) defines computer forensics as a subset of multimedia and digital evidence, and states that computer forensics is a scientific process because it goes as far as examining evidence scientifically, analysing evidence and evaluating its legal admissibility.

Considering the above definitions, the researcher deduces that forensics has an investigative and a legal connotation when preparing the requirements for admissible PDE in a court of law during the presentation of legal matters. Consequently, although the digital sources might be complex environments, Carrier (2003) highlights that for legal practitioners to understand the significance of digital evidence identification and analysis, requirements for admissibility have to be included.

Based on the definitions that have been put across, the researcher therefore coined the following new definition of digital forensics:

“Digital forensics employs scientifically proven methods for purposes of electronic discovery of information that has a possibility of being admitted as probable evidence during legal, civil or criminal proceedings in a court of law.”

2.4 Digital Evidence

Different views have been put across regarding the nature of digital evidence, and why it has increasingly been used in judicial proceedings. Firstly, digital evidence was presented by Casey (2000) as “data that is stored or may be transmitted using a computer”. This supports or refutes the theory of “how an offense occurred or addresses critical elements of the offense, such as intent or an alibi”. Moreover, Casey (2000) argues that it goes further by representing data that has a possibility of linking a crime and a suspect. Secondly, SWGDE (2013) presents digital evidence as digital information that has a probative value, which is transmitted digitally. Thirdly, Carrier and Spafford (2006) view digital evidence from the perspective of security incidents, and present it in a generic definition as data that is transmitted digitally and that has a possibility of supporting or refuting a hypothesis regarding digital events during legal proceedings. Lastly, the Association of Chief Police Officers (ACPO) on digital evidence defines digital evidence as stored data and information of investigative value that can be transmitted by computer (ACPO, 2007).

Basically, digital evidence should have the potential to establish whether a digital crime was committed in a particular instance. In essence, it may well be information from digital devices that will be used as Digital Forensic Evidence (DFE) in legal proceedings. Such information may include audio files, video recordings, digital images and text files that may be related to computing devices. From a legal perspective, digital evidence must comply with admissibility requirements before it is accepted in a court of law. Admissibility encompasses the legislative rules that are set by a given jurisdiction to allow digital evidence to be presented in court. Thus it is the author's opinion that, although digital evidence is circumstantial in nature, it should be acceptable in a court of law if it has been collected by means of scientifically proven techniques. Most important of all, it should be accepted when it is able to satisfy the constitutional and statutory provisions as well as the legal requirements and aspects on admissibility of digital evidence within a given jurisdiction.

2.5 Legal Requirements for Admissibility of Digital Evidence

Digital forensics as presented by Ryan and Shpantzer (2009) is a very technical course comprising of computer science, physics and mathematics, and grounded in science. This implies that deep knowledge and professional judgment is required when countering a digital investigation process. On the same note, before accepting digital evidence in a court of law, it should be authenticated by means of a testimony. This testimony should openly establish whether the digital evidence has been handled responsibly by law enforcement agents, qualified digital forensic experts and other qualified personnel. The main reason for such authentication is to present complete proof that the evidence has not been tampered with in any way.

Because digital evidence can easily be manipulated or distorted, it has to be subjected to legal scrutiny (Marcella et al., 2007). Therefore, before digital evidence is presented in a courtroom, it should be subjected to high standards of proof. According to Jacko et al. (2003), this is done so that the legality of the potential evidence can be maintained and to determine whether prosecutorial offices and Law Enforcement Agencies (LEA) can rely on this evidence.

Even though digital evidence can be presented in many forms (i.e. hearsay, image, audio, video or text), it is important for these forms of evidence to satisfy all the requirements that

are stipulated by a given jurisdiction. Therefore, to assist LEAs and prosecutorial offices, due process should be followed when acquiring digital evidence. This can be done through the incorporation of computer forensic processes in crime scenes, jurisdictions and the courtrooms.

Due to technological advances and the subsequent proliferation of evidence, the next subsections present the considerations that are applicable in a given jurisdiction and the requirements that must be fulfilled (based on the rules of evidence) to determine whether evidence can be considered authentic and admissible. It is worth noting again that the ultimate goal of digital forensics is to enable the purported evidence to prove or disprove a fact in a court of law.

2.5.1 Legal Governance and Consideration to Evidence Collection and Monitoring

Digital evidence can be extracted either from networked computers, stand-alone computers, mobile devices, digital devices or websites. However, digital evidence first has to satisfy a number of conditions for admission in court. In meeting these conditions, the integrity of the evidence may not be affected, and trained personnel should handle the digital evidence and document all the processes for purposes of review.

The conditions for legal governance and for the acceptability of digital evidence vary across different jurisdictions; in other words, what might be accepted in one country might not be accepted in another country. According to the United States' Electronic Communication Privacy (ECPA) Act of 1986, which deals with digital evidence, intercepted electronic evidence, electronic communications and computer records must be collected to facilitate prosecution in the judicial system unless one of the parties has given prior consent. For example, the US's Wiretap Act prohibits all acts of interception of electronic data communications, but allows statutory exceptions to be considered when activities affect incident response procedures. Moreover, the Wiretap Act allows the use of tools such as TCPDump and Etherpeek to collect digital evidence that may be used to intercept and examine content.

Additionally, the Stored Communications Act (SCA) 18 US Code 2701 states that it becomes unlawful to intentionally access an electronic facility without authorisation (Jarret and Bailie, 2002). However, Section (c) of Code 2701 provides for exceptions if the entity or the person provides a wire or an electronic communication service if there is consent, for law enforcement purposes, provider exception or an emergency situation (Scolnik, 2004); (Scolnik, 2009).

The good practices guide for digital evidence published by the United Kingdom's Association of Chief Police Officers (ACPO) highlights the principle that all digital evidence has to be subjected to the rules and laws that apply to documentary evidence. Moreover, the ACPO guide highlights the provision that before digital evidence is captured from a scene, the people responsible for seizure should have the necessary equipment and they should know the potential sources of evidence (ACPO, 2012).

On the other hand, the South African Government Gazette highlights the following legislation that contains information regarding the right to privacy of user information: the Electronic Communications and Transactions (ECT) Act (Gereda, 2006), the Protection of Personal Information (PoPI) Act (PoPI, 2013) and the Regulation of Interception of Communications and Provision of Communication-Related Information Act (RICA) (RICA, 2001). The purpose of the ECT Act is to regulate users' electronic communications and transactions; while the PoPI Act strives to maintain the right to privacy by safeguarding personal information as prescribed by the SA Constitution. The purpose of the RICA Act is to regulate the interception and monitoring of communication. Section 14a of the ECT Act on information admissibility and retention highlights that, "where a law requires information to be presented or retained in its original form, that requirement is met by a data message" (Gereda, 2006).

Furthermore, Section 15 of the ECT Act states that in no case should the prosecution apply the rules of evidence to deny admissibility of a data message during legal proceedings. A constitutional provision highlighted in Chapter 4 of the PoPI Act provides an exemption that allows for the violation of information privacy if this is in the interest of national security. Consequently, Section 6 of the RICA Act states that an employer is allowed to intercept indirect communication; however, such interception or monitoring may occur only under the following circumstances:

1. If monitoring is done to investigate or help with the detection of the unauthorised use of the telecommunications system that is being provided by the employer.
2. If the system controller has obtained consent – whether expressly or implicitly.
3. If there is partial use of the telecommunications system or if the telecommunications system has a connection partly or wholly to a specific business.
4. If the system controller makes an effort to inform the person intending to use the system that indirect communications may be intercepted (Schoeman & Jones, 2004).

The above provisions allow the employer to alert employees to the fact that they are being subjected to monitoring. Furthermore, the ECT, PoPI and RICA Acts may be disregarded if monitoring or interception occurs for law enforcement purposes (Scolnik, 2009);(Gereda, 2006).

2.5.2 Requirements for Admissibility of Digital Evidence

According to Ryan and Shpantzer (2009), digital forensic evidence qualifies to be admissible if it satisfies the conditions of being relevant and being able to be derived through a scientific method. Moreover, the scientific process should be supported by a validation process. These requirements are used to determine the legality of digital evidence so that a hypothesis can be formulated that might help to arrive at a justified conviction or exoneration of wrongfully convicted defendants. With reference to the Federal Rules of Evidence (FRE) on legal governance in respect of the admissibility of digital evidence, exceptions are highlighted when the case involves records dealing with issues that are pertinent to computer forensics. An example if this are cases that require the testimony of the expert witness. Detailed explanations are given in the next two subsections.

Based on the FRE, Nolan, O’Sullivan, Branson and Waits (2005) argue that if a company chooses to do logging as a practice, then the logs should be considered admissible when presented in a court of law. However, the FRE 1001(3) state that whenever there is data stored in a computing device and it is readable by sight or if it reflects the data accurately, then it is deemed to be original data. The following examples show the case law reviews in which digital evidence has been portrayed as admissible.

2.5.2.1 Case Law Review: Daubert v. Merrell Dow Pharmaceuticals

In 1993, the US Supreme Court used two court cases – Frye v. United States, 293F and Daubert v. Merrell Dow Pharmaceuticals – to determine the standard for admitting expert testimony in federal courts. They did this by subjecting new scientific techniques to admissibility rules (Zonana, 1994). Rule 702 of the Federal Rules of Evidence (FRE), which governs the testimony by expert witness, states that “A witness who is qualified as an expert by knowledge, skill, experience, training, or education may testify in the form of an opinion or otherwise if:

- a) the expert’s scientific, technical, or other specialised knowledge will help the trier of fact to understand the evidence or to determine a fact in issue;
- b) the testimony is based on sufficient facts or data;
- c) the testimony is the product of reliable principles and methods; and
- d) the expert has reliably applied the principles and methods to the facts of the case (Fed. R. Evid. 702), (Zonana, 1994).

With regard to the Daubert v. Merrell Dow Pharmaceuticals case review – if the principle of which evidence is given in a court of law has general acceptance in a particular field, then the scientific evidence may be admissible. The petitioners in this case were the parents of Jason Daubert and Eric Schuller who sued the respondent Merrell Dow Pharmaceuticals because both babies had been born with serious birth defects. The petitioners blamed these defects on the mothers’ use of Bendectin, a drug used to treat nausea. Even though the respondent claimed that the petitioners could not provide admissible evidence that was able to prove beyond doubt that Bendectin caused the birth defects, the petitioners were able to oppose the respondent’s judgments. They did this by bringing in eight qualified experts who later admitted that Bendectin could well cause birth defects, based on the experiments conducted in animals. The experts were also able to reach this conclusion based on previously published epidemiological research that showed similarities with Bendectin and the causality it has on birth defects.

The court found both the petitioners’ study that was used to link Bendectin to animals and the epidemiological studies as inadmissible, inter alia due to the divergence of the practices of the qualified experts. Congress adopted the Federal Rules of Evidence in 1975 and in Rule

702 of the Federal Rules of Evidence recommends that admission of scientific evidence should be subject to reliable inquiry, general observation status, falsifiability, refutability and testability (Watkins, 1994).

2.5.2.2 Case Law Review: United States v. Mosley

Based on the video surveillance of a bank robbery, an FBI agent and expert in photographic comparisons testified against Mosley who was being charged with six counts of bank robbery. The video was subjected to digital image processing, a procedure that sharpened and enhanced the images. The FBI agent identified a mark on the robber's face, and thus he was able to compare this mark with the mark on Mosley's face in the mug shot. Even though the defendant argued that the court was wrong to admit such analysed digital evidence, the court stated that the evidence was admitted properly and it could help the jurors (United States v Mosley, 1994).

Regarding the expert witness's testimony, the court ruled that digital evidence may be accepted provided that it is properly handled, not manipulated when seized and handled by forensically competent personnel who maintain the chain of custody. In this context, the chain of custody is a process that shows a roadmap of movement and location of evidence, which begins from the seizure of evidence to its presentation in a court of law. According to Ngomane (2010), a number of requirements may be considered by the court of law to increase chances of admissibility of digital evidence. These requirements include the following: Evidence should be in its original form, reliable, authentic, and legal.

The reader has been introduced to the important requirements that are needed in order for digital evidence to satisfy admissibility across diverse jurisdictions. Most importantly, this evidence is usually used to prove or disprove a fact during a digital forensic investigation process. As a result, in the next section the reader is introduced to a discussion on digital forensic investigations.

2.6 Digital Forensic Investigations

A Digital Forensic Investigations (DFI) is concerned with the retrieval, acquisition, analysis and examination of potential digital evidence (PDE) in such a way that the evidence will be accepted in a court of law. Conventionally, evidence presented in court is more inclined

towards the field of physical forensics. However, the theories that are developed and tested during DFIs are associated with the processes that use science and technology. These theories can therefore be presented in courtrooms to help answer questions about how digital events occurred (Carrier & Spafford, 2004).

According to Ioeng (2006), a DFI is a process that is used to decide whether extracted information considered as digital evidence is relevant and whether a court or jury can use it to draw conclusions. It is the task of the digital forensic investigator to extract factual data that can be used for judicial review purposes. A DFI itself is a reactive process that is used to obtain PDE after a potential security incident has been detected. Rowlingson (2004) agrees that a DFI is implemented as a post-event response – in other words after a potential information security incident occurred. Carrier (2004) prefers a DFI over a physical forensics investigation because a DFI answers more limited questions, identifies an object and determines the class of that object as opposed to physical forensics which is a mere physical crime scene investigation.

Carrier and Spafford (2004) in turn present a DFI as a scientific process that involves the examining of digital objects. The examined objects are subsequently used to develop and test theories that can be used in courtrooms to prove or disprove facts. Often the main focus of questions about the alleged security incident is the examination of a digital device in order to extract digital evidence. The forensic examination of digital devices therefore plays a crucial role when staging a DFI.

The outcome of any DFI relies on the scientific techniques that are used to extract PDE and the possibility that the extracted evidence may be admitted in a court of law. The main objective in this context is to rely on an examination of digital devices used with a view to extracting potential digital evidence. This implies that being in possession of a digital device and not being able to conduct examination using scientifically proven methods, it becomes trivial. This implies that DFI process should be conducted methodically (Kohn et al., 2013).

2.7 Digital Forensic Investigation Process

Digital Forensic Investigation Process (DFIP) represents the entire range of activities that are performed during a computer forensic investigation (Yusoff, Ismail & Hassan, 2011). In

order for a DFI to be launched, more comprehensive and proven methods have to be used so that any evidence that arises from such investigation may satisfy the requirements for being accepted during litigation. Already in traditional DFIs, it was a requirement for digital evidence to satisfy a number of conditions before it could be considered admissible. Figure 2.1 illustrates the classes of digital investigation processes pertaining to the ISO/IEC 27043. The processes shown in this figure are discussed later in this chapter.

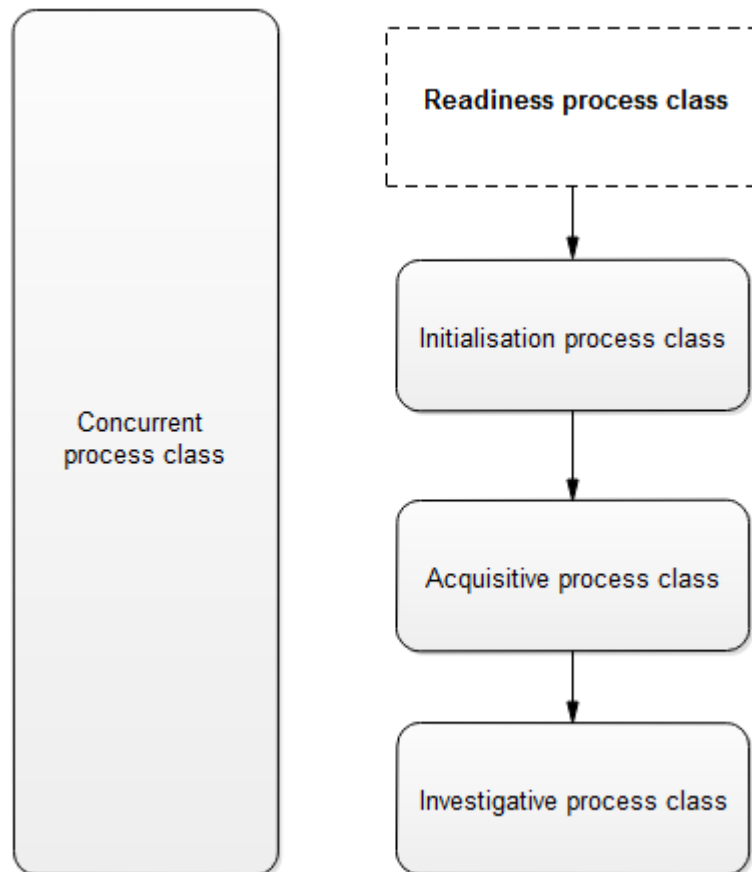


Figure 2.1 Classes of Digital Investigation Process (Source: ISO/IEC 27043:2015)

On the same note, Pollitt (2007) highlights that a suitable path has to be taken for digital evidence to be admissible. This path may include the physical context that encompasses the media, the logical context that encompasses the data being examined, and the legal context information that leads one to evidence. The DFIP fits into the initialisation, acquisitive, investigative and concurrent classes of DFIs processes that are shown in Figure 2.1. Forensic readiness is also included in this class, but it will be discussed later in the chapter.

2.7.1 Initialisation Process Class

An Initialisation Process Class (IPC) is a process that deals with the way a digital investigation starts. In fact, it is the initial starting point of the digital investigation process. The processes in an IPC include incident detection, first response, planning and preparation (ISO/IEC 27043, 2015). “Incident detection” is the application of various methods for the purpose of identifying intrusions, compromise of security, attacks or violations of user policies. “First response” represents the action that is taken by the incident response team to determine the root causes of a security incident, and “planning” and “preparation” are mechanisms for getting ready for a physical investigation.

2.7.2 Acquisitive Process Class

The Acquisitive Process Class (APC) deals with how a case may be investigated physically and how Potential Digital Evidence (PDE) should be handled. It includes PDE identification, PDE acquisition, PDE transportation and PDE storage. PDE identification shows the sources from which digital evidence is likely to be extracted, while PDE acquisition typically describes the mechanism of acquiring data by creating an exact copy on storage media while preserving their integrity. Transportation and storage of potential evidence show how the evidence is handled and how it is stored before the start of the digital investigative process.

2.7.3 Investigative Process Class

This class deals with ways of uncovering PDE. The processes, according to ISO/IEC 27043: 2015, include the following: PDE examination and analysis; digital evidence interpretation; reporting; presentation, and investigation closure. Examination and analysis of PDE is an assessment made to determine what is significant during the investigative process. Interpretation, on the other hand, is the ability to show how relevant digital evidence should be while conducting an investigation. Reporting is a process of producing examination notes in the form of remarks, conclusions and results emanating from the analysis of potential evidence. Finally, presentation allows the giving of the findings of an investigation, while investigation closure terminates the investigative process.

2.7.4 Concurrent Process Class

The Concurrent Process Class (CPC) as defined by ISO 27043: 2015, happens alongside other processes. As the other process classes occur, the CPC allows them to be executed so that PDE may be assured of being admissible in the legal system. The CPC consists of a number of processes, namely: How authorisation is obtained; how processes are documented; how the management of information flows in investigative processes is handled; tracking the roadmap of events through maintaining the chain of custody; how the collected evidence is digitally preserved; and lastly, how a link between an incident and a perpetrator is established during a physical forensic investigation.

2.8 Digital Forensic Investigation Process Models

According to Kohn, Eloff M and Eloff (2013), the DFIP models help to explain how specific the evidence extracted from digital devices is. The origin of the DFIP can be traced back to the early theories proposed on computer forensics. Notwithstanding that, ideal process models are supposed to identify the steps that are necessary to achieve investigative goals. The process models are supposed to succeed even when there have been technological changes. Ever since the first Digital Forensic Research Workshop (DFRWS) conference in 2001 in Utica, New York, up to the time of writing this research thesis, there has not been an accepted standard digital forensic process model that is able to support Digital Forensic Investigations (DFIs). Due to this inability to standardise, a number of DFIP models with different phases and tasks are defined. More often than not, a process model follows a number of iterations and these iterations represent the tasks and activities involved when conducting DFIs. Table 2.1 shows different proposed DFIP frameworks and models. These models have been formulated on the account of identification, preservation, analysis and presentation. The legal establishments and law enforcement agencies always rely on the proposed DFIPs to provide factual information as potential evidence. As a result, PDE may be admitted in courtrooms provided that the investigation processes followed a set of scientifically proven and accepted methods.

Reith, Carr and Gunsch (2002) have for example put forward the Abstract Digital Forensic Model (ADFM), which is made up of the following processes: Identification; preparation; approach strategy; preservation; collection; examination and analysis. Next, Carrier and Spafford (2003) proposed the Integrated Digital Investigative Process (IDIP) which involved

the following phases: Readiness; deployment; physical crime scene investigation; and review phase. This was followed by the Enhanced Digital Investigation Process (EDIP) model (EDIP) of Baryamureeba and Tushabe (2004), which included the following phases: Readiness; deployment; trace back; dynamite, and review.

Another researcher, Ciardhuain (2004) proposed an Extended Model of Cybercrime Investigation (EMCI) with the following phases: Awareness; authorisation; planning; notification; evidence search; collection; transportation; storage; examination; hypothesis; proof of hypothesis, and information dissemination. Next, Kohn, Eloff and Olivier (2006) also suggested a Framework for a Digital Investigation (FDI) with preparation, investigation and presentation phases, which allow for the incorporation of the previously proposed frameworks.

According to a document prepared by Kent, Chevalier, Grance and Dang (2006) for the National Institute of Standards and Technology (NIST), named a Guide to Integrating Forensic Techniques into Incident Response (GIFTIR) (Special Publication 800-86), a four-phase forensic process with collection, examination, analysis and reporting phases was proposed. Also, Perumal (2009) proposed a Digital Forensic Model (DFM) based on the Malaysian Investigation Process (DFMMIP) with the following phases: Planning; identification; reconnaissance; analysis; result; proof and defence, and diffusion of information. In addition, Agarwal (2011) put forward the Systematic Digital Forensic Investigation Model (SDFIM) that has the following phases: Preparation; securing of the scene; survey and recognition; scene documentation; communication shielding; evidence collection, preservation, examination, analysis, presentation and review. Table 2.1 shows various proposed DFIP models.

Table 2.1 Existing Digital Forensic Investigation Process Models

s.no	Year	Model/Framework	Authors	Phases
1	2001	National Institute of Justice (NIJ)	Ashcroft	5
2	2001	DFRWS Model	Palmer	7
3	2002	Abstract Digital Forensic Model	Reith, Carr & Gunsh	9
4	2003	The Integrated Digital Investigative Process	Carrier & Spafford	17
5	2004	Enhanced Digital Investigation Process Model(EDIP)	Baryamureeba & Tushabe	4
6	2004	An extended Model of Cybercrime Investigation	Ciardhuain	13
7	2004	A Hierarchical, Objectives-Based Framework	Beebe & Clark,	6

		for the Digital Investigations Process		
8	2006	Framework for a Digital Investigation	Kohn, Eloff & Olivier	4
9	2006	The Four-phase Forensic Process	Kent, Chevalier, Grance & Dang	4
10	2009	Digital Forensic Model based on Malaysian Investigation Process	Perumal	7
11	2011	The Systematic Digital Forensic Investigation Model	Agarwal	11
12	2012	Harmonised Digital Forensic Investigation Process Model	Valjarevic & Venter	12

In conclusion, Valjarevic and Venter (2012) proposed a comprehensive, iterative and multi-tiered Harmonised Digital Forensic Investigation Process Model (HDFIPM) with the following phases: Incident detection; first response; planning; preparation; incident scene documentation; potential evidence identification; potential evidence collection; evidence transportation; evidence storage; analysis; presentation, and investigation closure.

The HDFIPM actually formed part of the ISO/IEC 27043. The HDFIPM that has been mentioned in ISO/IEC 27043: 2015 mainly consists of three processes, namely the initialisation, acquisitive and investigative processes. This was discussed in detail in the earlier sections of this chapter.

The digital forensic investigation processes that have been discussed in this chapter are usually employed during the DFI process; however, Digital Forensic Readiness (DFR) is also part of this process. Even though DFR might be optional, it still forms part of this process and is therefore discussed in the next section.

2.9 Digital Forensic Readiness

Digital Forensic Readiness (DFR) presents a proactive process that is used to manage security incidents before they occur. Security incidents are risks or vulnerability that may occur in any organisation. Consequently, DFR plays an important role in preventing or detecting the possibility of security incidents. Beebe and Clark (2004) describe this as a preparation phase, which has the goal to maximise digital evidence availability through response, detection and deterrence. Normally, organisations become wary of the cost of DFR, but according to Grobler and Louwrens (2007), proactive digital forensic management makes

business structures well due to the ability to retain essential data. This essential data is what may proactively be used if a potential security incident is detected. The propositions in this research thesis applies proactive processes that are used to retain and manage potential evidence.

The ultimate goal of this section is to familiarise the reader with essential aspects of DFR, how DFR fits in among the classes of the digital investigation process that are mentioned in ISO/IEC 27043, and the essence of the Readiness Process Groups (RPGs) in the digital investigation context. Furthermore, this section shows the importance of enforcing DFR in any organisation. In the next subsection, a definition of DFR is introduced.

2.9.1 Defining Digital Forensic Readiness

A number of opinions on what exactly DFR is have been put forward. Thus, Tan (2001) defined the concept of DFR by means of two objectives and since then a number of researchers have developed different intuitions regarding it. The work that is presented in this research thesis is strongly inclined towards Tan's (2001) and Rowlingson's (2004) objectives.

Tan (2001) presents the objectives of DFR as maximising an environment's ability to collect credible digital evidence and minimising the cost of digital forensic investigations during an incidence response. Further, Tan's views revolve around how an organisation can be forensically ready through the identification of the key elements of DFR. Rowlingson (2004), on the other hand, sees DFR as a corporate goal that facilitates an organisation's ability to use digital evidence when needed. Additionally, DFR as viewed by Rowlingson is inclined towards the organisational perspective; hence it is authors' opinion that at least every organisation requires an investigative capability. For the purpose of this research thesis, DFR is defined as follows, based on Rowlingson's (2004) organisation perspective:

“Organisational investigative capability of reducing cost and time of performing a digital forensic investigation by retaining critical and sensitive information that is related to possible security incidents.”

The definition above goes on to state how DFR is being achieved in the cloud environment, which is discussed in the subsequent chapters of this research thesis.

Having looked at DFR, it is important to know the goals of DFR in an organisation. Hence, in the next section, the goals of DFR in a business environment are discussed.

2.9.2 Goals of Digital Forensic Readiness

Traditionally, the process of digital forensics begins when a security incident or crime has occurred. However, from an organisation's perspective, activities that involve proactive forensics are a major requirement as they limit potential business risks. When an organisation is not able to respond to security incidents, then the prevailing security incidents will eventually affect that organisation in terms of growth and performance. Thus, when actions of assessing and managing security risks are planned in any organisation, one needs to ensure that there are effective security-incident management strategies as well as an implementation priority plan for post-incident response approaches.

On the same note, the inclusiveness of DFR as a preparation phase and a platform for proactive activities represents a phenomenon where any given organisation may have sufficient assurance on the effectiveness of forensic readiness during a DFI. Rowlingson (2004) proposed the goals of forensic readiness for an enterprise as follows:

- To gather admissible evidence legally and without interfering with business processes.
- To gather evidence targeting the potential crimes and disputes that may have an adverse impact on an organisation.
- To allow an investigation to proceed at a cost in proportion to the incident.
- To minimise interruption of the business by any investigation.
- To ensure that evidence makes a positive impact on the outcome of any legal action.

Furthermore, Rowlingson (2004) makes recommendations to organisations to limit the future impact of evidence. He recommends that an organisation should establish policies for the securing, storing and handling of potential digital evidence that may be required in future.

Having looked at the goals of DFR in an organisation, our focus now shifts to the DFR process class that is explained in the next section.

2.10 Digital Forensic Readiness Process Class

This section contains a discussion of the Digital Forensic Readiness (DFR) process class, which has also been mentioned in the ISO/IEC 27043. This class deals with pre-incident investigation processes that cover the following aspects:

- Scenario definition.
- Identification of potential digital evidence sources.
- Planning of pre-incident gathering.
- Storage and handling of data representing potential digital evidence.
- Planning pre-incident analysis of data representing potential digital evidence.
- Planning incident detection.
- Defining system architecture.
- Implementing system architecture.
- Implementing pre-incident gathering.
- Storage and handling of data that represents potential digital evidence.
- Implementing pre-incident analysis of data that represents potential digital evidence.
- Implementing incident detection.
- Assessment of implementation; and assessment of results.

Based on this discussion, it is necessary to introduce the reader to the DFR process group as is discussed in the next section.

2.11 Digital Forensic Readiness Process Group

This section presents a discussion of the DFR process group. This aspect justifies discussion because core research that has been presented in this thesis is based on the DFR process group of ISO/IEC 27043. Moreover, this section also shows how the ISO/IEC 27043 handles DFR.

Forensic readiness in the Readiness Process Groups (RPGs) has been defined as a process that precedes incident detection (see Figure 2.2), in other words it is a proactive process. Nevertheless, ISO/IEC 27043 defines RPGs that can maximise the potential use of digital evidence in order to reduce the cost of conducting a DFI process. This is done as a measure to improve the level of information security in organisational systems. The RPGs are classified

into three groups: Planning process group, implementation process group and assessment process group.

Each of the three main groups contains a number of sub-processes. The *Planning Process Group (PPG)* has sub-processes that perform the following tasks: defining the scenario; identifying PDE sources and planning pre-incident collection; storage of PDE; pre-incident analysis planning; planning of incident detection, and defining the system architecture. The *Implementation Process Group (IPG)* implements all of the PPG activities, while the *Assessment Process Group (APG)* assesses the implemented process and implements assessment result processes. The concurrent process shown by the arrow pointing downwards in Figure 2.2 indicates that the processes are executed as continuous processes.

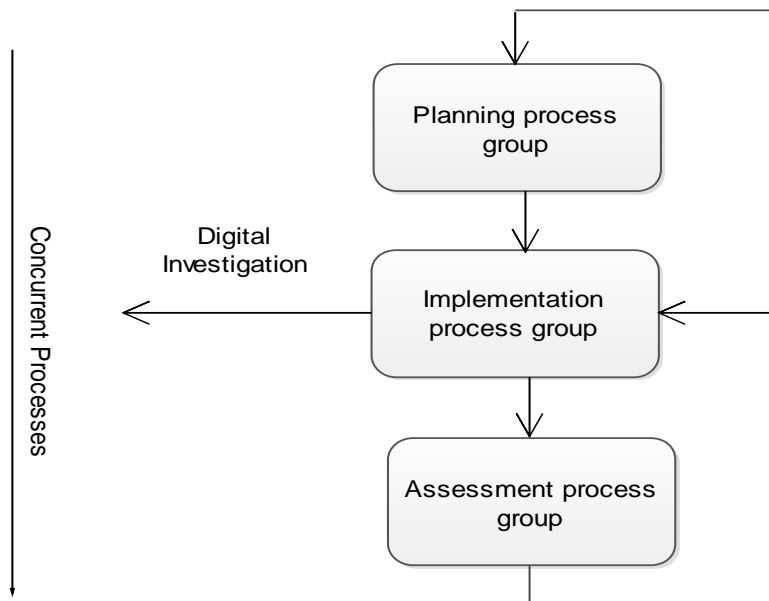


Figure 2.2 Readiness Process Groups (Source: ISO/IEC 27043:2015)

Implementing DFR in any organisation has definite cost implications, for example, when organisations are required to plan before potential security incidents can be detected. However, digital forensic readiness also holds great cost benefits for organisations, as will be discussed in the next section.

2.12 Cost Benefits of DFR in an Organisation

The absence of DFR in organisations allows the rise of fraudulent activities, which may have huge costs for an organisation during a DFI. The cost of performing a DFI can be reduced if a

comprehensive readiness framework is in place. Grobler, Louwen and Von Solms (2010) state that organisations need to introduce a comprehensive framework that will give assistance in the implementation of a DFR programme. Rowlingson (2004) also notes that organisations can reduce the cost of digital crimes if they implement the necessary measures to collect and retain digital evidence even before incidents are detected.

2.13 Conclusion

This chapter started off with a discussion of digital forensics, after which the researcher gave a definition of digital forensics and highlighted how forensics is viewed as a science. Digital evidence, as well as the legal requirements for admissibility of digital evidence, was discussed and then the reader was introduced to legal governance and the conditions for evidence collection and monitoring. In addition, the reader was introduced to the legal considerations and requirements that digital evidence has to meet across varying jurisdictions in order to be admitted in a court of law. Exploring these legal requirements and considerations was an important step in helping the reader to understand the territorial provisions that exist across different jurisdictions.

Afterwards, a discussion followed on Digital Forensic Investigation (DFI) and the DFI Process (DFIP) model. Finally, the concept of DFR was introduced and the researcher highlighted the digital forensic Readiness Process Groups (RPGs), goals of DFR and the cost benefits of implementing DFR in a business environment.

In the next chapter, the reader is introduced to cloud computing aspects.

Chapter 3: Concepts of cloud computing

3.1 Introduction

Is it a new wine or just a new bottle? This question represents the sentiments that have been expressed by Agrawal et al. (2010) on cloud computing, simply because it looks like the normal “client-server architecture” where a mainframe computer acts as a distributor of services to various nodes. Additionally, the cloud has been exemplified by internet technology, distributed mega data centres, powerful servers and the proliferation of more connected devices.

Nevertheless, cloud computing has emerged as one of the most talked-about technologies when it comes to business in recent times, and this has caused enterprises to shift their focus to the benefits and cost effectiveness of the services that the cloud offers. The move by organisations to move their data and applications to the cloud has been inspired by reduced operational costs and increased benefits. However, before an organisation sends its data to the cloud, it should have achieved organisational maturity, because one might not have an idea where your data resides in the cloud. Organisational maturity is a situation whereby the readiness of a given organisation is expressed in terms of the perceptions of its people, the processes and the data in that organisation.

The survey on cloud computing for business conducted by Gartner Inc. (2014) highlights that cloud computing promises economic advantages, speed, agility, innovation and elasticity. Moreover, the survey predicts that by 2016-2017, 20% of all the services offered by the cloud will be consumed by internal and external brokerages. Users are also anticipated to have trouble deciding which cloud computing environment to choose and whether to trust the security and privacy involved (Gartner, 2014).

The infrastructure of cloud computing allows unlimited resource usage for consumers by offering on-demand shared resources through multi-tenancy. The provision of these resources through a cloud model furthermore ensures proper service availability for everyone in the cloud environment. As the cloud is an analogy of the internet, its virtual resources are offered through the internet and the services get delivered from data centres located across the world.

The focus of this research thesis is to show comprehensively how the cloud can be made forensically ready for DFIs.

The remainder of the chapter is structured as follows: A definition of cloud computing is given in Section 3.2, after which Section 3.3 discusses the cloud computing architecture. Next, Section 3.4 discusses the role of the Cloud Service Provider (CSP), Section 3.5 deals with virtualisation and cloud computing, and Section 3.6 discusses the adoption of DF. This is followed by a discussion of DFR in the cloud in Section 3.7, and the chapter concludes with Section 3.8.

3.2 Defining Cloud Computing

A number of definitions have been put forward regarding how the cloud is perceived by different researchers. The National Institute of Standards and Technology, NIST, defines cloud computing as a model that is based on on-demand network access that can be run over resources that are configurable (Mell & Grance, 2011). These resources can be managed efficiently by Cloud Service Providers (CSPs). The on-demand feature of the cloud implies that the cloud user may gain access to a virtual instance whenever he/she needs it and afterwards cease to use it when access is no longer needed. The resources that can be provided in this context include applications, storage services, network services and network servers, and the cloud model is able to provide these services in an effective and convenient manner possible.

Armbrust et al. (2009) present a Berkeley view of cloud computing as those services and applications that are provisioned over the internet, including the systems that represent the software and hardware contained in the data centres that are able to deliver these services. The data centre that contains the hardware and the software is known as the cloud (Armbrust et al., 2009).

Barkley, Stanoevska-Slabeva and Wozniak (2009), together with Wozniak and Ristol (2009) summarise the features of the cloud by pointing out important aspects that make the cloud a preferable mode of computing:

- The cloud's virtual, scalable and on-demand nature.

- The services provided by the cloud can be presented through a web browser or via a defined Application Programming Interface (API).
- Infrastructure resources (including hardware, system software and storage) and applications are provided by X-as a Service (XaaS), in other words the services are offered by an independent provider.

Based on the definitions highlighted above, the researcher has been able to coin the following definition of cloud computing in this research thesis:

“Cloud computing is a scalable infrastructural paradigm that is coupled with internet-centric software that allows people to deploy, manage and access technology-enabled services that are provisioned over the internet.”

All the aforementioned definitions are focused towards the same goal and the same problem, but the most important aspects of cloud computing include the scalability, elasticity and provisioning of services, and the on-demand nature of the cloud. These aspects are normally employed as basic building blocks of a cloud computing architecture and therefore cloud computing is discussed in more detail in the next section.

3.3 Cloud Computing Architecture

The architecture of the cloud is represented in the form of different levels of abstraction with a different set of architectural elements that constitute the structure of the system. It comprises different cloud resources, software components, services and the relationship between these services. Cloud computing architecture is described according to five essential characteristics, three service models and four deployment models. Figure 2.1 gives the visual representation of the NIST definition of cloud computing (Cloud Security Alliance, 2009), after which each of the components of the model are explained in detail.

3.3.1 Essential Characteristics of Cloud Computing

A study by researchers Gong, Liu, Zhang, Chen and Gong (2010) presents the characteristics of cloud computing based on the following essentials: The cloud has its' own conceptual, technical, economic and user experience characteristics. Additionally, it has a service-

oriented conceptual characteristics that are able to abstract details of how implementation are done. Through virtualisation, the cloud architecture is able to be abstracted and elements of the underlying architecture may be accessed by the cloud user.

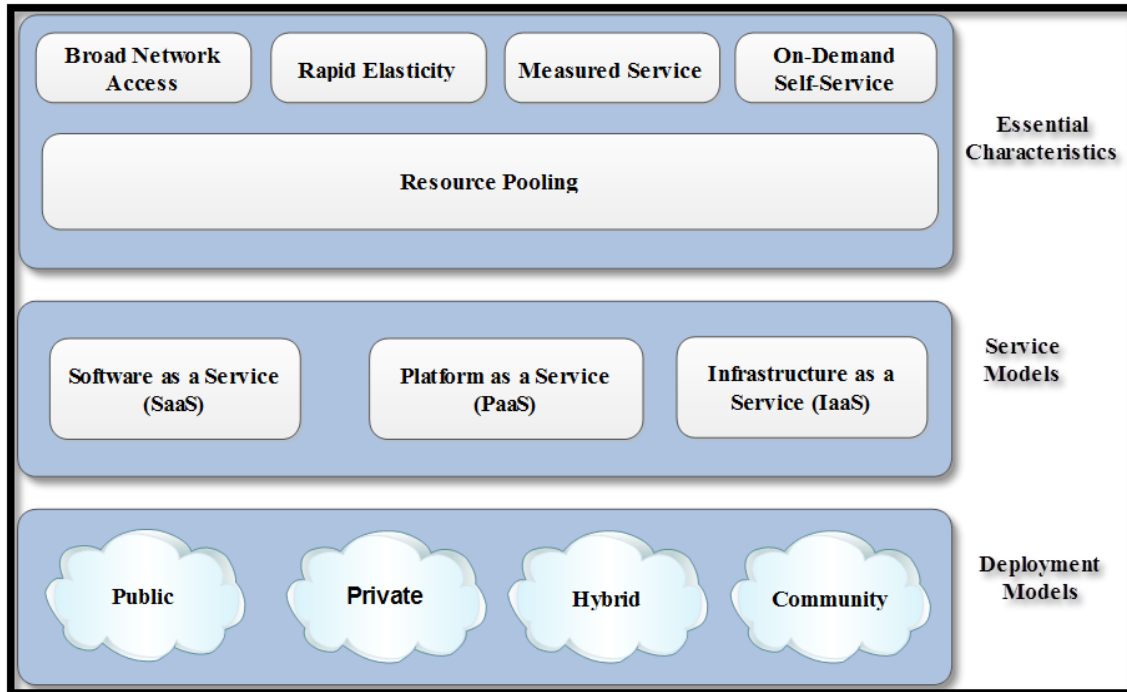


Figure 3.1 A Visual Model of NIST Cloud Computing Definition (CSA, 2009)

Loose coupling and strong fault tolerance are presented as technical characteristics where the platform used is an abstract layer that is able to isolate various applications running on the cloud – hence it is presented as a client-server model. This is followed by a business model with economic characteristics and ease of use as a specific user-experience characteristic. Subsequently, a study presented by Cloud Security Alliance (CSA) highlights five essential characteristics of cloud computing architecture as shown in Figure 3.1. These characteristics allow cloud services to be reached at and from any given end-point (tablet, mobile device, PC, etc.). Since cloud services are accessible wherever there is a network infrastructure, cost is reduced.

In the subsequent sections, all the components within this cloud computing model will be discussed in detail. The following characteristics are highlighted in the NIST’s special publication 800-145 (Mell & Grance, 2011).

3.3.1.1 On-Demand Self-Service

According to Olive (2011), the cloud service should always be available and it should be possible to modify the service received by the client organisation. Cloud consumers must be able to scale the infrastructure that they need without interfering with other host operations. This provision will allow cloud consumers to access cloud services through a controlled online panel whenever they require services, without requiring any human interaction. Bachiega et al. (2014) agree that on-demand computational services should be provided without human intervention or without the provider.

3.3.1.2 Broad Network Access

Cloud resources are made available through devices that enable the resources to be accessible from various locations. Consumers prefer this characteristic because they are able to have online access from a wide range of locations that go beyond any specific network, as well as from any computing device. According to Sriram and Khaejah (2010), cloud resources may be accessed over a given network by means of heterogeneous devices like laptops and other mobile devices. Olive (2011) also views it as a mechanism that is typically accomplished using the built-in web browsers' ubiquitous device.

3.3.1.3 Resource Pooling

A resource pool can be an object that has set of resources to be managed (Gulati, Shanmuganathan, Holler & Ahmad, 2011). This depicts an instance whereby multiple organisations are able to share the physical cloud infrastructure (Olive, 2011). Resource pooling is mainly employed to remove obstacles from the paths that clients use to access resources. Additionally, when it is done to servers, the time spent in maintaining resources is reduced and the inhibitors in the cloud are also removed. Within the cloud environment, resource pooling can be done in respect of the following services: bandwidth, storage and processing. Consequently, a resource pool as presented by Gulati et al. (2012) is used when dividing and sharing the aggregate range of a group of Virtual Machines (VMs) or users.

3.3.1.4 Rapid Elasticity

Elasticity is the ability to provide scalable services in the cloud environment. In reference to this concept, Herbst, Kounev and Reussner (2013) state that elasticity is related to the ability of a system to adapt to changes in workloads and demands. Clients are able to make requests of services in the cloud and the providers need to provide unlimited resources by allocating and de-allocating resources through scalable provisioning. This implies that it is the amount of resources provided to the user that may be changed as the resource demand changes (Herbst et al., 2013).

3.3.1.5 Measured Service

All the cloud components that are offered to clients are precisely measured and configured to deliver services according to the clients' expectations. This allows usage of services to be monitored and controlled for effectiveness. Mahmood (2011) presents a measured provision that is able to optimise the way the resources are allocated for purposes of billing, which eventually reduces the cost of provisioning new resources. In this case, a client is allowed to pay the Cloud Service Provider (CSP) or the hosting party only for the exact resources that are consumed.

The above-mentioned incentives are presented as significant factors employed by a majority of organisations that have enforced cloud computing. However, they depend on the type of cloud computing service model that is preferred by different organisations in order to provide services to their clients.

Now that the reader has been familiarised with the essential characteristics of the cloud, the different cloud computing service models are discussed in the next section.

3.4 Cloud Computing Service Models

Cloud computing service models are resources that are delivered over the internet and hence they are also regarded as web services. The main role of these services is to maximise the benefits that are being propelled by cloud computing. The service models have been categorised into three groups: Infrastructure as a Service (IaaS); Platform as a service (PaaS) and Software as a service (SaaS) (see Figure 3.1).

3.4.1 Infrastructure as a Service (IaaS)

IaaS as a provision gives the cloud consumer the ability to process, store, deploy and use networks and other computing resources that include the Operating System (OS) and other applications as on-demand services. According to Sriram and Khajeh-Hosseini (2010), IaaS is presented as a low-level abstraction through which users are able to access the infrastructure by the use of VMs. The resources are fully outsourced, which means the consumer does not need to buy equipment like virtualisation, storage, hardware, server or networking components and software. IaaS supports multi-tenancy, which implies that many users can use the same piece of hardware concurrently. Also, resources in IaaS are distributed as a service, which allows operation support at any given location.

3.4.2 Platform as a Service (PaaS)

PaaS gives the consumer a way to deploy applications to the cloud by providing services and provider tools. This is mainly a computing platform that allows cloud clients to effectively and quickly develop, test and deploy web applications without having to maintain the software or the underlying infrastructure. PaaS allows different web-based interface tools that help to create applications from different scenarios. According to Boniface et al. (2010), PaaS user-developer, PaaS provider and PaaS hoster are some of the components that are contained in PaaS stakeholder. Nevertheless, the researcher maintains that PaaS is a multi-tenant architecture that allows many tenants operating concurrently to make use of the same development environment. PaaS can easily be integrated with web services and databases; it supports project planning, team collaboration, subscription management, and finally, it is scalable and supports failover and load balancing.

3.4.3 Software as a Service (SaaS)

SaaS is software that is hosted off-premise and that is delivered over the internet (Godse & Mulik, 2009). SaaS is normally managed from a central location by a CSP or any other vendor and software services are then deployed through the network, normally the internet. Users can access this service through the client devices and through the web browser as thin clients. SaaS is normally delivered as a utility service based on a pay-per-use tariff, where the SaaS CSPs remain the owners of the software who are able to store the software system and user data in a centralised server (Guo, 2009).

Having looked at the cloud computing service models, the focus now shifts to cloud deployable models, which are explained in the next section.

3.5 Cloud Deployable Models

This section deals with different cloud deployable models. The elasticity of the cloud enables it to be dynamically deployed into different models. This capability allows applications from different locations to be deployed to different infrastructure during runtime. When choosing a deployable model, cloud consumers should be wary of the following concerns: security, cost, compliance and Quality of Service (QoS). Furthermore, the deployment model should be distinguishable by size, access mode and ownership. The architects of the cloud provide the following deployable models as previously shown in Figure 2.1.

- Public Cloud Model
- Private Cloud Model
- Community Cloud Model
- Hybrid Cloud Model

3.5.1 Public Cloud

Public cloud model resources are normally made available to the general public on a pay-per-use basis in a virtualised environment (Cloud Security Alliance, 2009). Resources that are generally involved in this model range from applications, storage and networks – all of which are provided over the internet. The public cloud is scalable because the resources are available on demand, and they are cost effective because the setup for hardware, bandwidth and applications is covered by the CSP. Because of multitenancy, the model is reliable, flexible and location independent. Examples of offered public cloud services include the Windows Azure Services Platform, Amazon Elastic Compute Cloud (EC2), Google AppEngine, IBM's Blue Cloud and Sun Cloud.

3.5.2 Private Cloud

Chahal et al. (2010) describe a private cloud as an environment that consists of shared multi-tenant and virtualised infrastructure. This model is based on a secure environment or enterprise that computes with specific clients and services provided within virtualised environments. An organisation that operates through a private cloud disregards third-party-

hosted services but offers the same cloud features. It also controls the consumer data, security and matters that are related to regulatory compliance. The model is scalable and supports multi-tenancy; resources are on-demand and support the processing of complex jobs. Additionally, a private cloud minimises security concerns through limiting the number of people who have access to data and giving organisations control of their data. Examples of providers that deploy private cloud infrastructure include VMware and Rackspace.

3.5.3 Community Cloud

CSA describes a community cloud as an infrastructure that is shared by several organisations. It may also support a specified community that has shared concerns governed by third-party service providers (CSA, 2009). On the same note Liu, Vlassov and Navarro (2014) describe a community cloud as a cloud that provides alternative choices by allowing clients to manipulate their different entities in the cloud without any restrictions from any public provider. The target might be a limited number of employees or some organisation (NIST SP 800-145), for example heads of companies, businesses, research and applications. The concerns supported by this model include regulatory compliance, audit requirements and performance requirements. Additionally, it is a centralised facility that supports multi-tenant infrastructure as well as different levels of security policies, and it can be hosted either within or outside the premises. A community cloud infrastructure may be shared by a number of organisations like governments, universities or central banks. For example Google applications for Government and Microsoft community cloud for government.

3.5.4 Hybrid Cloud

A hybrid cloud is a homogenous cloud service that uses both the private and public cloud to deliver its services within the same organisation. According to Annapureddy (2010), a hybrid cloud is able to combine resources and retain control of valuable data. In fact, the model enables client enterprises to store important data and then migrate un-important data to the cloud. Additionally, Taylor and Metzler (2010) declare that the hybrid cloud is composed of deployment options for different cloud services. Hence they are able to pave the way for organisations to use the computing resources of the public cloud so that users' needs can be met temporarily. Furthermore, a hybrid cloud can be implemented by a separate group of CSPs who may provide combined private and public services or a complete hybrid package through individual CSPs. In addition, different organisations that have private clouds may

integrate them into a public cloud. For example, Eucalyptus is software used to enable a private cloud to connect to public clouds.

Having looked at cloud deployable models, in the next section the reader is introduced to the roles played by the CSPs.

3.6 Role of Cloud Service Provider

This section discusses the roles that CSPs play in the cloud environment. CSPs are entities that are used to provide services to the users of the cloud in the form of management, service delivery and the dynamic provision of infrastructure and virtualised resources. On the same note, in PaaS, the CSP provides and manages the infrastructure and middleware for consumers, which includes development, deployment and testing of administrative tools. In IaaS, the CSP provides and manages processing and storage by hosting environment and the cloud infrastructure (NIST SP 500-291). Examples of cloud providers include Amazon, Apple, Cisco, Citrix, Google, Salesforce.com, Verizon and Rackspace.

CSPs need to be totally accountable and responsible for the services they provide. Although they provide different services to the cloud consumers, the CSPs have very clear responsibilities. These responsibilities and activities are based on the cloud service models and deployment models.

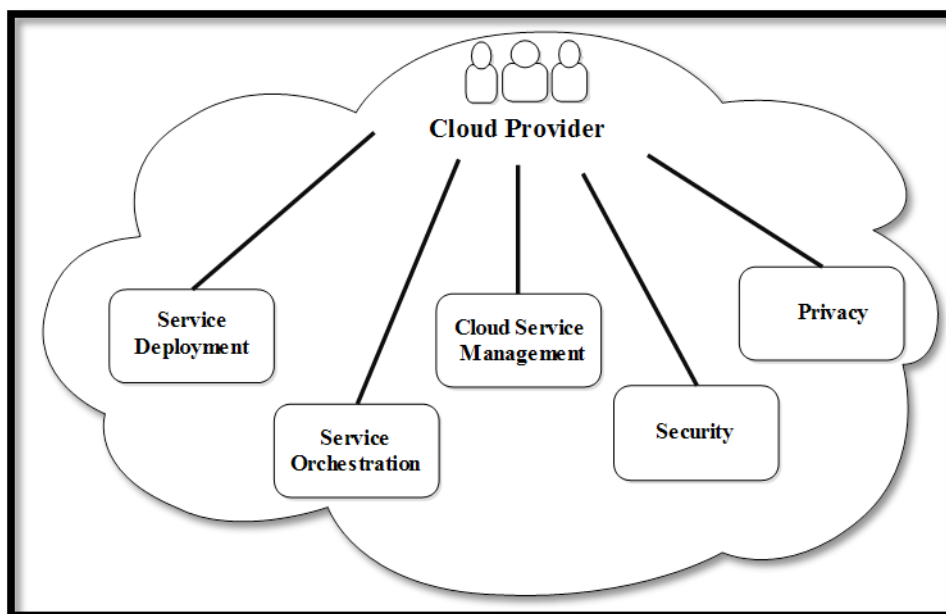


Figure 3.2 Roles of Cloud Service Provider (Adopted from NIST SP 500-291, 2013)

The most important aspect of the contract that exists between the CSPs and the consumer involves the Service Level Agreements (SLAs) that clearly define the policies and procedures for all the requirements needed during operation. Figure 3.1 shows how NIST SP 500-291 categorises the roles of a cloud provider. The roles include service deployment, service orchestration, cloud service management, security and privacy.

Security is an important feature of information protection and individual privacy. According to the Federal Cloud Computing Strategy (FCCS), a transparent secure environment should be created between the CSPs and cloud consumers to protect their privacy and the national security (Kundra, 2011). Finally, privacy as a role protects personal information stored within the cloud environment since the cloud users represent half of all the internet users (Seybent and Reinecke, 2014). There should be greater control over data that is stored in the cloud data centres. For example, in the USA, the Electronic Communication Privacy (ECPA) Act of 1986 governs how the privacy of electronic communications must be ensured.

Having looked at the role of the CSPs, a brief discussion on virtualisation and cloud computing appears in the next section.

3.7 Virtualisation and Cloud Computing

Virtualisation is certainly not a new concept in Information Technology (IT); it has existed in data centres ever since the 20th century as a means of consolidating servers. Ottenheimer and Wallace (2012) view virtualisation as “the creation of virtual resources from physical sources”. At the same time, virtualisation is a component of cloud computing through which multiple guest OS, servers, storage or network resources can be supported by a single host. The multiple guest OSs that are supported by the host are considered as VMs. Cloud computing, on the other hand, allows the creation of VMs through the presence of a hypervisor that acts as management software. The main role of the hypervisor is to manage the communication that exists between CPU, server memory, processing power and VMs. Decommissioning and provisioning of VMs can also be done using the hypervisor.

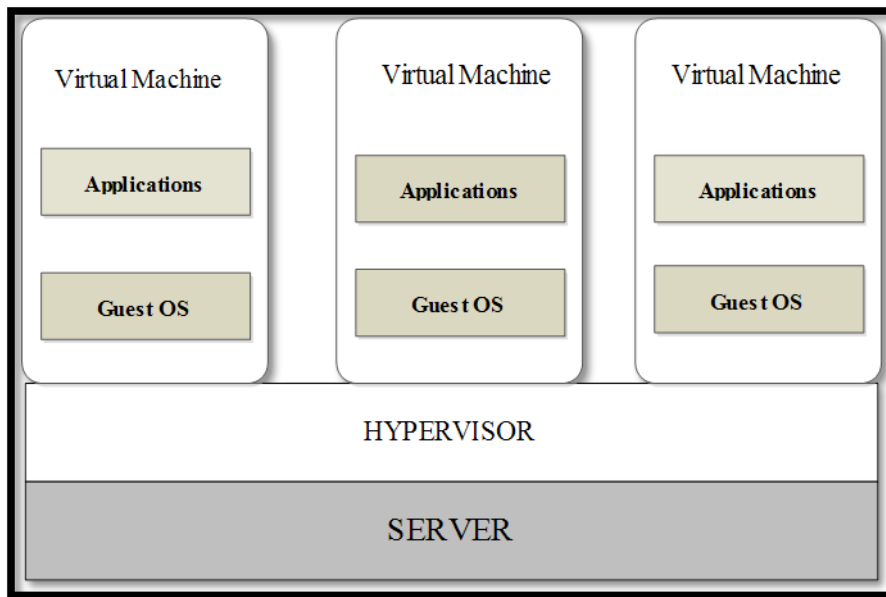


Figure 3.3 Example of Virtualisation

Figure 3.3 shows an example of virtualisation; the rectangle labelled server is used as a physical resource that creates a virtual computing environment through the hypervisor. The hypervisor allows the virtual machines to be provisioned and decommissioned. Each VM is able to support a guest OS and applications.

Having looked at virtualisation, in the next section, the reader is introduced to a discussion on how digital forensics can be adopted to cloud computing environment.

3.8 Adoption of Digital Forensics Readiness in the Cloud

This section deals with how digital forensic processes can be adopted in cloud computing environments. The cloud has not evolved to a degree where it can fully adopt traditional forensic processes to facilitate the investigation of crimes. This is due to technical challenges related to the nature of digital evidence, legal aspects and operational challenges (Birk, 2011). Regarding the technical aspects, there still exist no widely accepted standardised guidelines, frameworks and process models that can facilitate a digital investigation process in the cloud environment (Kent et al., 2006). Data from the cloud is often stored in different locations, which imply that data or potential evidence may be stored in different multi-jurisdictions that have different rules. In Section 2.9, the researcher highlighted the fact that the legal requirements might differ from one jurisdiction to the next, i.e. an action may be legal in one

jurisdiction and illegal in another. This might prove to be a challenge to digital forensic investigators.

Nevertheless, the growths of the cloud and the increase in data have caused the adoption of digital forensics in the cloud to continue at a slow pace and this is a big disadvantage to the forensic community. According to Marangos, Rizomiliotis and Mitrou (2012), this has happened because of issues like interoperability, conflicting legislation and a lack of standardisation. The adoption of digital forensics processes is essential for the proper establishment of capabilities that can help to fast-track investigations in the cloud. However, this inadaptability results from the fact that cloud computing is a new technique and neither researchers nor organisations have yet outlined guidelines and recommendations on how to conduct a DFI in the cloud.

The main aim of enforcing DFR in the cloud environment is to shorten the Digital Forensic Investigation (DFI) process through the collection of Potential Digital Evidence (PDE). Based on the concepts mentioned by Tan (2001), it is the researcher's opinion that the collection, preservation and presentation of digital evidence from digital sources for purposes of furthering a reconstruction of events are the most important relevant digital forensic readiness aspects.

Nevertheless, an environment that is forensically ready minimises the effort needed when conducting a DFI. Readiness can only be achieved through the collection, validation and preservation of critical information that is related to crimes as mentioned in the ISO/IEC 27043:2015 international standard, which was earlier discussed in Chapter 2. A forensically ready CSP should be able to respond to a security incident rapidly by taking the necessary steps to reduce the effort needed to perform a DFI while at the same time maintaining the credibility of collected potential evidence (Endicott-Popovsky, Frinke & Taylor, 2007). If a DFI is conducted in an environment that is not forensically ready, the process could cost an organisation a fortune and a lot of time.

Having looked at the adoption of digital forensics readiness processes, the next section the reader is introduced to the collection of digital evidence from the cloud.

3.9 Digital Forensic Evidence Collection from the Cloud

This section discusses the collection of digital evidence from cloud computing environments. The contrast between traditional forensics and cloud forensics is that one would need to create an image before examination in the former and not in the latter. Forensic logs should be provided by the CSPs to indicate whether digital evidence was compromised or not. This is because the cloud itself is elastic and different Service Level Agreements (SLAs) exist between the CSP and the consumer. The SLAs should explicitly highlight the ownership of data that the CSP is bound to retain during resource provisioning, as this data may end up being used as potential evidence if a security incident is detected.

Since the cloud is not jurisdictional, there is no guarantee that the data stored in the cloud will remain within a particular jurisdiction because there is bound to be geographical barriers. It is upon the SLAs to highlight that potential evidence should be subjected to the law of a given jurisdiction, based on its constitutional and statutory provisions. Also, the CSPs should comply with the regulations that allow data that exists as evidence to be provided in a given jurisdiction, because laws may differ from place to place. According to Dykstra (2013), searching and seizing electronic evidence from the cloud entails a stage of criminal prosecution that has to comply with the constitutional provisions. Delpont, Kohn and Olivier highlight the fact that to get evidence linked to an incident, a cloud instance can be isolated. Based on the above-mentioned propositions, conducting an investigation in the cloud will be dependent on the collection of evidence to the CSPs' data centres and availing such evidence to forensic experts when needed by the law enforcements agencies.

3.10 Conclusion

This chapter introduced the reader to the basic aspects of cloud computing, cloud computing service models, deployable models and virtualisation. The researcher drew attention to the important aspects that make the cloud a preferable mode of computing. Because it is necessary to understand the architecture of the cloud before conducting forensic processes and methodologies in it, a visual model of the NIST cloud computing architecture with its characteristics, service models and deployable models was presented in Section 3.3.

Since the proposals in this research study can easily be adopted by any CSP, the roles of the CSPs have been also been discussed in Section 3.4. The chapter closed with a discussion on virtualisation, DFR and its adoption in the cloud.

Having looked at cloud computing, it is essential to know how a botnet operates, its administration and control, the protocols it uses and its ability to collect digital information. As a result, the reader is introduced to botnets in the next section.

Chapter 4: Botnets

4.1 Introduction

The previous chapter introduced the reader to the literature on a cloud model, the characteristics of the cloud, virtualisation as a component of cloud computing and how digital forensic readiness (DFR) can be incorporated in the cloud. Based on the information gathered so far, this chapter now introduces the reader to botnets, which in this research thesis are presented as forensic agents that are able to collect PDE from the cloud environment in a non-malicious way and in a DFR approach.

In recent years, botnets have been considered as the biggest threat that exists in network security. This is mainly because of their topologies that give them the capability to accomplish various tasks effectively and to survive for long periods within networks. In most cases, this strength is attributed to the ability of the botnet operator to send commands that are able to fetch information on machines that are compromised. According to the FBI, the act of using botnets for adversarial attacks is on the rise, and has caused a number of organisations and financial institutions to lose millions of dollars (Rodriguez-Gómez, 2011). In an attempt to maintain the objectives of the botnets, their developers have researched new ways of hardening the botnet infrastructures and hence, the resilience of compromised machines is not guaranteed.

In the cloud environment, botnets have become very versatile and they are able to propagate and collect traffic in distributed mode – after which they can then process the collected data using distributed rules in cloud technology (Han, Chen, Xu & Liang, 2012). Furthermore, a large group of machines are owned by botnet operators, which allows them to install malicious software to launch cloud-based botnets that are able to collect information illegally.

