A Cloud-based Intrusion Detection and Prevention System for Mobile Voting in South Africa

By

Moloiyatsana Dina Moloja

2018

© Central University of Technology, Free State

A CLOUD-BASED INTRUSION DETECTION AND PREVENTION SYSTEM FOR MOBILE VOTING IN SOUTH AFRICA

Master Dissertation

by

Moloiyatsana Dina Moloja

Supervisor: Dr Noluntu Mpekoa

Co-supervisor: Prof Darelle van Greunen

This dissertation was written and submitted in fulfilment of the requirements for the degree Magister Technologiae:

Information Technology

at the

Central University of Technology, Free State, South Africa

© Central University of Technology, Free State

DECLARATION

I, Ms Moloiyatsa Dina Moloja, student number , declare that the work in this dissertation is my own, original work and has been submitted for the award of Magister Technologiae: Information Technology at the Central University of Technology (CUT), Free State. The dissertation was conducted under the supervision of Dr Noluntu Mpekoa at CUT, Free State and co-supervision of Prof Darelle van Greunen from Nelson Mandela University, Port Elizabeth. I further declare that this work has not been submitted to any other institution of higher education. Wherever contributions of other people are involved, every effort has been made to indicate this clearly by means of references.

Signature: Date: May 2018

© Central University of Technology, Free State

ACKNOWLEDGEMENTS

Firstly, let me take this opportunity to thank my Lord, who is my best Friend, my Keeper, my Saviour, the one King who kept me sane when situations compelled me to lose my peace as I walked this challenging and exciting journey. Thank you, Jesus Christ.

A special thank you to my supervisor, Dr Noluntu Mpekoa, who encouraged me to always do my level best. Throughout this research project, she has been nothing but a friend, a sister in Christ, a mentor, and an inspiration. Whenever I felt like giving up, I would remember her persistence and passion for her work. I am grateful for the sacrifices she made, both personally and professionally. I also truly appreciate my co-supervisor, Prof Darelle van Greunen, for all she has done for me.

Also, I would like to extend my sincere gratitude to my daughter (mommy's Pearl), Olerato Glorious Moloja, who understood, not in so many words, that mommy had to work hard to finish this project; my mother, Alina Moloja, father, Daniel Moloja, brothers Michael and Molete, and my sisters, Mami, Sannah, and Phumelele, for their never-ending support in prayer and their availability whenever I needed them. I could not have done this without them. Their love and words of encouragement kept me going.

I would also like to acknowledge everyone who has assisted me while working on this project, in particular Motoeli John Nyareli, for availing himself whenever I needed his assistance.

They all have significantly contributed towards the completion of this dissertation.

© Central University of Technology, Free State

3.1 LIST OF PUBLICATIONS

The publications contained herein have either been published or accepted for publication. The following publications stemmed directly from the work in this dissertation:

Moloja, D & Mpekoa, N., 2017. Securing M-voting Using Cloud Intrusion Detection and Prevention System: A New Dawn, IST-Africa 2017 Conference Proceedings, Paul Cunningham and Miriam Cunningham (Eds), IIMC International Information Management Corporation, 2017, ISBN: 978-1-905824-57-1, IEEE Xplore.

Moloja, D. & Mpekoa, N., 2017. Towards a Cloud Intrusion Detection and Prevention System for M-voting in South Africa. *IEEE International Conference On Information Society (iSociety 2017)*. Dublin, Ireland, 17 July–19 July 2017, 978-1-908320-80-3.

Moloja, D., Mpekoa, N., & Van Greunen, D., 2018. Cloud Intrusion Detection and Prevention System for M-voting Application in South Africa: Suricata vs Snort. Shahram Latifi (Ed), Las Vegas, Nevada, USA, 16–18 April 2018.

3.1.1 Other publications

In addition to the papers mentioned above, below is a paper that the researcher co-authored which is relevant to this study:

Moloja, D. & Tlale, C., 2017. Cloud Computing: A Paradigm Shift for Central University of Technology. Shahram Latifi (Ed), *Springer*. Las Vegas, Nevada, USA, 10–12 April 2017, pp. 343–347.

© Central University of Technology, Free State

ABSTRACT

Information and Communication Technology (ICT) has given rise to new technologies and solutions that were not possible a few years ago. One of these new technologies is electronic voting, also known as e-voting, which is the use of computerised equipment to cast a vote.

One of the subsets of e-voting is mobile voting (m-voting). M-voting is the use of mobile phones to cast a vote outside the restricted electoral boundaries. Mobile phones are pervasive; they offer connection anywhere, at any time. However, utilising a fast-growing medium such as the mobile phone to cast a vote, poses various new security threats and challenges. Mobile phones utilise equivalent software design used by personal computers which makes them vulnerable or exposed to parallel security challenges like viruses, Trojans and worms.

In the past, security solutions for mobile phones encountered several restrictions in practice. Several methods were used; however, these methods were developed to allow lightweight intrusion detection software to operate directly on the mobile phone. Nevertheless, such security solutions are bound to fail securing a device from intrusions as they are constrained by the restricted memory, storage, computational resources, and battery power of mobile phones.

This study compared and evaluated two intrusion detection systems (IDSs), namely Snort and Suricata, in order to propose a cloud-based intrusion detection and prevention system (CIDPS) for m-voting in South Africa. It employed simulation as the primary research strategy to evaluate the IDSs. A quantitative research method was used to collect and analyse data.

The researcher established that as much as Snort has been the preferred intrusion detection and prevention system (IDPS) in the past, Suricata presented more effective and accurate results close to what the researcher anticipated. The results also revealed that, though Suricata was proven effective enough to protect m-voting while saving the computational resources of mobile phones, more work needs to be done to alleviate the false-negative alerts caused by the anomaly detection method.

© Central University of Technology, Free State

This study adopted Suricata as a suitable cloud-based analysis engine to protect a mobile voting application like XaP.

Keywords: cloud computing, intrusion detection and prevention system, mobile voting,

© Central University of Technology, Free State

TABLE OF CONTENTS

© Central University of Technology, Free State

© Central University of Technology, Free State

© Central University of Technology, Free State

© Central University of Technology, Free State

© Central University of Technology, Free State

LIST OF TABLES

© Central University of Technology, Free State

LIST OF FIGURES

© Central University of Technology, Free State

© Central University of Technology, Free State

ABBREVIATIONS

1G – First Generation

2G – Second Generation

3G – Third Generation

4G – Fourth Generation

Amazon EC2 – Amazon Elastic Compute Cloud

AVD- Android Virtual Device

B-SIPS – Battery-Sensing Intrusion Prevention System

CIDPS – Cloud-based Intrusion Detection and Prevention System

CPU – Central Processing Unit

DoS – Denial-of-Service

DRE – Direct Recording Electronics

E-democracy – Electronic Democracy

E-voting – Electronic Voting

FN – False-Negative

FNR- False-Negative Rate

FP – False-Positive

FPR- False-Positive Rate

GSM – Global System for Mobile Communication

HIDS – Host-based Intrusion Detection and Prevention System

HTTP – Hypertext Transfer Protocol

IaaS – Infrastructure as a Service

ICMP – Internet Control Message Protocol

© Central University of Technology, Free State

ICT – Information and Communication Technology

IDPS – Intrusion Detection and Prevention System

IDS – Intrusion Detection System

IEEE – Institute of Electrical and Electronics Engineers

IP – Internet Protocol

IPS – Intrusion Per Second

LFI – Local File Inclusion

LTE – Long-term Evolution

M-voting – Mobile Voting

NIDS – Network-based Intrusion Detection and Prevention System

OS – Operating System

PaaS – Platform as a Service

PC – Personal Computer

PDA – Personal Digital Assistant

SA – South Africa

SaaS – Software as a Service

SMS – Short Message Service

TCP – Transmission Control Protocol

TN – True-Negative

TNR- True-Negative Rate

TP – True-Positive

TPR- True-Positive Rate

US – United States

© Central University of Technology, Free State

WLAN – Wireless Local Area Network

© Central University of Technology, Free State

CHAPTER 1: INTRODUCTION

© Central University of Technology, Free State

The main focus of this study is on mobile voting security. This introductory chapter is organised in nine sections.

Section 1.1 presents the background of the study. The research problem is described in Section 1.2. The respective objectives of the study are presented in Section 1.3. Section 1.4 presents the design of the study and the methods that were utilised. In Section 1.5, the significance of the study and its contributions are presented. The scope of the study is discussed in Section 1.6. In Section 1.7, the ethical considerations are presented. The structure of the dissertation is presented in Section 1.8. Finally, Section 1.9 summarises the chapter.

1.1 BACKGROUND

Voting and democracy are important components of any democratic process. Democracy allows nations to elect their governments and express their preferences as to how they want to be administered, while voting is a method used to express individual opinion regarding who will lead them for a specific period of time through electoral processes (Achen & Bartels, 2017; Delaune, Kremer & Ryan, 2006; Kalaichelvi & Chandrasekaran, 2012; Kiayias, Korman & Walluck, 2006). The integrity of the voting procedure is vital to the integrity of democracy itself (Ghate, Talewar, Taware & Katti, 2017; Kalaichelvi & Chandrasekaran, 2012).

For decades, South Africa (SA) has been utilising the traditional paper-based voting system, which does not deliver convenience and efficiency as does a mobile voting system. Omitted ballot papers, invalid votes and miscalculations of votes are some of the challenges related to the traditional voting system (Al-Ameen & Talab, 2013; Delaune *et al*., 2006; Goyal, Hemrajani, Sharma, Sharma & Goyal, 2013; Jacobs & Pieters, 2009; Kalaichelvi & Chandrasekaran, 2012; Kiayias *et al.*, 2006; Mpekoa & Van Greunen, 2016). Electronic voting (e-voting) has been drawing a lot of attention and sparking research from all over the world for the past years, for it has some notable benefits over traditional paper-based voting (Mpekoa, 2014; Olusola, Olusayo, Olatunde & Adesina, 2012).

The developments in Information and Communication Technology (ICT) have changed almost every aspect of everyday life (Button, Harrington & Belan, 2014; Jacobs & Pieters, 2009). Modern societies are now fully dependent on ICT for

© Central University of Technology, Free State

commercial, labour, and leisure activities, excluding voting. Using ICT for democratic elections is still quietly in its early stages. The changes in and extensive use of ICT, however, is shifting the way societies observe voting processes and will ultimately change the manner in which they vote. By utilising ICT, specifically mobile phones, traditional voting procedures can be simplified to accept the cost of social assets and time (Ajiboye, Adewole, Jimoh & Oladipo, 2013; Al-Ameen & Talab, 2013; Mpekoa, 2014).

Mobile devices are used in all aspects of life because of the benefits they provide (Mpekoa, 2014). They offer innovative computing and connectivity functionalities, whereas past mobile platforms had restricted functionality (Adigun, Fagbola & Adegun, 2014; Marforio, Jayaram, Soriente, Kostiainen & Čapkun, 2016; Sommers, Yegneswaran & Barford, 2004). Almost all communication and processes (transfer of documents, social networking, online shopping, etc.) can now be carried out through mobile tools, facilitating daily life (Okediran, Olabiyisi, Omidiora & Ganiyu, 2011).

The ability of the mobile phone to provide convenience and flexibility encouraged Mpekoa (2014) to develop a mobile voting (m-voting) system with the South African context in mind. Voting plays a vital part in democracy and m-voting allows voters to use their mobile devices to cast their votes anywhere outside voting stations (Mpekoa, 2014). Nevertheless, using a fast-growing medium such as the mobile phone as a tool to cast a vote may arouse questions when it comes to security challenges (Campbell, Tossell, Byrne & Kortum, 2011; Eilu & Baguma, 2013).

Corresponding to the hasty growth of mobile phone usage, the threat of security attacks is also growing. According to Adigun *et al.* (2014), mobile phones face a wide range of new security challenges, including malicious threats and intrusions, because they are gradually being used to store sensitive personal information such as financial data used for mobile banking. Also, they can now be used as a tool for casting a vote during elections (Eilu & Baguma, 2013; Mpekoa, 2014).

Except for the fact that mobile phones have the capabilities of normal computers, they also have the benefit of being portable. They comprise diverse operating systems (OSs), such as Microsoft Windows, Linux, Android, etc. However, this makes it possible for an attacker to transfer a wide range of different malware from the internet to telecommunication networks (Breitinger & Nickel, 2010; Burguera, Zurutuza &

© Central University of Technology, Free State

Nadjm-Tehrani, 2011; Ghallali, El Ouadghiri, Essaaidi & Boulmalf, 2011). Also, as these ever-present devices utilise the identical software architecture than personal computers, they are vulnerable or exposed to the same security challenges such as viruses, Trojans, and worms (Burguera *et al.*, 2011; Houmansadr, Zonouz & Berthier, 2011; More, Shaikh, Awaskar, Ghongde, Wattamwar & Tadpelliwar, 2015).

A malware attack against an m-voting system may intentionally violate either the secrecy, privacy or the integrity of the voter (Eilu & Baguma, 2013; Fong & Yan, 2008). The mobile phone virus named Cabir spreads through the Bluetooth interface of mobile phones (Houmansadr *et al.*, 2011; Zonouz, Houmansadr, Berthier, Borisov & Sanders, 2013). Another mobile phone security study revealed that Trojans, using voice-recognition algorithms, can steal spoken sensitive information via mobile phones (Ahson & Ilyas, 2017; Houmansadr *et al.*, 2011). Intrusions like that do not only invade the privacy and security of mobile phone users but also succeed in generating synchronized extensive attacks on communiqué infrastructures by creating so-called botnets (Houmansadr *et al.*, 2011; Raja, 2013; Zonouz *et al.*, 2013).

In the past, security solutions for mobile phones encountered several restrictions in practice. Some researchers developed a lightweight intrusion detection software that operates directly on the mobile device. However, such security arrangements neglect to give viable and effective security as they are repressed by the confined memory, storage, computational resources, and battery energy of mobile phones (Houmansadr *et al.*, 2011; Raja, 2013).

Fundamentally, there is a need for a secure and effective security solution for m-voting to be implemented in SA. In this study, a cloud-based intrusion detection and prevention system (CIDPS) for m-voting in SA is proposed. This system aims to identify any entity that attempts to compromise the confidentiality, integrity or availability of m-voting devices.

The next section outlines the problem description of this study.

© Central University of Technology, Free State

1.2 PROBLEM DESCRIPTION

Mobile phones are pervasive and ambitious; they offer connection anywhere, at any time. The mobile phone penetration is increasing every day as it becomes more and more affordable to acquire these devices. This is the reason why m-voting has attracted a lot of attention from researchers and innovators (Eilu & Baguma, 2013; Eilu, Baguma & Petterson, 2014; Thakur, Olugbara, Millham, Wesso & Sharif, 2014).

The advantages of mobile phones providing advanced technology and extra services compared to traditional phones give people the opportunity to have remote access to and be in charge of their information, anywhere and anytime (Burguera *et al*., 2011). However, the fast improvement of technology regarding miniaturisation and computing, predominantly in wireless mobile networks, conveys a new dimension to security threats (Ghallali *et al*., 2011; Marforio *et al*., 2016).

Although these portable devices have limited memory, they are accompanied with the computing and networking power of personal computers (PCs) (Breitinger & Nickel, 2010). Also, they connect through various network technologies such as third generation (3G), Bluetooth, infrared and Wireless Local Area Network (WLAN) or IEEE 802.11, which cause them to become extremely vulnerable to different types of attacks (Burguera *et al*., 2011). Denial-of-Service (DoS) and flooding attacks form part of the real security dangers to internet communication as it disturbs correspondence over the system and blocks network devices to authorised users (Ghallali *et al*., 2011). It has been stated that wireless network devices, for example, mobile phones, are more exposed to these kinds of attacks than wired network devices (Marforio *et al*., 2016).

A few strategies in arranged security recommend “defence-in-depth”, which involves various layers of security around the basic foundation monitoring information; this is viewed as a powerful protection method against attacks (Catania & Garino, 2012). Fusing intrusion detection and intrusion prevention as a defence-in-depth technique is one basic part of system security checking (Lin, Ke & Tsai, 2015). An intrusion detection system (IDS) dissects and records the information passing through a network; if an intrusion is detected, it creates an alarm (Alrajeh, Khan & Shams, 2013).

© Central University of Technology, Free State

This study is concerned with preventing increasing security threats against these mobile devices (Goyal *et al.*, 2013; Kowalski & Goldstein, 2006; Mitchell & Chen, 2014), even more so when they are used to cast votes during the voting process (Eilu & Baguma, 2013; Kalaichelvi & Chandrasekaran, 2012). If SA were to adopt and use m-voting, it is important to determinedly start addressing the current issues and challenges of m-voting systems, including security issues. If left unattended, this could be one substantial threat to citizen participation in m-voting where authorities are unable to ensure the security and secrecy of electronic ballots.

The problem statement for this study is therefore:

There is no cloud security solution, specifically for m-voting, in SA.

1.3 RESEARCH OBJECTIVES

Research objectives are the expected outcomes reached by the researcher at the completion of the research journey (Cameron, 2009; Creswell & Poth, 2017).

The main objective of this study is to compare and evaluate two IDSs in order to propose a suitable CIDPS for m-voting in SA.

To accomplish the main objective mentioned above, the following secondary objectives were pursued:

1) investigating the essential components of a CIDPS;
2) comparing two CIDPSs (analysis engines) and choosing a suitable system for m-voting;
3) linking a cloud-based analysis engine and XaP voting system; and
4) utilising evaluation criteria to test the analysis engines.

© Central University of Technology, Free State

1.4 RESEARCH DESIGN AND METHODOLOGY

A research methodology is a systematic way to solve a problem (Creswell, 2013; Creswell & Poth, 2017; Ivankova, Creswell & Stick, 2006). The techniques that are implemented in this study are based on the research problem, taking into consideration the research questions and objectives already demarcated, and the characteristics of the different research techniques available. The study makes use of three techniques, as depicted in Figure 1.1:

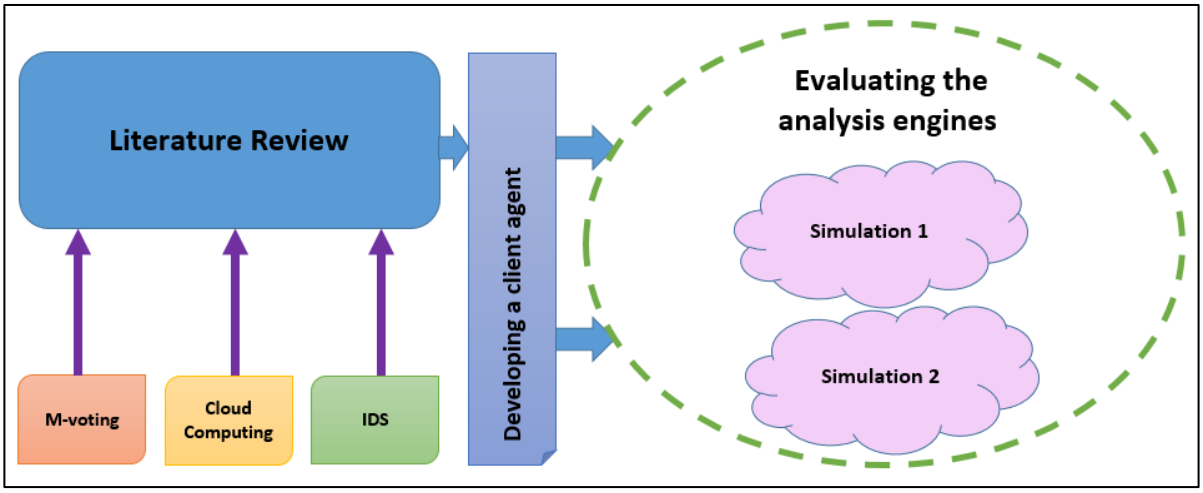


Figure 1-1: Research design

1.4.1 Literature review

A “literature review” is defined as an important summary and assessment of the existing body of recorded work dealing with information produced by other researchers, scholars, and practitioners in a given field (Pickering & Byrne, 2014). It comprises the gathering and appraisal of documentation relevant to a study. Sources may include:

- distributed reports, contemplations, contextual investigations, and so forth;
- conference abstracts, poster presentations, and materials on CD;
- newspaper articles and other media material; and
- any other accessible source of information that is significant and relevant.

© Central University of Technology, Free State

The literature review for this study aides in understanding the different kinds of attacks on mobile phones and discloses the solutions other researchers have developed in order to resolve the security problem. More details are included in Chapter 2.

1.4.2 Developing a client agent

The collected data from the literature study was used to get an understanding of the components needed to simulate a CIDPS. The literature review also assisted the researcher in comparing and evaluating two IDSs, namely Snort and Suricata, for m-voting.

With this information, the researcher developed a client agent. The client agent is the client software running on the mobile phone. The client agent monitors and collects user-sensor inputs and outputs from the device interface in runtime and sends it to the cloud analysis engine to perform an intensive malware scan.

The client agent listens for notifications from the cloud analysis engine and warns the user by displaying a message if a threat is detected and giving instructions on how to deal with the threat. More details on the system are included in Chapter 5.

1.4.3 Evaluating the analysis engines

System evaluation means to establish a subject's importance, substance, and significance in a systematic and rigorous way, according to measures controlled by conventional principles (Archibald, 2016; Blanco, Halpin, Herzig, Mika, Pound, Thompson & Tran Duc, 2011). For this study, the main objective for carrying out the evaluation was to compare the two CIDPSs (Suricata and Snort) and to verify which system is more suitable for m-voting.

To execute a comprehensive evaluation of the proposed CIDP framework/system, appropriate evaluation criteria that address the framework's execution issues were set up. As it is almost impossible to test the proposed system in a real environment, it was evaluated by means of a simulator. More details on the evaluation of the proposed system are included in Chapter 6.

© Central University of Technology, Free State

1.5 SIGNIFICANCE OF THE STUDY

There is inadequate documented research on m-voting in the South African context, and no documented research evidence could be found regarding the comparison of CIDPSs for m-voting. Although some researchers have compared IDPSs for mobile phones, it does not apply to m-voting specifically and most of these systems are not cloud-based.

No research has been done that combines intrusion detection and intrusion prevention techniques with cloud-based computing services for the purpose of m-voting. The researcher's proposed mechanism integrates two essential detection techniques, namely anomaly detection and signature detection. These work together to detect and prevent various kinds of attacks. This research marks therefore the first attempt in utilising a CIDPS for securing m-voting in SA.

The next section presents the scope of the study.

1.6 SCOPE OF THE STUDY

Mobile phones are the focus of both telecommunication networks and internet providers, meaning that mobile phones are associated with both the internet and telecommunication networks (Breitinger & Nickel, 2010; Butler, 2011). Figure 1.3 below illustrates this fact:

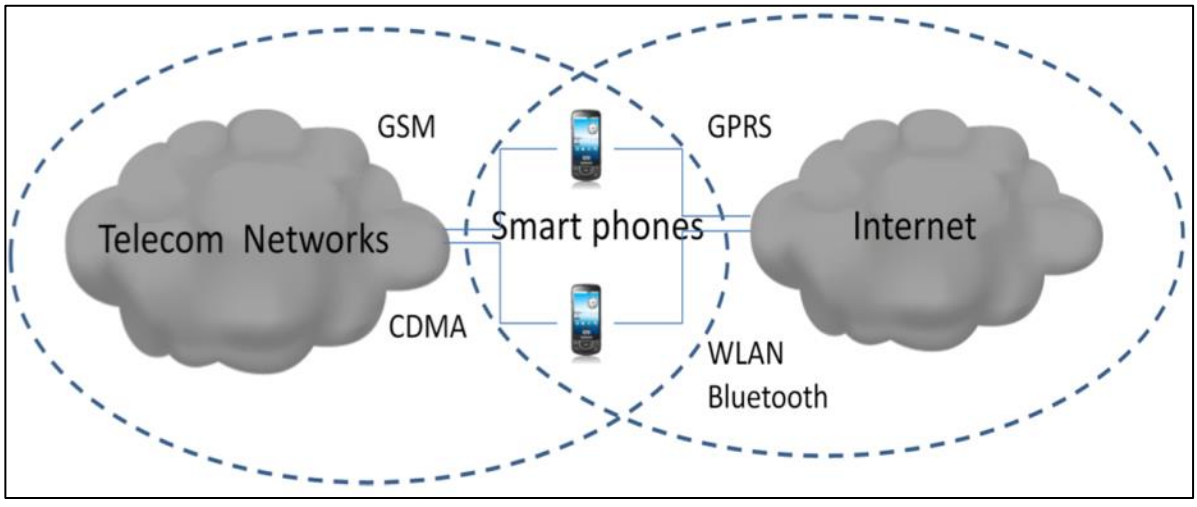


Figure 1-2: Mobile phone connection to the internet and telecommunication networks (Ghallali *et al.*, 2011)

This study does not:

© Central University of Technology, Free State

- deal with the security issues related to the internet connection;
- deal with the security issues related to the telecommunication connection;
- deal with the security issues related to data (e.g., the vote) while being transmitted;
- deal with the security issues associated with voting (e.g., anonymity); but
- focuses on improving the security of the mobile device whilst the user is casting a vote (i.e., protecting the XaP application whilst casting a vote).

The scope of this study is to compare and evaluate two CIDPSs (Suricata and Snort) and to propose the best system for securing m-voting in SA.

1.7 ETHICAL CONSIDERATIONS

Ethical issues were taken into consideration during the course of this entire research process in order to ensure that the results and the final study report truly represent all data and relevant conditions.

Ethics can be defined as behavioural norms and standards that guide moral choices with regard to people's relationships with others (Chang & Ramachandran, 2016). During this research, however, the researcher did not interact with any human participants and this means that none of the human-related ethical guidelines and principles applied.

1.8 ORGANISATION OF THE DISSERTATION

This research attempts to solve the security issues specific to m-voting with regard to mobile phones and their limited computing resources by comparing and evaluating two CIDPSs. These systems are intended to protect the integrity, confidentiality, and secrecy of voters by providing a secure environment while casting a vote via their mobile phones.

Following, is an overview of the impending chapters:

© Central University of Technology, Free State

Chapter 2: Literature review

This chapter gives a summary of the literature which was reviewed. The researcher shows in detail how this study fits in with what has already been done, its significance and how the study leads to new knowledge. In addition, the researcher compares the contents of two IDPSs to determine which one is best for m-voting and chooses the most suitable cloud-based computing service for the proposed security system.

Chapter 3: Proposed security system

This chapter presents related work which entails the IDPSs developed by other researchers for m-voting or other purposes. Also, it gives a brief discussion of the proposed security solution.

Chapter 4: Research design and methodology

This chapter includes a detailed outline of how the investigation was conducted: how the data were collected, which instruments were employed, how the instruments were used, as well as a description of the intended means for analysing the collected data.

Chapter 5: CIDPS components

This chapter discusses the components that are involved in the proposed security solution, as well as their operation.

Chapter 6: System evaluation and findings

In this chapter, various evaluation criteria are used to evaluate Snort and Suricata. Also, the results and findings of the evaluation are interpreted. The discussion in this chapter indicates whether the results of the study confirm – either fully or partially – the researcher's original expectations or predictions.

Chapter 7: Conclusions and recommendations

The study's main points of evidence are summarised in this chapter. Recommendations for future research are also offered.

© Central University of Technology, Free State

1.9 SUMMARY

This chapter has presented the background of the study and described the problem at hand. The CIDPS was briefly discussed and the research objectives were highlighted. The research design and the methodologies used were also considered. The significance of this study was presented, as well as the contributions it has made to the body of existing knowledge regarding this field. Table 1.1 links the chapters with the objectives of this study:

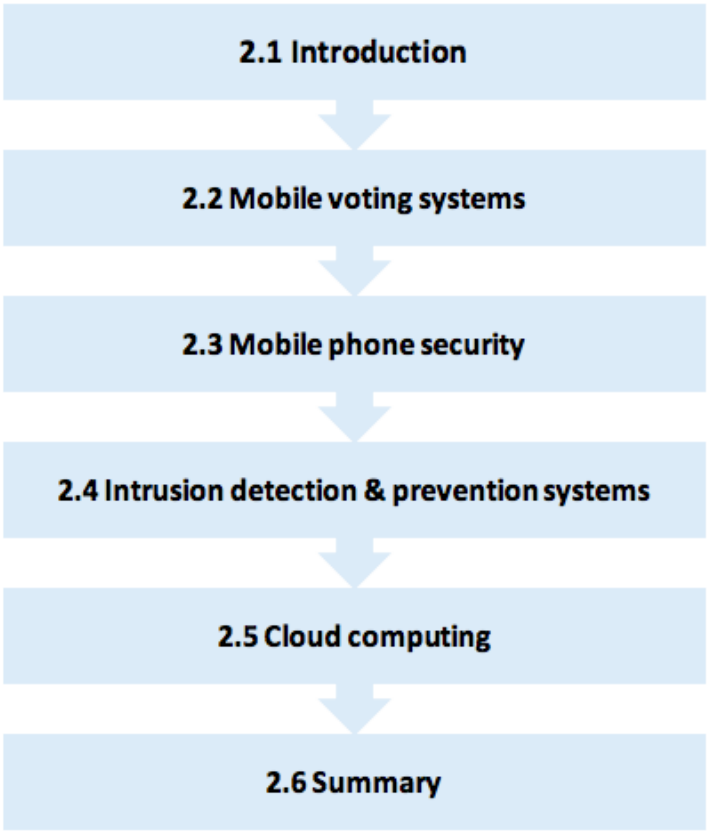
Table 1-1: Linking of chapters with the objectives

Research Objectives	Chapters	Data collection method
1.To investigate the essential components of a CIDPS.	Chapter 2 and Chapter 3	Literature review
2. To compare two CIDPSs (analysis engines) and choose the one that best secure m-voting.	Chapter 5	Literature review and system evaluation
3. To link a cloud-based analysis engine with an XaP voting system	Chapter 6	Literature review
4. To use various evaluation criteria in testing the analysis engines.	Chapter 6	Literature review and system evaluation

In the next chapter, the literature review is thoroughly discussed.

© Central University of Technology, Free State

CHAPTER 2: LITERATURE REVIEW



© Central University of Technology, Free State

This chapter presents the elements that helped in solving the research problem. It discusses the background of cloud computing, m-voting together with mobile devices, as well as IDPSs as indicated in Figure 2.1:

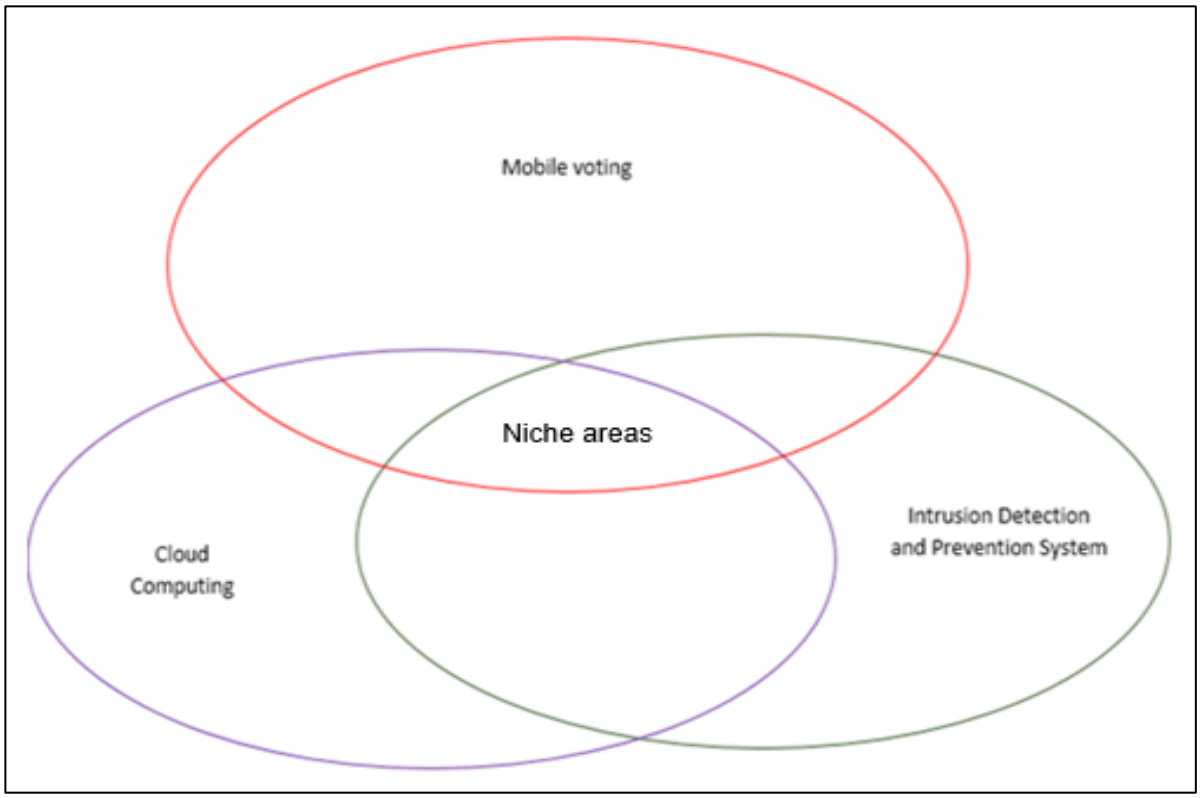


Figure 2-1: Major areas of the study

Section 2.1 defines democracy, ICT, and e-voting. Section 2.2 introduces m-voting and discusses its advantage and disadvantages. In Section 2.3, IDPSs are taken under the lens. Cloud-based computing and the layers involved are explained in Section 2.4. Lastly, the chapter is summarised in Section 2.5.

2.1 INTRODUCTION

2.1.1 Democracy and ICT

The vital task of democracy is to engage citizens in the government of the country (Mpekoa, 2014). Democracy is a system used by the government to listen to the people's voice; it is a process in which the power is entrusted to people and used directly by them under a free democratic system (Okediran *et al*., 2011; Persson, Sundell & Öhrvall, 2014). Voting, on the other hand, is one of the most vital tasks of democracy. Not only does voting deliver a systematic transfer of power, but it

© Central University of Technology, Free State

strengthens citizens' trust and confidence in government (Okediran *et al*., 2011; Olusola *et al.*, 2012).

The history and tradition of democracy and elections go back more than 2 500 years. Recently, however, technology has influenced the way elections are approached (Okediran *et al.*, 2011; Rubner, 2012). The use of ICT to help smooth democratic processes has offered opportunities to shape the future of democracy.

Furthermore, using ICT in governments has offered opportunities to address and adapt to broadening the understanding of political representation, its transparency, participation, and accountability (Persson *et al*., 2014). Additionally, the use of ICT in a democracy can assist in creating an interactive structure for interconnectivity, service delivery, efficiency and effectiveness, interactivity, decentralisation, and liability.

According to Magomelo, Mavhemwa and Ndumiyana (2013), the use of ICT has presented citizens with possibilities for additional involvement and engagement in the democratic procedure. The best, most effective way to have citizens involved in democracy and voting is through the use of ICT (Akonjom & Ogbulezie, 2014; Gibson, Römmele & Ward, 2004).

The introduction of ICT to democracy has birthed what is called electronic democracy (e-democracy) (Gibson *et al.*, 2004). Using ICT to cast electronic votes is known as e-voting (Okediran *et al.*, 2011).

2.1.2 Electronic voting

According to Okediran *et al.* (2011), ICT has given rise to new technologies and solutions that were not possible a few years ago. One of these new technologies is e-voting, which is a voting method whereby the electorate's votes are collected and counted by electronic means (Al-Ameen & Talab, 2013; Okediran *et al.*, 2011). Recently, e-voting has attracted a lot of attention (Al-Ameen & Talab, 2013). Many researchers believe that e-voting is better than manual voting because it is more effective and proficient (Ahmad, Shanmugam, Idris & Samy, 2013a).

The implementation of e-voting in other countries has unquestionably enabled those voters to cast their votes, regardless of where they are situated. E-voting has opened

© Central University of Technology, Free State

up gates to voters with infirmities or those who experience problems in being physically present at the poll site. It has the potential to increase voter turnout by offering an additional voting channel (Al-Ameen & Talab, 2013; Mpekoa, 2014; Okediran *et al*., 2011).

Many governments have recently begun to introduce e-voting in their voting processes (Ansper, Heiberg, Lipmaa, Overland & Van Laenen, 2009; At, Burkart & Lee, 2011). In the United States (US), various kinds of e-voting devices have been used for some time now. Estonia carried off internet voting during two parliamentary elections, and various Swiss referendums was run using the internet (Al-Ameen & Talab, 2013).

Trust in e-voting technology is fostered when the operational system of the technology conforms to high security criteria (Eilu & Baguma, 2013; Eilu *et al.*, 2014). E-voting has been applied, either on small or large scale, to a real environment in countries such as Estonia and Namibia, and it has been scrutinised based on the security challenges it contained (Ahmad, Musa, Nadarajah, Hassan & Othman, 2013b; Al-Ameen & Talab, 2013). However, the concept has faced opposition, despite the benefits voters and authorities can expect from using it.

It appears that security is the fundamental concern and the reason why some countries find it hard to implement e-voting systems (Ahmad *et al*., 2013b; Thakur *et al.*, 2014). According to the study conducted by Thakur *et al*., (2014), the challenge that e-voting faces is how to ensure that the technology (mobile phone or personal computer) with which voters cast their vote, is secure. This is due to the threat that malware or other intrusions pose to the integrity of the elections, as well as the privacy of the voter. The situation is exacerbated when the e-voting system is remote, as this factor increases the security vulnerabilities. Those vulnerabilities include the risk of malicious software on the user's device, malicious network nodes, and the selling of votes (Thakur *et al*., 2014).

Literature indicates that there are many different types of e-voting systems, which will be briefly discussed in the next section.

© Central University of Technology, Free State

2.1.3 Background on electronic voting systems

Some e-voting systems involve the use of lever arch machines, direct recording electronics (DRE), punched cards, optical scanning, and remote e-voting (Al-Ameen & Talab, 2013; Thakur *et al*., 2014). E-voting systems are divided into two categories, namely: A) e-voting systems employed in controlled (organised) environments, and B) e-voting systems implemented in uncontrolled environments (remotely). See Figure 2.2 below.

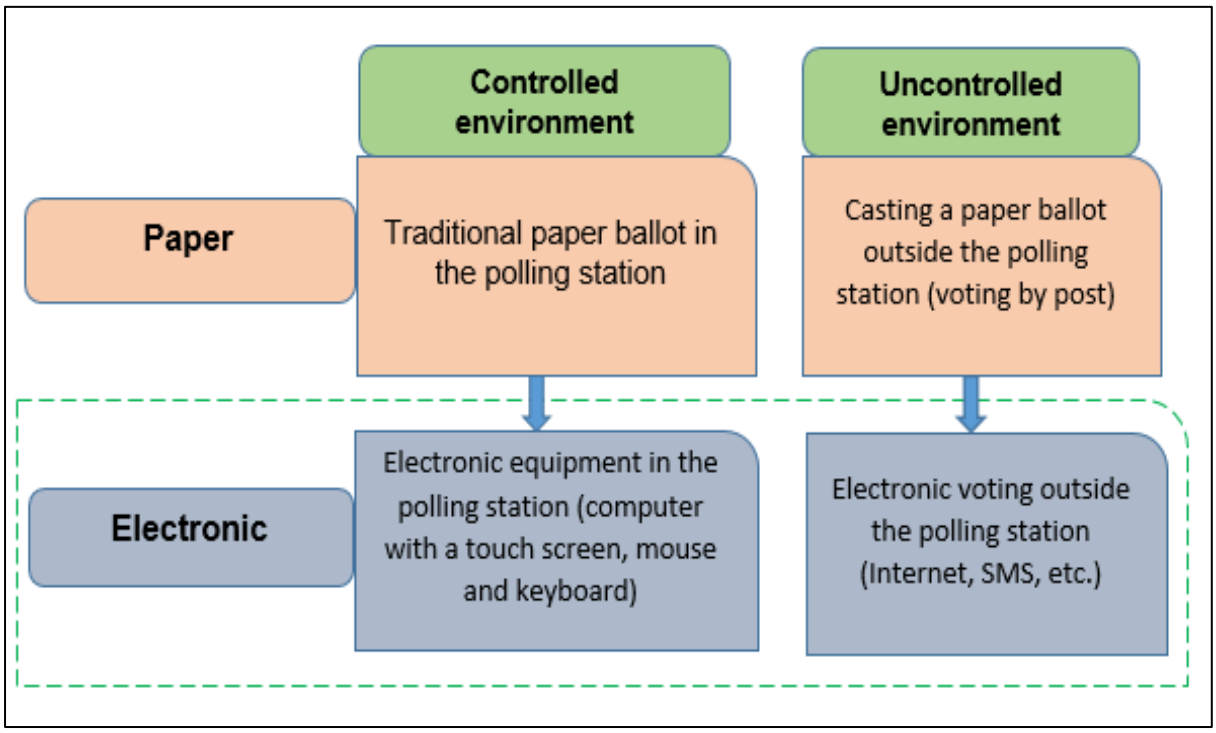


Figure 2-2: Types of e-voting systems (Rubner, 2012)

A. E-voting systems implemented in controlled environments

1. Lever arch machines:

The Australians used this type of voting system specifically to move away from the paper-based voting system. Myers Automatic Booth was the first lever arch voting machine to be offered in New York in 1892. It worked as follows: the voter entered the voting booth and pulled a handle which closed the curtains of the booth and also unlocked the voting levers. A voting ballot with a small switch near the name of each candidate was displayed and the voter made a choice by flipping this little lever next to the name of the favoured applicant. When the voter was satisfied with his/her

© Central University of Technology, Free State

decision, he/she would pull a large lever to cast the vote and opened the curtain (Kitsing, 2014; Mpekoa, 2014; Persson *et al.*, 2014).

2. *Direct recording electronics (DRE):*

   This system records votes with the assistance of a ballot display furnished with mechanical or electro-optical parts, for example buttons or a touch screen. Voters cast their votes using these and the voting data and ballot images are stored within the system's memory bank. DRE are still being used today as part of the voting processes of several nations. Notwithstanding, the absence of a paper trail when using this method makes auditing impossible (Delaune *et al.*, 2006; Goyal *et al.*, 2013; Kalaichelvi & Chandrasekaran, 2012).

3. *Punch card machines*:

   With punched card systems, a card and a little clipboard-sized device are used to record votes (Ansper *et al.*, 2009). The voters poke through the perforations on the card which match their choices and the card is put inside a box. The votes are then counted with a tabulation machine that scans the ballot card based on the passage of light through the holes (Okediran *et al.*, 2011).

4. *Optical scanning*:

   Electronic optical mark scanners have been designed and developed to register votes (Campbell *et al*., 2011). The voters cast their votes by marking the box to be rectangle, circle or bolt next to their favored applicant's name (Cetinkaya & Doganaksoy, 2007).

5. *Kiosk voting:*

   In this voting system, voting workstations are tamper-resistant and are situated in easily accessible and frequently visited places such as malls or schools. Voting takes place under the supervision of election officials and observers, and cameras are also used to ensure security, privacy, and the prevention of intimidation or other methods of interference with the voting process (Ahmad *et al*., 2013b).

© Central University of Technology, Free State

B. E-voting systems implemented in uncontrolled environment

Remote electronic voting:

This system is not under the supervision of any government authority representatives and involves voting from a voter's own or another person's computer via the internet or mobile phones (Al-Ameen & Talab, 2013). Remote e-voting includes the following two processes, namely:

- Internet voting enables a voter to vote via the internet using computers – this process tries to provide accuracy and security. Vulnerabilities of internet voting include DoS, spoofing and man-in-the-middle attacks (Ahmad *et al*., 2013a; Okediran *et al.*, 2011). Because voting can be done outside supervised locales, coercion and intimidation are inevitable (Akonjom & Ogbulezie, 2014).
- Mobile phone voting: Sometimes referred to as m-voting, it involves the ability of voters to cast their votes using a mobile phone without the supervision of electoral staff. Mobile phone voting allows the electorate to vote anywhere via the readily available Global System for Mobile Communication (GSM) network (Mpekoa, 2014; Olusola *et al*., 2012). Additionally, there are many ways in which a mobile phone can be used to cast a vote, for example by means of the Short Message Service (SMS) or using a third party voting application (Eilu *et al.*, 2014).

The next section discusses mobile voting systems in detail.

2.2 MOBILE VOTING SYSTEMS

2.2.1 Why mobile voting?

The ability of people to connect and work anywhere at any given time while on the move has given rise to major developments in mobile device penetration (Mpekoa, 2014; Rubner, 2012). After almost 40 years of development, mobile devices have evolved from pure telecommunication devices to small and ever-present computing platforms (Thakur *et al.*, 2014). With more than six billion mobile devices presently in use worldwide, it has become a vital part of daily life. These devices are capable of performing multiple computational tasks and offer different graphical communication user-interfaces that enable people to have universal access to a large diversity of

© Central University of Technology, Free State

services (Magomelo *et al.*, 2013; Mpekoa, 2014; Rubner, 2012; Thakur *et al.*, 2014; Titova & Talmo, 2014).

The growing influence of ICT, particularly mobile phone technology, on numerous areas of life has been observed before (Mpekoa, 2014), but a comprehensive analysis of its potential effects on government has only started recently (Eilu & Baguma, 2013). Whilst the potential of e-voting is still being investigated, the political impact of mobile phones can be seen in the broader context of democracy, for example, the mobilisation of activists (Ajiboye *et al*., 2013). Furthermore, any developing country wishing to cut costs and increase the participation of citizens in decision-making through ICT, should take advantage of the opportunities provided by mobile phones (Eilu & Baguma, 2013). Mobile phones are used everywhere and they can now be used as a tool to cast votes during elections in what is known as m-voting (Eilu *et al*., 2014).

The literature reviewed for this study reveals that m-voting is regarded worldwide as an attractive way of encouraging participation, particularly from among the youth (Eilu & Baguma, 2013; Mpekoa, 2014). Additionally, m-voting does not only offer mobility and flexibility to its users, but it also reduces logistical and administrative costs; the process of casting and counting votes is faster and more accurate, and it increases accessibility for the old and disabled (Eilu & Baguma, 2013; Mpekoa, 2014).

Despite all the opportunities it offers, m-voting has its own challenges. These challenges include security threats to mobile phones, as well as the limited resources (e.g., memory and processor) of these devices. The limited resources of mobile phones hinder the development of effective security systems for m-voting (Goyal *et al*., 2013; Kalaichelvi & Chandrasekaran, 2012).

2.2.2 Mobile phones as a device for voting

In general, the expression “mobile phone” refers to any handheld device with advanced computational abilities. This includes mobile phones, netbooks, and game machines, to mention just a few (Ghallali *et al.*, 2011). For the purposes of this study, a mobile phone refers to a modern mobile phone or smartphone which is a blend of three computing devices, namely a mobile phone (or cellular handset), an all-inclusive computing platform, and a web communicator (Harris & Patten, 2014).

© Central University of Technology, Free State

Individuals use mobile phones because of the various data services they offer such as text messaging, internet browsing, document editing, data storage, electronic games, and banking in addition to the usual voice services (He, Chan & Guizani, 2015).

Mobile devices are constantly getting more portable, affordable, convenient, and potent, and are able to deliver an abundance of advanced data input interfaces, enabling the user to interact with the device more effectively (Portokalidis, Homburg, Anagnostakis & Bos, 2010; Rahimi, Ren, Liu, Vasilakos & Venkatasubramanian, 2014). Because mobile devices are portable, easy to use and technologically advanced, their numbers are increasing day by day. As a result of this rapid growth, their operation systems have undergone major development and improvement, meeting consumer demands (Burguera *et al.*, 2011; Malisa, Kostiainen & Capkun, 2017; Rahimi *et al*., 2014).

Figure 2.3 compares the total number of computer and mobile phone owners in South Africa:

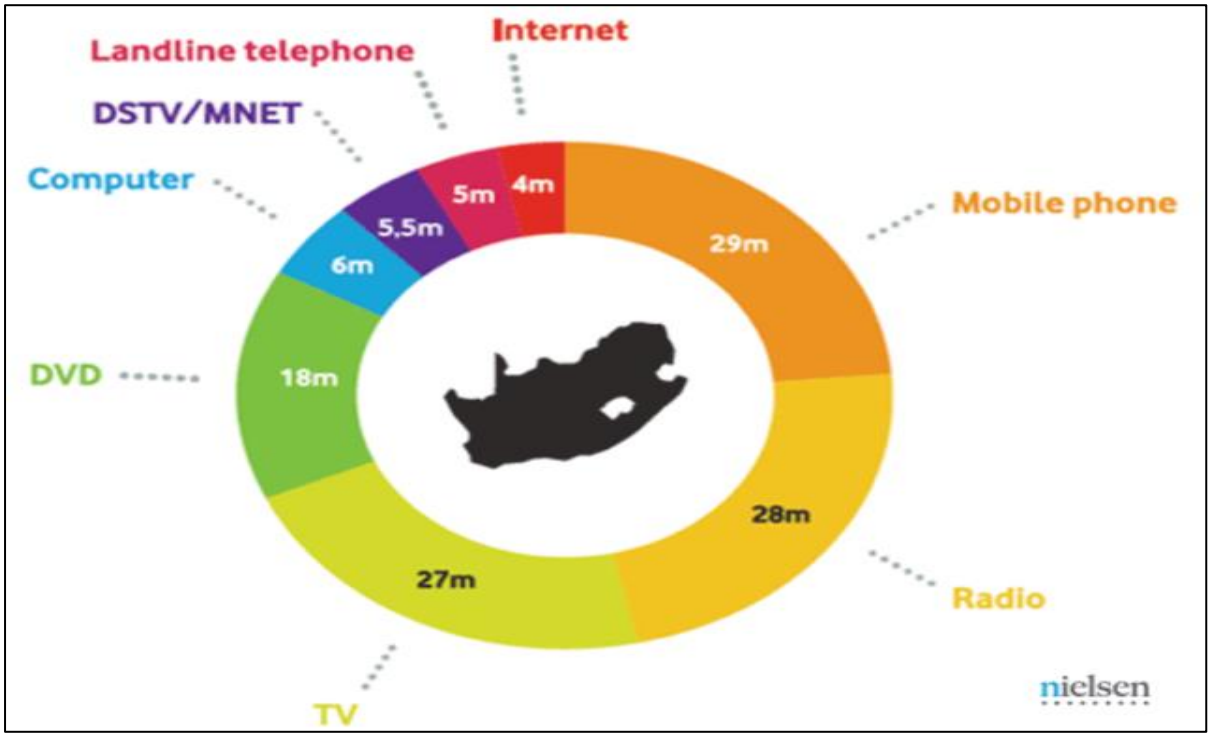


Figure 2-3: Computer and mobile phone ownership (Malisa *et al.*, 2017)

Although mobile phones are not as powerful as desktop or laptop computers, they can store huge amounts of data and provide effective services at the same time, such as

© Central University of Technology, Free State

internet sharing via tethering and intelligent voice assistants (Rahimi *et al.*, 2014; Raja, 2013). Moreover, they are increasingly used to cast votes during elections these days (Mpekoa, 2014).

Mobile phones work the same way as PCs as far as system administration or system networking, processing power, and data capacity are concerned which make them vulnerable to similar classes of attacks and threats than those facing PCs (Portokalidis *et al.*, 2010; Rahimi *et al.*, 2014; Raja, 2013).

2.2.3 Mobile phone networks

Mobile phones were initially intended to offer telephonic utilities by means of a cellular network (Butler, 2011). The first generation (1G) cellular network was set in motion in Japan in 1979, offering individuals the chance to interconnect with each other over the air waves using portable handsets only (Gandhewar & Sheikh, 2010).

Despite the fact that the first cellular communication systems used simple circuits and were very expensive, the second generation (2G) cellular network arrived on the scene 11 years later and gradually substituted the 1G. The 2G cellular network made use of computerised circuit switching technology, delivering both voice and data services (Portokalidis *et al.*, 2010). Out of these data services, the SMS made its appearance and mobile users now had the opportunity to send each other short instant messages, or “texting”, as it became known (Rahimi *et al.*, 2014).

A popular generation used to date is the third generation (3G); this generation features improved wireless technologies, such as high-speed transmissions, advanced multimedia access, and global roaming (Zonouz *et al*., 2013). 3G is generally used with mobile devices and handsets as a way of connecting to the internet or other internet protocol (IP) networks in order to make voice and video calls, downloading and uploading data, and surfing the internet (Rahimi *et al*., 2014; Raja, 2013; Zonouz *et al*., 2013).

Lastly, but not the least, the long-term evolution (LTE) network arrived in 2009 and became enormously popular among end users. Even though it was registered as fourth generation (4G) technology, its bandwidth does not meet the requirements of

© Central University of Technology, Free State

the fourth generation, so it is also known as pre-4G technology (Portokalidis *et al*., 2010; Rubner, 2012; Singh & Thapar, 2012).

2.2.4 Mobile device evolution and architecture

A comparison between the features of PCs or laptops and those of mobile phones reveals that the latter has a much more compressed architecture; the space in which it is constructed is very small, providing little access to the integrated circuits, and there is also a battery size restriction (Malisa *et al*., 2017; Raja, 2013; Zonouz *et al*., 2013).

In addition, several communication interfaces such as Wi-Fi, 3G/LTE, and Bluetooth are integrated into a single device, as depicted in Figure 2.4:

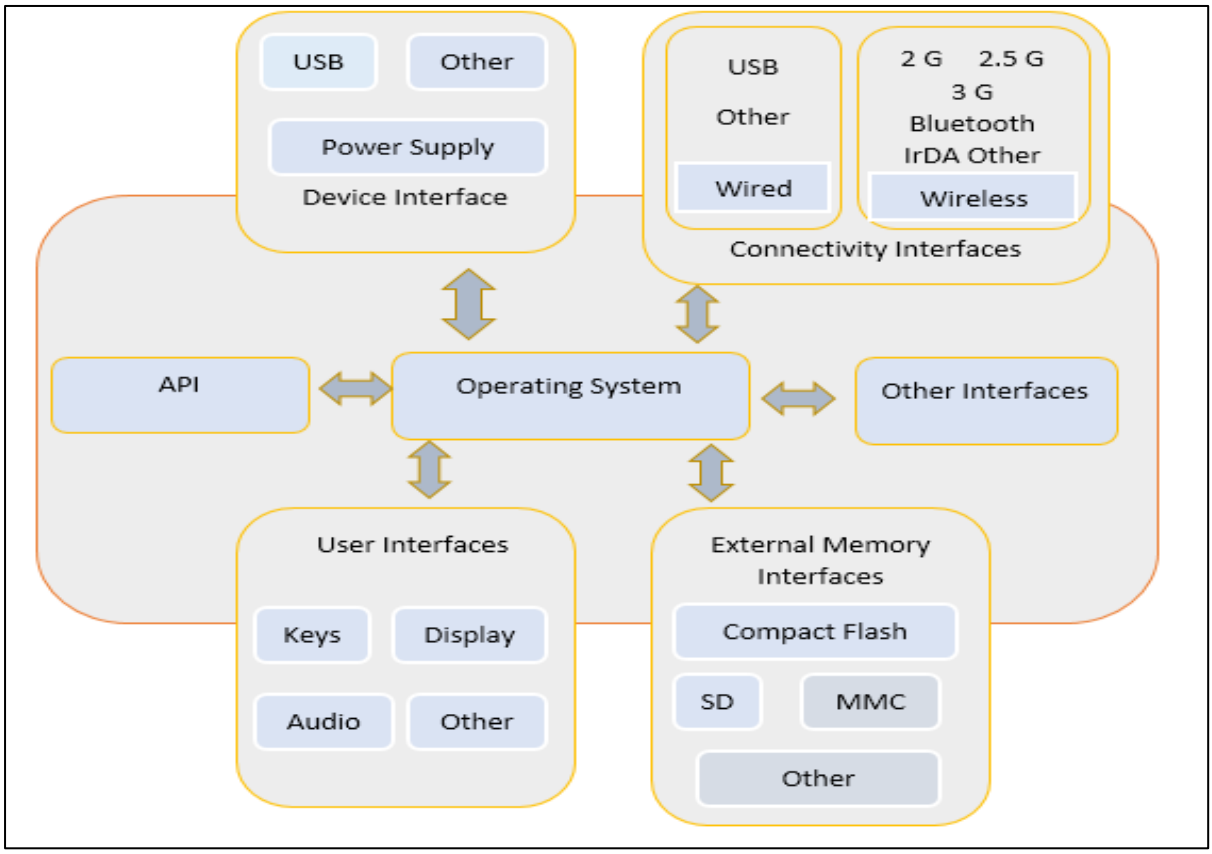


Figure 2-4: Architecture of a modern mobile device (Malisa *et al*., 2017)

Mobile phone devices have, without a doubt, overlapped the capabilities of a PC. They now boast a myriad advanced features enabling users to make fruitful use of their devices in a huge variety of ways (Malisa *et al*., 2017; Singh & Thapar, 2012). Typical innovative features are, amongst others: software keyboards displayed on a touchscreen instead of physical ones; cameras on the front and back of devices for

© Central University of Technology, Free State

video calls; and gyroscopes and magnetometers for computing or determining and maintaining the orientation of the device (Ghallali *et al*., 2011).

Another important point is the ease of access offered by mobile phones; users can be called at any given time or obtain information via any of the different wireless interfaces.

The most significant difference between laptops and mobile phone devices is the input interface of the mobile device (Hussain & Mkpojiogu, 2015). Formerly, mobile device users often operated their devices with PC-like hardware QWERTY keyboards in order to increase typing speed, or used a stylus-pen to import data or commands from displays on the interface. However, since the appearance of the first iPhone by Apple in 2007, most contemporary mobile devices today come with large touchscreens which are used as both output and input interfaces (Rahimi *et al*., 2014).

These revolutionary developments have changed the way how mobile devices are produced (Chin, Felt, Sekar & Wagner, 2012; Hussain & Mkpojiogu, 2015). The following section discusses the security of mobile devices.

2.3 MOBILE PHONE SECURITY

Kitsing (2014) states that the integrity of elections is equal to the integrity of democracy; thus, election systems must be designed in such a way that it can withstand any intrusion intending to harm it.

According to the study conducted by Al-Ameen and Talab (2013), every aspect of elections, especially the voting system, is important in terms of security. If the voting system is not planned and designed accordingly, it can undermine confidence in the whole electoral process.

Also, an election system must be adequate and understandable in order for voters, as well as the candidates to be voted for, to accept the results of the elections as they are presented (Okediran *et al.*, 2011; Olusola *et al*., 2012).

It would be a massive mistake to overlook the significance of security in elections, because the future of a country and the world rests on the public assurance that society has the power to choose their own government. Therefore, any intrusion that

© Central University of Technology, Free State

threatens the integrity of elections, or even the perceived integrity of the system, should be prevented at all costs (Ahmad *et al.,* 2013a; Magomelo *et al*., 2013; Olusola *et al*., 2012).

It is of the utmost significance to protect the efficiency, reliability, confidentiality, and security of elections, as well as the technology that is involved.

2.3.1 Mobile phone security vulnerabilities

As described above, current mobile phone devices provide many of the functions previously delivered by traditional personal computers. There are also these days many connectivity options, for example, IEEE 802.11 – overseen by the Institute of Electrical and Electronics Engineers (IEEE) – which represents wireless networks such as Wi-Fi and Bluetooth (Breitinger & Nickel, 2010; Malisa *et al*., 2017).

However, mobile devices also face a wide range of security challenges, including malicious threats and intrusions, for example, phishing attacks (Adigun *et al*., 2014; Zonouz *et al.*, 2013). Phishing attacks are an example of generic intrusion that intends to steal a user's logging information for a certain website (e.g., an online banking account) by mirroring the privacy and security measures of a mobile phone (Anirudha, Honale, Dhande & Chaudhari, 2013; Zonouz *et al.*, 2013).

The study conducted by Anirudha *et al*. (2013) discovered that Android-based mobile phones have a market share of around 50%. This fact has given growth to new types of attacks and penetration techniques aimed at these mobile devices. In fact, a large number of malware and viruses have been specifically developed to exploit vulnerabilities in such devices (Shahbazi, 2013), therefore this study will be focused on Android-based mobile phones.

Malware is a very big danger in the present technological world (Eilu & Baguma, 2013). It refers to a malicious software program that is intended to harm mobile devices such as smartphones, tablets, and Personal Digital Assistants (PDAs) (Suo, Liu, Wan & Zhou, 2013). A malware attack aimed at a mobile voting system may typically attempt to violate the secrecy and integrity of the vote, as well as the security of the system that is used (Shabtai, Tenenboim-Chekina, Mimran, Rokach, Shapira & Elovici, 2014).

© Central University of Technology, Free State

In Figure 2.5 below, it can be seen that the occurrence of malware has dramatically increased, especially two malware programs known as Trojan-Ransom and Trojan-Spy. A Trojan is a type of malware that is frequently concealed as legitimate software (Harris & Patten, 2014; Li, Ma & Guan, 2017). Trojans can be employed by cyber-thieves and hackers trying to obtain access to users' systems. Users are usually cheated by some form of social engineering into loading and executing Trojans on their systems (Malisa *et al*., 2017). Once it is activated, it can permit cyber-criminals to spy, steal sensitive data, and gain backdoor access to the user's systems (Batyuk, Herpich, Camtepe, Raddatz, Schmidt & Albayrak, 2011; Malisa *et al.*, 2017).

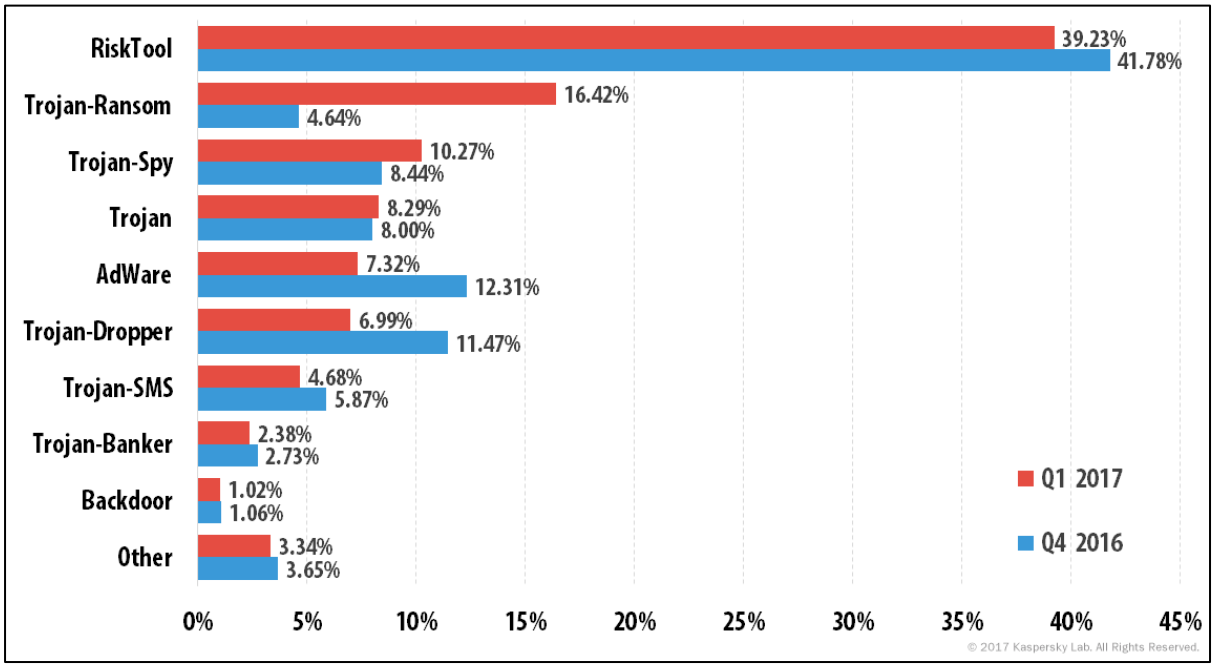


Figure 2-5: Mobile threats statistics (Li *et al.*, 2017)

A wide range of malware has been detected and the quantity of this malicious software is increasing annually. According to Li *et al*. (2017), the number of known malware has increased to 7.10 million since 2016 because of smartphone devices' OSs that allow constant connection to the internet. Studies conducted by Anirudha *et al*. (2013) and Malisa *et al*. (2017) revealed that malware can:

1. gain access to an information system, record and transfer data from the system to a third party without the user knowing;
2. disguise information; and
3. disable the system's security measures.