The emergence of botnets can be traced to early 1988 when the Internet Relay Chat (IRC) was invented (Kalt, 2000). IRC was an application layer protocol that was able to facilitate communication between parties through the transfer of messages. However, since then botnets have evolved to be very complicated and effective in camouflaging themselves within the network. In 1998, NetBUs and Back Orifice emerged as malware tools that could support a client and server version. Back Orifice allowed a client application that was able to run on

one machine to monitor and control a machine that was running the server applications (Symantec, 2007). Other features included locking the machine, restarting the machine, transferring files and displaying the passwords.

In this research thesis, however, botnets are used as positive forensic agents that are able to collect digital forensic information from the cloud environment in a non-malicious way to be used for DFR purposes. The researcher preferred to use botnets because when they are modified to act in a non-malicious way, they offer the following software agent attributes: autonomy, interactivity, mobility, intelligence, adaptability and interactivity (Biermann, 2009).

The remainder of the chapter is structured as follows: A definition of a botnet is given in Section 4.2, followed by the life-cycle of the botnet in Section 4.3. The anatomy of a botnet is covered in Section 4.4, and Section 4.5 presents a discussion on botnet control and administration. Section 4.6 discusses the use of botnets, Section 4.7 discusses how a botnet is considered as an a cloud attack vector and the chapter concludes with Section 4.8.

4.2 Definition of a Botnet

Leder, Werner and Martini (2009) presents a botnet or robot network as a generic term that describes a set of scripts written to perform predefined functions. Furthermore, the author highlights that a botnet is depicted as an alliance of computers that are interconnected and infected with malicious software. The bot itself is derived from “ro-bot”; from this perspective, bot represents the set of commands that is able to operate as an agent who is able to extract human activity. It is a group or network of computers that have been infected with a type of software that allows a so-called botmaster to control the infected systems. A botmaster is a human who is able to control the bots through the Command and Control (C&C) server.

Botnets work in the following manner: Bots are usually known to spread themselves across the internet by searching for unprotected and vulnerable computers. They are subsequently able to infect these computers via electronic mail, through malicious attachments or visited pages in a website. After infection, the bot is able to report to the C&C server channel. In this context, the role of the C&C server is to allow the botmaster to communicate and give

commands and direct the actions of the botnet.

The botmaster is a malicious actor who operates the bot clients from a remote location where he/she is able to command a chain of zombie computers. A zombie computer in this context is a computer that has been compromised by a bot. Botnets have always been attributed to crime-ware syndicates and they are also considered as the dark side of computing (Kebande & Venter, 2014). They are able to perform illegal activities ranging from information theft and spamming, to Distributed Denial of Service (DDoS) (Banday et al., 2009). They perform these activities through searching for a vulnerable computer for initial infection, after which the bot is distributed to clients (targets). Finally, they can connect to the botmaster for more instructions.

4.3 Life-Cycle of a Botnet

A botnet is made up of three key elements: Command and Control (C&C) servers, bots, and the botmaster. The botmaster sends commands and instructions to bots residing in the targets and controls the botnet through the C&C server. The main role of a bot is to infect the target computers without the owners' knowledge. Figure 4.1 contains a block diagram depicting the life-cycle of a botnet.

In Figure 4.1, the botmaster infects fresh targets by propagating bots in the box labelled 1 through the C&C server. This may be done through sending a malicious email attachment or spam, or by sharing malicious files. Once infected, the bot is able to connect back to the C&C server in the box labelled 2 and the infected computer becomes a zombie under the control of a botmaster. New updates and instructions are given to the bot through the C&C. Thereafter, in the part that is labelled 3, bots are able to listen to the C&C server periodically to receive more instructions from the botmaster. When the bot identifies a new command, it executes it and the report is sent back to the C&C server while awaiting new instructions (Schiller & Binkley, 2011).

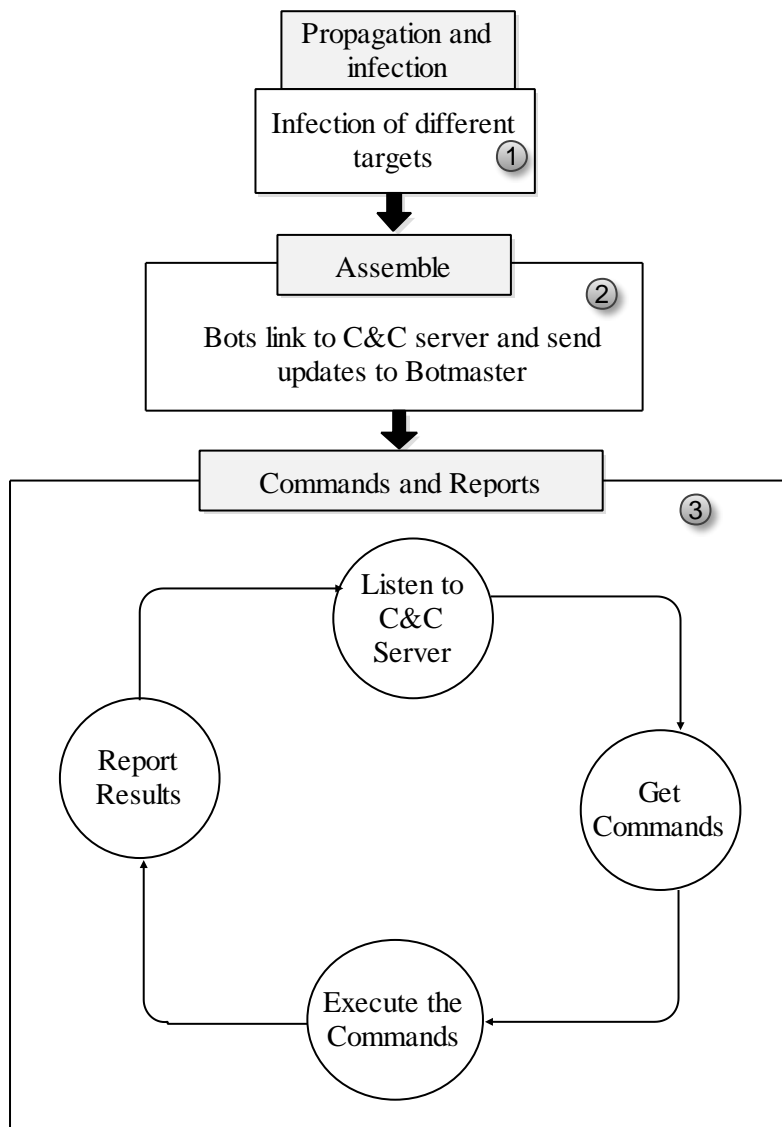


Figure 4.1 Life Cycle of a Botnet

Having looked at the lifecycle of botnets, the next section provides a discussion on the anatomy of botnets.

4.4 Anatomy of Botnets

According to the European Network and Information Security Agency (ENISA) the infrastructure of a botnet involves having a control entity that is either centralised or distributed (ENISA, 2011). Apart from that, the most crucial part of a botnet is the command and control infrastructure (C&C). ENISA also highlights the fact that the C&C server serves as the only way to control bots that are active in the botnet. The anatomy of the botnet shows how a botnet is structured, its behaviour, trends and how it distributes itself across different computers. In this context, the anatomy of the botnet can be dissected to display two distinct

architectures: a centralised and a decentralised C&C architecture. Figure 4.2 shows the architecture of a centralised botnet, after which each component will be explained below.

4.4.1 Centralised C&C Architecture

A centralised architecture that is used to control the bot clients is displayed by the C&C server. Nevertheless, according to Gu, Zhang and Lee (2008), in a centralised C&C architecture, bots are able to establish communication using a C&C server that is controlled by the botmaster. Furthermore, communication occurs simultaneously with the bots because of the ability of bots to connect to the C&C server. The architecture of a C&C is a central server that maintains requests, files and information being routed, and that coordinates all the instructions being sent from one point to the other. Figure 4.2 shows the architecture of a centralised botnet which consists of the botmaster, the internet, C&C server and bot clients. It also illustrates the four stages in a botnet's life cycle. An explanation of how the architecture works follows next.

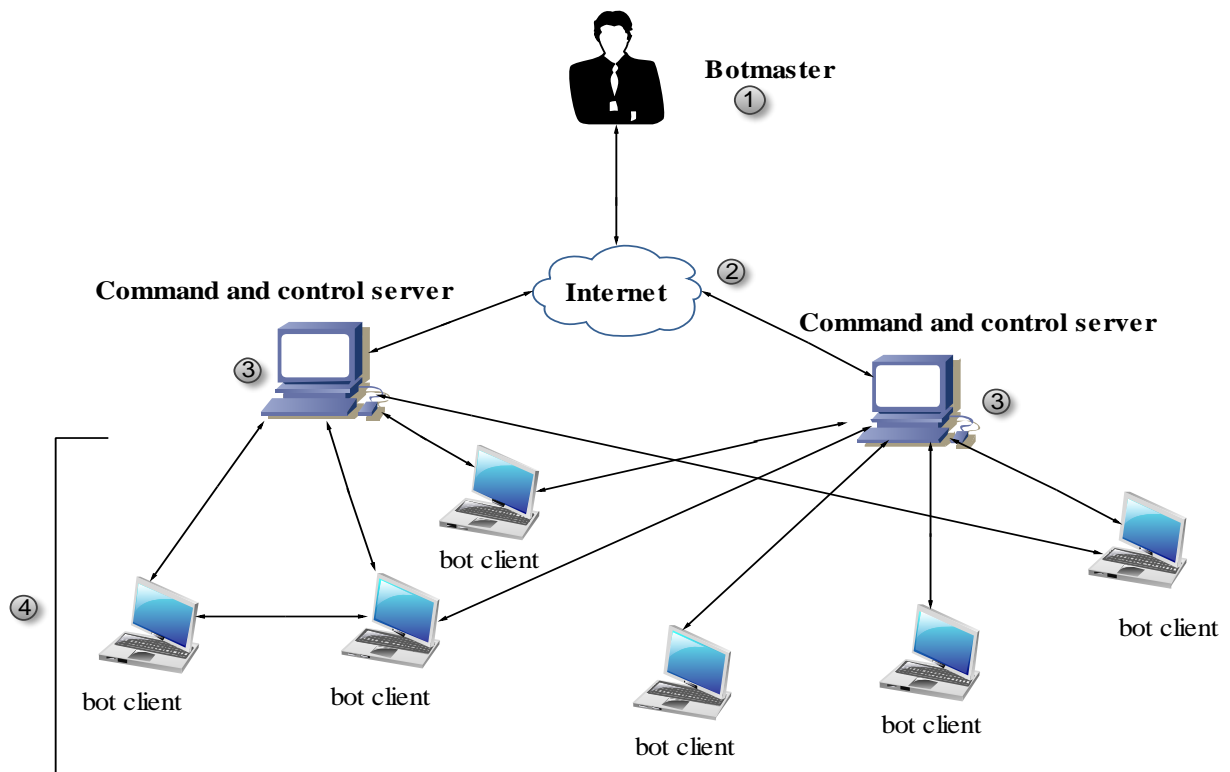


Figure 4.2 Architecture of a Centralised C&C Architecture

In Step 1, the botmaster is able to command new bot client computers remotely through the C&C server to spread bot binaries over the internet (see Step 2 of Figure 4.2). A bot in this context constitutes a malicious code that automatically propagates itself through the bot client computers. Through the C&C server involved in Step 3, an infection is executed to a new bot

in Step 4, when bot malware is downloaded by bot clients. Once that is done, the bot clients become zombie computers under the control of a botmaster. Next, the main binaries of the bot get fetched via Peer-to-Peer (P2P), Hypertext Transfer Protocol (HTTP) or File Transfer Protocol (FTP) by the shell code which is then able to install itself to the target client. After this, the bot clients are able to communicate with the botmaster through the C&C server which acts as a connection point.

According to Junewon (2011), a C&C server channel will be established by a new bot during propagation, which enables communication with the botnet. Attack commands are passed via the C&C channel where the bot will receive and execute them. According to Feily et al. (2009), connection between the bot and botmaster will be maintained when there is a need for bot binary updating so that the botnet can receive new instructions and operate in stealth mode. Any computer on which this programme can be installed becomes a zombie and is capable of executing the malicious code.

The decentralised C&C architecture is dealt with in the next section.

4.4.2 Decentralised C&C Architecture

According to Wang and Zou (2010), a decentralised architecture has no central C&C server and all the requests are handled by peers within the network. Furthermore, in decentralised C&C server architectures the links between the bots enable communication (ENISA, 2011). In this architecture, it is impossible to extract information that concerns the botnet distribution, because commands are sent directly to the botnet or to the peer device.

Figure 4.3 shows a decentralised C&C architecture that consists of the botmaster (Step 1), internet (Step 2), initial command injection (Step 3), and secondary infection (Step 4). The botmaster is able to inject commands into a single bot client over the internet in Step 3; he/she then uses the infected bot client as a C&C server to infect another client in a secondary infection in Step 4.

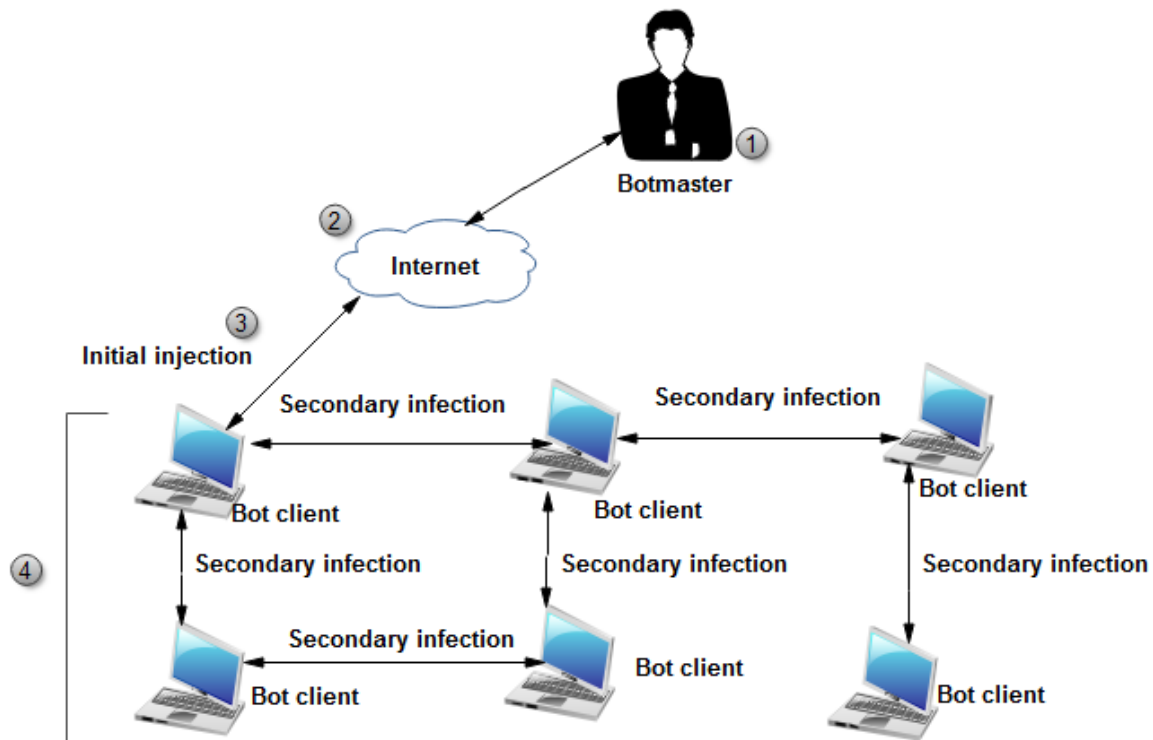


Figure 4.3 Architecture of a Decentralised C&C Architecture

The bot clients are distributed all over without a central server, which makes them slower, compared to centralised architectures. On the other hand, command transfer is done from one bot client to another and the botmaster is able to control a large number of computers by simply performing a login into a single compromised computer.

Having looked at botnet architectures, the reader is in the next section introduced to how botnets are controlled and administered.

4.5 Botnet Control and Administration

This section discusses the protocols that are used in administering and controlling botnets. It mainly introduces ways through which the compromised computers can be administered. The discussion is based on three protocols: Internet Relay Chat (IRC), HTTP and P2P.

4.5.1 Internet Relay Chat (IRC)

IRC came to life in 1989 and it was mainly used to control sessions in chat rooms. According to Jeff Fisher, the first IRC bot called Eggdrop appeared in 1993 and has since developed into

very powerful bots (Eggdrop, 1993). A botmaster trains botnets to communicate through IRC, to attack through a Distributed Denial of Service (DDoS) and spam, and to propagate through vulnerabilities.

Eggdrop, which is non-malicious in nature, was developed with the Expandable Tool Control Language (TCL) scripts that utilised C modules. It was also designed to run on Windows, SunOS, MacOS, Linux and BSD, and it supported five IRC networks, namely DALnet, EFnet, IRCnet, QuakeNet and Undernet (Eggdrop, 1993). Furthermore, Eggdrop is a supported IRC bot that has various options for channel management. A study on IRC networks by Oikarinen and Reed (1993) showed that while an IRC is set up, the client program is able to connect to an IRC server within an IRC network. The Transmission Control Protocol (TCP) server port for IRC is identified as port 6667. The added advantage of using IRC is that orders are received and answered without delay between the zombies and the botmaster. The disadvantage, however, lies in successfully setting up the IRC to control the botnet.

4.5.2 HTTP

According to Chiang and Lyod (2007), botnets can use Hypertext Transfer Protocol (HTTP) to communicate with the control servers. At the same time, botnets can trend in traffic when they mingle inside HTTP traffic; this means that HTTP botnets use HTML to communicate. Moreover, HTTP is used to host the C&C server where the bots connect through a web server to receive commands. Controlling a botnet via this method is very effective because the target computers can be connected through a web interface, which enables sending and receiving of bot commands. Furthermore, HTTP botnets are hosted on websites that have legally registered domains or websites that have been hacked. Grizzard et al. (2007) consider as a drawback of using HTTP over other forms of administration the fact that it depends entirely on one server, the C&C server, which means that if the server is removed by the service provider, control of the botnet will be lost. Additionally, if the C&C server address is changed from random domain names, control of the botnet will be lost too and it can be brought down. For example, HTTP communications occurs over Transmission Control Protocol/Internet Protocol (TCP/IP) and the default port is TCP 80. Additionally HTTP uses Multipurpose Internet Mail Extensions (MIME) constructs. MIME is able to format email and text character set and non-text attachments like audio, images, video and application

programs. The MIME constructs used by HTTP have also been defined in RFC 1521. Furthermore, HTTP uses Uniform Resource Identifiers (URIs) which are used to identify network data objects.

4.5.3 P2P

Considering the HTTP drawback discussed above, Peer-to-Peer (P2P) botnets were established to be widespread and widely used by their creators. In this case, the botmaster gets access to a single bot and then uses his/her own system to connect. Thereafter, the system is able to send and receive commands to and from the target computers, without the need of centralised distribution. According to Grizzard et al.,(2007), botnets in P2P appear to be very distinctive because their resilience is based on a P2P network. In fact, P2P protocols actually emerged in 1999 when the Napster bot was released. Napster allowed its peers to find and share files over the network with other peers (Grizzard et al., 2007). Furthermore, P2P bot administration has undergone massive developments including the use of customised protocols to administer bots. An advantage of using a decentralised P2P is that neutralising the botnet is not an easy task, due to the absence of a C&C server. Common examples of P2P networks are Skype, Bit Torrent, LimeWire and Gnutella.

In the next section, the usage of botnets is discussed.

4.6 Usage of Botnets

In this section, the reader is introduced to the different ways in which botnets are used, namely for purposes of cyber-extortion, traffic sniffing, key-logging, spam and DDoS.

4.6.1 Cyber-Extortion

Cyber-extortion attacks are lucrative practices that adversaries and cyber-extortionists use to threaten or incapacitate an individual's system, website or major components of an information system (Sulkoswki, 2007). It is a crime that comes in many forms and botnets are often used to threaten organisations and/or government institutions by demanding money to avert any potential attack. Botnets are set to hijack numerous computers and overwhelm them by employing sophisticated techniques for purposes of financial gain. Through the C&C server, the zombie computers are used to launch DDoS attacks on organisational servers by sending numerous requests. The extortionist would then demand money from organisations to

stop the attacks and a majority of organisations will end up making this payment because they find it cheaper than the downtime of their servers.

4.6.2 Traffic Sniffing

Clear text that passes through a system that has been compromised can be watched by bots. In the case of traffic sniffing, a botnet is designed to act as a packet sniffer to locate critical and sensitive information in a system, for example personal information or login details like usernames, passwords, etc. According to Liu, Xiao, Ghaboosi, Deng and Zang (2009), botnets are able to sniff command data in a victim's computer – they scan the host for significant data and listen to keyboard activities, which enables them to retrieve sensitive information. Once they have filtered meaningful inputs, they are able to report to the botmaster. Through traffic sniffing, a botnet may also be able to steal details of another botnet and gather essential information if that particular system has been compromised too.

4.6.3 Key-Logging

Bots use key-logging techniques to capture critical and sensitive information from a victim's system. Key-logging is done to bypass secure and encrypted communication channels like POP3s and the HTTPS. A key-logger can be installed remotely and can be distributed over different computers without the attacker requiring physical access to the victim's computer. Keystrokes are used to detect the user's input. Baig and Mahmood (2007) state that when a key in the keyboard is pressed, communication begins through a hardware interrupt, which is channelled to the system level message and a specific key value. The bot is also able to compromise thousands of systems running in parallel while collecting sensitive information.

4.6.4 Spam

A botmaster uses a sneaky way to misconfigure the victim's computer so that he/she might get access to private information. This is done through emails, advertisements or through posting malicious codes in websites in the form of free downloads (mostly games). The adversary is able to compromise many systems in parallel. According to Liu et al. (2009), a study reported that when SOCKS v4/v5 proxy (TCP/IP RFC 1928) is opened on the compromised hosts, a number of those machines will be used for nefarious jobs like spamming. Once infected, a number of bots are configured to enable the TCP/IP SOCKS proxy when a system is compromised. Usually, SOCKS is enabled to allow numerous

spamming of other computers. Additionally, a study by Zamil et al., (2010) has revealed that the recent spread of botnets is attributed to the exchange of large volumes of unsolicited mails.

4.6.5 Distributed Denial of Service

Distributed Denial of Service (DDoS) is an online attack made by botnets that make services unavailable by overwhelming the victim's computer with traffic. According to Specht (2004), DDoS is a coordinated attack that has the purpose of preventing legitimate users from using the resources provided by the network through many compromised computers. This attack normally targets thousands of systems, which makes them lose network connectivity through bandwidth consumption. Normally a botnet initiates this attack by sending TCP, SYN and UDP flood attacks to a target victim's computer. The first experience of known DDoS attacks was on Yahoo.com in 2002, which resulted in keeping Yahoo off the internet for close to two hours and the aftermath was a loss in advertising revenue (Kessler, 2000).

4.7 Botnet as Cloud Attack Vector

This section presents a discussion on how botnets are regarded to be attack vectors in the cloud environment. Botnets have emerged to be very dangerous predefined attack vectors that are able to easily take down the infrastructure of the CSPs. According to Lin and Lee (2012), a botnet can easily set up a C&C server in order to steal information from a victim's machine. The authors go ahead to illustrate that the cloud environment provides an ideal environment due to the availability of rich elastic computing resources like storage, bandwidth and processors that easily supports deployments.

Furthermore, the cloud provides an environment that one is not able to trace. Research by Jiang, Im and Koo (2012) has also shown that the SaaS is capable of elaborately being exploited in an unprecedented way by SaaS driven botnets as an attack vector. Consequently, according to Miao, Potharaju, Yu and Jain (2015) the cloud has become an attractive target for attackers to disrupt services, compromise resources and to launch network-based attacks. Based on the aforementioned discussions, it is evident that botnets are able to support cloud forensic research by being able to collect digital information that can be used for digital forensic purposes.

4.8 Conclusion

This chapter introduced the reader to the basic aspects of botnets, botnet control and administration, and the usage of botnets. The literature explained the structure and the capabilities of botnets over the internet. The chapter also showed the anatomy of botnets and gave a clear indication of the architecture of botnets.

Now that the reader has been introduced in chapters 2, 3 and 4 to the literature on which this research study was based, the next chapter discusses the model requirements that have been employed in this research study.

“Normal science does and must continually strive to bring theory and fact into closer agreement and the successive transition from one paradigm to another via revolution is the usual developmental pattern of mature science.”

-Thomas S. Kuhn-

Part Three: Model

Part Three of this thesis specifically discusses the contribution of the research. It deals with the model proposed in this research and is further divided into three chapters – Chapters 5, 6 and 7. Following the reviews presented in **Chapter 2** on the need for DFR, **Chapter 5** discusses the model’s requirements towards achieving DFR in the cloud. It explains the materials and methods used, as well as specific experiments that were conducted in the study. **Chapter 6** presents hypothetical case scenarios that were used to highlight the problem addressed in this thesis. These scenarios deal with fictitious scenes that provide a background on how DFR can help mitigate the effort and reduce the cost and time that will be needed by an organisation to conduct a DFI. The examples used while building the hypothetical case scenarios were used to introduce the prototype. The latter is discussed in **Chapter 8** and aims to show how DFR can be realised in the cloud environment. Finally, **Chapter 7** addresses the proposed Cloud Forensic Readiness as a Service (CFRaaS) model proposed by the researcher.

Chapter 5: Requirements of a Cloud Forensic Readiness Service Model

5.1 Introduction

The previous chapters presented the literature on Digital Forensics (DF), Digital Forensic Readiness (DFR), cloud computing and botnets. This has enabled the reader to more easily understand the focus of the research reported in this thesis. The problem statement addressed the lack of an easy way of conducting DFR in the cloud without tampering with the functionality or infrastructure of the existing cloud architectures, or the implementation thereof. However, the purpose of this chapter is to present the model requirements that are needed for the cloud to be forensically ready for digital forensic investigations. These requirements serve as a foundation for establishing DFR capabilities within any organisation. Furthermore, the requirements identified in this thesis comply with the guidelines that have been highlighted in ISO/IEC 27043: 2015, the international standard with forensic readiness investigation capabilities, though not focused on the cloud.

For purposes of internal and external investigation in any organisation that has enforced the cloud-based infrastructures, any considerations made should be able to proactively allow the identification of artefacts that will ensure that DFR is achieved. Hence, it is essential for a DFR model to prescribe forensic requirements to highlight those requirements that will support the deployment of a forensic agent between the cloud provider and its clients. These requirements are aimed at assisting and facilitating the performance of forensic activities since PDE is normally scattered between the providers and the clients.

In this research thesis, the researcher therefore proposes a number of requirements specifically for the CFR model that will be presented in Chapter 7 of this thesis. The requirements proposed here are aimed at facilitating the process of collecting PDE that can be used to support litigation or a hypothesis in a court of law. The researcher believes that the requirements proposed in this chapter will play a significant role in enhancing the DFR process in the cloud environment without having to modify the existing cloud architecture.

The researcher also wishes to present the unique requirements before introducing the actual model.

This chapter focuses on presenting the CFR model requirements in the best way possible for them to meet their main functional purpose, namely ease of use, proper interoperability of functions, proper integration, and communication with stakeholders. Each primary functioning process of the CFR model has been identified and each required input and output of the specific CFR model is also represented.

The rest of the sections in this chapter are structured as follows: The need for model requirements that will achieve DFR in the cloud environment is described in Section 5.2. Next, the requirements are described in Section 5.3, and a summary of the model requirements for achieving DFR in the cloud environment appears in Table 5.1. The chapter is concluded in Section 5.4.

5.2 Need for Model Requirements

Several requirements have been proposed to help in analysing and specifying the CFR model that is proposed in this research thesis. These requirements must allow a software application with the modified functionality of a botnet that operates like a forensic agent to be deployed in a highly scalable environment and perform forensic processes.

From a cyber-criminal's perspective, using the cloud to conduct digital crimes is more advantageous because a cyber-criminal can easily go unnoticed – as experienced in all the hypothetical case scenarios discussed in Chapter 6 of this research thesis. This is because of the challenging cyber-investigation processes brought about by the inadaptability of DF techniques in the cloud environment. Cross-cutting jurisdictions, multi-jurisdiction and lack of a common international law for cross-border cyber-investigation are a challenge too, as is locating data provenance, owing to the fact that a server may be located in a completely different territory or jurisdiction. More so, the distribution of datacentres may also act in favour of the cyber-criminal. In order for the LEAs and the DFIs to launch a DF investigation process in the cloud, CSPs should be empowered to employ the CFR model. The latter will make the cloud forensically ready by collecting useful information that can be used for DFI

purposes. The requirements discussed in this research thesis are aimed at creating a relationship between the CSPs, DFIs and LEAs.

The main goal of the proposed requirements is to ensure that the CFR model works according to the specification, which is to “infect” instances in the cloud environment and collect the necessary forensic information without having to modify existing cloud architectures or implementations. Additionally, the requirements ensure that functional relationships are established between the entities of the model. The essence of having functional relationships is to have a common understanding of the role of each requirement. The rationale for choosing each requirement is explained in each sub-section where the requirements are explained in detail. The requirements are considered important because of how they influence the design of a CFR model.

Having explored the reasons for model requirements, the requirements that are needed in order to achieve DFR in the cloud are highlighted in the next section.

5.3 Model Requirements for Achieving DFR in the Cloud

The presentation of the requirements for achieving DFR in the cloud environment introduces a way of supporting the structural properties of the system. It also provides a method of building a system that comprises of concepts, tools, frameworks and heuristics (Golden, 2003). The requirements are presented based on two distinct approaches: general and architectural requirements.

5.3.1 General Requirements

Any DF tool must satisfy specific requirements and guidelines in order for it to be considered as reliable during a scientific process of extracting evidence. When these requirements are adhered to, the forensic tools used are accepted by the forensic community. The forensic tools subsequently gain commercial support, which ends up making the collected evidence reliable, and increasing its chances of admissibility in court. For example, the Scientific Working Group on Digital Evidence (SWGDE) of USA requires the integrity of digital evidence that is to be presented for examination to be maintained through hashing (SWGDE, 2006). Also, the National Institute of Standards and Technology (NIST), through a project supported by Computer Forensic Tool Testing (CFTT), highlights the need for accuracy and measurability

as part of the requirements for tools that will ensure that integrity and useful information is achieved (CFFT, 2007). Consequently, essential characteristics of cloud computing have to be achieved for a DF tool to be well-suited or to successfully perform its functions in the cloud environment. For example, according to Dykstra (2012), a DF tool should be able to satisfy the following essential cloud computing characteristics: on-demand, rapid elasticity, measured service, resource pooling and scalability.

The researcher examined the factors listed below as general requirements that can satisfy a CFR model as proposed later in this thesis. The researcher also reviewed literature on the requirements for DFR (Rowlingson, 2004; Mouton & Venter, 2012; Yaninsac & Manzano, 2001; Kebande & Venter, 2016; Tan, 2001) and ISO/IEC 27043 and 27017 international standards. Rowlingson (2004) claims that gathering and using digital evidence is a business requirement for DFR in any organisation. On the same note, Tan (2001) believes that different technical aspects that relate to logging, timestamping and digital preservation are requirements for DFR. Also, Hou, Sasaki, Uehara and Yiu (2013) deem integrity and authenticity to be two fundamentally important aspects of digital preservation that must be considered in order for digital evidence to be admitted in a court of law. Other examples of significant literature that has helped to determine the factors for DFR include Carrier and Spafford (2004) on reconstructing events at digital crime scenes, Dong and Yong-Qing (2012) on how digital evidence events are analysed, and Khanna et al. (2006) on forensic characterisation. Security implementation is highlighted by Kuntze and Rudolph (2011) as well as by Richter, Kuntze and Rudolph (2010), while forensic reporting is emphasised in ISO/IEC 27043: 2015. Consequently, Schwerha (2004) highlights the constitutional and statutory provisions with regard to collection digital evidence. Other requirements whose literatures have not been explored – like non-modification of the functionality of the existing infrastructure or architecture and obfuscation – are propositions that have been made in this research thesis. The researcher identified the following general requirements of the CFR model in this research thesis:

- Organisation requirement
- Digital forensic governance
- Forensic logging capability and management
- Digital preservation

- Timestamping
- Digital evidence characterisation
- Non-modification of the functionality existing cloud architecture
- Security implementation
- Obfuscation
- Event reconstruction
- Constitution, legal and statutory provisions
- Forensic readiness reporting

These requirements are each discussed in more detail in the sections to follow.

5.3.1.1 Organisation requirement

As an organisation requirement, it is important for any organisation to collect potential evidence in order to be able to manage business risks. This is only possible through the collection of appropriate evidence that can be used prior to the occurrence of an incident (Rowlingson, 2004). Additionally, Yasinsac and Manzano (2002) and Wolfe-Wilson and Wolfe (2003) believe that computer forensics can be enhanced by the availability of enterprise policies and that organisations can control DFI by having proper procedures in place for preserving digital evidence.

5.3.1.2 Digital forensic governance

The CFR model complies with readiness processes that have been highlighted in the ISO/IEC 27043 guidelines for information technology, security techniques, incident investigation principles and processes. In addition, it complies with the code of practice for information security controls based on ISO/IEC 27002 for cloud services. While ISO/IEC 27043: 2015 is not focused on the cloud, ISO/IEC 27017: 2015 gives guidance on the information security aspects of cloud computing. Some of the guidelines that have been highlighted in the standards include *Planning and Preparation*, *Scenario Definition*, *Logging and Monitoring*, *Digital Preservation*, *Incident Detection*, *Pre-Incident Analysis* and *Concurrent Processes*. The above-mentioned standards make provision for a wider scope of requirements that enforce a comprehensive approach towards compliance by the CSPs. This enforcement of compliance allows the CSPs to have a proper understanding and interpretation of the requirements before enacting them to the clients.

5.3.1.3 Forensic Logging Capability and Management

Forensic logging is the processes of capturing or recording data that is being generated by a device. This may include data that is passing through a particular point in a networked computer system. Marty (2011) describes logs as the pieces of data in a cloud-based infrastructure that appear to have great importance. In the context of this research thesis, logging is defined as

“a mechanism of collecting and capturing data, analysing and preserving it for digital forensic readiness purposes”.

A number of techniques that prove that forensic logging can be done in the cloud have already been published (Marty, 2011; Zawoad & Hasan, 2013; Dykstra & Sherman, 2012). In addition, in the researcher has previously proposed a technique for collecting and digitally preserving forensic logs for purposes of a DFI (Kebande and Venter, 2014).

Log management should help to facilitate the process of conducting forensic logging in the cloud environment. More so, log management ensures that proper log records are retained, logs are centralised in a forensic database, support is given to different log formats, log integrity is maintained and finally that an audit trail for logs is kept. Since logging has been employed to collect PDE, a software application acting as a forensic agent is able to harvest log information from the cloud environments. The Forensic Logs (Fl) that are generated can be defined as follows:

$$Fl = \{Fl_{(IP)}, Fl_{(TP)}, Fl_{(KST)}, Fl_{(Username)}, Fl_{(UID)}, Fl_{(C_Usage)}, Fl_{(R_Usage)}\} \quad (5.1)$$

where $Fl_{(IP)}$ denotes the forensic log tagged with an Internet Protocol (IP) address, $Fl_{(TP)}$ denotes the forensic log with a timestamp, $Fl_{(KST)}$ denotes the forensic log that denotes a keystroke, $Fl_{(Username)}$ denotes the forensic log with a username, $Fl_{(UID)}$ denotes the forensic log with a User-ID, while $Fl_{(C_Usage)}$ and $Fl_{(R_Usage)}$ denote the forensic log with CPU usage and RAM usage respectively. The collected forensic logs should be retained in their original form since they must satisfy the test of admissibility in a court of law. The integrity of these

forensic logs should be maintained by the process of digital preservation, which is explained in the next subsection.

5.3.1.4 Digital Preservation

During a DFI process, preservation of digital evidence plays a crucial role in ensuring that digital evidence is admissible in a court of law. The objective of including Digital Preservation (DP) in the CFR model is to make sure that no changes are made to the forensically logged potential evidence (Endicott-Popovsky, Frincke & Taylor, 2007). The forensic database stores the blocks of forensically logged (B_fl) data that is collected from the cloud. The collected data is digitally preserved for purposes of forensic examination and analysis. Forensic examination and analysis may only commence if a security incident that needs a DFI has been detected. Digital preservation is presented using the equation given below:

$$DP = \{B_fl_1, B_fl_2, \dots, B_fl_n\} \quad (5.2)$$

where B_fl represents a block of forensically collected logs. Firstly, Hash values ($HshV$) for B_fl are created after the evidence that is captured, after which the B_fl are transferred to a storage area. The $HshV$ for each B_fl is represented as follows:

$$HshV = \{Hsh(B_fl)_1, Hsh(B_fl)_2, \dots, Hsh(B_fl)_n\} \quad (5.3)$$

where $HshV$ represents the hash value that is created and Hsh represents the hash for each B_fl . $HshV$ comes as a result of a hash function or a message digest. A hash function is able to utilise some mathematical functions and calculations that are able to generate numerical values based on the originality of the digital evidence that is collected. Therefore, the overall DP equation can be represented as follows:

$$DP = \{HshV\} = \{Hsh(B_fl)_1, Hsh(B_fl)_2, \dots, Hsh(B_fl)_n\} \quad (5.4)$$

According to Endicott-Popovsky et al. (2007), hashing is presented as the use of a mathematical function that aids in the creation of unique fixed strings that come from the length of a message. The main purpose of hashing is to help maintain evidence integrity by checking if the forensically logged data has been altered or not during a DFI process. Menezes, Oorschot and Vanstone (1996) define integrity as a condition in which digital data has not been altered in a manner that is unauthorised from the time it was created, transmitted, or stored by an authorised source. This becomes important during the digital forensic investigation process, because integrity implies that the digital evidence that is produced in a court of law has remained unaltered throughout the DFI process.

5.3.1.5 Timestamping

A timestamp is used to show the exact time (precise year, month, day, minutes and seconds) at which an event was recorded by the computer. It is important in the digital investigation process because DF investigators may need to know the exact moment that a digital event occurred. Although log files can be used to trace the intruder's actions, Koen and Olivier (2008) argue that a timestamp can act as an alternative, where its information can be used as a simple way of extracting the log of events that transpired. This is because the last actions or activities that are performed on a file may act as a valuable source of evidence when there is no any other alternative. The researcher defines a timestamp using the following equation:

$$Fl_{(TP)} = \{B_fl_j, B_fl_{j+1}, \dots, B_fl_{n+1}\}, \{Hsh(Hshk_0), Hshk_2 = Hsh(Hshk_1) \dots Hshk_i = Hsh(Hshk_{i-1})\} \quad (5.5)$$

where $Fl_{(TP)}$ denotes the timestamp for the forensic log and TP denotes the timestamp that depicts year, month and day. Next B_fl_j denotes the collected block of forensic data from the cloud environment. $\{Hsh(Hshk_0), Hshk_2 = Hsh(Hshk_1) \dots Hshk_i = Hsh(Hshk_{i-1})\}$ represents $Hshk_i$, which is used as a hash function or a signature while verifying the integrity of collected PDE as was mentioned in the previous section.

Activities that trigger digital events in the cloud environment may occur at different times and at different locations as denoted by TP , therefore, when critical information that is related to digital crimes is collected as PDE, it is crucial to record timestamps. There are different digital sources in the cloud that are able to record digital events, and environmental factors

like the variations in time zones are able to clock the location of the incident and other human factors like clock interference. Moreover, proper recording of appropriate evidence provenance is required to avoid timestamp ambiguity, which may lead to inconsistency in timelines. Evidence provenance in this context has been used to show the origin of the digital files that can be used to aid in extracting the log of events.

The time reference and the format of a timestamp are easily determined by the digital source or origin and by the existence of some digital event that can be used by a digital forensic investigator to prove that an event happened at a certain time.

5.3.1.6 Digital Evidence Characterisation

Characterisation of digital evidence is a means of grouping the evidence based on useful file formats for forensic analysis purposes (Kebande & Venter, 2015). During the analysis phase, Digital Evidence Characterisation (DEC) increases the chances of incident detection by isolating evidence that is relevant from evidence that may be non-relevant. For example, a digital forensic tool may capture unwanted data and while doing sifting; one might lose the exact artefact that is being investigated. Additionally, digital evidence can be characterised by measuring how similar two or more events are. According to the researcher, DEC occurs through clustering of evidence on field names based on occurrence, sources and similarity (Kebande & Venter, 2015). DEC is represented as follows:

$$DEC = \{field_N_1(Fl_1), field_N_2(Fl_2), \dots, field_N_n(Fl_n)\} \quad (5.6)$$

where *DEC* represent the characterised digital evidence, *field_N* represents the field name and *Fl* represents the forensic log.

5.3.1.7 Non-Modification of Existing Cloud Architecture

One important aspect of the CFR model that is presented in this research thesis is the fact that it does not require the functionalities of the existing cloud architecture to be modified. Eventually, this saves much cost and time because there is no need for reprogramming the infrastructure when a software application is used to collect PDE.

Once the collected PDE has been isolated from the cloud environment, any computation, manipulation, examination and analysis, DFR processes or other activities are not applied directly to the retained evidence inside the cloud environment. Since the whole process happens outside the cloud environment, the CFR model is not constrained by specific cloud architecture and it is versatile and flexible. Furthermore, because the researcher's approach of introducing a CFR model does not require the modification of the existing cloud infrastructure, we are still able to capture the required digital data within the cloud.

5.3.1.8 Security Implementation

Security as a requirement in the CFR model is implemented at different levels and these levels are based on how DFR process is executed in the cloud environment. For example, it is important to protect the Forensic Agent (FA) from external attacks. In this context, external attacks may be malicious and aimed at disrupting the FA and its functionalities. Based on the roles of the FA, Funfrocken and Mattern (1999) as well as Wilhen et al. (1999) are of the opinion that an agent is able to move to a number of hosts that use a specialised software and are pre-defined in order to maliciously infect the hosts so that a specific task can be achieved.

To enforce security levels and to deploy the FA in a trusted cloud environment, strong encryption and authentication techniques should be enforced by the CSP to ensure management, authority and maintenance of security keys. Encryption and authentication methods also provide protection to the FA.

5.3.1.9 Obfuscation

The purpose of obfuscation in this context is to deter surveillance of the FA. It is done for privacy and security reasons, i.e. to avoid tampering with the FA and preventing an attacker from disabling or infiltrating the FA. By obfuscating the FA, it is able to run continuously while capturing PDE. Should the FA be disabled, its purpose of finding evidence would be defeated.

When a forensic software application is deployed to the cloud environment, it is crucial to choose a random FA obfuscation vector x^0 that will be used to create an obfuscated software application free from detectors. The researcher represents this by using the following equation:

$$FA^o = OBx^o \quad (5.7)$$

where FA represents the forensic agent, OB denotes the obfuscation process and x^o represents the obfuscation vector. OB can be evaluated using a modified Jacobian equation (Khorram & Moosavian, 2015). Nevertheless, according to Li, Xu, Zheng and Xu (2009), application obfuscation is an application transformation that helps the application to evade detection from malwares through code transposition. OB can be represented as follows:

$$OB = \frac{\partial FA^o}{\partial OBx^o} \quad (5.8)$$

Since the equation takes the obfuscation vector $x^o \in FA^o$ and produces an output with the obfuscation vector $f(x^o) \in OB^o$, the obfuscation matrix for OB can be defined as follows:

$$OB = \left[\begin{array}{cccc} \frac{\partial FA^o}{\partial OBx^o_1} & \dots & \dots & \frac{\partial FA^o}{\partial OBx^o_n} \end{array} \right] \quad (5.9)$$

This can further be broken down so that each component of the FA is shown based on the Jacobian matrix as follows:

$$= \left[\begin{array}{cccc} \frac{\partial FA^o_1}{\partial OBx^o_1} & \dots & \dots & \frac{\partial FA^o_1}{\partial OBx^o_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial FA^o_m}{\partial OBx^o_1} & \dots & \dots & \frac{\partial FA^o_m}{\partial OBx^o_n} \end{array} \right] \quad (5.10)$$

By using the obfuscation vector x^o , it is possible to obfuscate ∂FA^o as a forensic agent in the cloud environment based on ∂OBx^o . The signatures of the forensic agent will be hidden from malwares, virus scanners and detectors in order to avoid detection. Therefore, using x^o as the obfuscation vector, the FA can be obfuscated using the following equation:

$$FA = [OB + (x^o)] \quad (5.11)$$

Taking the obfuscation vector into account, obfuscation of the software application in the cloud environment for information gathering FA is represented as follows:

$$OB = C[FA + x^0] \quad (5.12)$$

Where C represents the environment (cloud) and Equation 5.12 represents the obfuscated FA . The technique illustrated in Figure 5.1 shows how an application gets obfuscated and how it “infects” the virtual instances of computers in the cloud environment towards attaining DFR. Note that infection, which normally has a cynical implication, is used for good purposes in this case to capture digital forensic information.

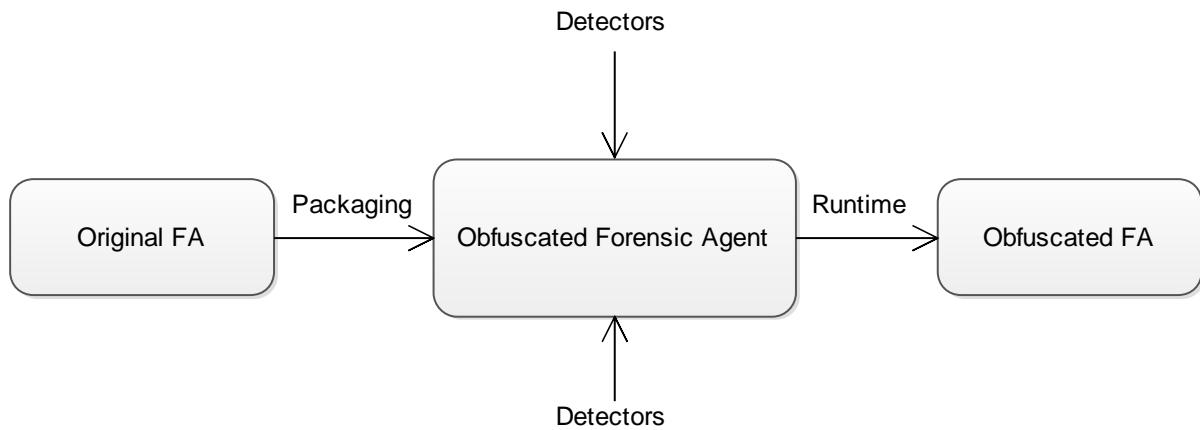


Figure 5.1 High-level View of the Process of Obfuscating a Forensic Agent

The rectangle on the left in Figure 5.1 portrays an application code that undergoes obfuscation. The original FA shown in that rectangle represents an original software application whose patterns have not been changed. It is the code that undergoes pattern changing. Pattern changing is a process that changes the structure of the code without changing the functionality. Pattern changing is the renaming of the symbols stored in a metadata in a nonsensical manner (Kebande & Venter, 2015). This is clear from the packaging process shown by the arrow pointing to the middle box labelled 'obfuscated FA '. The detectors (virus scanners, malwares, Trojans, etc.) are expected to detect the malicious code pattern of the FA . The obfuscated FA represents a hidden code whose patterns have been changed. This allows the structure of the FA to change without changing its functionality in order for the software application to run in stealth mode. The outcome is the process of obfuscated digital evidence capturing FA within the cloud.

At runtime, an infecting obfuscated code is generated in the rectangle on the right-hand side. This is the non-malicious obfuscated *FA* that will be deployed (as a software application) to infect virtual instances in the cloud environment after a stealth installation.

5.3.1.10 Event Reconstruction

According to Carrier and Spafford (2004), event reconstruction is used to examine the evidence and identify why it portrays specific characteristics. Event reconstruction provides a thorough examination and analysis of all the events by revisiting their characteristics and determining the sequence of events. This is done by means of checking if the gathered PDE satisfies admissibility and why that particular evidence must be considered as evidence. Furthermore, it helps to identify if a causal relationship exists between the events in the sequence.

Event reconstruction in a CFR model is also presented by KEBANDE and VENTER (2015) as a process that is able to distinguish events, discover the structure of events and distinguish one event from the other. This is done by focusing on the relationship that exists between the events and how the behaviour of events can be predicted. Carrier and Spafford (2004) argue that event reconstruction is able to question why PDE is treated as an object, why it has some properties and where those properties originate from. Hence it is the researcher's opinion that every reconstructed event should have a cause, and characterisation of the potential events should help to identify this causality.

As a consequence, event reconstruction in this context is used for purposes of developing a hypothesis that is used to answer forensic questions about a crime in a court of law. This hypothesis is built on the digital evidence that is collected by the software application from the cloud environment. More so, this evidence is perceived as potential evidence that plays a role in an event where a potential security incident was detected. This is very important, because a digital forensic investigator must be able to defend the hypothesis by proving or disproving the existence of the evidence.

One would want to know why the characteristics of PDE are of so much importance while performing event reconstruction. When digital devices are visited, a trace must always be left somewhere. Digital data may reside anywhere, and electric signals are easily converted into

digital data (Carrier & Spafford, 2004). Since digital data possesses many properties, the characteristics of PDE are used in the identification of data.

Table 5.1 Summary of General Model Requirements for Achieving Digital Forensic Readiness in the Cloud (Kebande & Venter, 2016)

	Requirement	Summary
1	Organisation requirement	It is a requirement for organisations to collect potential evidence that can be used for an investigation if an incident is detected.
2	Digital forensic governance	Complies with standardised digital forensic readiness processes and the code of practices for security in the cloud.
3	Forensic logging capability and management	Forensic logs to be used as digital evidence should be collected in a virtualised environment and managed effectively. It is also important to know how logging is done, what is logged and when to log.
4	Digital preservation	The retained digital evidence should be digitally preserved.
5	Timestamping	Each log should have a timestamp in order to prove its integrity. All events and activities should have timestamp representation.
6	Digital evidence characterisation	Digital evidence should be grouped in respective file formats for possible incident identification. Activity analysis should be conducted to isolate potential security incidents.
7	Non-modification of existing cloud architecture	Functionalities of existing cloud architecture are not modified or tampered with and this can only be possible since activities like computation of evidence and analysis are conducted outside the cloud environment..
8	Security implementation	The software application solution should be protected against other malicious activities. The software application should be deployed in a trusted environment
9	Obfuscation	Software applications' patterns are changed in a nonsensical manner to deter surveillance. Obfuscation is enforced for privacy reasons.
10	Event reconstruction	A hypothesis that should prove a fact in a court of law should be developed based on events. The structure and occurrence of events should be easily distinguished.
11	Constitution and statutory provisions	The legal perspective and provisions across diverse jurisdictions should be known prior to a digital forensic investigation.
12	Forensic readiness reporting	A forensic readiness report should be generated that shows the interpretation process as a result of digital evidence retention.

5.3.1.11 Constitutional, Legal and Statutory Provisions

The constitutional, statutory provisions and legal requirements on admissibility of digital evidence vary across different jurisdictions throughout the world. There are numerous

constitutional and statutory provisions with regard to how digital evidence should be acquired from CSPs. For example, in the USA, privacy protection of the citizenry to prevent government intrusion has been highlighted by the Fourth Amendment Thompson (2013). However, Rule 16 of the USA’s Federal Rules of Criminal Procedure has a provision that enables a responding party to request data that is in possession by another party. The CSPs and the cloud clients are considered the responding party in the context of cloud computing.

5.3.1.12 Forensic Readiness Reporting

It is a requirement to have a forensic report that shows possible causality before the commencement of a DFI. In order to have proper forensic planning and preparation in place before or during a DFR process, there should be a forensic report that shows descriptions, steps and activities that are taken towards analysing and examining digital evidence. Forensic reporting as presented by ISO/IEC 27043: 2015 is a process of interpreting the results obtained from digital evidence. This involves the roadmap of the readiness process and the event properties that are interpreted and presented to digital forensic investigators and LEAs.

5.3.2 Architectural Requirements

The purpose of this section is to propose architectural requirements for realising a CFR model that allows DFR processes to be achieved effectively in the cloud environment without necessarily modifying the functionality and/or infrastructure of the existing cloud architecture. The nine architectural requirements of a CFR system have been divided into two groups: four Functional Requirements (FRs) and five Non-Functional Requirements (NFRs) (see Table 5.2). FRs are shown in the upper half of Table 5.2, while NFRs are shown in the lower half. FRs are next discussed in Section 5.3.2.1 and NFRs in Section 5.3.2.2 respectively.

Table 5.2 Functional and Non-functional Requirements

	Functional Requirements	Summary
1	Standard implementation of DFR in the cloud	The model should allow standardised DFR processes to be implemented in the cloud.
2	Collaboration with legal competent bodies	The CFR model should allow collaboration and interaction with LEA, DF investors and other investigation bodies.
3	Incident response procedure	When potential digital evidence is availed IRP processes should be able to be conducted.

4	Efficacy and ease of use	The CFR model should be effective by providing an interface for conducting digital investigation in the cloud.
Non-Functional Requirements		Summary
1	Scalability	The DFR process should be able to accommodate demands of users and DF processes.
2	Security	Forensic services or forensic agent solution should be prevented from other malicious attacks.
3	Usability	Forensic processes should be relatively easy to use for novices.
4	Flexibility	The model should enable proper execution of tasks and be able too incorporate other investigative technologies.
5	Auditability	Once forensic process have been conducted, an audit of the processes should be conducted.

5.3.2.1 Functional Requirements (FRs)

In this section, the researcher presents a description of the system architectures' Functional Requirements, mainly the primary system requirements. According to Pohl (2010), FRs are statements of services that the system is supposed to provide. When FRs are met, it means that a system is able to behave as required. Pohl (2010) adds that meeting its FRs means that a system is able to react when it receives particular inputs in different scenarios. Table 5.2 shows the list of FRs that the architecture deals with.

5.3.2.1.1 Standard Implementation of DFR in the Cloud

A CFR model should allow a standardised DFR process to be implemented in a cloud environment. Furthermore, the standardised implementation of DFR in the cloud allows collected evidence to be accepted as admissible evidence. The DFR process proposed in this thesis complies with the ISO/IEC 27043: 2015. It allows a forensic agent that was initially considered to be malicious to be used as a software application that is able to collect relevant PDE. This is a proactive approach that can be used to prepare and plan for potential security incidents in the cloud environment. The benefit of this process is that it requires no modification or alteration of, or tampering with the functionalities of the existing cloud architecture – simply because all DFR activities are conducted outside the cloud. Nonetheless, the DFR process also adheres to a standard process highlighted in ISO/IEC 27017 and ISO/IEC 27001: 2013 international standards. ISO/IEC 27001 highlights the specifications and requirements for an Information Technology (IT) security management

system (ISMS). The standard stipulates that such a system should involve the detailed design of an intensive security model. Furthermore, it also contains a provision regarding the CSPs' responsibility to log and analyse activities performed by their employees so that they can prevent fraudulent activities and unauthorised access (ISO/IEC 27001: 2013).

5.3.2.1.2 Collaborative with Competent Legal Bodies

A CFR model should allow interaction and collaboration with LEAs, DFIs and other competent investigation bodies within a given jurisdiction. One is able to conduct competent investigations by maintaining the chain of custody through the collaboration of distributed DF investigators. The DF legal framework should provide rules that allow the admissibility of digital evidence if it is lawfully admitted in a court of law during a trial. Although it does not guarantee full control over the processes and evidence in the cloud, the architecture should be relevant and have a scope of collaborating with regard to identification of the digital artefacts.

5.3.2.1.3 Incident Response Procedures

Incident Response Procedures (IRPs) are DF tasks that are associated with competent bodies and that contain instructions for detection and response to potential security incidents. One should be able to perform IRPs when digital evidence is available that has been collected through DFR process. A CFR model should ensure that IRPs are adhered to during the collection and preservation of digital evidence. This should ensure that the integrity of PDE remains immaculate during an investigative process so that PDE can be accepted as admissible in a court of law.

5.3.2.1.4 Effectiveness and Ease of Use

A CFR model should allow effective communication between the different tasks by providing an interface when preparing the cloud for digital investigation. The model should be found user friendly and simple whenever users interact with it, and it should help preparing for the investigation of security incidents and completing it within a shorter time.

5.3.2.2 Non-Functional Requirements (NFR)

According to Malan and Bredemeyer (2001), Non-Functional Requirements (NFRs) are used to describe various constraints and qualities of a system in which stakeholders are interested. This means that NFRs have the capability to affect the stakeholders' degree of satisfaction, which eventually implies that the NFRs should be prioritised in any system. As a result, an investigation by the researcher identified scalability, security, usability, flexibility and auditability as the NFRs that the CFR model has to meet. A summary of the NFRs also appears in Table 5.2.

5.3.2.2.1 Scalability

The process of conducting DFR in the cloud should accommodate the demands of users and DF processes. The CFR architecture should be sound enough to meet the needs of emerging processes within the shortest time possible. Additionally, the CFR model should be able to withstand overstraining and tolerate errors. If the system is not able to scale to the business volume, then obstacles may arise that may hinder the DFR process.

5.3.2.2.2 Security

Security as a requirement ensures that the forensic services are prevented from potential attacks. It ensures that a forensic agent solution is protected from malicious attacks that might want to infiltrate it or defeat its purpose. Tampering with digital evidence is one aspect through which the security of the system might be compromised. If security is not enforced at all levels, then the contamination, tampering or theft of forensic evidence might be experienced. Ritcher, Kuntze and Rudolph (2010) believe that for digital evidence to be admissible in a court of law, the system must be secure and the data within must be authentic and possess integrity.

5.3.2.2.3 Usability

For any system to be effective, it must be tangible and relatively easy to use. Novices should be able to easily learn the forensic processes and tasks that are based on this model. The same should apply to more experienced forensic experts, where the CFR model should allow the users to gain an understanding of exactly what the intent of the system is. Different collaborating legal bodies in different jurisdictions should also be able to interact well, as

well as other casual users. If usability is not enforced properly, the performance objectives pursued by a user and the forensic tasks that he/she might wish to perform may be hindered.

5.3.2.2.4 Flexibility

Carrier and Spafford (2004) suggest that a framework that needs to support future technologies should be flexible enough. Hence, a DF process should incorporate the quality of flexibility. This enables proper execution of DF tasks and helps to incorporate other investigative technologies at the same time. In addition, flexibility should ensure the ability to support the system processes by reacting fast to internal and external changes.

5.3.2.2.5 Auditability

One of the CFR model's requirements that was discussed earlier in this chapter, is the ability of the system to perform forensic logging. The collected forensic logs are then isolated and used as PDE. Once the investigation has been closed, one should be able to perform an audit of the processes conducted by the system – either during or before PDE presentation. Irrespective of that, auditing may also be conducted by using the reconstructed events as mentioned in the CFRaaS model that was presented in Chapter 7 of this research thesis. Therefore, it is a requirement that an audit must be conducted after forensic processes have been completed.

Having looked at the model requirements that are needed in order for the cloud to achieve forensic readiness, the chapter is concluded next.

5.4 Conclusion

The chapter introduced the reader to the requirements needed for the achievement of DFR in the cloud environment: General requirements and architectural requirements. The researcher described each general requirement and gave a detailed summary in Table 5.1, followed by a highlight of architectural requirements and a detailed summary in Table 5.2. Table 5.1 summarises the general requirements that are needed by the CFR model to achieve DFR in the cloud environment when a software application is used as a forensic agent. The first column shows the respective requirements, while the second column contains a specific summary of each requirement. Likewise, a detailed explanation of the architectural requirements presented as functional and non-functional requirements was given too.

The reader should now have an understanding of the requirements for realising the CFR model. In the next chapter, the reader is introduced to hypothetical case scenarios that will be used in the remainder of this research thesis.

Chapter 6: Hypothetical Case Scenarios

6.1 Introduction

This chapter presents hypothetical case scenarios of digital crimes that are linked to the cloud environment. The scenarios concern fictitious companies and they have been used to show the applicability of the CFR model that will be proposed in the subsequent chapters. The first hypothetical case scenario explores an information and security breach in the cloud environment and the problems that relate to the acquisition of forensic data. The second scenario involves using the cloud as a target for conducting digital crimes. The third hypothetical scenario explores sexual harassment and child pornography in the cloud.

The researcher considers these three hypothetical case scenarios because they portray the need for DFR techniques and processes. The case scenarios also highlight what would be needed if the cloud were to be made forensically ready for digital investigations. Furthermore, the scenarios give guidance to the reader on the required standard investigative strategies that are needed to identify potential security incidents and intrusions in cloud-based environments. Because of its prevailing characteristics such as lack of transparency and an open nature, the cloud can sometimes be (mis)used by cyber-criminals to conduct digital crimes. Thus, it is important for organisations to enforce DFR as a mitigation strategy and the researcher proceeds to highlight how DFR can be adopted as a cost-effective approach in any organisation. Since organisations that lack DFR are unable to collect, digitally preserve, examine and analyse PDE, such organisations ultimately become unable to uncover important digital information about security incidents.

Before the solutions aimed at achieving DFR in the cloud are explored, the researcher first introduces the above three typical investigative hypothetical case scenarios that support and explain the research presented in this thesis. The case scenarios represent fictitious companies and show how a lack of incident preparedness can have extremely serious security consequences. The scenarios are repeatedly referred to throughout the thesis and a full description of each is given in the next section.

The remainder of the chapter is structured as follows: The hypothetical case scenarios are presented in Section 6.2 – investigative case scenario I, II and III in Section 6.2.1, 6.2.2 and

6.2.3 respectively. Section 6.3 subsequently discusses observations made on the basis of these case scenarios. The chapter is concluded in Section 6.4.