© Central University of Technology, Free State

In line with the growth of mobile devices, the malicious software industry is also soaring in both technological and structural terms. Figure 2.6 below highlights the threats and weaknesses of mobile OSs.

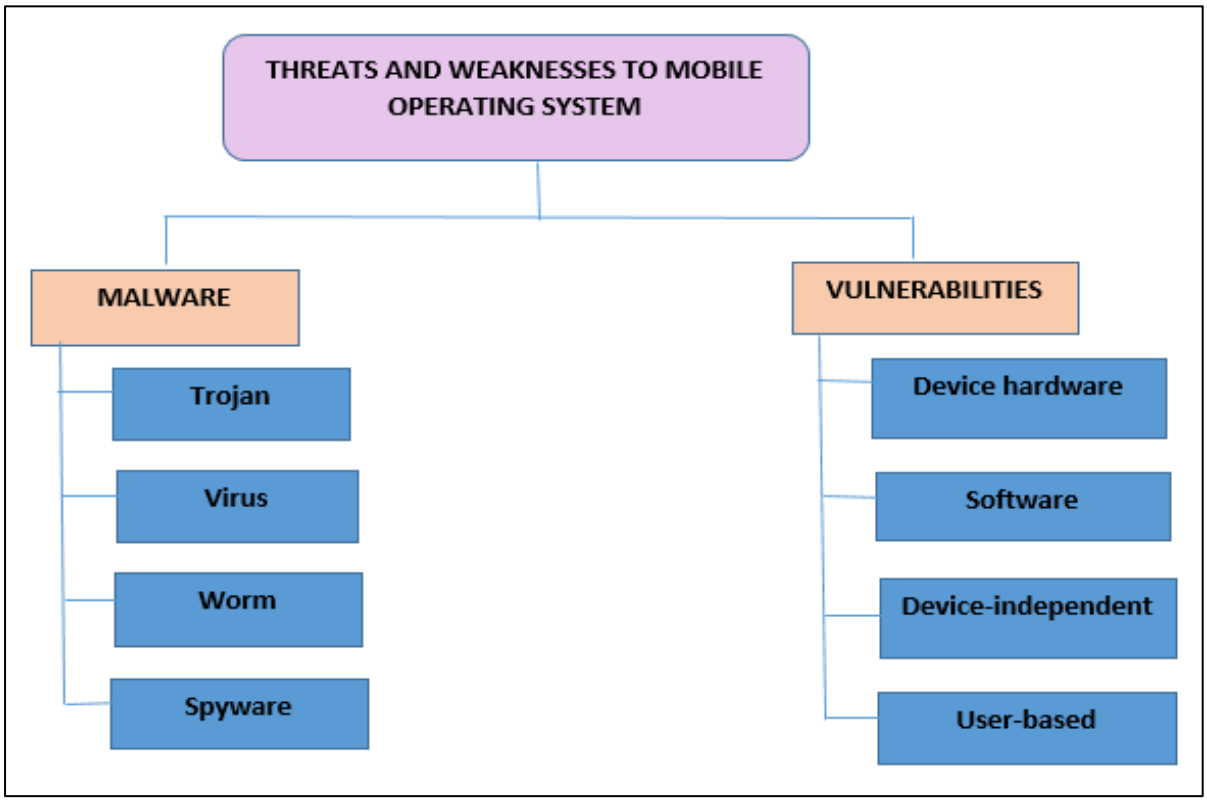


Figure 2-6: Threats and weaknesses of a mobile OS (Li *et al.,* 2017)

Following is a brief description of the malware types listed in Figure 2.6:

1. **Virus:** a piece of code that can duplicate itself. Different replicates of a virus can infect other programs, boot areas or files by incorporating or attaching itself to them (Harris & Patten, 2014; Kushwaha & Kushwaha, 2011; Li *et al*., 2017; Shabtai *et al*., 2014).

2. **Worm:** a program that reproduces itself and spreads to other devices using different transport mechanisms without any user intervention. For instance, the worm Cabir spreads through the Bluetooth interface of mobile phones (Harris & Patten, 2014; Kushwaha & Kushwaha, 2011; Li *et al*., 2017; Suo *et al.*, 2013).

3. **Trojans:** these are malicious programs which are not able to self-replicate, but they perform actions that have not been authorised by the user. They always require user intervention to be activated. Once triggered, the malware can cause

© Central University of Technology, Free State

serious harm to the device by contaminating and disabling other applications or the phone itself (He, Chan & Guizani, 2015; Li *et al*., 2017; Shabtai *et al*., 2014).

4. **Spyware:** this malware threatens mobile device users by collecting, monitoring, using, and distributing the user's personal or private information without the user's permission and awareness (He *et al.*, 2015; Shahbazi, 2013).

Vulnerabilities can be classified in numerous ways. A classification made by He *et al.* (2015) and Shabtai *et al*. (2014) groups attacks towards mobile devices into four main categories. These categories are: device hardware, software-based, device-independent, and user-based vulnerabilities. The definition of each category is explained below:

1. **Hardware-based vulnerability**: Hardware-based vulnerabilities establish a broad component of mobile security. Attacks cannot expose user information, however, there is access to the device which results in putting the information contained on the device at risk (Shabtai *et al.*, 2014).

2. **Device-independent**: According to Wang, Zheng, Lou and Hou (2015), these types of vulnerabilities expose mobile devices to attacks which are aimed directly at the mobile device user. It intends to disrupt the privacy of the user's personal data through wireless connection or wiretapping.

3. **Software-based**: This type of vulnerability creates backdoors into the device through the use of third-party software or spyware that the user has unintentionally downloaded from the internet or by opening attachments to a malicious email (He *et al.*, 2015; Shabtai *et al.*, 2014; Soldani & Manzalini, 2015).

4. **User-based**: This type of vulnerability means the mobile phone device is not open to malicious software but to attacks which are launched through "social engineering". These attacks are aimed at extracting private information and occur often (Suo *et al*., 2013). For example, DoS attacks are not directed through applications or malware installed on mobile phones, but are using the weaknesses produced by distorted text messages (He *et al.*, 2015).

In addition to these vulnerabilities or attack vectors, there are also other kinds of attacks. Nevertheless, the aim of all attacks is to discover the victim's vulnerabilities

© Central University of Technology, Free State

and to take action using a well-thought-out process and application (Suo *et al*., 2013). It is therefore vital to ensure a secure mobile phone device in order for mobile voting to take root (He *et al.*, 2015; Shabtai *et al.*, 2014; Soldani & Manzalini, 2015).

2.3.2 Mobile phone protection

It is worth noting that the development of mobile phones does not come without concerns, as these devices become more and more personal and are increasingly used to store personal and sensitive information of users (Harris & Patten, 2014; Li, Dai, Ming & Qiu, 2016). As a result, it is imperative to avoid information from being stolen by securing either the device itself or the mobile OS that runs on it.

The fact that these devices are continually improved with better functionality and are so widely used is one of the main reasons why there is such a substantial rise in the number of malware that targets the OSs of these devices (Gai, Qiu, Zhao, Tao & Zong, 2016; Li *et al*., 2016; Rahimi *et al*., 2014). Therefore, mobile device security is an emerging need.

Information security and m-voting security encompass five principles listed in Table 2.1 below. Mobile device security refers to the importance of the physical mobile device (hardware and software) complying to these principles (Li *et al*., 2016; Rahimi *et al*., 2014).

© Central University of Technology, Free State

Table 2-1: Description of major mobile voting security principles

Security Principles	Description
Confidentiality	Preventing the disclosure of the user's information to unauthorised individuals or systems and protecting privacy and proprietary information (Ansper *et al.*, 2009; Eilu & Baguma, 2013).
Integrity	Systems or mechanisms cannot be modified or unauthorised, and must be free of corruption (Eilu *et al.*, 2014; Goyal *et al.*, 2013).
Availability	Warrants timely and steady access to any system or information when it is required (Kalaichelvi & Chandrasekaran, 2012; Kitsing, 2014).
Authenticity	Warrants that data, transactions, communications or documents are genuine and original (Kitsing, 2014; Okediran *et al.*, 2011).
Non-repudiation	Implies that neither party can deny having participated in a sending or receiving transaction (Eilu & Baguma, 2013; Eilu *et al.*, 2014).

A trustworthy m-voting system for SA is crucial in order to gain the confidence of the population, as democracies are built on this confidence (Eilu *et al.*, 2014). As a result, this study introduces a CIDPS as a security solution for m-voting in SA.

The following section discusses IDPSs.

2.4 INTRUSION DETECTION AND PREVENTION SYSTEMS

IDPSs have become valuable tools in keeping information systems or system devices secured (Ashfaq, Wang, Huang, Abbas & He, 2017; Lo, Huang & Ku, 2010). These systems are configured to automatically detect and respond to security threats, thereby reducing the risk to network devices or independent devices (Mitchell & Chen, 2014).

Figure 2.7 depicts how an IDPS captures and processes information. The data that is captured on the mobile device is extracted, then sent to an information management system for analysis and the creation of a user profile. If an alert is triggered, an

© Central University of Technology, Free State

appropriate action is taken by the response management system (Ashfaq *et al.*, 2017; Patel, Taghavi, Bakhtiyari & JúNior, 2013;).

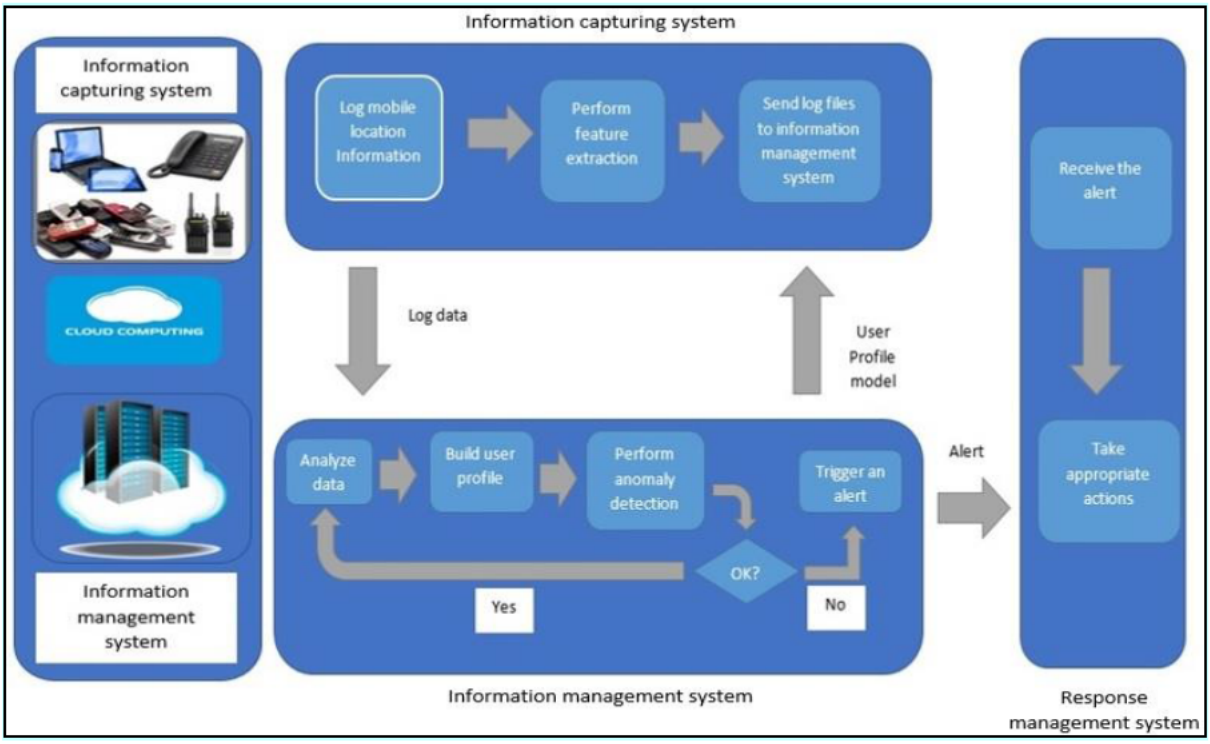


Figure 2-7: IDPS architecture (Lin *et al.*, 2015).

IDPSs can monitor networks in a variety of ways to recognise malicious packets. These systems are classified into three categories, namely: architecture, detection methods, and response approach, as illustrated in Figure 2.8.

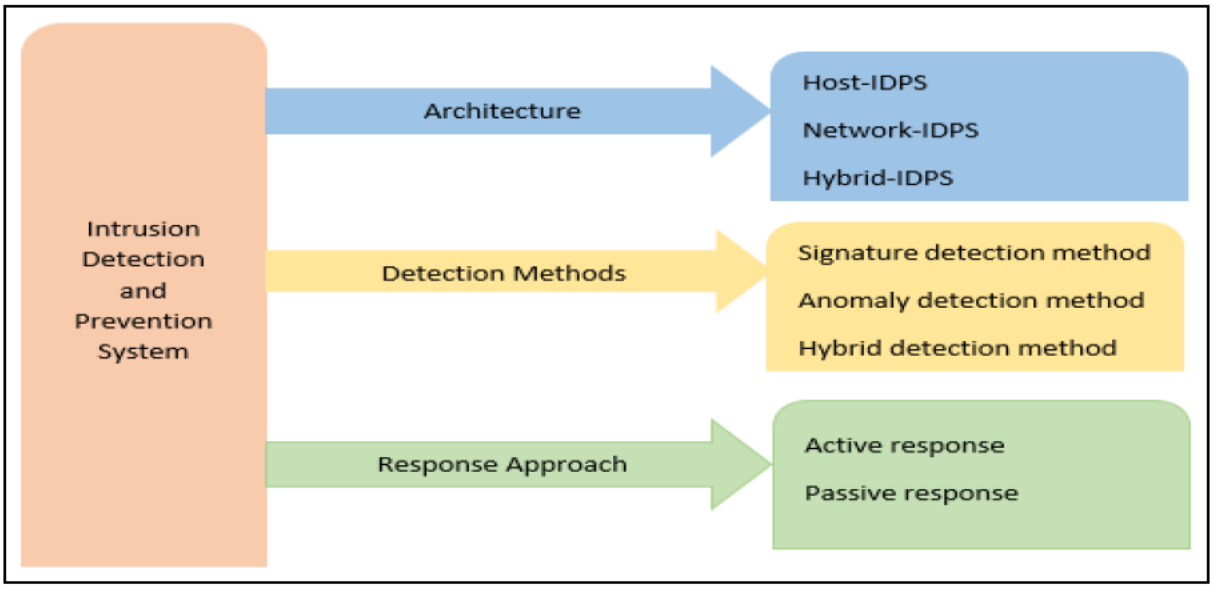


Figure 2-8: Categorisation of IDPS *(Ashfaq et al., 2017)*

© Central University of Technology, Free State

2.4.1 Intrusion detection and prevention system architecture

IDPSs can be categorised into three types of architectures: host-based IDPSs (HIDPSs), network-based IDPSs (NIDPSs), and hybrid IDPSs (Hirsh, 2015; Wang, Hao & Huang, 2010).

Host-based IDPS: A host-based IDPS monitors a device to detect intrusions and/or exploitations and responds by recording the activity and alerting the authorised personnel (Alzahrani, Stakhanova, Ali & Ghorbani, 2014). Furthermore, an HIDPS can be seen as an agent that is used to monitor and evaluate whether there is an internal or external intrusion threatening the device's security and responds by taking an action (Rassam, Maarof & Zainal, 2012; Singh & Thapar, 2012).

Network-based IDPS: This is a type of IDPS that uses a network adapter to monitor and analyse traffic in real-time, meaning it monitors incoming and outgoing traffic as it is transmitted across the network (Gul & Hussain, 2011). This type of IDPS notifies, alerts, and takes action once the attack has been detected. The responses are based on how harmful the attack is to the system, but typically include administrator notification, connection termination or session recording for deep analysis and evidence that the attack took place (Gul & Hussain, 2011).

Hybrid IDPS: This kind of IDPS is a combination of host-based and network-based IDPSs (Lo *et al.*, 2010; Mehmood, Habiba, Shibli & Masood, 2013).

This study has adopted a host-based IDPS which will run on the mobile phone. This will provide comprehensive and complex attack detection and prevention protection for m-voting.

2.4.2 Intrusion detection and prevention methods

Based on the tactic used to discover network intrusions, intrusion detection and prevention methods can also be categorised into three key types: signature-based, anomaly-based, and hybrid systems (Abduvaliyev, Lee & Lee, 2010; Hirsh, 2015; Khune & Thangakumar, 2012). Figure 2.9 illustrates the methods used to detect malware:

© Central University of Technology, Free State

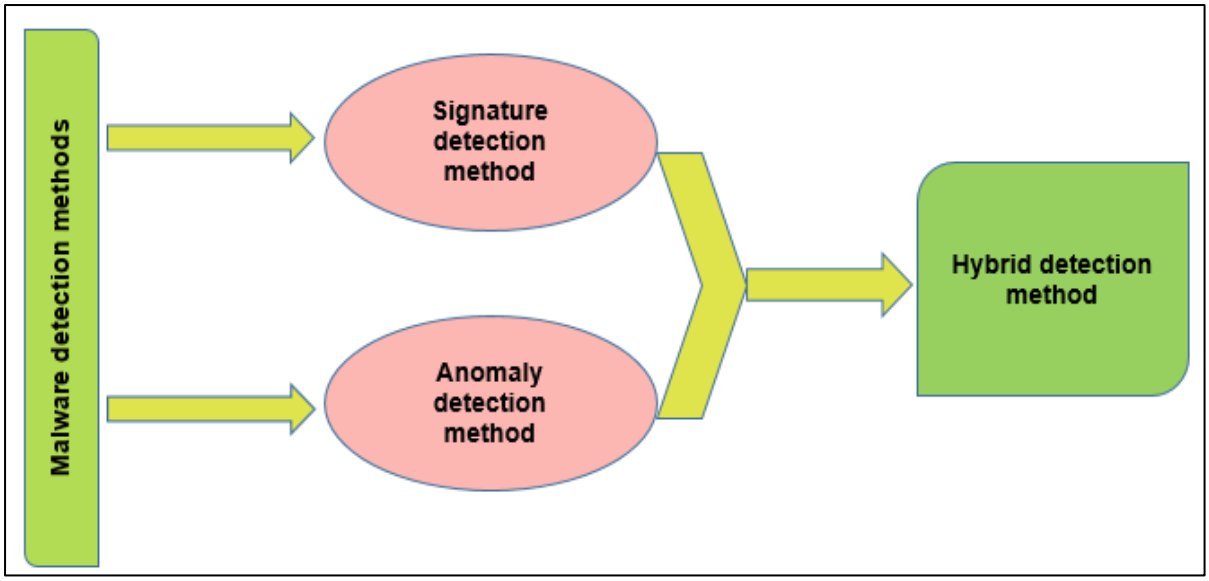


Figure 2-9: Malware detection methods

Signature detection method: It is also known as the misuse detection method. Most of the commercial antivirus companies use signature detection to identify malware. A signature detection method analyses the content of the file dictionary of malware where evidence of the recorded intrusions will be found (Mitchell & Chen, 2014). Every attack leaves a trail, like the nature of data packets, an unsuccessful attempt to run an application, or failed logins, file and folder access. These trails/footprints are called signatures and are used to identify and prevent the same attacks in future (Ahmed, Mahmood & Hu, 2016; Moon, Pan & Kim, 2016).

This method needs a huge database in which to store the detected malware signatures (Hubballi & Suryanarayanan, 2014; Kim, Lee & Kim, 2014). A signature database is therefore created based on the attacks detected. The IDPS does not need to keep on recording the signatures of malware that have been detected previously.

Many organisations are currently employing signature-based IDPSs as part of their internet security, because these systems normally provide accurate and clear detection results (Mitchell & Chen, 2014). The drawback of using this type of IDPS is that only known trails stored or recorded in the signature database can be used when analysing network traffic. If a new type of attack is launched, its signature will be unrecognisable to a signature database. Nevertheless, an advantage of a signature-based system is that as soon as the new attack is known and understood, its signature

© Central University of Technology, Free State

can be stored in the signature database and always be recognised in future (Hubballi & Suryanarayanan, 2014; Kim *et al*., 2014).

Anomaly-based detection: This intrusion detection and prevention method compares classifications of actions that are considered normal or abnormal against the incoming and outgoing traffic to identify whether or not a certain event is harmful (Ahmed *et al.*, 2016). The characteristics of the event are monitored over a certain period and a profile is developed. The IDPS then matches the characteristics of the current event to thresholds related to the profile that was developed.

This method can be very effective and efficient in detecting formerly unknown attacks. Common problems with this method are, among others, the unintended inclusion of malicious activity within a profile, causing high false-positive alerts (Hirsh, 2015).

As much as there are advantages using the anomaly-based method, there are two challenges that make this method less desirable to be used in the industry. Firstly, this method is only effective when the right thresholds have been set as a baseline for system normalcy which is challenging to maintain in most cases (Hirsh, 2015). The second drawback is that this method is computational and resource wise very expensive due to its nature of tracking network streams and storing state information (Hubballi & Suryanarayanan, 2014).

Hybrid detection method: According to Catania and Garino (2012), this method is an amalgamation of both the signature- and anomaly-based detection methods. It was developed as a solution to the limitations of the other two methods. A hybrid IDPS uses the advantageous features of both the signature- and anomaly-based methods to detect attacks without the limitations of each. (Hirsh, 2015).

2.4.3 Intrusion detection system response approach

IDSs can be classified into two different response approaches which are **active** and **passive**:

An active IDS is also known as an intrusion detection and prevention system (IDPS) (Mitchell & Chen, 2014; Rassam *et al.*, 2012). This type of IDPS is configured in such a way that it consistently blocks suspected intrusions without any intervention required by the network administrator (Mitchell & Chen, 2014).

© Central University of Technology, Free State

A passive IDS, on the other hand, is only capable of triggering an alarm when an intrusion is detected; it does not perform any defensive or corrective tasks on its own (Li *et al.*, 2009; Mitchell & Chen, 2014).

Table 2.2 provides a comparison of the two response approaches:

Table 2-2: IDS response approaches

Active Response	Passive Response
• Triggers an alarm • Takes action against attack • Blocks intrusion	• Triggers an alarm • Collects other information to be sure • Does not have blocking capabilities

Due to the limited resources, for example, memory and central processing unit (CPU), of mobile phones, it is a challenge to run an active IDPS directly on a mobile phone (Mitchell & Chen, 2014; Rassam *et al.*, 2012). Therefore, the security solution proposed in this study uses cloud computing services which offers unlimited storage capacity.

The next section discusses cloud computing technology.

2.5 CLOUD COMPUTING

Cloud computing is the process of using a system of remote servers on the internet to store, manage, and process data instead of using a local server or any other data storage device (Hashem, Yaqoob, Anuar, Mokhtar & Gani, 2015; Zhang, Jiang, Li, Liu, Vasilakos & Liu, 2016). This facility provides various benefits, as can be seen from Figure 2.10 below:

© Central University of Technology, Free State

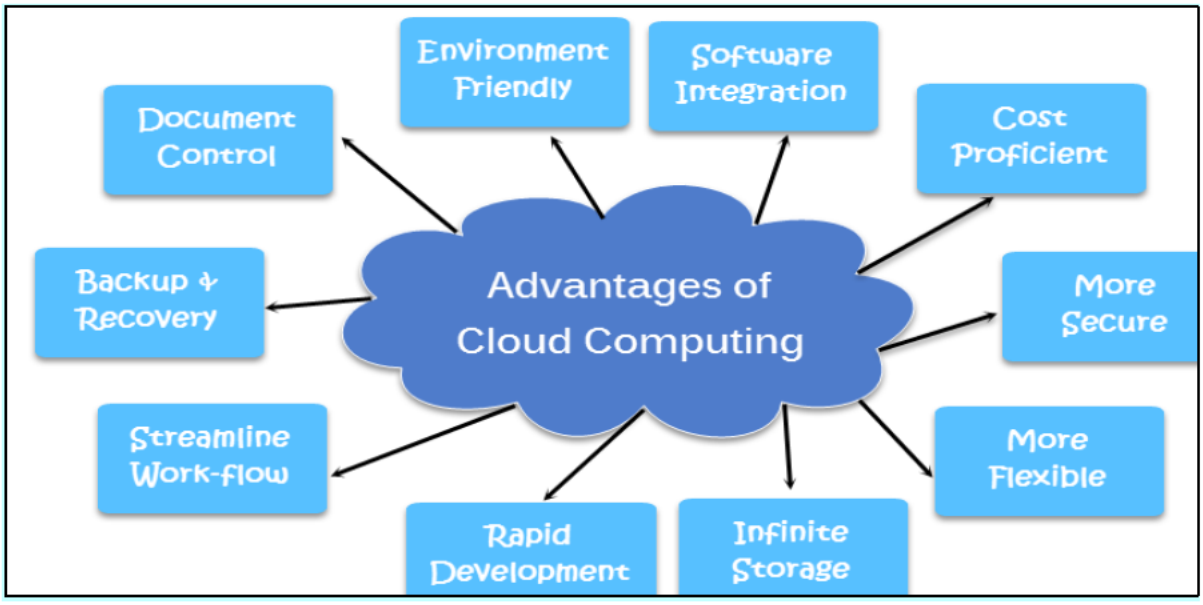


Figure 2-10: Cloud computing benefits (Hashem et al., 2015; Zhang et al., 2016)

2.5.1 Characteristics of cloud computing

According to Mitchell and Chen (2014), the main characteristics of cloud computing are:

1. On-demand self-service that allows users to use computing services (e.g., applications and storage) as they need them (Papamartzivanos, Damopoulos & Kambourakis, 2014; Zonouz *et al.*, 2013).

2. Resource pooling that enables merging computing services (e.g., hardware, software, processing, network bandwidth) to aid numerous customers with their needs (Bhat, Patra & Jena, 2013; Gupta, Kumar & Abraham, 2013; Mehmood *et al.*, 2013; Patel *et al.*, 2013).

3. Fast elasticity and scalability that allow any system to adapt or scale well when there is growth in demand (Patel *et al.*, 2013; Zissis & Lekkas, 2011).

4. Measured provision to improve resource distribution and to allow a metering ability to regulate the usage for billing purposes, give extension to existing hardware, and application resources. As a result, the rate of extra resource provisioning is reduced (Houmansadr *et al.*, 2011; Mehmood *et al.*, 2013; Raja, 2013).

© Central University of Technology, Free State

2.5.2 Cloud computing deployment models

Cloud computing allows users to have options regarding operating their infrastructures whilst saving costs and delegate different responsibilities to third-party providers (Gupta *et al.*, 2013). It has become an essential part of technology and business models, and has forced businesses to adapt to innovative technology tactics (Kolpyakwar & Bhute, 2015).

Therefore, the demand for cloud computing has fuelled the development of a unique market offering that represents different cloud service and delivery models. These models considerably improve the range of existing choices and task organisations with problems such as which cloud-computing model to use (Carroll, Van der Merwe & Kotze, 2011).

There are three commonly used cloud deployment models, namely private, public, and hybrid deployment models. A community cloud is regarded as an additional deployment model, though it is not often used (Modi, Patel, Borisaniya, Patel, Patel & Rajarajan, 2013; Modi & Acha, 2017).

Private clouds are not open to public users, because their infrastructure is controlled by private organisations (Carroll *et al*., 2011). This deployment model is normally used by large organisations that wish to have the benefit of scalability, availability, and structural transparency with a strict administration of data security (Chang, Kuo & Ramachandran, 2016).

A **community cloud** is a shared effort in which infrastructure is pooled between numerous organisations from a particular community with common concerns, such as security, compliance, and jurisdiction, that are hosted internally or externally (Chang *et al*., 2016; Gul & Hussain, 2011).

Most end users prefer to use **public clouds** because of their rapid set-up time and low capital cost (Chang *et al*., 2016). The providers of this type of cloud usually divide their physical servers and rent them to the cloud users. Therefore, the end users have unlimited computational power and storage capacity (Hashem *et al*., 2015; Houmansadr *et al*., 2011).

© Central University of Technology, Free State

The **hybrid cloud** model uses the features of both public and private clouds (Khune & Thangakumar, 2012). Organisations use this cloud whenever they have software or hardware compatibility issues with outside cloud providers but still want to benefit from the huge storage space and other resources provided by public clouds. Another reason to opt for hybrid clouds is the extensibility in revealing corporation assets for a limited time to public users. Thus, a corporation's resources can be partly revealed on the public side of the cloud, rather than endangering everything on the public cloud (Patel *et al.*, 2013).

Figure 2.11 summarises the four cloud deployment models:

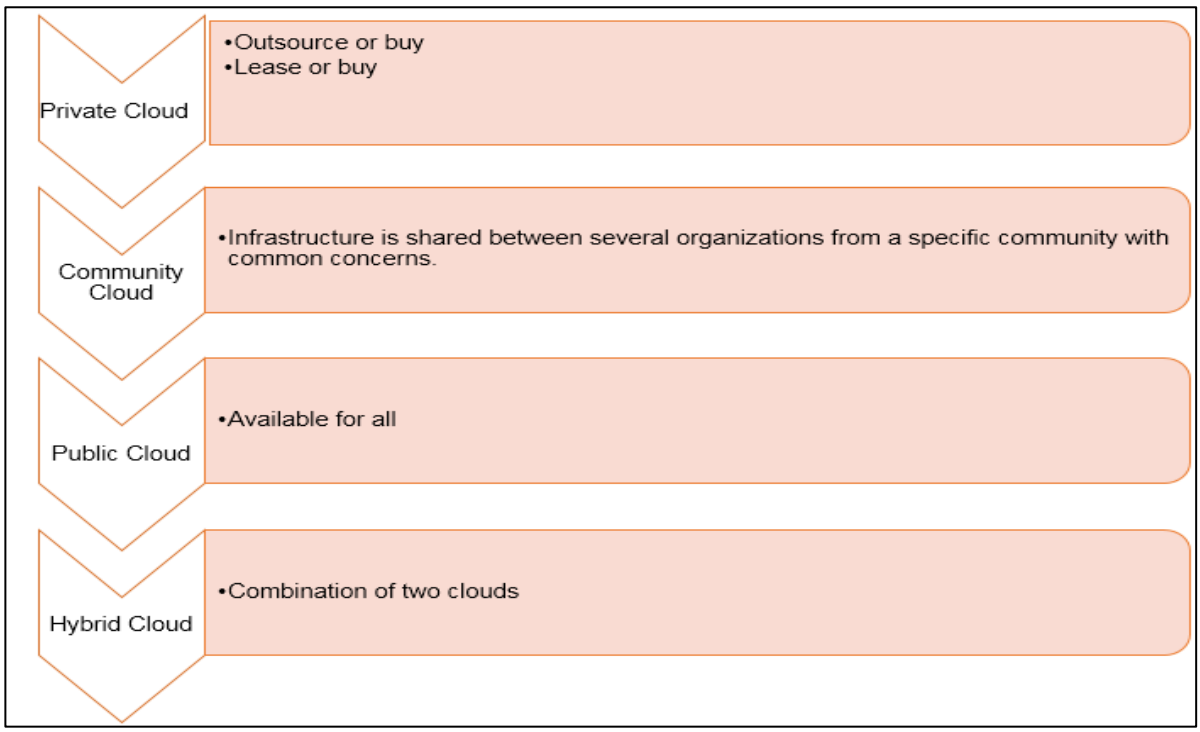


Figure 2-11: Cloud deployment models (Chang *et al*., 2016)

Cloud computing consists of two different service mechanisms for end users, namely software and hardware over the internet (Ravale, Marathe & Padiya, 2015). In addition, there are various cloud service delivery models which are categorised into three layers depending on the services provided by the cloud (Khune & Thangakumar, 2012).

As seen in Figure 2.12 below, the layer at the bottom provides elementary infrastructure mechanisms such as CPU, memory, and storage and is henceforth

© Central University of Technology, Free State

referred to as *Infrastructure as a Service* (IaaS) (Dinh, Lee, Niyato & Wang, 2013; Wang *et al.*, 2015).

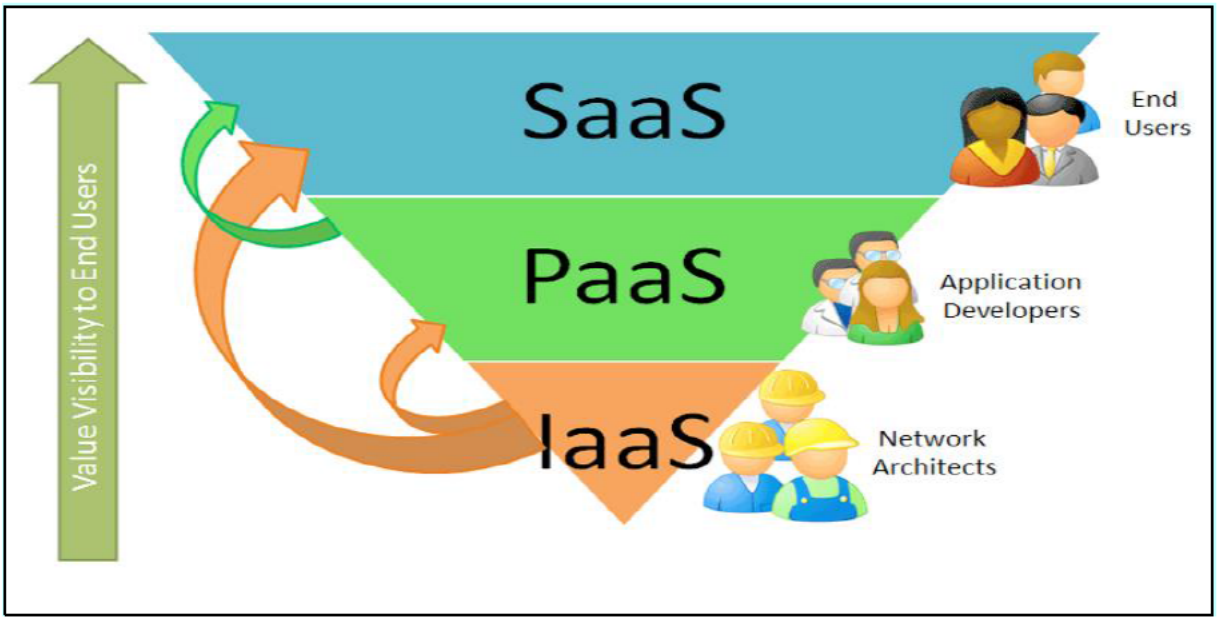


Figure 2-12: Cloud computing layers (Khamphakdee, Benjamas & Saiyod, 2015)

Just above the IaaS, is the middle layer called *Platform as a Service* (PaaS). Platform-oriented facilities allow the usage of a hosting environment intended for a specific need. Google App Engine, Force.com, Microsoft Azure, and Apache Stratos are examples of PaaS (Hashem *et al.*, 2015; Wang *et al.*, 2015). Finally, the top layer provides users with applications that are available over the internet at any given time; this layer is also known as *Software as a Service* (SaaS) (Chang & Ramachandran, 2016; Chang *et al.*, 2016).

Each one of these layers provides a unique service to users in the cloud computing environment (Chang & Ramachandran, 2016; Chang *et al.*, 2016; Zhang *et al.*, 2016). For the purposes of the security solution proposed in this study, PaaS is the chosen cloud computing service to be used for the storage of the IDPS, data used for m-voting, and other related information.

The following section gives a summary of this chapter.

© Central University of Technology, Free State

2.6 SUMMARY

The main objective of this study is to compare and evaluate two cloud-based intrusion detection and prevention systems in order to propose a CIDPS for m-voting in SA. The study focuses on the security of m-voting and argues that in order to effectively secure m-voting, multiple technologies must work together. As a result, the study combines intrusion detection and intrusion prevention systems, as well as cloud computing, to effectively and efficiently secure an m-voting application.

This chapter discussed these elements: democracy, ICT, e-voting, m-voting, mobile phone security, IDPSs, and cloud computing.

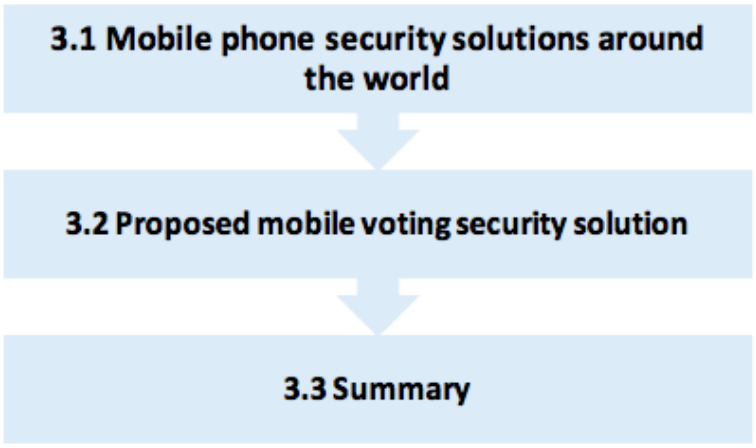
The literature that was reviewed indicated that more and more users rely on mobile phones to meet their everyday needs. Mobile phones have swiftly accumulated popularity by delivering more innovative computing and connectivity functionalities compared to personal computers, which have restricted functionality.

This chapter introduced the concept of m-voting and discussed its advantages and disadvantages. The ability of mobile phones to provide convenience and flexibility opens doors to intrusions and security threats. Previously, other researchers have attempted to secure mobile phones by developing security solutions that operated directly on the device itself, but due to the limited computational resources of mobiles phones, these systems have failed to provide effective security. The features and various types of IDPSs and cloud computing technologies were therefore briefly discussed.

The next chapter covers related work done by other researchers who developed similar security solutions (cloud-based IDPS and lightweight IDPS). The chapter also briefly discusses the operation of the proposed security system for m-voting.

© Central University of Technology, Free State

CHAPTER 3: MOBILE VOTING SECURITY SOLUTIONS



© Central University of Technology, Free State

This chapter discusses work done by other researchers who developed similar security solutions such as IDPSs for mobile devices – cloud-based or lightweight. The chapter also briefly discusses the operation of the proposed security system for m-voting.

Section 3.1 presents mobile phone security solutions from around the world and Section 3.2 briefly discusses the security solution proposed in this study. The chapter is concluded with a brief summary in Section 3.3.

3.1 MOBILE PHONE SECURITY SOLUTIONS FROM AROUND THE WORLD

A number of approaches to mobile phone security solutions exists. The literature reviewed for this study indicates that many of these approaches support running a light-weight intrusion detection software directly on the mobile phone (Raja, 2013). Some approaches have developed a full IDS running on the mobile phone (Adigun *et al*., 2014), and other IDPS solutions employ the cloud environment (Houmansadr *et al*., 2011; Zonouz *et al*., 2013). However, up to now researchers have unsuccessfully tried to develop an effective security solution for mobile phones; their systems fail to cater to both malware detection and prevention while saving computational resources.

Portokalidis *et al*. (2010) developed a prototype security model for Android mobile phones. According to this model, a tracer records all the necessary information on the mobile phone to precisely echo its operation. The recorded execution trace is conveyed to the cloud through an encoded channel, where a copy of the phone is running on a simulator. In the cloud, a re-player accepts the trace and re-runs it within the simulator.

The prototype uses a network proxy for internet connection which permits the developers to intercept and temporarily store incoming traffic. The re-player accesses the proxy to obtain the data needed for replaying. This way, the tracer does not have to re-transmit the data received over the network to the replica.

The evaluation results of this prototype showed that it is both practical and scalable: no more than two *kibibytes* per second and 64 bytes per second of trace data for high-loads and idle operation respectively were generated, and the prototype security

© Central University of Technology, Free State

model is capable to maintain more than a hundred replicas operating on a single server.

Kim, Smith and Shin (2012) presented a power-aware, malware-detection IDS that monitors, detects, and examines new energy-depletion threats. The power-aware malware-detection system comprises two agents, namely a power monitor and a data-analyser which exist in combination. The power monitor runs directly on a mobile phone, taking samples of the power consumption which are then used to build a power consumption history. The data-analyser processes the power consumption history on the host IDS.

During the evaluation of this system, it rendered 27% detection time. This resulted in high false-positive alerts due to slow processing, since everything is running on the device itself. The developers realised that this IDS requires more computational resources for it to functions effectively.

Zonouz *et al*. (2013) identified the main challenge when building effective mobile phone security solutions: the computational resources. As a result, these researchers developed a lightweight IDS that operates on a mobile phone.

This IDS requires the mobile phone to be registered in the simulated environment of the cloud in real time. The system then captures and logs all information to fully replicate the registered mobile phone. The IDS records all input to the device, such as incoming traffic and physical sensors, and replays it to the simulator with no need for periodic checks between the device and the simulator in the cloud.

The evaluation of this system revealed that their approach is resource-intensive, which results in high volumes of overhead traffic being generated by the device.

Likewise, the study conducted by Houmansadr *et al*. (2011) addressed the critical challenge of keeping mobile phones secure by developing a cloud-based intrusion detection system. These researchers applied a working prototype of the intrusion forensics analysis engine for the Linux kernel on mobile phones. The forensics engine comprises two sources of information: 1) a set of IDSs that monitors several parts of the system; and 2) system calls, which are logged by a loadable kernel module that was established by manipulating the system call table and replacing each system call function with a wrapper logging function.

© Central University of Technology, Free State

Any incoming traffic is analysed to identify any abnormal behaviour. Their results indicated that, despite the computational and storage resource limitations of mobile phones, the forensics engine performed a comprehensive and in-depth analysis of the mobile phone. All the investigations were carried out on a simulated device in a cloud environment.

Also, Buennemeyer, Nelson, Clagett, Dunning, Marchany and Tront (2008) developed a Battery-Sensing Intrusion Prevention System (B-SIPS) for mobile phones in Microsoft C# in the .NET Compact embedded environment. This system is designed to detect and send alerts when abnormal behaviour is detected.

B-SIPS provides a threshold-monitoring and alert notification in a host application, which triggers when there is a change in power usage of the mobile device. The system consists of a host application that works as a sensor in a wireless network, and also the server that runs the analysis. The B-SIPS focuses on certain power depletion attacks, such as floods, buffer overflows, and several DoS attacks which are profiled by their patterns.

The researchers argued that some attacks create temporary spikes in power usage and are more difficult to pattern. B-SIPS monitors the power consumption of a mobile phone with Bluetooth and Wi-Fi communication activity. When abnormal behaviour and an attack activity are detected, it is reported to the server for correlation with Snort alerts.

The study results indicated that their system managed to detect abnormal behaviours; however, the detection rate was lower than what they anticipated which created a lot of false-negative alerts.

Another study conducted by Kolpyakwar and Bhute (2015) developed a cloud-based IDS for mobiles phones. Their system comprised two parts: 1) the mobile host agent, and 2) the proxy server. The mobile host agent is a lightweight process that operates on the device and inspects every activity taking place. The proxy server acts as an intermediary which mirrors the incoming and ongoing traffic between the mobile device and the cloud. Also, it is the job of the proxy server to send all traffic to the cloud services where further analysis is performed on the basis of behaviour patterns and code signatures.

© Central University of Technology, Free State

The researchers' system turned out to be feasible and effective since it provided a 79% rate of true-positive alerts and a 21% rate of false-negative alerts. The system provides more comprehensive security for mobile phones.

It is evident that there is no single technique that can guarantee threat detection whilst at the same time prevent those attacks from taking place. It is necessary to keep on searching for an effective security solution to protect the confidentiality and integrity of m-voting in SA while saving mobile phone resources.

3.2 PROPOSED M-VOTING SECURITY SOLUTION

The main aim of this study is to compare and evaluate two IDSs in order to propose a suitable CIDPS for m-voting in SA.

The proposed security solution comprises two parts: the first part is the client software or client agent running on a mobile phone; the second part is the cloud IDPS or cloud analysis engine that uses PaaS as a cloud computing layer. The client agent monitors and collects user-sensor inputs and outputs from the device interface in runtime and sends it to the cloud analysis engine to perform an intensive malware scan.

The cloud analysis engine uses a hybrid detection method where the signature and anomaly detection methods work in parallel. The cloud analysis engine makes use of a malware library to scan for intrusions. The client agent listens for notifications from the cloud analysis engine if a threat is detected.

For the purposes of this study, an m-voting application that has already been developed was used. The criteria that the researcher used in choosing a suitable m-voting application was:

1. The researcher must have access to code.

2. The application must have been developed with the South African context in mind.

Table 3.1 below lists some of the applications that are available online and summarises the criteria that were used to choose a m-voting application.

Table 3-1: List of m-voting applications and the criteria

© Central University of Technology, Free State

M-voting applications	Access to code?	Developed with South African context in mind?	References
SMS Voting	No	No	Warrier (2010)
eVOTZ	No	No	Klein (2010)
XaP	Yes	Yes	Mpekoa (2014)
GSM mobile voting	No	Yes	Feng, Ng and Schwiderski-Grosche (2006)
m-voting system	No	No	Ekong and Ekong (2010)
GSM vote system	No	Yes	Gentles and Sankaranarayanan (2012)

The m-voting application that met the criteria was XaP. It was therefore used in this study as the m-voting application to be secured.

More details on the proposed CIDPS are presented in Chapter 5, Section 5.1.

3.3 SUMMARY

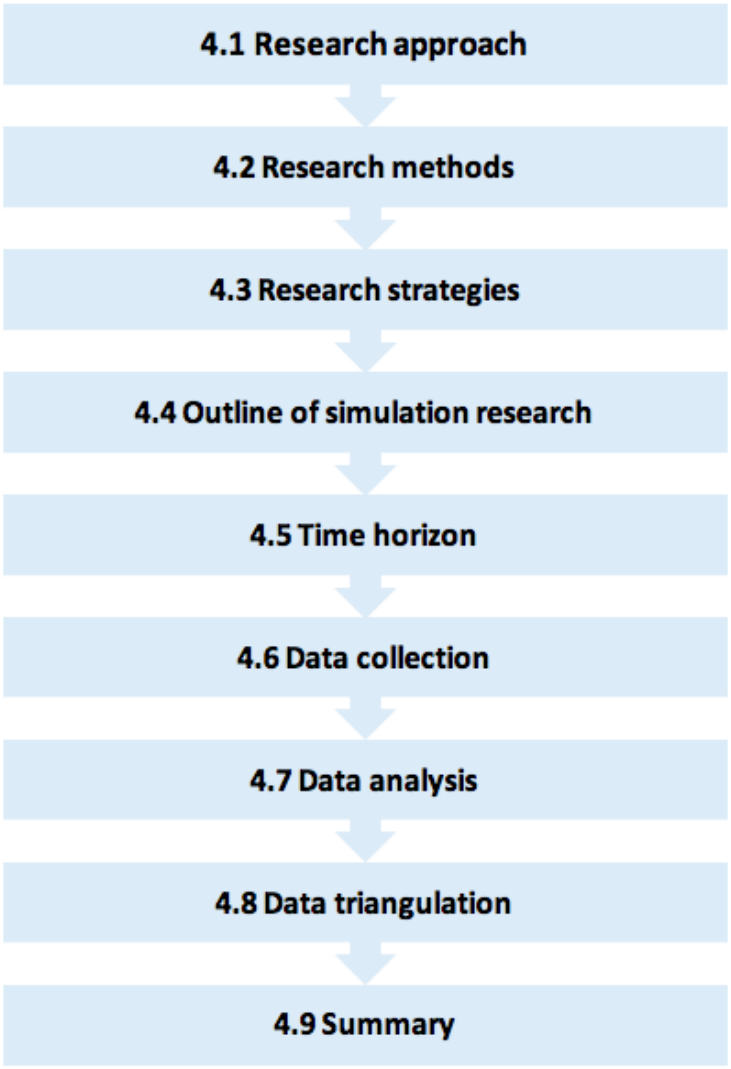
In this chapter, the related work that was done in this field by other researchers was presented. The literature revealed that the security solutions of other researchers are limited regarding the computational resources of mobile phones which results in the failure of their systems.

Hence, this study uses cloud computing services to house the proposed security solution. The last section of the chapter briefly described the proposed security system for this study.

The next chapter presents the methodology that was used during this study.

© Central University of Technology, Free State

CHAPTER 4: RESEARCH DESIGN AND METHODOLOGY



© Central University of Technology, Free State

This chapter presents the chosen research design and research methodologies used in this study. Section 4.1 presents the research approach adopted for the study. In Section 4.2, the research methods that were used are discussed. Section 4.3 covers the research strategies used in the study and Section 4.4 presents the outline of simulation research. The time-horizon that was used is discussed in Section 4.5, while Section 4.6 presents the data collection methods employed. Section 4.7 covers data analysis methods and Section 4.8 presents data triangulation. Lastly, the chapter is summarised in Section 4.9.

4.1 RESEARCH APPROACH

A research approach is the way a researcher deals with the studied phenomenon in order to analyse and examine it; it can be deductive, inductive or abductive (Thomas, Silverman & Nelson, 2005). The common denominator between deductive, inductive, and abductive approaches is the significance of assumptions to the study (Boell & Cecez-Kecmanovic, 2014).

The deductive approach tests the legitimacy of the assumptions in hand, also known as theories or hypotheses, whereas the inductive approach supports the development of new theories in a broader view (Creswell, 2013). Abductive research, on the other hand, begins with facts and the whole research journey is dedicated to finding their explanations. Table 4.1 below demonstrates the major differences between the three approaches:

Table 4-1: Research approaches (Creswell & Poth, 2017)

Deductive	Inductive	Abductive
The conclusion is fully dependent on the premises obtained.	Known evidence is used to generate experimental conclusions.	Evidence is used to generate testable conclusions.
Data collection is used to evaluate a proposition or hypotheses related to an existing theory.	The collected data are used to investigate a phenomenon and to classify themes and patterns in order to construct a conceptual framework.	The collected data are used to explore a phenomenon, to find themes and patterns, and to generate a testable conceptual

© Central University of Technology, Free State

Deductive	Inductive	Abductive
		framework through successive data collection and so forth.

Figure 4.1 below further illustrates the differences between these research approaches:

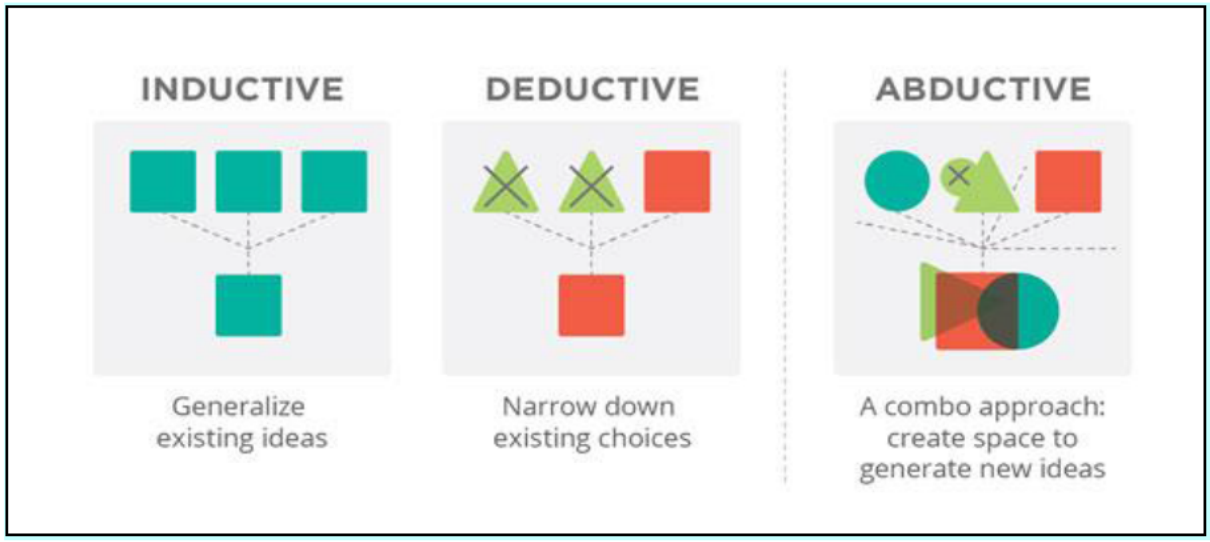


Figure 4-1: Differences in research approaches (Reichertz, 2007)

For this study, an **inductive** research approach was adopted.

Based on the definition offered by Creswell (2013), this study endeavoured to develop a new theory by using known evidence from literature. It attempted to test and propose a CIDPS for m-voting in SA. Hence, the inductive approach was considered most appropriate.

The inductive approach was chosen because the study explored, compared, and evaluated two analysis engines for the purposes of securing m-voting. A theory was generated from the obtained results and the conclusion was drawn that the analysis system named Suricata is the best CIDPS to effectively secure m-voting in SA. Additionally, some researchers in this area have used this approach; amongst others is Ravale *et al*. (2015).

© Central University of Technology, Free State

4.2 RESEARCH METHODS

Research methods are ways in which research problems are systematically solved (Cameron, 2009; Creswell, 2013). There are currently three major research methods, namely:

1. **Quantitative research:** Is used to quantify the phenomenon under study by generating numerical data or data that can be converted into meaningful statistics. It is also used to quantify attitude, behaviour and so forth. The data collection methods used with this research method are much more structured than those of qualitative research (Creswell, 2013; Håkansson, 2013).

2. **Qualitative research:** It is used to obtain a better understanding of important reasons, opinions, and motivations. It offers insights into the problem at hand and aids in developing ideas for potential quantitative research (Creswell & Klassen, 2011; Gioia, Corley & Hamilton, 2013). Qualitative data collection methods are carried out in various ways, such as using unstructured or semi-structured practices. Some common methods include focus groups, individual interviews, and observations (Creswell & Clark, 2011; Thomas *et al*., 2005).

3. **Mixed research:** This is a research drive that moves beyond the paradigm wars by offering logical and practical alternatives. With this research method, both quantitative and qualitative research techniques, methods, approaches, concepts, and languages are combined into a single study. Moreover, mixed research is an extensive and inventive form of research; it does not limit the researcher. It is comprehensive, diverse, and complementary (Cameron, 2009; Creswell, 2007; Creswell & Clark, 2011).

An **quantitative method** was adopted for this study.

The quantitative method was used because the researcher used simulation as a strategy to test and compare the effectiveness of the two analysis engines, Snort and Suricata. A simulation process is primarily used to support quantitative data in order to produce quantitative results in a structured process (Creswell, 2013).

© Central University of Technology, Free State

This ties in with Christ (2009) who suggests that traditional quantitative research serves as a methodology of verification rather than discovery. Therefore, in this study the researcher compared and verified the results after the evaluation of the two analysis engines in order to choose a suitable analysis engine to secure m-voting.

4.3 RESEARCH STRATEGIES

According to Gioia *et al*. (2013), a research strategy is a general plan that assists the researcher in answering the research questions efficiently and effectively. Following are a few common research strategies:

1. **Grounded theory:** This research strategy is associated with qualitative methodology, draws on an inductive approach, and includes the documentation and incorporation of categories of meaning from data. It is both the process of category identification and integration (as method), and its product (as theory) (Creswell, 2013; Pickering & Byrne, 2014).

2. **Surveys:** This strategy is normally used in quantitative research projects and involves test groups, which are the representative section of the population (Creswell, 2007). Surveys produce quantitative data that can be analysed empirically and are usually used to observe causal variables between dissimilar data (Creswell, 2013; Pickering & Byrne, 2014).

3. **Ethnography:** This strategy entails close examination of people, observing their cultural interaction and their meaning (Pickering & Byrne, 2014). When using this research strategy, the observer conducts the research from the perspective of the people who are being observed, and tries to understand the differences of meaning or their behaviours from their point of view (Creswell, 2013; Pickering & Byrne, 2014).

4. **An archival research strategy:** With this strategy, research is carried out by using existing materials (Creswell, 2013). Systematic literature reviews may be performed to determine what has been discovered about or is known about a particular study field so far (Pickering & Byrne, 2014).

5. **Action research:** It includes investigating the practice to see if it corresponds to the best approach. This form of research is common in teaching or nursing

© Central University of Technology, Free State

professions, where experts evaluate ways in which they can enhance their professional modus operandi and understanding (Creswell, 2013; Pickering & Byrne, 2014).

6. **Case study research:** This is the assessment of an individual unit in order to establish its main features and draw generalisations (Gioia *et al*., 2013). It can help to improve the understanding of the specific nature of any example and can indicate the importance of culture and context when there are differences between cases (Pickering & Byrne, 2014).

7. **Experimental research:** It examines the results of an experiment against the expected results during a research process (Gioia *et al*., 2013). It can be used in all research disciplines and considers a relatively limited number of factors (Creswell, 2013; Pickering & Byrne, 2014).

In this study, an **experimental** research strategy was adopted.

Experimental research was adopted because the researcher measured and compared the effectiveness of the two analysis engines in order to draw concrete conclusions as to which one is best for securing m-voting.

According to literature, there are three types of experiments, namely:

1. **Laboratory experiments,** which allow the researcher to identify relations between a small number of variables that are studied intensively in a laboratory situation. Quantitative, analytical techniques are used with the goal to make general statements pertinent to life situations (Creswell, 2013; Pickering & Byrne, 2014).

2. **Field experiments** extend laboratory experiments into real-life situations. In this way, realism is achieved and it would be more difficult to judge a situation as being false (Gioia *et al*., 2013). In practice, not many organisations are willing to be quinea pigs and it is even more difficult to control situations sufficiently in order to make replication feasible (Creswell, 2007).

© Central University of Technology, Free State

3. **Simulation** is an abstraction of reality and is used in situations where it is challenging to scientifically solve problems. It normally involves the outline of random variables (Creswell & Clark, 2011). According to Gioia *et al*. (2013), simulation is a technique using computer software to imitate the set-up of real-world processes, systems or events. In research terms, simulations have been employed in process analysis and evaluation operations (Creswell & Clark, 2011; Gioia *et al*., 2013).

Creswell and Poth (2017) suggest that simulations are useful for:

- helping any scholar to understand basic research principles and analytical methods;
- investigating the effects of problems that might arise prior to the execution of research; and
- exploring the accuracy and utility of novel analytical practices that are pragmatic to difficult data structures.

In this study, **simulations** were used as a tool to perform the experiments.

Simulations were chosen as a tool mainly because the researcher did not have all the necessary and required equipment to conduct the study in a real-world environment. Therefore, simulations using computer software were used to model the operation of the real-world environment. It helped the researcher to explore the accuracy of the analysis engines and to test their effectiveness in protecting and securing m-voting.

4.4 OUTLINE OF SIMULATION RESEARCH

This study is a quantitative study that made use of simulations. The following simulation steps proposed by Creswell (2007) as well as Thomas *et al.* (2005), underpinned the study (see Figure 4.2):

© Central University of Technology, Free State

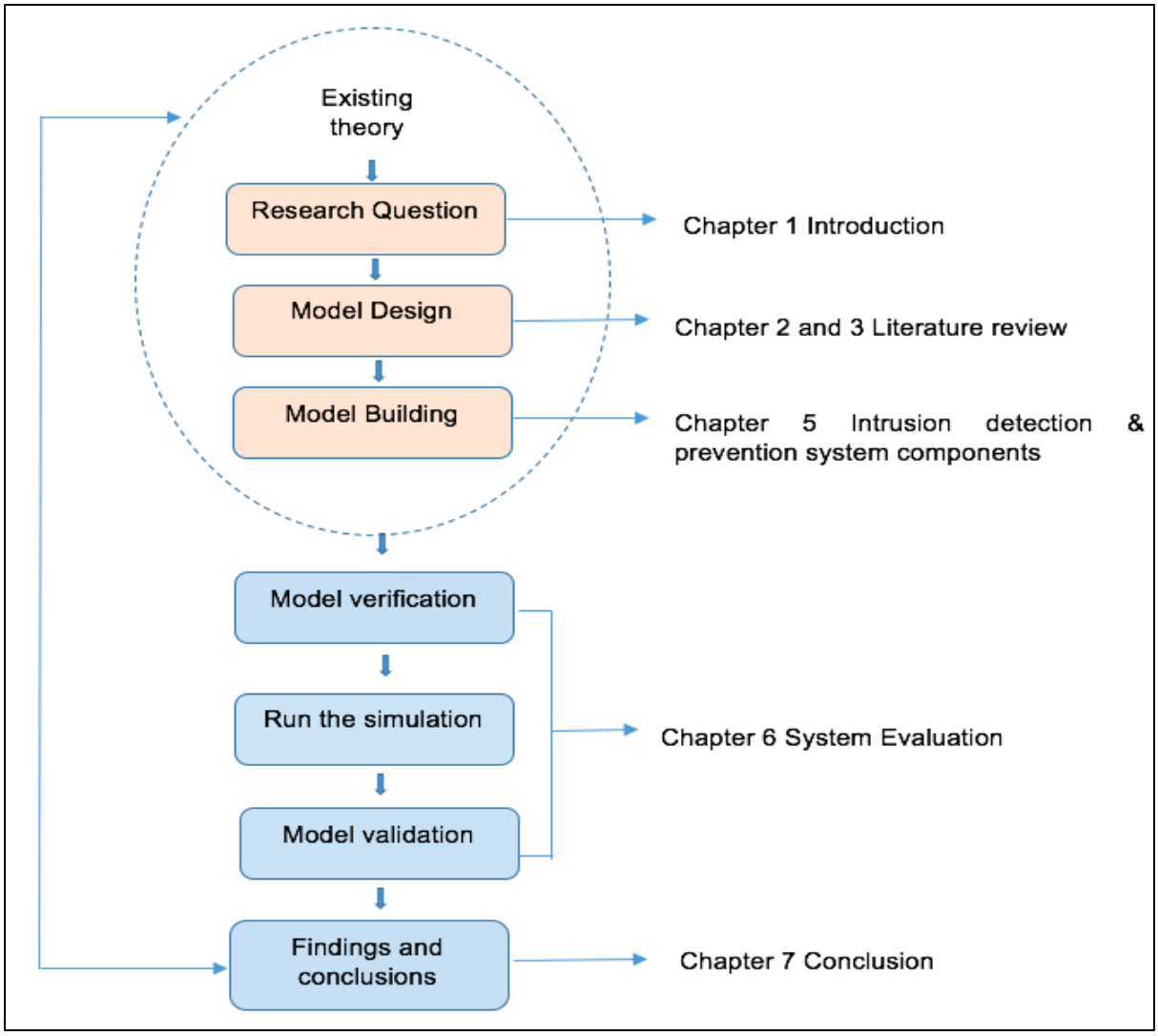


Figure 4-2: Steps in simulation research (Creswell, 2007; Thomas *et al.*, 2005)

1. **Research question:** Formulate a research question that is suitable for study by simulation (Thomas *et al.*, 2005).

2. **Model design:** Includes setting up specifications of the target to be modelled and for a suitable simulation method to be selected. There are a number of different methods to choose from, however, the chosen method will depend on the problem at hand (Creswell, 2007; Thomas *et al.*, 2005).

3. **Model building:** A number of software programs exist that are used to support particular simulation methods, but if there is no suitable software package available, a program can be written to do that (Creswell, 2007; Thomas *et al.*, 2005).

© Central University of Technology, Free State

4. **Model verification:** Verification includes running and testing the simulation to see if the model is working as it should. If any problem occurs during the simulation, it ought to be rectified (Creswell, 2007; Thomas *et al.*, 2005).