6.2 Hypothetical Investigative Case Scenarios

The researcher presents hypothetical investigative case scenarios that assisted him to test for the requirements and specifications by using experiments presented in the subsequent chapters of this research thesis. The Oxford Dictionary defines a scenario as "the script that has details of a sequence of future events that are imagined" (Oxford, 2007). Therefore, the hypothetical case scenarios that are being presented in this section do not reflect any specific incident from any organisation. They are purely hypothetical, generic and fictitious. However, they help to highlight the existence of digital crimes in cloud environments that are very difficult (or impossible) to investigate with conventional digital forensic investigation techniques.

Even though fictional, all the hypothetical case scenarios are presented as cloud-related crimes and they have been used to represent crimes that happen in everyday life. The following issues experienced in these scenarios have helped to distinguish between the kind of digital forensic investigation that was employed in this research thesis and traditional digital forensic approaches.

- Availability of digital evidence when needed.
- Collaboration by the Cloud Service Providers (CSPs).
- Difficulty in acquiring digital forensic information.

The above-mentioned issues are also portrayed in each of the hypothetical case scenarios below.

6.2.1 Case Scenario I: Information Security Breach and Identity Theft

It is common in a corporate environment to find adversaries with malicious intent who can make an organisation a target of identity and information theft through unauthorised login. Adversaries can do this by planting malware that is able to collect the administrator's login details and report back to them, which eventually helps adversaries to access confidential

information. The researcher proposes the following ‘consumer-provider’ investigative scenario.

The scenario involves BlueBerry, a purchasing and logistics company that was able to gather and store its employees' information without protecting the confidentiality of such information, thereby putting the privacy of all its employees at risk. Nax.com was a CSP that offered private cloud services to various clients, including the clients in a company called BlueBerry. BlueBerry set up a service in the cloud and its employees were able to access these services promptly. BlueBerry furthermore had approximately 5000 employees who were in possession of BlueBerry's corporate credit and debit cards. BlueBerry outsourced the company's data and applications to be managed by the CSP. This allowed the data of BlueBerry's employees to be stored across multiple servers in the cloud environment without any control over the location of their data.

For a period of time, there were no signs of any suspicious activity within BlueBerry that could trigger any digital investigation – until a number of employees raised issues related to the following: Invalid transactions because the card could not support the type of transaction; debit card transactions that returned a statement reading “Transaction Not Allowed”; lack of sufficient funds in the debit cards because the credit limit had been reached; and unauthorised transactions displayed in the bank statements of the clients. One employee officially lodged a complaint to the client service department that stated the following issues:

- When he tried to withdraw some amount using his debit card, he realised that the account balance was below what he expected and the account statement showed 10 consecutive withdrawals that he had not made.
- Each withdrawal was done between 12:00 midnight and 12:01 am of the next day over a span of one week.
- The monthly statement also showed a number of unauthorised purchases that were made in another country to which the client had never travelled.

BlueBerry gradually came to suspect that there was a potential security breach in its computer system. Firstly, the CEO of BlueBerry thought it was an inside job by someone on the inside with access privileges who had colluded with the hackers to steal employees' confidential

information. However, he was not sure and without taking chances BlueBerry decided to hire security consultants and digital forensic investigators. The Chief Executive Officer (CEO) also involved the LEAs and informed financial bodies (like the banks that issued the credit cards) about the possible security breach. BlueBerry chose not to disclose this potential security breach to the public until the digital forensics investigations had been completed, because there was no proactive monitoring of event data or daily transactions in the system and the company feared losing its credibility.

BlueBerry disclosed the number of cards affected to the investigators as 2000, and then decided to file for protection in accordance with the law to prevent further disclosure of the attack details to the public. The affected banks and financial institutions, however, sued BlueBerry and put the number of affected cards at 4500. This discrepancy highlighted the fact that BlueBerry and its CSP did not have any accurate log data that could be used for forensic analysis.

A DFI should be triggered by the occurrence of such incidents. A more significant issue that should worry digital forensic investigators and LEAs in this case scenario, would be to identify who really was responsible for these security attacks. Where did the attacks originate and how could they be prevented in future?

6.2.2 **Case Scenario II: Intrusion, information theft, information tampering and framing**

In the second scenario, the cloud was used as a target for committing cyber-crime. The scenario involves a situation whereby a disgruntled employee (*W*) exploited a security administrator's system in the cloud and stole confidential information. *W* then wiped all traces of evidence and managed to shut down his Virtual Machine (VM). Later on, *W* was able to frame the administrator (*H*) in what led to the arrest of an innocent person.

5 August 2016 was *H*'s first day to work as a computer security administrator for company *PQR*. *PQR* dealt with electronic supply chain systems whose data had been stored in the cloud environment. *PQR* had different trading partners who could conduct e-commerce in terms of SLAs. As a security administrator, *H* had to manage the security of all the retailers' confidential information and transactions, and he also had to keep track of possible vulnerabilities that could leave the supply chain process vulnerable to attack.

H gave his job top-level devotion because of the discretion of the IT system's trading secrets. The nature of this job allowed information sharing and interconnection among trading partners, financial institutions, manufacturers, vendors, associates and customers. In November 2016, *H* became aware of a possible security intrusion. While performing a routine system security check, he discovered that a new program called “iscanned” had altered a number of files. The altered files had their format changed and the size of the files had increased rapidly. He was concerned because he had personally kept track of all the programs that had been installed in the system.

Before *H* could report this matter to Manager *M*, he decided to do a preliminary scrutiny of the system to try and see if he could find traces of potential evidence. After performing a series of tests, he discovered that a number of critical information sources had been deleted, but he could not trace the origin of the IP address that performed this malicious action. Not being able to find any further trace, *H* decided to inquire about the attack from *W* who was working as his immediate supervisor, unsuspecting that *W* was a disgruntled employee and that it was she who had accessed the system remotely from the cloud. *W* had remotely installed a rootkit. The rootkit was able to delete information and steal confidential information, and afterwards *W* was able to cover her traces by shutting down the VM. However, *W* was not able to cover the traces of his IP address entirely. When *H* reported the matter to *W*, *W* told *H* that it was possible to run a scan in order to check for possible causes. *H* agreed and *W* instead used the cloud to install another stealth program that wiped all his traces and instead showed that the attack originated from *H*'s system.

The pilot investigation by the two employees found that there was no potential digital evidence. Knowing the obligations involved in safeguarding customers' confidential information, they reported the matter to Manager *M*. *M* triggered a digital investigation immediately by informing digital forensic investigators and the LEAs. A preliminary examination on *H*'s system revealed that the attacks originated from his system and he was responsible. *H* was arrested immediately while digital investigators continued with further digital forensic investigations. Company *PQR* tried to ascertain the reasons as to why their security administrator was able to compromise their system.

Why was *W* able to steal very confidential information? In this instance there was not any forensic process that could collect valuable information in real time or remotely while the intruder was planting the malware. Thus, was company *PQR* forensically prepared for any of these incidents? Was it fair that *H* would be charged and eventually serve a sentence for a digital crime that he did not commit?

6.2.3 Case Scenario III: Sexual harassment, child pornography and framing

The scenario involves a sexual harassment case and a child pornography aggravated paedophile that ends up framing his immediate neighbours for a crime they did not commit. *X* and *Y* stayed in *RCY* neighbourhood which was enjoying cloud computing services from *S_NET.com*. At the same time, *P* who was a morally corrupt paedophile was also living in *RCY* neighbourhood. *X* and *Y*, the parents of a 6-year-old child, caught *P* red-handed trying to molest the 6-year-old victim. *X* who was the child's mother, decided to report the matter to the LEAs. As a result of his creepy behaviour, *P* was forced to sign the sex offenders' register once every week for a period of two years. This angered *P* immeasurably and as a result *P* was looking for vengeance against *X* and *Y* who were his immediate neighbours, unfortunately, *X* and *Y* had no immediate plans of relocating from the neighbourhood.

At a given instance, *P* decided to set up a service in the cloud in order to retaliate against the initial actions of *X* and *Y*. *P* started his campaign of vengeance by downloading Wi-Fi hacking software and he managed to crack *X* and *Y*'s Wireless Equivalent Privacy (WEP) key. This enabled *P* to get hold of their unique router ID, and he next managed to open a fake account on the social site TB.com, bearing *X*'s name while in the cloud. Thereafter, he was able to rent a good amount of space in the cloud and started posting pubescent explicit videos and pictures of young boys and girls. Afterwards, *P* managed to make a collection of child pornography materials which he emailed to *X* and *Y*'s workmates, including the CEO of companies *ABC* and *DEX* respectively, using their router as a host. (Keep in mind that *X* and *Y* were working in company *ABC* and *DEX* respectively.) Due to these insensitive acts, *X* was fired from her workplace immediately and *Y* was summoned to give an answer about the emails in his workplace. *Y* denied everything and maintained that he was not responsible for any wrongdoing.

ABC decided to seek the services of a digital forensic investigator. At first the investigators tried to trace the IP address in *X* and *Y*'s network in order to get to the root cause of these incidents. They failed because *P* had covered his tracks well. A preliminary inspection by a digital forensic investigator in TB.com also failed to yield any information, because there were no credible packet logs that could show the e-mail sessions that were sending the child pornography material.

Eventually *X* and *Y* were held responsible for the emails and the fake accounts set up for child pornography. They were each charged on two counts of child pornography and were later sentenced to five years in jail each.

6.2 Observations

Each of the aforementioned scenarios has highlighted the need for potential digital forensic evidence in an organisation that has embraced cloud computing. Moreover, the case scenarios are tailored towards the lack of DFR in the cloud environment and identify broad issues that need attention from the forensic side. In this context, a fair degree of digital forensic preparation is needed in any organisation to curb or mitigate the adverse impact of a major security incident as was illustrated in the case scenarios.

Digital forensics is a very thorough process of investigation that requires great cost and time. At some point, the inability to unearth exact evidence might lead to wrongful or unsuccessful prosecution, as in the particular cases illustrated in the case studies above. Within the cloud, instances may get lost as soon as they appear, which might make it hard to extract evidence during the digital investigation process. Nevertheless, if an adversary decides to terminate a VM and the CSP decides to tamper with the logs, then an investigator may face serious challenges. This is because digital evidence at some instance may actually be the image of a VM, logs owned by CSPs or even the files that are stored somewhere in the cloud. It is evident nowadays that investigators depend on the CSPs to get logs that can be used as evidence (Dykstra, 2013). A malicious activity performed inside a VM or incidents like in the case scenarios could possibly be traced from the logs that would have been collected by the CSPs. If an investigator needs access to specific logs, the investigator should issue a subpoena to the CSP, and once the investigator gets access to logs, it becomes the task of the investigator to prove that an incident occurred. However, what would happen in a case where

incident planning and preparedness are lacking? Answers to this uncertainty are given in the subsequent chapters of this research thesis.

If the above case scenarios are to be investigated in a reactive process and presented in a court of law, then the defence team would retaliate by disregarding the acquired evidence and using the following questions:

- ✓ Can the investigator prove that the digital evidence is admissible in a court of law? Can he provide logs to prove provenance of the data?
- ✓ How can the investigator prove the integrity of the logs? If the logs were tampered with, how can their correctness be proved?
- ✓ Can the investigator prove the physical location of a VM if the VM was terminated?
- ✓ Were there standardised processes that could have been used in digital preservation, documentation, maintaining the chain of custody, documentation and managing of information flow, while conducting a digital investigation process?
- ✓ If the LEA manages to get a subpoena against Nax.com and S_NET.com, would prosecution be allowed to continue if it was a case of cross-cutting or multi-jurisdiction?

Based on the hypothetical case scenarios that have been presented, it is evident that the CSPs have greater control over the generation of logs that can be used in the digital forensic process. Nevertheless, proactive forensic logging is essential in the process of digital investigation. The researcher highlighted problems that arise as a result of using the cloud environment without being forensically ready.

The reader was introduced to hypothetical case scenarios that helped them to understand the need for DFR in any organisation. The next section concludes the chapter.

6.3 Conclusion

Chapter 6 presented typical investigative hypothetical case scenarios. In the first scenario, the researcher presented a case of intrusion that led to the theft of personal information that was stored in a company's central database. In the second scenario, the researcher presented a case where a disgruntled employee planted malware on a security administrator, which led to his arrest. Finally, in the third scenario, a sex offender hacked into the network of his

neighbours after he was caught trying to molest their 6-year-old child. The investigative scenarios presented in this chapter will constantly be referred to in the rest of this research thesis. Suggestions for improving the DFR investigative process will also be made, so as to help organisations to be forensically ready for operating in the cloud.

In the next chapter the researcher introduces the proposed model.

Chapter 7: Cloud Forensic Readiness as a Service (CFRaaS) Model

7.1 Introduction

The previous chapters introduced and discussed various literature that were consulted in this research thesis. Chapter 5 discussed the requirements for the Cloud Forensic Readiness as a Service (CFRaaS) model in order to achieve DFR in the cloud environment. In this chapter, the researcher proceeds to present a CFRaaS model that is aimed at achieving these requirements and specifically attempts to realise the proactive process of DFR in the cloud environment.

As the cloud steadily becomes ubiquitous, trust needs to be built and CSPs are increasingly concerned about how to control security incidents because criminals are progressively using the cloud as a platform for launching unprecedented attacks. Trenwith and Venter (2013) argue that employing a traditional DFI model centred on searching and seizing digital evidence might not scale up well, due to the existence of a large number of data centres and hosts in the cloud. Consequently, based on the limitations of the existing models and the lack of DFR models in the cloud at the time of writing this research, it is essential to conduct an investigation and propose a suitable model with digital forensic capability that is focused on cloud computing environments.

The main aim of the proposed CFRaaS model is to make the cloud environment forensically ready for DFIs, through compliance with standardised guidelines that have been highlighted in the ISO/IEC 27043: 2015. (The latter advocates the use of DFR principles.)

The remainder of the chapter is structured as follows: A formal description of the cloud model based on formalisations appears in Section 7.2, while the formalisation of the cloud architecture is discussed in Section 7.3. This is followed by a high-level overview of the model in Section 7.4 to give the reader a holistic understanding of the model. Section 7.5 discusses the detailed CFRaaS process model design, while Section 7.6 provides a

comparison between the proposed CFRaaS model and other existing forensic readiness models. The chapter ends with Section 7.7.

7.2 Formalisation of the Cloud Model

The researcher uses the formalism that is based on the actions between the Cloud Service Providers (*CSPs*) and cloud clients (*Cl*s) to logically model a formal cloud where the CFRaaS model is based. In the cloud environment, interactions occur between the *CSP* and the *Cl*s. Logically, the underlying infrastructure between these cloud-based technologies (the *CSP* and the *Cl*s) can be separated through the concept of virtualisation, which is represented as loose coupling (Gong et al., 2010). Loose coupling allows different components in a system to interconnect for purposes of interdependence. In this context, the cloud services and applications are offered by the *CSP* and the *Cl*s. The cloud services are able to interact with the cloud servers and data centers. (Keep in mind that the cloud operates on the client-server architecture.) Consequently, the *Cl*s do not have any control over the data in the cloud. To present a formal logic model of the interactions between the *CSP* and the *Cl*s that support the CFRaaS model, the researcher describes a Cloud Model (*CM*) that is represented by a *Cl*s and *CSP*s as shown in Equation 7.1:

$$CM = \{CSP_1, CSP_2 \dots CSP_m\}, [m \geq 1] \quad (7.1)$$

where

$$CSP = \{Cl_1, Cl_2 \dots Cl_p\}, [p > 0] \quad (7.2)$$

Based on Equation 7.1 and 7.2 of the *CM*, Gong et al. (2010) highlight the fact that coupling between entities can be represented as a set. It is clear from Equation 7.1 that the *CM* is made up of *m* number of *CSP*s, which implies that in every *CM* there should exist $m \geq 1$. This means at least one *CSP* should exist in a *CM*. Based on equation 7.1 and 7.2, the *Cl*s and the *CSP* are represented as a set as shown:

$$Set(Cl_i, CSP_n)$$

This is then followed by showing the independence of the *Cl*s and the *CSP*s which is shown in equation (7.3) and (7.4) respectively.

$$Cl_i \cap Cl_n = \varnothing, (0 \leq i, n \leq p, i \neq n) \quad (7.3)$$

$$CSP_i \cap CSP_n = \varnothing, (0 \leq i, n \leq p, i \neq n) \quad (7.4)$$

The above implies that the sets of *Cl*s and the *CSP* are independent among *n* number of clients and *n* number of *CSP*s; however, when they are loosely coupled, the *Cl*s are able to connect to the *CSP*s as shown in Equation 7.5. Consequently, the interconnection between the *Cl*s and the *CSP*s that shows how independent each entity is has been shown in Equation 7.5.

$$Set(Cl_{i1}, CSP_{n1}) \cap Set(Cl_{i2}, CSP_{n2}) = \varnothing, [0 \leq i, n \leq p, i \neq n] \quad (7.5)$$

This shows that the *Cl*s of the cloud are able to access the multiple provisioned services in the cloud at the same time; however, the data centres remain independent, irrespective of the cloud deployment model. Based on the formalisms that have been highlighted above, it is evident that in the *CM*'s logic plays a big part during the separation of the infrastructure and the process of virtualisation. Additionally, the cloud platform is represented as an abstract layer that is capable of separating a variety of applications that run in the *CM*. In spite of that, once the *Cl*s are loosely connected to the *CSP*s, the *Cl*s have no control over the data.

7.3 Formalisation of the Cloud Architecture

In this section, the researcher provides a formalism based on the entities of the cloud architecture. This formalism gives a description of the entities that allow the normal operation of the cloud architecture. Mathematical formulations have also been employed coupled with set theory. Based on the formalism that has been mentioned, a number of definitions are given too.

The cloud architecture consists of different designs that are aimed at allowing applications to be built on the underlying infrastructure (Varia, 2008). The main role of the cloud architecture in this context is to set a platform through which cloud-based activities can be monitored effectively. The cloud architecture comprises of services that are deployed to deliver the roles of a datacentre where the main goals remain to be reliability, scalability, availability and effectiveness. The cloud architecture also has the following components: a Physical server (*Ps*), a Virtual server (*Vs*), Operating System (*OS*), applications and services (*appns*). The *Ps* comprises of a Datacentre (*Dc*), the *Vs* allows the deployment on VMs and the *OS* allows one to add different *appns* over the internet.

In order to implement DFR in the cloud environment, it is necessary for the cloud to adapt to digital forensic processes; however, the *apps* should also be accessible at the user levels. Based on the Cloud Model (*CM*) that was presented using Equations 7.1 and 7.2, the CSP comprises of a set of clients *Cl* that can be represented as follows:

$$CSP = \{Cl_1, Cl_2, \dots, Cl_i\}, i \in N \quad (7.6)$$

Equation 7.7 shows that the cloud is normally distributed across datacentres (*Dc*) – which can be a limited number that is > 0 . This is represented below:

$$CSP = \{ \{Cl_i \{Dc = \{Dc_1, Dc_2, \dots, Dc_j\}, j \in N, Dc > 0\} \} \} \quad (7.7)$$

In the context of Equation 7.8, the CSPs extend their services to the deployable models in the cloud that are represented below.

$$CSP = \left\{ \begin{array}{l} Cl^{Pr}_1, Cl^{Pr}_2, \dots, Cl^{Pr}_n \\ Cl^{PB}_1, Cl^{PB}_2, \dots, Cl^{PB}_n \\ Cl^{HB}_1, Cl^{HB}_2, \dots, Cl^{HB}_n \end{array} \right\} Cl \geq 1 \quad (7.8)$$

where *Cl* is the client that represents the users of the cloud, *Pr* represents the private cloud deployable model, *PB* represents the public cloud deployable model and *HB* represents the hybrid deployable model. Services are deployed to one or more clients as shown in Equation 7.8.

A *Dc* constitutes a network that has a *Vs* that allows the deployment of the VM. In addition, there exists the *Ps* that is able to give support to the OS. In other words, a *Dc* is composed of entities *Ps*, *Vs* and OS respectively. Together with the CSP, this is represented as shown in Equations 7.9 and 7.10:

$$Dc = \begin{bmatrix} Ps \\ Vs \\ Os \end{bmatrix} \quad (7.9)$$

$$CSPs = \{Cl_i\{Dc_j\} = \{Ps, Vs, Os\}, j \in N, Dc > 0\} \quad (7.10)$$

Ps and Vs are able to support a number of cloud resources (R_n) and a VM_n , while the OS is able to support $appns_n$ as shown in Equations 7.11, 7.12 and 7.13 respectively.

$$Ps = \{R_1, R_2, \dots, R_n\} \quad (7.11)$$

$$Vs = \{VM_1, VM_2, \dots, VM_n\} \quad (7.12)$$

$$OS = \{appns_1, appns_2, \dots, appns_n\} \quad (7.13)$$

Therefore, the overall logic formulation for the entities of the cloud architecture is given as follows in Equation 7.14:

$$CSP = \left\{ Cl_i \{ Dc_j \} = \begin{cases} Ps = \{R_1, R_2, \dots, R_n\} \\ Vs = \{VM_1, VM_2, \dots, VM_n\} \\ OS = \{appns_1, appns_2, \dots, appns_n\} \end{cases}, j \in N, Dc > 0 \right\} \quad (7.14)$$

where $Ps = \{R_1, R_2, \dots, R_n\}$ represents a number of cloud resources, and $Vs = \{VM_1, VM_2, \dots, VM_n\}$ represents the VMs that are able to support virtualisation. Finally, $OS = \{appns_1, appns_2, \dots, appns_n\}$ represents the applications and services in the cloud environment. Basically, the formalism of the cloud architecture provides a theoretical approach that is aimed at making the cloud forensically ready for DFIs. The entities Ps , Vs and OS have been employed to help in the execution of cloud services.

Having looked at the formalisation of the cloud model and architecture, the researcher's focus now shifts to the proposed model and in the next section, the reader is introduced to the high-level overview of the proposed CFRaaS model.

7.4 High-Level Overview of the CFRaaS Model

The section presents a high-level overview of the CFRaaS model. The Cloud Forensic Readiness as a Service (CFRaaS) model is a well-defined recurring process model that has been used in a step-by-step approach to forensically plan and prepare the cloud for digital forensic investigations. Additionally, the CFRaaS model has been represented as a proactive process, which means it deals with pre-incident-detection strategies. The high-level CFRaaS model is divided into five distinct layers as shown in Figure 7.1, which enables communication between the other processes.

The layers (labelled 1-5) include: Provider layer (layer 1); Virtualisation layer (layer 2); Digital Forensic Readiness (DFR) layer (layer 3); Incident Response Procedures (IRP) (layer 4); and Concurrent Processes (Layer 5). Layer 3 and 5 correspond with and adhere to the guidance of the Incident Investigation Principles and Processes international standard (ISO/IEC 27043: 2015) while layer 4 is a reactive process. Each of these processes is mentioned briefly in this section, after which each process is discussed in detail in the subsections to follow.

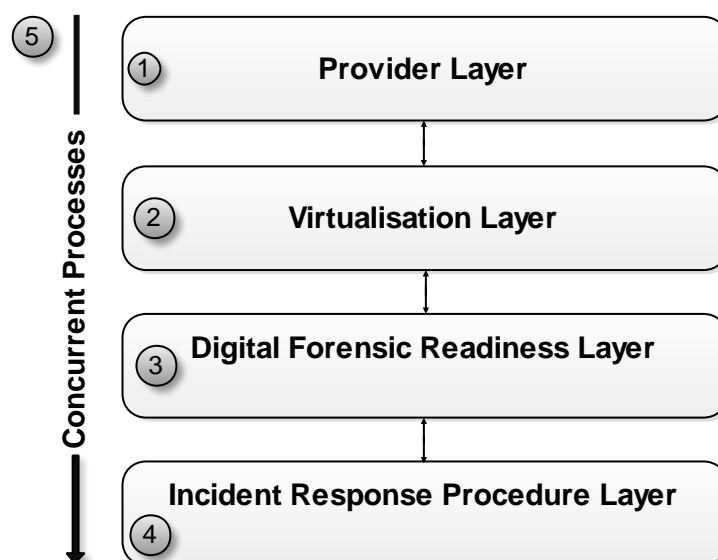


Figure 7.1 High-Level Overview of the CFRaaS Model

The Provider Layer (PL) ensures that the Cloud Service Providers (CSPs) are able to provide services over the internet through virtualisation layer. Next, digital information that can be used as Potential Digital Evidence (PDE) is captured using a bot client which forms part of a Non-Malicious Botnet (NMB) in a DFR Layer labelled 3 in Figure 7.1. Note that the bot client plays the role of a “non-malicious bot” that is deployed in the cloud environment to collect digital information legitimately. The collected PDE is digitally preserved in a forensic database, then pre-analysed for possible incident detection purposes in a DFR approach layer in the process (labelled 3). Finally, the IRP layer (labelled 4) is a reactive process that allows proper forensic examination and analysis of evidence by DF investigators and LEAs.

The arrow pointing downwards in the process (labelled 5), represents *concurrent processes* that are taken verbatim from ISO/IEC 27043: 2015. The concurrent processes are executed simultaneously alongside the other processes shown in Figure 7.1. According to ISO/IEC 27043: 2015, the main aim of the concurrent processes is to assure the admissibility of digital evidence in a given legal system. This can only be achieved by following proper digital-evidence-handling techniques as highlighted by these processes in ISO/IEC 27043: 2015. If these concurrent processes are not employed in a model like this, then PDE may be regarded as unsuitable due to potential improper handling thereof. The *Concurrent Processes* are meant to run in parallel, i.e. concurrently with the other processes. The reason for this parallelism will become apparent when discussed later. Additionally, more details on the concurrent processes are discussed later in the detailed CFRaaS model in this chapter.

The CFRaaS model represents a proactive approach that allows collection of digital evidence from the cloud environment. Such an approach assists organisations to prepare and plan before a security incident can occur. Thus, in the event of one actually occurring, it is possible to gather enough intelligence and store it forensically so that it will not be lost due to the cloud's normally volatile nature. In the context of this research thesis, the researcher employed the notion of a modified form of a botnet through which a bot client, collects PDE based on the digital evidence collection requirements discussed previously.

More high-level details of the composition of the CFRaaS model are discussed in the next section by systematically explaining each of the components mentioned here in detail.

7.5 Detailed CFRaaS Model

Section 7.4 of this chapter gave a high-level overview of the CFRaaS model through a brief description of the various processes that the model consists of. In this section, a detailed CFRaaS model is presented, which is an expansion of the high-level model. (The detailed CFRaaS model is later shown as a block diagram in Section 7.6 in Figure 7.15.)

7.5.1 Provider Layer

The provider layer (PL) (labelled 1) in Figure 7.1 is the business layer that comprises of the services that are provisioned by the CSPs over the internet. In this layer, convenient, secure and reliable services are provisioned to different cloud clients in terms of properly agreed SLAs. Implementing an SLA ensures that the forensic monitoring process is executed, while the clients' data is protected at the same time.

Considering the PL (labelled 1) in Figure 7.2, individual cloud roles and services can be deployed to different consumers, depending on the cloud model and the business requirements that suit a given organisation. Still, depending on the type of workload, the services may be deployed by creating, handling and managing the number of role instances. Furthermore, service orchestration allows a well-planned provisioning and automation of different DFR tasks in the cloud environment that rely solely on the rules for collecting PDE.

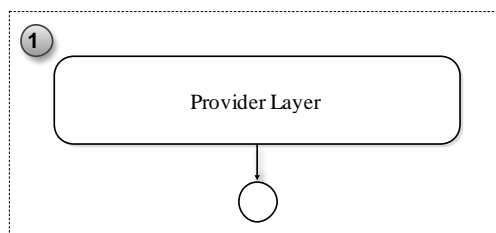


Figure 7.2 CFRaaS Provider Layer

Potential digital information that is related to digital crimes is also collected through monitoring. This is achieved by deploying a bot client, which forms part of a modified form of a botnet that acts as a forensic agent and is able to “infect” VMs in the cloud environment. It should be noted that infection in this context has a positive connotation, and it happens through a well-governed, secure virtual administration process.

7.5.2 Virtualisation Layer

Virtualisation (labelled 2) in Figure 7.1 is an enabler that ensures that resources are scaled within the cloud environment. It gives room for the separation of VMs from the physical infrastructure, which allows PDE to be collected. The VM is isolated in a runtime environment like the OS or applications. Delport, Kohn and Olivier (2011) highlighted the fact that isolating an instance for digital forensic investigations helps to preserve the integrity of forensic evidence. A further advantage of using virtualisation is that it enhances the forensic monitoring process in multiple sources. Moreover, in this context, the virtualised resources are provided as services within the cloud environment.

Virtualisation as shown in Figure 7.3 consists of the hardware, hypervisor, operating system, VMs and an NMB that is executed inside the VM. More details about the NMB that is executed in the VM are explained in the next section. The hardware represents a physical platform which supports the running of the OS. The hypervisor acts as managing or controlling software between the hardware resources and the VMs. Multiple software applications (bot clients) are executed inside the VMs in order to collect forensic information. Through automation, a secure forensic monitoring process is ensured by managing the configuration of VMs and forensic databases as is shown in Figure 7.3.

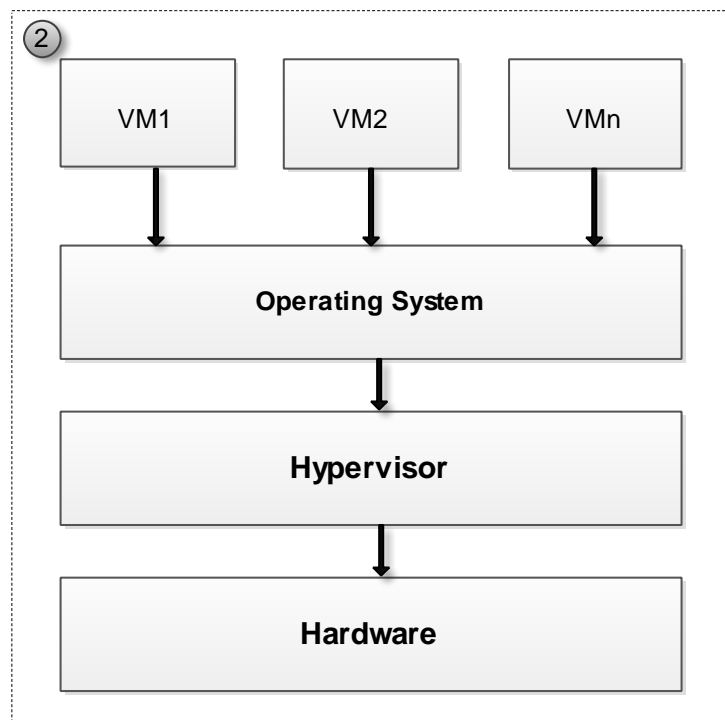


Figure 7.3 CFRAaS Virtualisation Layer

Having considered virtualisation, there is a need for the reader to understand how the NMB performs its purposes; this is discussed in the next section.

7.5.3 Digital Forensic Readiness Layer

In this research thesis, the Digital Forensic Readiness (DFR) layer is presented as a proactive process that happens before incident detection (see Figure 7.4). The uppermost process that is labelled *a* in Figure 7.4 represents the existence of a *forensic readiness policy*.

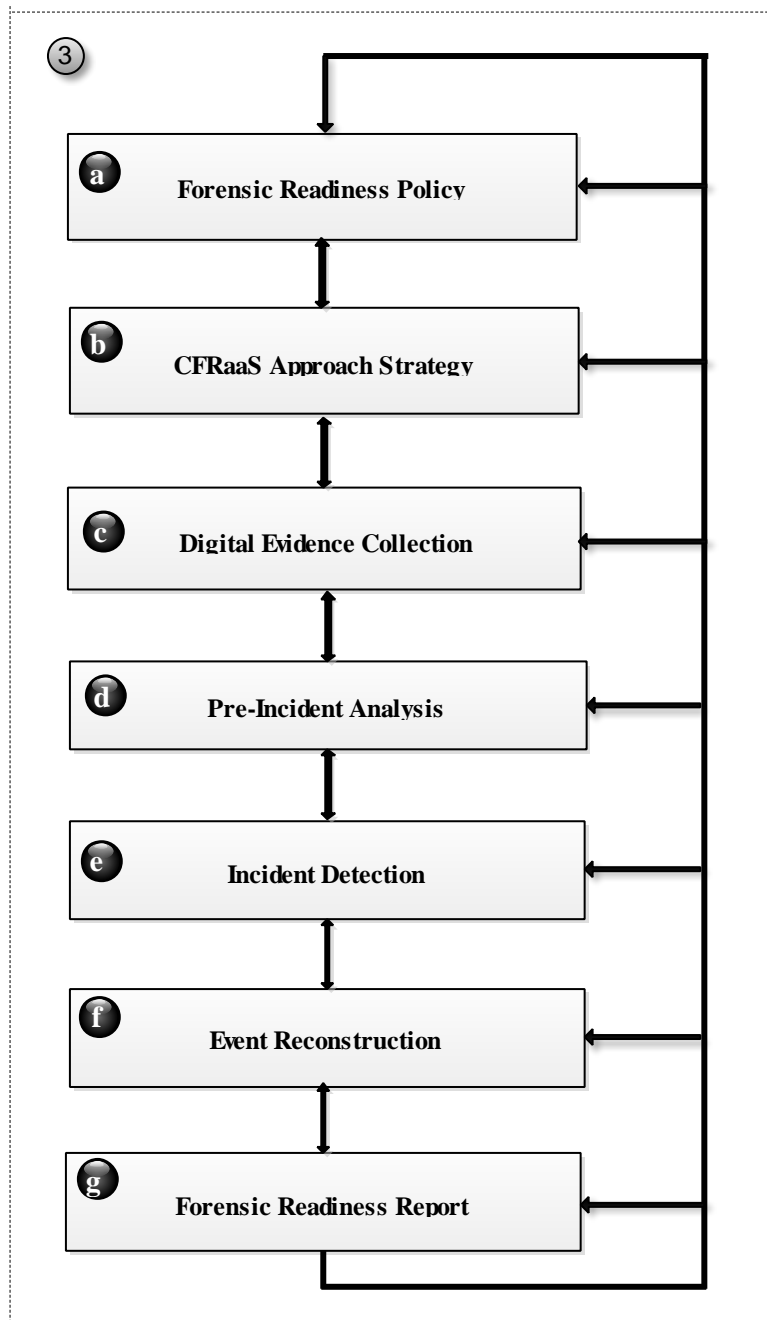


Figure 7.4 Overview of Digital Forensic Readiness Layer

The main objective of employing a *forensic readiness policy* is to ensure that there is a legal premise on how PDE, that is deemed admissible, may be extracted and presented in a legal process. The *forensic readiness policy* process may also involve the practices that are employed in any organisation in order to minimise the cost of conducting a DFI. Consequently Elyas, Ahmad, Maynard and Lonie (2015) puts forward a view that forensic readiness policy may dictate a number of factors like technology to be used, standards and also the impacts a system architecture will have in an organisation.

The next process illustrated in Figure 7.4 is the *CFRaaS Approach Strategy* (labelled **b**). The *CFRaaS Approach Strategy* is a DFR mechanism that is used to define the processes to be used by the CSPs while collecting digital evidence that may be accepted as suitable PDE for litigation. Furthermore, the *CFRaaS Approach Strategy* may be used in preparing the cloud environment for digital investigations.

The process labelled **c** represents the *digital evidence collection* mechanism that is used to collect PDE that can be admitted in a court of law. This involves employing a forensic agent – in this context, the researcher employed a botnet with modified functionalities, i.e. a non-malicious botnet – where a bot client is deployed to legitimately perform an infection.

The process labelled **d** is *pre-incident analysis*, and it represents an approach whereby potential digital evidence is reviewed in detail if a security incident is detected. Reviewing of evidence enables one to be able to increase chances of incident detection.

The process that is labelled **e** is known as *incident detection*, and it mainly involves the identification of security incidents from the collected PDE. *Incident detection* is concerned with instituting guidelines that allow the detection of security incidents through notification procedures. Moreover, it involves classifying and describing a security incident that is capable of triggering the DFI process.

Next, follow the *event reconstruction (ER)* and *forensic readiness report* in the processes labelled **f** and **g** respectively. *Event reconstruction* is a mechanism that is used to trace events that have previously occurred in order to be examined. Digital evidence is usually examined to determine the significance of evidence characteristics. Additionally, ER ensures that PDE is presented in a human understandable manner in order to enable LEAs, forensic experts and

DF investigators to analyse previously experienced events at digital crime scenes. On the other hand, the *forensic readiness report* (labelled **g**) is an interpretation of forensic readiness results to assist forensic investigators and LEAs to commence and interpret what could have happened. It should reveal the proactive details of a potential incident. It may also portray recommendations, opinions and conclusions that can be interpreted by investigators.

Each of the processes labelled **a-g** and highlighted in Figure 7.4 will be explained in the subsequent subsections. First is a discussion of a *forensic readiness policy*.

7.5.3.1 Forensic Readiness Policy

A *Forensic Readiness Policy* (labelled **a** in Figure 7.4) governs different procedures for collecting, storing and handling PDE, and outlines the standardised methods of conducting DFR. It also provides a channel that serves as a legal basis for collecting PDE that can satisfy the requirements for admissibility in a court of law when such evidence is required for a legal process. Furthermore, the broad effect of this policy is to reflect different procedures of maximising the use of PDE when needed. This also helps to minimise the cost and adverse security effects of potential incidents in an organisation. On the same note, a *forensic readiness policy* helps to safeguard the interests of an organisation when conducting a DFI and determines what type of information security should be handled (Hone and Eloff, 2002). The *forensic readiness policy* also outlines an organisation's capability to conduct proactive forensic monitoring through the collection of admissible digital evidence, as well as the examination and presentation of such evidence for legal purposes (see Figure 7.5).

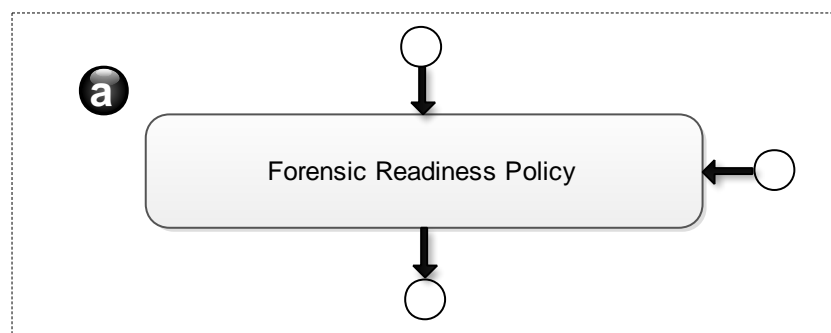


Figure 7.5. Forensic Readiness Policy

Note that each process previously shown in Figure 7.4 is further subdivided into sub-processes and this has been shown in subsequent figures. The *CFRaaS Approach Strategy* is discussed next.

7.5.3.2 CFRaaS Approach Strategy

To begin with, the process labelled *b*, which is presented as *CFRaaS Approach Strategy* in Figure 7.4, consists of the following sub-processes: *Planning and Preparation*, *Scenario Identification* and the *Non-Malicious Botnet (NMB) deployment* as a forensic agent. The researcher opted to call this process *CFRaaS Approach Strategy* because it is used throughout this research thesis to demonstrate the impact of strategically making the cloud ready for Digital Forensic Investigations (DFI). The *CFRaaS Approach Strategy* provides an ideal basis for defining activities that deal with the pre-incident collection of potential evidence in a manner that will make a particular cloud environment forensically ready for digital investigations. Moreover, this approach identifies the risks that might identify vulnerabilities and threats, scenarios and techniques of gathering PDE in the cloud environment – as is shown in Figure 7.6.

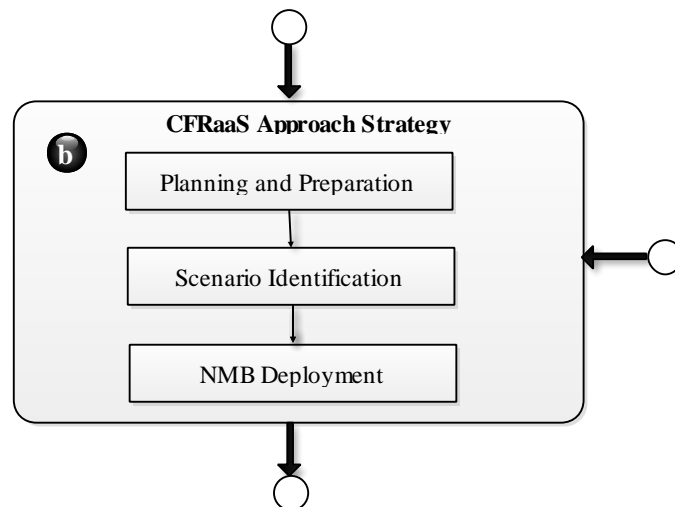


Figure 7.6 CFRaaS Approach Strategy

Each of the sub process contained in *CFRaaS Approach Strategy* has been explained in the subsections to follow.

7.5.3.2.1 Planning and Preparation

Planning and Preparation is a collection of processes that are concerned with the day-to-day operations in an organisation and that are able to formulate the procedures that an organisation will follow if the unexpected is experienced. The main focus of *Planning and Preparation* is to plan and prepare for major risks (such as security incidents) that an

organisation is likely to experience. In addition, organisational functions are protected in such a manner that, should a security incident occur, the necessary requirements will be in place to aid investigators and cause minimal disruption of business processes.

Nonetheless, it is essential to have a comprehensive plan that shows how activities related to forensic readiness are to be handled. This plan should cover all aspects dealing with *Planning and Preparation* activities. Nevertheless, the organisation's critical functions, security events, the people involved, the technology that is going to be used and the resources that are to be employed are all defined at this phase. Rowlingson (2004) highlights that incident preparedness can be targeted as a corporate goal (i.e. a policy decision) when implementing a forensic readiness approach. An example of the technology that may be employed in *Planning and Preparing* includes planning for the implementation of intrusion detection tools. The *Planning and Preparation* process was also defined in the readiness process groups as depicted in the ISO/IEC 27043: 2015.

In the case of the hypothetical case scenario 1 (as discussed in Chapter 6), there was no proper planning strategy in place on how a proactive process could have monitored security events. The information security breach and identity theft that was experienced in BlueBerry Company could have been prevented if Intrusion Detection Systems (IDS) that are able to monitor digital traffic had been in place. Another problem concerns the discrepancy between the number of affected cards that BlueBerry reported to investigators (2000), whereas the affected banks and financial institutions put forward a different figure of 4500. This discrepancy was highlighted between the CSP and BlueBerry. In scenario 1, Nax.com had not enforced proper *Planning and Preparation* procedures for these particular incidents. Due to a lack of incident preparedness, it would be hard for BlueBerry to recover without disruptions. The post-event response mechanism was likely to be costly because insufficient *Planning and Preparation* had been done before these security events could occur. A post-event response mechanism is a DFI process that is employed to gather the necessary digital evidence when a serious or potential security incident has been detected (Rowlingson, 2004).

Having looked at planning and preparation sub-processes, the next section gives a discussion on scenario identification.

7.5.3.2.2 Scenario Identification

The next phase in Figure 7.6 is *Scenario Identification*, which involves an assessment of potential risks. This assessment enables the identification of environments, threats and vulnerabilities that can be introduced as prerequisites for achieving DFR within a given organisation. It is worth noting that, at any given time, different organisations will experience different threats, vulnerabilities and incidents. Additionally, it might become significantly difficult to respond to these threats, vulnerabilities, attacks and incidents due to their nature of sophistication in the absence of DFR.

Again with respect to hypothetical case scenario I, the following could have been identified as the probable risks that led to a security breach: unauthorised login, unprotected Wi-Fi and un-encrypted client information in the central database. It was essential for BlueBerry to ensure that strategies were in place that could ensure the timely identification of risks. This is particularly important because the earlier risks were identified, the sooner plans could be put in place to mitigate them.

The researcher also identified probable risks in hypothetical case scenario II, namely intrusion, information theft, information tampering and framing. These actions allowed disgruntled employee *W* to use the cloud as a tool of crime in company *PQR*. *W* was able to plant malware in the form of a rootkit and frame the security administrator up to a point that resulted in the latter being unfairly arrested. Based on this case scenario, the following were identified as the probable risk indicators that led to a security breach: lack of forensic monitoring tools strategically positioned for the cloud; lack of malware filters or detectors; unprotected file system; unrestricted access to files. In this case scenario, company *PQR* could have implemented a number of procedures that would have helped to minimise the risk of a malware attack. Even after the attack, company *PQR* still did not have sufficient forensic evidence that could be used by investigators to identify the perpetrator.

In hypothetical case scenario III (dealing with sexual harassment, child pornography and framing using the cloud), the risks that could have been identified and mitigated include employing a forensic monitoring device; identity deception; unprotected router ID; lack of monitoring; unavailability of Netflow logs. In this case, the actual risks were never identified and as a result, there were dire consequences for an innocent couple, which allowed *P* to easily frame *X* and *Y* and causing them to end up spending time in jail.

Due to the risks highlighted in the hypothetical case scenarios, Equation 7.15 is the risk equation that highlights an instance where threats are likely to correspond to threat levels and the cost that is incurred by a given organisation (Risk, 2016). The main goal in this case is to set procedures for identifying, evaluating and mitigating risks that may arise in a business environment. Therefore, in the context of this research thesis, the existence of potential risks can be defined as shown in Equation 7.15 (Risk, 2016):

$$\text{Potential_Risk} = \text{Org [P_Threats]} \times \text{Org [P_Vulnerabilities]} \times \text{Org [Cost]} \quad (7.15)$$

- **Org [P_Threats]:** This factor is represented as the frequency at which adversarial events are likely to be experienced on an organisation. Even though the threats do not have a detrimental effect, the likelihood that they may occur in an organisation makes them a measurable entity. Moreover, an organisation should be able to estimate or measure the rate of occurrence of threats that are likely to be experienced. It is worth noting that since the majority of threats constantly change, the researcher's focus is on the threats that are accelerated by humans. In all the hypothetical case scenarios, threats existed because major risks were evident; however, these threats were not taken into account.
- **Org [P_Vulnerabilities]:** This factor is used to denote the likelihood that a given threat may happen in an organisation. Normally this is seen as a weakness that an attacker might use to jeopardise the normal running of the system – provided that a system is susceptible to attacks and the attacker is able to use this susceptibility to his advantage. A lack of forensic monitoring in all the hypothetical case scenarios meant that the systems remained vulnerable and in due course they were exploited.
- **Org [Cost]:** In this context, the effects that may be experienced as a result of a threat are related to the cost incurred by an organisation based on the potential attack. For example, the cost of dealing with an incident was the biggest factor that the researcher considered. In all of the case scenarios, conducting digital investigations was going to cost the organisation dearly, due to lack of incident preparedness or DFR. If all the organisations could have discovered that the risks mentioned were not going to pose

any threat, or if they discovered that they were not vulnerable to these risks, then the *Org [Cost]* was going to be zero. It is also worth to note that *Org [Cost]* includes the cost of doing analysis, assessment or risks.

Having looked at *Scenario Identification*, in the next section the researcher gives a discussion on *NMB Deployment process*.

7.5.3.2.3 NMB Deployment

The last process of the *CFRaaS Approach Strategy* is the *NMB Deployment* process. The process is conducted by the CSPs for purposes of forensically monitoring the client's activities (Ac_i as shown in Equation 7.16 below) within a cloud environment. In this process, botnets, which had originally been used for malicious purposes, were modified for non-malicious purposes, i.e. that act as forensic agents. These non-malicious agents or bot clients were deployed in the cloud environment to forensically collect digital information that may be used as admissible evidence in a court of law.

The *NMB Deployment* process allows the gathering of digital forensic data in order to keep track of a list of actions in the cloud environment. This is mainly done for digital forensic readiness purposes, in preparation of a digital investigation if a security incident has been detected. This approach is based on a relationship between the CSPs' Service Level Agreements (SLAs), existing constitutional and statutory provisions, legal considerations and compliance, and on forensic monitoring in a given jurisdiction. Firstly, a domain that represents a CSP is defined. Thereafter, the CSP gives services to clients that are represented as a set of activities Ac_1 to Ac_n (see Equation 7.16). Also refer to Equations 7.1 and 7.2 respectively where the CSP and the Clients (*Cl_s*) were also defined as entities of the Cloud Model (*CM*).

$$CSP = \{ \{ Cl_i \} = \{ Ac_1, Ac_2, \dots, Ac_n \} \} \quad (7.16)$$

where CSP represents the cloud provider and Ac represents the monitored clients' activities in the cloud environment. Based on this formulation, a more detailed explanation of how digital evidence collected is presented next.

7.5.3.3 Digital Evidence Collection

The process labelled *c* in Figure 7.7 named *Digital Evidence Collection* consists of the following sub-processes: *Digital Evidence Collection Requirements* (labelled 1); *Bot client infection* (labelled 2); *Digital Evidence Capture* (labelled 2); *Digital Preservation* (labelled 3) of forensically collected evidence, Storage in *Forensic Database* (labelled 4) as payload and hash data. Digital evidence that can be used to develop a hypothesis in a court of law is collected in this phase. According to Rowlingson (2004), the techniques for gathering digital evidence for DFR purposes should be done through Intrusion Detection Systems (IDS) so as to target major security incidents.

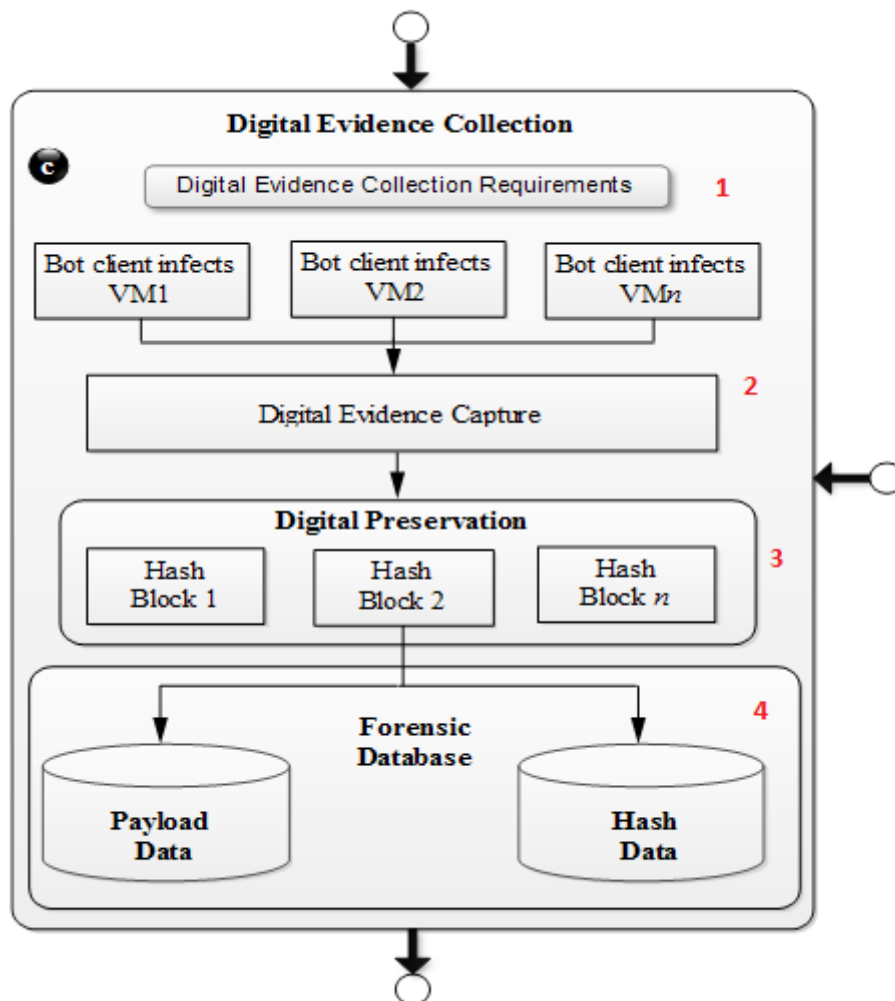


Figure 7.7 Digital Evidence Collection

Each of the sub process that has been represented in Figure 7.7 has been explained in subsections 7.5.3.3.1 to 7.5.3.3.2.

7.5.3.3.1 *Digital Evidence Collection Requirements (1)*

Meeting the *Requirements* of the *Digital Evidence Collection* process is a vital step while collecting PDE from the cloud environment. It is employed as a strategy that ensures digital evidence is collected based on the Standard Operating Procedures (SOPs). The process should ensure that the following requirements are satisfied while or before collecting PDE:

- That the integrity of the forensic logs is maintained and proved at the same time in case there is evidence manipulation or tampering.
- That digital forensic investigators are able to check if the integrity of the logs is violated at a given time.
- That, based on the legal requirements of a given jurisdiction, an individual's privacy is preserved while collecting evidence.
- That hashes are created for all the forensic logs that are collected.
- That all the SOPs and the SLAs regarding evidence collections are adhered to prior to evidence collection.
- That a strategy to secure the relevant collected forensic logs is put in place.
- That consideration is made for the legal requirements and considerations while collecting digital evidence based on territories and jurisdictions.
- That, owing to the constantly changing cloud environment, the collection and retention of the forensic logs should be continuous.
- That efforts are made to determine what forensic log is relevant during evidence seizure.

Monitoring of potential evidence sources occurs in the cloud environment as shown in Figure 7.7. At this stage, a technique of retaining digital evidence through gathering, preserving and storing is defined.

7.5.3.3.2 *Bot Client Infection and Digital Evidence Capture (2)*

Figure 7.7 shows forensic (bot) clients that basically contain the execution of the NMB processes in the VMs. The underlying assumption in this model is that there are n VMs executing in the cloud environment, through which a modified form of a botnet that consists of forensic agents named bot client is able to administer the collection of PDE. By means of this process, sensitive and critical information can be gathered from the cloud in a forensic logging approach from n number of VMs (see Figure 7.7).

The captured potential evidence that is in the form of logs is digitally preserved and then

stored in the forensic database as payload and hash data. Hashing is performed to potential evidence as a block of hashes (*block 1, block 2.... block n*) in order to maintain the integrity of the data. In this context, hashes are created from the forensically captured logs from a particular VM.

7.5.3.3.3 *Digital Preservation and Forensic Database (3&4)*

After the collection of the forensic logs, the process of hashing is applied to the forensic logs to maintain integrity of collected logs. Hash values are generated for each block of forensically collected log. The hashed block of the forensic logs can be used by digital forensic experts to check the integrity of the data and determine if the potential evidence was tampered with or not during the collection process.

The purpose of *digital preservation* is to ensure changes are not made to the gathered potential evidence. PDE that is to be used for DFR purposes has to be retained in its original form without contamination or modification. This is to ensure that the collected digital evidence satisfies admissibility, as it may eventually be used to reach formal conclusions in a court of law or during court proceedings.

7.5.3.4 Discussion based on hypothetical case scenarios

Reflecting on hypothetical case scenario I on Information Security Breach and Identity Theft (see Chapter 6 of this thesis) the following is evident: when BlueBerry initially suspected a possible break into its systems, a decision was made to keep the incidents from the public, based on the company's own interests. Even though there existed no legal provision that necessitated BlueBerry to report the incident, it was evident that the CSP, Nax.com, had not taken effective security measures for *Planning and Preparation* before *Incident Detection*. This complacency was, in one or more ways, going to cost BlueBerry a lot of money when a digital investigation was to be conducted.

Similar events have also been seen in hypothetical case scenario II, where a computer security administrator for company PQR was arrested for a crime of intrusion, information theft, information tampering and framing. In this case, there was insufficient forensic evidence that could prevent the security administrator from being arrested or help him to be exonerated.

Likewise, the cloud service provider S_NET.com did not have in place sufficient proactive forensics that could enable the prosecutors to compile a case against a morally corrupt paedophile P in RCY neighbourhood (see hypothetical case scenario III on Sexual Harassment, Child Pornography and Framing). This led to the wrongful prosecution of the victim's parents, while the actual perpetrator walked free.

In each of the hypothetical case scenarios, there is a higher impact emanating from the lack of a sufficient proactive process that could help prove a fact in a court of law. If the CSPs had been forensically ready, these vulnerabilities and breaches that allowed intrusions, framing and theft could well have been noticed. The latter would have been possible if the CSPs had performed forensic logging, which according to Rowlingson (2004) could point to the following:

- Digital forensic evidence capture could provide evidence to be gathered for acting in the company's defence if there was a lawsuit.
- Digital forensic evidence capture could be used as a digital evidence collection method that would deter insider threats. Lack of this approach could help to cover a cyber-criminal's tracks.
- Digital forensic evidence capture could have reduced the cost and time that could have been required to conduct an investigation.

In this context, forensic logs could have been used as potential digital evidence in a forensic readiness approach. Arguably, it is evident that if organisations have an understanding on a forensic readiness policy, how evidence is acquired from evidence sources, and how to perform cost-effective digital forensic evidence capture, then they may be eager to implement forensic readiness and this may lead to pre-incident detection and proper post-incident process. This can only be achieved if digital forensic evidence capture is done based on the readiness processes that have been highlighted in the ISO/IEC 27043: 2015.

The Forensic logs (*FLs*) that exist as PDE are collected based on the timeline of cloud-based activities Ac_i . It is essential to perform analysis on *FLs* that exist as digital evidence when they are collected from different cloud sources before a DFI is conducted. *FL* files can be represented with a tag name (t_n) and the number of times (To_i) the logging activity occurs. A tag name (t_n) is an identifier that is used to label the identity of a given *FL*. This can be represented using Equation 7.17.

$$Ac_i = \{(Fl_{t1}, To_1), (Fl_{t2}, To_2) \dots (Fl_m, To_j)\} \quad (7.17)$$

where Fl_m is used to denote the identifier of the Fl or the tag name and To_i denotes the number of times that Fl_m occurs – which can be represented as $To_i (Fl_m)$. This is the actual occurrence (x) of a particular Fl file. It implies that in each particular activity that (Ac_i) generates, one or more Fl file can be used as PDE. With reference to hypothetical case scenario I (Security Breach and Identity Theft), illegal debit card transactions and exhaustion of credit card limits are some of the activities Ac_i that were experienced in company BlueBerry. In order to compile a digital forensic case, PDE could have been used to prove a fact in a court of law. PDE must exist in the form of forensic logs Fl_{ti} which should also show the timestamps (TP) when Ac_i occurs.

Similarly, a number of adversarial activities are experienced in hypothetical case scenarios II and III respectively. The moment W is able to intrude and plant malware in hypothetical case scenario II, an activity Ac_i is said to have occurred. In a DFR approach, a timestamp TP would have been generated. There is also an adversarial activity Ac_i when P manages to crack A and Y's WEP encryption in order to extract the router_ID in hypothetical case scenario III. The TP for this activity would have been generated if DFR had been enforced by *TB.com*. The occurrence of Ac_i , Fl and TP are represented by Equation 7.18.

$$Ac_i = \{(Fl_{TPi1}, Fl_{TPi2} \dots Fl_{TPip})\}, i \in [1, p] \quad (7.18)$$

where n is the maximum number that is continuously taken by i . Each time PDE is collected from a Cl from the cloud environment, i represents the origin or the source from which an Fl is extracted. Additionally, a series of Possible Security Events (PSEs) that exhibit different attributes (at) may be registered as an Fl . The at in this context, represents the time of occurrence, frequency of occurrence, size of the forensic log, source, and destination of the Fl . Security event logs for all the parties that were compromised in the hypothetical case scenarios could have been recorded in a DFR approach through effective digital forensic evidence capture, as was highlighted in Figure 7.6. This is an important factor that could have enabled digital forensic investigators (DFIs) to locate the potential source of DFE. It is quite obvious that the presence of these Fls could have made it hard for the perpetrators in all the

hypothetical case scenarios to cover their tracks. The source could have been traced by using either using the *IP address*, *timestamps*, and the *possible location* of the perpetrator(s). The presence of *at* could have enabled the identification of the attacker's patterns in a forensic readiness state, in preparation of an incident response. Thus, the existence of these possible security events may be represented by Equation 7.19:

$$Fl_{ij} = \{e_{ij1}, e_{ij2}, \dots, e_{ijm}\}, i \in [1, n], j \in [1, p] \quad (7.19)$$

where, (e_{ij}) represents Possible Security Event (PSE) that has a probability of occurring given some Ac_i , m is the maximum number of time that e_{ij} occurs, while j takes some integer values between $[1, p]$ continuously where p is the maximum value. The *at* for an event e_{ij} may be represented by varying records that may include timestamp (*TP*), occurrence (x) and size (s) of the forensic log (*Fl*). An event e_{ij} has been used as an information security incident or a threat that has the intent of jeopardising or compromising the normal operation of a system. This can only happen through detecting and exploiting vulnerabilities.

It is also apparent that the employee's information was not secured by company BlueBerry in hypothetical case scenario I. The intruders took advantage of this vulnerability by gaining access to confidential information in company PQR. This situation is similar to hypothetical case scenario II, where W was able to defeat the main control of the system by planting malware that compromised the operations of the system. An even more serious event was when the perpetrator covered his own tracks, which led to a wrongful conviction. Exploited vulnerabilities in the WEP encryption in hypothetical case scenario III also led to online child pornography, sexual harassment and framing.

Based on the aforementioned vulnerabilities, perpetrators are able to syphon confidential information. Notwithstanding the syphoning of information that occurred in the aforementioned case scenarios, the events e_{ij} and timestamp *TP* of occurrence x are hardly detected due to the absence of *Fl* that could prepare for DFR. The sequences of occurrence of these events are represented by Equation 7.20:

$$e_{ij} = \{at_{ij1}, at_{ij2}, \dots, at_{ijk}\}, i \in [1, n], j \in [1, p], k \in [1, m] \quad (7.20)$$

Based on the formulations that were presented, the components of the CFRaaS model have been formalised and therefore, the equation representing the CFRaaS model process 2 (*CFRaaS Approach Strategy*) and process 3 (*digital evidence collection*) illustrated in Figure 7.1 are represented as follows:

$$CFRaaS = \{CSP = \{ \{Cl_i\} = Ac_i \{Fl_t \{e_{ij} \{at_{ij}\}\}\} \} \Leftrightarrow dp \} \quad (7.21)$$

where, Ac_i represents the set of monitored activities, Fl_t represents forensic log with an identifier or tag name, e_{ij} represents a possible security event, at represents the attributes, while dp is used to represent the process of digital preservation.