5. **Run the simulation:** Researchers regard simulations as virtual experiments where a sequence of experiments are run in different environments and can be varied as required. Gioia *et al*. (2013) pinpoint five key features of these types of experiments: the initial conditions, the time structure, outcome measurement, the number of iterations and any variation in model parameters or initial conditions (Creswell, 2007; Thomas *et al.*, 2005).

6. **Model validation:** Verifies that the simulation is suitable for the target by comparing the outcomes of the simulation with empirical data. Validation can be a difficult process because of the nature of simulation and potential limitations on available empirical data (Creswell, 2007). Nonetheless, as Thomas *et al.* (2005) highlights, it is imperative that the model is adequately reliable for people to be confident acting on the insights it yields.

7. **Findings and conclusions:** The findings and conclusions should answer the research questions and the outcomes must be distributed (Thomas *et al.*, 2005). Cameron (2009) states that providing sufficient data for the study to be replicated whilst avoiding burying the reader in detail, can be a challenging task when reporting simulation research (Creswell, 2007; Thomas *et al.*, 2005).

4.5 TIME-HORIZON

The time-horizon is the probable period of time in which a project will be completed (Wurm, Tomasik & Tesch-Römer, 2010). The research conducted by Creswell and Klassen (2011) revealed that a study can be portrayed in a snapshot lookalike, or can have a diary-like viewpoint.

A snapshot horizon is termed as “cross-sectional”, while the diary-like viewpoint is called “longitudinal”. Furthermore, Wurm *et al*. (2010) suggest that the time perspective to research, whether cross-sectional or longitudinal, does not depend on the research strategy.

© Central University of Technology, Free State

Longitudinal research is chosen when changes or developments that occur over a period are to be studied. Wurm *et al*. (2010) suggest that longitudinal studies are very convenient when studying human behaviour. Longitudinal studies do have a restriction when time is a concern (Creswell & Clark, 2011).

On the other hand, in cross-sectional research the researcher focuses on variables at a given point in time, and data from which to draw conclusions are collected in one go (Creswell & Clark, 2011).

This study made use of a **cross-sectional** time-horizon, because data were collected and analysed under a defined time frame.

4.6 DATA COLLECTION

Data collection is a vital part of any kind of research study. If data collection is done incorrectly, it can affect the results of the study and will eventually lead to invalid results (Christ, 2009).

Depending on the nature of the information to be gathered, different instruments can be used for data collection. These instruments include questionnaires, interviews, surveys, literature reviews, system evaluations, experts' reviews, classroom observation, etc. (Best & Kahn, 2016; Håkansson, 2013).

In order for the researcher to compare and evaluate the analysis engines, this study depended heavily on **reviewed literature**. Once the analysis engines were compared, **system evaluation** was conducted.

Below is a brief explanation of each tool that was used to collect data for this study:

1. **A literature review** is a summary of published research that is relevant to a specific research topic. The literature review of this study informed the researcher about related research that has already been conducted by other researchers.

© Central University of Technology, Free State

According to Creswell (2013), the purpose of a literature review is to create understanding of present ideas on the research topic at hand and may justify and encourage future research on previously unobserved areas.

The main sources used in this study were: books, journal papers, unpublished reports, published reports, research studies, conference abstracts, poster presentations, newspaper articles and other media coverage, internet articles, and any other available source of information that were relevant and documented.

Chapter 2 and Chapter 3 present the literature that the researcher reviewed, as well as related work that was done by other researchers regarding the areas that were identified earlier in this dissertation, namely: e-voting, cloud computing, and IDPSs.

2. **System evaluation** is a continuing process during the performance testing of a system or a project. System evaluation offers better value when piloted early in the testing of a system or a project (Blanco *et al.*, 2011).

   The purpose of system evaluation is to determine how the system is likely to perform in a real-life situation. It is a systematic determination of a subject's merit using criteria governed by a set of standards (Hirsh, 2015; Hubballi & Suryanarayanan, 2014).

   During system evaluation, information is gathered about the entire system, the functions of the system, the projected results and any other details that are helpful in achieving the needs of the system. This information gives a basis for gathering the performance goals and requirements, characterising the capacity, generating performance-testing tactics and plans, and evaluating project and system threats (Hermawan & Sarno, 2012). However, it is important to understand the system that needs to be evaluated in order to successfully test its performance.

   This study evaluated the merits of two IDPSs called Suricata and Snort by using experiments which tested and compared the two analysis engines in a virtual environment. The experiments evaluated system performance by measuring detection time, detection rate, and accuracy.

© Central University of Technology, Free State

The detection time aspect was measured by taking the amount of time that elapsed between intrusion and detection thereof (Abduvaliyev *et al*., 2010), whereas the detection rate was measured by determining the rate at which traffic and audit events were processed (Gupta *et al*., 2013). The accuracy of the two analysis engines was measured by subjecting both to malicious traffic in controlled tests and comparing the alerts created by each application (Abduvaliyev *et al*., 2010; Gupta *et al*., 2013).

According to Abduvaliyev *et al*. (2010) and Gupta *et al*. (2013), there are nine system performance criteria (see Table 4.2):

Table 4-2: System performance criteria (Abduvaliyev *et al*., 2010; Gupta *et al*., 2013)

Criteria	Description
1.Efficiency	The system must use less time and memory for intrusion detection.
2. Effectiveness	The system must be able to detect intrusions and to measure the percentage of false alarms.
3.Portability and reliability	The system must be convenient, and easy to assemble and disassemble.
4. Accuracy	Accurate detection of attacks with the absence of false alarms.
5. Technology maturity	When the technology used is mature and well tested, it can significantly improve the consistency and performance of the system.
6.Expected costs	The system must offer the best return of investment for the user's needs and operational environment.
7.Operation and Maintenance ease	Once the system is installed and operational, it should be easy to maintain.
8. Detection rate	The rate at which traffic and audit events are processed.
9. Detection time	Time elapsed between intrusion and detection.

© Central University of Technology, Free State

Due to time constraints, not all evaluation components could be tested but only a few. Mehmood *et al*. (2013) and Modi and Acha *(*2017) state that for an IDPS to be trusted, it must be evaluated based on accuracy, detection rate, and detection time. These three criteria are the most important in determining the effectiveness and efficiency of any IDPS (Mehmood *et al*., 2013; Modi and Acha *(*2017)).

Hence, these criteria were used to evaluate the two analysis engines and are presented in Table 4.3.

Table 4-3: Evaluation criteria for CIDPS

Criteria	Description	References
Accuracy	Correct detection of attacks with the absence of false alarms.	Abduvaliyev *et al*.(2010); Gupta *et al*. (2013)
Detection rate	The rate at which traffic and audit events are processed.	Abduvaliyev *et al*. (2010); Gupta *et al*. (2013); Volden *et al*. (2011)
Detection time	Time elapsed between intrusion and detection.	Abduvaliyev *et al*. (2010); Gupta *et al*. (2013); Volden *et al*. (2011)

In order to evaluate the IDPSs' performance and detection accuracy, the ratio of four possible events was scrutinised. These events are illustrated in Figure 4.3, followed by their brief definitions:

© Central University of Technology, Free State

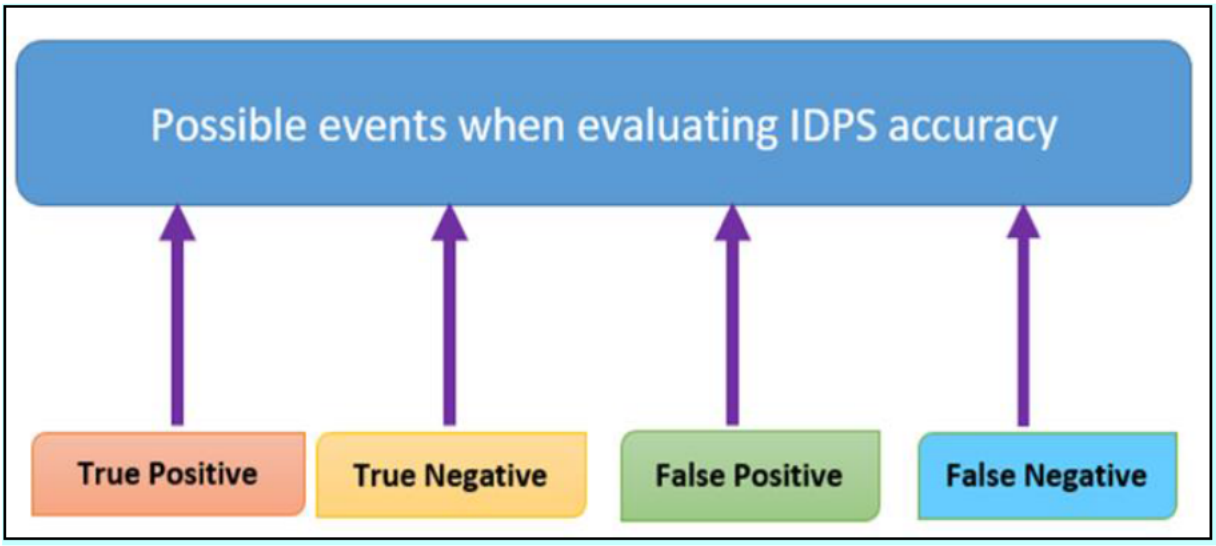


Figure 4-3: Detection events when evaluating IDPS accuracy (Hirsh, 2015)

1. **True-Positive (TP):** This is a situation where an intrusion is present and the system managed to detect it. The signature of the intrusion is correctly identified as an attack against the network (Elhag, Fernández, Bawakid, Alshomrani & Herrera, 2015; Lin *et al.*, 2015).

2. **True-Negative (TN):** This is a situation in which no alarm was triggered, because there was no intrusion present (Elhag *et al.*, 2015; Ravale *et al.*, 2015).

3. **False-Positive (FP):** This is a situation in which an alarm is triggered even when there is no intrusion present (Lin *et al.*, 2015; Ravale *et al.*, 2015).

4. **False-Negative (FN):** This is a situation where a security system fails to detect an intrusion although there is a signature in the signature database designed to detect the intrusion. Thus, an intrusion was present, but the system failed to detect and act (Elhag *et al.*, 2015; Ravale *et al.*, 2015).

Based on the definitions of the events above, the researcher concludes that a system is effective when there is a high number of TP and TN alerts. Likewise, the system is considered unsuccessful if there is a high number of FP and FN alerts. The lower the number of FP and FN alerts, the more effective the system is.

Chapter 5 presents the results of the system evaluation and the interpretation thereof in detail.

© Central University of Technology, Free State

4.7 DATA ANALYSIS

Data analysis is the manner in which questions are answered through examining and interpreting data (Östlund, Kidd, Wengström & Rowa-Dewar, 2011). Data analysis helps the researcher to ask questions, solve problems, and derive important information (Creswell, 2013). Below are different data analysis methods accompanied by their brief definitions:

1. **Descriptive:** describes the main aspects of the data being analysed (Best & Kahn, 2016; Neyeloff, Fuchs & Moreira, 2012).
2. **Exploratory:** here the researcher aims at finding new relations. This type of analysis is great for finding new relations and to provide future recommendations (Best & Kahn, 2016; Cox, 2017).
3. **Inferential:** infers from the sample to the population. This method determines the probability of the character of the population based on the character of the chosen sample (Best & Kahn, 2016; Cox, 2017).
4. **Predictive:** this type of analysis predicts what will happen in the future by observing current and past facts (Best & Kahn, 2016; Cox, 2017).
5. **Casual:** this method is used to determine the effect on one variable when changes are made to some other variable (Best & Kahn, 2016; Cox, 2017).

Descriptive analysis was used in this study.

The descriptive analysis method was chosen because it allows the researcher to quantitatively describe or summarise features of the collected data. It also allows the researcher to present the results graphically by means of graphs or tables representing quantities or frequencies (Best & Kahn, 2016).

© Central University of Technology, Free State

4.8 DATA TRIANGULATION

Triangulation exists to define the grouping of two or more data collection techniques, methodological approaches, investigators or theoretical viewpoints (Creswell & Poth, 2017). Triangulation allows researchers to make a more inclusive, holistic, and related interpretation of a research process (Creswell, 2007). There are five types of triangulations:

4.8.1 Data triangulation

Data triangulation involves using different kinds of information in order to increase the validity of a study. This type of triangulation is the most common and the easiest to implement in research studies due to the fact that it allows the researcher to make use of different sources (Archibald, 2016; Creswell, 2007).

4.8.2 Methodological triangulation

Methodological triangulation uses different approaches, from qualitative to quantitative methods in order to study the problem (Archibald, 2016). This triangulation is done to detect convergence, inconsistency, and contradiction by exploring a research question from different perspectives (Creswell, 2007). It can also refers to data collection methods or research designs (Creswell & Clark, 2011). If the conclusions drawn from each of the methods are identical, then validity is proven. Whilst this method is common, it usually needs time and resources (Best & Kahn, 2016).

4.8.3 Investigator triangulation

Investigator triangulation involves using a number of different “investigators” in the analysis process. Whilst this method is helpful for establishing validity, it may not always be feasible to gather different investigators due to time constraints and individual programmes (Archibald, 2016; Torrance, 2012).

© Central University of Technology, Free State

4.8.4 Theory triangulation

Theory triangulation includes the use of various perceptions to understand a set of data. This method can be inefficient and may not be viable in all situations (Creswell & Klassen, 2011; Torrance, 2012).

4.8.5 Environmental triangulation

This type of triangulation includes the use of different environments, surroundings, and some important factors related to the location in which the study was conducted, such as the time, day or season (Creswell, 2007). It is crucial to identify which environmental factors, if any, may impact the information that is collected during the study. If the results are the same under different environmental settings, then validity has been proven (Creswell & Klassen, 2011).

> In this study, three types of triangulation were adopted, namely **data, investigator, and environmental** triangulation.

This study uses two investigators to run the simulations – one was situated in Bloemfontein and another one in Welkom, Free State. This arrangement offered investigator, environmental, and data triangulation. Benefits of this combination include: more confidence in the validity of the data, a deeper understanding of the issue at hand, and new perspectives on the study topic.

4.9 SUMMARY

As stated earlier, the aim of this study was to compare and evaluate two IDSs in order to propose a suitable CIDPS for m-voting in SA.

This research was produced by following the Design-Science research model as it enabled the researcher to investigate and assess two IDSs with the clear expectation of enhancing the security of an m-voting application. A Design-Science process model was performed. The model helped the researcher to synthesise the selected literature on the topic.

© Central University of Technology, Free State

This chapter described and discussed the research methodology, including the research approach, data collection instruments, and data analysis methods that were used during the study. The study's time-horizon and data triangulation methods were also discussed. Table 4.4 summarises the various research design and methodology options, as well as those that were adopted for this study:

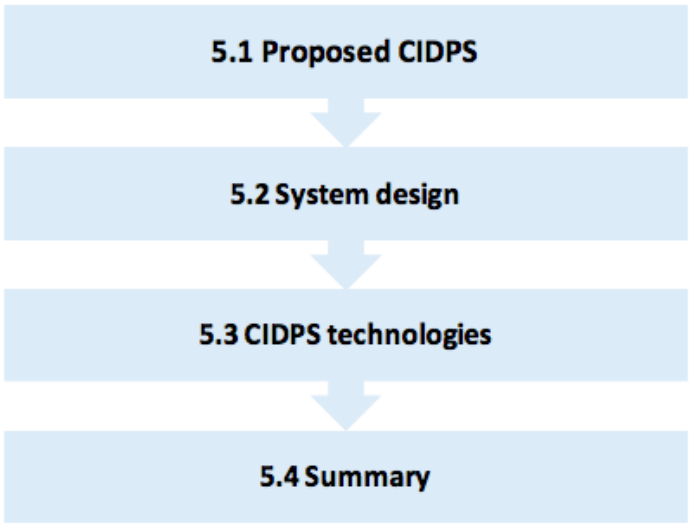
Table 4-4: Research design and methodology

Research Methodologies	Types	Adopted type
Research approach	Inductive Deductive Abductive	Inductive
Research methods	Qualitative Quantitative Mixed methods	Quantitative research method
Research strategy	Survey Ethnography Archival Action research Case study Experimental	Experimental
Time-horizon	Longitudinal Cross-sectional	Cross-sectional
Data collection	Questionnaires Interviews Literature reviews Observations System evaluations Surveys	Literature review System evaluations
Data analysis	Descriptive Exploratory Inferential Predictive Casual	Descriptive

The next chapter, Chapter 5, presents detailed information on the CIDPS that is proposed as a security solution, the technologies and methodologies that were used.

© Central University of Technology, Free State

CHAPTER 5: INTRUSION DETECTION AND PREVENTION SYSTEM COMPONENTS



© Central University of Technology, Free State

This chapter describes the components of the proposed security solution referred to as the CIDPS. In Section 5.1, the proposed security solution is discussed in depth by presenting the system, security, and end user requirements, system architecture, and key role players. Section 5.2 presents the system design by using USE cases and a flowchart. CIDPS technologies are discussed in Section 5.3. The chapter is concluded with a brief summary in Section 5.4.

5.1 PROPOSED CLOUD-BASED INTRUSION DETECTION AND PREVENTION SYSTEM

It is important to state the system, security, and end user requirements for the proposed security solution. The following sections present these three features:

5.1.1 System requirements

The two investigators used Dell Latitude E6540 laptops with the following specifications:

Microsoft Windows 10 Enterprise version 1703

Intel(R) Core(TM) i7-4810MQ CPU @ 2.80GHz

64-bit operating system, x64-based processor

8.00 GB RAM

5.1.2 Security requirements

Security requirements are very important aspects that the proposed system needs to meet. Those requirements that need to be complied with by any IDPS have already been discussed in Chapter 2, Section 2.3.2. The security requirements in Table 2.1 (page 30) include confidentiality, integrity, authenticity, non-repudiation, and availability.

© Central University of Technology, Free State

5.1.3 End user requirements

In the context of this study, end user requirements specify what the end user should have and know in order to use the system successfully. Below are the end user requirements:

1. should own a mobile phone or a smartphone using the Android OS;
2. should be able to read and understand English; and
3. must have knowledge of the registration and voting process.

5.1.4 Cloud-based intrusion detection and prevention system architecture

The figure below (Figure 5.1) illustrates the proposed architecture of the CIDPS for m-voting in SA. The proposed system was constructed from two parts:

- Sensor (client agent); and
- Analyser (cloud analysis engine).

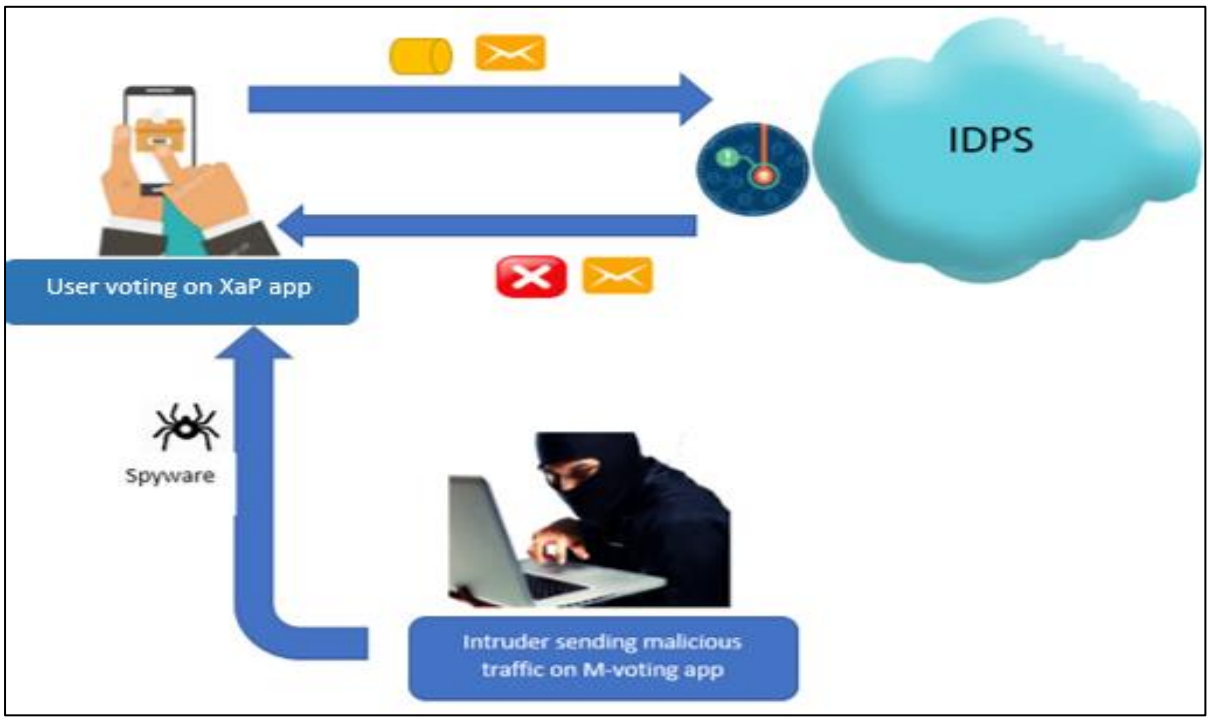


Figure 5-1: CIDPS architecture

Sensors collect data such as network traffic (e.g., incoming and outgoing calls and SMSs), and record files and system trace files. When data is collected, it is forwarded to the cloud analysis engine where an intensive intrusion scan happens. The cloud

© Central University of Technology, Free State

analysis engine is responsible for determining if there are any intrusions among the data sent by the client agent. The cloud analysis engine uses a malware library to scan for intrusions. It makes use of a hybrid detection method where the signature- and anomaly-based methods work in parallel. If an intrusion is detected, the analysis engine's output is either an alarm or an action (block).

Then the client agent listens for notifications from the cloud analysis engine in case a threat is detected. If an intrusion is detected by the cloud analysis engine, its signature is compared with the known signatures stored in the signature database. If there is a match according to the signature detection method, an alert is triggered to warn the client agent. On the other hand, if a match is not found in the signature database and the behaviour of the voter is not normal, the proposed model does consider it as abnormal behaviour according to the anomaly detection method. Also, an alert is triggered to warn the client agent. If it is an intrusion, the signature of the detected intrusion is saved as a new signature within the signature database.

The proposed m-voting security solution was linked with a back-end client agent system developed using Android Version 6.0 for Android phones with a REST API called CloudRail to freely access cloud services. CloudRail was used because it offers a full package service which includes services such as user profile storage, SMS, API's for any programming language, etc. (Luo, Jin, Song & Dong, 2011; Paikaray, Mohapatra & Rath, 2015).

Android is the latest OS used in most mobile phones and dominates the market share (Kolpyakwar & Bhute, 2015; Shabtai & Elovici, 2010). As indicated in Chapter 3, Section 3.2, XaP was chosen as the relevant m-voting application as it allowed the researcher to have access to the XaP code. XaP was developed for Android mobile phones (Mpekoa, 2014), but the security solution can be implemented in any other m-voting application in order to secure the mobile phone whilst voting.

5.1.5 Key role players in the CIDPS

A few key role players are required for the successful implementation of the proposed CIDPS. They are: the mobile voter, telecommunication network (e.g., 3G/LTE), the XaP application, the client agent, the cloud analysis engine, the attacker, and the cloud. The roles of these role players are defined in Table 5.1 below:

© Central University of Technology, Free State

Table 5-1: CIDPS key role players

Key role players	Roles
Telecom network	Allows the mobile voter to be wirelessly connected to the internet so that the IDPS can be active.
Mobile voter	A qualified voter using a mobile device to cast a vote.
XaP application	A mobile voting system which allows users to cast a vote using their mobile phones.
Client agent “mobile sensors”	Monitors and collects user-sensor inputs from the device interface in runtime, and sends it to the cloud analysis engine.
Cloud analysis engine	An IDPS running on the cloud that collects data from the client agent to perform intensive malware scans.
Attacker/Intruder	Any individual who tries to gain access to the mobile phone while the voter is voting or send malicious traffic to disturb the voter.
Cloud storage	Houses the cloud analysis engine, signature method, anomaly method, and user profile.

© Central University of Technology, Free State

5.2 SYSTEM DESIGN

5.2.1 Cloud analysis engine USE case diagram

In system software, a **USE case** is a list of activities or event steps, normally outlining the connections between a role (known in the Unified Modelling Language as an actor) and a system to achieve an objective (Hermawan & Sarno, 2012). The actor can be a human or another external system (Ranjini, Kanthimathi & Yasmine, 2011).

Figure 5.2 depicts the cloud analysis engine USE case diagram:

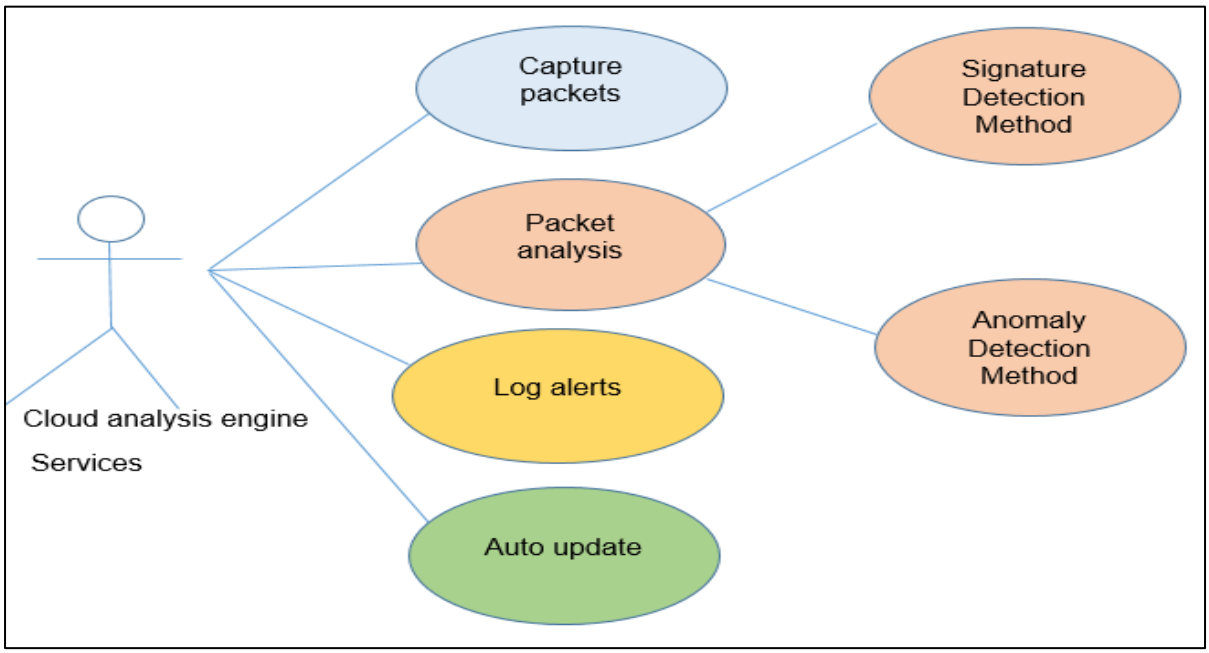


Figure 5-2: Cloud analysis engine USE case diagram

The cloud analysis engine USE case diagram identifies one main actor with four main actions. Below is a brief discussion of the main actions in table format:

© Central University of Technology, Free State

Table 5-2: Cloud analysis engine service: Capture packets

Actor	Capture packets service
Pre-conditions	Client agent is configured to direct traffic to the hosting cloud server
Description	All the traffic originating from the client mobile phone will be analysed by the HIDS to be scrutinised by the detection engine
Exceptions	N/A
Post-conditions:	Mobile device's outgoing traffic is captured and ready for analysis

Table 5-3: Cloud analysis engine service: Packet analysis

Actor	Packet analysis service
Pre-conditions	Mobile phone's incoming and outgoing traffic is captured
Description	Signature- and anomaly-based detection techniques are executed against the sniffed traffic to detect misbehaviour and suspicious packets
Exceptions	Packets that follow abnormal behaviour is warned without being blocked
Post-conditions:	Mobile phone's incoming and outgoing traffic is analysed

Table 5-4: Cloud analysis engine service: Log alerts

Actor	Log alerts service
Pre-conditions	Infected packets are detected
Description	Once misbehaviour and suspicious packets are detected, it will be stored in the log repository or database
Exceptions	N/A
Post-conditions:	Signature database is updated with the recently detected intrusions

Table 5-5: Cloud analysis engine service: Auto update

© Central University of Technology, Free State

Actor	Auto update service
Pre-conditions	New signatures are added to the signature database
Description	Must be able to define criteria to recognise attacks and system vulnerabilities
Exceptions	N/A
Post-conditions:	Automatically updates database with new signatures

5.2.2 Client USE case

The following figure (Figure 5.3) depicts the client USE case diagram – the client of the CIDPS being the mobile voter.

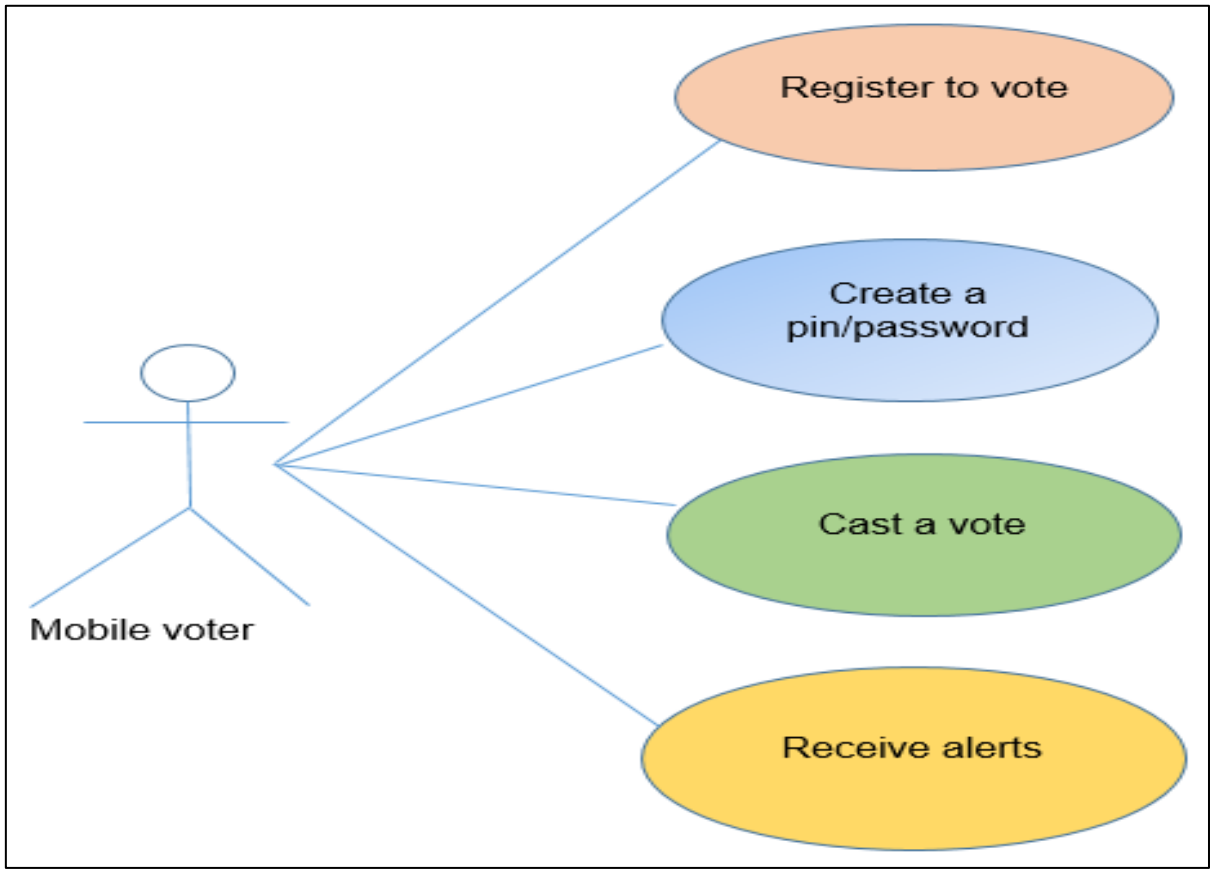


Figure 5-3: Client USE case diagram

The client USE case diagram identifies one main actor with four main actions. Below is a brief discussion of each of the identified actions in table format:

© Central University of Technology, Free State

Table 5-6: Mobile voter: Register to vote

Actor	Mobile voter: Register to vote
Pre-conditions	• Connected to Wi-Fi/mobile data • Download XaP application • Register to vote
Exceptions	None
Post-conditions:	Warning notification message appears to users if they attempt to do anything other than registering

Table 5-7: Mobile voter: Create a pin/password

Actor	Mobile voter: Create a pin/password
Pre-conditions	• Connected to Wi-Fi/mobile data • Download XaP application • Register to vote • Create a pin • Cast a vote
Description	ID number and pin are required to cast a vote
Exceptions	Cast a vote
Post-conditions:	Warning notification message appears to users if they attempt to do anything other than voting

Table 5-8: Mobile voter: Cast a vote

Actor	Mobile voter: Cast a vote
Pre-conditions	• Log in on XaP application • Cast a vote
Exceptions	None
Post-conditions:	Warning notification message appears to users if they attempt to do anything other than voting

© Central University of Technology, Free State

Table 5-9: Mobile voter: Receive alerts

Actor	Mobile voter: Receive alerts
Pre-conditions	• Mobile phone connected to Wi-Fi access point or mobile data • VPN established between client and service endpoints • HIDS service is up and running • Client agent monitors XaP application installed on the mobile phone
Description	Client agent will keep monitoring the XaP application and will keep listening to CIDPS for new intrusions detected; the client agent will show warning notification to the voter
Exceptions	N/A
Post-conditions:	Warning notification message appears to the voter

5.2.3 CIDPS flowchart

A CIDPS for m-voting in SA is proposed in this study. Figure 5.4 exhibits the key processes of a CIDPS in the form of a flowchart. The flowchart depicts the movement of information between the mobile voter, the client agent and the analysis engine.

© Central University of Technology, Free State

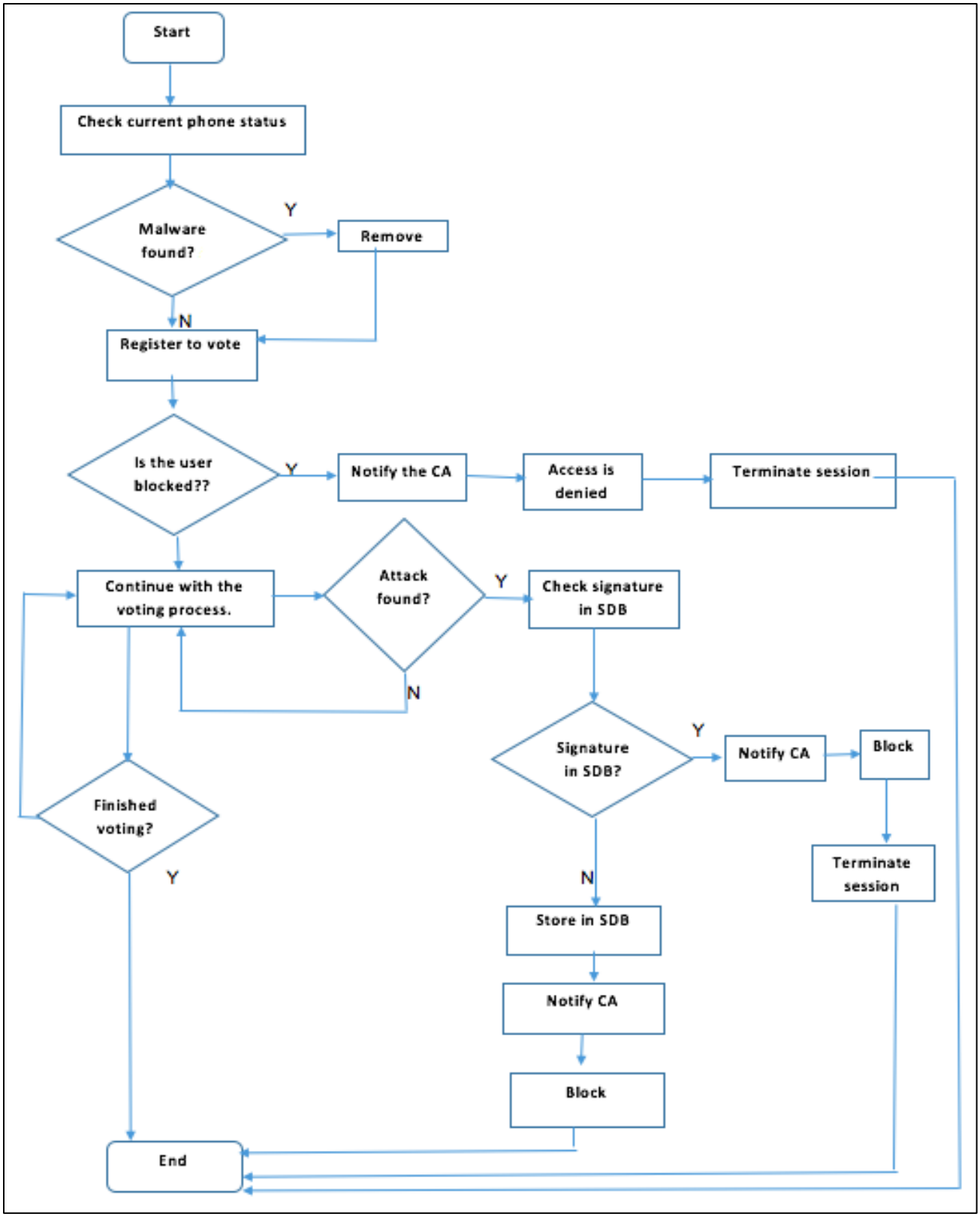


Figure 5-4: CIDPS flowchart

© Central University of Technology, Free State

The mobile phone voter has to download and install the XaP application; once it is installed, the CIDPS becomes active and ready to begin its work. The client agent collects the user's inputs as he or she registers to vote. Once the user clicks the register button, the cloud analysis agent checks if the user has been blocked. If the user is blocked, the CIDPS informs the client agent about the status of the user, access is denied and the system automatically logs the user out. If the user is not blocked, he or she continues with the voting process while the client agent monitors and collects user-sensor inputs and outputs from the device interface in runtime and keeps on sending it to the cloud analysis engine for an intensive malware scan.

If intrusion or malware is detected, the system checks if the malware signature is known. When the malware signature is known, the cloud analysis agent blocks and removes the intrusion and notifies the client agent. If the malware signature is not known – meaning no match is found in the signature database – it is stored in the signature database and the threat is blocked and removed. If no malware or intrusions are detected, the voter continues casting a vote.

The next section discusses the technologies and methodologies used to develop the CIDPS.

5.3 CIDPS TECHNOLOGIES

As previously mentioned, the aim of this study was to compare and evaluate a suitable CIDPS for mobile voting in SA.

Table 5.10 below shows a list of CIDPS technologies and available options, some of which were used for the proposed security system.

© Central University of Technology, Free State

Table 5-10: CIDPS technologies

Components	Available options
Analysis engines	Snort Suricata
Cloud storage	Amazon EC2 Google app engine Microsoft Azure
Simulation tools	Android Emulator VMware Genymotion

Each of the above will now be briefly described:

5.3.1 Analysis engines

Analysis engines, also known as packet sniffers, are technologies that are used to monitor legitimate or bogus traffic on a network. Two analysis engines were used during this study:

1. Snort

Snort is a packet analyser and recorder used as a lightweight IDS (Xing, Huang, Xu, Chung & Khatkar, 2013). Snort was originally established in 1998 by Martin Roesch to examine the functions of an application layer of network data packets (Khamphakdee *et al.*, 2015; Roesch, 1999). It is one of the most common and widely used open-source signature-based intrusion detection engines (Khamphakdee *et al.*, 2015).

Snort is designed to operate for long periods without the need for monitoring or administrative maintenance, and it can also be utilised as a central part of network security infrastructures. Snort is an open-source solution, a mature product that has been available for more than a decade (Khamphakdee *et al.*, 2015). Its current modular design is known as Snort 2.9. Snort permits developers to build and add additional features without the need to change anything on the detection engine (Khamphakdee *et al.*, 2015).

© Central University of Technology, Free State

In this study, Snort was utilised because it is a packet-oriented, signature-based, and open-source IDPS (Ahmad *et al*., 2013a). It produces alarms using signature rules and has a language to explain new rules. Its architecture makes it possible to add new functionalities at the time of compilation (Khamphakdee *et al*., 2015). The filtered traffic passes through the Snort attack detection engine in order for various active and passive attacks to be detected. With the help of Snort, the researcher was able to analyse and monitor live traffic and packet flow along with anomalies in the network.

Snort consists of the following four components, illustrated in Figure 5.5:

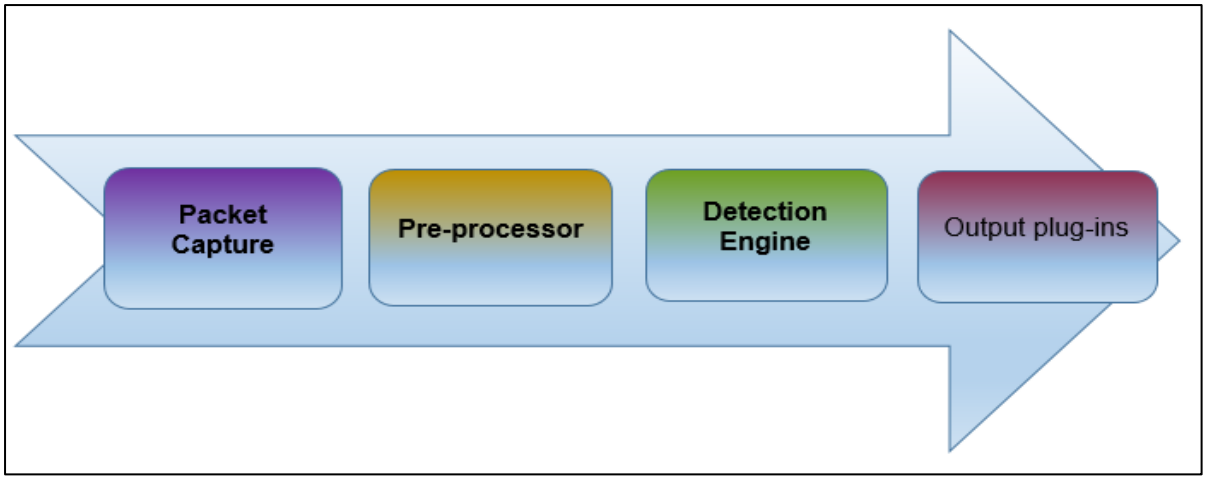


Figure 5-5: Snort components (Aydin *et al.,* 2010)

Below is a description of each component (Roesch, 1999):

- **Packet capture:** The Snort analysis engine uses the packet-capturing library (libpcap) written in Lawrence Berkeley National Laboratories. Analysis engines handle captured packets in order for it to conform to the network-level. The packets needs re-decoding for upper-level protocols, which are transmission control protocol (TCP) and User Data Protocol(UDP) for any mobile device.

- **Pre-processor plug-ins:** This step intends to inspect and process packets before they are sent to the detection engine. Every pre-processor scrutinises the packets for a different attribute and decides whether or not to pass the packet to the detection engine without changes.

- **Detection engine:** The detection engine tests packets for different attributes specified in the Snort rules definition file. Detection plug-ins offer extra detection functions.

© Central University of Technology, Free State

- **Output plug-ins:** The Snort output plug-ins receive triggered alarms from the detection engine, pre-processors or decoding engine.

2. Suricata

Comparable to Snort, Suricata is also an open-source IDPS (Catania & Garino, 2012). Suricata was established by the Open Information Security Foundation (OISF) in 2009 and was only released in 2010 (Albin & Rowe, 2012; White, Fitzsimmons & Matthews, 2013).

It is a multi-threaded system that delivers a higher performance and improved scalability than Snort, which is single threaded (White *et al.*, 2013). This analysis engine inspects incoming data until the internet connection is terminated (Catania & Garino, 2012). The packets are inspected using the sliding window concept (as depicted in Figure 5.6 below).

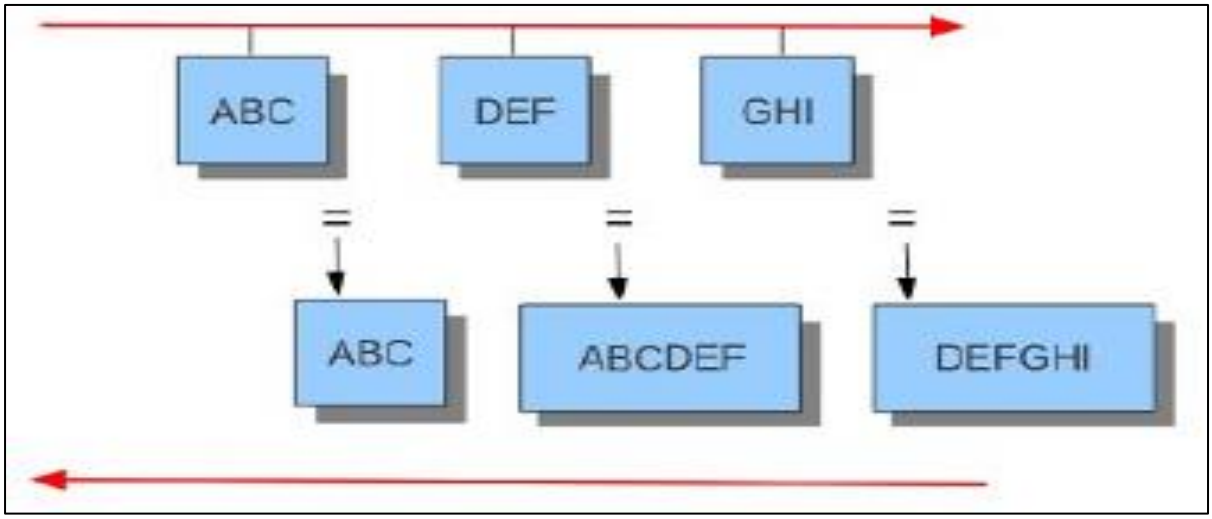


Figure 5-6: Sliding window used in Suricata (Day & Burns, 2011)

According to Jiang, *Z*hang, Xie, Salamatian and Mathy (2013), Suricata receives the first segment and immediately inspects it. Then it receives the second segment, puts it together with the first, and inspects it. At the end, it catches the third segment, cuts off the first one, puts together the second segment with the third segment and inspects it. Once the full packet is inspected and no malware was found or intrusion detected, the packet is sent to the receiver (Albin & Rowe, 2012).

© Central University of Technology, Free State

Also, Suricata has two modes that act on intrusion when it is detected, namely a Drop mode and/or a Reject mode (Day & Burns, 2011):

- **Drop (IPS mode):**
  - If a signature containing a drop action matches a packet, it is immediately discarded and will not be send any further.
  - The receiver does not receive the message, resulting in a time-out.
  - All following packets from the same sender are dropped.
  - An alert for a packet is generated.

- **Reject (both IDS and IPS mode):**
  - This is an active rejection of the packet; both the receiver and sender receive a rejected packet.
  - If the packet concerns TCP, it will be a reset-packet, otherwise it will be an ICMP-error packet.
  - An alert for a packet is generated.

White *et al.* (2013) state that Suricata is capable of lower average memory usage and lower average CPU utilisation than Snort. Another study that proves the effectiveness of Suricata was conducted by Day and Burns (2011). Their study, which analysed the performance of Snort and Suricata network intrusion detection and prevention engines, revealed that Suricata has the potential to be more flexible and efficient than Snort (Day & Burns; 2011).

3. Comparing Snort with Suricata

The following comparison (see Table 5.11) between Snort and Suricata is made on the basis of different parameters, such as signatures, false alarm, flexibility, deployment, interface, and OS capability.

© Central University of Technology, Free State

Table 5-11: Comparison of Snort and Suricata

Parameter	Snort	Suricata
Contextual signatures	No	Yes
Flexible site customisation	Medium	High
False alarm	Medium	Small
Large user community	Yes	Yes
Configuration GUI	Yes	Yes
Installation /deployment	Easy	Easy
OS compatibility	Any OS	Any OS

When choosing an IDPS, the two signature-based detection engines can both function effectively depending on the environment and the deployment of the rule set (Day & Burns, 2011). The development in the Suricata engine allowed the researcher to employ multi-threaded operations for the simulation, which is very important and crucial when it comes to network bandwidth (White *et al.*, 2013). Both Snort and Suricata were used in the simulations and their results were compared.

5.3.2 Cloud storage

Cloud storage is a service where data is remotely maintained, managed, and backed-up (Mehmood *et al.*, 2013; Wang *et al.*, 2010). The researcher had to choose between the three most popular cloud storage facilities, briefly described below:

1. Amazon EC2

The Amazon Elastic Compute Cloud (Amazon EC2) web service allows users to execute applications in computing environment. For example, a statistician can work with an R-server operating on EC2 as if it was local to his or her device. The platform hides the density of the cloud-computing infrastructure and the R-server is accessed with a simple URL. Several statisticians can connect all at once to the same EC2-R server and analyse data collaboratively via a set of broadcasted views (Khamphakdee *et al.*, 2015).

2. Google App Engine

Houmansadr *et al.* (2011) state that the Google App Engine is a cloud computing platform for developing web-based applications. With this tool, developers are able to

© Central University of Technology, Free State

code applications without worrying about server management or hardware configurations, because this is managed by the engine (Manzalini & Crespi, 2015). The Google App Engine Development Kit offers tools for developing and testing the entire application locally. Google App Engine also scales all the computing resources that an application may need automatically (Houmansadr *et al.*, 2011; Manzalini & Crespi, 2015; Srivastava *et al*., 2012).

3. Microsoft Azure

According to Mehmood *et al*. (2013), Microsoft Azure is a cloud storage system that offers customers the ability to store vast amounts of data, and it can be stored for as long as the user wants. Data stored in Microsoft Azure is long-lasting due to the local and geographic duplication that aids disaster recovery. The storage consists of blobs (user's files), tables (structured storage), and queues (messaging). The data is highly durable, available, and immensely accessible. It is exposed through REST APIs, client libraries in .NET, Java, Node.js, Python, PHP, and Ruby (Luo *et al*., 2011; Mehmood *et al*., 2013; Srivastava *et al*., 2012).

As mentioned in Chapter 1, the main reason for running an IDPS on the cloud and not directly on a mobile device, is because of the limited computational resources of the mobile phone. Hence, the researcher used Microsoft Azure as the cloud platform for this study. Microsoft Azure provides the storage that the proposed system needs and allows the data to be durable and available, as well as accessible.

The researcher used the REST API called CloudRail for configuring the analysis engines. The CloudRail REST API permits access to various cloud services via a single interface (Paikaray *et al.*, 2015). Data flows from one point to the other without a middleware, whilst everything stays automatically up to date. It is also free of charge (Luo *et al.*, 2011).

© Central University of Technology, Free State

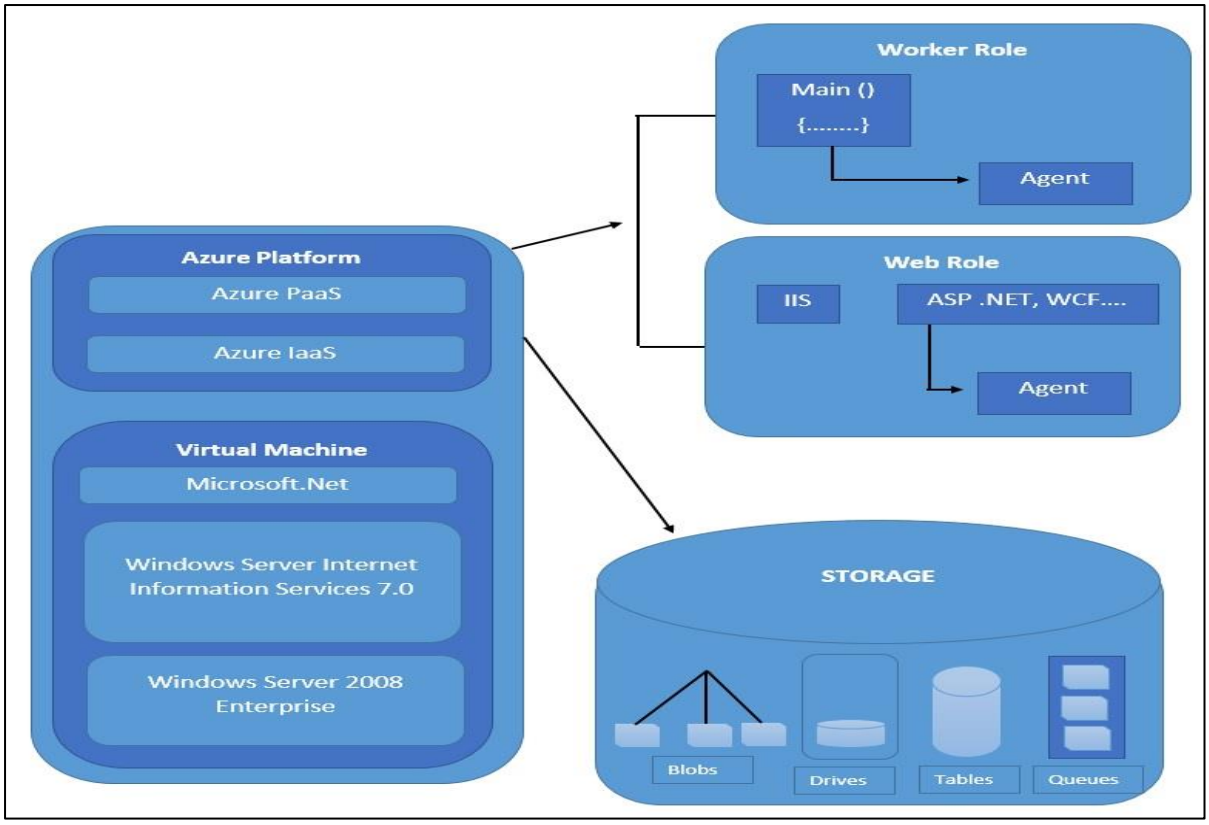


Figure 5-7: Basic architecture of Microsoft Azure (Mehmood *et al.*, 2013)

By integrating CloudRail into an application like XaP, developers can access Dropbox, Google Drive, Microsoft OneDrive, Box, and more (Mehmood *et al*., 2013). Figure 5.7 above, illustrates the architecture and service offering of Microsoft Azure .

5.3.3 Simulation tools

Simulation tools allow developers to develop and test their model. This is usually done when implementing the model in a real-world environment is not feasible (Lo *et al.*, 2010). The most common mobile simulation tools are discussed below:

1. Android Emulator

Android Emulator is a tool that can be used by Android developers to simulate any Android device and Android application (Malkawi, Khasawneh, Al-Jarrah & Barakat, 2009). The Android Emulator has two components: the Android Virtual Device (AVD) Manager and the Android Virtual Devices. The AVD manager provides a graphical user-interface in which a developer can create, test, and manage AVDs which are required by the Android Emulator (Rashida, 2013). An AVD is a configuration that defines the characteristics of an Android phone, tablet, Android Wear or Android TV

© Central University of Technology, Free State

device that a developer wants to simulate in the Android Emulator (Gandhewar & Sheikh, 2010).

2. VMware

VMware is a network virtualisation product where a physical server can be partitioned into multiple virtual machines (Gandhewar & Sheikh, 2010). VMware provides extensive virtualisation, management, resource optimisation, application availability, and operational automation abilities in a combined subscription. It works with Windows, Linux, Netware or any other OS. Recently, VMware introduced the Mobile Virtualisation platform (MVP) which allows users to run multiple OS instances on top of a mobile phone's physical platform. That means anyone can manage a personal and work phone on one device (Rashida, 2013).

3. Genymotion Emulator

Genymotion is an Android emulator which is very easy to use and is based on VirtualBox. The Oracle VM VirtualBox installs automatically as the developer installs Genymotion and it enables the virtualisation of Android OSs (Kushwaha & Kushwaha, 2011). Genymotion can emulate specific devices and allows the developer to install, run, and test applications on the devices. Genymotion has a version for both Windows and Mac computers and also a free version (Mahmood, Esfahani, Kacem, Mirzaei, Malek & Stavrou, 2012).

This study used Genymotion as a simulation tool. Genymotion provides better performance compared to Android Emulator and VMware. Genymotion runs on top of x86 architecture through VirtualBox. It also offers more features than the other two emulators, such as filtering through Android devices, adding new Android devices, battery and mobile device sensors, and easy-to-use widgets.

5.4 SUMMARY

In this chapter the architecture and the technologies required for the proposed security solution were presented. The key role players of the proposed system were listed and their roles and responsibilities explained. USE cases and a flowchart were used to present the proposed security solution.

© Central University of Technology, Free State

This chapter theoretically compared and evaluated two analysis engines, namely Snort and Suricata. Three cloud storage providers were also compared – Microsoft Azure was chosen as the most appropriate cloud platform and storage facility for the XaP application and the analysis engines. For evaluation of the two analysis engines, Genymotion was chosen as simulation tool.

A summary of the technologies discussed in this chapter and which were used for the comparison and evaluation of the CIDPS is portrayed in Figure 5.8:

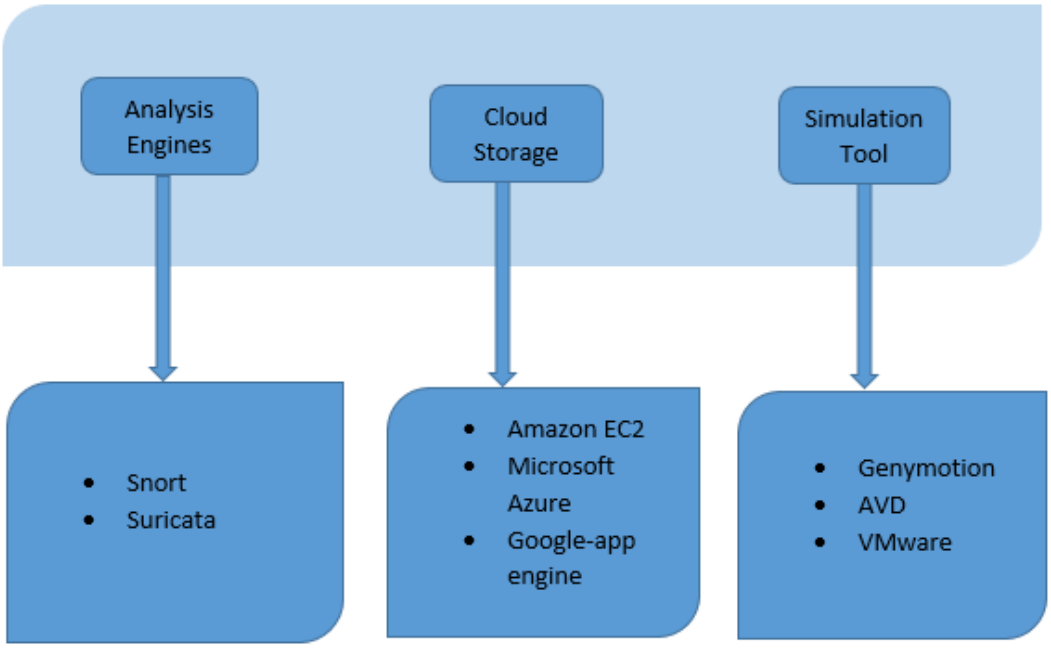
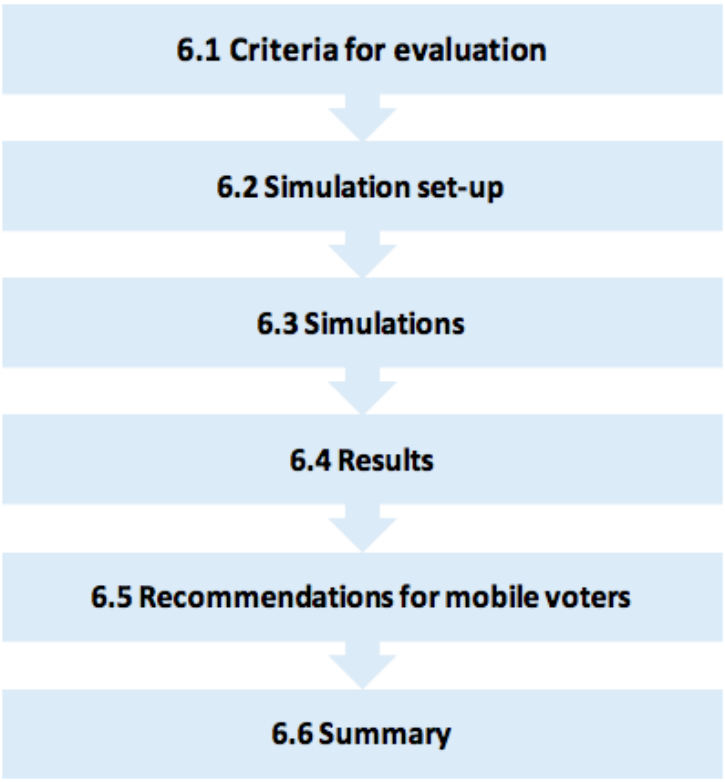


Figure 5-8: CIDPS technologies

The next chapter is an evaluation of the two analysis engines in order to choose the one most suitable for secure m-voting. The chapter presents the evaluation set-up, the evaluation process and finally, the results.

© Central University of Technology, Free State

CHAPTER 6: SYSTEM EVALUATION



© Central University of Technology, Free State

This chapter describes the evaluation process of the proposed security solution. The chapter is organised into six sections. In Section 6.1 the criteria used to evaluate the proposed security system are discussed. Section 6.2 presents the simulation set-up and Section 6.3 discusses the simulation, while Section 6.4 presents the results. Recommendations for mobile voters are presented in Section 6.5 and the chapter concludes with a summary in Section 6.6.

6.1 CRITERIA FOR EVALUATION

With the specific end goal to perform a comprehensive evaluation of the proposed security system, suitable evaluation criteria that tended to system execution issues were produced. These criteria are portrayed in Table 6.1 underneath, as well as in Chapter 4, Section 4.6.

Table 6-1: CIDPS evaluation criteria

Criteria	Description	References
Accuracy	Correct detection of intrusions and the nonappearance of false alarms.	Abduvaliyev *et al*. (2010); Gupta *et al*. (2013); Volden *et al*. (2011)
Detection rate	The rate at which traffic and audit events are processed.	Abduvaliyev *et al*. (2010); Gupta *et al*. (2013); Volden *et al*. (2011)
Detection time	Time elapsed between intrusion and detection.	Abduvaliyev *et al*. (2010); Gupta *et al*. (2013); Volden *et al*.(2011)

To effectively evaluate the CIDPS's performance and detection accuracy, the ratio of four possible events was scrutinised. The four possible events, which occur upon alerts produced by the IDS, are: true-positive (TP), true-negative (TN), false-positive (FP), and false-negative (FN).

The TP appears when the detection engine triggers an alarm when an intrusion is detected (Lin *et al.*, 2015). The TN occurs when a detection engine triggers no alarm because no intrusion was detected (Elhag *et al.*, 2015). The FP happens when an alarm is triggered even when there was no intrusion detected (Ravale *et al.*, 2015).

© Central University of Technology, Free State

The most dangerous of conditions is the FN, which is a situation in which the detection engine fails to detect intrusion, thereby allowing it to go into the network without notice (Elhag *et al*., 2015).

Accuracy in intrusion detection is determined by the sum total of FP and FN alerts created by the detection engine (Lin *et al*., 2015). FP and FN conditions can happen in various ways. Defective rule configuration, including invalid signature data or improper rule language, is one way, yet this is unusual. Rules may likewise be inaccurate, because there is no unique signature for a particular intrusion (Elhag *et al.*, 2015). These kinds of issues would outcome in the occurrence of FP or FN alerts in all intrusion detection engines using similar rules. FP and FN alerts might also happen because of detection engine execution issues (Ravale *et al.*, 2015).