Having looked at digital evidence collection, it is important to perform a pre-analysis because it will determine if an event, e_{ij} can be used as potential evidence.

7.5.3.5 Pre-Incident Analysis

The process that is labelled **d** in Figure 7.8, namely *Pre-incident Analysis (Pi_A)*, consists of the following sub-processes: *Pre-incident Planning (Pi_P)*, *Incident Description (Pi_Des)*, *PDE Assessment (PDE_A)*, *Relevant PDE (R_PDE)*, *Non-relevant PDE (NR_PDE)* and *Pre-Incident Analysis report*. In this process, a careful examination and assessment of PDE is made for the possible detection of incidents. *Pi_A* is a process that reviews digital evidence in order to ascertain whether it has been compromised or not. This is done with the main aim of conducting a reconstruction of security incidents. In this phase, examination and assessment of PDE is conducted with a view to the possible detection of security incidents. The output of this process should highlight the security incidents experienced as a result of the *incident detection* process (Note that this was previously shown process 4 of Figure 7.4). On the other hand, Figure 7.8 gives an overview of *pre-incident analysis* processes of the CFRaaS model.

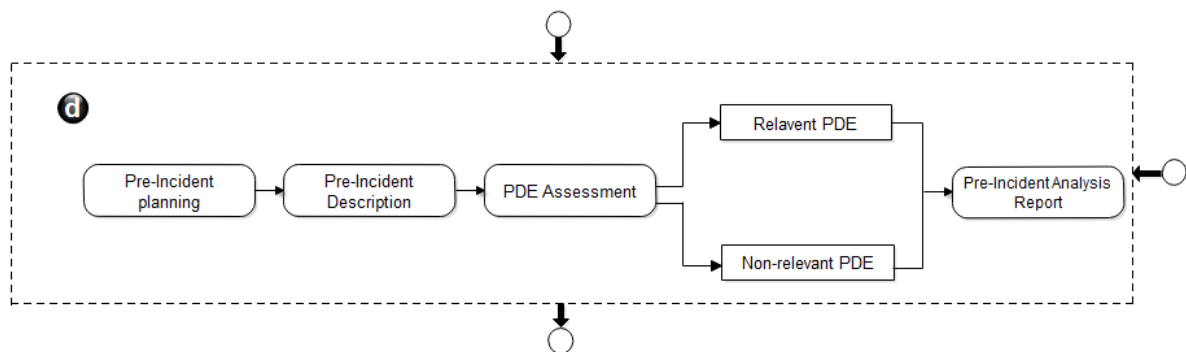


Figure 7.8 Pre-Incident Analysis

Pi_P defines activities that should be performed on the collected, digitally preserved and stored PDE by enabling a given organisation to manage security incidents in a manner that allows effective and fast incident response measures.

Pi_Des in this case provides information on the type or nature of the incident that is detected as highlighted in the previous section. PDE_A is done to check if PDE is relevant or not. PDE is said to be relevant if it has a potential of containing suspicious activities; otherwise it is non-relevant. Details on how R_PDE is distinguished from NR_PDE are explained in later sections of this research thesis. A final process is a Pi_A report that is able to reach a conclusion based on the relationship that exists between the collected PDE and possible events that may arise as a result of PDE seizure.

Once all the companies in all the hypothetical case scenarios (see Chapter 6) realised that they had been compromised in some way, they knew that the only way the perpetrators could be apprehended would be through conducting a DFI. As a post-response process, this would be a very costly exercise due to the companies' forensic unpreparedness. Implementation of DFR could have ensured a preliminary awareness and proper understanding of the need to conduct an extensive PDE_A , based on the highlighted descriptions of incidents. This proposition would allow an effective measure of having to separate R_PDE that may be used to conduct an investigation from NR_PDE . This is ultimately presented in a Pi_A report (see Figure 7.8).

The following presents a formulated equation for process 2 (*CFRaaS Approach Strategy*) and process 3 (*digital evidence collection*) in Figure 7.4 of the CFRaaS model as follows:

$$CFRaaS = \{CSP = \{\{Cl_i\} = Ac_i\{Fl_{ii}\{e_{ij}\{at_{ij}\}\}\}\} \Leftrightarrow dp\}$$

where (dp) is the collected and digitally-preserved PDE. Based on the formulation, dp is represented as a target while Pi_P , Pi_Des , PDE_A and Pi_A are presented as activities (Ac_i) that are performed on the object dp . In this context, the object qualifies to be a digital file, a

forensic log, software or hardware. If DP is a set of objects, then $dp \in DP$, where DP is given by the following:

$$DP = \{dp_1, dp_2, dp_3, \dots, dp_i\}, i \in N \quad (7.22)$$

Nevertheless, $dp \in DP$ is considered to have a number of properties. These properties can be represented using the following:

$$DP = \{p \in P_o | dp \alpha_o p\} \quad (7.23)$$

where p_o represents the properties that give a description of the target from which digital evidence is extracted. p_o may include forensic log (Fl) name, timestamp (TP) of the Fl , the size (s) of the Fl and the overall log file metadata. α_o provides a relationship that is able to merge the target $dp \in DP$ to the property $p \in P_o$. The activities Pi_P , Pi_Des , PDE_A and Pi_A are operations that have to be executed after the collected digital information from the cloud environment has been stored as PDE. If $\{Pi_P \cup Pi_Des \cup PDE_A \cup Pi_A\} = Ac_i$ = the set of activities, then $at \in Ac_i$ where;

$$Ac_i = \{at_1, at_2, at_3, \dots, at_j\}, j \in N \quad (7.24)$$

The aforementioned activities (Ac_i) are represented as general operations that occur at the digital locations where the target is stored. Following Figure 7.8, the overall activities (Ac_i) are represented as follows:

$$\{Pi_P \cup Pi_Des \cup PDE_A \cup Pi_A\} = Ac_i \Leftrightarrow \{R_PDE, NR_PDE\} \quad (7.25)$$

where, R_PDE and NR_PDE are represented as relevant PDE and non-relevant PDE respectively, which may contain potential security events e_{ij} , that are represented as activities Ac_i registered within Fl . It is important to note that $\{Pi_P \cup Pi_Des \cup PDE_A \cup Pi_A\}$ represents activities that are conducted during *pre-incident analysis* and these are activities that will end up determining the R_PDE and NR_PDE that digital forensic investigators will concentrate on. Mostly the concentration should be on R_DE , however, NR_PDE is still retained.

It is therefore paramount to know how to pull the plug when security incidents are detected in any business setting. Firstly, the main way of managing a post-event response after an incident has been detected is to ensure proactive measures exist in a given business setting. Secondly, it is essential to detect the exact incident efficiently and to respond to the incident, and thirdly it is equally essential to perform a DFI effectively. Therefore, in the next section, the process of *incident detection* is discussed.

7.5.3.6 Incident Detection

This section discusses the mechanism of detecting incidents which comprises of three sub-processes. The process labelled *e* in Figure 7.4 represents *Incident Detection*, which is made up of the following sub-processes: *Incident Detection Rate (IDR)*, *Incident Classification (IC)* and *Incident Response Mechanism (IRM)*. Thus, *Incident Detection* is a process that may trigger a digital investigation. Incidents can be any potential security threats that have the capability of exploiting vulnerabilities in a system. In this context, *Incident Detection* involves techniques that are used to identify intrusions that may trigger a DFI. Basically, with the availability of DFR, *Incident Detection* enables efficient discovery, notification and reporting of threats and attacks that have the potential to compromise a system. According to ISO/IEC 27043: 2015, *incident detection* is a process that deals with the detection of *potential* incidents. An incident has a possibility of occurring whenever a suspect performs an unlawful action on a stand-alone computer or a networked computer.

Moreover, incidents are normally related to security threats or attacks that pose the risk of jeopardising the business processes of a running organisation. These threats or attacks can be intrusions, a resource breach, subversion, system exploitation or an intruder leveraging unauthorised access. Such subsequent intrusions happen as a result of adversarial breaches and these intrusions end up causing a given organisation huge financial losses if DFR is not enforced.

Coping with the kind of attacks that were experienced in the hypothetical case scenarios I, II and III brings some degree of uncertainty. However, even though the scope of the incidents in the case scenarios differs in some way, very similar damage was caused to the companies BlueBerry, PQR, ABC and DEX. The escalation of these particular incidents was caused by the absence of proper planning and preparation before the incidents could occur.

Nevertheless, the characteristics of the incidents experienced in all the case scenarios show that the primary focus was on intrusion. In each case scenario, the perpetrator was able to gain easy access to confidential information without being detected. In addition, the link between the three scenarios shows that they were cyber-related incidents and all of them had unprecedented impacts on their companies. The presence of forensic readiness could have facilitated timeous and focused investigations from the Computer Emergency Response Teams (CERTs). This would have happened only if the incidents had been reported through a proactive approach.

Once an incident has been reported, information regarding the time and date of occurrence, kind of incident, network, host and how it was detected, should be documented for a proper post-incident response mechanism. Figure 7.9 shows the sub-processes in the incident detection process.

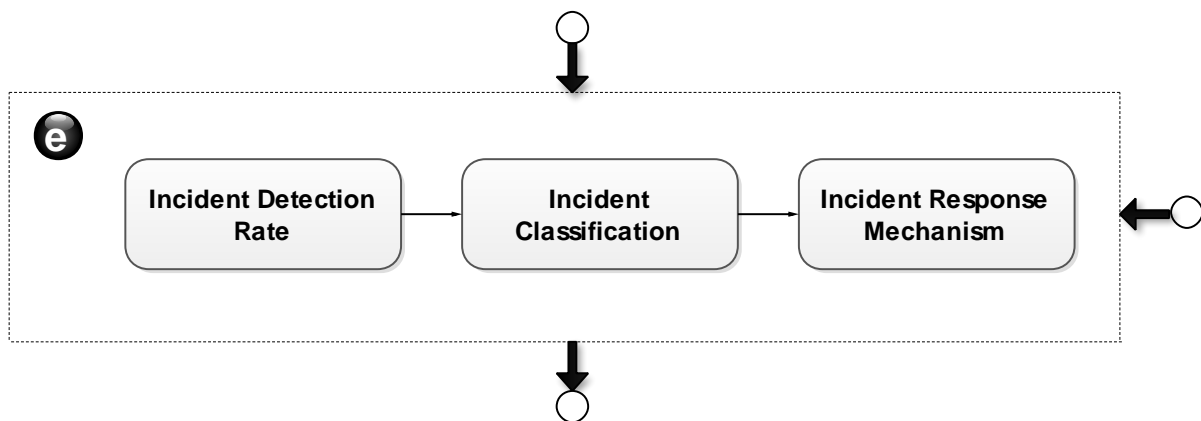


Figure 7.9 Incident Detection

The process elements involved in *Incident Detection* as shown in Figure 7.8 include *Incident Detection Rate (IDR)*, *Incident Classification* and *Incident Response Mechanism (IRM)*. IDR is used to estimate the actual frequency at which security incidents occur. Therefore, in the context of this research thesis, IDR is computed as follows:

$$IDR = \left(\frac{\text{Number_of_Incidents_Detected}}{\text{Number_of_Real_Incidents}} \right) + \{ \text{False_Alarms} \}$$

(7.26)

With regard to the hypothetical case scenario I on information security breach and identity theft, we can compute IDR based on the number of incidents that were experienced. Initially BlueBerry reported that the total number of personal cards affected was 2000, whereas the bank put the total number of affected cards at 4500. Note that the calculation of the real incidents is an estimation that helps digital forensics to approach an investigations. If we agree with the banks and put the number of incidents at 4500, for example, while the real number stood at 3500 and the false alarms at 10, then we can compute IDR follows:

$$IDR = \left(\frac{4500}{3500} \right) + \{10\} \quad (7.27)$$

$$= 11.28 \text{ incidents per given occurrence } (x)$$

x is used to denote a given time when the security incidents occurred, i.e. week, month or year. For example, if we take 11.28 incidents/week, we are able to calculate the growth rate of incidents as a percentage. The growth rate of the security incidents is calculated by assessing the past and the present occurrence of security incidents. If a given rate of 11.28 incidents/week has been detected and the present number of security incidents stands at 15.5 incidents/week, for example, then the growth rate is calculated using the formula shown next in Equation 7.27.

$$Incident_Growth_Rate = \left(\frac{Present_Security_Incidents - Past_Security_Incidents}{Past_Security_Incidents} \right) * 100 \quad (7.28)$$

Therefore IGR is represented as follows:

$$Incident_Growth_Rate = \left(\frac{15.5 - 11.28}{11.28} \right) * 100$$

$$= 37.41\%$$

The *IGR* has been computed based on x that is experienced on a weekly, monthly or an annual basis. The main requirements needed while computing IGR are regular incident occurrence values for time.

Once a potential incident has been detected, it naturally leads to an IRM. However, an incident classification is done first to show the type of incident through a description of the actual incident. The presence of forensic readiness in all the hypothetical case scenarios would have classified the incidents based on the magnitude of their occurrence – either as a potential incident, slight incident and a serious incident. Classification of an incident allows proper identification of the type of incident and based on the incidental information gathered at this level a DFI can be triggered.

A potential incident shows that a threat or an attack is imminent, lest a prevention action is enforced. Normally, the quicker the time that is taken to respond to a potential incident, the smaller chances of that attack being real. Additionally, a slight incident shows that the risk of occurrence of a serious incident may be obvious. In this case, alerting a forensic readiness administrator may help to avoid such an incident.

The importance of having a DFR approach is that it allows the institution to prepare for proper IRM, based on the classification of incidents as shown in the third sub-process of Figure 7.8. A team that comprises of personnel with technical and legal expertise, called a Computer Security Incident Response Team (CSIRT), is able to provide IRM. A CSIRT is a team that considers the basic procedures for responding to a security incident and it is normally responsible for handling incidents. IRM is used to review all information that is related to Potential Security Incidents (PSI), therefore in this research thesis we can compute IRM as follows:

$$IRM = IDR + \{Incident_Description\} \quad (7.29)$$

Where IRM is the incident response mechanism that shows how the post-event response technique is conducted. In the context of this research thesis, the researcher has considered IDR to check the incident detection rate and a description of each incident.

Having explored the incident detection process, the next section contains a discussion on event reconstruction.

7.5.3.7 Event Reconstruction

Event Reconstruction (ER) (labelled *f*) in Figure 7.4, which allows a study of the characteristics of PDE, consists of the following sub-processes: *Retrieval of Relevant PDE*, *Locate Relevant PDE in fields*, *Searching Events*, *Performing Similarity Measure* and *generation of Causality Report*. Consequently, ER analyses and examines PDE in order to identify why it holds certain characteristics. This helps in identifying the causality of the PSEs and also helps to build a hypothesis before a DFI is conducted. The first step when performing an ER procedure is evidence collection and examination, in order to find the causality of an event. Afterwards, the creation and sequencing of event segments must be done based on the evidence (Gardmer & Bevel, 2002).

This involves a discrete collection of digital data to be examined for PDE related to the occurrence of a security incident. ER thus tends to question why digital evidence has certain properties and characteristics, and during ER, these characteristics and properties are analysed and examined to show the exact causes of events.

In addition, while reconstructing digital events in a forensically ready environment, we have to revisit the characteristics and sequence of digital events and check whether the collected PDE satisfies admissibility in a court of law (Kebande & Venter, 2015). For the sake of this research thesis, ER is illustrated as in Figure 7.10 and each of the sub-processes of the figure is explained below.

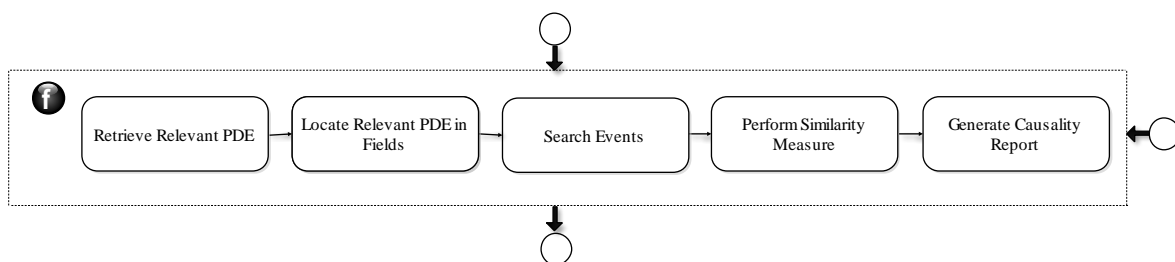


Figure 7.10 Event Reconstruction

7.5.3.7.1 Retrieval of Relevant PDE

Retrieve Relevant PDE entails the act of finding relevant evidence by means of an information retrieval approach, and subsequently taking such PDE as tasks and preparing it

for effective discovery of potential security events. Retrieval of the collected PDE makes it possible for a forensic expert to extract relevant forensic information and filter out what needs to be analysed. This can be achieved by locating R_PDE that is stored in fields in a forensic database.

7.5.3.7.2 Location of Relevant PDE in Fields

Evidence is normally stored in groups or fields, using various patterns. *Location of Relevant PDE in Fields* allows for the identification of real evidence that is organised in such fields and that can be used as admissible evidence. Patterns of evidence that are similar or closely related to each other fall under the same field.

When a DFI is conducted, fields are formed that help to locate PDE and increase chances of detecting events – initially based on their occurrence and later their similarity. In order to arrange the PDE in different fields, the researcher uses the technique of first proposing to locate PDE in the fields based on the categories of different forensically logged data, and then checking for the occurrence of PSEs.

The researcher furthermore considers PDE and the occurrence (x) of different fields A_n of events e_i with timestamp (TP). The PDE is reconstructed based on field name ($Field_N$) and the interval between e_i , which is denoted as the distance (d). Whenever d exists between e_i in a field A_n , it shows that e_i exhibits a difference in TP based on the intervals of x . Based on these variations, an assumption is made that a security incident may have occurred. Nevertheless, the pattern of occurrence of e_i can be computed using the distance function $d(e_i)$ as shown in Equation 7.30 (Kebande & Venter, 2015).

$$\begin{array}{c}
 d(e_i) \\
 \left. \begin{array}{l}
 A_{1(Field_N)} \longrightarrow \{ W_{1TP} \longrightarrow W_{2TP} \longrightarrow W_{3TP} \dots \dots \dots W_{nTP} \} \\
 A_{2(Field_N)} \longrightarrow \{ X_{1TP} \longrightarrow X_{2TP} \longrightarrow X_{3TP} \dots \dots \dots X_{nTP} \} \\
 A_{3(Field_N)} \longrightarrow \{ Y_{1TP} \longrightarrow Y_{2TP} \longrightarrow Y_{3TP} \dots \dots \dots Y_{nTP} \} \\
 \vdots \\
 A_n(Field_N) \longrightarrow \{ Z_{1TP} \longrightarrow Z_{2TP} \longrightarrow Z_{3TP} \dots \dots \dots Z_{nTP} \}
 \end{array} \right\} e_i
 \end{array}
 \tag{7.30}$$

The assumptions that have been used to derive at the approach shown in Equation 7.30 are that a field A_n has a specific field name ($Field_N$). Thereafter, e_i , which is shown on the right-hand side of Equation 7.30 represents the following PSEs (e_i): w_{1TP} , x_{1TP} , y_{1TP} and z_{1TP} , which further represent possible first events for fields A_1 , A_2 , A_3 , A_n respectively. This is followed by w_{2TP} , x_{2TP} , y_{2TP} and z_{2TP} as the second PSE for fields A_1 , A_2 , A_3 and A_n . The third and last events are represented using w_{3TP} , x_{3TP} , y_{3TP} and w_{nTP} , x_{nTP} , y_{nTP} and z_{nTP} for fields A_1 , A_2 , A_3 , A_n respectively. For example, if there are three fields A_1 , A_2 , A_3 that are represented using *IPs*, *Usernames* and *Access logs* respectively, then a digital forensic investigator should be able to arrange the data based on *IPs*, *usernames* and *access logs* as A_1 (*IP*), A_2 (*User_name*) and A_3 (*Access_logs*). This grouping can be used to represent the fields that fall under different categories and thereafter the DFI will realise that there is a need for searching for PSE (Kebande & Venter, 2015).

Based on the similarity between two events that has been highlighted above, the researcher approaches this by considering the fact that whenever there exist two events, for example X and Y, it is possible they may be similar if they occur in the same context. This explains why the distance between the two events X and Y may vary-that is if they occur under different contexts. Based on this concept, security incidents/events may occur under different or similar circumstances-which is the basis of computation on how the similarity of two events may be a factor of consideration, hence the concentration on the distance function (d). This is because, always there will exist a distance between two events, however, d can only be computable under the following circumstances:

$$\begin{aligned}
 d(e_1, e_2) &\geq 0 \\
 d(e_1, e_2) &= 0 \text{ if _ and _ only _ if } (e_1 = e_2) \\
 d(e_1, e_2) &= d(e_2, e_1)
 \end{aligned}$$

This shows that at least $d(e_1, e_2)$ must be computable when checking the similarity of two events X and Y.

The distance function (d) between these events X and Y plays a part in showing they may be similar or not based on the context that they appear. For example, Mannila and Moen (1999) has shown that, if a study is conducted on how a website information is provided to users, it is possible that there might exist similarity if two or more websites are giving users exactly the same information.

7.5.3.7.3 Searching PSEs from Fields

Search Possible Security Events (PSEs) involves performing a look-up by using an Event Search Function (*ESF*) to detect the PSEs that are present within the fields and that contain forensic logs (*Fls*) from the retrieved PDE – these are presented as $A_1(IP)$, $A_2(User_name)$, and $A_3(Access_logs)$ respectively. PDE is assumed to be stored in fields through which e_i or incidents might be detected. Figure 7.11 illustrates how an *ESF* performs a look-up in PDE. S is a search function that returns the fields $A_1, A_2, A_3...A_n$ with field names (*Field_N*). Furthermore, e_i represents the PSE in A_n while $w_1, x_1, y_1...z_n$ represent the PSE in $A_1, A_2, A_3...A_n$ respectively.

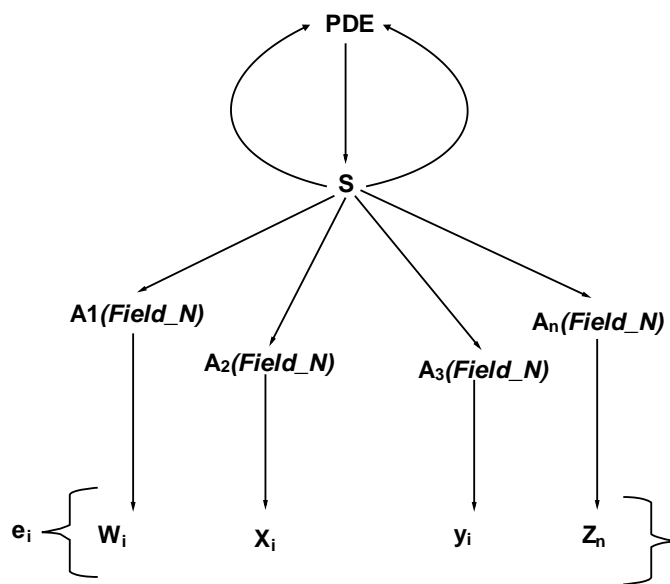


Figure 7.11 Event Search Function

7.5.3.7.3.1 Checking Event Similarity Measure (ESM)

An ESM of the PSEs in a given field is used to check how similar the events are by determining the pattern of occurrence of events. PDE is stored in different fields $\{A_1, A_2, A_3...A_n\}$ as shown in Figure 7.11.

- A revisit of the hypothetical case scenario I on Information Security Breach and Identity Theft shows that when one of the employee tried to withdraw an amount using his debit card, he realised that his account statement showed 10 consecutive withdrawals that he had not made.
- Each withdrawal was done between 12:00 midnight and 12:01 am of the next day within the span of one week.

Considering the events that transpired in case scenario I, it is important for a forensic investigator to try and track the perpetrator's moves by checking the pattern of event occurrence based the frequency of the PSEs and the time intervals at which they occurred. Based on the finding (*consecutive withdrawals* and *withdrawals at 12:00 midnight and 12:01 am the next day*), it is possible to conclude whether the PSEs are originating from one location or one perpetrator, or whether they are happening at the same time.

If $\{A_1, A_2, A_3 \dots A_n\}$ are distinct fields with events e_i (say an *Account withdrawal occurs 48 times after an interval of 10 seconds*), then ESM and the pattern of occurrence (x) of two given events e_i – say w_{1TP} and w_{2TP} – can be computed using the distance function $d^{MD}(w_{1TP}, w_{2TP})$ where w_{1TP} , is the first event (*first withdrawal*) and w_{2TP} is the second event (*second withdrawal*) and $d(e_i)$ is the distance between the events. The distance $d(e_i)$ in this context can be the difference in time between the *first withdrawal* and the *second withdrawal*. Therefore, ESM can be computed using the general distance function equation, which is given by the Minkowski distance metric that is shown in Equation 7.31. The latter is a generalisation of the Euclidean distance and the Hamming distance (De Amorim, 2011), which can be used to calculate the distance between two events in order to show the similarity of those events. The Minkoski distance metric has been chosen in this context because of its accuracy, effectiveness and ability to yield results, even when the chosen data sets are distinct or appear to be separate from each other (Singh, Yadav & Rana, 2013). As a result, ESM is calculated as follows:

$$ESM = d^{MD}(w_{1TP}, w_{2TP}) = \sqrt[p]{\sum_{i=1}^n |w_{1TP} - w_{2TP}|^p} \quad (7.31)$$

where p in Equation 7.31 $= [1, 2 \dots n]$ for the distance function of an event w_{1TP} and w_{2TP} , and $d^{MD}(w_{1TP}, w_{2TP})$ is the distance between first event w_{1TP} and second event w_{2TP} . The ESM, behaviour and pattern of the events e_i can be calculated using the distance function $d^{MD}(w_{1TP}, w_{2TP})$, given p (see Equation 7.31). It is worth noting that the ESM between the events (w_{1TP}, w_{2TP}) should be recorded so as to ascertain the initiator of the event if the process is not continuous.

7.5.3.7.3.2 Testing ESM Using Non-Negative Attributes

Based on the ESM that is mentioned in the previous section, the researcher conducted a comparative analysis of the ESM distance function with the values of p and using the attributes that e_i exhibits. By revisiting Equation 7.30 ($p=1, 2,..n$), we take p as the number of possible occurrences (x) of PSE during ER (see Equation 7.31). On the same note, an assumption is made that the attributes (at) of the given two events w_{1TP} and w_{2TP} are represented as a set. The researcher therefore took random non-negative values for size (s), time (t) and occurrence (x) attributes as w_{1TP} (1, 5, 3) and w_{2TP} (6, 7, 5) (see Table 7.1). Afterwards, the researcher tested the attributes when $(p)=1$, $(p)=2$ and when $(p)>2$ to ∞ , using Equation 7.31.

Different formulas have been used for computation because of the following reasons:

1. To check if there exist variations as a result of the similarity between the events (w_{1TP} , w_{2TP}) that are based on the distance metric $d(e_1, e_2)$.
2. To quantify if there may exist other interrelationships like dissimilarity or correlation on $d(e_1, e_2)$.

Table 7.1 w_{1TP} and w_{2TP} Events With Attributes for ESM Testing

event (e_i)	S	t	x
w_{1TP}	1	5	3
w_{2TP}	6	7	5

When $p=1$ in Equation 7.31, the absolute difference between the pair of event attributes is calculated by examining the absolute value distance. This is given by the following equation:

$$d(w_{ij}) = \sum_{k=1}^n |w_{ik} - w_{jk}| \quad (7.32)$$

When $p=2$ in Equation 7.31, we find the root of square differences between the set of event attributes by examining the distance metric. This is given by the following equation:

$$d(w_{1TP}, w_{2TP}) = \sqrt{\sum_{i=1}^n (w_{1TP} - w_{2TP})^2}$$

$$(7.33)$$

When $p > 2$ to ∞ in Equation 7.31, the maximum value distance is checked by examining the absolute difference in magnitude between the set of event attributes. The distance metric is given by the following equation:

$$d(w_{1TP}, w_{2TP}) = \max_i |w_{1TP} - w_{2TP}| \quad (7.34)$$

7.5.3.7.4 Evaluation of ESM and Finding

Based on the hypothetical case scenarios that were discussed in Chapter 6, the researcher gives an evaluation of ESM. If PDE was collected in all the hypothetical case scenarios for purposes of DFR, then ESM could be applied to the attributes of that PDE. For example, if the PDE consisted of forensic logs that had attributes – IP address, size of log, number of times the log occurred in a given time interval – then ESM based on the aforementioned attributes could have been calculated based on Equations 7.32, 7.33 and 7.34 respectively.

To evaluate the ESM approach, the researcher selected random non-negative numbers to represent the attributes (size, time and occurrence) based on two sets shown in Table 7.1. Thereafter, the researcher evaluated ESM using the proposed Equations 7.32, 7.33 and 7.34 respectively, in order to find the inferences shown in Table 7.2. Furthermore, the selected non-negative attributes were used to represent the PSE, e_i . On the other hand, Figure 7.11 represents ESM experimental findings that are depicted in Table 7.2, which shows the convergence of the distance metric $d(e_i)$ when p is tested with $[1, 2 \text{ and } >2 \text{ to } \infty]$ in Equations 7.32, 7.33 and 7.34 respectively. The computed values of $p=1$, $p=2$ and $p > 2$ to ∞ are shown in Table 7.2, which lists the distance metric based on the occurrences of p that are shown in Equation 7.31.

Table 7.2 w_{1TP} and w_{2TP} Events With Attributes and Values when $p=1, 2 > \text{to } \infty$

No	Event (e_i)	s	t	x	$p=1$	$p=2$	$p=2 > \text{to } \infty$
1	w_{1TP}	1	5	3	8	5.744	5
2	w_{2TP}	6	7	2			

Based on the value of p , the researcher showed the ESM with respect to the distance metric d for the attributes that represent e_i . Figure 7.12 shows the corresponding experimental findings. The notion portrayed in Equation 7.30 is that $d^{MD}(w_{1TP}, w_{2TP})$ represents the

Minkowski distance metric between two e_i , w_{1TP} (*first withdrawal*), and w_{2TP} (*second withdrawal*). Equations 7.32, 7.33 and 7.34 represent the distance metric based on the occurrence of p . From the observations made from Table 7.2, when the value of $p = 1$, and the distance metric for w_{1TP} and w_{2TP} is computed as 8. In contrast, the second value for p has a slightly lower value, and in this case when $p=2$, the distance metric is computed as 5.744. Finally, when $p > 2$ to ∞ the distance metric is computed as 5. The rationale behind the methods used while observing the values for p is to check the variations that were observed as a result of the similarities between events (w_{1TP} , w_{2TP}) that are based on the distance metric between the events.

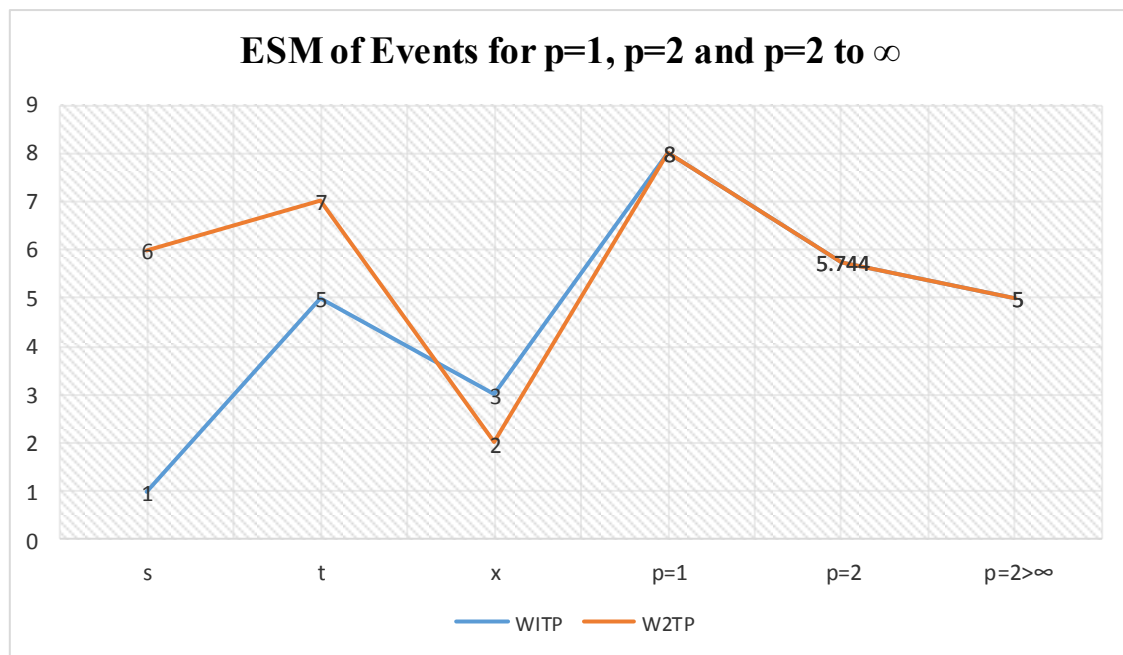


Figure 7.12 ESM of Events for $p=1$, $p=2$ and $p=2$ to ∞

Additionally, as highlighted in Figure 7.12, the random selection of the non-negative numbers that are shown in Table 7.2 suits the prediction that two or more given events (w_{1TP} , w_{2TP}) can possess different characteristics. It is worth noting once again that these variations result from different incidental data that was gathered during the readiness approach. Besides, Figure 7.12 shows that there is a convergence when $p=1$, which depicts a similarity between PSEs. In the researcher's opinion, the outcome of Figure 7.12 on the ESM between w_{1TP} and w_{2TP} shows a close match and convergence when $p=1$; therefore, this outcome is sufficient to prove the concept of ESM in this context.

7.5.3.8 Forensic Readiness Report

Forensic Readiness Report, labelled *g* in Figure 7.4, is an integral part of the DFR process and it typically contains the information and descriptions of all the steps taken towards potential evidence examination, classification and how ER process is formulated. Furthermore, a *forensic readiness report* serves as the ultimate outcome of a forensic examination. It involves two sub-processes namely *Examination notes* and *causality*.

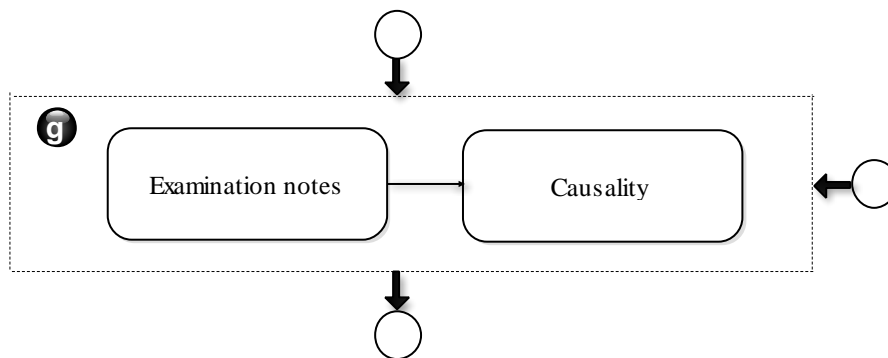


Figure 7.13 Forensic Readiness Report

The sub-process *Examination notes* outlines the examination process and the significant potential evidence that may be collected from the cloud environment. The *examination notes* should be presented in a human-understandable manner and the presentation should be accurate. Moreover, *Examination notes* should lead to conclusions based on the potential evidence that is examined. The ISO/IEC 27043: 2015 describes these conclusions as results from digital evidence interpretation process.

In the context of this research, causality serves as a summary of the activities or evidence that was collected, examined and concluded as being related to digital crimes. Causality involves a process of preliminary investigation that shows the relationship between entities. According to Sowa (2000), causality postulates that the occurrence of a given entity X of a given set may depend on an entity Y of another group. This is simply used to show the link that exists between X and Y and ultimately, based on this relationship, X is considered to be the cause and Y the effect.

Having looked at the DFR layer, focus now shifts to IRP layer that was shown previously shown in Figure 7.1.

7.5.4 Incident Response Procedure Layer

This section contains a detailed discussion on the *Incident Response Procedure* (IRP) (labeled *d*) layer that was labelled 4 in Figure 7.1. Casey (2005) argues that whenever suspicious behaviour culminating in a potential security incident is detected, then IRP becomes necessary. IRP involves the steps that LEAs follow to tackle incidents and how to collaborate with competent bodies. IRP is a reactive process and therefore not part of the DFR functions; it occurs after incident detection, which is the actual process of DFI.

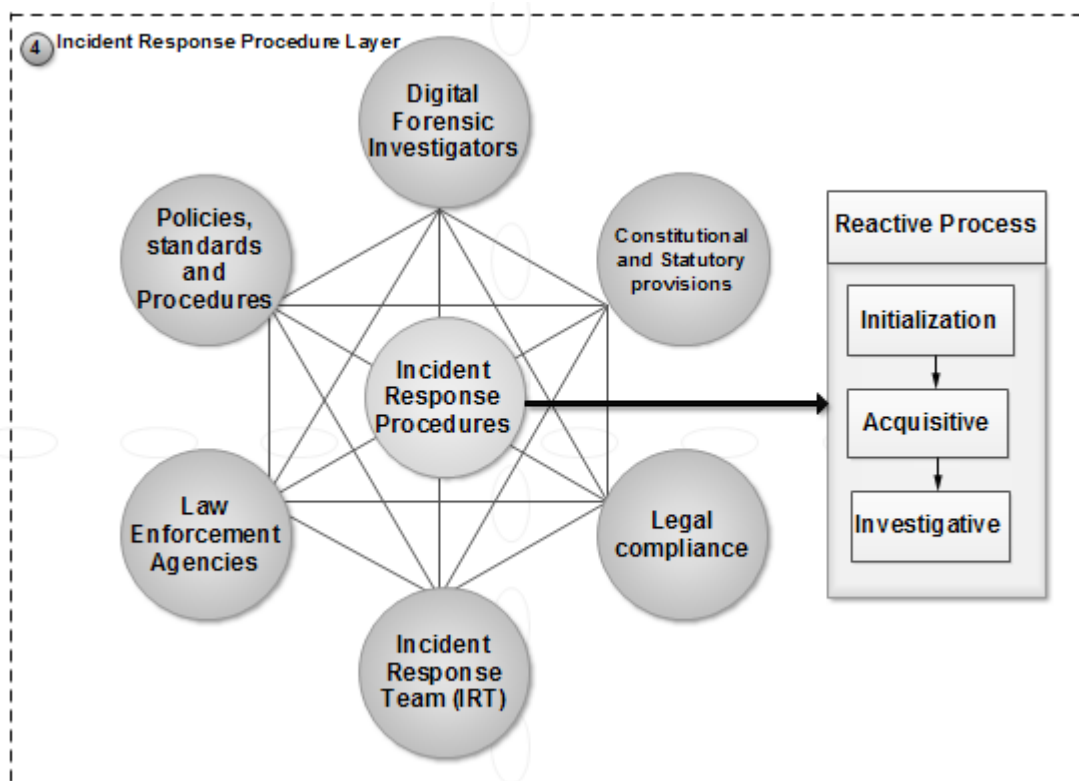


Figure 7.14. IRP Process

In this context, IRP corresponds with and adheres to the guidance of the ISO/IEC 27043: 2015. IRP consists of *Initialisation*, *Acquisitive* and *Investigative* processes, as shown on the right-hand side in Figure 7.14. Valjarevic and Venter (2015) highlight these processes in their comprehensive and harmonised digital forensic investigation process model. Initialisation deals with the inception of the digital investigation process and represents *incident detection*, *first response*, *planning* and preparing for a *post-incident response*. Note that the concurrent

processes discussed later in this chapter are implemented alongside the IRP layer. The acquisitive process *identifies PDE and collects evidence*, and then it follows the process of *acquisition, storage, transportation and preservation of PDE*. Finally, the investigative process performs the following: *PDE examination and analysis, interpretation, reporting, presentation of digital evidence and investigation closure*. IRP relies on the DFR layer to conduct a DF investigation where the requirements as shown in Figure 7.14, namely *policies, standards and procedures, Digital Forensic Investigators, Constitutional and Statutory provisions, Legal compliance, Incident Response Team (IRT) and LEAs* are adhered to.

Policies, standards and procedures provide a legal premise for how the LEAs, IRT and digital forensic investigators should conduct the IRP process in an organisation if a potential security incident is detected. The IRT is a team of senior experienced executives who may be able to contain an incident and thus enable the organisation to recover in case of a security incident. Examples of IRT members include information security experts, senior executives in management, legal counsel and IT auditors. The IRP should comply with the legal aspects and constitutional and statutory provisions applicable to a given jurisdiction, before a DFI begins.

Having studied the IRP layer, the reader is introduced to the concurrent processes in the next section.

7.5.5 Concurrent Processes

The concurrent processes are discussed in this section. The last process (labelled 5) of Figure 7.1 that is shown by an arrow pointing downwards involves the *concurrent processes*, as was mentioned in ISO/IEC 27043: 2015. The concurrent processes are executed alongside the other processes. They provide a proper method of handling digital evidence to enable a holistic approach to DFIs. The main aims of these processes are to ensure the admissibility of digital evidence in a legal system (ISO/IEC 27043: 2015) and to adhere to DF principles.

The tasks involved in concurrent processes (according to ISO/IEC 27043: 2015) include documentation, managing information flow, obtaining authorisation, preserving the chain of custody and digital evidence. Documentation is a mechanism that involves the taking of examination notes based on the outcome of the digital investigation process, while information flow allows the automation of on-going processes.

Next, obtaining authorisation after a security incident has been detected, allows interaction that enables a forensic administrator to perform activities dealing with physical investigations. Chain of custody as a process shows a roadmap of the preservation of digital evidence, in other words how each and every form of evidence is collected even when changes are made. Digital preservation of the collected evidence is done through hashing to maintain the evidence in its original form.

The CFRaaS model was discussed and a more inclusive CFRaaS model was shown in Figure 7.15; however, in the next section the reader is introduced to a comparison between the CFRaaS model and other existing models.

7.6 Comparing the CFRaaS Model with Existing Readiness Models

To check the effectiveness of the proposed model, the researcher compared and mapped the CFRaaS model to other existing forensic readiness models to highlight scientific principles that will contribute to a better understanding of the CFRaaS model. The results of this comparison between the proposed CFRaaS model and different proposed forensic readiness models are presented in a summarised format in Table 7.3.

None of the models used in the comparison – apart from the proposed CFRaaS model – were at the time of writing this research thesis focused on the cloud environment. The proposed model adopted a holistic approach, in order to cover a majority of the processes contained in the other forensic readiness models. Consequently, the CFRaaS model employs the execution of a botnet with modified functionalities as a forensic agent to collect digital evidence that can further be applied in a reactive process. This is one of the novelties of this research, which (according to Table 7.3) has not been explored in any of the existing forensic readiness models as yet.

A number of the existing forensic readiness frameworks shown in Table 7.3 target different environments. For example, Barske et al. (2010) framework targeted forensic readiness for Small to Medium Enterprises (SMEs), Rowlingson (2004) defined forensic readiness focused on the corporate environment, Mouton and Venter (2011) targeted forensic readiness of wireless sensor networks, while Ngobeni and Venter (2012) modelled forensic readiness for

wireless local area networks. Valjarevic and Venter (2012) targeted incident investigation principles and processes, Trenwith and Venter (2013) targeted digital forensic readiness in the cloud. Additionally, Ab Rahman, Glisson, Yang and Choo (2016) have targeted the cloud and lastly Do, Martini and Choo (2016) have targeted mobile cloud in their research. Still, none of the processes that were defined in this model comprehensively covers the entire proposed CFRaaS model.

Therefore, the researcher introduced an *event reconstruction* process in the model for purposes of revisiting the characteristics and properties of accumulated PDE while reconstructing the sequence of events – which is another novelty employed in the CFRaaS model. The *event reconstruction* process, together with the rest of the processes, constitutes a holistic and effective approach to the process of DFR in the cloud environment. In the proposed model, different measures, such as the ESM, were used to check inconsistencies in potential evidence. In addition, the researcher introduced *concurrent processes*, which previously were employed in ISO/IEC 27043: 2015 to allow the CFRaaS model processes to be executed continuously so as to increase the admissibility of digital evidence.

5) Concurrent processes

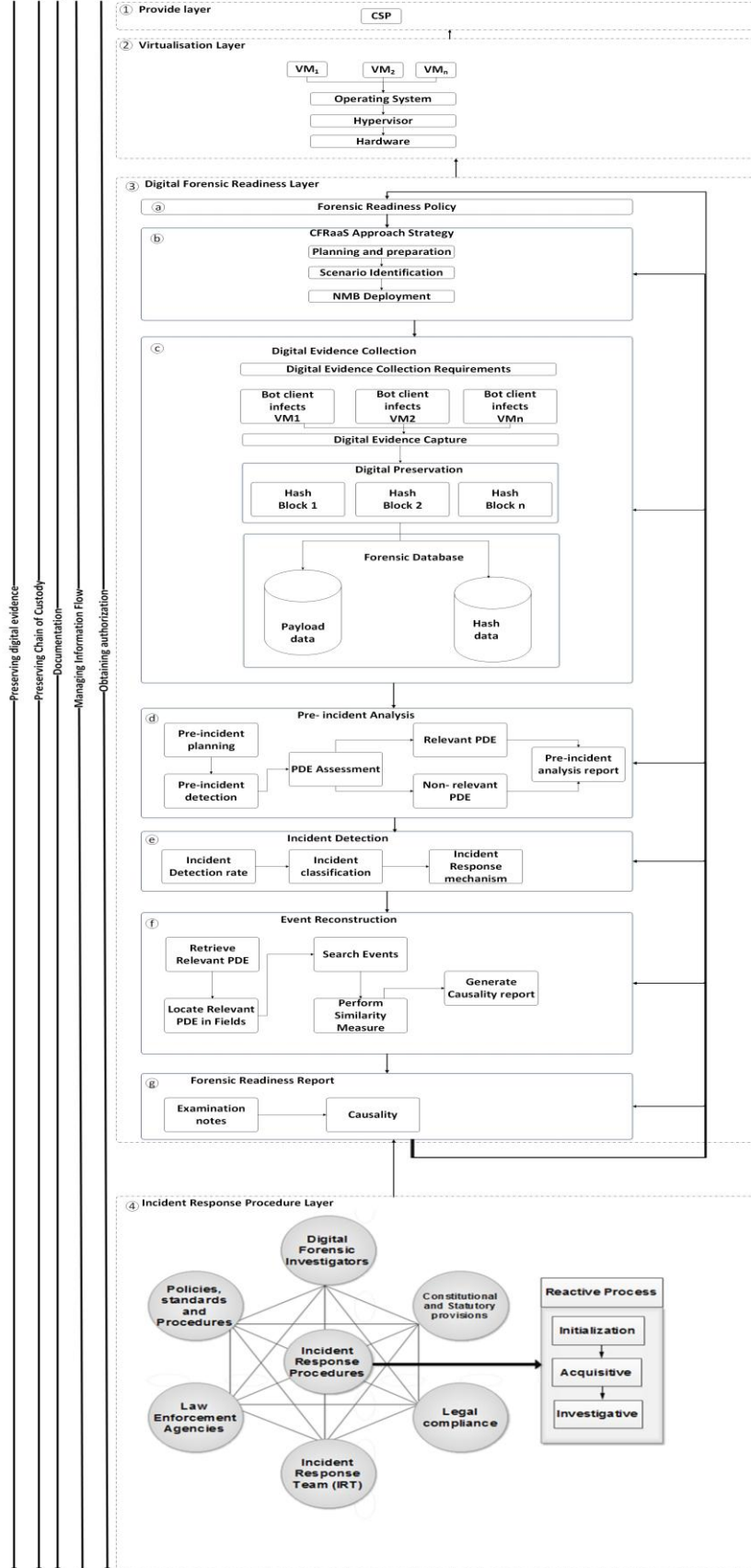


Figure 7.15 Block Diagram of the Detailed CFraaS Model

Table 7.3 Comparing the Proposed CFRaaS Model with Existing Forensic Readiness Models

	Proposed CFRaaS Model	Carrier & Spafford (2004)	Barske et al. (2010)	Tan (2001)	Pooe (2012)	Rowlingson (2004)	Mouton & Venter (2011)	ISO/IEC 27043:2015	Ngobeni & Venter (2012)	Valjarevic & Venter (2012)	Trenwith & Venter (2013)	Ab Rahman et al.(2016)	Do,Martini & Choo(2015)
Forensic Readiness Processes													
Target and processes	Cloud		SMEs			Organisation	Wireless Sensor Networks	Incident investigation principles and processes	Wireless LAN	Incident investigation principles and processes	Cloud	Cloud	Mobile Cloud
1	Planning and Preparation	Survey for digital evidence		How logging is done		Forensic readiness planning		Planning and preparing		Planning & preparation		Planning	
2	Scenario identification		Determining scenarios that potentially require digital evidence	What should be logged		Defining business scenarios that require evidence		Scenario definition			Identification	Evidence sources & Risk Management	Identification
3	Non-malicious Botnet execution												Injection
4	Digital evidence capture	Operation readiness	Maximising the value of logs as potential digital evidence	Determining how logging is done		Collection of admissible evidence	Packet logging	Pre-incident collection	Logging	Evidence collection	collection	Plan Pre-incident collection	Forensic copy
5	Digital preservation	Preservation Phase		Evidence preservation	Authenticating evidence	Preservation of evidence required for corporate governance	Preservation of evidence	Preservation of digital evidence	Preservation	Preserving evidence	Authentication to proof integrity	Evidence handling procedures	Preservation
6	Pre-incident analysis		Acquisition and analysis		Analysis	Monitoring of purpose to deter security	Analysis	Planning pre-incident analysis	Analysis	Evidence analysis		Plan Pre-incident analysis	Examination

						incidents							
7	Storage	Infrastructure readiness	Ensuring secure digital evidence storage	Forensic acquisition		Establishing policies for handling and storing evidence securely		Storage and handling		Evidence storage	Storage	Plan Pre-incident storage	
8	Incident detection			Intrusion detection systems		Stating when an escalation to a full investigation should start	Incident response plan	Incident collection		Incident detection		Plan Pre-Incident Detection	
9	Event Reconstruction	Reconstruction Phase											
10	Forensic Reporting	Reporting		Reporting				Reporting	Reporting	Presentation			
11	Forensic Readiness Policy		Identifying the policies that are needed to achieve digital forensic readiness			Establishing a policy for secure handling and storage							
12	Digital Evidence Collection Requirements					Determining evidence collection requirement		Legal requirements					
13	Concurrent Processes							Concurrent processes		Actionable principles			

In view of the above, it is the researcher’s opinion that the modification of botnets to act as forensic agents for purposes of forensic readiness is an important contribution. This is because the processes proposed in the CFRaaS model facilitate proactive activities that allow for an effective response to potential security incidents when the cloud environment is digital forensically ready. In fact, the availability and isolation of forensically collected potential evidence also allow companies to have litigation preparedness without disrupting any business processes. Based on this holistic approach, the researcher strongly believes that the scope covered by the proposed model is worth being explored by digital forensic practitioners and forensic experts.

The researcher also managed to compute the total number of processes that each of the compared forensic readiness models possesses (see Figure 7.16). This figure shows the variations that exist between the proposed CFRaaS model and other existing readiness models.

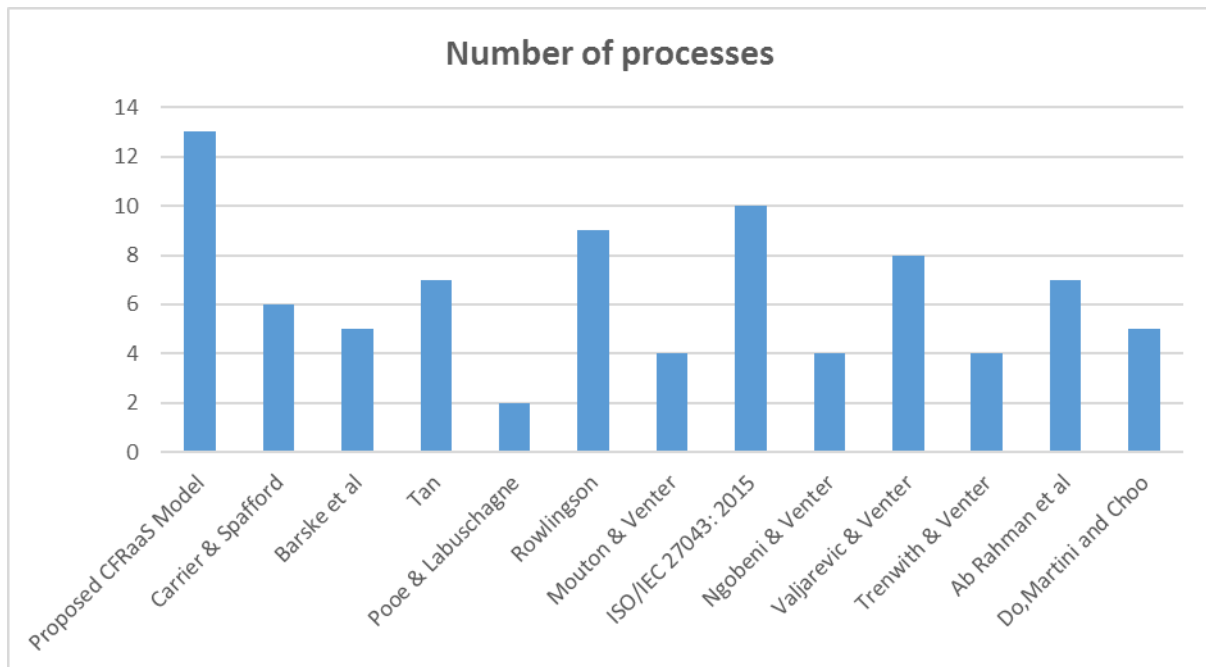


Figure 7.16 Comparisons of Proposed CFRaaS and Existing Forensic Readiness Models

Figure 7.16 illustrates that the proposed CFRaaS model consists of 13 processes, which is slightly more than the other models. There is, however, a number of common processes.

Some of the models have fewer processes because their readiness process starts from a collection of PDE – unlike the proposed model, which begins with planning and preparation. It is worth noting again that none of the models highlighted in Figure 7.16 focuses on the cloud environment. The proposed model is also an ad hoc model, which means other relevant processes can be incorporated easily.

Having considered the comparison of the proposed model with other existing forensic readiness model, the next section concludes this chapter.

7.7 Conclusion

This chapter focused on the proposed CFRaaS model. Firstly, the reader was introduced to the high-level discussion of the CFRaaS model in Figure 7.1, which was divided into five processes: *Provider layer*, *Virtualisation layer*, *Digital Forensic Readiness layer (DFR)*, *Incident Response Procedure layer (IRP)* and the *Concurrent Processes*.

Thereafter, a detailed CFRaaS process model design followed – presented as block diagrams from Figure 7.2 to Figure 7.14. A final detailed diagram to illustrate the CFRaaS model was shown in Figure 7.15. The detailed design showed the techniques for achieving DFR in the cloud when an NMB is used as a forensic agent. A bot client that forms part of an NMB was used as a forensic agent to collect PDE that could be used for forensic readiness purposes. A comparison of the CFRaaS model and other existing models was subsequently presented in Table 7.3 and the total number of processes used by the CFRaaS model was compared to those of other existing forensic readiness processes.

In the next chapter, the reader is introduced to the design of the CFRaaS prototype.

“To give a causal explanation of an event means to deduce a statement which describes it, using as premises of the deduction one or more universal laws, together with certain singular statements, the initial conditions ... We have thus two different kinds of statement, both of which are necessary ingredients of a complete causal explanation.”

-Karl R. Popper-1959

Part Four: Prototype

Part Four, which explains the practical steps needed to build a prototype, consists of **Chapter 8** and **Chapter 9**. Chapter 8 introduces the design of CFRaaS prototype while Chapter 9 introduces the prototype implementation as a proof of concept on the best way to conduct DFR in the cloud environment. Chapter 8 begins by highlighting an overview of the prototype with the prototype requirements. Thereafter, the chapter 9 shows the novel architecture of the prototype and discusses and describes the prototype that the researcher developed for achieving DFR. The prototype shows how PDE can be collected from the cloud environment using a botnet with modified functionalities that is able to work in a non-malicious fashion.

Chapter 8: CFRaaS Prototype Design

8.1 Introduction

The previous chapter introduced the reader to the architectural design of the Cloud Forensic Readiness as a Service (CFRaaS) model and its capabilities. CFRaaS has been presented as a basic building block for a functional prototype that can be used to achieve DFR.

The distribution and elasticity of the cloud has, in this regard, made it a challenge to conduct Digital Forensic investigations (DFIs) because of the costs associated with modifying and manipulating with the functionality and/or infrastructure of the existing cloud architecture (Kebande & Venter, 2014a). However, the aim of DFR while conducting these investigations is to maximise the potential use of Digital Forensics Evidence (DFE) while minimising the cost of performing a conventional DFI (Rowlingson, 2004).

In order to achieve DFR in the cloud environment, the researcher modified a botnet's functionality to act as an agent-based solution in a cloud environment. This involves deliberate, but controlled 'infection' of a VM by using a bot client to collect digital forensic information that can be used for DFR purposes. The basic concept of this design is to optimise the botnet to act in a non-malicious way, so that it is able to collect DF information that can facilitate DFR for organisations.

This chapter's main focus is to present the design of the CFRaaS prototype in the best way possible to show how an NMB infection can be realised in virtualised environments. The chapter highlights the design approaches that have helped to achieve the objectives proposed in this research thesis.

The remainder of the chapter is structured as follows: The chapter continues by giving an overview of the CFRaaS prototype in Section 8.2. Section 8.3 subsequently introduces the CFRaaS prototype main components which are then followed by the design of the CFRaaS prototype in Section 8.4. The chapter is concluded with Section 8.5.

8.2 CFRaaS Prototype Overview

This section describes the CFRaaS prototype that is able to forensically collect digital forensic evidence from a constantly changing environment. In order to illustrate the realisation of the proposed CFRaaS model, a CFRaaS prototype that acts as a proof of concept was implemented. The prototype is presented as a software application whose functionalities were modified to act as a Non-Malicious Botnet (NMB), where a bot client which forms part of the NMB is able to collect digital information as Potential Digital Evidence (PDE). The main functionalities of CFRaaS include being able to perform digital forensic evidence capture, communicate with a Command and Control (C&C) centre, store the evidence and at the same time preserve the integrity of logged information from the cloud environment.

Employing a prototype for CFRaaS would significantly increase the chances of admissibility of PDE that can be used in a court of law for litigation, civil or criminal cases. This is owing to the fact that admissibility of digital evidence requires a Digital Forensic Investigation (DFI) process and a DFI requires the application of standardised processes. This claim can be supported by using ISO/IEC 27043: 2015 guidelines in this research thesis. From the perspective of an enterprise, this can be achieved by maximising the use of PDE when needed (Rowlingson, 2004). This can help to plan and prepare for, and possibly thwart potential security incidents in any organisation. In addition, the collected digital evidence is used to facilitate the DFR process. The processes used to realise the prototype complies with the readiness processes that have been mentioned in the ISO/IEC 27043: 2015.

8.3 CFRaaS Prototype Main Components

In this section, the researcher is devoted to explaining the CFRaaS prototypes' main components. The CFRaaS prototype consists of two main functional components, namely the C&C server and the bot client(s). The C&C server is used to give and register new instructions to the bot client, while the bot client is executed in the VMs as part of an "infection" approach. Infection, which in the context of this research thesis has a positive connotation, was achieved by employing the bot client's infection vectors that are executed on the hosts' VMs (see Figure 8.1). In this context, an infection vector is an automatic propagation or transmission of malicious code, which is mostly done via the exploitation of a

vulnerability. Its main task is to “infect” a target VM by installing the bot client executable (see Figure 8.1). Note that the VM should be installed with a deliberate vulnerability in order for the VM to be “infected” by a bot client.