The following section describes the simulation set-up.

6.2 SIMULATION SET-UP

This study made use of two investigators (one was situated in Bloemfontein and another one in Welkom, Free State) to run the simulations. This arrangement offered investigator, environmental, and data triangulation. The simulation set-up comprised the following components:

1. Computers to run the simulation
2. M-voting system
3. Client agent
4. Analysis engine
5. Intruder
6. Simulation tool

6.2.1 M-voting system

Mpekoa (2014) developed the XaP application for casting a vote via a mobile phone. XaP is a fully developed system that is currently available on Google store and is accessible to everyone. Figure 6.1 depicts the availability of XaP on Google store.

© Central University of Technology, Free State

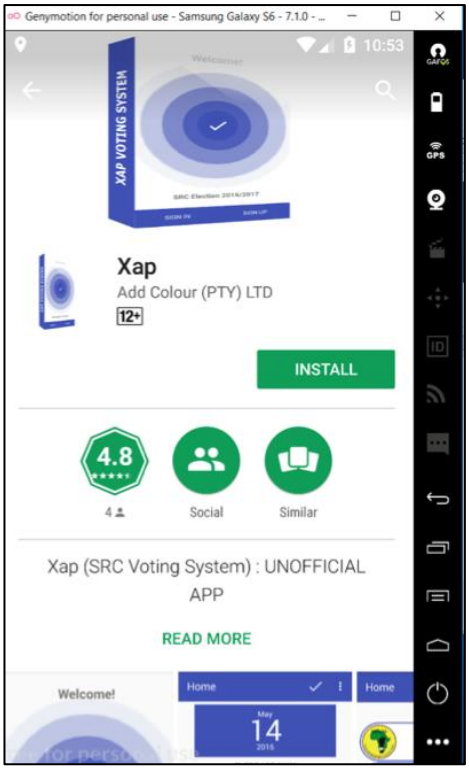


Figure 6-1: XaP on Google store

Firstly, the voter has to download and install the XaP application on his or her mobile phone. Once it is downloaded and installed, the voter has to open XaP to sign up, as depicted in Figure 6.2 below.

Figure 6-2: XaP sign up page

© Central University of Technology, Free State

For demonstration purposes, the mobile voting application was designed for Student Representative Council (SRC) elections, as seen in the figure above. For the sign-up process, voters have to put in their student number, surname and initials, and a password. Once they have signed up, they are allowed to sign in using their credentials (student number and password), as seen in Figure 6.3 below.

Figure 6-3: XaP sign in page

6.2.2 Client agent

The client agent which was developed by the researcher for the proposed system is a back-end system that connects XaP and the cloud IDPS (Snort/Suricata), as indicated by Figure 6.4 below.

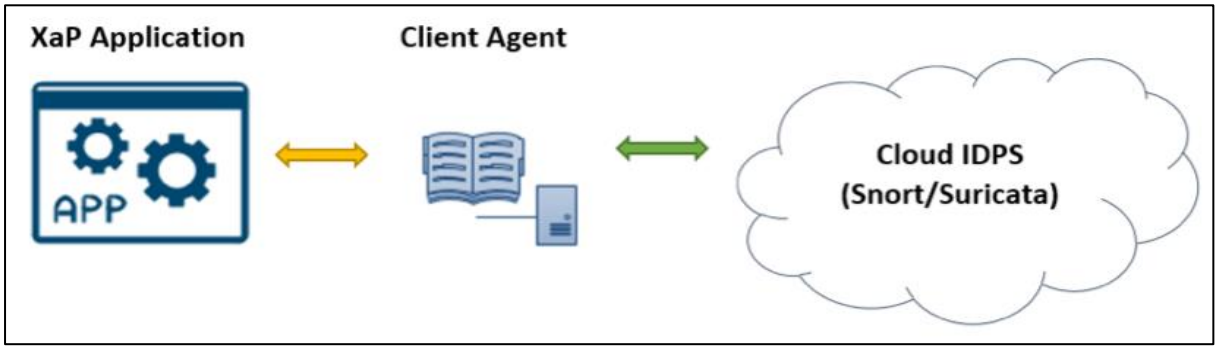


Figure 6-4: Client agent with XaP and the cloud IDPS

Immediately when the voter signs into the voting application, the client agent begins to gather information about the mobile phone and sends the mobile phone's status to the cloud analysis engine for detailed analysis. The cloud analysis engine checks for

© Central University of Technology, Free State

viruses, intrusions, and other malware. Once this analysis has been done, a message alert is sent to the client agent. If the mobile phone is “clean” and without viruses, the client agent will open the full functionality of XaP (to allow the voter to cast a vote). If a virus or worm has been found on the mobile phone, a message is sent to the voter, indicating steps on how to fix the problem (e.g., disable the wireless interface that is being attacked).

When the voter casts the vote, the client agent listens to all ports on the mobile phone and monitors any changes that occur, for example, incoming SMSs and calls. Any changes in the status of the mobile phone are sent to the cloud analysis engine. To avoid opening any form of communication channel that may make the mobile phone vulnerable to threats and attacks, the voter cannot access any other applications or perform any activities with his or her mobile phone once the XaP application is opened. If there is an incoming phone call while using XaP, the client agent will notify the voter by sending a message to him or her.

6.2.3 Analysis engine

This study made use of two analysis engines, namely Snort Version 2.9 and Suricata Version 4.0. Snort and Suricata have been discussed in detail in Chapter 4, Section 4.3.1.

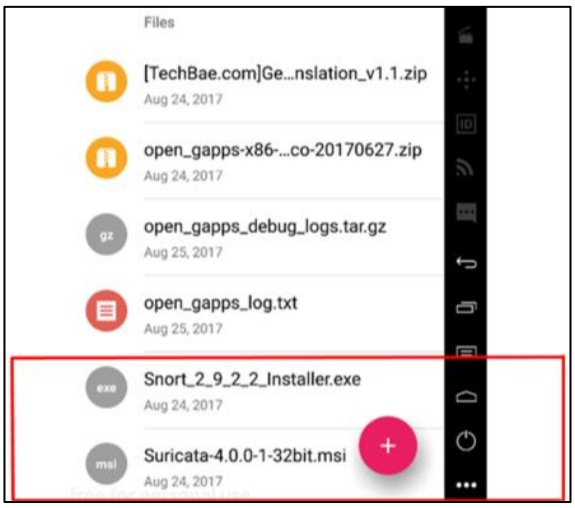


Figure 6-5: Snort and Suricata after drag-and-drop to Genymotion

Figure 6.5 above depicts the position of Snort and Suricata after they were dragged and dropped into Genymotion.

© Central University of Technology, Free State

6.2.4 Intruder

According to Symantec, the American software company, an intrusion signature is the information trail left behind by the attackers that can be picked by the analysis engine to identify an attack (Elhag *et al.*, 2015; Patel *et al*., 2013). Pytbull is an IDS/IPS testing tool for Snort, Suricata, and any other IDS/IPS that produces an alert file. The testing tool includes 300 tests, grouped in 11 testing modules:

- **Bad Traffic**: Non-RFC packets are sent to the server for additional preparations.
- **Brute Force**: checks the ability of the server to distinguish brute force attacks (e.g., FTP) and furthermore uses custom principles for Snort and Suricata.
- **clientSideAttacks**: this module uses a reverse shell with the intention to direct the server in downloading remote harmful data. This module challenges the capacity of the IDS/IPS to shield against client-side attacks.
- **denialOfService**: looks at the limit of the IDS/IPS to shield against DoS attacks.
- **Evasion Techniques**: a few avoidance techniques are used to test whether the IDS/IPS can recognise them.
- **Fragmented Packets**: various broken-up payloads are dispatched to the server to check or look at its capacity to recompose them and detect the attacks.
- **ipReputation**: checks the capacity of the server to identify traffic from/to low reputation servers.
- .**Normal Usage**: activities that are parallel to normal usage.
- **pcapReplay**: empowers to replay pcap records.
- **Shellcodes**: guide distinctive shell-codes to the server on port 21/TCP to check the capacity of the server to recognise/dismiss shell-codes.
- **Test Rules**: essential rules for testing. These intrusions are developed to be picked up by the rule sets incorporated into the IDS/IPS.

There are mainly five types of tests:

- **Socket**: opens a socket on a specific port and guides the payloads to the remote target on the same port.
- **Command**: directs a command to the remote target with the subprocess.call() python function.
- **Scapy**: refers to distinct crafted payloads created on the Scapy syntax.

© Central University of Technology, Free State

- **Client side attacks**: uses a turnaround shell on the remote target and sends commands to it to make them being processed by the server (typically wget commands).
- **Pcap replay**: allows echoing of traffic based on pcap files.

In this study, Pytbull was used to test the detection and blocking abilities of both Snort and Suricata. The performance of these two analysis engines were then compared with each other. Configuration modifications were also compared and lastly, configurations were checked and/or validated.

6.2.5 Simulation tool

The simulation tool that was used is called Genymotion. Genymotion was installed on the Dell laptops, as well as the virtualisation application named Oracle VM VirtualBox Version 5.0.28. (Figure 6.6 and Figure 6.7)

Figure 6-6: Oracle VM VirtualBox setup wizard

© Central University of Technology, Free State

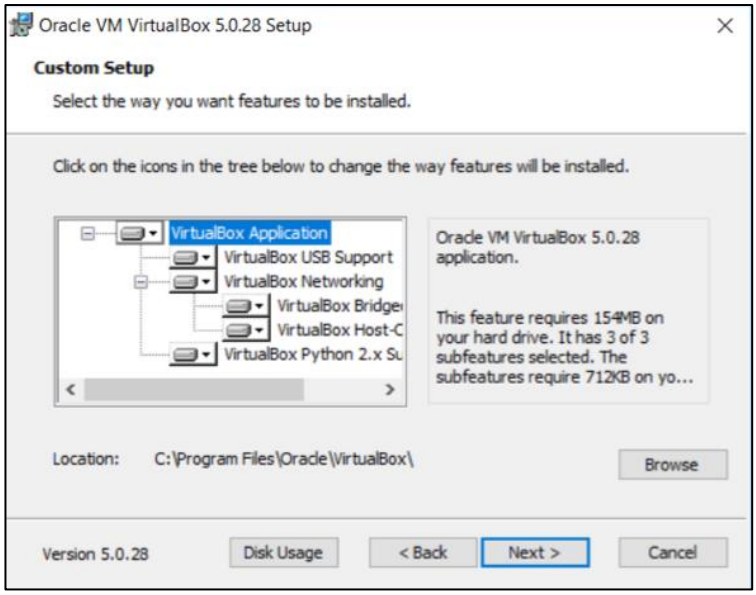


Figure 6-7: Oracle VM VirtualBox installation

Once Oracle VM VirtualBox and Genymotion were successfully installed, the researcher could create a virtual device. Any device could be chosen from a huge list of Android devices; for the purpose of this experiment, the Samsung Galaxy S6 mobile phone was used, as indicated in Figure 6.8 below.

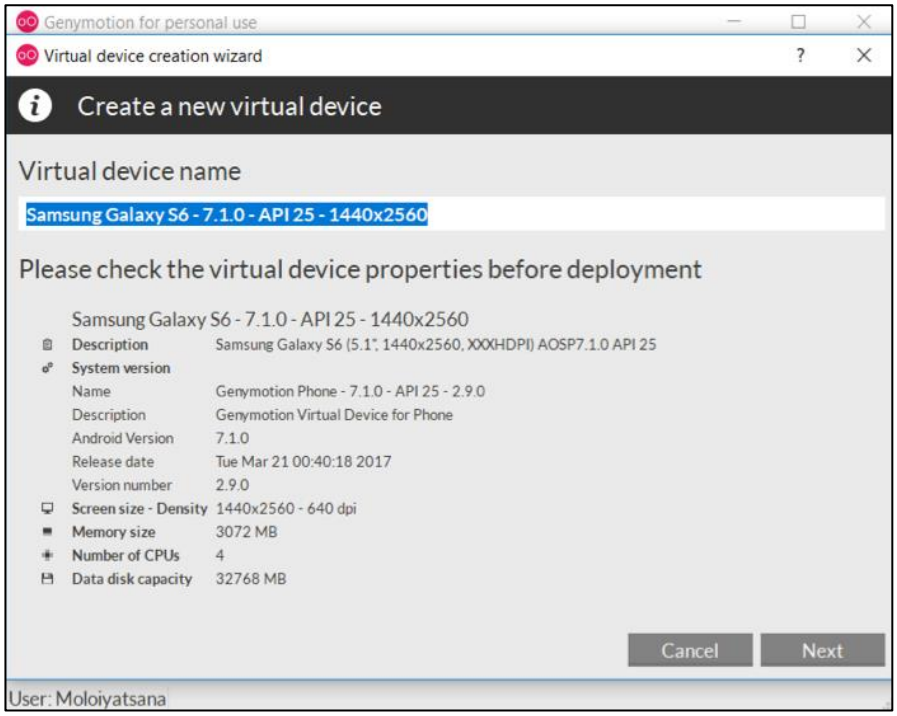


Figure 6-8: Creating a new virtual device

Once the new virtual device was in place, the virtualisation application was upgraded to VirtualBox 5.1.26. at the time of the experiment, as this was the latest version. The

© Central University of Technology, Free State

Samsung S6 was deployed successfully, however, the virtual mobile phone did not have access to Google Play Store.

A plug-in allowing access to Google Play Store by the virtual mobile phone had to be added. The virtual phone was then flashed with ARM-Translation Version 1.1, where after it was restarted to effect the changes. Figure 6.9 depicts this process:

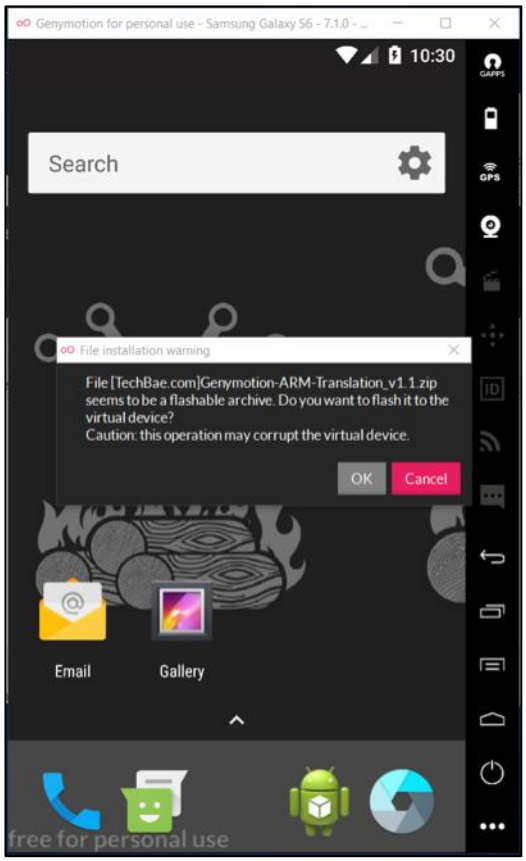


Figure 6-9: Flashing the ARM-Translation application

Google Play Store was available on the virtual mobile phone after it was restarted. The store was searched for the XaP application and the results are shown in Figure 6.10 below:

© Central University of Technology, Free State

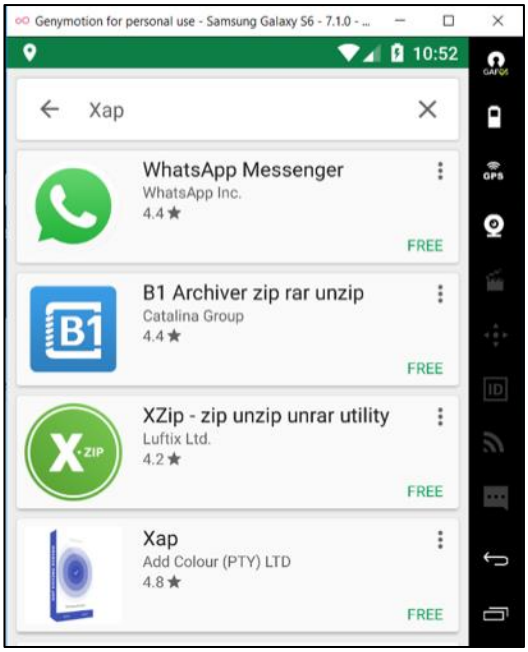


Figure 6-10: XaP in the Google Play Store

XaP was downloaded and then installed on the virtual mobile phone, as shown in Figure 6.11 below.

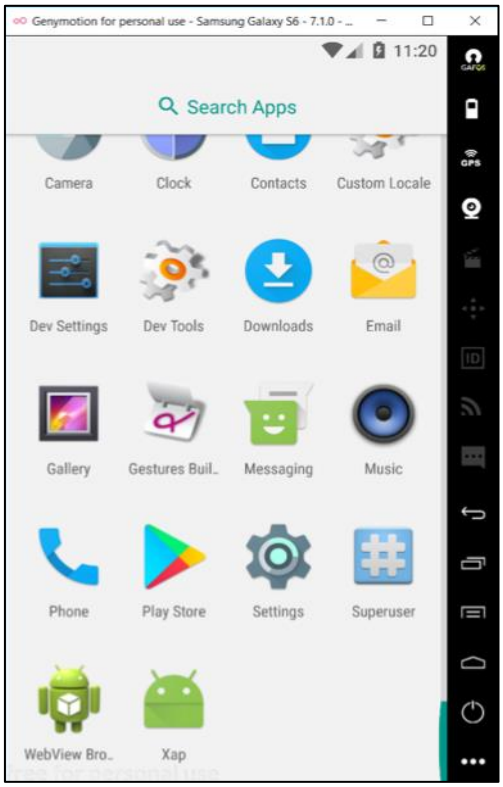


Figure 6-11: XaP after installation

© Central University of Technology, Free State

After XaP had been installed on the virtual mobile phone, the client agent was also automatically installed. When XaP was opened, the client agent automatically ran in the background.

6.3 SIMULATIONS

Each investigator ran two simulations for Snort and Suricata. They also had to run two instances of each detection engine (two times Snort and two times Suricata) in order to validate the first instance. Each instance ran separately and independently from the other one. In this way, the researcher could measure how well each IDS detects various malicious packets. The experiment occurred over several days.

As mentioned earlier, the experiment tested how precisely or accurately Snort and Suricata detected malicious or abnormal traffic. Using Pytbull, the experiment produced numerous tests comprising suspicious or malicious payloads and sent them through to the analysis engines to generate alerts where necessary. These tests were separated into nine groups: malformed traffic, client-side attacks, DoS, analysis engine evasion, fragmented packets, malware identification, shell-code, common rule testing, and failed authentication.

To put the simulation in motion, the researcher started off by opening Genymotion to start the detection engine (either Snort or Suricata). After the detection engine was launched and began to examine the network interface, Pytbull was run, directed at the address of the virtual mobile phone. The application finished the task list, exited, and created an HTML report indicating the attempted intrusions and the alert response from the analysis engines, if any.

At this point, the researcher paused all the services and the analysis engines. Performance information was recorded, analysed, and collated. The outcomes are discussed in the following section.

6.4 RESULTS

The results revealed that for most rules, Suricata produced more alarms than Snort with regard to the same network traffic. Despite the fact that both analysis engines loaded the same rule sets, error messages occurred and a few rules did not perhaps load properly on one analysis engine. Other reasons for the discrepancy could be a

© Central University of Technology, Free State

bug in the implementation of the rules on an engine, or an issue with the algorithm used to examine the traffic. Additional research on the packet level is needed to determine what had happened in each situation.

This experiment evaluated the detection accuracy of Snort and Suricata when exposed to known and unknown malicious packets. The researcher conducted 54 tests divided into nine categories, on both Snort and Suricata. The results are presented below:

6.4.1 Performance of the analysis engines

1. Suricata

Suricata generated 12 FN alerts where it did not detect the malicious traffic. The most likely reason for these FN alerts is that the rule used was not loaded or optimised for the particular threat.

During the client-side attack, Suricata detected both tests, but during the evasion-technique attack, it did not recognise that an evasion attack was underway.

FP alerts were more challenging to measure taking in consideration the composite nature of the Pytbull tests. If an alarm is an obvious FP, such as an alarm for a Trojan infection during an Nmap scan, it will be categorised as a FP. Nmap is a protocol that is used to discover hosts, protocols, open ports, services, and vulnerabilities on networks (Albin & Rowe, 2012; Jiang *et al*., 2013).

Table 6.2 below presents the summarised results from the Suricata experiment:

Table 6-2: Summary of Suricata alerts

Test Category	Suricata True (+)	Suricata True (-)	Suricata False (+)	Suricata False (-)
Client-Side attack	75	78	8	11
Test rules	85	56	9	12
Bad traffic	77	77	13	15
Fragmented packets	51	77	18	16
Multiple failed logins	65	65	21	9
Evasion techniques	89	91	15	9

© Central University of Technology, Free State

Test Category	Suricata True (+)	Suricata True (-)	Suricata False (+)	Suricata False (-)
Shell-codes	81	58	14	10
DoS	63	63	14	34
PCAP replay	61	30	47	12

2. Snort

Snort generated 16 FN alerts where it did not detect the malicious traffic. As was the case with Suricata, FN alerts were observed and a possible reason for this is that the rule used by both was not loaded or optimised for the particular threat.

During the client-side attack, Snort detected only one test and during the evasion-technique attack, it recognised that an evasion attempt was underway (compared to Suricata which did not detect the attack).

Table 6.3 presents the summarised results from the Snort experiment:

Table 6-3: Summary of Snort alerts

Test Category	Snort True (+)	Snort True (-)	Snort False (+)	Snort False (-)
Client-Side attack	71	68	10	13
Test rules	84	56	10	16
Bad traffic	77	77	12	12
Fragmented packets	51	55	16	16
Multiple failed logins	65	71	19	11
Evasion techniques	70	75	17	9
Shell-codes	80	57	15	7
DoS	63	63	11	9
PCAP replay	61	31	43	10

3. Comparison of Snort and Suricata

Below is the analysis of the detection success for each category of attacks as depicted in Table 6.2 and Table 6.3 above:

© Central University of Technology, Free State

- **Bad traffic**: These tests contained unusual network packets in which either the flags in the TCP header were not set correctly, or the type of packet did not correspond to its header. This test used Nmap and Scapy which are packet manipulating tools for a network to produce malformed packets (Day & Burns, 2011). Both Snort and Suricata recognised one out of the three malformed packet tests. However, none of the analysis engines created an alert on the Scapy-modified packets.
- **Client-side attacks**: The tests simulated the action after the user has downloaded the XaP application. The system must be able to pick up any malware that may have infected XaP during the downloading process. Suricata produced 75 TP alerts from the malicious downloads. On the same rule set, Snort generated 71 TP alerts. There were a few alerts that was FP, as depicted in Table 6.2 and Table 6.3.
- **DoS:** It is difficult to test DoS attacks without causing a genuine denial of service. Pytbull can perform two DoS attacks: one which uses the utility *hping* to produce an Internet Control Message Protocol (ICMP) ping flood to the target machine; the second one attempts to attack a particular application with a DoS attack. The researcher could not simulate the hping DoS attack. Both Snort and Suricata recognised the attack attempt, however, none of these analysis engines identified it as a DoS attack.
- **Evasion techniques:** These tests used basic procedures for evading the analysis engines. The initial two tests used the decoy function inside Nmap to hide the source address of the attacker. The tests failed to get accurate results because the Pytbull installation was on a virtual machine that performed network address translation, so the attempts by Nmap to use distinctive IP addresses brought about the same IP address, defeating the evasion attempt. In the next test, hexadecimal encoding was used in an effort to evade the detection engines. It was detected by neither detection engines. Notwithstanding, in contrast with Suricata, Snort could distinguish the use of Nmap scripting and created a suitable alarm.
- **Fragmented packets**: For these tests, Pytbull executed two types of fragmented packet intrusions; a Ping-of-Death assault where the packet

© Central University of Technology, Free State

segments, when reassembled, were bigger than the permitted size in the protocol specification, and a Nestea attack where the sequence of reassembly is out of order. Neither Snort nor Suricata recognised the Nestea attack. Suricata recognised the Ping-of-Death attack, while Snort created a FP alarm for an outbound Secure Shell check.

- **Pcap replay**: This is the replay of a previously captured malicious payload to test how well the analysis engines can detect other malware. In this test, Suricata performed much better than Snort because of its multi-threaded capabilities.
- **Shell-codes:** A shell-code refers to a written code that starts a command shell (Catania & Garino, 2012; White *et al*., 2013). Pytbull sent 13 different shell-code attacks to the analysis engines. Out of the 13 attempts, Suricata managed to detect 10 and Snort detected only 9.
- **Test rules**: This test evaluated how well the detection engines reacted to a wide range of intrusions. Included were Local File Inclusion (LFI) attacks and different network scans. For these tests, Pytbull used Hypertext Transfer Protocol (HTTP), alongside the Nmap applications. Both Snort and Suricata produced various TP and FP alarms for the test traffic. Pytbull produced a Nmap full-connection scan across all ports and both analysis engines accurately identified it as an Nmap scan. There was an issue when both Snort and Suricata created a FP alarm for a possible network Trojan attack which actually did not take place.
- **Multiple failed logins:** Using a known weak username and password combination, Pytbull tried a number of times to sign into the XaP application under different circumstances to retrieve the username and password of the voter. Suricata created a FP alarm for these attempts as a general login attempt, but not as a failed attempt. Snort, nevertheless, considered the attempt as a bad login alarm.

To summarise these evaluations: when Snort and Suricata were loaded with the similar rule set, sometimes both failed to produce an alarm on known malicious activity. In cases where both failed, it can be said objectively that it is because of the rules and not the detection engines.

© Central University of Technology, Free State

There were a few cases where the Snort and Suricata alarms were inconsistent with one another. It might be contributed to the differences between Snort and Suricata regarding the implementation of the rule language. Another reason might be the different ways in which the detection algorithms are implemented inside each application – this could have influenced the way how the detection engines inspected packets.

6.4.2 Detection rate of the analysis engines

The researcher has hoped to obtain a 100% TP rate, a 100% TN rate, a 0% FN rate and a 0% FP rate. Below are the results obtained for the Snort and Suricata analysis engines regarding accuracy, detection rates, and detection time.

For each analysis engine, the False-Positive Rate (FPR), True-Positive Rate (TPR), False-Negative Rate (FNR) and True-Negative Rate (TNR) were calculated in order to determine the **accuracy** of the analysis engine:

Suricata:

False-Positive Rate (FPR) = FP / (FP+ TN)
= 9 / (9+56)
= 9/ 66
= 0.14 * 100
= 14%

True-Positive Rate (TPR) = FP / (FP+ TN)
= 85 / (85+12)
= 85/ 97
= 0.87 * 100
= 87.6 = 88%

False-Negative Rate (FNR) = FN / (FN+ TP)
= 12 / (12+85)
= 12/ 97
= 0.12 * 100
= 12%

© Central University of Technology, Free State

True-Negative Rate (TNR) = TN / (TN+ FP)

= 56 / (56+9)

= 56/ 65

= 0.86 * 100

= 86%

Accuracy was calculated using the following formula:

Accuracy = (TP+ TN) / (TP+ TN+ FN+ FP)

= (88+ 86) / (88+86+12+14)

= (174)/ (200)

= 0.87 * 100

= 87%

2. Snort

False-Positive Rate (FPR) = FP / (FP+ TN)

= 10 / (10+56)

= 10/ 66

= 0.15 * 100

= 15%

True-Positive Rate (TPR) = FP / (FP+ TN)

= 84 / (84+16)

= 84/ 100

= 0.84 * 100

= 84%

False Negative Rate (FNR) = FN / (FN+ TP)

= 16 / (16+84)

= 16/ 100

= 0.16 * 100

= 16%

© Central University of Technology, Free State

True-Negative Rate (TNR) =TN / (TN+ FP)

= 56 / (56+10)

= 56/ 66

= 0.848 * 100

= 85%

Accuracy was calculated using the following formula:

Accuracy = (TP+ TN) / (TP+ TN+ FN+ FP)

= (84+ 85) / (84+85+16+15)

= (169)/ (200)

= 0.845 * 100

= 85%

6.4.3 Detection time of the analysis engines

For this study, intrusion detection time is defined as the time interval between the start of an intrusion and the time the intrusion is actually detected as an intrusion (Rassam *et al*., 2012).

The two analysis engines are configured in such a way that they are able to detect an intrusion/incident/congestion/queue as soon as possible. The sooner the intrusions are detected, the easier it is to contain their effects and return traffic flow conditions to normalcy. The researcher calculated the detection time using the unit of intrusion per second (ips).

Figure 6.12 below depicts the results the researcher obtained for each analysis engine regarding their detection time.

© Central University of Technology, Free State

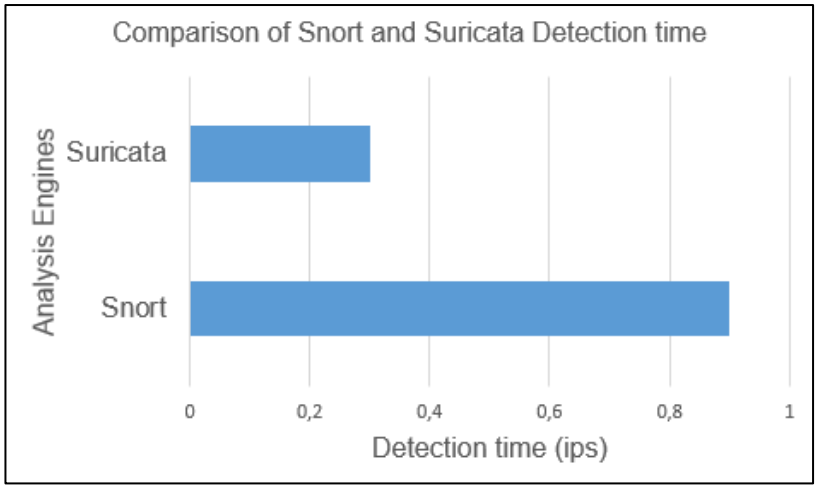


Figure 6-12: Comparison of Snort and Suricata detection times

Suricata provides speed movements and capabilities that are not available in Snort, therefore Suricata has a lower detection time.

In conclusion: to have an effective IDPS, both FPR and FNR must be minimised, together with maximised TPR and TNR. A hybrid detection method, which is the combination of signature- and anomaly-detection methods as stated in Chapter 2, helped the researcher in achieving acceptable alarm rates as far as TP, FN, TN, and FP are concerned.

This study adopted Suricata as a suitable cloud-based analysis engine to protect a m-voting application called XaP. The researcher established that as much as Snort has been the preferred IDPS in the past, during the evaluation simulation that was run for this study, Suricata presented more effective and accurate results close to what the researcher anticipated.

6.5 RECOMMENDATIONS FOR MOBILE VOTERS

IDPSs are just one security technology that can be used to protect a mobile device. Many other solutions and technologies exist that can also be incorporated to provide the desired protection. Some of these solutions include mobile antivirus protection for smartphones that can detect and clear known and unknown malware, provide SMS anti-spamming and more.

The main aim of all attacks is essentially to find the victim's vulnerabilities and to perform an attack using a well-thought-out process and application. It is of great

© Central University of Technology, Free State

importance that certain safety precautions should be taken for mobile tools on which private information and documents are stored. Some of the recommendations for mobile users include:

- Enable device-locking and use a PIN to lock SIM card (Breitinger & Nickel, 2010);
- Never allow obscure devices to connect through Bluetooth. Never leave Bluetooth switched on permanently and never put Bluetooth in “always discoverable” mode (Ghallali *et al*., 2011);
- Use Bluetooth in hidden mode so that even if the device is using Bluetooth, it will be invisible to others. Change the name of the device to prevent others from identifying the mobile phone model. Switch off Bluetooth when it is not transmitting information (Ghallali *et al*., 2011);
- Connect only to networks you trust. Use Wi-Fi only when required and switch off the service when not being used. Mobile users must be cautious when connecting to public networks as they may not be secure (Ghallali *et al*., 2011);
- Use a password while pairing with other devices (Breitinger & Nickel, 2010);
- Never keep sensitive data such as user names or passwords on mobile phones;
- Update the mobile OS frequently by upgrading to the most recent version. Only install applications from trusted sources (Ghallali *et al*., 2011);
- Consider installing security software from a trustworthy provider and update it frequently (Ghallali *et al*., 2011); and
- It is essential to check the features before downloading an application. Some applications may use the mobile phone user’s personal data (Breitinger & Nickel, 2010).