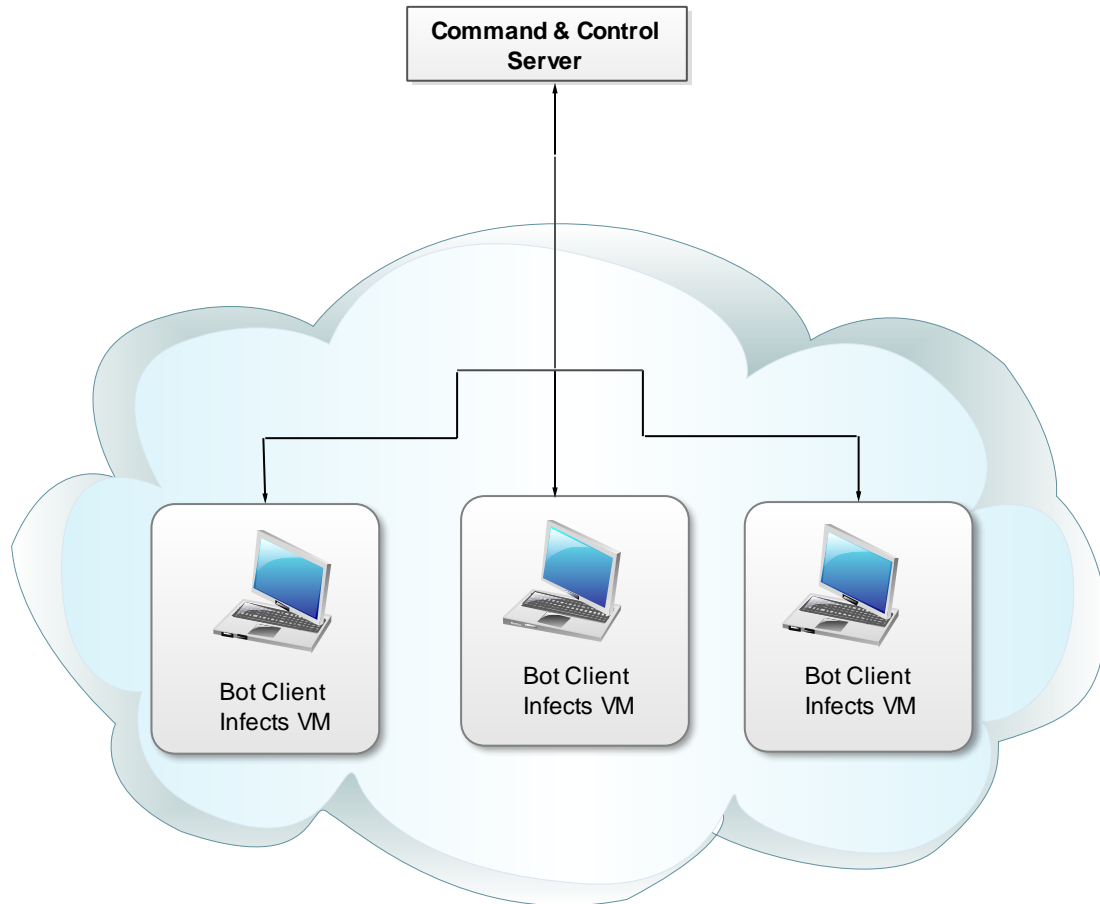


Figure 8.1 CFRaaS Prototype Main Components

In the context of this research, the C&C server is able to collect and gather PDE from the hosts’ VMs and send it to the forensic database for possible analysis. Infection vectors are used to execute the bot client to the target VM, as was shown in Figure 8.1. Each of the above-mentioned functional components will be discussed in the subsections to follow.

Having presented an overview of the main components, the researcher begins with a discussion on the C&C server in the next section.

8.3.1 Command and Control Server

In order for forensic agents to be able to receive instructions and transmit digital data, they should be able to communicate with their specific C&C server (Dietrich, Rossow & Pohlmann, 2013). Thus, the C&C server listens for incoming connections from active bot clients, receives data from the bot client and then forward commands to the bot client to collect digital information from VMs.

The C&C server is optimised for multi-threading so that it can handle multiple connecting bot clients. Once the bot client has been deployed, the C&C server listens for incoming connections from a bot client that has already been executed in the system (active bot client). At this instance, the C&C server operator issues new commands for execution or saves the digital information that the bot client is sending to the C&C server.

A more detailed discussion on how the C&C server relates to the other components mentioned in this research thesis follows in the later sections of this chapter. First, however, the reader is briefly introduced to the bot client.

8.3.2 Bot Client

In the context of this research thesis, a bot client was used as a forensic agent that can be executed in the VM for purposes of collecting digital forensic information and then reporting back to the C&C server. The bot client is executed in the VM in an infection approach, but this is done legitimately – with a positive as opposed to a malicious connotation. Information that is gathered by the bot client can be used in a Digital Forensic Readiness (DFR) approach if a potential security incident is detected. To achieve all these tasks, the bot client should be executed on the VMs in the simulated cloud environment through the infection vectors. A more detailed explanation on how this process is realised follows in the subsequent sections of this chapter.

The overview of the CFRaaS prototype main components is followed in the next section by a discussion of the design of the CFRaaS prototype.

8.4 Design of the CFRaaS Prototype

This section reports on the design of the prototype. The researcher focuses on the technical goals, deployment and virtual environment, technical specifications, functionalities and detailed components that were used to realise the prototype. The CFRaaS prototype, which is designed to be implemented in the cloud environment, has the aim of collecting – in a forensically sound manner – data in a bid to preserve it as PDE. The tasks to be performed include building a software prototype with (as mentioned before) the functionality of a modified botnet that acts in a non-malicious fashion so that it can collect PDE that can be used as admissible evidence in a court of law. To ensure that PDE will exist in a legally acceptable (i.e. forensically sound) manner, we list the design aspects of the CFRaaS prototype (Sections 8.4.1 to 8.4.4) and then give an explanation of each design aspect.

The CFRaaS prototype was designed based on the CFRaaS model proposed in Chapter 7 of this research thesis. The CFRaaS model was the basis for developing a software prototype with functionalities of a Non-Malicious Botnet (NMB). The NMB is able to collect digital forensic information from the cloud and use it for Digital Forensic Readiness (DFR) purposes. The researcher considered processes *a* (*CFRaaS Approach Strategy*), *b* (*Digital Evidence Collection*) and *c* (*Pre-incident Analysis*) in Layer 3 – the Digital Forensic Readiness Layer (DFRL) – (see Figure 7.4). Table 8.1 shows the processes and sub-processes of the CFRaaS processes that were considered in the design of the CFRaaS model.

Table 8.1 CFRaaS Design Processes and Sub-Process

	Digital Forensic Readiness Layer Process	Considered sub-processes
a	CFRaaS Approach strategy	<ul style="list-style-type: none"> • NMB Deployment
b	Digital Evidence Collection	<ul style="list-style-type: none"> • Bot client infection • Digital evidence capture • Digital preservation (Hashing) • Storage in forensic database as payload and hash data
c	Pre-Incident Analysis	<ul style="list-style-type: none"> • PDE assessment

Note that the processes and sub-processes that were not mentioned in Table 8.1 are considered as future work and an explanation for this decision is given later in this chapter.

Having looked at the design of the CFRaaS prototype, the technical goals are discussed in the next section.

8.4.1 Technical Goals

In order to find PDE that can link a suspect to a crime in the cloud environment, intruders' footprints are examined based on the forensic log files that are collected using a DFR approach. Due to the existence of large volumes of data resulting from the proliferation of digital devices and an increase in the number of users in the cloud, it might be impossible to track the source of potential attacks. However, having a human, using a manual approach, to filter this data or logs to look for evidence related to digital crimes is an extremely tedious and time-consuming task.

Such limitations can be overcome using the CFRaaS prototype that the researcher developed. The prototype pursues the following technical goals:

- It monitors activities in the cloud environment using a bot client that acts as a distributed forensic agent.
- It gathers digital information that is forensically preserved and then transmits it to a centralised forensic database for secure analysis.

The above goals have been motivated by the fact that monitoring digital forensic activities and digitally preserving evidence provides a more secure and a centralised approach for managing digital evidence. Moreover, these forensic actions are able to support real-time monitoring of potential security incidents.

In order to achieve the two technical goals mentioned, the reader needs to understand the environment in which the CFRaaS prototype is deployed and in the next section, the deployment and virtual environment of the CFRaaS are discussed in detail.

8.4.2 Deployment and Virtual Environment

Martini and Choo (2012) pointed out that data generated by the CSPs is important in the analysis of the evidence. Based on this stance, the researcher thinks that information from cloud instances can be monitored and as a result, the prototype is deployed in the cloud

environment where monitoring, digital forensic evidence capture and digital preservation are conducted. Moreover, it becomes a challenge to collect digital evidence for DFR purposes in the cloud because cloud instances are destroyed (i.e. deliberately dismantled by its owner) as easily as they are created. Cloud instances exist as VMs, which are usually placed in different physical nodes and are offered through virtualisation. Sometimes a conflict may arise during job scheduling based on how the cloud instances are placed at different physical nodes (Guo, Qian, Han & Zhang, 2015). Apart from that, instances may also appear and disappear as a result of the following:

- If a host goes down as a result of network failure, the instance may go down or disappear at the same time.
- There is usually some competition of disks for Input/Output (I/O) between the cloud instances. If one of the disks used by a cloud instance manages to go down, the instance may disappear too.
- An adversary with malicious intent may choose to disrupt a given instance. For example, an adversary may remotely inject a script that targets a vulnerability on the hypervisor to disable it.

As a result of the malicious actions that can occur to the VMs as listed above, the bot client is deployed to the VMs to legitimately “infect” them and gather digital information. This information is then sent to a centralised server for analysis.

The cloud OS acts as an intermediary between the VMs and the physical resources. Physical resources represent application servers, storage or data centres. The Oracle VirtualBox 5.1.20 virtualisation tool that was used in this approach was preferred as a hypervisor component because of its rich way of supporting large numbers of guest OSs and the fact that it has pre-built VMs (Vahidi & Ekdahl, 2013). The researcher was able to simulate the cloud environment because both VirtualBox and the cloud environment are able to host virtual instances.

Following the discussion of its deployment and virtual environment, the next section helps the reader to understand the technical specifications of the CFRaaS prototype.

8.4.3 Technical Specifications

An overview of the technical specifications (i.e. programming language, platform, database specifications and integrity checking tool) to realise the CFRaaS prototype is presented in this section.

8.4.3.1 Programming language

The prototype code was developed in a Windows-based environment using the C++ programming language and PHP application framework as a server-side scripting language. C++ was the most preferred language for this environment because of its capabilities in terms of networks, components, managing memory, support to various protocols and efficient CPU utilisation (openP2P, 2017). In addition, C++ was preferred because it provides the capabilities for the bot client to run on multiple platforms. To maximise portability across the cloud environment, the researcher used the standard C libraries within C++ and custom built all the components of the bot client that are executed in the cloud environment.

8.4.3.2 Platform

A web-based platform was used while deploying the prototype to allow digital evidence to stream in from the simulated cloud environment. A web-based platform was preferred because the cloud resources are provisioned over the web. The prototype was implemented by using the Software-as-a-Service (SaaS) cloud service model, which makes it ideal for communicating with the C&C server in the course of collecting digital forensic information. SaaS applications can easily be hosted centrally since they are provided in a client-server context. Another factor that makes SaaS ideal, is that SaaS applications can easily be deployed to computing devices over the internet. In developing and testing the prototype, the researcher used a physical computer (that was able to host the VMs) with the specifications as shown in Table 8.2.

Table 8.2 System Specifications for Development and Testing

Prototype Development System Specifications	
Operating System	Windows, 64-bit
Processor	Intel® processor Core™ i5-750 CPU
Processor Base Frequency	2.66GHz
RAM	4.0 Gigabyte
Hard drive	500 Gigabyte
Network Adapter	Intel® 82572EI Gigabit Ethernet Controller
Cache speed	8M cache

The next section presents a discussion on the database that was used to develop the prototype.

8.4.3.3 Database

The prototype was implemented using the MYSQL open-source relational database (MySQL, 2001), which was preferred because of its scalability, speed, security and ability to be accessed on a cross platform (Dong & Li, 2015). MYSQL database was also preferred because of its strong data security layers that are able to protect data from intruders (Zhang, Ning & Yang, 2016), and its ability to handle huge amounts of data (the prototype has to deal with huge amounts of PDE). Lastly, MYSQL was preferred due to its ability to run on different OSs and the fact that it implements confidentiality by means of encryption.

8.4.3.4 Integrity Checking Tool

In order to verify the integrity of the PDE, the researcher used the MD5 & SHA-1 Checksum Utility tool. This utility was used to generate hash values and at the same time to verify the integrity of the captured forensic logs. The MD5 & SHA-1 Checksum Utility tool was designed to generate MD5 and many more types of hash values for any log or data that is given as an input. It is also able to support multi-threaded checksum calculations and to create checksums from directories and subdirectories. A more detailed explanation on how this tool was used to check the integrity of the collected digital evidence follows in the later sections of this chapter.

The focus next shifts to the operation of the prototype.

8.4.4 CFRaaS Prototype Operation

This section contains a detailed discussion of the operation of the CFRaaS prototype. In particular, the researcher describes the operations that led to digital information collection and the functions that are involved in these activities.

8.4.4.1 High-Level NMB Process

In order to effectively present the NMB process, a high-level overview of the process is presented first, followed by a detailed explanation of the various threads within the NMB process. Figure 8.2 shows a high-level diagram of the NMB process consisting of the following parts that are implemented as threads and labelled **A** to **H** respectively: Main thread, connection listener thread, installer thread, ping server thread, evidence capture thread, update thread, destroy thread and send data thread.

Each of the above-mentioned threads is discussed individually and each is recognisably different from the other, i.e. each thread can readily be distinguished from the other. However, the threads are bound by communication links. How these links were used to bind the threads will become apparent later in this chapter and a merged figure is presented in Figure 8.12.

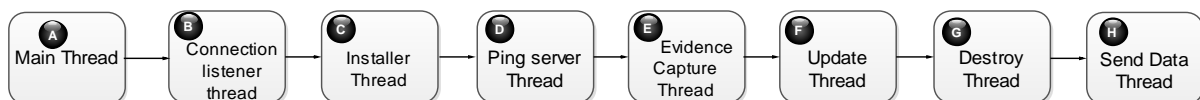


Figure 8.2 High-level view of the NMB process

Some of the threads are initiated by the C&C server while others are initiated by the VM. Threads A, B, C, F and G are initiated by the C&C server while threads D, E and H are initiated in the VM by the bot client. These processes are illustrated in Figure 8.3.

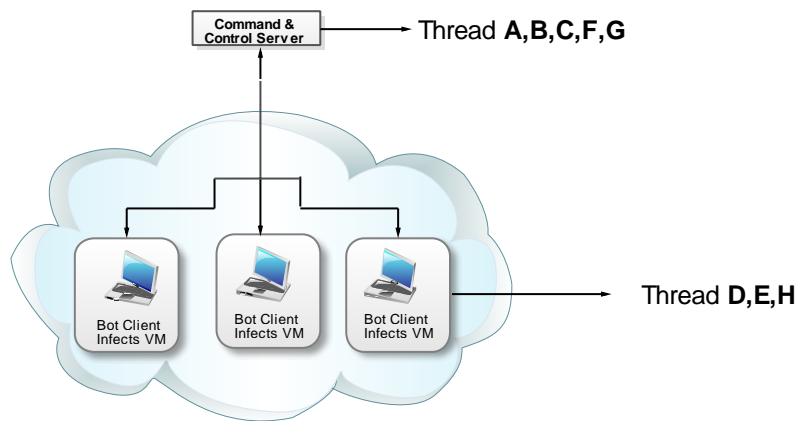


Figure 8.3 Processes Initiation by the C&C Server and the Bot Client

Although the threads are initiated by the C&C server and the bot client (see Figure 8.3), note that the actual execution of the threads take place as follows: Threads A, B, C, F, G execute on the C&C server while threads D, E, H execute on the VM. (See later also Figure 8.12).

For efficient communication of the NMB process (threads **A** to **H**), frequent requests and responses occur between the bot client and the C&C server. The next section gives a detailed explanation of the NMB process.

8.4.4.2 Detailed NMB Process

In this section, the researcher discusses each thread in the NMB process (Figures 8.4 to 8.11). A combined flow diagram of the entire NMB process appears later in Figure 8.12. However, before looking at each thread in detail, the communication scheme used between the threads is discussed next.

The communication scheme that was used between these threads is presented in Table 8.3, and explanations for each thread and its respective communication link are presented in the subsections that follow.

Table 8.3 Representation of Communication Links

Message ID	Source	Destination	Message Code	Message content
(I)	A1	B10	X	Start connection listener thread
(II)	A2	C16	X	Start installer thread
(III)	D23	B11	P1	Ready state after start of installer thread
(IV)	D23	B11	P2	Receive payload data
(V)	D23	B11	P3	Receive hash data
(VI)	B12	B13	X	Formulate evidence capture/update or destroy command
(VII)	B14	A5	X	Process payload data
(VIII)	B15	A7	X	Process hash data
(IX)	C20	D21	X	Start ping server thread
(X)	D23	B11	X	Send ping code
(XI)	B13	D24	X	Receive evidence capture/update or destroy command
(XII)	D25	E28	X	Start evidence capture thread
(XIII)	D25	F36	X	Start update thread
(XIV)	D25	G42	X	Start destroy thread
(XV)	E34	H46	X	Start send data thread
(XVI)	F39	G44	X	Delete bot executable from disk
(XVII)	G46	F41	X	Update bot client
(XVIII)	H47	D23	X	Transmit data

Based on Table 8.3, the communication between the bot client and the C&C server is presented in the following communication format, consisting of five fields separated by colons:

{(Message ID): Source: Destination: Message code: Message content}

where **Message ID** is a unique identifier that is used in the identification of communication links. The range of values consists of Roman numerals. **Source** and **destination** represent the origin and target of the communication link respectively, and are numbered according to the particular thread and action. For example, A1 would refer to action number 1 in thread A. **Message code** was used as a tag that identifies or confirms that a particular type of message sent or received. Symbols used for the message code could either be P1, P2 or P3, of which the meanings will be explained later. This field is often not used and then an X is used to

indicate it as such. Lastly, **message content** is the subject matter that the communication links aim to achieve, often referred to as the message payload.

The above communication format between the bot client and the C&C server is used to discuss the threads in the NMB process in subsequent subsections. Nevertheless, the researcher would like to state upfront that whenever a message code does not apply, a symbol (**x**) is used.

As an example, consider Message (I) in Table 8.2:

{(I): A1: B10: X: Start Connection listener}.

The first field (**I**) refers to the **message ID**, **A1** refers to the source action while **B10** represents the destination action. **X** was used to show the absence of the message code while **Connection listener** is the message content that is fulfilled by the actions.

As mentioned in Section 8.4.4.1, the NMB process consists of eight threads numbered by symbols A to H (see Figure 8.2): main thread, connection listener thread, installer thread, ping server thread, evidence capture thread, update thread, destroy thread and send data thread. Whenever a thread is mentioned, its corresponding symbol is shown in brackets after the name of the thread, for example, *main thread (A)*. (This is done for easy reference to the threads in Figure 8.12.) Each thread, in turn, consists of a number of actions.

A description of the first thread, i.e. the *main thread (A)*, as well as its actions, is given in the next subsection.

8.4.4.2.1 Main Thread (A)

The details of the *main thread (A)*, which ensures that the *connection listener thread (B)* and the *installer thread (C)* are up and running, are shown in Figure 8.4 (see the outgoing arrows labelled **B10** and **C16** respectively). The *main thread (A)* also listens for incoming data that is collected by the bot client that would originate from the *send data thread (H)*, which is also shown by incoming arrows labelled **B14** and **B15** respectively. Based on the order that was

previously highlighted, the communication links for the *main thread* (A) are shown in Table 8.4.

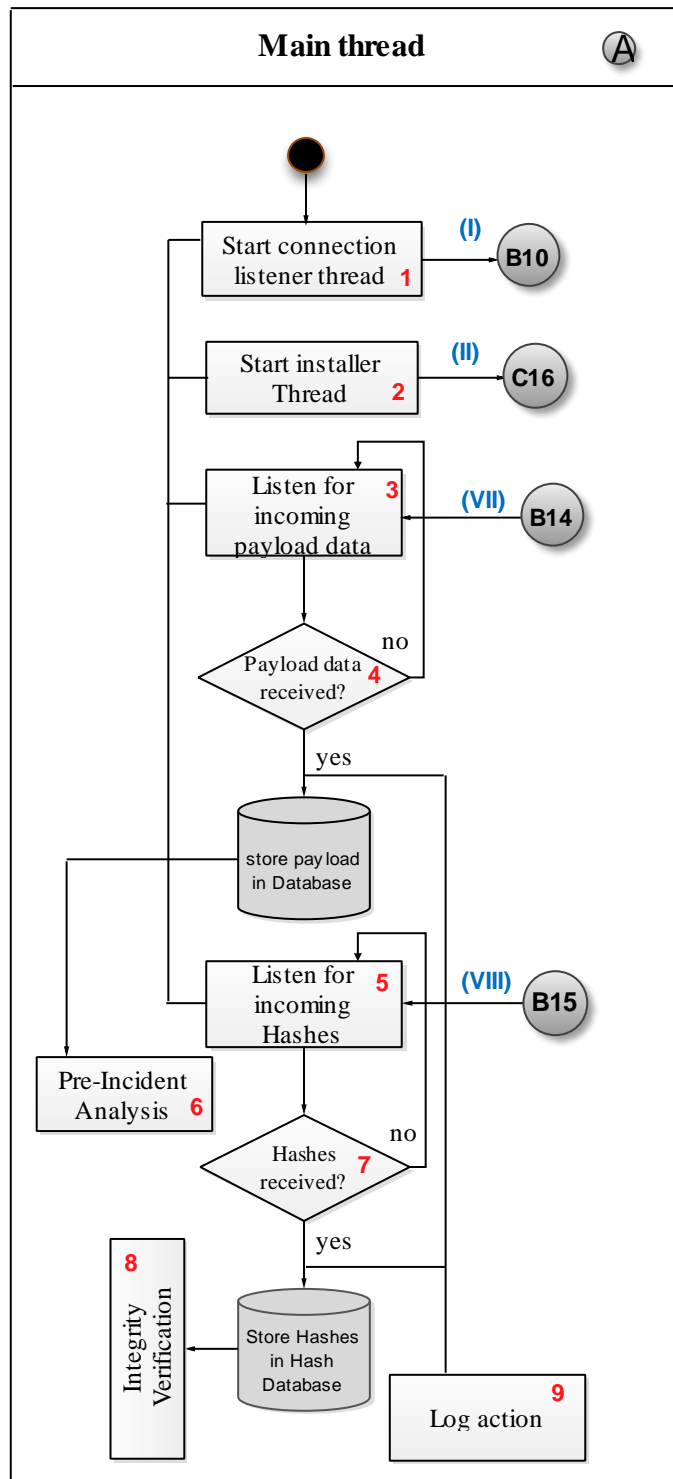


Figure 8.4 The Main Thread

The first column gives the message format, while the second column gives a short functional description of the message. Note that the messages in Table 8.4 and subsequent tables are discussed at the thread from which they originate. For example, the *main thread (A)* involves four messages, i.e. message (I), (II), (VII) and (VIII); however, only messages (I) and (II) are discussed at *main thread (A)*, since these two messages originate at *main thread (A)*. Messages (VII) and (VIII) originate from thread B and will be discussed there. Also note that messages that have already been discussed at a particular thread, will not be stated and discussed again in subsequent threads. This practice is followed in subsequent tables.

Table 8.4 Communication Links of the Main Threads

Message	Functional Description
{(I): A1: B10: X: Start Connection listener thread}	Connection listener is started to ensure the bot client connects to the C&C server
{(II): A2: C16: X: Start installer thread}	Installer is started so as to ensure that the bot client is installed in the VM

Message (I) is read as follows: Initiate communication at action A1 (that is, action block 1 in thread A), indicating that the connection listener thread should be started in thread B, action block 8, while no message code is required (indicated by the field with the X). The rest of the messages can be read in a similar way.

From Figure 8.4, the actions for *Start connection listener thread*, *Start installer thread*, *Listen for incoming payload data* and *Listen for incoming hashes* are executed at the same time at the start. When data is received from a bot client (via the *send data thread (H)* from the bot client), the *main thread (A)* assesses the kind of data that is received. If the data is payload data (from **B14**), it is transmitted to the payload database. If it is hashes (from **B15**), they are transmitted to the hash database. Actions such as the successful receiving of payload data and hashes are logged for each payload or hash data transaction. Note that arrows that were used throughout these threads represent the source and destination communication links between the threads and the actions.

The *main thread (A)* basically ensured the house-keeping of the NMB process. In the next section the *connection listener thread (B)* is discussed.

8.4.4.2.2 Connection Listener Thread (B)

The *connection listener thread (B)* that is shown in Figure 8.5 listens for incoming ping requests from the *ping server thread (D)*.

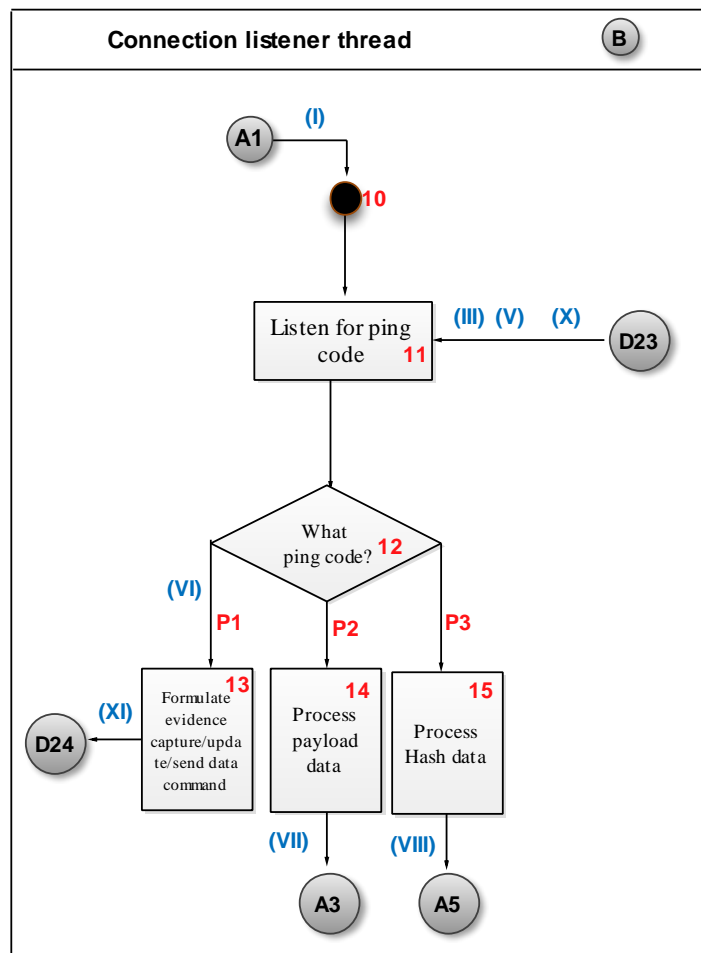


Figure 8.5 Connection Listener Thread

Once the *connection listener thread (B)* listens and receives the ping code from the *ping server thread (D)* in **D23**, it determines what kind of ping is received. The received ping codes are represented by codes **P1**, **P2** and **P3** respectively. Ping code **P1** was used to show that the *connection listener thread (B)* is in a ready state to dispatch commands (evidence capture, update or destroy) after the start of the *installer thread (C)*. These commands are dispatched to the *ping server thread (D)* as is shown by an outgoing arrow from **B13** to **D24**. Next, ping code **P2** was used to show that the payload data from the *send data thread (H)* has been received in **B9**. Whenever the *connection listener thread (B)* receives code **P2**, it is able

to process payload data, which is sent to the *main thread (A)*. This was shown by means of an outgoing arrow labelled **B14** to **A3**. Lastly, ping code **P3** shows that the hash data from the *send data thread (H)* was received in **B11**. Whenever the *connection listener thread (B)* receives code **P3**, it is able to process hash data, which is sent to the *main thread (A)*. This was shown using an outgoing arrow labelled **B15** to **A5**.

Note that as long as the *connection listener thread (B)* has not received a ping code, it will keep listening for ping requests. The communication links for the *connection listener thread (B)* are represented in Table 8.5.

Table 8.5 Connection Listener Thread Communication Links

Message	Functional Description
{(XI): B13: D24: X: Formulate Command}	Connection listener dispatches evidence capture, update and destroy command to the ping server thread
{(VII): B14: A3: X: Process Payload data}	Payload data is sent to the main thread
{(VIII): B15: A5: X: Process Hash data}	Hash data is sent to the main thread

The *connection listener thread (B)* ensures that for each request that is received, a response should be dispatched. However, a bot client needs to be installed in the client VM. The next thread accomplishes this.

8.4.4.2.3 Installer Thread (C)

The *installer thread (C)* is started by the *main thread (A)* and this is shown by the incoming arrow labelled **A2**. The *installer thread (C)* checks if the infection vectors have installed the bot client to a target VM and, if not, installs them as shown in Figure 8.6. If the bot client is installed, then it is definite that the infection process has occurred. Once the installation is complete, it makes the necessary system changes when the operating system starts up, without having to change the hypervisor.

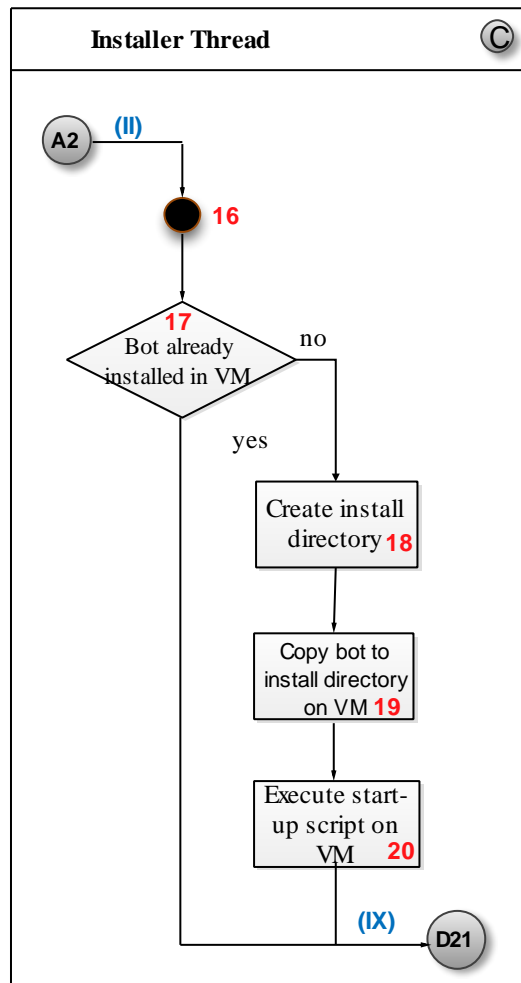


Figure 8.6 Installer Thread

The binaries of the bot client are sent in the form of a worm via the network to exploit a deliberate vulnerability on the VM without requiring any user interaction. Hence, the bot client is able to launch this exploit from the host VM. Having launched the exploit, the bot client itself executes a start-up shell script on the target VM and then it triggers the *ping server thread (D)* (see the outgoing arrow labelled **D21** in Figure 8.6). Once executed in the VM, the bot client is able to interact with the VM by collecting digital information and then communicating back to the C&C server. The *installer thread (C)* creates an install directory through which a bot client is copied to and then it is able to execute the start-up script, which in turn triggers the *ping server thread (D)*. The communication link for the *installer thread (C)* is represented as is shown in Table 8.6.

Table 8.6 Installer Thread Communication Link

Message	Functional Description
{(IX): C20: D21: X: Start ping}	Installer thread triggers the ping server thread to connect to the C&C server

The *installer thread (C)* shows how a bot client can be installed to perform an infection to the target VM by exploiting a deliberate vulnerability. To achieve this, a number of requests are made between the *connection listener thread (B)* and the *ping server thread (D)*. A discussion on how the *ping server thread (D)* achieves this, follows in the next section.

8.4.4.2.4 Ping Server Thread(D)

The *ping server thread (D)* shown in Figure 8.7 is triggered by the *installer thread (C)*, as shown by an incoming arrow labelled **C20**. It works on a timer and is able to contact the *connection listener thread (B)* periodically. Every minute, the *ping server thread (D)* wakes up and contacts the *connection listener thread (B)* (outgoing arrow labelled **B11**) after receiving a command from the *send data thread (H)*.

The command that is received from the *send data thread (H)* (incoming arrow labelled **H48**) is meant to transmit data to the *main thread (A)*. After this, the *connection listener thread (B)* is able to communicate back by sending a command that is destined to a specific thread (i.e. evidence capture, update or delete). This action is shown by an incoming arrow that is labelled **B13**. In this context, a specific thread represents any thread that is executed by the *ping server thread (D)* after a response is received from the *connection listener thread (B)*. For example, based on the response that is received from the *connection listener thread (B)*, the *ping server thread (D)* can capture evidence, update a bot client or destroy a bot client.

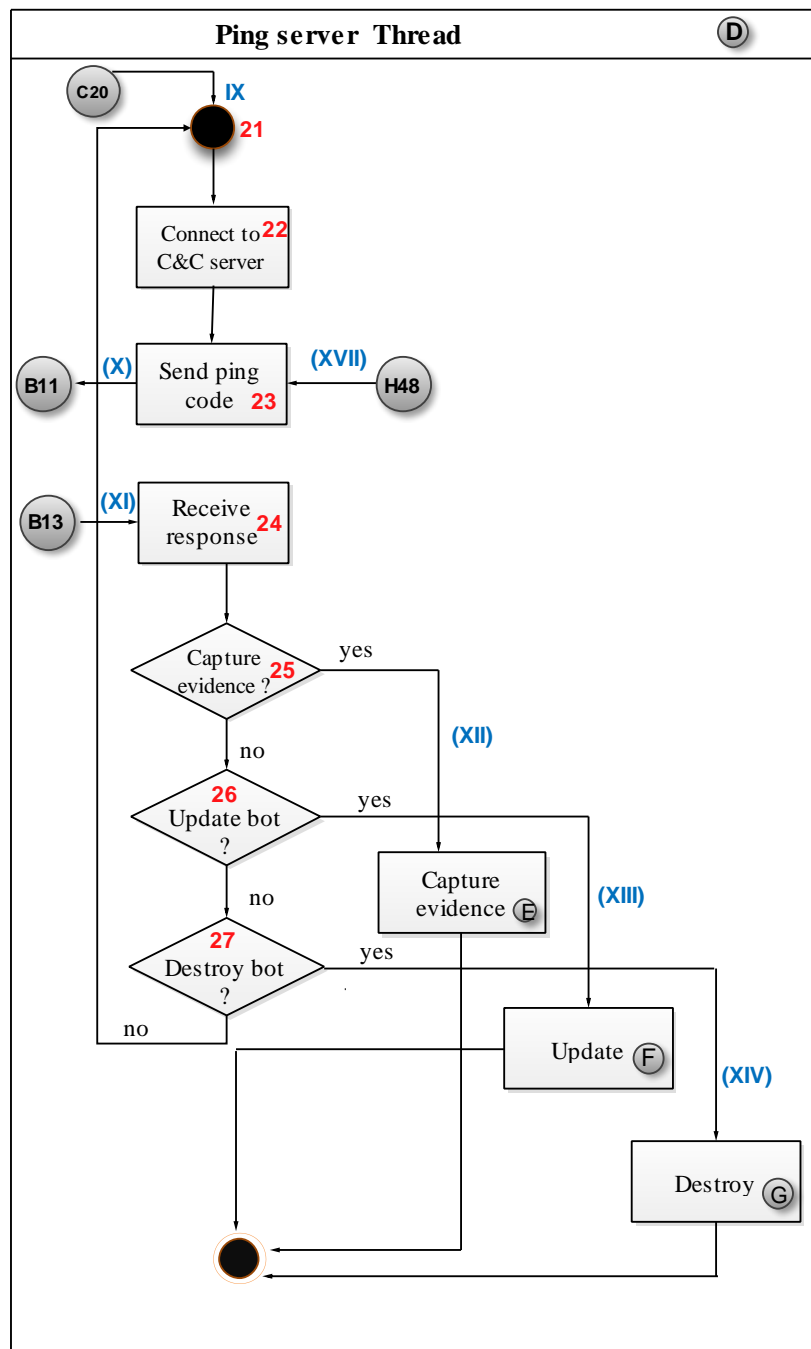


Figure 8.7 The Ping Server Thread

The communication links for the *ping server thread* (D) are shown in Table 8.7.

Table 8.7 Ping Server Thread Communication Links

Message	Functional Description
{(X): D23: B11: X: Send Ping code}	Ping code is sent to the connection listener thread
{(XII): D25:E28:X: Start Evidence capture}	Evidence capture thread is started
{(XIII): D26:F37: X: Start Bot client Update}	Bot client Update thread is started
{(XIV): D27:G43:X: Start Bot client Destroy}	Bot client destroy thread is started

The *ping server thread (D)* has shown that the bot client is able to communicate to the C&C server to be certain that the communication is alive. The main reason why the *ping server thread (D)* checks if communication is alive, is to ensure that it is able to receive commands that enable digital evidence to be captured. This process is discussed next.

8.4.4.2.5 Evidence Capture Thread (E)

The *evidence capture thread (E)* is triggered by a command from the *ping server thread (D)* that allows digital evidence to be captured from the VM (shown by an incoming arrow labelled **D25** in Figure 8.8). After gathering digital evidence, evidence is added to the buffer. If the buffer is not full, this process of capturing and adding evidence to the buffer continues until the buffer is full. When the buffer gets full, it is copied to a file. Once the file has reached the maximum capacity that needs to be collected, the file is hashed and sent to the C&C server (shown by an outgoing arrow labelled **E47**). After this, the file is reset. The main reason why digital information is transmitted to the C&C server after a given file size has been reached, is to allow the process to return to the initial state of capturing digital information.

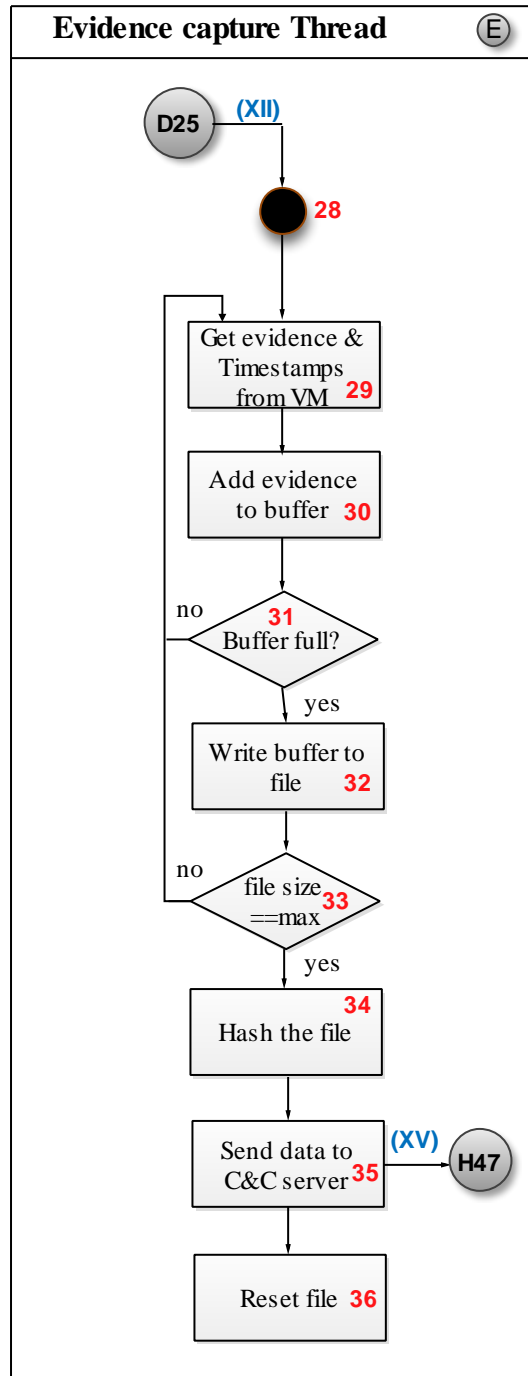


Figure 8.8 The Evidence Capture Thread

The communication link for the *evidence capture thread (E)* is shown in Table 8.8.

Table 8.8 Evidence Capture Thread Communication Link

Message	Functional Description
{(XV): E35: H47: X: Start Send Data}	Send data thread is triggered to transmit data to the C&C server

The *evidence capture thread (E)* enables the capturing of digital evidence and the posting of digital data to the C&C server. It is imperative for the reader to know how the evidence-gathering bot client can be updated and in the next section, the reader is introduced to the *update thread (F)*.

8.4.4.2.6 Update Thread (F)

The *update thread (H)* is triggered by a command from the *ping server thread (D)* that allows a bot client to be updated (shown by an incoming arrow labelled **D26** in Figure 8.9). The *update thread (H)* is able to download a new version of the bot client and then installs it. This also allows the operator to add a new functionality to the bot client.

Once the update has been triggered, the file and file size meant to update the bot client are received from the steps of the update thread (F) shown in Figure 8.9. Once the file and the file size are received, the new executable is copied to a temporary file, after which the old executable is destroyed (shown by an outgoing arrow labelled **G4**). Next, the new executable (temporary file) file is renamed with the existing executable name and then *ping server thread (D)* is started.

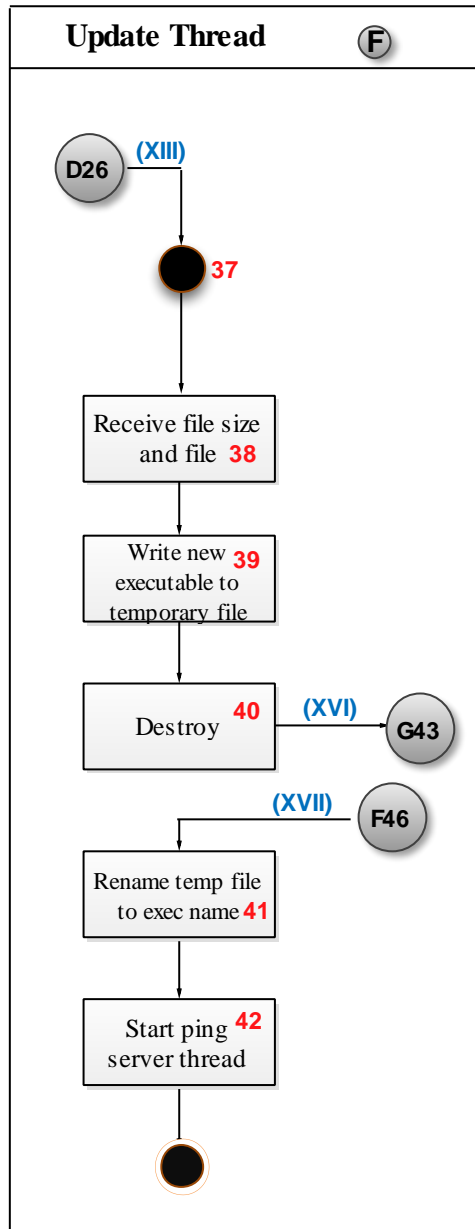


Figure 8.9 The Update Thread

The communication link for the *update thread* (F) is represented in Table 8.9.

Table 8.9 Update Thread Communication Link

Message	Functional Description
{(XVI): F40: G43: X: Destroy Bot client}	The bot executable is removed from the VM.

The discussion of the *update thread (F)* shows how a bot client can be updated with new instructions; however, the bot client can also be removed from the VM – as is discussed next.

8.4.4.2.7 Destroy Thread (G)

The *destroy thread (G)* is triggered by a command from the *ping server thread (D)* that allows a bot client to be destroyed when a newer version of a bot client is initiated or if the C&C server wants to completely demolish the bot client. This action is shown by an incoming arrow labelled **D27** in Figure 8.10.

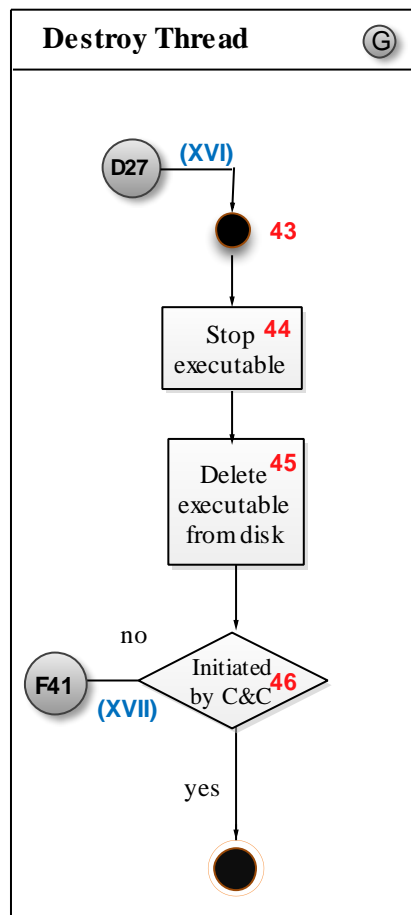


Figure 8.10 The Destroy Thread

Firstly, as is clear from Figure 8.10, the executable is stopped and deleted from the disk. If the command to destroy the bot client is initiated from the C&C sever, then the process terminates. Otherwise, if it is the *update thread (F)* that requested the *destroy thread (G)*, the latter moves back to the *update thread (F)* in **F41** where the old bot executable is replaced with a new bot executable.

The communication link for the *destroy thread (G)* is next represented in Table 8.10.

Table 8.10 Update Thread Communication Link

Message	Functional Description
{(XVII): F46: G41: X: Update bot client}	If the bot is requested by the destroy thread, then the bot client is updated; otherwise, if it is the C&C, it terminates.

Having looked at the *destroy thread (G)*, we need to see how digital data is transmitted to the C&C server using the *send data thread (H)* and this is discussed in the next section.

8.4.4.2.8 Send Data Thread (H)

The *send data thread (H)* is triggered by the *evidence capture thread (E)*, which allows a bot client to gather digital information (shown by an incoming arrow labelled **E35** in Figure 8.11). The *send data thread (H)* is able to transmit digital data collected to the C&C server via the *ping server thread (D)*, as shown by an outgoing arrow labelled **D23**.

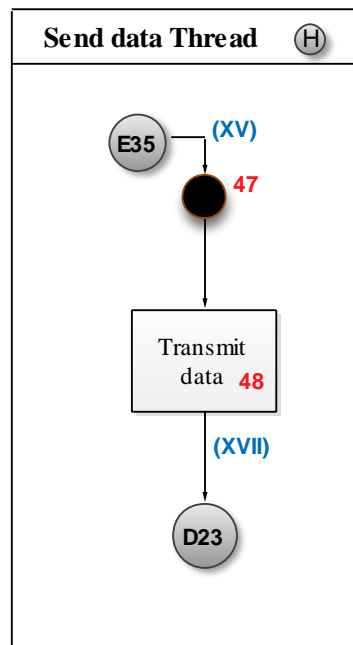


Figure 8.11 The Send Data Thread.

The communication link for the *send data thread (H)* is represented in Table 8.11.

Table 8.11 Evidence Capture Thread Communication Link

Message	Functional Description
{(XVII): H48: D23: X: Transmit_Data}	The captured digital data is transmitted to the C&C sever via the ping server thread.

Figures 8.4 to 8.11 highlighted the program flow of the NMB process and they described the following: *Main thread (A)*, *connection listener thread (B)*, *installer thread (C)*, *ping server thread (D)*, *evidence capture thread (E)*, *update thread (F)*, *destroy thread (G)* and *send data thread (H)*. However, Figure 8.12 shows a merged program flow of the NMB process, which consists of all the aforementioned threads discussed and presented in Figures 8.4 to 8.11.

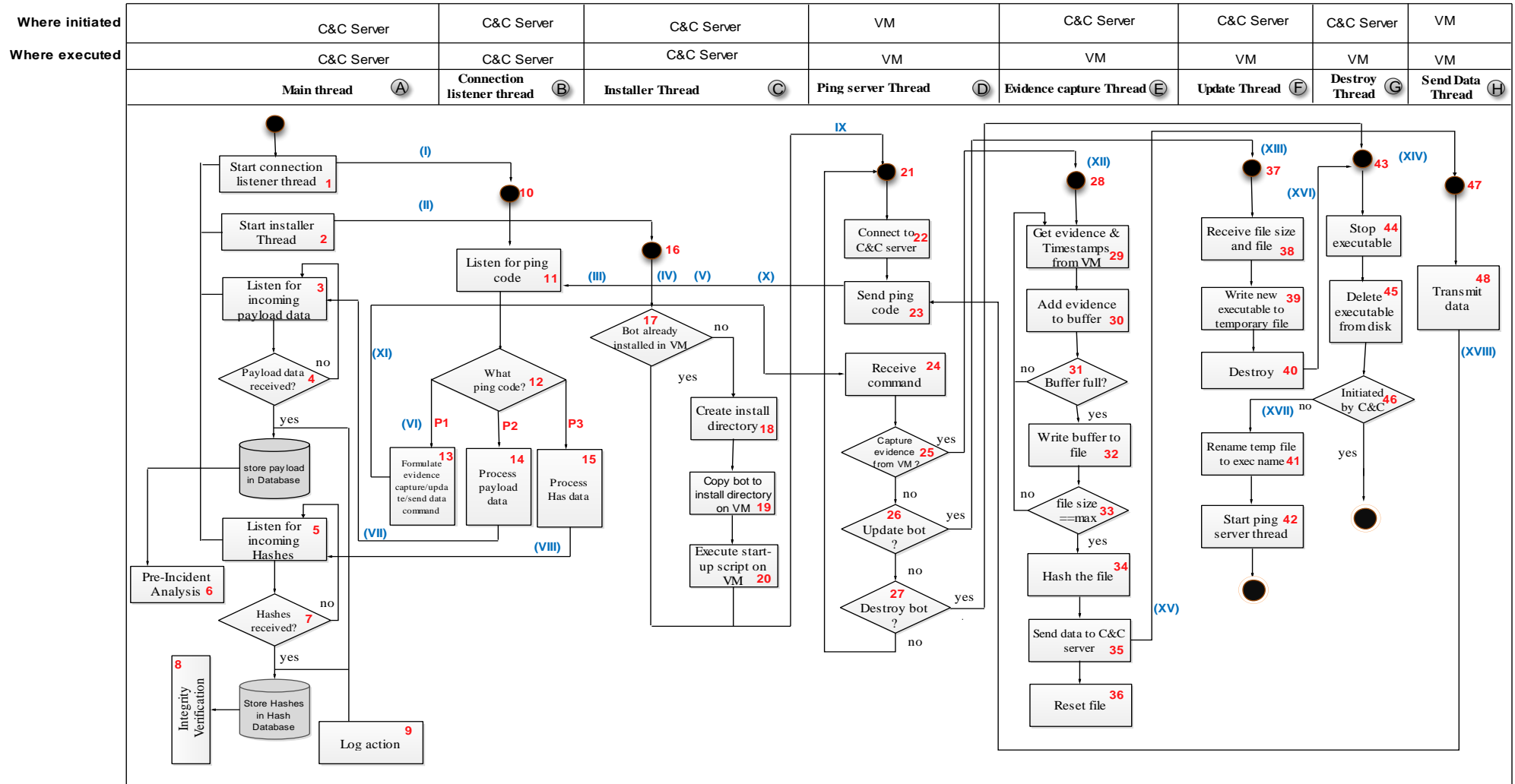


Figure 8.12 Detailed Flow of the NMB Process

The previous sections showed the NMB process that comprises of threads that are responsible for the process of capturing digital data. The chapter is concluded in the next section.

8.5 Conclusion

This chapter presented the design of the CFRaaS prototype, which consists of the basic building blocks needed to develop a functional prototype. Based on the scope that was covered in this chapter, the description of the CFRaaS design components shows that the CFRaaS design facilitates the development of a functional prototype that is aimed at achieving DFR in cloud environments.

The researcher discussed the following in this chapter: An overview of the CFRaaS prototype, the prototype requirements (by showing the main components that are needed for implementation) and, lastly, the design that shows how an NMB infection is realised.

In the next chapter, the implementation of the CFRaaS prototype is dealt with.

Chapter 9: CFRaaS Prototype Implementation

9.1 Introduction

The preceding chapter presented the design of the Cloud Forensic Readiness as a Service (CFRaaS) prototype and highlighted the basic building blocks that are needed to realise a functional prototype. Furthermore, the CFRaaS design helped to show how a Non-Malicious Botnet (NMB) infection can be realised in virtualised environments.

This chapter describes the CFRaaS prototype implementation and then demonstrates the proof of concept. This chapter's main focus is to introduce the reader to the CFRaaS prototype based on the previously proposed model. Each of the three hypothetical case scenarios that were staged in Chapter 6 requires a Digital Forensic Investigation (DFI) and as a result, a CFRaaS prototype is presented, based on the model presented in Chapter 7. This helps the reader to gain an understanding of the CFRaaS concept.

As mentioned previously, the CFRaaS prototype is represented as a botnet with modified functionalities that operates in a

3.

1`non-malicious fashion through the collection of vital digital information for Digital Forensic Readiness (DFR) purposes. All these functionalities were developed to prepare the cloud to be forensically ready for a Digital Forensic Investigation (DFI). It is worth noting again that the processes that were used comply with forensic readiness processes highlighted in the ISO/IEC 27043: 2015.

The remainder of this chapter is structured as follows: The CFRaaS prototype set-up is given in Section 9.2. Thereafter, Sections 9.3 introduce the experiment that was conducted. Section 9.4 presents the phases that were not implemented, owing to the fact that the researcher chose to give priority to the primary functionality, that is, to provide a proof of concept. The implementation challenges are presented in Section 9.5, and the chapter is concluded in Section 9.6.

9.2 CFRaaS Prototype Set-up

This section concentrates on discussing the prototype setup. The execution of the CFRaaS prototype occurs in a simulated cloud environment and involves physical systems and VMs. The cloud provides a proficient and flexible environment for deploying a bot client. In order to realise a bot client “infection”, the researcher used a laboratory set-up and a virtualised environment (see Figure 9.1). The physical and virtual environments were used for different reasons. For example, for purposes of targeting accurate results while conducting the experiment, physical computers were used. However, research by Sanabria (2007) revealed that in order to conduct analysis of a botnet in a laboratory, effectiveness can only be achieved by the use of virtualisation software. Figure 9.1 shows the environment and set-up that was used for experimentation.

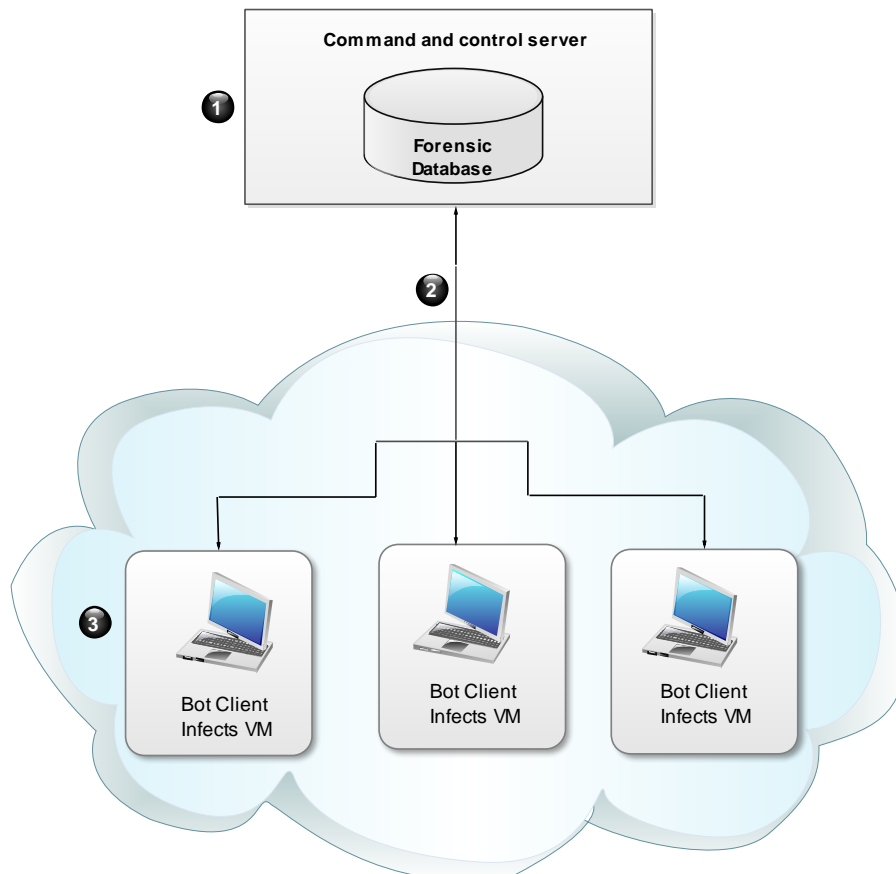


Figure 9.1 CFRaaS Experimental Set-up

Figure 9.1 shows that various components were used to set up an environment for deploying a bot client to collect traffic. An explanation of each component shown in Figure 9.1 is given next.

The task of collecting digital information is invoked by an operator. The operator uses the C&C server (Step 1) to send instructions and receive feedback from the bot client. The sending and receiving of feedback is shown by the bi-directional arrows in Step 2. In Step 3, the bot clients are executed inside the VMs in an infection approach. It is worth noting again that infection in this context carries a positive connotation. After an infection has occurred, digital information is collected and a cryptographic hash of the collected digital information is created. Next, the hashed logs are sent to the forensic database that is shown in Step 1 (inside the C&C server). Hashing is done to make sure that this information is retained in its original form during verification.

The experiment that was conducted as part of this research study is explained in the next section.

9.3 Experiment to Identify Intrusion, Theft of Personal Information and Framing among organisations

This section presents an experiment that was conducted to collect digital forensic information from a simulated cloud environment using a modified form of botnets. The experiment, which involved deploying a bot client to collect useful information from a virtualised environment, checked the activities of an intruder and reported the results that were obtained.

9.3.1 Motivation

A lack of pre-incident detection strategies can be costly and time consuming for any organisation that has to conduct a digital forensic investigation when a security incident has been detected. In the current study, the researcher illustrated the steps that a digital forensic investigator would apply to extract potential evidence that can be used for digital forensic investigation purposes. Explanations of the purpose, scenarios, execution and approaches of this experiment are provided next.

9.3.2 Experiment Purpose & Hypothetical Scenarios Used

This experiment was meant to forensically prepare a cloud environment for digital forensic investigation through the collection of potential evidence that identifies intruders who use vulnerabilities to commit digital crimes. The experiment focused on adversaries who use the cloud as an instrument of crime and suggested hypothetical case scenarios that depict digital crimes in the cloud. This experiment was conducted to test three hypothetical case scenarios that were elaborated on in detail in Chapter 6. To refresh the reader's memory, the scenarios are summarised briefly below.

In the first scenario, entitled **Information Security Breach and Identity Theft**, a company called BlueBerry that stored data in the cloud, suffered from a data breach and theft of the personal data of approximately 5000 of its employees. The stolen data was maliciously used to make cash withdrawals by using employees' debit cards in other countries.

The second scenario, entitled **Intrusion, information theft, information tampering and framing**, depicted a situation where a disgruntled employee exploited company PQR's security administrator by setting up a VM and managing to steal the administrator's credentials from the cloud environment. The disgruntled employee wiped all traces of evidence, managed to shut down his VM, and eventually succeeded in framing the administrator, in what led to the arrest of an innocent security administrator.

In the third scenario, entitled **Sexual harassment, child pornography and framing**, a convicted paedophile P managed set a service in the cloud to retaliate against the earlier actions of witnesses X and Y who had testified against him in a situation that led to his imprisonment. P downloaded Wi-Fi hacking software and managed to crack X and Y's Wireless Equivalent Privacy (WEP) encryption. This enabled P to get hold of their unique router ID. After this, P managed to open a fake account on *TB.com* (a social site) bearing X's name while in the cloud, and was able to rent a good amount of space in the cloud. He was then able to post sexually explicit videos and pictures of young boys and girls, and managed to make a collection of child pornography materials, which he emailed to X and Y's workmates (including to the CEO of company ABC and DEX respectively) while using X and Y's router as a host. Keep in mind that X and Y were working in company ABC and DEX respectively.

The purpose of these hypothetical scenarios was to show the reader how digital crimes can be conducted in the cloud environment and how the cloud itself can be used as an instrument of crime. As a result, the actions in the next section are discussed to show how the experiment was executed.

9.3.3 Execution of the Experiment

The experiment was executed by using a vulnerability to deploy a modified form of botnet that acted as a forensic agent in the cloud environment. The aim was for it to collect Potential Digital Evidence (PDE) that could be used for digital forensic readiness purposes and for investigating the claims that were highlighted in the scenarios presented in Section 9.3.3. A bot client was deployed from a C&C server to VMs to collect digital evidence. The researcher developed a software prototype that was used as a bot client and that was executed in the VM to collect digital forensic information. The developed software prototype was able to use an exploit that travels as a worm, and managed to exploit a deliberate vulnerability that executed shell code on a target VM in order to collect digital information. The digital information is collected and removed from the cloud to avoid changing the functionality of the cloud infrastructure.

This prototype was implemented to test the possibility of collecting potential evidence from a simulated cloud environment by modifying the functionality of a botnet to act as a distributed forensic agent. The prototype consisted of a forensic database that handles the storage of the collected digital information and its functionalities enabled forensic monitoring, digital evidence capture, digital preservation and pre-incident analysis.

There is always a need to follow standardised approaches while conducting forensic processes, and in the next section, the approaches that were followed in this experiment are discussed.

9.3.4 Approach Strategy adopted in the Experiment

This section further explores the approach strategy that was adopted to develop the CFRaaS prototype. Based on this approach, a digital forensic administrator is able to collect digital

information by deploying a bot client in a simulated cloud environment to gather digital forensic evidence, to digitally preserve it and store it in a forensic database.

The results of the experiment discussed later in this chapter show that it is possible to proactively collect digital information that can be used for digital forensic purposes. In this particular experiment, the bot client was able to monitor and collect traffic like CPU usage, RAM usage, keystrokes, IP addresses and timestamps using a proactive process. In summary, the following was achieved by conducting this experiment:

- The bot client executable was deployed through a deliberate “infection” approach in the VM.
- Digital information was collected in a proactive approach.
- The bot client managed to transmit collected digital information to the C&C server as potential digital evidence.
- The timestamps were recorded.
- A cryptographic hash for the collected digital information was created.
- The hashed digital information was posted to the forensic database.

The CFRaaS prototype was able to perform essential operations, referred to as threads. These threads (shown in Figure 9.2) were implemented in the design of the CFRaaS prototype (see Sections 8.4.4.1 and 8.4.4.2) and they included the following: *main thread*, *connection listener thread*, *installer thread*, *ping server thread*, *evidence capture thread*, *update thread*, *destroy thread* and *send data thread*. Other functionalities in this section were implemented inside some of these threads. For example, the *pre-incident analysis* and *integrity verification* functionalities were implemented inside the *main thread*. Similarly, *digital preservation* functionality was implemented inside the *evidence capture thread*.

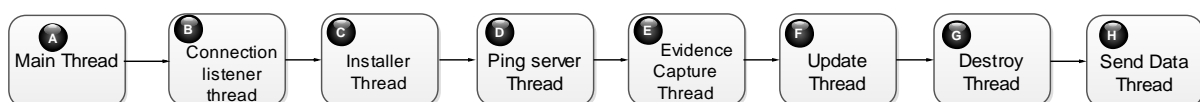


Figure 9.2 Overview of the NMB Process Threads

The threads in Figure 9.2 relate to the NMB process that was highlighted in Chapter 8 of this research thesis. The *main thread* and the *connection listener thread* were implemented

together owing to the fact that the accepted incoming commands that are destined for the *main thread* (C&C server) have to pass through the *connection listener thread*. Similarly, the *installer thread* and the *ping server thread* were implemented together since it is the installer thread that triggers the ping server thread.

Although the *forensic readiness reporting* functionality has not been fully part of the NMB process, it was discussed in this section because it was used to extract the RAM and CPU processes in the *pre-incident analysis* functionality. (Note that the *update thread* and the *destroy threads* were not implemented.)

A description of the first combined thread, i.e. the *main and connection listener thread* follows in the next subsection.

9.3.4.1 Main and Connection Listener Thread

The essence of the *main and connection listener thread*, i.e. the C&C server, is to allow the execution of the bot client in a virtualised environment and to listen for incoming data after the bot client infected the VM, and collected and transmitted data to the forensic database at the C&C server. Figure 9.3 shows an example of the C&C server that the researcher used to initiate the bot client installation. It displays different IP addresses, machine IDs, the timestamps that show when the bot clients are created and updated, and the time that the logs are received from the bot client.

IP	Machine ID	Creation Date	Last Log Received Date	Actions
196.249.12.226	309c2361-3044-47b5-b392-371f241573b8	2016-06-06 15:54:48	2016-06-06 15:58:06	Start
196.249.12.226	7bb470d0-3db-4a7d-b46e-7ce828936fa3n	2016-06-06 15:53:52	2016-06-06 15:53:52	Stop
196.249.12.226	7bb470d0-3db-4a7d-b46e-7ce828936fa3	2016-06-06 15:49:29	2016-06-06 15:50:31	Stop
196.248.150.84	734b5693-6720-4b8a-b344-12ef5dc69df	2016-05-24 12:42:41	2016-05-24 12:53:51	Start
196.248.141.195	38953bee-5525-492a-9f94-68ab2b84685d	2016-05-24 12:42:36	2016-05-24 12:56:21	Start
196.248.130.250	58543a91-5960-4566-8b77-5b82eed68e6	2016-05-24 12:42:29	2016-05-24 12:53:10	Start

Figure 9.3 Forensic Log Extraction Control Panel

Through the *main and connection listener thread*, the bot client is designated to handle specific functions during execution. For example, the bot client is a software application with

modified functionalities that depict a Non-Malicious Botnet (NMB) that should collect digital information and communicate back to the C&C server.

As was mentioned earlier, *integrity verification* and *pre-incident analysis* can be conducted inside the main thread once digital information has been received from the send data thread. These two functionalities are discussed the next subsections.

9.3.4.1.1 Integrity Verification

This section explains how the integrity of the collected forensic logs is verified. *Integrity verification* checks if the captured digital data in the CFRaaS prototype is retained in its original form and whether its authenticity and integrity is intact.

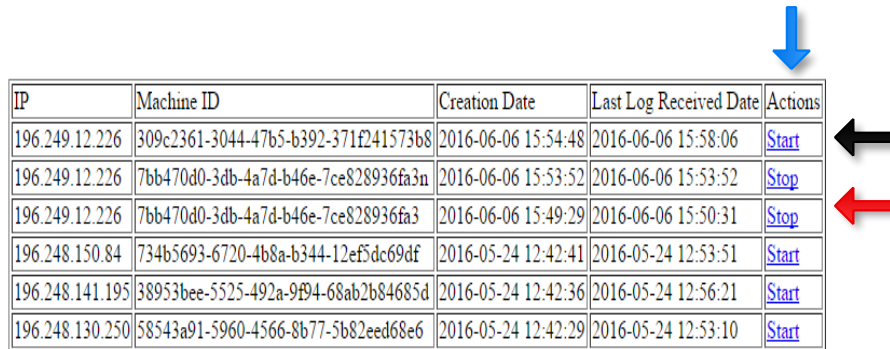
9.3.4.1.1.1 Motivation for Verifying Integrity

The integrity of PDE must be verified and confirmed by the CFRaaS prototype to ensure that the captured potential evidence is digitally preserved. Verifying the integrity of data is also meant to ensure that the evidence collected by the CFRaaS prototype has been retained in its original form. This can be achieved by verifying the generated hashes, which are discussed next.

9.3.4.1.1.2 Hash Values Generation Technique

The last column of Figure 9.4 (C&C server) is the actions column (shown by the vertical blue arrow). When the user initiates the “start” process that is shown by a black arrow, the prototype immediately begins the process of digital information capturing from the cloud environment. The “stop” functionality is not automatically enabled since digital information has to be captured first before the process can be stopped. Once the process of digital information capture is ongoing, the stop functionality is automatically invoked which gives room for the process to be stopped. Once a user clicks the ‘stop’ functionality (shown with the red arrow), the CFRaaS prototype creates an MD5 hash value that corresponds to each forensic log file that is captured. After this, the forensic log file is stored in the forensic database. The integrity of the captured forensic log is checked by verifying the hash values to determine whether the original forensic log has been altered or not. A number of factors can cause modification or alteration of forensic logs. For example, an adversary can tamper with

the forensic logs, a virus may infect the logs, or the forensic logs can suffer from a malware attack.



IP	Machine ID	Creation Date	Last Log Received Date	Actions
196.249.12.226	309c2361-3044-47b5-b392-371e241573b8	2016-06-06 15:54:48	2016-06-06 15:58:06	Start
196.249.12.226	7bb470d0-3db-4a7d-b46e-7ce828936fa3n	2016-06-06 15:53:52	2016-06-06 15:53:52	Stop
196.249.12.226	7bb470d0-3db-4a7d-b46e-7ce828936fa3	2016-06-06 15:49:29	2016-06-06 15:50:31	Stop
196.248.150.84	734b5693-6720-4b8a-b344-12ef5dc69df	2016-05-24 12:42:41	2016-05-24 12:53:51	Start
196.248.141.195	38953bee-5525-492a-9f94-68ab2b84685d	2016-05-24 12:42:36	2016-05-24 12:56:21	Start
196.248.130.250	58543a91-5960-4566-8b77-5b82eed68e6	2016-05-24 12:42:29	2016-05-24 12:53:10	Start

Figure 9.4 Actions of Creating a Hash

It is worth noting that the researcher’s focus was on forensically capturing the logs and digitally preserving them – rather than on what is contained in the forensic logs. The way in which the integrity of the collected forensic logs was verified is explained next.

9.3.4.1.1.3 Integrity Verification Technique

The researcher used the original captured forensic log files as shown in Figure 9.5 and encoded them with MD5 & SHA-1 Checksum Utility 1.1 to verify the integrity of the collected log files.



id	rawData	hash	timeReceived	ip	machineid
119	eyJkYXRhbjp7Im1hY2hpbmVVUjEjaWNGM1MTIzZWVhbnMmZS...	93711a89cc4ac1cc0b0ee0605608a124	2016-03-08 13:12:21	196.248.99.47	6
120	eyJkYXRhbjp7Im1hY2hpbmVVUjEjaWNGM1MTIzZWVhbnMmZS...	93de39c aab9aee79cf8da55a473812e2	2016-03-08 17:37:31	196.248.159.209	6
121	eyJkYXRhbjp7Im1hY2hpbmVVUjEjaWNGM1MTIzZWVhbnMmZS...	cfc017a318967c2b4606a8171f91127	2016-03-08 17:38:37	196.248.159.209	6
122	eyJkYXRhbjp7Im1hY2hpbmVVUjEjaWNGM1MTIzZWVhbnMmZS...	f0100a6577bd301e1af728d35c fac89	2016-03-09 00:41:05	196.248.96.30	6
123	eyJkYXRhbjp7Im1hY2hpbmVVUjEjaWNGM1MTIzZWVhbnMmZS...	8c6754070926157c42ca87c538a0c412	2016-03-09 06:26:45	196.248.99.38	6
124	eyJkYXRhbjp7Im1hY2hpbmVVUjEjaWNGM1MTIzZWVhbnMmZS...	3bd1b347d8a91b63cbd34da80f4fbc7	2016-03-09 06:30:36	196.248.117.128	6
125	eyJkYXRhbjp7Im1hY2hpbmVVUjEjaWNGM1MTIzZWVhbnMmZS...	9870c782d3aa846e8920b75fec ee809	2016-03-09 07:11:23	196.248.117.128	6
126	eyJkYXRhbjp7Im1hY2hpbmVVUjEjaWNGM1MTIzZWVhbnMmZS...	a9787097ca7710cbe58898e417420d1	2016-03-09 07:11:32	196.248.99.38	6
127	eyJkYXRhbjp7Im1hY2hpbmVVUjEjaWNGM1MTIzZWVhbnMmZS...	5865b097edd58949f75b7c1da3aeed3	2016-03-09 07:11:43	196.248.143.177	6

Figure 9.5 Hash of Captured Forensic Logs

Figure 9.6 shows the MD5 & SHA-1 Checksum Utility 1.1 that was used to perform integrity verification. Although the “MD5 & SHA-1 Checksum Utility” tool was not part of the development process of the prototype, the researcher employed it as secondary tool to generate and verify the hash values of the captured forensic logs.

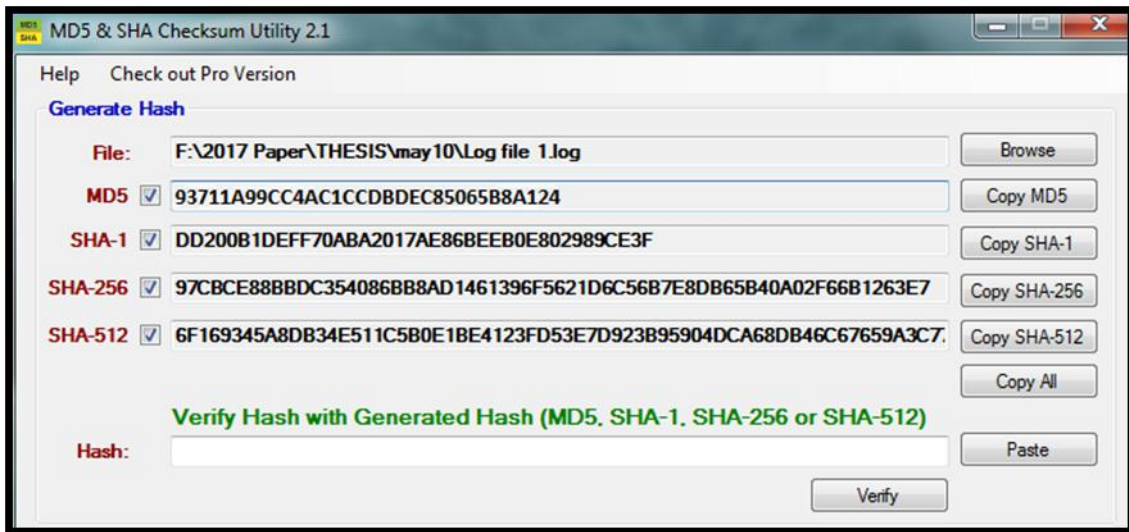


Figure 9.6 MD5 & SHA-1 Checksum Utility

The hash values that were generated by the CFRaaS prototype and stored in the forensic database (see Fig 9.5) were compared with the hash values generated by the utility tool. Note that the researcher used MD5 hash values to draw the comparison since encoding was done using MD5. Thus, the other hash values that are shown (like SHA-1, SHA-256 and SHA-512) in this utility are not considered again hereafter. If the hash values stored in the forensic database are the same as the generated ones, then the files are considered to be tamper free, otherwise files are considered to be altered, tampered with or modified.

9.3.4.1.1.4 Integrity Verification Results

This section concentrates on explaining the experimental results of integrity verification. Figures 9.7 to 9.10 show the matched MD5 hash value for the first, second and third captured log files that were generated previously by the prototype in Figure 9.5. The author also altered the contents of log file 3 and the resultant hash is shown.

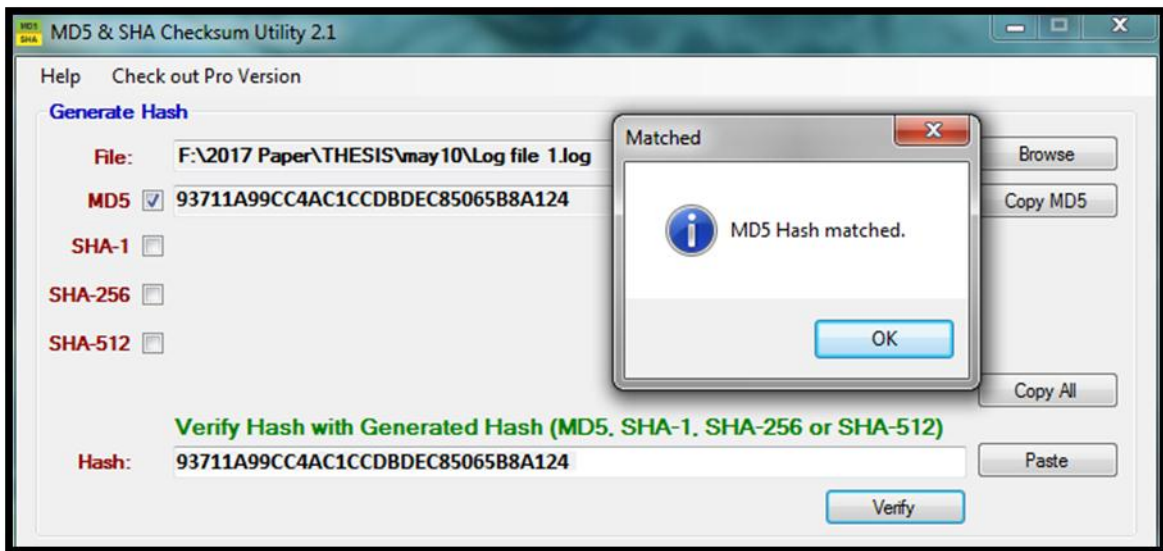


Figure 9.7 Matched MD5 Hash for Log File 1

Figure 9.7 shows that the researcher generated an MD5 hash and compared it to the hash value that was previously stored in MYSQL forensic database. The hash values matched successfully. The same was done in log file 2, which is shown next.

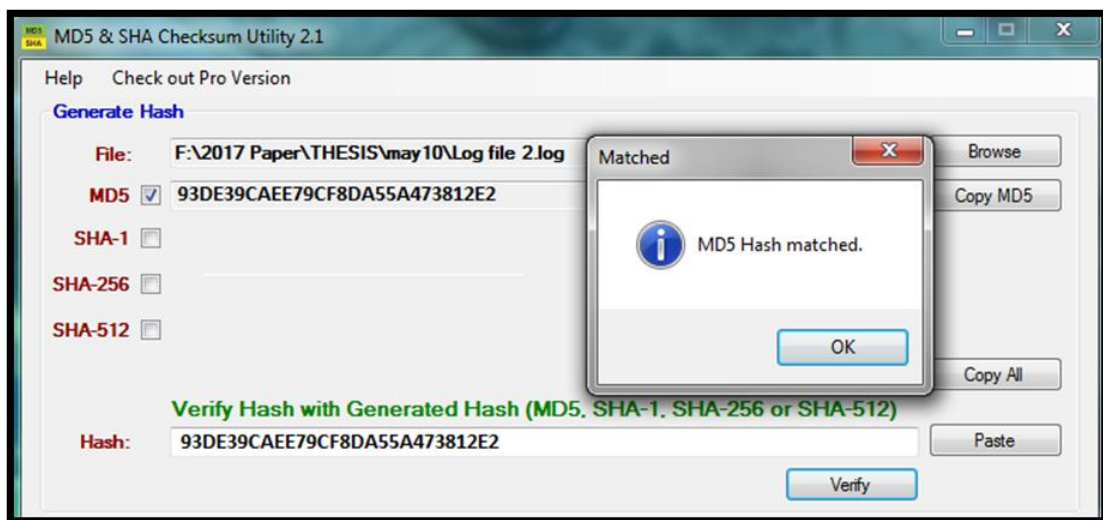


Figure 9.8 Matched MD5 Hash for Log File 2

As shown in Figure 9.8, the MD5 hash for log file 2 successfully matched with the hash that is stored in the prototype (see Figure 9.5). The same was done for log file 3, which is shown in Figure 9.9.

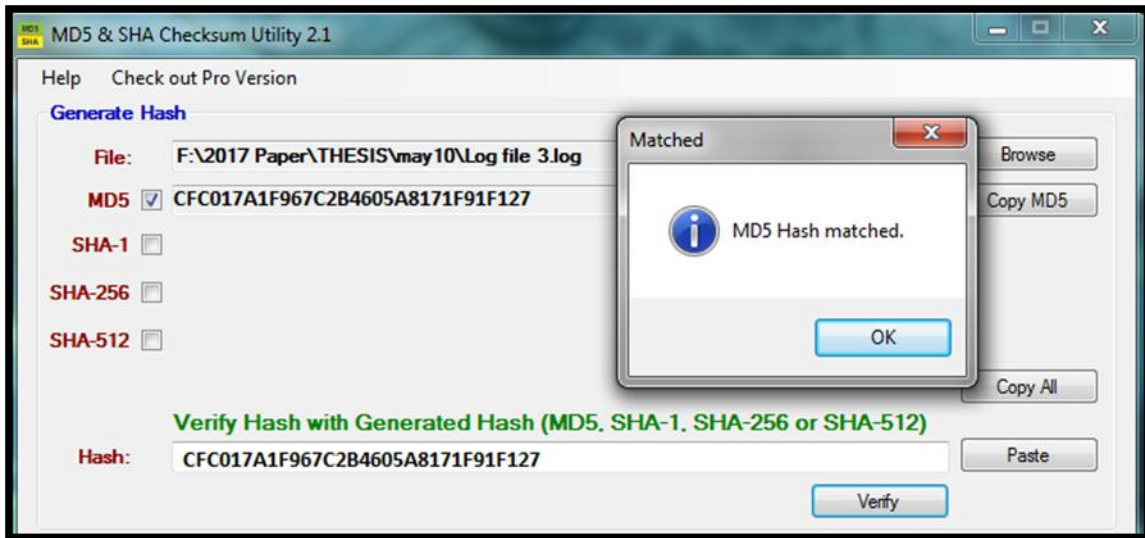


Figure 9.9 Matched MD5 Hash for Log File 3

Figure 9.9 shows that the MD5 hash value for log file 3 also matched the existing hash value that was captured by the prototype as shown in Figure 9.5.

In Figure 9.10, the researcher altered the contents (rawData) of log file 3 and the resultant hash was matched with the hash that was initially captured by the prototype. The message “Hash does not match” was returned.

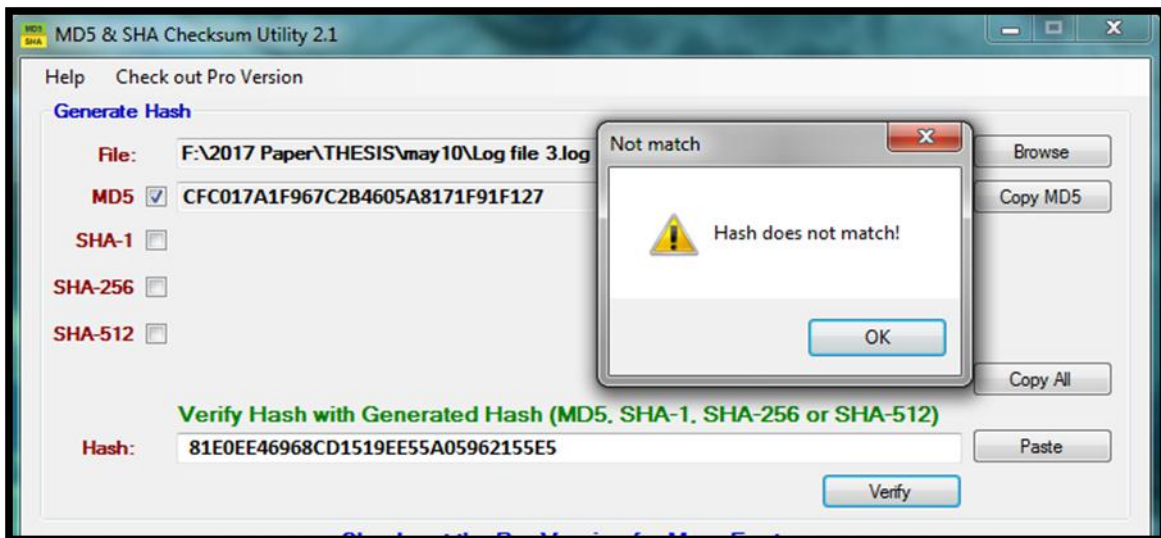


Figure 9.10 Unmatched MD5 Hash for Altered Log File 3

A table that was used to draw comparisons of the MD5 hash values that were initially collected by the prototype and it shows the hash values that were generated by MD5 & SHA-1

Checksum Utility. The researcher tested the first three hash values for log file 1, log file 2 and log file 3 respectively and then altered the contents of log file 3. A summary of this process is presented in Table 9.1.

Table 9.1 A Comparison of MD5 Hash Values Generated by CFRaaS Prototype and MD5 & SHA-1 Checksum Utility

log file Input	CFRaaS prototype Generated MD5 Hash	MD5 & SHA-1 Checksum Utility Generated MD5 Hash Value	Status
Log file 1	93711a99cc4ac1ccdbdec85065b8a124	93711A99CC4AC1CCDBDEC85065B8A124	Matched
Log file 2	93de39caee79cf8da55a473812e2	93DE39CAEE79CF8DA55A473812E2	Matched
Log file 3	cfc017a1f967c2b4605a8171f91f127	CFC017A1F967C2B4605A8171F91F127	Matched
Log file 3 (Altered)	cfc017a1f967c2b4605a8171f91f127	81E0EE46968CD1519EE55A05962155E5	Not Matched

Table 9.1 shows the output of the hash values that the researcher tested. The first three inputs for log file 1, log file 2 and log 3 matched successfully. The contents of the last input log file were altered after the prototype had generated a hash value. This alteration caused the resulting hash value to change and did not match the initial hash value. The alteration was made to show that when the originality of the captured log files is not retained, the logs may not qualify to be considered as potential evidence.

Since the respective hash values for log files 1, 2 and 3 from the prototype were a positive match with the ones generated by the MD5 & SHA-1 Checksum Utility, the integrity of the log files is said to have been maintained. This means that the forensic logs qualify to be used as potential digital evidence.

Having dealt with integrity verification (i.e. the processes involved and the results), the focus now shifts to pre-incident analysis, which is presented in the next subsection.

9.3.4.1.2 Pre-Incident Analysis

This *pre-incident analysis* allows a preliminary analysis to be made of the collected forensic logs that are stored as potential digital evidence prior to incident detection. It is worth noting that having deployed the bot client, the researcher was not able to collect human readable evidence; however, the generated graphs could depict the running processes. Pre-incident

analysis was performed by checking the live traffic that represents RAM and CPU utilisation and this procedure is explained in Figure 9.11.

id	name	username	value	total	description	date	logEntryId
14820	CPU	cnamenlyn	26	100	CPU Load	1457565683668	143
14821	RAM	cnamenlyn	60	4172963840	Ram usage	1457565684528	143
14822	CPU	cnamenlyn	30	100	CPU Load	1457565684684	143
14823	RAM	cnamenlyn	60	4172963840	Ram usage	1457565685543	143
14824	CPU	cnamenlyn	49	100	CPU Load	1457565685700	143

Figure 9.11 Collected CPU and RAM Usage, Timestamp Information

In *pre-incident analysis*, the researcher only considered the analysis of RAM and CPU usage graphs, which are explained in the next two subsections.

9.3.4.1.2.1 RAM usage analysis

The researcher deployed the bot client to collect traffic and observe the RAM usage in real time. This process involved gathering digital information that would depict the running processes. After this, this evidence was analysed and observed to detect if there was any suspicious activity. The RAM usage graph that is shown in Figure 9.12 depicts the running processes that utilised RAM. The graph was generated based on the digital data that was pushed to the forensic database.

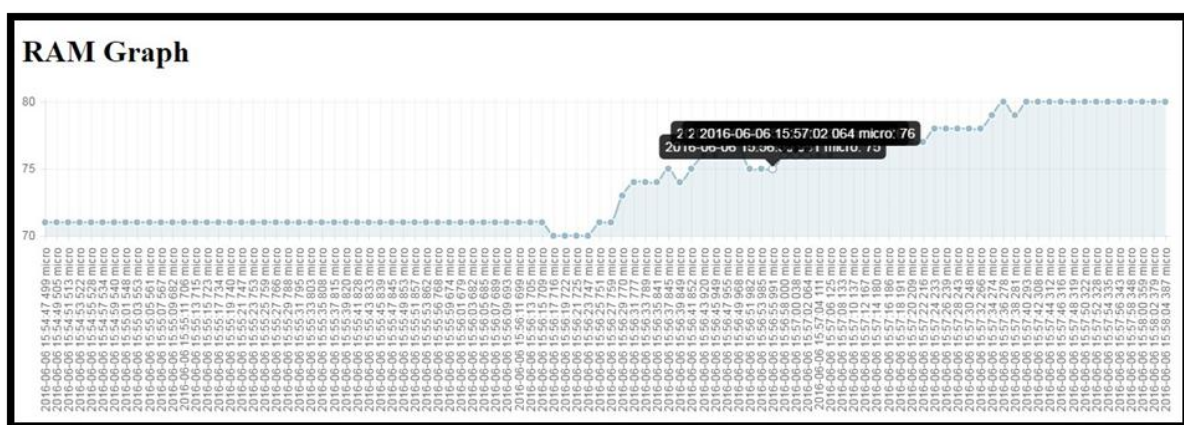


Figure 9.12 Pre-Analysis of RAM Usage Graph

The graph seen in Figure 9.12 shows the running processes and below the processes, the timestamps. Timestamps can also be seen by placing the cursor at the graph end points as

shown in the graph. The timestamp aspect also applies to CPU usage, which is discussed next.

9.3.4.1.2.2 CPU usage analysis

The researcher examined the CPU utilisation by collecting digital information that can be used to analyse the possibility of discovering anomalies (see Figure 9.13). The CPU usage graph is important to help analyse if there is or may be any unusual activity that might consume the processing power of the CPU. Monitoring this action might help to detect any unusual activity, since the aspect of CPU usage monitoring was also considered as a running process. The CPU report can be generated based on the computer name, date or username, as shown in Figure 9.13.

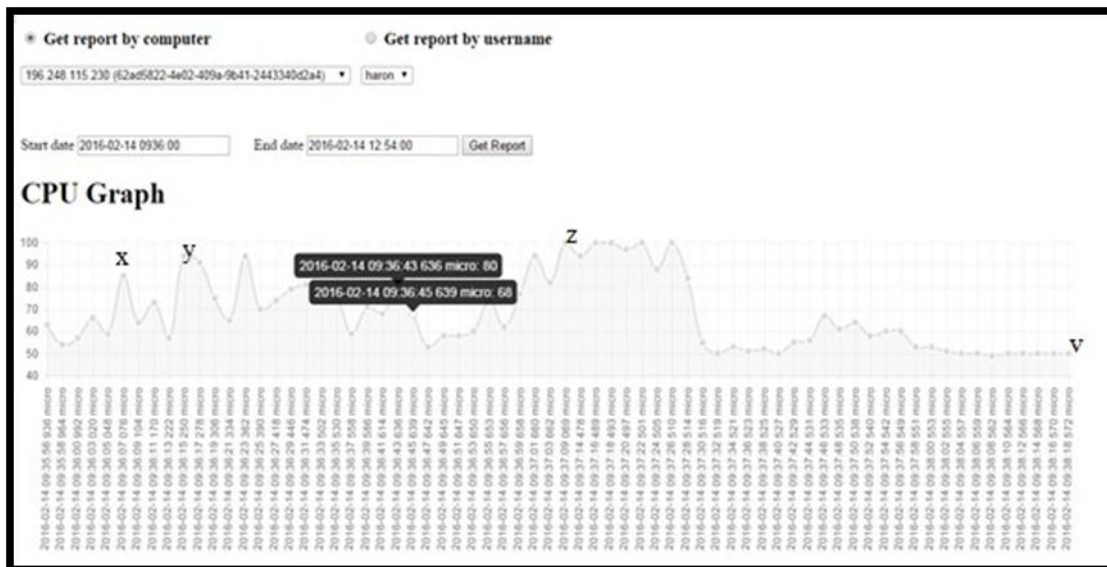


Figure 9.13 Analysis of CPU Usage Graph

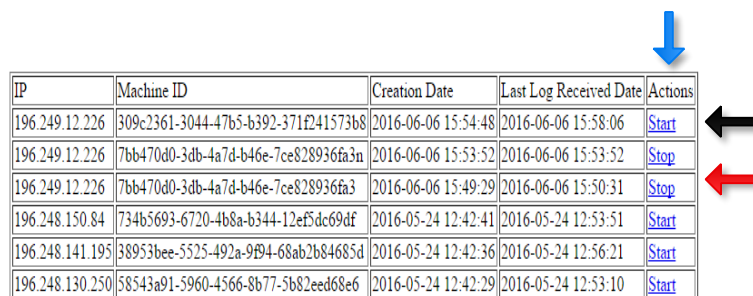
Below the CPU graph in Figure 9.13 there are timestamps that show the exact times when the CPU activities occurred. Different points of Figure 9.13 are labelled as *x*, *y*, *z* and *v*. The labelled points are examples of points that help one to monitor how the CPU is being utilised. For example, if there is an anomaly that is meant to divert the processor's power, then the points (*x*, *y*, *z* and *v*) may not remain constant. Under normal circumstances, the CPU and RAM usage patterns would not necessarily remain constant, but from the performance statistics that are shown in Figure 9.3, one would easily be able to tell when there are anomalies in terms of usage by using the usage statistics to do a normal usage prediction.

According to Jordan (2015), the following parameters might create anomalies in the CPU energy consumption rate: CPU load, memory consumed, network packets received, network packets transmitted, disk reads and disk writes.

Having discussed the *pre-incident analysis*, the next section is dedicated to a discussion of the *installer thread* and the *ping server thread*.

9.3.4.2 Installer and ping Server Thread

This *installer and ping server thread*, which is initiated at the C&C server, allows one to initiate or halt the deployment of the bot client through an infection. In this thread, one is able to control the bot client and direct it to a specific VM to perform an infection. Actions that are shown in the last column of Figure 9.14 represent control, which is used to start and to stop the infection process. Figure 9.14 shows that whenever “start” is clicked, a ping is activated and a ping code is sent to the *main and connection listener thread* where a command to capture evidence can be initiated. Alternatively, by clicking “stop”, the VM infection or the process of collecting digital information is halted.



IP	Machine ID	Creation Date	Last Log Received Date	Actions
196.249.12.226	309c2361-3044-47b5-b392-371e241573b8	2016-06-06 15:54:48	2016-06-06 15:58:06	Start
196.249.12.226	7bb470d0-3db-4a7d-b46e-7ce828936fa3n	2016-06-06 15:53:52	2016-06-06 15:53:52	Stop
196.249.12.226	7bb470d0-3db-4a7d-b46e-7ce828936fa3	2016-06-06 15:49:29	2016-06-06 15:50:31	Stop
196.248.150.84	734b5693-6720-4b8a-b344-12ef5dc69df	2016-05-24 12:42:41	2016-05-24 12:53:51	Start
196.248.141.195	38953bee-5525-492a-9f94-68ab2b84685d	2016-05-24 12:42:36	2016-05-24 12:56:21	Start
196.248.130.250	58543a91-5960-4566-8b77-5b82eed68e6	2016-05-24 12:42:29	2016-05-24 12:53:10	Start

Figure 9.14 Analysis of CPU Usage Graph

After installing the bot client at the host VM, a bot client infection process commences (as shown in Figure 9.15). This is shown by the black arrow pointing at a host location (**Host: *Logger.xp3.biz***). It is worth noting again that the motivation for this experiment was to collect digital forensic information that may be used as potential evidence.

```

RAM usage: 61% of 4083007488b

CPU Total usage: 49%
Connecting...
POST /sendata.php HTTP/1.1
Content-Type: application/octet-stream
Host: logger.xp3.biz
Content-Length: 18540

eyJkYXRhIjp7Im1hY2hpbmUuUuU1EIjo injJhZDU4MjItNGUwMi00MD1hLT1iNDEtMjQ0Mz00M0M0Q0yVTRy
IiwvY29udGUudCI6M3siZGUzY3JpcHRpb24iOiJSVW0gdXNhZ2UuILCJuYVU1Ii jo iUkFNi iwidmFsdWUu
OjI2MSIsInRvdGFsIjo inDA4Mz0uNz04OCIsInUzZXJuYVU1Ii jo iaGFyb24iLCKjYXR1Ii jo iMTQ1NTQ0
NTk3NTY4MjY5LHsiZGUzY3JpcHRpb24iOiJDUUFuG9hZCI5In5hbWUioiJDUUFuILCj2VUx1ZS16IjUu
IiwidG90VWUuioiLMDA1LjC1c2UybmFtZS16Imhhen9uIiwidGF0ZS16IjE0NTU0Nz05Nz03OTIifSx7
ImRlc2NyaXB0aW9uIjo iUmFtIHUzYVU1IiwidmFtZS16IjB1IjB1IiwidmFtZS16IjB1IjB1IiwidmFtZS16IjB1IjB1
IiwidXNlc5hbWUioiJ0YXJvbiIsImRhdGUioiIXNDU1NDc1OTc2ODAzIn0seyJkZXNjcm1udGlobiI6
IjJhbS1c2FnZSIsIn5hbWUioiJSQU0iLj2VUx1ZS16IjYxIiwidG90VWUuioiI0MDgzMDA3NDg4Iiwu
dXNlc5hbWUioiJ0YXJvbiIsImRhdGUioiIXNDU1NDc1OTc3NzExIn0seyJkZXNjcm1udGlobiI6Ikwu
USBMB2Fki iwidmFtZS16IknQUSIsInZhbHU1Ii jo iNTA1LjCj0b3RhbCI6IjEwMCI5InUzZXJuYVU1Ii jo
  
```

Figure 9.15 Installed and Running Bot Client

Following the discussion on *installer and ping server thread*, the next subsection explains how digital forensic evidence is captured.

9.3.4.3 Evidence Capture Thread

The *digital evidence capture thread*, which is initiated from the C&C server (see Figure 9.14), allows an infecting bot client to capture digital traffic in a forensically sound manner from the simulated cloud environment. There are two options in the last column of Figure 9.14 namely “Start” and “Stop”. By clicking “Start”, the C&C server invokes the bot client to a designated IP, say *196.249.12.226*, and then traffic starts to be captured. Once “Stop” is clicked, the process of capturing forensic logs is halted. Note that the options do not start automatically; they have to be initiated from the C&C server when traffic needs to be captured, and halted when the traffic collecting process must be stopped. A black arrow in Figure 9.16 shows a block of PDE that is captured when the bot client is executed in the host VM.

```

RAM usage: 61% of 4083007488b

CPU Total usage: 49%
Connecting...
POST /sendata.php HTTP/1.1
Content-Type: application/octet-stream
Host: logger.xp3.biz
Content-Length: 18540

eyJkYXRhIjp7Im1hY2hpbmUuUuU1EIjo injJhZDU4MjItNGUwMi00MD1hLT1iNDEtMjQ0Mz00M0M0Q0yVTRy
IiwvY29udGUudCI6M3siZGUzY3JpcHRpb24iOiJSVW0gdXNhZ2UuILCJuYVU1Ii jo iUkFNi iwidmFsdWUu
OjI2MSIsInRvdGFsIjo inDA4Mz0uNz04OCIsInUzZXJuYVU1Ii jo iaGFyb24iLCKjYXR1Ii jo iMTQ1NTQ0
NTk3NTY4MjY5LHsiZGUzY3JpcHRpb24iOiJDUUFuG9hZCI5In5hbWUioiJDUUFuILCj2VUx1ZS16IjUu
IiwidG90VWUuioiLMDA1LjC1c2UybmFtZS16Imhhen9uIiwidGF0ZS16IjE0NTU0Nz05Nz03OTIifSx7
ImRlc2NyaXB0aW9uIjo iUmFtIHUzYVU1IiwidmFtZS16IjB1IjB1IiwidmFtZS16IjB1IjB1IiwidmFtZS16IjB1IjB1
IiwidXNlc5hbWUioiJ0YXJvbiIsImRhdGUioiIXNDU1NDc1OTc2ODAzIn0seyJkZXNjcm1udGlobiI6
IjJhbS1c2FnZSIsIn5hbWUioiJSQU0iLj2VUx1ZS16IjYxIiwidG90VWUuioiI0MDgzMDA3NDg4Iiwu
dXNlc5hbWUioiJ0YXJvbiIsImRhdGUioiIXNDU1NDc1OTc3NzExIn0seyJkZXNjcm1udGlobiI6Ikwu
USBMB2Fki iwidmFtZS16IknQUSIsInZhbHU1Ii jo iNTA1LjCj0b3RhbCI6IjEwMCI5InUzZXJuYVU1Ii jo
  
```

Figure 9.16 A block of Captured Potential Digital Evidence

To illustrate why it is important to capture this kind of information, the researcher gives an instance where a malware might be consuming the CPU processing power or be diverting it for unwanted tasks. By capturing this kind of information, a digital forensic analyst can detect if and when a given malware is consuming the processing power, as this might end up interfering with the overall performance of the system.

Having familiarised the reader with the *digital evidence capture thread* that was used to collect digital forensic information, the researcher next discusses how the collected data is digitally preserved (hashing).

9.3.4.3.1 Hashing

Hashing shows how the CFRaaS prototype is able to preserve the integrity of the forensically logged data. The objective of digital preservation is to ensure that the captured digital forensic information is retained in its original form. Captured blocks of digital forensic information are stored as forensic logs (see Figure 9.17).

9.3.4.3.2 MD5 Encoding

The captured block of digital evidence is encoded using MD5 and the resulting hash value is stored in a forensic database (see the hash column with its corresponding log file). The generated hash is shown by a black arrow pointing vertically to the hash column in Figure 9.17. It is important to store the hashed files in the forensic database to allow verification and checking of the forensic logs' integrity when a digital forensic investigation is needed.



id	rawData	hash	timeReceived	ip	machineld
119	eyJkYXRhbjp7Im1hY2hpbmVUUUEljojNGM1MTIzZWmtNmMyZS...	93711a99cc4ac1ccdbdec85065b8a124	2016-03-08 13:12:21	196.248.99.47	6
120	eyJkYXRhbjp7Im1hY2hpbmVUUUEljojNGM1MTIzZWmtNmMyZS...	93de39caab9aee79cf8da55a473812e2	2016-03-08 17:37:31	196.248.159.209	6
121	eyJkYXRhbjp7Im1hY2hpbmVUUUEljojNGM1MTIzZWmtNmMyZS...	cfc017a31f967c2b4605a8171f91f127	2016-03-08 17:38:37	196.248.159.209	6
122	eyJkYXRhbjp7Im1hY2hpbmVUUUEljojMjVhODdmZGYtNDg3Ni...	f0100a6577bd301e61af728d35cfac89	2016-03-09 00:41:05	196.248.96.30	6
123	eyJkYXRhbjp7Im1hY2hpbmVUUUEljojNGM1MTIzZWmtNmMyZS...	8c6754070926157c42ca87c538a0c412	2016-03-09 06:26:45	196.248.99.38	6
124	eyJkYXRhbjp7Im1hY2hpbmVUUUEljojNGM1MTIzZWmtNmMyZS...	3bd1b347d8a91b63cbd34da8fbf4bc7	2016-03-09 06:30:36	196.248.117.128	6
125	eyJkYXRhbjp7Im1hY2hpbmVUUUEljojMjVhODdmZGYtNDg3Ni...	9870c782d3aa646e8920b75feccec80f9	2016-03-09 07:11:23	196.248.117.128	6
126	eyJkYXRhbjp7Im1hY2hpbmVUUUEljojMjVhODdmZGYtNDg3Ni...	a9787097ca7710cbe5f8998c417420d1	2016-03-09 07:11:32	196.248.99.38	6
127	eyJkYXRhbjp7Im1hY2hpbmVUUUEljojMjVhODdmZGYtNDg3Ni...	58b5bb97edd9894f97f5b7c1da3aed3	2016-03-09 07:11:43	196.248.143.177	6

Figure 9.17 Stored Cryptographic Hash

id	rawData	hash	timeReceived	ip	machinelid
119	eyJkYXRhbjp7Im1hY2hpbmVVVUIEjoiNGM1MTIzZWMTNmMyZS...	93711a99cc4ac1ccdbdec85065b8a124	2016-03-08 13:12:21	196.248.99.47	6
120	eyJkYXRhbjp7Im1hY2hpbmVVVUIEjoiNGM1MTIzZWMTNmMyZS...	93de39caab9aee79cf8da55a473812e2	2016-03-08 17:37:31	196.248.159.209	6
121	eyJkYXRhbjp7Im1hY2hpbmVVVUIEjoiNGM1MTIzZWMTNmMyZS...	cfc017a31f967c2b4605a8171f91f127	2016-03-08 17:38:37	196.248.159.209	6
122	eyJkYXRhbjp7Im1hY2hpbmVVVUIEjoiMjVhODdmZGYtNDg3Ni...	f0100a6577bd301e61af728d35cfac89	2016-03-09 00:41:05	196.248.96.30	6
123	eyJkYXRhbjp7Im1hY2hpbmVVVUIEjoiNGM1MTIzZWMTNmMyZS...	8c6754070926157c42ca87c538a0c412	2016-03-09 06:26:45	196.248.99.38	6
124	eyJkYXRhbjp7Im1hY2hpbmVVVUIEjoiNGM1MTIzZWMTNmMyZS...	3bd1b347d8a91b63cbd34da8bf4fbc7	2016-03-09 06:30:36	196.248.117.128	6
125	eyJkYXRhbjp7Im1hY2hpbmVVVUIEjoiMjVhODdmZGYtNDg3Ni...	9870c782d3aa646e8920b75fec ec80f9	2016-03-09 07:11:23	196.248.117.128	6
126	eyJkYXRhbjp7Im1hY2hpbmVVVUIEjoiMjVhODdmZGYtNDg3Ni...	a9787097ca7710cbe5f8998c417420d1	2016-03-09 07:11:32	196.248.99.38	6
127	eyJkYXRhbjp7Im1hY2hpbmVVVUIEjoiMjVhODdmZGYtNDg3Ni...	58b5bb97edd9894f97f5b7c1da3aeed3	2016-03-09 07:11:43	196.248.143.177	6

Figure 9.19 Forensic Log, Hash, Timestamp, Machine IP Address, Machine ID

Figure 9.20 highlights the data that was obtained as a result of key logging and transmitted to the forensic database as potential digital evidence. The figure portrays the key values that were entered every time that a key on the keyboard was pressed (shown by an arrow that points to the key value column in Figure 9.8). It also captures the timestamp associated with every time that the key was pressed and shows the log entry ID of the forensic logs that were posted to the forensic database.



id	name	username	value	total	description	date	logEntryId
10386	Keyboard	haron	k	0	Keystroke	1457004340436	87
10387	Keyboard	haron	l	0	Keystroke	1457004340653	87
10388	Keyboard	haron	k	0	Keystroke	1457004340659	87
10389	Keyboard	haron	j	0	Keystroke	1457004340731	87
10390	Keyboard	haron	d	0	Keystroke	1457004340743	87
10391	Keyboard	haron	s	0	Keystroke	1457004340753	87
10392	Keyboard	haron	c	0	Keystroke	1457004340821	87
10393	Keyboard	haron	l	0	Keystroke	1457004340849	87
10615	Keyboard	haron	[Enter]	0	Keystroke	1457004567780	90
10738	Keyboard	haron	w	0	Keystroke	1457004700695	93

Figure 9.20 Forensically Captured Keystrokes in a Readiness Approach

Figures 9.18 to 9.20 show the digital data that was captured, hashed and transmitted to the forensic database. In the next subsection, a discussion follows on forensic readiness reporting.

9.3.4.5 Forensic Readiness Reporting

The outcome of a digital investigation process or the steps taken to collect potential digital evidence are presented in a report. According to ISO/IEC 27043: 2015, reporting is an integral part of any digital forensic investigation process as it provides the findings emanating from a digital investigation process. Although the reporting module was used in the pre-incident analysis thread while extracting the CPU and the RAM usage graphs, it was implemented separately. Since the reporting module was implemented separately in this prototype, it should (in the researcher's opinion) form part of future work in this field.

Having looked at the different phase that were implemented, there is need for the reader to have a glimpse of the phases that were *not* implemented.

9.4 Phases Not Implemented

In the previous sections, the researcher stressed the fact that the focus of the CFRaaS prototype was to present a proof of concept of the initially proposed CFRaaS model (see Chapter 7), which aimed to be implemented in organisations operating in the cloud environment. Therefore, not all the functionalities of the CFRaaS prototype were implemented. The proof of concept of the prototype was conducted in a simulated cloud, as opposed to a fully-fledged cloud environment. Since collecting large amounts of data from the cloud environment for forensic readiness purposes can be tedious and time consuming, the researcher concentrated on the primary functionalities that prove that DFR can be achieved in the cloud environment with the help of a software application that operates as a botnet. Thus, the functionalities that were not implemented in the CFRaaS prototype could be considered as future work during a later phase. The phases listed below were not implemented in the CFRaaS prototype.

- **Incident Detection**– This phase was not implemented, but it will not require much effort to be implemented since it happens before the incident response. *Incident response* mainly focuses on restoring normal services and integrating forensic analysis practices into Incident Response Procedures (IRP) (Freiling & Schwittay, 2007).

- **Event Reconstruction** – This phase was not implemented in the CFRaaS prototype; however, if it were to be implemented, a reconstruction page could be added. This page would be based on the collected PDE that would match the detected events to the original PDE. For example, Carrier and Spafford (2004) argue that *event reconstruction* can be used to create a model that is able to automate the reconstruction process. This functionality would be enabled to search events from fields, locate relevant events, and check how similar those particular events are.
- **Incident Response Procedures (IRP)** – This phase significantly corresponds with the digital investigation processes. It involves a coordinated approach and outlines procedures and guidelines used in responding to security incidents (By and Krzewinski, 2013). The CFRaaS model requires policies, standards and procedures, as well as constitutional and statutory provisions across different jurisdictions to be implemented before PDE may be considered as admissible. Given that the CFRaaS model is able to achieve PDE collection timeously before *incidence response procedures* are implemented, the researcher did not consider IRP as a primary functionality. However, since it has some importance in the digital investigation process, the researcher considers IRP as future work.
- **Forensic Reporting** – As stated in Section 9.3.5.5, this functionality was only implemented separately for the prototype; however, it was used in the extraction of CPU and RAM usage graphs. This phase was not discussed into detail, rather it has been presented with the aim of showing how it fits within the CFRaaS model. According to ISO/IEC 27043: 2015, *forensic reporting* plays an integral role and reports on observations and examined processes. Tanner et al. (2012) agree that it is a process that is able to communicate the results of an investigation. Hence, *forensic reporting* will form part of future work.
- **Update and Destroy Phase** – The *update and destroy* phase that was mentioned in Chapter 8 was not implemented due to time constraints; however, this functionality is considered as future work.

In the next section, the implementation challenges that were encountered are discussed.

9.5 Implementation Challenges

This section concentrates on the implementation challenges faced by the researcher while developing the CFRaaS prototype. The researcher was able to deal with a number of challenges (listed in Sections 9.5.1 to 9.5.4) that are discussed next, based on a general, technical and operational point of view. However, for a full list and explanation of, as well as a proposal for the high-level solutions of other identified challenges, see **Section 10.6**. The implementation challenges are presented below.

9.5.1 NMB obstruction

The first challenge was how to implement the NMB in a suitable cloud environment without having it tampered with. The existence of an NMB in the cloud environment can be affected by the availability of disinfection strategies, and these strategies are able to remove the forensic clients from their functionalities. The bot client is infiltrated by many or big fake malicious programs in an attempt to disrupt communication. As a measure, the bot client binaries were obfuscated by changing their code patterns.

9.5.2 NMB implementation

A botnet is traditionally perceived to be of a devious nature and to have bad connotations. The NMB (with modified functionalities of a botnet) also collects useful information, but it is considered malicious because it captures information illegally. Although the non-malicious botnet implemented in this study operates in a non-malicious fashion, the presence within the network of possible multiple malicious activities that infiltrate the bot client may disrupt communication during a DFR approach. Furthermore, since the botnet creators modify the botnet architecture from time to time with the intent of making it more resilient, it is very difficult for researchers to implement the botnet with the existing architecture.

9.5.3 Live evidence acquisition

Since the cloud is distributed and volatile, there is no easy way of logically acquiring live PDE in the cloud environment for purposes of DFR once the bot client has gathered and retained critical digital information related to crimes.

9.5.4 Legal authority

The issue of legal admissibility of digital evidence is encountered when PDE is gathered across different jurisdictions. Issues arising from cross-cutting jurisdictions also hold a challenge when digital evidence is collected as part of the DFR process. Digital evidence that is stored in multiple jurisdictions cannot be accessed because what is considered legal in one country might be illegal in another. The lack of proper Internet laws and consistent legislation may make governments unwilling to cooperate and thus lead to delays in warrants being served (and consequently to delays in prosecution).

9.6 Conclusion

This chapter concentrated on the CFRaaS prototype and demonstrated its proof of concept. The prototype showed how PDE can be collected using a software application that acts as a non-malicious botnet. The results presented in this chapter illustrated that the CFRaaS prototype successfully performed digital forensic evidence capturing.

It is important to note that some phases of the CFRaaS prototype were not implemented. This is owing to the fact that the researcher chose to give priority to the primary functionality, namely to provide a proof of concept of the CFRaaS model (initially proposed in Chapter 7 of this research thesis).

“Thus science must begin with myths, and with the criticism of myths; neither with the collection of observations, nor with the invention of experiments, but with the critical discussion of myths, and of magical techniques and practices—in which errors are systematically criticized and fairly often, in time, corrected—that is, to play a part in the critical discussion, in the elimination of error.”

-Karl R. Popper-1959

Part Five: Conclusion

Part Five – the final part of the thesis – acts as a driver for critical evaluations of the proposals offered in this research, and shows how these propositions were implemented from the researcher’s point of view. It is the concluding part of this research study and consists of two chapters, Chapters 10 and 11. A critical evaluation of the research conducted in this study is presented in **Chapter 10**, followed by a detailed evaluation of the proposed cloud forensic model, the prototype and the research questions. **Chapter 11** is the concluding chapter that contains the novel contributions, recommendations and suggested avenues for future work.

Chapter 10: Critical Evaluation of Research Study

10.1 Introduction

In view of the ubiquity of cloud computing, many criminals have opted to use the cloud as a platform for launching malicious attacks. This it has become essential for clients of CSPs to understand how digital forensic processes can be adapted in cloud computing environments. In this thesis, the researcher emphasises the need to understand the technology and challenges that are linked to the collection of digital information from the cloud for DFR purposes. Numerous analogies regarding the inadaptability of the cloud and digital forensics processes have been put across by lawmakers, legal practitioners, policy makers, law enforcement agencies and digital forensic experts. Even though at the time of writing this thesis no effective digital forensic investigation framework that is focused on the cloud has been implemented, a good number of organisations venture to take their businesses to the cloud because of its flexibility. Therefore, digital forensics practitioners need to constantly address the intricacies experienced in the cloud, matters dealing with digital investigations in the cloud and the technology that is employed in the cloud.

In Chapters 5 to 9 of this research thesis, the reader was introduced to an environment in which a modified form of a botnet posing as a forensic agent was deployed to collect vital digital forensic information from VMs in the cloud environment as part of a proactive approach. This approach led to the development of different proactive security levels and eventually to the proposal of a Cloud Forensic Readiness as a Service (CFRaaS) model. Furthermore, in this research thesis, the proposed CFRaaS model was used as a basis for the design and implementation the CFRaaS prototype, as presented in Chapters 8 and 9.

This chapter aims to critically evaluate the extent to which the research problem presented in this research has been addressed. Next, the chapter concentrates on providing a critical evaluation of the proposed model, proposed prototype, research objectives and research questions, as well as on discussing the findings. In addition, the chapter discusses how DFR can be achieved in the cloud environment, without necessarily changing the functionality and/or infrastructure of any existing cloud architectures.

The rest of the sections in this chapter are structured as follows: A critical evaluation of the proposed CFRaaS model appears in Section 10.2. Next, a critical evaluation on the proposed CFRaaS prototype is given in Section 10.3. This is followed by an evaluation of the set objectives in Section 10.4, a discussion of the research questions in Section 10.5. After this, Section 10.6 presents the CFRaaS implementation challenges and the chapter being concluded in Section 10.7.

10.2 Critical Evaluation of the Proposed CFRaaS Model

In this section of this research thesis, the researcher presents a critical evaluation and a discussion of the CFRaaS model. The guidelines that have been used while coming up with this model comply with ISO/IEC 27043: 2015, namely Information Technology - Security techniques - Incident investigation principles and processes. At the time of writing this research thesis, the technique of using bot clients as digital forensic agents in the cloud environment for DFR purposes was still a novel concept in the process of digital forensic planning and preparation for a DFI.

In the researcher's opinion, digital forensic processes in the cloud environment are increasing exponentially due to the steep increase in the uptake of computing – specifically cloud computing – devices. As discussed in Chapter 4, botnets are known to be cynical in nature, as hostile botnets are able to capture digital information illegally. This happens when the botnet's code is malicious and when it is not used for digital forensic purposes, in other words when it has a negative connotation (Kebande & Venter, 2014). However, the NMB that was introduced in this research thesis has a positive connotation in that it operates in the cloud environment for DFR purposes, which are non-malicious. This concept was achieved by considering the legal acts as well as the constitutional and statutory provisions of a given jurisdiction. These acts show when it is allowed to gather digital information for law enforcement purposes and when not.

On the same note, research on the jurisdiction of the CFRaaS prototype has shown that it is important for the LEAs to have harmonious laws that can assist them across diverse jurisdictions (Hooper, Martini & Choo, 2013). Regarding regulation and governance, Choo (2010) argues that it is important to determine the law of jurisdiction where a Service Level

Agreement (SLA) exists. Additionally, if there are matters of interest –like national security – the CSPs may be obliged to monitor, search and report on these matters, depending on the law that applies in the applicable jurisdiction where the physical machine that needs to be investigated, is located (Gellman, 2009).

Whilst there also exist implications of collecting digital forensic information from the cloud, the laws in different jurisdictions have a provision depending on whether the information is for law enforcement purposes or to facilitate prosecution in a judicial system (Kebande & Venter, 2016c). For example, in the USA, the CSPs are governed by the Stored Communication Act (SCA) of 1986, which incorporates cloud computing (Bellia, 2003). Based on the SCA Act, parties may be compelled to disclose records and electronic communications that are stored (Kerr, 2004). However, when this Act was passed, the modern cloud characteristics were not yet considered; cloud computing was still considered a remote computing service. According to Thompson (2015), cloud computing uses the same characteristics (timesharing) as those considered when the Act was enacted. Therefore, based on the reflections of the SCA Act, parties may be compelled to disclose records and electronic communications that are stored (Kerr, 2004).

Capturing information for digital forensic purposes without consent and using a malicious code (conventional bot clients) deployed in stealth mode, might be offensive and might have legal implications when the captured information is not explicitly intended for law enforcement purposes (Kebande & Venter, 2014b). To ensure legal compliance of the CFRaaS model with regard to monitoring and admissibility of digital forensic evidence, a number of Acts were considered. The Acts considered for this research thesis include the following: Rule 702 of The Federal Rules of Evidence of USA (Watkins, 1994); Case laws for USA (United States v Mosley, 1994; Daubert v. Merrell Dow Pharmaceuticals); The Association of Chief Police Officers (ACPO-UK) (ACPO, 2007), (ACPO, 2012); Electronic Communications and Transactions (ECT) Act, Act 25 of 2002 (South Africa) (Gereda, 2006); Regulation of Interception of Communications and Provision of Communication-related Information Act (RICA), Act 70 of 2002 (South Africa) (RICA, 2001); and the Protection of Personal Information (PoPI) Act, Act 4 of 2013 (South Africa) (Gereda, 2006).

The proposed CFRaaS model not only provides a well-coordinated proactive approach that has a comprehensive view of cloud security, it is also able to orchestrate digital forensic

activities through the following approaches: collecting potential digital evidence; digitally preserving the collected information; storing the collected digital evidence; reconstructing potential security events; and reporting the results of potential security incidents.

The advantages of the proposed CFRaaS model (based on its processes) are summarised below:

1. The methodologies used in the CFRaaS model are applicable and relevant, and they will be able to support future digital forensic investigative capability and technologies.
2. The CFRaaS model complies with the standard of ISO/IEC 27043: 2015 on PDE collection processes, thus ensuring higher admissibility of digital evidence.
3. The proposed CFRaaS model is interactive and provides multi-jurisdictional collaboration, which implies that it can easily be integrated with the applicable laws. This will help during the investigation of cloud-based incidents.
4. The structure of the proposed CFRaaS, which allows deliberate infection, is very effective when performing a DFR process.
5. The proposed model provides an easy way of conducting DFR in the cloud environment without tampering with or modifying the existing architecture/infrastructure of the cloud environment.

The outcome of the hypothetical case scenarios showed that if the CFRaaS model were to be fully adopted, it might be suitable to conduct DFR processes in the cloud environment. Furthermore, the hypothetical case scenarios have shown that implementing the CFRaaS model may accelerate digital forensic investigation processes in the cloud. The researcher trusts that employing the CFRaaS approach may help to reduce the difficulties faced by forensic communities and digital forensic investigators in the cloud environment.

At the time of writing this research thesis, there existed no cloud forensic readiness model that does not require the cloud infrastructure to be changed. Thus, the researcher is convinced that the order of the processes provided in this research thesis is suitable for supporting future proactive investigative technologies in the cloud environment.

Having evaluated the CFRaaS model, the reader needs to evaluate and gain insight into the CFRaaS prototype.

10.3 Critical Evaluation of the Proposed CFRaaS Prototype

This section critically evaluates the proposed CFRaaS prototype that was developed by the researcher and that uses modified functionalities of botnets to conduct DFR in the cloud environment.

The prototype that was presented in Chapter 9 where its proof of concept was demonstrated. The CFRaaS prototype represents a systematic approach towards implementing DFR in the existing cloud environment, without having to modify the existing cloud environment. As defined previously, DFR is a proactive process that not only allows any organisation to maximise the potential use of digital evidence, but also minimises the cost of performing a DFI (Rowlingson, 2004). The CFRaaS prototype is able to support the following processes: Collecting and retaining digital information; protecting the integrity of the collected digital information; and accelerating the digital investigation process by using the collected information to plan a post-incident response when a security incident is detected or suspected.

The CFRaaS prototype processes comply with the readiness process standards that have been proposed by the international standard ISO/IEC 27043: 2015 (ISO/IEC 27043, 2015). This means that when digital evidence is required for purposes of supporting a legal process in a court of law, one would follow the processes outlined in the prototype to conduct digital forensic readiness in the cloud environment. The CFRaaS prototype, which was built based on the CFRaaS model defined in Chapter 7, allows a bot client that acts as a forensic agent to legally infect VMs in the cloud under agreed SLAs. The aim was to collect digital forensic information in a digital forensically sound way, i.e. the collected information was digitally preserved, and eventually stored as payload and hash data in a forensic database. A simulated cloud environment offered a uniform basis for the development of the prototype.

The security breaches that were experienced in all the hypothetical case scenarios (see Chapter 6) show that there were vastly insufficient proactive approaches, and a serious lack of log collection and analysis techniques that could help link the suspects to a given crime. The presence of the CFRaaS prototype could have saved the affected companies the time and

money needed to conduct a Digital Forensic Investigation (DFI). Additionally, if there was going to be a proper prosecution in a court of law, the Cloud Service Providers (CSPs) could have established a digital forensic evidence capture mechanism coupled with digital evidence preservation techniques, based on the requirements that are highlighted in ISO/IEC 27043: 2015, ISO/IEC 23037: 2012 and ISO/IEC 10118-2: 2010 international standards. On the same note, all systems could have been configured to monitor and record relevant events and to maintain sufficient historical information from which digital evidence could have been extracted.

The demonstrations of the CFRaaS prototype showed that DFR could be achieved by modifying the functionality of a malicious botnet to become non-malicious, so that it can act as a forensic agent in the cloud environment. It is important to note again that the CFRaaS prototype did not require the functionality of the existing cloud architecture to be modified, because digital forensic activities and manipulation of the collected digital evidence were conducted outside the cloud (Kebande & Venter, 2015b).

The cost of conducting a DFI is much higher when DFR has not been employed. One would prefer to have a running application that is able to forensically prepare relevant cloud data for immediate post-event response when a security incident is suspected or detected. Rowlingson (2004) remarks (as part of DFR requirements in general) that there should be minimum disruption of business processes while conducting DFI. Deploying a bot client to the cloud environment ensured that digital evidence related to potential digital crimes and with the ability to affect any organisation, could be gathered in a forensically sound manner. This was done with minimal disruption because after collection, PDE was isolated from the cloud for purposes of analysis.

The captured forensic logs were also organised in chronological order based on the time of occurrence. This would enable a digital forensic investigator to track down the time interval and the exact time that, for example, a ping was sent to the C&C server. When all these activities of tracking events were considered carefully, it became apparent that the forensic logs were captured. Since all the captured forensic logs and the timestamps reflected the actual time when an event occurred, it is apparent that these evidence was captured in a forensically sound manner. Consequently, the prototype demonstrated the following:

1. The forensic logs were captured in a way that maintained their integrity.
2. The functionality and/or infrastructure of the existing cloud architecture was not tampered with because the forensic activities of the captured data were conducted outside the cloud.
3. The proactive collection and storage of the forensic data ensured that DFR was enforced.
4. While capturing forensic logs as PDE, no business process was meant to be interrupted.

Based on the prevalent characteristics of the cloud, it is worth noting that implementing CFRaaS processes in the cloud environment poses a number of challenges. As a result of the implementation of the proposed model, the researcher was able to identify a number of challenges (mentioned in Chapter 9). However, other challenges resulting from the inadaptability of conventional digital forensics processes in the cloud environment are highlighted in **Section 10.6**. The researcher was able to map these challenges with the available literature, as shown in **Table 10.2**. The researcher also suggested that the aforementioned challenges can be mitigated through the proposed high-level solutions shown in **Table 10.2**.

Having taken note of the evaluation of the CFRaaS prototype, it is important for the reader to know how the initially proposed objectives were addressed.

10.4 Evaluation of Research Objectives

This section is devoted to the evaluation of the research objectives addressed in this research thesis. The study investigated an acceptable way of conducting DFR in the cloud environment without having to modify the functionality and/or infrastructure of the existing cloud architecture. The study also focused on designing and implementing a CFRaaS software prototype. The specific objectives evaluated in this research thesis are discussed below and summarised in Table 10.1.

- **Objective 1: To conduct a comprehensive literature review on digital forensics, cloud computing and botnets**

In order to achieve this objective, **Chapter 2** explored literature on digital forensic investigation process models, digital investigation processes, digital evidence and the DFR process. After that, the concepts of cloud computing were explored in **Chapter 3**, which showed different characteristics of the cloud and various cloud deployable models which could support the deployment of the CFRaaS prototype. **Chapter 4** discussed botnets and also showed various architectures that botnets use to propagate themselves across the network. Botnet architectures were furthermore explored to enable the reader to understand how a non-malicious botnet would be useful. Based on the literature explored in Chapters **2**, **3** and **4**, a CFRaaS model was proposed in **Chapter 7**.

- **Objective 2: To propose the requirements and techniques used to attain DFR in cloud computing environments**

Chapter 5 presented the CFRaaS model requirements that were needed to achieve DFR in the cloud environment. The requirements presented in **Chapter 5** were aimed at allowing the researcher to follow well-defined processes in the build-up for the CFRaaS model that was presented in **Chapter 7**.

- **Objective 3: Propose a novel forensic cloud model to perform DFR and propose systematic processes that can be used during PDE collection from the cloud environment:**

As at the time that this research was commencing, there existed no cloud forensic readiness model to the researchers' knowledge, which could help prepare for security incidents following the processes highlighted in the ISO/IEC 27043: 2015. Nonetheless, ISO/IEC 27043: 2015 was not focused on the cloud. The researcher, therefore, proposed a CFRaaS model with significant processes in **Chapter 7** that aimed at planning and preparing effectively for potential security incidents. The proposed CFRaaS model enabled one to follow the processes, some of which are standardised processes that are mentioned in ISO/IEC 27043: 2015. The advantage of employing the

standardised processes is to enable collected digital evidence during pre-incident detection to satisfy admissibility when presented in a court of law, as advocated by ISO/IEC 27043: 2015.

- **Objective 4: To contribute towards a prototype that acts as a proof of concept of how a proactive DFR process can be achieved in a cloud environment**

To begin with, a well-defined CFRaaS model was presented in **Chapter 7** that acted as the basis for designing and implementing the CFRaaS prototype. The model consisted of information security functional components that allowed the design and implementation of the CFRaaS prototype in **Chapters 8** and **9**. The proposed prototype provides a novel approach that can be used while conducting DFR in the cloud environment.

The CFRaaS prototype was able to capture digital forensic evidence of user and system activities from a simulated cloud environment. The captured evidence was digitally preserved and stored in a forensic database and the integrity of the data was maintained.

- **Objective 5: To show the effectiveness of the proposed model in a virtualised environment**

The CFRaaS prototype was tested in a virtual environment where it was able to collect reliable digital forensic evidence. A deliberate vulnerability was used to propagate the bot client to infect the VMs. This later allowed implementation of the digital forensic readiness process through logging actions in a non-malicious fashion. A summary of the objectives achieved in this thesis is given in Table 10.1. All the objectives have been met in full.

Table 10.1 A Summary of Achieved Research Objectives

Chapter	Summary of Objectives
Chapter 2	<ul style="list-style-type: none"> • Forensic science • Definition of Digital Forensics • Digital Evidence • Digital Forensic Investigation Process • Digital Forensic Readiness • Digital Forensic Investigative model

	<ul style="list-style-type: none"> • Legal Requirements for Admissibility of Digital Evidence
Chapter 3	<ul style="list-style-type: none"> • Defining Cloud computing • Cloud computing Architecture • Cloud computing service models • Roles of Cloud Service Providers • Virtualisation and Cloud Computing. • Adoption of Digital Forensics in the Cloud • Digital Forensic Readiness in the cloud
Chapter 4	<ul style="list-style-type: none"> • Definition of a Botnet • Life-Cycle of a Botnet • Anatomy of Botnets • Botnet Control and Administration • Usage of Botnet
Chapter 5	<ul style="list-style-type: none"> • Model Requirements for achieving Digital Forensic Readiness in the Cloud • Descriptive methodology • Scientific methodology • Formulative mathematical approaches
Chapter 7	<ul style="list-style-type: none"> • High-level CFRaaS Model • Detailed CFRaaS Model • Comparison of the CFRaaS Model with other existing Readiness Models
Chapter 8 & 9	<ul style="list-style-type: none"> • Design of CFRaaS Prototype • CFRaaS Prototype • Technical Goals • Functionality Modification • Experimental Results • Effectiveness in the virtual environment

Following an evaluation of the research objectives that were proposed in this study, the researcher next evaluates the research questions to show the extent to which they have been answered.

10.5 Evaluation of Research Questions

The research presented in this thesis focused mainly on how to achieve DFR in the cloud environment using a botnet with modified functionalities. The results presented in **Chapters 8 and 9** portray a process that begins with an “infection” in the cloud environment. The sections that follow answer the main research question and the sub-questions dealt with in this study.

10.5.1 Main Research Question

The main question addressed in this research was defined in **Chapter 1** (Section 1.2):

Is it possible to proactively prepare and plan digital forensic readiness process in the cloud environment?

The researcher focused on using a non-malicious botnet as a software application to infect the host VMs in the cloud environment to gather PDE. The researcher was mainly interested in capturing, digitally preserving and storing the captured PDE for purposes of DFR. Therefore, in this research, PDE was extracted, which could help in the creation of a hypothesis that could be used to prove or disprove a fact during criminal or legal proceedings in a court of law.

A bot client with a list of instructions that were initiated from the C&C server was able to execute a start-up shell script on the target VM. The bot client managed to extract running processes that helped forensic investigators to capture the sequence of events during post-event response.

Therefore, the research findings that address the main research question in this research thesis is as follows:

Research findings show that digital information that is extracted from the cloud environment using the CFRaaS software prototype includes the following: Running processes such as CPU usage, RAM usage, keystroke analysis, IP addresses and timestamps. With regard to this research, the collected information can be used to reconstruct potential security events, which can be used to forensically prepare a cloud environment for DFIs.

Although the main research question solves the problem addressed in this research thesis, this research was further divided into sub-questions that are explained in the following section.

10.5.2 Research Sub-Questions

The research sub-questions addressed in this research thesis were also defined in **Chapter 1** (Section 1.2) as follows:

1. What are the suitable techniques of conducting DFR in the cloud environment?

The purpose of this sub-question was to reveal ways through which DFR can be achieved in the cloud environment. To investigate this problem, the researcher built a software prototype that was able to collect digital information (see **Chapters 8 and 9**). This aim was highlighted in this thesis through the collection, retaining and digital preservation and storing of information from the cloud environment to be used in preparing and planning for security incidents.

The findings suggested that the use of a software prototype constituted a proactive approach that could help one to achieve DFR in the cloud. This software prototype was able to collect digital information that could qualify as admissible digital evidence in a court of law. The process of collecting PDE complied with the forensic readiness processes that had been proposed in the standard of ISO/IEC27.43: 2015.

Therefore, if an organisation was able to follow a well-defined and accepted digital forensic process, methods and practices, then the collected PDE could be considered as evidence that could be used to create a hypothesis to prove or disprove a fact during legal, civil or criminal proceedings in a court of law. Based on the research findings, sub-question 1 can be answered as follows:

After implementing a CFRaaS prototype, the most suitable way of conducting DFR in the cloud environment is to accumulate and retain digital evidence from cloud sources that are related to digital crimes and may have an adverse effect on an organisation.

2. Is it possible to conduct DFR in the cloud environment without having to change the functionality and/or infrastructure of the existing cloud architectures?

The main purpose of this sub-question was to provide a possible way of implementing DFR capability in the cloud environment. Hence, hypothetical case scenarios were introduced in **Chapter 6**. The scenarios discussed in this research thesis were aimed at presenting instances where DFR should have been a requirement; however, DFR was not implemented in any of the case scenarios. Based on the research findings, the CFRaaS prototype that was proposed is not compelled to be executed in a specific architecture of the cloud. This makes the architecture adaptable, flexible and customisable by any organisation and also the CSP.

The PDE that is gathered by the bot client is isolated from the cloud environment, which means that all forensic activities like computation, manipulation and DFR operations that are related to the captured PDE, happen outside the cloud. An advantage of collecting digital evidence and conducting forensic activities outside the cloud is that the use of a bot client as a software application does not allow tampering, modification or alteration of the main functionalities of the existing cloud computing architecture/infrastructure. The different layers of the CFRaaS discussed in **Chapter 7**, namely Provider layer (PL), Virtualisation layer, Digital Forensic Readiness (DFR) layer, Incident Response Procedure (IRP) layer and other sub-modules are still able to communicate and they can gather relevant PDE through suitable communication channels. Therefore, this sub-question can be answered as follows:

Research findings show that the functionality and/or infrastructure of the existing cloud architecture is not altered or tampered with when Digital forensic operations or manipulations of PDE are conducted.

3. How can one digitally preserve PDE harvested from the cloud environment so that it can be used for DFR purposes? In other words, how can one preserve the integrity of collected PDE?

This sub-question was aimed at addressing different methods through which the gathered potential evidence can be preserved to maintain its integrity. PDE tends to be fragile, unlike in the conventional investigation processes. This is because, at some instance, a perpetrator or

a colluder might erase or change the contents of digital evidence before it is secured – without leaving a trace. This may in the long run present a digital forensic investigator with a tough assignment.

In **Chapter 8** the researcher highlighted a technique that created a cryptographic hash for the captured evidence before it could be stored in a forensic database. This technique ensured that the collected evidence was retained in its original form. Also, dates and auditable timestamps were collected to provide assurance when a chain of custody had to be established. Therefore, this research sub-question can be answered as follows:

Research findings show that the integrity of the gathered PDE can be maintained by creating cryptographic hashes of the gathered information and providing auditable timestamps of each digital event.

4. Can a software application that was originally applied for malicious purposes be used – without being detected – to capture PDE in a cloud environment? Can this be done in a non-malicious fashion for DFR purposes?

In this sub-question, the researcher wanted to ascertain whether it is possible to use a software application that was initially considered malicious to help conduct DFR in the cloud environment. To answer this research sub-question, the researcher developed a software application with the modified functionality of a botnet that was able to operate in a non-malicious fashion. As seen in **Chapter 9**, a bot client executed in VMs in the cloud was able to perform digital forensic evidence capture. Patterns of the bot client were further changed in an obfuscation process for deterrence purposes, which involved the exploitation of a deliberate vulnerability present in a VM. If the surveillance of the bot client is not deterred in the process, then it might defeat its purpose, since the user should not be aware of the interaction and workings of the NMB. Therefore, this sub-question can be answered as follows:

An obfuscated bot client that operates in a non-malicious fashion can be used to collect useful information that is related to crimes in the cloud environment. The bot client was however obfuscated to deter surveillance.

5. What issues and challenges are encountered when conducting DFR by using a non-malicious software application in the cloud environment? What are the possible high-level solutions?

The aim of this sub-question was to identify the challenges that were encountered while making use of a bot client in the cloud environment. In answering this research question, the researcher was able to identify challenges that affected the implementation of the cloud-based botnet solution from a technical, legal and operational point of view. Additionally, the researcher mapped the identified challenges (explained in detail in **Section 10.6**) with the existing literature on cloud forensic challenges and identified a number of implementation challenges (listed in **Chapter 9**). The researcher was also able to propose a high-level solution for each identified challenge (see **Table 10.2**). Therefore, this research sub-question can be answered as follows:

A number of challenges were identified from a technical, legal and operational standpoint. Furthermore, the researcher was able to propose a high-level solution to each of the identified challenges.

The next section introduces the reader to the CFRaaS implementation challenges.

10.6 CFRaaS Implementation Challenges

This section presents the CFRaaS implementation challenges that the researcher was able to encounter while conducting this research.

10.6.1 Overview

The preceding chapters have discussed the CFRaaS model and the CFRaaS prototype that may help organisations to maximise the potential use of evidence while minimising the cost of conducting DFI at the same time. The CFRaaS model that was presented in Chapter 7 provides generic guidelines that can be accepted and implemented at the same time in multiple cloud environments. The set of requirements highlighted in this model has been used to bridge the gap between the theoretical forensic readiness challenges in the cloud and the technicalities involved when using a software application to conduct forensic readiness in the cloud environment. Moreover, to understand the concepts of conducting digital forensic

investigations in the cloud, it is paramount to comprehend the technology, issues and challenges that are associated with monitoring or capturing electronic information.

Unfortunately, despite there being no forensic readiness standards that are focused on the cloud environment apart from ISO/IEC 27043 that focuses on incident investigation techniques, the CFRaaS model acts as a technical approach that can be used to implement forensic guidelines in the cloud. By mainly relying on collected useful forensic artefacts, file metadata and other credentials that are authenticated as PDE, the researcher is able to forensically prepare for DFIs. Carrying out a Digital Forensic Investigation (DFI) process in the cloud during incident response faces multiple challenges. The primary challenge is how the data of vendors and consumers can be protected. Cloud consumers' data residing in the cloud is basically aggregated in multi-tenant environments and these environments are hardly ever shared by Cloud Service Providers (CSPs).

According to Birk (2011) on a technical aspect of digital forensic investigations in the cloud, the amount of PDE that can be available to a DFI strongly diverges between the CSP and deployment models. This serves as an indicator when compounding the hurdles while trying to achieve Digital Forensic Readiness (DFR) in the cloud environment. The cloud continues to grow enormously with major organisations preferring the usage of Virtual Machines (VMs), which have enabled them to venture into privately owned clouds. Major organisations have moved their applications, systems and data because of the economies of scale and scalability through centrally powerful and effective hosted virtual servers. While this has benefited some organisations by freeing 35 to 50% of operational and infrastructure resources, not enough proactive solutions to mitigate potentially inherent security risks have been enforced (Gartner, 2011), (Wilcox, 2011). Cloud exploits against cloud consumers are major risks that are expected due to the open nature of the cloud. Moreover, the CSPs cannot mitigate this by leveraging the exploding number of users.

In the previous chapters, the researcher has extrapolated that without modifying the functionality and/or infrastructure of existing cloud architectures, a modified form of a botnet acting as a distributed Agent-Based Solution (ABS) could be deployed within the cloud environment to forensically capture PDE for purposes of DFR. Thereafter the captured information was digitally preserved to aid in the reactive process when conducting a DFI.

Discounting that, this section investigates the issues and challenges faced when conducting DFR in the cloud environment as a result of implementing the CFRaaS prototype as shown in Chapter 8. The contribution of this section is presented in four phases. Firstly the researcher discusses the motivation then identifies the challenges in the cloud as a result of CFRaaS prototype implementation. Next, the researcher matches all the challenges to the related work on cloud forensic challenges. Finally, the researcher proposes possible high-level solutions to these issues and challenges from a general, technical and operational point of view.

The rest of the sections are structured as follows: The section begins with a description of a motivation and the scope of exploring challenges in Section 10.6.2 Thereafter, Section 10.6.3 discusses related work. This is then followed by a discussion on the general, technical and operational challenges of the model, as well as the proposed high-level solutions in Section 10.6.4.1, 10.6.4.2 and 10.6.4.3 respectively. After this, Section 10.6.5 gives a discussion.

10.6.2 Motivation for Exploring Challenges

The cloud has not fully adapted to traditional DFI processes because potential evidential data is distributed and there is still a lack of standardised processes. In fact, the complexity of the cloud has given rise to many open issues. These issues include the inability to conduct digital investigation due to the difficulty involved while trying to gain physical access. Another issue is the inability of the Law Enforcement Agencies (LEA) to trace the provenance of a digital object. Additionally, ISO/IEC 27043 international standard highlights the standardised Digital Forensic Readiness Investigation Process (DFRIP), which can be used to conduct DFR in a given organisation without the disruption of any business processes. However ISO/IEC 27043 is not focused on the cloud. Notwithstanding that, using an NMB to conduct DFR in the cloud has been motivated by the fact that, there is no alteration or modification of the functionality and/or infrastructure of existing cloud architecture. Despite that, the scope of this study shows the manner or capability through which DFR can be conducted in the cloud with a focus on SaaS. At the outset of conducting DFR, the CSPs will be assured that the process will save money and time because of lack of having to reprogram the infrastructure time and again.

10.6.3 Related Work on Challenges

In this section, the researcher present work that is somewhat related to the research on challenges and DFR approaches that is being presented in this thesis. To begin with, a survey on information security incident handling for mitigating risks to confidentiality, integrity and availability in the cloud environment by Rahman & Choo (2015) presented different methods of handling incidents in digital forensics and the existing gaps in the cloud environment were identified. Moreover, the researcher was able to identify clouds organisational data, as one of the challenges encountered while handling security incidents and a comparative summary of different international incident handling models was presented. A final study on the same research provides a summary of cloud security challenges. While this research work on cloud forensic challenges was very informative, its major focus was mainly on incident handling mechanisms.

Another paper aimed at addressing cloud forensic technical challenges and solutions by Martini & Choo (2014c) reviewed various prominent technical publications. Research on these publications was aimed at providing conceptual solutions to CSPs with better systems for forensic evidence collection while implementing them in the cloud environment. On the same note, a conceptual framework to integrate DF tools into different ways of developing a cyber-physical cloud that was aimed at helping an organisation to recover from cyber-physical attacks addressed the following factors: Risk management, forensic readiness, incident handling, laws and regulations, hardware and software requirements and industry-specific requirements. The main need for this framework was to point out cloud-specific forensic challenges like multi-jurisdictional, multiple versions and data extraction issues as highlighted by Rahman, Glisson, Yang & Choo (2016).

Research on challenges in digital forensics by Vincze (2015) has identified the following operational challenges: Diversity, scale and cloud resources, digital evidence seizure, privacy, hiring, training and development. In this research, the author categorically identifies the major challenges in DF but also acknowledges that there is still more work to be done on the same. Also, Simou et al., (2014) has addressed major forensic challenges and issues of the cloud from a review perspective and using a model. The researcher has been able to categorise identified challenges to the following stages that are applicable to IaaS, PaaS and SaaS: Identification, preservation-collection, Examination-analysis, presentation and

uncategorised. Also, research by Delpont, Olivier and Kohn (2011) has proposed an isolation of a cloud instance through instance relocation, server farming address relocation, failover and sandboxing in order to prevent contamination, tampering and losing instances of possible digital forensic evidence.

On the same note, after a survey of existing literature on draft NISTIR 8006, the NIST cloud computing forensic science working group (NCC FSWG) documented a list of challenges in cloud computing environments (NISTIR 8006, 2014). In this document, NIST listed 65 challenges related to cloud forensics, based on a normalised formula of four variables. These variables include the following:

- Stakeholders (noun) – this variable identifies the affected stakeholders by the challenge that has been identified. Examples include first responders, investigators and cloud consumers (NISTIR 8006, 2014).
- Action (verb) – This represents the activities that the stakeholder intends to do, for example gaining access, imaging and decrypting (NISTIR 8006, 2014).
- Object – This identifies the specific item on which the action is to be performed, for example data, audit logs, evidence and time stamps (NISTIR 8006, 2014).
- Reason – This refers to the primary challenges that the stakeholder faces so that he can perform the specific action on the object (NISTIR 8006, 2014).

Throughout this research and based on the normalised formulae, NIST specifically identified the general challenges without proposing any solutions.

Whilst these challenges are documented well, the researcher has realised that implementing the CFRaaS prototype in the cloud poses many challenges in respect of which the researcher provide his thoughts on possible high-level solutions later in this research thesis. The next instances of related work do not deal with challenges but present related work on DFR. They are included because they present a DFR approach and we also make use of a DFR approach.

In a Harmonised Digital Forensic Investigation (HDFI) model that was proposed by Valjarevic and Venter (2013), the researcher highlight the need for DFR phases before incident detection. The HDFI is a vital comprehensive digital forensic investigation model

that has already been published as ISO/IEC 27043: 2015 international standard. We are adopting the forensic readiness processes that have been mentioned in the various classes of digital investigation process to aid with the conducting of DFR in the cloud.

A prototype on DFR in the cloud environment developed by Trenwith and Venter (2013) has the following requirements: Identification, collection, transportation, storage and examination. In this prototype, there was a collection of PDE from each virtual machine in the hybrid cloud that was later used as an operating system (OS) Application Programming Interface (API) to conduct a backup. Furthermore, the prototype had a communication channel and implemented encryption, compression and authentication. It shortened the DFI process through data acquisition in a proactive process that involved a remote and a central evidence server – this portrayed the effectiveness of forensic readiness. The study by Trenwith and Venter (2013) portrays a mechanism of conducting DFR in which the emphasis lies on identifying and collecting information. This research attempts to do the same, however, it uses an NMB to accomplish this. On the other hand, according to Cohen (2009), it is important to identify and preserve relevant log files and audit data. In spite of that, Cohen (2009) highlights that all these potential evidence should be linked to the servers used to send, receive, process and store the evidence. This is preferred in situations where PDE might be sought if an incident is detected.

10.6.4 Challenges and Proposed High-Level

Solutions

This section of the research thesis highlights a contribution towards assessing the prevailing issues and challenges when an NMB is used to conduct DFR in the cloud environment. The issues and challenges mentioned in this section emanate from the CFRaaS implementation. Based on the comprehensive literature study provided in Section 10.6.3 of this thesis and the CFRaaS implementation, the researcher is able to identify the existing gaps. The researcher has therefore harmonised the existing gaps between the existing literature and the CFRaaS implementation by identifying issues and challenges and proposing high-level solutions for each challenge. The challenges are classified into three categories: general, technical and operational. Figure 10.1 illustrates the categories of challenges in a hierarchical structure. A more detailed explanation of the challenges follows for each category by highlighting the

sub-challenges and proposing a high-level solution. Each of these challenges is explained in a separate section to follow.

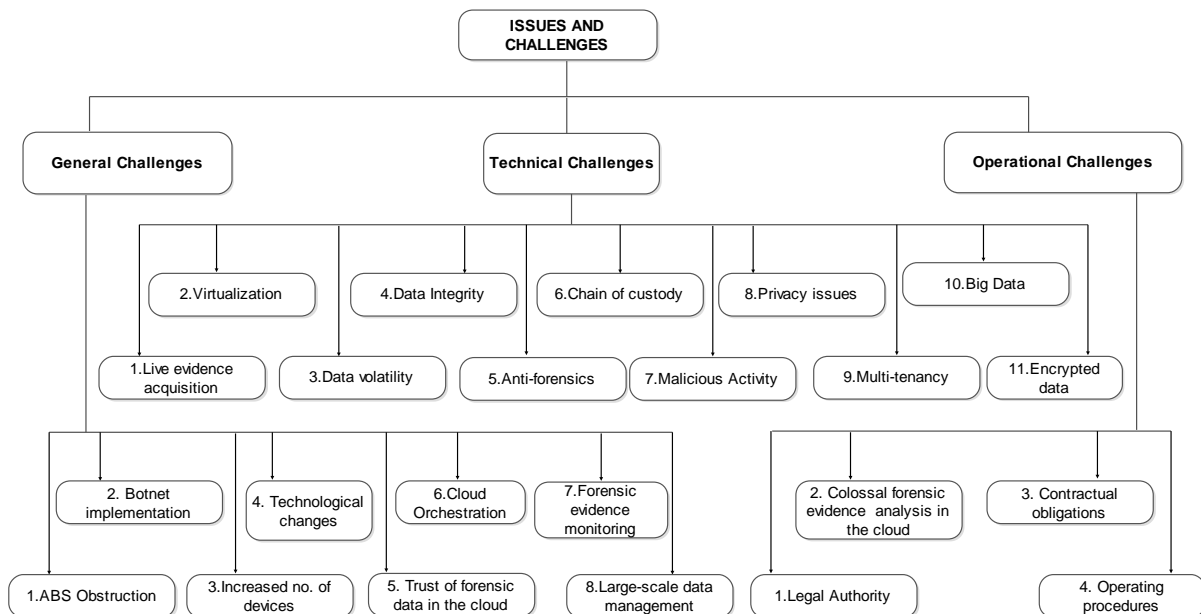


Figure 10.1 A Hierarchical Classification of CFRaaS Implementation Issues and Challenges

10.6.4.1 General Challenges

In this section the researcher provides a brief discussion on the general challenges by breaking down the main challenges into sub-challenges and then proposing a high-level solution to each of the challenges. The challenges arise as a result of implementing CFRaaS using an NMB. The proposed high-level solutions are also discussed in each sub-section. A summary of this discussion appears later in Table 10.2.

10.6.4.1.1 NMB Disinfection

The existence of an NMB can be affected by the availability of disinfection strategies. As a result, these strategies are able to remove the forensic clients from their functionalities. Agarwal (2010) in his research highlighted that in random mitigation strategies that disrupt communication, there exists a method whereby a set of clients are randomly removed from their functionalities. This method offers the possibility of an attack in the form of infiltration. The NMB is infiltrated by many or big fake malicious programs with a view to disrupting communication. Obfuscating the NMB solution from this activity according to KEBANDE and VENTER (2015) might prevent the possible infiltration and takedown of the non-malicious activity.

10.6.4.1.2 NMB Implementation

The implementation of an NMB involves two factors. Firstly, the botnet has originally been perceived to be of a devious nature and to have bad connotations. Secondly the NMB (modified functionalities of a botnet) collects useful information. Moreover, it is considered malicious because it captures information illegally. However, although the implemented botnet operates in a non-malicious fashion, the presence within the network of possible multiple malicious activities that infiltrate the NMB solution may disrupt communication during a DFR approach. Furthermore, the botnets creators from time to time keep modifying the botnet architecture with the intent of making it more resilient. This has made it very difficult for researchers to implement the botnet with the existing architecture. It is essential to keep up with the current network architectures as they support a wide range of cloud services and architectures that sustain a broad range of emerging technologies and services.

10.6.4.1.3 Increased Number of Devices

The increase in distributed computing devices makes it a challenge to monitor the origin of data objects when an NMB gathers digital information. A potential issue uncovered with the presence of these devices is when data is moving to the cloud. When this happens, direct control of that data is lost. A publication by NIST 800-37 highlights the fact that “organisations are accountable for the risk incurred by use of services provided by external providers and address this risk by implementing compensating controls”(NIST 800-37, 2010).

Since most devices have gone mobile and ad hoc, locating a specific device is not easy and the distributed nature of most devices poses a real challenge. Furthermore, the increase in device numbers means an increase in usage, as well as an increase in volume, velocity and variety of data in the cloud, which makes evidence segregation hard. A proper cloud evidence management system for DFR purposes helps in the extraction, mapping and segregation of PDE.

10.6.4.1.4 Technological Changes

The way in which the cloud environment operates is distinctly affected by the implementation of new and upcoming NMB and network technologies. The cloud is embraced as economical and has replaced the act of hosting own resources due to the new technologies that have proved to be efficient, cost effective and flexible (Biggs, 2009). DFR in the cloud is affected by current operational changes and on-demand solutions. This is due to the fact that the current DFIP in the cloud do not match the existing cloud computing characteristics, i.e. the cloud is not adapting to DF processes easily.

These technological changes tend to change the existing cloud-computing architectures in some instances. Furthermore, the change in network interoperability brings about changes in the NMB operation. In solving these predicaments, the new evidence-capturing forensic tools should be at par with the technological changes so that these technological concepts do not affect the confidentiality or authenticity of the forensic evidence being transmitted.

10.6.4.1.5 Trust of Forensic Data in the Cloud

Conducting DFR in the cloud environment involves users' data gathered and retained by an NMB. For this reason, the chain of trust in the integrity and confidentiality of the data stored in the cloud is not endorsed fully by users, since the cloud is an untrusted execution environment. In this context, trust (as presented by Pearson (2013) from a cross disciplinary view) is a psychological state that comprises the intention to accept vulnerability based on positive expectations of the intentions or behaviour of another). The process of capturing PDE is therefore faced by the issue of protecting critical components such as the hypervisor against run-time attacks. A hypervisor in this context is a piece of software that runs or operates the virtual machines. This is the most important key function in securing the PDE collected. To mitigate this challenge, a chain of trust should be built on the identities, cloud infrastructures and data that are stored in the cloud. This can be done through hardening the hypervisor and VMs so that the risk of run-time attacks on the hypervisors can be reduced. It will also ensure that the privacy and integrity of PDE is maintained.

10.6.4.1.6 Large-Scale Data Management

There is a plethora of data when gathering PDE from the cloud environment in a non-

malicious way by means of an NMB. This is due to increased multi-tenancy and lack of effective scalable data management systems that can manage large-scale PDE. According to Abadi (2009), managing the vast volumes of data in the cloud is more difficult and time consuming due to the lack of a timely and efficient back-end data management system in the cloud. This is mainly because data in the cloud comes in huge volumes in heterogeneous environments and it is often replicated across large geographical distances making data management a challenge. Hence, the cloud has to manage a number of resources with different patterns that range from digital maps, images, large files, structured and unstructured data. This is a very tedious exercise of mapping the actual potential evidence that can be admissible in a court of law. Furthermore, large-scale data requires many server instances to speed up data processing.

An ideal cloud potential forensic evidence processing system that is based on MapReduce that can manage the increased scale of generated data in the cloud brings about effectiveness in PDE management (Kebande & Venter, 2015) . This makes evidence aggregation and correlation easy through evidence characterisation. Furthermore, this will reduce the need for more server instances required to process more data.

10.6.4.1.7 Forensic Evidence Monitoring

Normally digital forensic evidence is aggregated in a multi-tenant environment. This environment is complex for it consists of numerous applications, VMs and hardware. The data is also organised in distributed and scalable fashion. The process of monitoring the cloud-based NMB solution is not very effective because complexity implies an increase of virtual server triggers and an increase of data load, which eventually leads to component overload. If the captured PDE is compromised, evidence may well be contaminated before a digital investigation can be performed. Since evidence monitoring is done on per user basis, a given cloud may have thousands and millions of monitoring tasks that might not be very effective in the long run. Moreover, data centres run a huge number of cloud services, and this may lead to an overlap in the metric being monitored.

By enforcing Monitoring-as-a-Service (MaaS) enforcement at the application level of the prototype, there can be an opportunity to improve the efficiency of the evidence being monitored. In spite of that, we should ensure that there is no storage of unnecessary forensic

evidence. This means based on potential evidence that is collected from different sources, it will be assessed to see if it is relevant or non-relevant.

10.6.4.2 Technical Challenges

This section gives an overview of technical challenges encountered when conducting DFR in the cloud. The challenges will be highlighted mainly from a technical point of view and thereafter high-level solutions will be proposed. A typical approach of cloud forensic readiness begins with the gathering and retention of PDE using an NMB as discussed in Chapter 9. The PDE collected for DFR purposes might result from hypervisor error logs, network logs, activity logs, virtual images, application logs, database activities, monitoring, cloud carrier logs, etc. All these comprise forensic logs, service artefacts and monitored data that are contained in a cloud forensic storage database. According to Ananthanarayanan et al., (2009) the key infrastructure element of the cloud stack is a storage layer that is able to support large petabytes of data. A number of technical challenges arise as a result of events and log cycles and will be discussed next.

10.6.4.2.1 Live Evidence Acquisition

Since the cloud is distributed and volatile, there is no easy way of logically acquiring live PDE in the cloud environment for purposes of DFR after an NMB solution has gathered and retained critical digital information related to crimes. According to Casey (2002) acquisition of digital artefacts is the initial step in the forensic process. In this context, if a VM is shut down by an adversary, it becomes impossible to collect forensic evidence because of the VM's volatile nature. Moreover, a challenge of verification arises due to changes in the data. Traditionally, data changes as the systems keep running, which brings about a difference in the evidence collected and the evidence acquired.

Live evidence depends on the system that the suspect is using, but Carrier (2006) argues that a suspect's system cannot be trusted. In spite of that, Birk (2011) highlights that in overcoming live acquisition, the CSPs can make access potential evidence read-only through an API. On the other hand, proper verification of the integrity of collected evidence should be enforced.

10.6.4.2.2 *Virtualisation*

According to the draft NIST 8006, virtualisation is a simulation of the software or hardware upon which other software's run. A majority of network defence systems are based on physical networks and most data centres support static virtualisation. When the NMB is installed in the virtual instances, monitoring the security threats of VMs becomes difficult as a result. Hypervisors consequently remain vulnerable to attacks and in order to prevent these, the CSPs must provide a perimeter security as a firewall. This is done to isolate the virtual resource spaces from further potential attacks.

10.6.4.2.3 *Data Volatility*

Once the VM is shut down, PDE is lost. This is because all the data residing in the VM is volatile, making it a difficult task to locate the whereabouts of data. In nature, volatile data tends not to survive when there is power failure. To mitigate this challenge, the provenance of digital data must be proved or accounted for. Zawoad and Hasan (2013) highlight the fact that evidence should be stored in a persistent database so that if an adversary attempts to shut down a VM, evidence can still be gathered. Additionally, Dykstra and Sherman (2013) suggest that a cloud management that uses an IaaS model can enable evidence gathering when the VM is terminated.

10.6.4.2.4 *Data Integrity*

An NMB solution is able to collect vital data as PDE. On the same note, it is not easy to guarantee the perfection and correctness of essential and critical data that exist as PDE in the cloud environment because PDE streams in from different points at different times. Consequently, it is also not easy to differentiate at different levels which kind of data is essential and which one is not. The integrity of data should be enforced to help to prove in a court of law that the evidence being presented is the same evidence that was captured forensically. This also will increase the chances of admissibility on PDE. In this context, this is a big challenge for DF investigators and LEAs, because even if they manage to acquire the necessary evidence, it might not be an easy task to verify the integrity of data and its origin.

According to Grispos et al., (2012) data stored in the cloud should be subjected to hashing, which enables integrity checking. Additionally, ISO/IEC 27037, 10118-2 and ISO/IEC

27043 stresses on the need to avoid cases of intentional and un-intentional evidence deletion, evidential integrity through using hash functions of all the bits in each media that contain PDE (ISO/IEC 27043:2015; ISO/IEC 27037: 2012).

10.6.4.2.5 Anti-Forensics

The draft NIST 8006 highlights anti-forensics as a set of techniques that are used specifically to prevent or mislead forensic analysts. In this context, these tools frustrate the process of achieving DFR. NIST further categorises anti-forensics as a process of obfuscation and hiding data, as well as the use of malicious codes with the intent of compromising the integrity of PDE. The use of anti-forensics reduces the quality of PDE deliberately by interfering with pre-incident analysis of PDE.

According to Jahankhani and Beqiri (2010) to mitigate anti-forensics; computer forensic tools should be improved through improvement of signature analysis and time-stamp analysis. All forensic tools should be hardened because forensic tools like Encase and FTKs do not check for signatures.

10.6.4.2.6 PDE Handling

Handling and managing the collection of evidence is a daunting task because of the distributed nature of resources and cross-cutting jurisdictional issues. PDE seizures, control, transfer and trails have to be documented systematically according to accepted cross-jurisdictional standards, procedures and technology. Vacca (2005) highlights the fact that there has to be a roadmap showing how evidence is collected, analysed and preserved in order for PDE to qualify for admission in a court of law. In the preparation of DFR, the policies regarding retention, collection, planning and evidence acquisition must be documented chronologically. When this approach is not followed, PDE loses quality and may not be admissible. Furthermore, the good practices of the UK's Association of Chief Police Officers (ACPO, 2007) describe documentation as a way in which evidence was managed before being presented in a court of law.

10.6.4.2.7 Malicious Activity

A cloud-computing platform is a ready target for malicious activity. Protecting an NMB in the cloud from adversaries is necessitated by the existence of numerous threats and attacks

because of the pervasiveness of the cloud. VMs are bound to be attacked and this brings a possibility of intentional data tampering, which in the long run makes the cloud platform vulnerable to attacks. The aspect introducing Intrusion Detection Systems (IDS) to monitor the entire network traffic and suspicious behaviour protects an NMB from intentional attacks. KEBANDE and VENTER (2014) uncovered a mechanism that was able to detect how malicious botnets or potential malicious activity can be detected in the cloud, namely the Artificial Immune System (AIS).

This was based on the malicious pattern that the botnet traffic uses. According to FLOOD and KEANE (2012), a system should be trained on the possibility of responding to the user interaction. Enforcement of these criteria is a key to protection in the cloud and performing a comprehensive assessment regularly. On the other hand, CLAYCOMB and NICOLL (2012) argue that “transparency into overall information security and management practices, determining security breach notification process and encrypting data in the cloud” are some of the key solutions to the challenge of the presence of malicious activity in the cloud.

10.6.4.2.8 Privacy Issues

Gathering and retaining PDE from the cloud by using an NMB poses a huge challenge to cloud consumers, because PDE collected for DFR purposes is not achieved through deliberate planning. Capturing user information within the cloud may have jurisdictional issues because electronic transactions that can lead to disclosure of personal information. This may be treated as a contravention of an individual’s right to privacy. A review of the legal perspective on admissibility of DE shows that although the requirements vary across different jurisdictions, some legislation provides exemptions to allow interference with privacy of information, provided that it is a matter of national security or a research activity. These Acts include the Electronic Communications Privacy Act (ECPA), Act of 1986 of the USA; the UK’s Association of Chief Police Officers (ACPO) Good Practice Guide for Digital Evidence; the Electronic Communication and Transactions (ECT) Act of South Africa; the Protection of Personal Information (POPI) Act of South Africa and the Stored Communications Act (SCA) of the USA.

Moreover, the cases from the United States of America, which include the presentation of digital evidence, are treated under Rule 702 of the federal rules of evidence. This is the rule

of evidence that says “If scientific, technical, or other specialised knowledge will assist the trier of fact to understand the evidence or to determine a fact in issue, a witness qualified as an expert by knowledge, skill, experience, training, or education, may testify thereto in the form of an opinion or otherwise”. The Daubert (1993) case applied these rules.

10.6.4.2.9 Multi-tenancy

A single instance of a software application may have many tenants where an NMB is deployed. However, when multiple VMs tend to share the same physical infrastructure in the cloud environment, computing gets distributed. As a result, potential vulnerabilities arise from the VM technologies used by cloud vendors, which eventually results to a number of security issues while gathering PDE at VM level. A higher chance of evidence contamination is likely because evidence come from different parts of the multi-tenant environment; hence it is left to the DF investigator to prove whether the evidence is strongly associated with the malicious user, while the privacy of other tenants is preserved.

This is because data is shared among multiple computers within multiple locations with numerous tenants and with numerous forensic evidence. Additionally, the process of proving whether the evidence has a relationship with the user in this environment is faced with timestamp inconsistencies and difficulty in assessing forensic evidence. According to Zawoad and Hassan (2013), the DF investigator must prove two instances: firstly, that the forensic evidence is not mingled and secondly that the privacy of other users is preserved.

10.6.4.2.10 Big Data

Data gathered by an NMB from the cloud comes in vast volumes. This data, which is characterised by velocity, variety, complexity and volume, is a threat when DFR is to be achieved. Big data remains unstructured, and performing pre-incident analysis on this data is a challenge because data comes from different sources. Big data is a focal challenge and it affects almost every activity being conducted within the cloud environment. During a DFR approach, it mainly affects data security: when sensitive, critical data related to crime and personal information tends to exist within big data, it becomes a big risk.

As a result of increased devices and the large exploding amount of terabytes of digital data that is seized as forensic evidence, Quick & Choo (2014e); Quick & Choo (2014f) have

highlighted that the effects of this leads to large backlogs, increased the size of devices per case, increased number of cases that deals with digital evidence and an increase in the size of data. The researcher in this context have gone ahead to propose a digital forensic data reduction and data mining framework that is able to reduce the storage demands while providing a capability of conducting a review on subset data for purposes of intelligence analysis, archival, research and historical review purposes.

On the other hand, big data comes in all shapes and sizes, and it contains very heavy schemas that further cause pre-incident analysis problems. Another issue is the need to scale data centres rapidly and the cost of maintaining the Relational Database Management System (RDBMS), which is very high. The researcher has proposed a functional architecture that is able to do analysis of large-scale PDE in the cloud using MapReduce (Kebande & venter, 2015). Through this computing is transferred to low commodity cluster computers by mapping and reducing the colossal amount of data using a reducer.

10.6.4.2.11 Encrypted Data

Adding an additional layer of security to data may make data unusable because encryption is a strong security mechanism for data protection. This involves making data meaningless to unauthorised users or adversaries. Encryption is embraced because cloud users want protection, compliance and control of their stored data. From a DFR perspective, if encrypted PDE is collected by an NMB, it becomes a task to decrypt the same, due to key management constraints. A major issue regarding data moving to the cloud is how it is handled if there is a breach, as well as the status of data at rest and data in motion. For the purposes of effectiveness in conducting DFR, a data security plan should be laid out in accordance with which data should be encrypted.

Data in motion and data at rest should be encrypted because data protection is a critical aspect of security. Finally, key management should be performed by the CSPs as encryption providers. Logging as forensic evidence collection policies should be controlled by the CSPs and at the same time they need to enforce measures to restrict access to sensitive and critical management tools.

Table 10.2 Issues and challenges faced in implementing an NMB in the cloud for purposes of digital forensic readiness

	Category and Identified Challenges	Related work on Identified Challenges	Proposed High-Level Solutions
GENERAL CHALLENGES			
1	NMB Obstruction	xxx	Obfuscate the NMB cloud solution
2	Implementation	xxx	Use current network architecture
3	Increased no. of devices	(Quick & Choo, 2012)	Introduce a cloud evidence management system
4	Technological changes	(Draft NISTIR 8006) (Palmer, 2001)	Introduce new evidence-capturing tool, tools at par with new changes
5	Trust	(Draft NISTIR 8006) (Dykstra & Sherman, 2012) (Simou et al.,2014d) (Delpont, Kohn & Olivier(2011)(Daryabar, Dehghantanga & Udzir, 2013)	Built trust based on identities, infrastructure and information
6	Data Management	(Martini & Choo, 2013)(Quick and Choo, 2014e)	Develop a cloud forensic readiness management system, it will reduce the need for more server instances
7	NMB Monitoring	(Martini & CHoo, 2012)(Quick & Choo, 2013)(Hooper,Martini & Choo, 2013d)(Martini & Choo, 2014b)(Marty, 2011)(Dykstra & Sherman, 2012) (Martini & Choo, 2014c) (Vincze, 2015) (Simou et al.,2014) (Alqahtany, Clarke, Furnell & Reich, 2015) (Damshenas et al.,2012) (Meyer & Stander, 2012) (Sandez, 2015)	Monitor as-a-service at application level
TECHNICAL CHALLENGES			
1	Live evidence acquisition	(Draft NISTIR 8006) (Martini & Choo, 2014c) (Sibiya, Venter & Fogwill, 2012) (Alqahtany, Clarke, Furnell & Reich, 2015) (Sandez, 2015), Grispos et al (2011)	Access evidence in read-only through an API
2	Virtualisation	(Draft NISTIR 8006) (Quick & Choo,2013a) (Quick & Choo,2013a) (Martini & Choo,2013) (Martini & Choo, 2014b) (Alqahtany, Clarke, Furnell & Reich, 2015)	Provide perimeter security as a firewall
3	Volatile data	(Draft NISTIR 8006) (Quick & Choo,2013a) (Quick & Choo,2013b) (Vincze,2015) (Delpont, Kohn & Olivier(2011) (Alqahtany, Clarke, Furnell & Reich, 2015) (Damshenas et al.,2012) Azfar et al (2016)	Store evidence in a non-persistent form

4	Integrity of collected evidence	(Draft NISTIR 8006) Quick & Choo, 2013) (Delpport, Kohn & Olivier(2011) (Alqahtany, Clarke, Furnell & Reich, 2015)	Subject data in cloud to hashing and encrypt data at rest and in motion
5	Anti-forensics	(Draft NISTIR 8006)	Harden forensic tools by improving signature analysis and time-stamp analysis
6	Potential evidence handling	(Draft NISTIR 8006) (Martini & Choo, 2012) (Quick & Choo, 2013a)(Martini & Choo, 2013)(Marty,2011)	Preserve document evidence as a way of management
7	Malicious activity	(Draft NISTIR 8006)	Determine security breach notification and encrypt data in the cloud
8	Privacy issues	(Draft NISTIR 8006))(Hooper,Martini & Choo, 2013d) (Delpport, Kohn & Olivier, 2011)	Handle the issue in accordance with requirements of jurisdictional acts
9	Multi-tenancy	(Draft NISTIR 8006))(Hooper, Martini & Choo, 2013d) (Delpport, Kohn & Olivier,2011)	Get a digital forensic investigator to prove all instances for consistency
10	Big-data	(Delpport, Kohn & Olivier, 2011)	Use MapReduce for analysis of big data
11	Encrypted data	(Draft NISTIR 8006) (Quick & Choo, 2013c)(Martini & Choo,2013) (Delpport, Kohn & Olivier, 2011) (Sibiya, Venter & Fogwill, 2012)	Get CSPs to handle key management and to have control of logging policies
OPERATIONAL CHALLENGES			
1	Legal authority	(Draft NISTIR 8006) (Martini & Choo, 2012) (Hooper, Martini & Choo,2013)(Dykstra & Sherman, 2012) (Ab Rahman et al., 2015)(Meyer & Stander, 2012)(Sibiya, Venter & Fogwill, 2012) (Van Eecke, 2011)	Formulate international legislation to implement global law
2	Colossal data analysis	(Martini & Choo, 2012) (Quick & Choo, 2013) (Martini & Choo,2013) (Alqahtany, Clarke, Furnell & Reich, 2015) (Damshenas et al,2012)	Implement a cloud-based analysis management system that uses MapReduce
3	Contractual obligations	(Hooper,Martini & Choo, 2013d) (Marty(2011) (Ab Rahman & Choo,2015) (Ab Rahman et al.,2016) (Delpport, Kohn & Olivier(2011)	Make it a requirement that evidence generated in different layers be accessible to different stakeholders of the system
4	Standard Operating procedures	(Hooper,Martini & Choo, 2013d)] (Ab Rahman et al.,2016)	Adopt the proposed readiness standards by ISO/IEC 27043

10.6.4.3 Operational Challenges

This section first discusses operational challenges and then proposes high-level challenges when conducting DFR in the cloud. Operational challenges consist of the following sub-challenges: legal authority, colossal data analysis, contractual obligations and operating procedures.

10.6.4.3.1 Legal Authority

The issue of legal admissibility of DE is encountered when gathering PDE across different jurisdictions. Issues arising from cross-cutting jurisdictions hold a challenge when digital evidence (DE) is collected as part of the DFR process. DE that is stored in multiple jurisdictions cannot be accessed because what is considered legal in one country might be illegal in another. This is because of the lack of proper Internet laws and consistent legislation, which may cause governments to be unwilling to cooperate and thus lead to delays in warrants being served (hence delays in prosecution).

On the concept of legal authority, the Security Techniques Advisory Group (STAG) on lawful interception has highlighted on its ETSI Technical Report (ETR) that a provision on guidance is given to service providers and network operators with lawful interception of telecommunications (TC-STAG, 1996). Based on this the general requirements that have been highlighted recapitulates that the CSP will intercept, retrieve and store content of communication the entire period and one can monitor or choose to permanently record the results arising from interception. However in order for PDE to be admissible it must satisfy the legal requirements of the particular jurisdiction because the legal requirements for admissibility of digital evidence vary across different jurisdictions.

According to Cohen (2009), most communications are interstate and international when it comes to jurisdiction and cooperative agreements should solve evidence-giving problems. On the other hand, Biggs and Vidalis (2009) suggest that in overcoming cross-border challenges, an international unity should be formed to introduce international legislation that can implement globally accepted legislation.

10.6.4.3.2 Colossal Forensic Evidence Analysis in the Cloud

The distributed NMB is very versatile, as it captures huge amounts of data with high velocity, volume and complexity. When performing DFR pre-incident analysis and examination, performing analysis of incoming traffic is a challenge because of the mentioned characteristics. In this research thesis, researcher has proposed cloud storage with MapReduce, as it can process the collected data by performing computations of the large-scale DE through trace and analysis conducted in parallel (Kebande & Venter, 2015). Chen et al., (2013) also proposed a cloud-based forensic analysis managing system that could analyse the traffic in the cloud.

10.6.4.3.3 Contractual Obligations

Normally, contractual obligations exist between the CSPs and the cloud consumers. During PDE capturing, a dispute is encountered on how the captured logs are to be administered by the client and CSPs. Furthermore, any change in the implementation of Service Level Agreements (SLAs) may breach the pact between the parties, which might lead to conflict. Mainly, this involves a situation in which CSPs have full control of forensic evidence in the cloud, where customers do not have any control. According to Zawoad & Hassan (2013); Riley et al.,(2011) , it is a requirement that PDE that is generated in different layers must be accessible to different stakeholders of the system.

10.6.4.3.4 Standard Operating Procedures (SOPs)

There is lack of proper Standard Operating Procedures (SOPs) in the cloud environment on how DFR is conducted. It is difficult to develop tools that can conduct DFR because of cross-platform developments and a lack of standardised infrastructures. According to Aminnezhad et al., (2013) conducting DFR processes in the cloud can be extremely challenging when it includes thousands of virtual machines. It may also lead to disruption of service to other users. ISO/IEC 27403 was proposed as a standard for security techniques, incident investigation principles and processes. This standard also defines the DFR process as a process that can be conducted before incident detection. Additionally,

ISO/IEC 27403: 2015 presents the main aim of DFR in an organisation as “to maximise the potential use of digital evidence while minimising the cost and preserving the level of information security systems within the organisation (ISO/IEC 27043, 2015). The next section gives a discussion.

10.6.5 Discussion

In this research, the inadaptability of DF in the cloud has revealed the challenges faced when CFRaaS prototype is implemented. (See the CFRaaS model as discussed in Chapter 7 and CFRaaS Prototype in chapter 9.) The chapter reports on research that was based on digital forensic gathering and retention mechanisms for purposes of DFR. Based on the existing literature the researcher conducted a literature review of the research conducted by other researchers on cloud forensic challenges (Table 10.2) and other well-documented research papers (Quick & Choo, 2012); (Quick & Choo, 2013a); (Hooper et al.,2013d); (Chung et al.,2012); (Marty,2011); (Dykstra & Sherman, 2012); (Rahman, 2015).

Not only was the researcher able to identify challenges as a result of implementing DFR in the cloud by means of an NMB, but the researcher was able to identify other challenges from the existing literatures that are also attended to in this research study. Based on the current study, the researcher has been able to make a contribution in respect of all the challenges faced in the cloud by proposing high-level solutions. As a result of a review of the related work from the researcher and the challenges arising from the NMB implementation, the researcher managed to list all these challenges in a summarised format in 10.2 for possible comparison and contrasting. Challenges that appeared in both the researcher and the model were represented with their respective references, while challenges emanating from the model but not in the related work were represented using (XXXX). Furthermore, the specific high-level solution for each challenge was summarised in the fourth column of Table 10.2. It is worth noting that in the researchers’ opinion, the proposed high-level solutions may be enhanced as a result of further research in this context.

If the recommended solutions in this research are adopted fully, it is the researchers' opinion that this could significantly facilitate effective proactive processes of pre-incident detection in the cloud environment as is highlighted in ISO/IEC 27403:2015 standard. Digital information retention, correlation and management constitute the most comprehensive way of conducting DFR in the cloud without interfering or modifying the cloud architecture. This can only be possible if the retained potential evidence is correlated, manipulated and managed outside the cloud.

If we explore the challenges from the stakeholders' perspective, the study significantly points out a number of important counter-measures that can help the stakeholders to understand the impacts of DF challenges with respect to future technologies in DFR approaches that are implemented for organisations. This is a big concern to the DF community, especially to the LEAs and DF investigators.

From a legal point of view, the CFRaaS prototype's use of an NMB in capturing user information might interfere with an individual's privacy. This was highlighted by the Regulation of Interception Communications and Provision of Communication-Related Information Act (RICA), 70 of 2002. However, Section 6 (2) of the RICA is a provision that states that "interception can be made for reasons of investigating unauthorised use of that communication system". This can only take place if the investigator has consent from law enforcement authorities (RICA, 2002).

Finally, possible applicability of the aforementioned procedures can be enforced if the CSPs are entrusted to collect evidential information in a forensic readiness process. For each of the contributions mentioned above, the researcher believe that DFR as a process will be effective and that the cloud model will still be able to offer state-of-the-art services.

Having looked at an evaluation of the research questions, the chapter is concluded in the next section.

10.7 Conclusion

In this chapter, the researcher critically evaluated the proposed CFRaaS model that was discussed in **Chapter 7** and the proposed CFRaaS prototype that was discussed in **Chapters 8** and **9**. This was followed by an evaluation of the research objectives, and a discussion on the extent to which the main research question and the sub-questions had been answered. Additionally, the researcher has introduced the CFRaaS implementation issues and challenges that are faced when trying to achieve DFR in the cloud environment. The researcher has proposed a contribution by assessing the possible solutions from a general, technical and operational point of view. Furthermore, the researcher has proposed a high-level solution for each of the contributions. The researcher also compared encountered challenges with other challenges based on existing literature.

The chapter that concludes this research thesis follows next.

Chapter 11: Conclusion

11.1 Introduction

This chapter concludes this research thesis and provides suggestions for future work. An increase in cloud security incidents and cyber-related attacks has changed the pattern of cyber-crime, hence there is need for the forensic community to adopt or develop effective digital investigation approaches that can help to combat cyber-crime. The lack of effective proactive approaches in the cloud environment has caused an increase in security incidents.

Based on this premise, a Cloud Forensic Readiness-as-a-Service (CFRaaS) process model that has Digital Forensic Investigation (DFI) capabilities was proposed in **Chapter 7**. The proposed model was designed to gather Potential Digital Evidence (PDE) from a cloud environment as a way of planning and preparing for potential security incidents. In addition, the proposed model employs a botnet that was initially considered malicious, but that now operates in this context in a non-malicious fashion due to modified functionalities. The CFRaaS prototype that was designed and implemented in **Chapters 8** and **9** furthermore demonstrated a proof of concept of the proposed model.

The remainder of this chapter is organised as follows: The novelty of the proposed concepts is discussed in Section 11.2, future research work is discussed in Section 11.3, and the chapter is concluded in Section 11.4.

11.2 Discussion on Novel Contributions

A number of novel contributions were presented in this research thesis. It is worth noting that the identified contributions aimed to address the problem stated in this research thesis. To the best of the researcher's knowledge, the proposal of a model to achieve DFR in the cloud environment is a novel contribution. At the time of writing this research thesis, a DFR model with a focus on the cloud was not yet available and there was no literature that linked the cloud forensic readiness model aspect to the cloud.

In order to investigate the existence of a potential security incident in the cloud environment, there should exist a reliable digital forensic tool and processes implemented in accordance with the law. The researcher highlighted how the inadaptability of Digital Forensic (DF) processes in the cloud is a challenge that needs a lasting solution. According to Sherman (2008), it is important for law enforcement agencies to be prepared for challenges that deal mainly with the acquisition of digital data. Therefore, the researcher proposed that the best way of conducting DFR in the cloud environment would be through the collection of PDE based on ISO/IEC 27043: 2015 guidelines.

The next novel concept introduced in this research thesis is the use of a botnet that was initially considered malicious for positive purposes. Botnets are known to be hostile information-stealing applications. However, the researcher proposed a novel concept that allows a modified botnet's functionalities to be used as a service in the cloud environment. This was achieved by sending a bot client to infect VMs in the cloud in order to gather, preserve and store digital forensic information. To the best of the researcher's knowledge, this was still a novel concept at the time of writing this research thesis.

A third novel contribution by this research is the idea of the CFRaaS model. This model provides a mechanism for implementing the prototype, by using a non-malicious botnet in a Software-as-a-Service (SaaS) fashion within the cloud environment to collect digital information for DFR purposes. This allows for collaborative DFI processes with competent legal bodies across diverse jurisdictions and happens through a standardised implementation of DFR processes mentioned in the ISO/IEC 27043: 2015. Although the ISO/IEC 27043: 2015 is not focused in the cloud environment, the CFRaaS model complies with its readiness processes and even provides more detailed processes that focus on the cloud environment.

The next section presents work that has been considered as future work.

11.3 Future Work

The research objectives that were identified in this research thesis were achieved based on the propositions presented in preceding sections. However, the researcher was able to identify a number of issues and challenges or shortcomings, and hence managed to come up with suggestions that can be considered as future work.

In this research thesis, the achievement of DFR is limited to a simulated cloud environment as opposed to a fully-fledged cloud environment. This is because the researcher perceived that the proposed model might be very sensitive to the local laws of a given jurisdiction. Since a variety of constitutional and statutory provisions apply regarding the acquisition of digital forensic evidence, the researcher suggested the following as topics of research to be conducted in future.

The current research focused on gathering digital information that may be related to potential crimes; however, due to the proliferation of digital devices, one may want to know how to deal with the combinatorial explosion of big data. Based on the arguments that have been put forward throughout this research thesis, this aspect presents research worth exploring. It is suggested that a further extension of the prototype should investigate ways to locally cache big data at remote locations (i.e. remotely clustered locations dedicated to big data forensics) and techniques for forensically analysing the traffic in real time, or near real time, to enable more efficient incident detection.

Not all distinctive attributes of the CFRaaS model were implemented in the prototype. For example, not all the digital information in the cloud environment could be captured, due to complexity, large volumes and the velocity at which the data was streaming in. The digital information that was captured was therefore limited to keystrokes, RAM usage and CPU usage, IP addresses and timestamps. These elements were used as a podium for testing the objectives of this research study. It would therefore be important to gather other relevant digital forensic information like complete VM images, hypervisor error logs, network logs and database logs. Moreover, incident detection, event reconstruction and incident response procedure phases – all of which are post-event

response phases – were not implemented, but were recommended as future work, already in **Chapter 8** of this research thesis. Hence the CFRaaS prototype can in future be improved to handle all these aspects.

At the time of preparing this research thesis, there existed no DFR standards that had their focus on the cloud environment. It is the researcher's opinion that more research should be conducted so that the proposed CFRaaS model can be refined further. Additionally, it would be realistic to improve the methodologies and techniques used in developing the prototype with a view to its possible standardisation. Inasmuch as the CFRaaS prototype could prove the concept based on the CFRaaS model, the researcher believes that more research, centred in a technical organisational setting, might further highlight the feasibility of the CFRaaS prototype.

Nevertheless, much work remains to be done on the cloud challenges. As the cloud continues to hold sway, numerous technical, legal and operational challenges are being experienced. The researcher was able to identify a number of challenges in this study and therefore concentrated on proposing only high-level solutions (see **Section 10.6**). It is therefore suggested that technical, legal and operational solutions should be pursued that will ensure that the legal aspects of cloud computing keep pace with the advances in technology.

Lastly, the researcher encouraged a futuristic evaluation of the proposed model, as it would provide an assessment that would help enforce operational tests that may end up providing confidence, based on the performance of the model. Further improvements will allow the model to adapt easily to various scenarios. It is the researcher's opinion that if extensive research on the suggested future work is conducted, then DFR processes in the cloud environment will be more acceptable to the forensic community.

In the next section, the researcher draws a final conclusion on the value and contributions of this research thesis.

11.4 Final Conclusion

The research that was presented in this research thesis provides significant and novel contributions on how DFR can be achieved in the cloud environment and how it can facilitate a post-event response process that will limit the costs and time needed to conduct a DFI. These conclusions can be applied when any organisation wants to establish a proactive approach towards digital forensics to help prepare for security incidents in the cloud. Based on his evaluation of the proposed approach, the researcher understands that accumulating potential evidence increases the effectiveness of an investigation when an incident is detected. The researcher trusts that the digital forensic experts, policy makers, legal practitioners, law enforcement agencies and the forensic community as a whole will adopt the findings of this research to maximise the use of digital evidence and save time and money needed during the reactive process.

BIBLIOGRAPHY

- American Academy of Forensic Sciences (2016). Proceedings of the Academy of Forensic Sciences 68th Annual Scientific Meeting. <http://www.aafs.org/wp-content/uploads/2016Proceedings.pdf> [Accessed on July 4th 2016].
- Abadi, D. J., (2009). Data management in the cloud: limitations and opportunities. *IEEE Data Eng. Bull.*, 32(1), 3-12.
- ACPO, (2007). Association of Chief Police Officers. Good Practice Guide for Computer Based Electronic Evidence.
- Agarwal, A., Gupta, M., Gupta, S. and Gupta, S. C. (2011). Systematic digital forensic investigation model. *International Journal of Computer Science and Security (IJCSS)*, 5(1), 118-131.
- Agarwal, S.,(2010). Performance Analysis of Peer-To-Peer Botnets using “The Storm Botnet” as an Exemplar (Doctoral dissertation, University of Victoria).
- Agrawal, D., Das, S., and El Abbadi, A. (2010). Big data and cloud computing: new wine or just new bottles? *Proceedings of the VLDB Endowment*, 3(1-2), 1647-1648.
- Almirall, J. R., and Furton, K. G., (2003). Trends in forensic science education: Expansion and increased accountability. *Analytical and bioanalytical chemistry*, 376(8), 1156-1159.
- Aminnezhad A, Dehghantanha A, Abdullah MT, and Damshenas M. (2013) Cloud Forensics Issues and Opportunities.
- Annapureddy, K., (2010). Security challenges in hybrid cloud infrastructures. Aalto University.
- Ananthanarayanan R, Gupta, K, Pandey, P, Pucha, H, Sarkar, P, Shah M, and Tewari, R., (2009) Cloud analytics: Do we really need to reinvent the storage stack. In *Proceedings of the 1st USENIX Workshop on Hot Topics in Cloud Computing (HOTCLOUD’2009)*, San Diego, CA, USA.
- Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A. and Zaharia, M. (2010). A view of cloud computing. *Communications of the ACM*, 53(4), 50-58.

- Azfar, A., Choo, K. K. R. and Liu, L. (2016). An Android Social App Forensics Adversary Model. In 2016 49th Hawaii International Conference on System Sciences (HICSS) (pp. 5597-5606). IEEE.
- Bachiega, N. G., Martins, H. P., Spolon, R., Cavenaghi, M. A., Lobato, R. S. and Manacero, A. (2014). Open Source Cloud Computing: Characteristics and an Overview. Available at:. Last access, 10.
- Baig, M.M., Mahmood, W., (2007). A Robust Technique of Anti Key-Logging using Key-Logging Mechanism," Digital EcoSystems and Technologies Conference, 2007. DEST '07. Inaugural IEEE-IES , vol., no., pp.314,318, 21-23 Feb.
- Barske, D., Stander, A., and Jordaan, J., (2010). A digital forensic readiness framework for South African SME's. In Information Security for South Africa (ISSA), (pp. 1-6). IEEE.
- Banday, M. T., Qadri, J. A. and Shah, N. A. (2009).Study of Botnets and their threats to Internet Security.
- Baryamureeba, V., and Tushabe, F. (2004). The enhanced digital investigation process model. In Proceedings of the Fourth Digital Forensic Research Workshop.
- Beebe, N.L.,(2009). Digital forensics research: the good, the bad, and the unaddressed. In: Fifth annual IFIP WG 11.9 international conference on digital forensics; January 2009.
- Beebe, N. L., and Clark, J. G. (2005). A hierarchical, objectives-based framework for the digital investigations process. *Digital Investigation*, 2(2), 147-167.
- Bevel, T., and Gardner R.M.,(2002). *Bloodstain pattern analysis: with an introduction to crime scene reconstruction*". 2nd ed. Boca Raton, FL:CRC Press.
- Biermann, E. (2009). A framework for the protection of mobile agents against malicious hosts (Doctoral dissertation).
- Biggs, S. and Vidalis S.,(2009) Cloud computing: The impact on digital forensic investigations. In *Internet Technology and Secured Transaction. ICITST 2009. International Conference for* (pp. 1-6). IEEE.
- Birk, D and Wegener, C.,(2011). Technical Issues of Forensic Investigations in Cloud Computing Environments," *Systematic Approaches to Digital Forensic*

- Engineering (SADFE), 2011 IEEE Sixth International Workshop on , vol., no., pp.1,10, 26-26.
- Birk D. (2011) Technical challenges of forensic investigations in cloud computing environments. In Workshop on Cryptography and Security in Clouds, pp. 1-6.
- Boniface, M., Nasser, B., Papay, J., Phillips, S. C., Servin, A., Yang, X. and Kyriazis, D. (2010). Platform-as-a-service architecture for real-time quality of service management in clouds. In Internet and Web Applications and Services (ICIW), 2010 Fifth International Conference on (pp. 155-160). IEEE.
- Business Dictionary (2015), [online], Available at - <http://www.businessdictionary.com/definition/data-logging.html>
- Brittany M., (2016). Direct Evidence: Definition, Law & Examples. [online] Available at: <http://study.com/academy/lesson/direct-evidence-definition-law-examples.html> [Accessed March 26, 2016]
- By, L. R., and Krzewinski, K. 3.0 RESPONSIBILITY.
- Casey, E., (2011). Digital evidence and computer crime: Forensic science, computers, and the internet. Academic press.
- Casey E., (2002) Handbook of Computer Crime Investigation. Academic Press. Boston.
- Case Law Review: Unites Stetes v. Mosley (1994), United States Court of Appeal Ninth circuit [online]: Available: <http://www.crime-scene-investigator.net/admissibilitydigitalevidencecriminalprosecutions.html>.
- Carrier B.D.,(2006) Risks of live digital forensic analysis. Communications of the ACM, 49(2), 56-61.
- Carrier, B., and Spafford, E. H., (2004). An event-based digital forensic investigation framework. In Digital forensic research workshop (pp. 11-13).
- Carrier B. D., and Spafford E.H.,(2004). Defining event reconstruction of digital crime scenes. Journal of forensic sciences, 49(6), 1291-1298, 2004.
- Carrier, B., (2003). Defining digital forensic examination and analysis tools using abstraction layers. International Journal of digital evidence, 1(4), 1-12.
- Carrier, B., and Spafford, E. H. (2003). Getting physical with the digital investigation process. International Journal of digital evidence, 2(2), 1-20.

- Chahal, S., Hahn-Steichen, I. J., Kamhout, D., Engineer, I. I., Kraemer, R., Li, I. H., and Peters, I. C. (2010). An Enterprise Private Cloud Architecture and Implementation Roadmap. IT@ Intel White Paper, USA.
- Chambers, J., (Speaker). (2009). Collaborate with Confidence: Securely connect, Communicate, Conduct Business in Decentralized / Highly Collaborative Environment (Online only recording of a speech presented at the RSA).
- Chen, G. (2012). Suggestions to digital forensics in Cloud computing ERA, In Third IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC),
- Chen Z., Han F. J., Cao, X., Jiang, S., and Chen., (2013). Cloud computing-based forensic analysis for a collaborative network security management system. *Tsinghua Science and Technology*, 18(1), 40-50.
- Chiang, K. and Lloyd, L., (2007). A Case Study of the Rustock Rootkit and Spam Bot. Paper presented at the First Workshop on Hot Topics in Understanding Botnets, Cambridge, MA.
- Choo, K. K. R. (2010). Cloud computing: challenges and future directions. *Trends and Issues in Crime and Criminal justice*, (400), 1.
- Chung, H., Park, J., Lee, S., and Kang, C. (2012). Digital forensic investigation of cloud storage services. *Digital investigation*, 9(2), 81-95.
- Committee on Identifying the Needs of the Forensic Sciences Community, National Research Council (2009). *Strengthening Forensic Science in the United States: A Path Forward*, National Academy of Sciences.
- Cohen, F. B., (2009). *Digital forensic evidence examination*. Asp Press.
- Cohen F., (2008) *Challenges to digital forensic evidence*. Fred Cohen and Associates.
- Claycomb W.R., Nicoll A.,(2012) Insider threats to cloud computing: Directions for new research challenges. In *Computer Software and Applications Conference (COMPSAC)*, 2012 IEEE 36th Annual (pp. 387-394). IEEE.
- Daubert v. Merrell Dow Pharmaceuticals,(1993) Inc., 509 U.S. 579, 113 S. Ct. 2786, 125 L. Ed. 2d 469.
- Delpont, W., Köhn, M., and Olivier, M. S. (2011, August). Isolating a cloud instance for a digital forensic investigation. In *ISSA*.

- De Amorim, R. C., (2011). Learning feature weights for K-Means clustering using the Minkowski metric (Doctoral dissertation, PhD thesis. Department of Computer Science and Information Systems, University of London).
- Dietrich, C. J., Rossow, C., & Pohlmann, N. (2013). CoCoSpot: Clustering and recognizing botnet command and control channels using traffic analysis. *Computer Networks*, 57(2), 475-486.
- Dong, X., & Li, X. (2015). A Novel Distributed Database Solution Based on MySQL. In *Information Technology in Medicine and Education (ITME), 2015 7th International Conference on* (pp. 329-333). IEEE.
- Do, Q., Martini, B., and Choo, K. K. R. (2015). A forensically sound adversary model for mobile devices. *PloS one*, 10(9), e0138449.
- Dykstra, J., and 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.
- Dykstra, J., (2013). Seizing electronic evidence from cloud computing environments. Eggdrop,(1993). Open source IRC bot. “<http://www.eggheads.org/>, .
- Endicott-Popovsky, B., Frincke, D. A., and Taylor, C. A., (2007). A theoretical framework for organizational network forensic readiness. *Journal of Computers*, 2(3), 1-11.
- Elyas, M., Ahmad, A., Maynard, S. B., and Lonie, A. (2015). Digital forensic readiness: Expert perspectives on a theoretical framework. *Computers & Security*, 52, 70-89.
- Federal Rule of evidence: Rule 901. Authenticating or Identifying Evidence [online] at- https://www.law.cornell.edu/rules/fre/rule_901[Accessed,January 7, 2016]
- Feily, M., Shahrestani, A., and Ramadass, S., (2009). A survey of botnet and botnet detection. In *Emerging Security Information, Systems and Technologies, 2009. SECURWARE'09. Third International Conference on* (pp. 268-273). IEEE.
- Flood, J., and Keane A., (2012). A Proposed Framework for the Active Detection of Security Vulnerabilities in Multi-tenancy Cloud Systems. In *Third International Conference on Emerging Intelligent Data and Web Technologies (EIDWT)*, pp. 231-235. IEEE.

- Freiling, F., & Schwittay, B. (2007). A common process model for incident response and digital forensics. Proceedings of the IMF2007.
- Funfroeken, S., and Mattern, F.,(1999). Mobile Agents as an Architectural Concept for Internet-based Distributed Applications. The WASP project approach-In: Proceedings of KiVS;99. Steinmetz. Springer: PP 32-43.
- Gereda, S.L., (2006). The Electronic and Communications and Transactions Act. Telecommunications Law in South Africa.
- Godse, M., and Mulik, S., (2009). An approach for selecting software-as-a-service (SaaS) product. In Cloud Computing, 2009. CLOUD'09. IEEE International Conference on (pp. 155-158). IEEE.
- Golden, B., (2013). A unified formalism for complex systems architecture (Doctoral dissertation, Ecole Polytechnique X).
- Gong, C., Liu, J., Zhang, Q., Chen, H., and Gong, Z. (2010). The characteristics of cloud computing. In Parallel Processing Workshops (ICPPW), 2010 39th International Conference on (pp. 275-279). IEEE.
- Grispos, G., Storer, T., and Glisson, W. B., (2011). A comparison of forensic evidence recovery techniques for a windows mobile smart phone. Digital Investigation, 8(1), 23-36.
- Grispos G, Storer T, Glisson W., (2012). Calm before the storm: The challenges of cloud computing in digital forensics. International Journal of Digital Crime and Forensics, 4(2), 28-48, 2012.
- Grizzard, J. B., Sharma, V., Nunnery, C., Kang, B. B., and Dagon, D. (2007). Peer-to-peer botnets: Overview and case study. In Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets (pp. 1-1).
- Guo, P., (2009). A survey of Software as a Service Delivery Paradigm. In Seminar on.
- Guo, J., Qian, K., Han, D., & Zhang, G. (2015, April). A private cloud instances placement algorithm based on maximal flow algorithm. In Information Science and Control Engineering (ICISCE), 2015 2nd International Conference on (pp. 59-62). IEEE.
- Gu, G, Zhang, J., and Lee, W., (2008). BotSniffer: Detecting botnet command and control channels in network traffic.

- Gulati, A., Shanmuganathan, G., Holler, A. M., and Ahmad, I. (2011). Cloud Scale Resource Management: Challenges and Techniques. *HotCloud*, 11, 3-3.
- Gulati, A., Holler, A., Ji, M., Shanmuganathan, G., Waldspurger, C., and Zhu, X. (2012). Vmware distributed resource management: Design, implementation, and lessons learned. *VMware Technical Journal*, 1(1), 45-64.
- Han, F., Chen, Z., Xu, H., and Liang, Y., (2012). Garlic: A distributed botnets suppression system. In *Distributed Computing Systems Workshops (ICDCSW), 2012 32nd International Conference on* (pp. 634-639). IEEE.
- Herbst, N. R., Kounev, S., and Reussner, R., (2013). Elasticity in Cloud Computing: What It Is, and What It Is Not. In *ICAC* (pp. 23-27).
- Höne, K., and Eloff, J. H. P. (2002). Information security policy—what do international information security standards say?. *Computers & Security*, 21(5), 402-409.
- Hooper, C., Martini, B., & Choo, K. K. R., (2013). Cloud computing and its implications for cybercrime investigations in Australia. *Computer Law & Security Review*, 29(2), 152-163.
- Hou, S., Sasaki, R., Uehara, T., & Yiu, S. (2013, March). Verifying data authenticity and integrity in server-aided confidential forensic investigation. In *Information and Communication Technology-EurAsia Conference* (pp. 312-317). Springer Berlin Heidelberg.
- Hybrid Cloud definition <http://csrc.nist.gov/publications/nistpubs/800-146/sp800-146.pdf> [Accessed in May 7, 2015].
- <Http://openP2P.org/main-page>, [Online]-Accessed at April 2017.
- Internet Crime Report (2017), [Online]-Accessed-January 2017 at https://pdf.ic3.gov/2016_IC3Report.pdf.
- International Data Corporation (IDC) Top 10 prediction (2014). Accessed at <http://www.idc.com/research/Predictions14/index.jsp;jsessionid=0F5F27EF62AF965596E9D2177A95391B>.
- ISO/IEC 27043: 2015- Information technology -- Security techniques -- Incident investigation principles and processes. <https://www.iso.org/standard/44407.html>

- ISO/IEC 27001 - Information security management: Available from: in <http://www.iso.org/iso/iso27001>[Accessed in August 5, 2015].
- ISO/IEC 27017:2015 □ Information technology -- Security techniques -- Code of practice for information security controls based on ISO/IEC 27002 for cloud services. Available from: in http://www.iso.org/iso/catalogue_detail?csnumber=43757. [Accessed in September 10th 2015].
- ISO/IEC 27037: (2012). Information technology -- Security techniques -- Guidelines for identification, collection, acquisition and preservation of digital evidence.[online], Accessed at http://www.iso.org/iso/catalogue_detail?csnumber=44381 [Accessed in September 10th 2015]
- Jarrett H.M., and Bailie M.W (2002). Searching and Seizing Computers and Obtaining Electronic Evidence in Criminal Investigations. Available at <https://www.justice.gov/sites/default/files/criminal-ccips/legacy/2015/0>.
- Jacko, J.A., Stephanidis, C. and Harris, D., (2003). Human-computer interaction: Theory and practice. London: Lawrence Erlbaum Associates.
- Jahankhani H, Beqir E., (2010). Digital evidence manipulation using anti-forensic tools and techniques. Handbook of Electronic Security and Digital Forensics, 411.
- Jiang, B., Im, E. G., and Koo, Y. (2012, May). SaaS-Driven Botnets. In *PAISI* (pp. 198-206).
- Junewon, P. (2011). Acquiring digital evidence from Botnet attacks: procedures and methods (Doctoral dissertation, Auckland University of Technology).
- Kalt, C., (2000). Internet relay chat: Architecture.
- Kebande, V.R., and Venter H.S., (2016c). On Digital Forensic Readiness in the Cloud Using a Distributed Agent-Based Solution: Issues and Challenges, in *Australian Journal of Forensic Sciences*.
- Kebande V. R., and Venter H.S., (2017). Novel Digital Forensic Readiness Techniques In the Cloud Environment, *Australian Journal of Forensic Sciences*.
- Kebande V. R., and 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 Publishers, United Kingdom.

- Kebande, V. R., and Venter, H. S., (2015). Adding event reconstruction to a Cloud Forensic Readiness model. In Information Security for South Africa (ISSA), 2015 (pp. 1-9). IEEE.
- Kebande, V.R., & Venter, H.S., (2016a). Requirements for Achieving Digital Forensic Readiness in the Cloud Environment Using an NMB Solution. In 11th International Conference on Cyber Warfare and Security: ICCWS2016 (p. 399). Academic Conferences and publishing limited.
- Kebande V. R., and Venter, H.S., (2016b). 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., and 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, and Venter H.S., (2014). A cognitive approach for botnet detection using Artificial Immune System in the cloud, Cyber Security, Cyber Warfare and Digital Forensic (CyberSec), 2014 Third International Conference on , vol., no., pp.52,57.
- Kebande, VR., & Venter, H.S., (2015). Towards a Model for Characterizing Potential Digital Evidence in the Cloud Environment during Digital Forensic Readiness Process. In ICCSM2015-3rd International Conference on Cloud Security and Management: ICCSM2015 (p. 151). Academic Conferences and publishing limited.
- 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 Limited.
- Kebande V.R, Venter, H. S., (2015) A Functional Architecture for Cloud Forensic Readiness Large-scale Potential Digital Evidence Analysis. In Proceedings of the 14th European Conference on Cyber Warfare and Security 2015: ECCWS (p. 373). Academic Conferences Limited.

- Kent, S K., Chevalier, T.G, and Dang, H.,(2006). Guide to integrating forensic techniques into incident response, NIST Special Publication, pp. 800-86.
- Kerr, O. S., (2004). A User's Guide to the Stored Communications Act, and a Legislator's Guide to Amending it. Available at SSRN 421860.
- Kessler, G. C., (2000). Defenses against distributed denial of service attacks. SANS Institute.
- Khanna, N., Mikkilineni, A. K., Martone, A. F., Ali, G. N., Chiu, G. T. C., Allebach, J. P., & Delp, E. J. (2006). A survey of forensic characterization methods for physical devices. *digital investigation*, 3, 17-28.
- Khorram, M., & Moosavian, S. A. A. (2015, October). Modified Jacobian transpose control of a quadruped robot. In *Robotics and Mechatronics (ICROM), 2015 3rd RSI International Conference on* (pp. 067-072). IEEE.
- Koen, R., & Olivier, M. S. (2008, July). The Use of File Timestamps in Digital Forensics. In *ISSA* (pp. 1-16).
- Köhn, M.D., Eloff, J. H and Olivier, M. S (2006). Framework for a Digital Forensic Investigation. In *ISSA* (pp. 1-7).
- Kohn, M.D., Eloff, M. M., and Eloff, J. H. (2013). Integrated digital forensic process model. *Computers & Security*, 38, 103-115.
- Kumar Alluri, B.K.S.P, Geethakumari, G., (2015). A digital forensic model for introspection of virtual machines in cloud computing, *Signal Processing, Informatics, Communication and Energy Systems (SPICES), 2015 IEEE International Conference on* , vol., no., pp.1,5, 19-21.
- Kuntze, N., & Rudolph, C. (2011, May). Secure digital chains of evidence. In *Systematic Approaches to Digital Forensic Engineering (SADFE), 2011 IEEE Sixth International Workshop on* (pp. 1-8). IEEE.
- Kundra, V., (2011). Federal cloud computing strategy.
- Lampe, V, K. (2015). *Organized crime: analyzing illegal activities, criminal structures, and extra-legal governance*. SAGE Publications.
- Leder, F., Werner, T., and Martini, P., (2009). Proactive botnet countermeasures: an offensive approach. *The Virtual Battlefield: Perspectives on Cyber Warfare*, 3, 211-225.

- Lillis, D., Becker, B., O'Sullivan, T., and Scanlon, M. (2016). Current Challenges and Future Research Areas for Digital Forensic Investigation. *arXiv preprint arXiv:1604.03850*.
- Li, J., Xu, M., Zheng, N., and Xu, J., (2009). Malware obfuscation detection via maximal patterns. In *Intelligent Information Technology Application, 2009. IITA 2009. Third International Symposium on (Vol. 2, pp. 324-328)*. IEEE.
- Lin, W., and Lee, D. (2012, June). Traceback Attacks in Cloud--Pebbletrace Botnet. In *Distributed Computing Systems Workshops (ICDCSW), 2012 32nd International Conference on (pp. 417-426)*. IEEE.
- Liu, J., Xiao, Y., Ghaboosi, K., Deng, H., and Zhang, J. (2009). Botnet: classification, attacks, detection, tracing, and preventive measures. In *EURASIP journal on wireless communications and networking (Vol. 2009, pp. 1184-1187)*. IEEE Computer Society.
- Liu, Y., Vlassov, V., and Navarro, L., (2014). Towards a community cloud storage. In *Advanced Information Networking and Applications (AINA), 2014 IEEE 28th International Conference on (pp. 837-844)*. IEEE.
- Mahmood, Z., (2011). Cloud computing: Characteristics and deployment approaches. In *Computer and Information Technology (CIT), 2011 IEEE 11th International Conference on (pp. 121-126)*. IEEE.
- Malan, R., and Bredemeyer, D., (2001). Architecture resources.
- Mannila, H., and Moen, P. (1999). Similarity between event types in sequences. *Data Warehousing and Knowledge Discovery*, 804-804.
- Marangos, N., Rizomiliotis, P., and Mitrou, L. (2012). Digital forensics in the cloud computing era. In *Globecom Workshops (GC Wkshps), 2012 IEEE (pp. 775-780)*. IEEE.
- Martini, B., and Choo, K. K. R., (2012). An integrated conceptual digital forensic framework for cloud computing. *Digital Investigation*, 9(2), 71-80.
- Martini, B., and Choo, K. K. R. (2013). Cloud storage forensics: OwnCloud as a case study. *Digital Investigation*, 10(4), 287-299.
- Martini, B., and Choo, K. K. R. (2014b). Remote programmatic vCloud forensics: a six-step collection process and a proof of concept. In *Trust, Security and Privacy in*

- Computing and Communications (TrustCom), 2014 IEEE 13th International Conference on (pp. 935-942). IEEE.
- Martini, B., and Choo, K. K. R., (2014c). Cloud forensic technical challenges and solutions: a snapshot. *IEEE Cloud Computing*, (4), 20-25.
- . Marty, R., (2011). Cloud application logging for forensics. In *Proceedings of the 2011 ACM Symposium on Applied Computing* (pp. 178-184). ACM.
- Marcella, A.J. and Greenfield, R.S. (2002). *Cyber forensics: A field Manual for collecting, examining and preserving evidence of computer crime*, London: Taylor & Francis.
- Masters, P. H., Lam, J. K., and Wong, K., (1991). Incident detection algorithms for COMPASS-An advanced traffic management system. In *Vehicle Navigation and Information Systems Conference, 1991* (Vol. 2, pp. 295-310). IEEE.
- Maurer, M., Emeakaroha, V. C., Brandic, I., and Altmann, J. (2012). Cost-benefit analysis of an SLA mapping approach for defining standardized Cloud computing goods. *Future Generation Computer Systems*, 28(1), 39-47.
- Menezes, A. J., Van Oorschot, P. C., and Vanstone, S. A. (1996). *Handbook of applied cryptography*. CRC press.
- Mell, P., and Grance, T., (2011). The NIST definition of cloud computing.
- Miao, R., Potharaju, R., Yu, M., and Jain, N. (2015, October). The dark menace: Characterizing network-based attacks in the cloud. In *Proceedings of the 2015 ACM Conference on Internet Measurement Conference* (pp. 169-182). ACM.
- Mouton, F., (2012). *Digital Forensic Readiness for Wireless Sensor Network Environments* (Doctoral dissertation, University of Pretoria).
- Mouton, F., and Venter, H. S. (2011). A prototype for achieving digital forensic readiness on wireless sensor networks. In *AFRICON, 2011*(pp. 1-6). IEEE.
- Mozilla Development Network (2015) [online]: Available: https://developer.mozilla.org/en-US/Learn/How_the_Internet_works[Accessed in September 10th 2015]
- Mouhtaropoulos, A., Li, C. T., and Grobler, M., (2012). Proactive Digital Forensics: The Ever-Increasing Need for Standardization. In *Intelligence and Security Informatics Conference (EISIC), 2012 European* (pp. 289-289). IEEE.

- Nelson, Bill, Phillips, Amelia, Enfinger, Frank, and Steuart, Chris Guide to Computer Forensics and Investigations Thomson Learning Inc. - Course Technology, Canada, 2004, p. 689.
- Ngomane, A. R., (2010). The use of electronic evidence in forensic investigation.
- Ngobeni, S., Venter, H. S., and Burke, I. (2012). The Modelling of a Digital Forensic Readiness Approach for Wireless Local Area Networks. J. UCS, 18(12), 1721-1740.
- Nolan, R., O'sullivan, C., Branson, J., and Waits, C., (2005). First responders guide to computer forensics (No. CMU/SEI-2005-HB-001). carnegie-mellon univ pittsburgh pa software engineering inst.
- NIST Cloud Computing Standards Roadmap (2013), Special Publication 500-291, Version 2-http://www.nist.gov/itl/cloud/upload/NIST_SP-500-291_Version_2_2013_June18_FINAL.pdf [Accessed in September 16th 2014].
- NIST SP 800-37 (2010). Guide for Applying the Risk Management Framework to Federal Information Systems, A security Life Cycle Approach. Available at <http://csrc.nist.gov/publications/nistpubs/800-37-rev1/sp800-37-rev1-final.pdf>.
- NIST. Computer Forensic Tool Testing (CFTT) Project Overview. Available at <https://www.nist.gov/content/computer-forensics-tool-testing-cftt-project>.
- Oikarinen, J and Reed, D. (1993). Internet relay chat protocol. At <http://tools.ietf.org/html/rfc1459.html>. [Accessed in January 8th 2015]
- Ottenheimer, D., and Wallace, M. (2012). Securing the virtual environment. Indianapolis, in: John Wiley and Sons.
- Owen P, and Thomas P., (2011). An analysis of digital forensic examinations: Mobile devices versus hard disk drives utilising ACPO & NIST guidelines. Digital Investigation, vol. 8, pp. 135-140.
- Olive, C., (2011). Cloud Computing Characteristics Are Key. General Physics Corporation, www.gpworldwide.com.
- Palmer G.A., (2001). Roadmap for Digital Forensic Research. Digital Forensics Research Workshop (DFRWS).
- Pearson, S., (2013). Privacy, security and trust in cloud computing. In Privacy and Security for Cloud Computing (pp. 3-42). Springer London.

- Pohl, K., (2010). Requirements engineering: fundamentals, principles, and techniques. Springer Publishing Company, Incorporated.
- Pollitt, M.M. (2004) Six blind men from Indostan. Digital forensics research workshop (DFRWS).
- Pollitt, M.M., (2007). An ad hoc review of digital forensic models. In Systematic Approaches to Digital Forensic Engineering, 2007. SADFE 2007. Second International Workshop on (pp. 43-54). IEEE.
- Pooe, A., Labuschagne, L. A, (2012). conceptual model for digital forensic readiness, in Information Security for South Africa (ISSA), IEEE.
- Qian, M., Wang, Y., Zhou, Y., Tian, L., and Shi, J. (2015). A Super Base Station based Centralized Network Architecture for 5G Mobile Communication Systems. Digital Communications and Networks.
- Quick, D., and Choo, K. K. R., (2014e). Impacts of increasing volume of digital forensic data: A survey and future research challenges. Digital Investigation, 11(4), 273-294.
- Quick, D., and Choo, K. K. R., (2014f). Data reduction and data mining framework for digital forensic evidence: storage, intelligence, review and archive. Trends & Issues in Crime and Criminal Justice, 480, 1-11.
- Quick, D., and Choo, K. K. R., (2013a). Forensic collection of cloud storage data: Does the act of collection result in changes to the data or its metadata?. Digital Investigation, 10(3), 266-277.
- Quick, D., and Choo, K. K. R., (2013b). Digital droplets: Microsoft SkyDrive forensic data remnants. Future Generation Computer Systems, 29(6), 1378-1394.
- Quick, D., and Choo, K. K. R., (2013c). Dropbox analysis: Data remnants on user machines. Digital Investigation, 10(1), 3-18.
- Quick, D., and Choo, K. K. R., (2014a). Google Drive: Forensic analysis of data remnants. Journal of Network and Computer Applications, 40, 179-193.
- Rahman, Ab, N. H., and Choo, K. K. R., (2015). A survey of information security incident handling in the cloud. Computers & Security, 49, 45-69.
- Rahman, Ab, Glisson, W. B., Yang Y., and Choo, K. K. R., (2016) Forensic-by-Design Framework for Cyber-Physical Cloud Systems, in IEEE Cloud Computing, vol. 3, no. 1, pp. 50-59,.doi: 10.1109/MCC.2016.5.

- Rahman Ab, N. H. and Choo, K. K. R. (2015). Integrating digital forensic practices in cloud incident handling: A conceptual cloud incident handling model. *Cloud Security Ecosystem*R..
- Ramgovind, S., Eloff, M. M., and Smith, E. (2010). The management of security in cloud computing. In *Information Security for South Africa (ISSA)*, 2010 (pp. 1-7). IEEE.
- Reith, M., Carr, C., and Gunsch, G. (2002). An examination of digital forensic models. *International Journal of Digital Evidence*, 1(3), 1-12.
- Regulation of Interception of Communications and Provision of Communication-related Information Act 70 of 2001.
- Richter, J., Kuntze, N., and Rudolph, C. (2010). Security digital evidence. In *Systematic Approaches to Digital Forensic Engineering (SADFE)*, 2010 Fifth IEEE International Workshop on (pp. 119-130). IEEE.
- Rodríguez-Gómez, R. A., Maciá-Fernández, G., and Garcia-Teodoro, P. (2011). Analysis of botnets through life-cycle. *SECRYPT*, 257-262.
- Rowlingson R.A., (2004). Ten Step Process for Forensic Readiness. *International Journal of Digital Evidence* volume 2, no. 3.
- Ryan, D. J., and Shpantzer, G. (2002). Legal aspects of digital forensics. In *Proceedings: Forensics Workshop*.
- Seybent. H, Reinecke. P (2014). Internet and cloud services - statistics on the use by individuals. Half of Europeans used the internet on the go and a fifth saved files on internet storage space in 2014. Available at http://ec.europa.eu/eurostat/statisticsexplained/index.php/Internet_and_cloud_services_-_statistics_on_the_use_by_individuals.
- Scientific Working Group on Digital Evidence. SWGDE/SWGIT Digital & Multimedia Evidence Glossary: Version 2.7, 2013. Available at: in <https://www.swgde.org>. [Accessed in December 10th 2015].
- Scolnik A.,(2004). Protections for electronic communications: The stored communications act and the Fourth Amendment. *Fordham L. Rev.*, 78, 2009, 34
- Specht, S. M., & Lee, R. B. Distributed Denial of Service: Taxonomies of Attacks, Tools, and Countermeasures. In *ISCA PDCS* (pp. 543-550).

- Scolnik, A., (2009). Protections for electronic communications: The stored communications act and the Fourth Amendment. *Fordham Law Review*, 78(1).
- Schwerha IV, J. J. (2004). Cybercrime: legal standards governing the collection of digital evidence. *Information Systems Frontiers*, 6(2), 133-151.
- Schiller, C., and Binkley, J. R., (2011). *Botnets: The killer web applications*. Syngress.
- Schoeman, M. H., and Jones M. M., (2004). *Legality of Monitoring E-Mail At The Workplace: A Legal Update*.
- Shorter Oxford English Dictionary (6th ed.), Oxford University Press, 2007.
- Simou, S., Kalloniatis, C., Kavakli, E., and Gritzalis, S., (2014). Cloud forensics: identifying the major issues and challenges. In *Advanced Information Systems Engineering* (pp. 271-284). Springer International Publishing.
- Singh, A., Yadav, A., and Rana, A., (2013). K-means with Three different Distance Metrics. *International Journal of Computer Applications*, 67(10).
- Sowa, J. F. (2000). Processes and causality. Retrieved April, 2, 2016. Accessed at <http://www.jfsowa.com/ontology/causal.html>.
- Sriram, I., and Khajeh-Hosseini, A., (2010). Research agenda in cloud technologies. arXiv preprint arXiv:1001.3259.
- Stanoevska-Slabeva, K., Wozniak, T., Thanos, G. A., Parrilli, D. M., Serrabou, B., and Luokkanen-Rabetino, K., (2009). Turning Grid Research into Business- Identification of Commercialization Barriers. In *Proceedings of the First International ICST Conference on Digital Business-DIGIBIZ*.
- Sulkoswki, A. J., (2007). Cyber-Extortion: Duties and Liabilities Related to the Elephant in the Server Room. U. Ill. JL Tech. & Pol'y, 19.
- Symantec security updates-<http://www.symantec.com/avcenter/warn/backorifice.html>.
- SWGDE Data Integrity Within Computer Forensics V1-0 (2006). Accessed at <https://www.swgde.org/documents/Archived%20Documents/SWGDE%20Data%20Integrity%20Within%20Computer%20Forensics%20V1-0>.
- Tanner, A., Dampier, D., & Thompson, J. (2012). On developing a conceptual modeling report management tool for digital forensic investigations. In *Homeland Security (HST), 2012 IEEE Conference on Technologies for* (pp. 445-450). IEEE.
- Tan, J. (2001). *Forensic readiness*. Cambridge, MA: @ Stake, 1-23.

- Taylor, S and Metzler, J (2010). Cloud Computing: reality vs Fiction. Accessed at Network World. [online]:-“ in <http://www.networkworld.com/article/2216572/lan-wan/cloud-computing--reality-vs--fiction.html>”[Accessed in November 11th 2015].
- Telecommunications Law in South Africa. The Protection of Personal Information Act, Vol. 581, No 4, 2013.
- TC-STAG, E., (1996). Security techniques advisory group (stag); definition of user requirements for lawful interception of telecommunications: requirements of the law enforcement agencies.
- Thompson II, R. M. (2013). Cloud Computing: Constitutional and Statutory Privacy Protections. CRS Report for Congress, 16.
- Trenwith, P. M., and Venter, H. S. (2013). Digital forensic readiness in the cloud. In Information Security for South Africa, 2013 (pp. 1-5). IEEE.
- The Botnet Chronicles: A journey to infamy(2010) Accessed at http://countermeasures.trendmicro.eu/wpcontent/uploads/2012/02/the_botnet_chronicles_-_a_journey_to_infamy__nov_2010_.pdf [Accessed in January 5th 2015]
- The Protection of personal Information act (2013) Vol 581 No 4.
- Vacca, J.R., (2005). Computer forensics: Computer Crime Scene Investigation. Charles River Media, 20 Downer Avenue, Suite 3, Hingham, MA, 02043, second edition.
- Varia, J., (2008). Cloud architectures. White Paper of Amazon, jineshvaria. s3.amazonaws.com/public/cloudarchitectures-varia.pdf, 16.
- Valjarevic, A., and Venter, H. S., (2015). A Comprehensive and Harmonized Digital Forensic Investigation Process Model. Journal of forensic sciences,60(6), 1467-1483.
- Valjarevic, A., and Venter, H. S., (2012). Harmonised digital forensic investigation process model. In Information Security for South Africa (ISSA), 2012 (pp. 1-10). IEEE.
- Vahidi, A., & Ekdahl, P. (2013). VETE: Virtualizing the Trusted Execution Environment.
- Vincze, E. A., (2015). Challenges in digital forensics. Police Practice and Research, 1-12.
- Watkins, H., (1994). Daubert v. Merrell Dow Pharmaceuticals, Inc.: General acceptance rejected. Santa Clara Computer & High Tech. LJ, 10, 259.

- Wang, P., Aslam, B., and Zou, C. C. (2010). Peer-to-peer botnets. In Handbook of Information and Communication Security (pp. 335-350). Springer Berlin Heidelberg. ISBN.
- Wilcox, J.(2011) Gartner: Most CIOs have their head in the cloud. Available at- <http://betanews.com/2011/01/24/gartner-most-cios-have-their-heads-in-the-clouds/>. [Accessed on January 8th 2016].
- Wilhelm, U.G., Staamann, S. and Buttyan, L., (1999). Introducing trusted third parties to the mobile agent paradigm. Secure Internet Programming: Security Issues for Mobile and Distributed Objects. Springer. PP 471-491.
- Wozniak, T., and Ristol, S., (2009) Grid and Cloud Computing. A Business Perspective on Technology and Applications. Springer Berlin Heidelberg.
- Yusoff, Y., Ismail, R., and Hassan, Z. (2011). Common phases of computer forensics investigation models. International Journal of Computer Science & Information Technology (IJCSIT), 3(3), 17-31.
- Zawoad, S., and Hasan, R., (2013). Cloud forensics: a meta-study of challenges, approaches, and open problems. arXiv preprint arXiv:1302.6312.
- Zawoad S, and Hasan R., (2013). Digital Forensics in the Cloud. Alabama Univ in Birmingham.
- Zawoad, S., Dutta, A. K., and Hasan, R., (2013). SecLaaS: secure logging-as-a-service for cloud forensics. In Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security (pp. 219-230). ACM.
- Zamil, M. F., Manasrah, A. M., Amir, O., and Ramadass, S., (2010, May). A behavior based algorithm to detect Spam bots. In Collaborative Technologies and Systems (CTS), 2010 International Symposium on (pp. 453-462). IEEE.
- Zhang, L., Ning, H. Y., & Yang, Y. (2016). A New Type MySQL Integrated Mutual Authentication Security Model. In Instrumentation & Measurement, Computer, Communication and Control (IMCCC), 2016 Sixth International Conference on (pp. 253-257). IEEE.
- Zonana, H. (1994). Daubert v. Merrell Dow Pharmaceuticals: A new standard for scientific evidence in the courts? Journal of the American Academy of Psychiatry and the Law Online, 22(3), 309-325.

APPENDIX A. PSEUDO-EQUATIONS AND ILLUSTRATIONS

Equation	Equation	Pseudo-Equation	Illustrative Example
5.1	$Fl = \{Fl_{(IP)}, Fl_{(TP)}, Fl_{(KST)}, Fl_{(Uname)}, Fl_{(UID)}, Fl_{(C_Usage)}, Fl_{(R_Usage)}\}$	<ul style="list-style-type: none"> ✓ $Fl_{(IP)}$-Forensic log with an IP address where it originates ✓ $Fl_{(TP)}$-Forensic log with a timestamp ✓ $Fl_{(KST)}$-Forensic log extracted as a result of a keystroke ✓ $Fl_{(Uname)}$-Forensic log denoting username ✓ $Fl_{(UID)}$-Forensic name ✓ $Fl_{(C_Usage)}$-CPU usage forensic log 	<p>An digital investigation about a suspected intrusion needs Forensic logs that have the following:</p> <ol style="list-style-type: none"> 1. Timestamp showing the following: <ul style="list-style-type: none"> • Time of occurrence • Date, month, year 2. IP address 3. Username
5.2	$DP = \{B_fl_1, B_fl_2, \dots, B_fl_n\}$	<ul style="list-style-type: none"> • DP=Digital Preservation • Digital Preservation=Set/col lection of forensically logged data. 	<p>Forensic Logs are collected as blocks. A block represents a periodic collection [each 1 minute]. After each one minute the block is digitally preserved.</p>

5.3	$HshV = \{Hsh(B_fl)_1, Hsh(B_fl)_2, \dots, Hsh(B_fl)_n\}$	<ul style="list-style-type: none"> ✓ HshV-Hash values created for forensically logged data ✓ Hsh(B_fl)-Hash for a block of forensic log 	<p>For each collected block, a hash is created.</p> <p>Data collected for CPU and RAM usage as is periodically collected, a hash is created before it is posted to the forensic database.</p>
5.4	$DP = \{HshV\} = \{Hsh(B_fl)_1, Hsh(B_fl)_2, \dots, Hsh(B_fl)_n\}$	DP=collection of hash values created for blocks of collected logs	The hash values that are created on the block of collected digital data constitutes the digital preservation process.
5.5	$Fl_{(TP)} = \{B_fl_j, B_fl_{j+1}, \dots, B_fl_{n+1}\}, \{Hsh(Hshk_0), Hshk_2 = Hsh(Hshk_1) \dots Hshk_i = Hsh(Hshk_{i-1})\}$	$Fl_{(TP)}$ -Collection of hashed log files that have respective timestamps	Each of the hashed log file should have a respective timestamp that shows how each particular event occurred.
5.6	$DEC = \{field_N_1(Fl_1), field_N_2(Fl_2), \dots, field_N_n(Fl_n)\}$	DEC=Digital Evidence characterization- collection of fields that contain forensic logs.	To distinguish forensic log, one needs to look at the fields that have the forensic logs.
5.7	$FA^o = OBx^o$	FA=Forensic Agent=Obfuscation combined obfuscation vector	A forensic Agent can be hidden by the presence of a vector that changes the patterns
5.8	$OB = \frac{\partial FA^o}{\partial OBx^o}$	OB=Obfuscation Obfuscation=Being hidden/undetected	This is a modified Jacobian Equation. This equation helps the forensic agent to be able to be transformed S.T it can easily evade detection.
5.9	$OB = \begin{bmatrix} \frac{\partial FA^o}{\partial OBx^o_1} & \dots & \frac{\partial FA^o}{\partial OBx^o_n} \end{bmatrix}$	The matrix represents modified Jacobian equation	Represents modified Jacobian matrix for partial derivatives of the forensic agent, obfuscation process

			and obfuscation vector. This is also the determinant of Equation 5.10.
5.10	$\begin{bmatrix} \frac{\partial FA^o_1}{\partial OBx^o_1} & \dots & \frac{\partial FA^o_1}{\partial OBx^o_n} \\ \cdot & & \\ \frac{\partial FA^o_m}{\partial OBx^o_1} & \dots & \frac{\partial FA^o_m}{\partial OBx^o_n} \end{bmatrix}$	Represents the Jacobian square matrix.	The square matrix is a representation of functions that represent the obfuscation process and it is a determinant of Equation 5.9.
5.11	$FA = [OB + (x^0)]$	A forensic agent can easily be obfuscated by introducing the obfuscation vector	The obfuscation Vector facilitates the process of hiding the forensic agent.
5.12	$OB = C[FA + x^0]$	Obfuscation process in the cloud= forensic agent combined with obfuscation vector	In the cloud environment the forensic agent can be hidden by the presence of the vector that changes the patterns in a way that is difficult to decipher
7.1	$CM = \{CSP_1, CSP_2, \dots, CSP_m\}, [m \geq 1]$	<ul style="list-style-type: none"> • CM=Cloud Model. • Cloud Model=Collection of Cloud service providers. • At least one CSP will exist in a cloud model. 	In a cloud set-up there will be at least one Cloud service Provider that will offer and loud infrastructure and resources.
7.2	$CSP = \{Cl_1, Cl_2, \dots, Cl_p\}, [p > 0]$	<ul style="list-style-type: none"> • CSP=Cloud Service Provider • Cloud Service Provider=Collection/ set of cloud clients 	A Cloud Service Provider must have clients who uses the resources being offered.

		<ul style="list-style-type: none"> In a CSP there will always be at least one client. 	
7.3	$Cl_i \cap Cl_n = \varphi, (0 \leq i, n \leq p, i \neq n)$	<ul style="list-style-type: none"> Cls and the CSP are independent among n number of clients and n number of CSPs; 	The clients operates on their own while making use of cloud resources
7.4	$CSP_i \cap CSP_n = \varphi, (0 \leq i, n \leq p, i \neq n)$	<ul style="list-style-type: none"> Cls and the CSP are independent among n number of clients and n number of CSPs; 	The CSP s operates on their own but depend on the clients to use the resources that they offer.
7.5	$Set(Cl_{i1}, CSP_{n1}) \cap Set(Cl_{i2}, CSP_{n2}) = \varphi, [0 \leq i, n \leq p, i \neq n]$	The interconnection between the Cl s and the CSP s that shows how independent each entity is.	There exist a relationship between the cloud clients and the CSP s in terms of resource provisions.
7.6	$CSP = \{Cl_1, Cl_2, \dots, Cl_i\}, i \in N$	Cloud Service Provider is a collection of cloud clients.	In every resource that the CSP offers, there also exist at least one client.
7.7	$CSP = \{ \{Cl_i \{Dc = \{Dc_1, Dc_2, \dots, Dc_j\}, j \in N, Dc > 0\} \} \}$	Cloud Service Provider is distributed across Data centers that holds client data.	The cloud operates on a Data center, with client that are provided resources by the CSP s
7.8	$CSP = \left\{ \begin{array}{l} Cl^{Pr}_1, Cl^{Pr}_2, \dots, Cl^{Pr}_n \\ Cl^{PB}_1, Cl^{PB}_2, \dots, Cl^{PB}_n \\ Cl^{HB}_1, Cl^{HB}_2, \dots, Cl^{HB}_n \end{array} \right\} Cl \geq 1$	Cloud Service Provider (CSP) = Collection of clients in private, public and hybrid cloud.	The CSP can operate in all the cloud models like the Private cloud, Public cloud and hybrid cloud.
7.9	$Dc = \{Ps, Vs, OS\}$	<ul style="list-style-type: none"> Dc=Data Center 	The physical server, virtual server

		<ul style="list-style-type: none"> Data Center=Collection of Physical server, Virtual server and Operating Systems 	and the operating systems, application and services constitutes a Data center.
7.10	$CSPs = \{Cl_i\{Dc_j\} = \{Ps, Vs, OS\}, j \in N, Dc > 0\}$	CSP=Collection of Data center and Clients	The cloud operates on a Data center, with client that are provided resources by the CSPs
7.11	$Ps = \{R_1, R_2, \dots, R_n\}$	<ul style="list-style-type: none"> Ps=Physical Server Physical Server=collection of resources. 	A physical server consist of a set of resources that are provisioned to clients
7.12	$Vs = \{VM_1, VM_2, \dots, VM_n\}$	<ul style="list-style-type: none"> Vs=Virtual Server. Virtual Server=Collection of Virtual Machines. 	A virtual server consist of VMs that are also provisioned so that they can be used during virtualization process.
7.13	$OS = \{appns_1, appns_2, \dots, appns_n\}$	<ul style="list-style-type: none"> OS=Operating Systems. Operating System=collection of applications and services. 	Applications and services constitutes the operating systems.
7.14	$CSP = \left\{ Cl_i \{ Dc_j \} = \begin{cases} Ps = \{R_1, R_2, \dots, R_n\} \\ Vs = \{VM_1, VM_2, \dots, VM_n\} \\ OS = \{appns_1, appns_2, \dots, appns_n\} \end{cases} \right\}, j \in N, Dc > 0$	<ul style="list-style-type: none"> CSP=collection of clients. CSP=set of data centers Data center=collection of Physical server, 	CSPs relies on Data centers and the Data center consist of physical servers, virtual servers, applications and services.

		Virtual server and Operating System.	
7.15	$Potential_Risk = Org [P_Threats] \times Org [P_Vulnerabilities] \times Org [Cost]$	Potential risk=Product of threats, vulnerabilities and cost	If one wants to calculate the risks that affect any organization, it is worth to consider the possible threats, vulnerabilities and the cost involved.
7.16	$CSP = \{ \{ Cl_i \} = \{ Ac_1, Ac_2, \dots, Ac_n \} \}$	<ul style="list-style-type: none"> • CSP=Cloud Service Providers. • CSP==Collection of activities performed by clients. 	
7.17	$Ac_i = \{ (Fl_{t1}, To_1), (Fl_{t2}, To_2) \dots (Fl_{tn}, To_i) \}$	<ul style="list-style-type: none"> • Ac_i=Activities • Activities=collection of forensic logs and the number of times logging occurs • Fl_{t1}=Forensic log with an identifier • To_1= the number of times (To_i) the logging activity occurs. 	Every extracted forensic log that has a timestamp and the number of time that forensic log occurs constitutes an activity.
7.18	$Ac_i = \{ Fl_{TPi1}, Fl_{TPi2} \dots Fl_{TPip} \}, i \in [1, n]$	Activities=collection of forensic logs with respective	Every extracted forensic log that has a timestamp and the number of time

		timestamps	that forensic log occurs constitutes an activity.
7.19	$Fl_{ij}=\{e_{ij1}, e_{ij2}, \dots, e_{ijm}\}, i \in [1,n], j \in [1,p]$	<ul style="list-style-type: none"> • Fl_{ij}=Forensic log with a timestamp. • Forensic log with timestamp=collection of potential security events. 	A forensic log with a timestamp constitutes an event.
7.20	$e_{ij}=\{at_{ij1}, at_{ij2}, \dots, at_{ijk}\}, i \in [1,n], j \in [1,p], k \in [1,m]$	e_{ij} =Potential Security Events Potential Security Events=Collection of event attributes.	Each event that occurs has a set of attributes.
7.21	$CFRaaS=\{CSP=\{\{Cl_i\}=Ac_i\{Fl_{ij}\{e_{ij}\{at_{ij}\}\}\}\} \Leftrightarrow dp\}$	CFRaaS Model=Collection of CSPs, cloud clients, activities conducted by clients, forensic logs generated from clients, possible security events and event attributes which eventually are digitally preserved.	The CFRaaS model in this context comprise of a provider that gives services to clients that carries out activities which results to extraction of forensic logs that have attributes. This are digitally preserved as potential evidence.
7.22	$DP=\{dp_1, dp_2, dp_3, \dots, dp_i\}, i \in N$	Digital preservation=Collection of digitally preserved objects/digital files.	Collection of digitally preserved data constitutes the digital preservation process.
7.23	$DP=\{p \in P_o dp_{\alpha, p}\}$	DP=Relationship between a digital object and how it is digitally preserved.	For each digital object that is preserved there exist a relationship between digital objects.
7.24	$Ac_i=\{at_1, at_2, at_3, \dots, at_j\}, j \in N$	<ul style="list-style-type: none"> • Ac_i=Activities. • Activities=Collection 	All activities conducted by clients have some attributes attached to

		n of attributes	them.
7.25	$\{ Pi_P \cup Pi_Des \cup PDE_A \cup Pi_A \} = Ac_i \Leftrightarrow \{ R_PDE, NR_PDE \}$	<ul style="list-style-type: none"> • Pi_P=Pre-Incident Planning • Pi_Des=Pre-Incident Description • Pi_A=Pre Incident Analysis $Pi_P \cup Pi_Des \cup PDE_A \cup Pi_A$ =collection of activities that shows relevant and non-relevant potential evidence	A combination of processes that begins in planning, incident detection and analysis constitutes activities which in the long run determines whether we have relevant potential evidence or non-relevant potential evidence.
7.26	$IDR = \left(\frac{\text{Number_of_Incidents_Detected}}{\text{Number_of_Real_Incidents}} \right) + \{ \text{False_Alarms} \}$	<ul style="list-style-type: none"> • IDR=Incident Detection Rate • Incident Detection Rate=division of the number of incidents detected and number of real incidents combined with false alarms. 	This calculation is based on the rate through which incidents may be experienced within a given organization. One would rely on the number of incidents, the real incidents and false alarms.
7.27	$IDR = \left(\frac{4500}{3500} \right) + \{ 10 \}$	IDR Computation	This calculation is based on the rate through which incidents may be experienced within a given organization. One would rely on the number of incidents, the real incidents and false alarms.

7.28	$Incident_Growth_Rate = \left(\frac{Present_Security_Incidents - Past_Security_Incidents}{Past_Security_Incidents} \right) * 100$	Incident Growth Rate computation considers a percentage of past and present security incidents	This calculation presents the rate through which incidents grows. It is based on the number of past security incidents and present security incidents as a percentage.
7.29	$IRM = IDR + \{Incident_Description\}$	<ul style="list-style-type: none"> • IRM=Incident Response Mechanism. • Incident Response Mechanism=The rate of detecting the incident and the description of the incident. 	
7.30		<ul style="list-style-type: none"> • $d(e_i)$=Distance between two events • A_n=Fields that contain logs that have possible events • W_{1TP}-first event • W_{2TP}-second event 	There will always exist a distance between two events. However, based on the similarity between two events that has been highlighted above, the researcher approaches this by considering the fact that whenever there exist two events, for example X and Y, it is possible they may be similar if they occur in the same context. This explains why the distance between the two events X and Y may vary-that is if they occur under different contexts. Based on this concept, security incidents/events may occur under different or similar circumstances-which is the basis of computation on

			<p>how the similarity of two events may be a factor of consideration, hence the concentration on the distance function (d).</p> <p>The distance function (d) between these events X and Y plays a part in showing they may be similar or not based on the context that they appear. For example, Mannila and Moen (1999) has shown that, if a study is conducted on how a website information is provided to users, it is possible that there might exist similarity if two or more websites are giving users exactly the same information.</p> <p>This is because, always there will exist a distance between two events, however, d can only be computable under the following circumstances:</p> $d(e_1, e_2) \geq 0$ $d(e_1, e_2) = 0 \text{ if _and_only_if } (e_1 = e_2)$ $d(e_1, e_2) = d(e_2, e_1)$ <p>This shows that at least $d(e_1, e_2)$ must be computable when</p>
--	--	--	--

			checking the similarity of two events X and Y.
7.30-x	$d(e_1, e_2) \geq 0$ $d(e_1, e_2) = 0$ if _and_ _only_ _if_ $(e_1 = e_2)$ $d(e_1, e_2) = d(e_2, e_1)$	$d(e_1, e_2)$ =Distance between two events	This shows that at least $d(e_1, e_2)$ must be computable when checking the similarity of two events X and Y.
7.31	$ESM = d^{MD}(w_{1TP}, w_{2TP}) = \sqrt[p]{\sum_{i=1}^n w_{1TP} - w_{2TP} ^p}$	ESM=Event Similarity Measure Event Similarity measure=Distance between two events W_{1TP} and W_{2TP}	The distance between two events $d(e_1, e_2)$ is measured as a root of the magnitude.
7.32	$d(w_{ij}) = \sum_{k=1}^n w_{ik} - w_{jk} $	$d(w_{ij})$ = Absolute difference between the pair of event attributes, which is a difference in magnitude	The distance between two events $d(e_1, e_2)$ is measured as a magnitude
7.33	$d(w_{1TP}, w_{2TP}) = \sqrt{\sum_{i=1}^n (w_{1TP} - w_{2TP})^2}$	$d(W_{1TP}, W_{2TP})$ = Root of square differences between the set of event attributes	The distance between two events $d(e_1, e_2)$ are measures as a Root of square differences between the set of event
7.34	$d(w_{1TP}, w_{2TP}) = \max_i w_{1TP} - w_{2TP} $	$d(W_{1TP}, W_{2TP})$ = Absolute difference in magnitude between the set of event attributes	The distance between the events $d(e_1, e_2)$ are measured as Absolute difference in magnitude between the set of event

APPENDIX B. Professional Publications

While conducting this research study, the researcher managed to present and publish a number of professional publications at peer-reviewed international conferences and peer-reviewed scientific journals. Some of the papers presented have formed the basis of the chapters that have been presented in this research thesis.

A research paper on “A Cognitive Approaches for Botnet Detection in the Cloud Environment” was presented in the CyberSec2014 conference at the Third International Conference on Cyber Security, Cyber Warfare and Digital Forensics, in Lebanon, Beirut in May 2014. The paper was published by IEEE Xplore. This was followed by a paper titled, “A Cloud Forensic Readiness Model using a Botnet as a Service” that was presented at the International Conference on Digital Security and Forensics (DigiSec2014), Ostrava, Czech Republic in July 2014. This has been discussed in **Chapter 5** and **7** respectively.

This was then followed by a paper titled, “Obfuscating a Cloud-based Botnet Towards Digital Forensic Readiness”, which has also been discussed in **Chapter 5** and **7** respectively of this research. This presentation was done at the 10th International Conference on Cyber Warfare and Information Security (ICCWS2014) held in Kruger, South Africa in March 2015. After this, a paper titled “A Functional Architecture for a Cloud Forensic Readiness Large-Scale PDE Analysis”. The paper was presentation was done at the 14th European Conference on Cyber Warfare and Security (ECCWS2015), Hatfield, the United Kingdom in July 2015 and this has been discussed in **Chapter 7** of this research thesis.

The next paper was on “Adding Event Reconstruction to a Cloud Forensic Readiness Model” which has been discussed in **Chapter 7** of this research thesis. This paper was presented in August 2015, at the 15th Information Security of South Africa (ISSA) Conference in Rosebank, Johannesburg, South Africa. After this presentation, it was

followed by a paper titled “A Model that Characterises PDE in the Cloud Environment during Digital Forensic Readiness Process”. This was presented at the 3rd International Conference on Cloud Security and Management (ICCSM2015), in Tacoma, University of Washington, USA. This has also been discussed in **Chapter 7** of this research thesis.

A paper on “Requirements for Achieving Digital Forensic Readiness in the Cloud Environment using an NMB Solution” was presented at the 11th international conference on Cyber Warfare and Security, ICCWS2016, in Boston, the USA in March 2016. This was discussed in **Chapter 5** of this research thesis. This was followed by a paper titled, “A prototype for Achieving Digital Forensic Readiness in the Cloud using a Distributed NMB Solution”. This paper was presented at the 15th European Conference on Cyber warfare and Security (ECCWS2016) at the Universitet de Bundeswehr, Munich, Germany in July, 2016. This has also been discussed in **Chapter 9** of this research thesis. Another paper titled, “A Generic Framework for Digital Evidence Traceability” was also presented at the 15th European Conference on Cyber warfare and Security (ECCWS2016) at the Universitet de Bundeswehr, Munich, Germany.

Nevertheless, a number of peer-reviewed scientific journal publications have also been published as a result. A journal paper titled “**On Digital Forensic Readiness in the Cloud Using a Distributed Agent-Based Solution: Issues and Challenges**” has formed a basis for **Chapter 9** of this research thesis. The paper was published in Australian Journal of Forensic Sciences, Taylor and Francis, in June 2016. This was followed by another journal paper titled “**Novel Digital Forensic Readiness Techniques in the Cloud Environment**” this paper formed a basis of **Chapter 6, 7 and 8** respectively and was published in Australian Journal of Forensic Sciences, Taylor and Francis in January 2017.

A list of scientific journals and Conference that have been published have been listed in **in the next section**

APPENDIX B.1 List of Papers Published in Peer Reviewed Journals and International Conferences

B.1.1 Peer Reviewed Scientific Journals

1. **Victor R. Kebande & H.S. Venter (2017): Novel Digital Forensic Readiness Techniques In the Cloud Environment.** In Australian Journal of Forensic Sciences, Published in January 17, 2017.
2. **Victor R. Kebande & H.S. Venter (2016): On Digital Forensic Readiness in the Cloud Using a Distributed Agent-Based Solution: Issues and Challenges.** In Australian Journal of Forensic Sciences, Published In June 2016.
3. **Victor R. Kebande & H.S. Venter (2017): CFRaaS: Architectural Design of a Cloud Forensic Readiness as a Service (CFRaaS) System Using an NMB Solution as a Forensic Agent.** Submitted in International Journal of Information and Computer Security. Inderscience Publishers, United Kingdom.
4. **Victor R. Kebande, N.M Karie & H.S. Venter (2018): Adding Digital Forensic Readiness as a Security Component to the IoT Domain.** In International Journal on Advanced Science, Engineering and Information Technology, published February 2018.
5. **Victor R. Kebande, N.M Karie & H.S. Venter (2018): Functional Requirements for Adding Digital Forensic Readiness as a Security Component in IoT Environments.** In International Journal on Advanced Science, Engineering and Information Technology, published February 2018, published February 2018.

B.1.2 Peer Reviewed International Conferences

1. **Victor R. Kebande & H.S. Venter, “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, 2016.

2. **Victor R. Kebande**, N.M Karie & H.S. Venter, Nickson Karie “**Taxonomy of Digital Forensic Evidence**“, Pan-African Conference on Communication and technology, 2017.
3. **Victor R. Kebande** & H.S. Venter, NM Karie “**A Generic Framework for Digital Evidence Traceability**”, In The European Conference of Cyber Warfare and Security, Bundeswar University, Munich, Germany, 2016.
4. **Victor R. Kebande** & H.S. Venter, “**Requirements for Achieving Digital Forensic Readiness in the Cloud Environment Using an NMB Solution**”, In The International Conference on Cyber Warfare and Security (ICCWS2016), Boston University, USA, 2016.
5. **Victor R. Kebande** & H.S. Venter, “**Towards a Model for Characterizing Potential Digital Evidence in the Cloud Environment During Digital Forensic Readiness Process**”, In The International Conference Cloud and Security Management (ICCSM2015), University of Washington, Tacoma, USA, 2015.
6. **Kebande, V. R., & Venter, H. S.** “**Adding Event Reconstruction to a Cloud Forensic Readiness Model**”. In Information Security for South Africa (ISSA), 2015 (Pp. 1-9). IEEE. Conference Proceeding/IEEE Xplore, Digital Library, 2015.
7. **Victor R. Kebande** & H.S. Venter, “**A Functional Architecture for Cloud Forensic Readiness Large-Scale Potential Digital Evidence Analysis**”, In The European Conference of Cyber Warfare And Security, University of Hertfordshire, United Kingdom, 2015.
8. **Victor R. Kebande** & H.S. Venter, “**Obfuscating A Cloud-Based Botnet Towards Digital Forensic Readiness**”, In The International Conference on Cyber Warfare and Security (ICCWS2015), Kruger, South Africa, 2015.
9. **Victor R. Kebande** & H.S. Venter, “**A Cloud Forensic Readiness Model Using a Botnet as a Service**”, In The International Conference Digital Forensic and Security, Czech Republic, 2014.
10. **Victor R. Kebande** & H.S. Venter, “**A Cognitive Approach for Botnet Detection Using Artificial Immune System in the Cloud**”, In The International

Conference Cyber Security, Cyber Warfare and Digital Forensic (Cybersec2014,
Beirut, Lebanon, Proceeding/ IEEE Xplore Digital Library, 2014.