6.6 SUMMARY

In this chapter the two analysis engines named Snort and Suricata were evaluated. The chapter commenced by stating the criteria used to evaluate the analysis engines. The simulation set-up was presented with a list of the components involved and the tools used for the simulation.

Both Snort and Suricata are effective IDPSs, each with its own particular qualities and shortcomings. The researcher tested Snort and Suricata on the same data in order to

© Central University of Technology, Free State

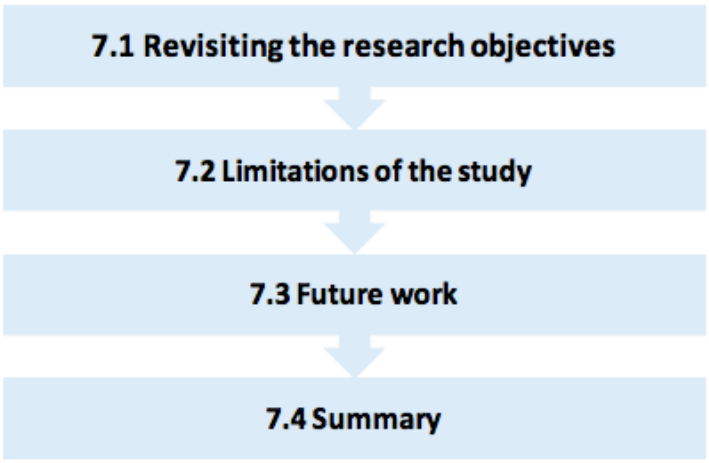
give a knowledgeable recommendation regarding the most suitable IDPS to be used with the XaP application.

Both Snort and Suricata performed well in the tests. Although they did produce FP and FN alerts as well, much of that can be ascribed to shortcomings in the rule set that was used for the tests. In the end, Suricata was adopted as the most suitable cloud analysis engine to protect an m-voting application such as XaP.

The study is concluded in the next chapter where the sub-objectives are assessed and lessons learned are presented. The scope for future studies is also discussed.

© Central University of Technology, Free State

CHAPTER 7: CONCLUSION



© Central University of Technology, Free State

This chapter summarises the study by revisiting the research objectives. It is organised into four sections. In Section 7.1, the research objectives are re-examined. The limitations of the study are presented in Section 7.2 and Section 7.3 discusses future work. Section 7.4 puts the content of this chapter in a nutshell.

7.1 REVISITING RESEARCH OBJECTIVES

The scope of this study was to compare and evaluate two IDSs in order to propose a suitable CIDPS for m-voting in SA. Figure 7.1 below presents the research overview:

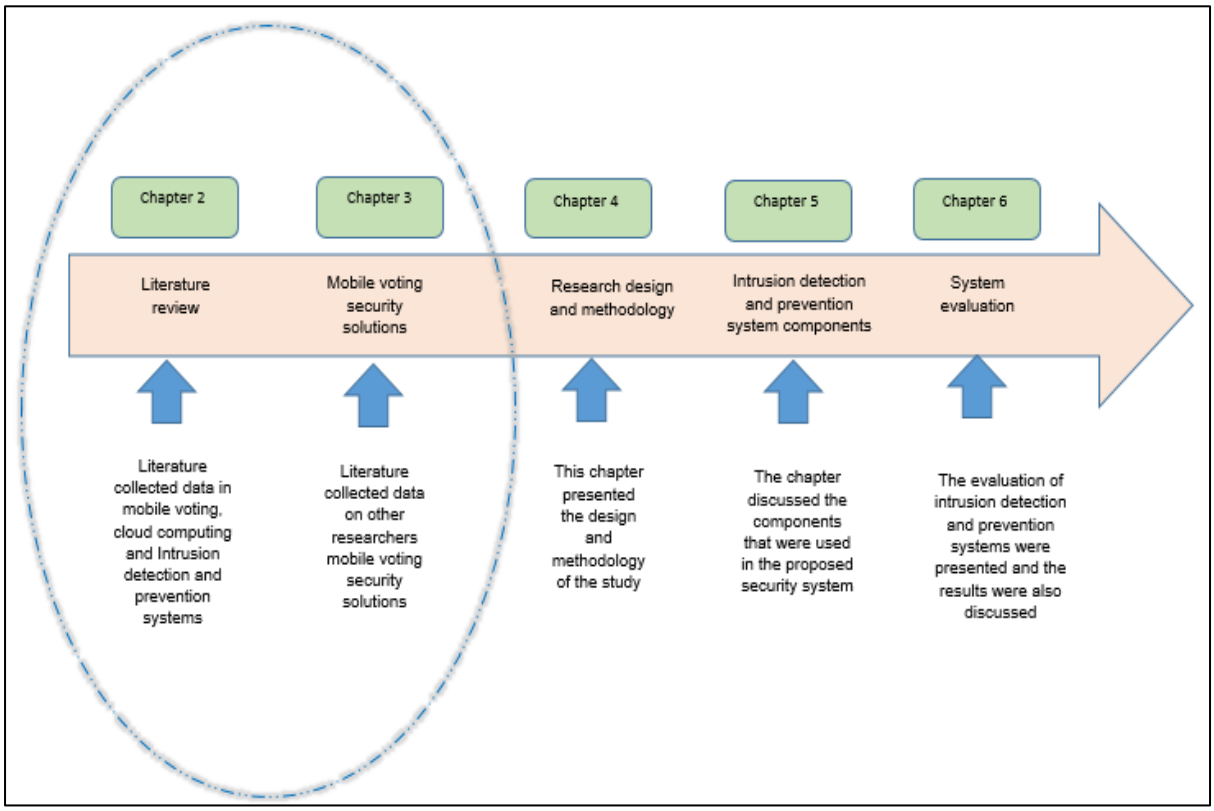


Figure 7-1: Research overview

To fully achieve the main objective of the study, the following secondary objectives were accomplished:

1. **To investigate the essential components of a cloud-based intrusion detection and prevention system**

At the beginning of Chapter 2, the researcher presented the elements that helped in solving the research problem. The elements/components include: cloud computing, m-voting and mobile devices, as well as IDPSs (see Figure 2.1 and Figure 7.2 below).

© Central University of Technology, Free State

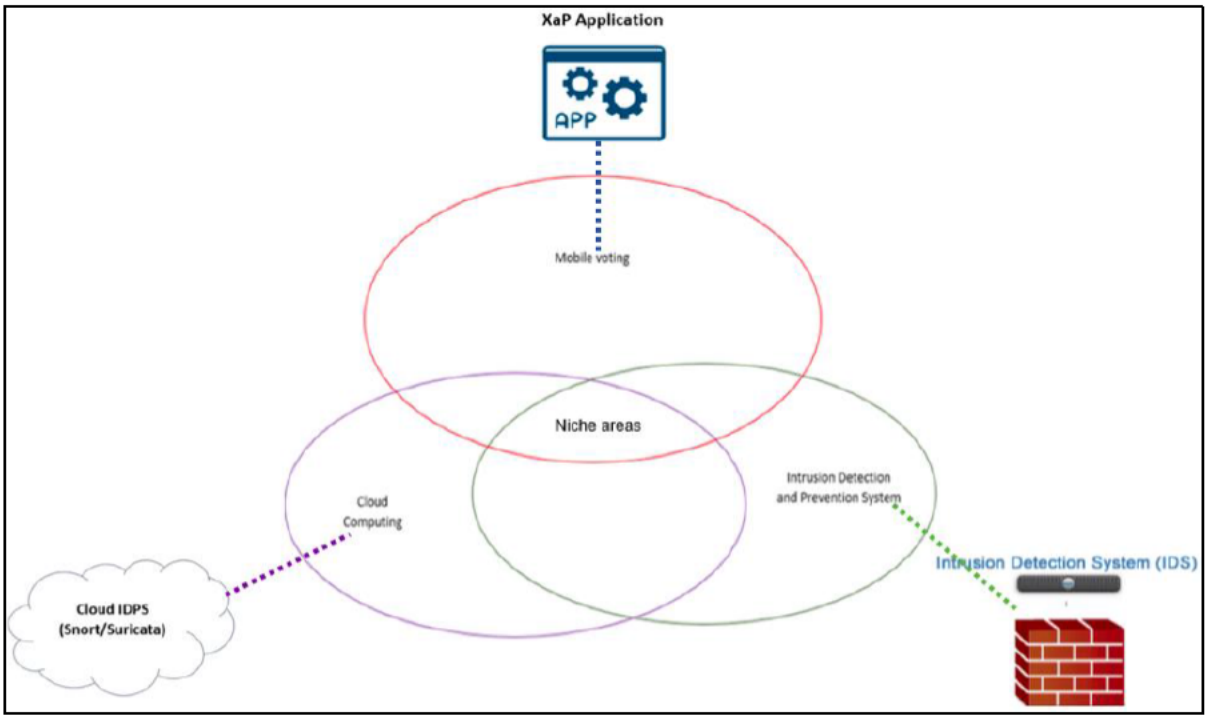


Figure 7-2: Essential components of a CIDPS

The literature review revealed (Chapter 2, Section 2.2 and Section 2.3, and Chapter 3, Section 3.1) that there are three essential components necessary for the implementation of the CIDPS for m-voting. The components include:

- The cloud computing environment
- The analysis engine
- The m-voting application

The following table lists the possibilities regarding these components and identifies the options chosen for this study.

Table 7-1: List of essential components

M-voting applications	Cloud storage	Analysis engine
SMS Voting eVOTZ XaP GSM mobile voting m-Voting system GSM vote system	Amazon EC2 Microsoft Azure Google App engine	Snort Suricata

© Central University of Technology, Free State

XaP was used as the m-voting application (Chapter 3, Section 3.2) and Microsoft Azure was chosen as the cloud storage environment. *Platform as a Service* (PaaS) was used as the public cloud computing layer. Details on cloud computing services were discussed in Chapter 5, Section 5.3. Both Snort and Suricata were evaluated in this study; details are in Chapter 6, Section 6.2.

The study found that there are key stakeholders involved in the proposed security solution. They include the mobile voter, telecommunication network (e.g., 3G/LTE), the XaP application, the client agent, the cloud analysis engine, the attacker, and the cloud.

2. **To compare two cloud-based intrusion detection and prevention systems (analysis engines) and choose the one that best secures m-voting**

Section 5.3.1 theoretically compared and evaluated two analysis engines, namely Snort and Suricata. Table 7.2 gives a summary of the major differences between these two analysis engines.

Table 7-2: Comparison of Snort and Suricata

Parameter	Snort	Suricata
Contextual signatures	No	Yes
Flexible site customisation	Medium	High
False alarm	Medium	Small
Large user community	Yes	Yes
Configuration GUI	Yes	Yes
Installation /deployment	Easy	Easy
OS compatibility	Any OS	Any OS

Chapter 6, Section 6.2, evaluated the two analysis engines. The experiments evaluated the performance of each by measuring their detection time, detection rate, and accuracy. The researcher tested Suricata and Snort on the hand of comparable information to deliver a well-versed recommendation for the XaP application. Both Suricata and Snort are well-known and very effective IDPSs, each with its own

© Central University of Technology, Free State

particular qualities and shortcomings. They both executed the tests to satisfaction, however, both also generated FP and FN alerts.

Suricata was adopted as the most suitable cloud-based analysis engine to protect an m-voting application such as XaP. It uses a hybrid detection method where the signature- and anomaly-detection methods work in parallel.

The analysis engines and the evaluation results were discussed in detail in Section 5.3 and Section 6.4.

3. To link a cloud-based analysis engine and XaP voting system

The XaP m-voting application and the analysis engines (Snort and Suricata) are independent systems that had to be linked together in order for the proposed security solution to function. Figure 7.2 below depicts the relationship between the client agent and the XaP application, as well as the relationship between the client agent and the cloud analysis engine.

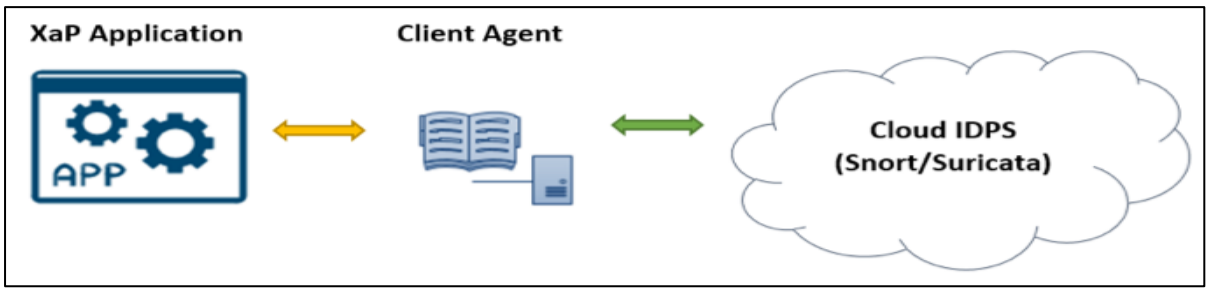


Figure 7-3: Client agent with XaP and the cloud IDPS

The client agent is a back-end software program that runs on the voter's mobile phone. The client agent monitors and collects user-sensor inputs and outputs from the device interface in runtime, and sends it to the cloud analysis engine to perform an intensive malware scan. The client agent listens for notifications from the cloud analysis engine to determine if threats were detected. The client agent is discussed in full detail in Section 5.1.

4. To use evaluation criteria to test the analysis engines

Certain evaluation criteria were developed in order to evaluate the suitability of the proposed security solution. These criteria are depicted in Table 7.3 below:

© Central University of Technology, Free State

Table 7-3: CIDPS evaluation criteria

Criteria	Description	References
Accuracy	The proper detection of attacks and the absence of false alarms.	Abduvaliyev *et al*. (2010); Gupta *et al*. (2013); Volden *et al*. (2011)
Detection rate	The rate at which traffic and audit events are processed.	Abduvaliyev *et al*. (2010); Gupta *et al*. (2013); Volden *et al*. (2011)
Detection time	Time elapsed between intrusion and detection.	Abduvaliyev *et al*. (2010); Gupta *et al*. (2013); Volden *et al*. (2011)

More details on the evaluation criteria are discussed in Section 4.6 and in Section 6.1; the implementation of the criteria (evaluation process) is discussed in Section 6.4.

7.2 LIMITATIONS OF THE STUDY

Some of the limitations identified while conducting the study are:

- There are few developed practical and experimental IDSs in reality (Patel *et al.*, 2013).
- Prevention has only recently been introduced as a feature of IDPSs; as a result, there are not many published research papers and articles on this feature (Patel *et al.*, 2013).
- As far as the literature review for this study is concerned, no published research or study has attempted to develop an IDPS for m-voting before. Nevertheless, all recent work on intrusion detection and prevention running on the cloud was considered for this study since it is expandable.

The main aim of this study was to compare and evaluate two IDPSs (Snort and Suricata) for a specific m-voting application, namely XaP. XaP is the only m-voting application that was available to the researcher at the time of the research.

© Central University of Technology, Free State

7.3 FUTURE WORK

The analysis of both the literature and research tools has provided several important insights. Following are some of the future work that can be done in this area:

- More work needs to be done in order to reduce the FN alerts observed, which were caused by the anomaly-detection method.
- More work still needs to be done on the proposed CIDPS so that it can be used to protect any m-voting system, not only the XaP application.
- The proposed system can also be developed further so that it is able to protect any mobile device using any OS, not only the Android OS.
- Additional research ought to be performed on the incorporation of signature- and anomaly-based intrusion detection methods to address obscure and persistent dangers.

7.4 SUMMARY

In this chapter, the objectives of the study were revisited. The limitations of the study and future work that can be done in this field were also addressed.

Although the benefits of mobile phones as a tool to cast a vote are frequently stated, there are major security concerns regarding this. Hence, this study set out to attempt finding a suitable solution to enhance the security of mobile phones when used to cast a vote.

From what was discussed and shared, one can conclude by saying that it did accomplish its overall intention which was to compare and evaluate two IDPSs (Snort and Suricata) in order to propose a suitable CIDPS as a security solution for m-voting in SA.

This has been a great learning journey. As the great Sipho Mnyakeni, the author and people development practitioner has said:

“We rest when we are done.

Not when we are tired.”

© Central University of Technology, Free State

REFERENCES

Abduvaliyev, A., Lee, S. & Lee, Y.K., 2010. Energy efficient hybrid intrusion detection system for wireless sensor networks. *2010 International Conference On Electronics and Information Engineering (ICEIE*), Kyoto, Japan, 1–3 August, 2010, IEEE, Vol. 2, pp. 2–25.

Achen, C.H. & Bartels, L.M., 2017. *Democracy for realists: Why elections do not produce responsive government*. Princeton University Press.

Adigun, A.A., Fagbola, T.M. & Adegun, A., 2014. SwarmDroid: Swarm Optimized Intrusion Detection System for the Android Mobile Enterprise. *IJCSI International Journal of Computer Science Issues,* May 2014, *11*(3), ISSN (Online): 1694–0784.

Ahmad, A., Shanmugam, B., Idris, N.B. & Samy, G.N., 2013a. Danger Theory Based Hybrid Intrusion Detection Systems for Cloud Computing. *International Journal of Computer and Communication Engineering, 2*(6), p. 650.

Ahmad, M.S., Musa, N.E., Nadarajah, R., Hassan, R. & Othman, N.E., 2013b. Comparison between Android and iOS Operating System in terms of security. In: *Information Technology in Asia (CITA), 8th International Conference on Information Technology. Kota Samarahan, Malaysia*, July 2013, IEEE, pp. 1–4.

Ahmed, M., Mahmood, A.N. & Hu, J., 2016. A survey of network anomaly detection techniques. *Journal of Network and Computer Applications, 60*, pp. 19–31.

Ahson, S.A. & Ilyas, M., 2017. *RFID handbook: applications, technology, security, and privacy*. CRC Press.

Ajiboye, A.R., Adewole, K.S., Jimoh, R.G. & Oladipo, I.D., 2013. Modeling and Evaluation of E-Voting System for a Sustainable Credible Election. *International Journal of Applied Information Systems 5*(3), pp. 8–14.

Akonjom, N.A. & Ogbulezie, J.C., 2014. Bandwidth and Resource Allocation for Full Implementation of e-Election in Nigeria. *International Journal of Engineering Trends and Technology, 10*(2), pp. 58–65.

© Central University of Technology, Free State

Al-Ameen, A. & Talab, S.A., 2013. E-voting systems security issues. *International Journal of Networked Computing and Advanced Information Management, 3*(1), p. 25.

Albin, E. & Rowe, N.C., 2012. A realistic experimental comparison of the Suricata and Snort intrusion-detection systems. In: *Advanced Information Networking and Applications Workshops (WAINA) 26th International Conference on Advanced Information Networking and Applications Workshops*. March 2012, IEEE, pp. 122–127.

Alrajeh, N.A., Khan, S. & Shams, B., 2013. Intrusion detection systems in wireless sensor networks: a review. *International Journal of Distributed Sensor Networks*, *9*(5), p. 167–575.

Alzahrani, A.J., Stakhanova, N., Ali, H.G. & Ghorbani, A., 2014. Characterizing Evaluation Practices of Intrusion Detection Methods for Smartphones. *Journal of Cyber Security and Mobility, 3*(2), pp. 89–132.

Anirudha, A.K., Honale, S., Dhande, P.M. & Chaudhari, P.A., 2013. A Review on Cloud-Based Intrusion Detection System for Android Smartphones. *International Journal of Advanced Research in Engineering & Technology (IJARET)*, *4*(6), pp. 238–245.

Ansper, A., Heiberg, S., Lipmaa, H., Overland, T. & Van Laenen, F., 2009. Security and Trust for the Norwegian E-voting Pilot Project E-valg 2011. *Identity and Privacy in the Internet Age*, September 2009, pp. 207–222.

Archibald, M.M., 2016. Investigator triangulation: A collaborative strategy with potential for mixed methods research. *Journal of Mixed Methods Research*, *10*(3), pp. 228–250.

Ashfaq, R.A.R., Wang, X.Z., Huang, J.Z., Abbas, H. & He, Y.L., 2017. Fuzziness based semi-supervised learning approach for intrusion detection system. *Information Sciences*, *378*, pp. 484–497.

At, C., Burkart, M. & Lee, S., 2011. Security-voting structure and bidder screening. *Journal of Financial Intermediation, 20*(3), pp. 458–476.

© Central University of Technology, Free State

Batyuk, L., Herpich, M., Camtepe, S.A., Raddatz, K., Schmidt, A.D. & Albayrak, S., 2011. Using static analysis for automatic assessment and mitigation of unwanted and malicious activities within Android applications. In: *6th International Conference on Malicious and Unwanted Software (MALWARE)*. October 2011. IEEE, pp. 66–72.

Best, J.W. & Kahn, J.V., 2016. Research in education. Pearson Education: India.

Bhat, A.H., Patra, S. & Jena, D., 2013. Machine learning approach for intrusion detection on cloud virtual machines. *International Journal of Application or Innovation in Engineering & Management (IJAIEM), 2*(6), pp. 56–66.

Blanco, R., Halpin, H., Herzig, D.M., Mika, P., Pound, J., Thompson, H.S. & Tran Duc, T., 2011. Repeatable and reliable search system evaluation using crowdsourcing. In: *Proceedings of the 34th international ACM SIGIR conference on research and development in Information Retrieval*, Beijing, China, July 2011, USA: ACM, pp. 923–932.

Boell, S.K. & Cecez-Kecmanovic, D., 2014. A hermeneutic approach for conducting literature reviews and literature searches. *CAIS*, *34*, p.12.

Breitinger, F. & Nickel, C., 2010. User Survey on Phone Security and Usage. *BIOSIG,* September 2010, pp. 139–144.

Buennemeyer, T.K., Nelson, T.M., Clagett, L.M., Dunning, J.P., Marchany, R.C. & Tront, J.G., 2008. Mobile device profiling and intrusion detection using smart batteries. In: *Proceedings of the 41st Annual Hawaii International Conference on System Sciences*. January 2008, IEEE, pp. 296–296.

Burguera, I., Zurutuza, U. & Nadjm-Tehrani, S., 2011. Crowdroid: behavior-based malware detection system for android. In: *Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*, Chicago, Illinois, October 2011:ACM, pp. 15–26.

Butler, M., 2011. Android: Changing the mobile landscape. *IEEE Pervasive Computing, 10*(1), pp. 4–7.

© Central University of Technology, Free State

Button, D., Harrington, A. & Belan, I., 2014. E-learning & information communication technology (ICT) in nursing education: A review of the literature. *Nurse Education Today, 34*(10), pp. 1311–1323.

Cameron, R., 2009. A sequential mixed model research design: Design, analytical and display issues. *International Journal of Multiple Research Approaches, 3*(2), pp. 140–152.

Campbell, B.A., Tossell, C.C., Byrne, M.D. & Kortum, P., 2011. Voting on a smartphone: Evaluating the usability of an optimized voting system for handheld mobile devices. In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 2011,* CA: Sage Publications, *55*(1), pp. 1100–1104.

Carroll, M., Van Der Merwe, A. & Kotze, P., 2011. Secure cloud computing: Benefits, risks and controls. In: *Information Security South Africa (ISSA),* August 2011, IEEE, pp. 1–9.

Catania, C.A. & Garino, C.G., 2012. Automatic network intrusion detection: Current techniques and open issues. *Computers & Electrical Engineering*, *38*(5), pp. 1062–1072.

Cetinkaya, O. & Doganaksoy, A., 2007. Pseudo-voter identity (pvid) scheme for e-voting protocols. In: *The Second International Conference on Availability, Reliability and Security.* ARES, 2007*. IEEE,* pp. 1190–1196.

Chang, V. & Ramachandran, M., 2016. Towards achieving data security with the cloud computing adoption framework. *IEEE Transactions on Services Computing, 9*(1), pp. 138–151.

Chang, V., Kuo, Y.H. & Ramachandran, M., 2016. Cloud computing adoption framework: A security framework for business clouds. *Future Generation Computer Systems, 57*, 2016, pp. 24–41.

Chin, E., Felt, A.P., Sekar, V. & Wagner, D., 2012. Measuring user confidence in smartphone security and privacy. In: *Proceedings of the eighth symposium on usable privacy and security*, Washington, D.C, July 2012, ACM, p. 1.

© Central University of Technology, Free State

Christ, T.W., 2009. Designing, teaching, and evaluating two complementary mixed methods research courses. *Journal of Mixed Methods Research*, *3*(4), pp. 292–325.

Cox, V., 2017. Exploratory Data Analysis: Translating Statistics to Make Decisions. Available at: https://en.wikipedia.org/wiki/Exploratory_data_analysis.

Creswell, J.W., 2007. *Research Design: Qualitative, Quantitative and Mixed Method Approaches.* 3rd edition. Los Angeles, London: Sage Publications.

Creswell, J.W., 2013. *Research Design: Qualitative, Quantitative and Mixed Methods Approaches.* 4th edition. Los Angeles, London: Sage Publications.

Creswell, J.W. & Clark, P.V.L., 2011. *Designing and Conducting Mixed Methods Research.* 2nd edition. London: Sage Publications.

Creswell, J.W & Klassen, A.C., 2011. Best practices for mixed methods research in the health sciences. *National Institutes of Health*, *29*, 2011, pp. 4–10.

Creswell, J.W. & Poth, C.N., 2017. *Qualitative Inquiry and Research Design: Choosing among Five Approaches*. London: Sage Publications.

Day, D. & Burns, B., 2011. A performance analysis of Snort and Suricata network intrusion detection and prevention engines. In: *Fifth International Conference on Digital Society*. Gosier, Guadeloupe, February 2011. pp. 187–192.

Delaune, S., Kremer, S. & Ryan, M.D., 2006. Verifying Properties of Electronic Voting Protocols. In: *Proceedings of the IAVoSS Workshop On Trustworthy Elections*, 2006, pp.45–52.

Dinh, H.T., Lee, C., Niyato, D. & Wang, P., 2013. A survey of mobile cloud computing: architecture, applications, and approaches. *Wireless communications and mobile computing*, *13*(18), pp. 1587–1611.

Eilu, E. & Baguma, R., 2013. Designing reality fit m-voting. In: *Proceedings of the 7th International Conference on Theory and Practice of Electronic Governance*, Seoul, Korea, ACM, New York, pp. 326-329.

© Central University of Technology, Free State

Eilu, E., Baguma, R. & Pettersson, J.S., 2014. M-voting in developing countries: Findings from Uganda. *Commonwealth Governance Handbook*, *15*, 2014, pp. 25–28.

Ekong, U.O & Ekong, V.E., 2010. M-voting: a panacea for enhanced e-participation. *Asian Journal of Information Technology*, *9*(2), pp.111–116.

Elhag, S., Fernández, A., Bawakid, A., Alshomrani, S. & Herrera, F., 2015. On the combination of genetic fuzzy systems and pairwise learning for improving detection rates on Intrusion Detection Systems. *Expert Systems with Applications*, *42*(1), pp.193–202.

Feng, Y., Ng, S.L & Schwiderski-Grosche, S., 2006. An electronic voting system using GSM mobile technology. *Department of Mathematics Royal Holloway, University of London Egham, England Technical Report RHUL–MA–2006, http://www. rhul. ac. uk/mathematics/techreports*, *26*.

Fong, S. & Yan, Z., 2008. Comparative study on m-commerce applications in various scenarios. In: *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology,* Sydney, Australia, December 2008, IEEE, *Vol 1*, pp. 943–946.

Gai, K., Qiu, M., Zhao, H., Tao, L. & Zong, Z., 2016. Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing. *Journal of Network and Computer Applications*, *59*, pp.46–54.

Gandhewar, N. & Sheikh, R., 2010. Google Android: An emerging software platform for mobile devices. *International Journal on Computer Science and Engineering*, *1*(1), pp. 12–17.

Gentles, D., & Sankaranarayanan, S., 2012. Application of biometrics in mobile voting. *I.J. Computer Network and Information Security,* p. 57–68.

Ghallali, M., El Ouadghiri, D., Essaaidi, M. & Boulmalf, M., 2011. Mobile phone security: the spread of malware via MMS and Bluetooth, prevention methods. In: *Proceedings of the 9th International Conference on Advances in Mobile Computing and Multimedia,* Ho Chi Minh, Vietnam, December 2006, *ACM,* pp. 256–259.

© Central University of Technology, Free State

Ghate, B., Talewar, S., Taware, S. & Katti, J.V., 2017. E-Voting system based on mobile using NIC and SIM. *International Journal of Computer Applications*, *165*(8).

Gibson, R., Römmele, A. & Ward, S., 2004. Electronic democracy: mobilisation, organisation and participation via new ICTs. In: Rachel K. Gibson, Andrea Rommele & Stephen J. Ward. eds., London: Routledge.

Gioia, D.A., Corley, K.G. & Hamilton, A.L., 2013. Seeking qualitative rigor in inductive research: Notes on the Gioia methodology. *Organizational Research Methods*, *16*(1), pp. 15–31.

Goyal, D., Hemrajani, N., Sharma, G., Sharma, R.S. & Goyal, R., 2013. Secure Voting System in a Mobile Network. *International Journal of Computer and Information Technology, 02*(02), pp. 286–290.

Gul, I. & Hussain, M., 2011. Distributed Cloud Intrusion Detection Model. *International Journal of Advanced Science and Technology*, *34*, 2011, pp. 71–82.

Gupta, S., Kumar, P. & Abraham, A., 2013. A profile-based network intrusion detection and prevention system for securing cloud environment. *International Journal of Distributed Sensor Networks*, viewed 22 Februery 2016, from https://doaj.org/article/2963d3d5aedd402d97fb694dfdbdba5d/

Håkansson, A., 2013. Portal of research methods and methodologies for research projects and degree projects. In: *Proceedings of the International Conference on Frontiers in Education: Computer Science and Computer Engineering FECS'13*, [ed] Hamid R, Arabnia, A, Victor A. Clincy Leonidas Deligiannidis George Jandieri, Las Vegas, USA: CSREA Press2013, p. 67–73.

Harris, M. & Patten, K., 2014. Mobile device security considerations for small- and medium-sized enterprise business mobility. *Information Management & Computer Security*, *22*(1), pp. 97–114.

Hashem, I.A.T., Yaqoob, I., Anuar, N.B., Mokhtar, A. & Gani, A., 2015. The rise of “big data” on cloud computing: Review and open research issues. *Information Systems*, *47*, pp.98–115.

© Central University of Technology, Free State

He, D., Chan, S. & Guizani, M., 2015. Mobile application security: malware threats and defenses. *IEEE Wireless Communications*, *22*(1), pp. 138–144.

Hermawan, H. & Sarno, R., 2012. Developing distributed system with service resource oriented architecture. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, *10*(2), pp. 389–399.

Hirsh, Å., 2015. IDPs at Work. *Scandinavian Journal of Educational Research*, *59*(1), pp. 77–94.

Houmansadr, A., Zonouz, S.A. & Berthier, R., 2011. A cloud-based intrusion detection and response system for mobile phones. In: *Dependable Systems and Networks Workshops (DSN-W), 2011 IEEE/IFIP 41st International Conference on Dependable Systems and Networks,* Sheraton Hong Kong, China: IEEE, June 2011, pp. 31–32.

Hubballi, N. & Suryanarayanan, V., 2014. False alarm minimization techniques in signature-based intrusion detection systems: A survey. *Computer Communications*, *49*, pp.1–17.

Hussain, A. & Mkpojiogu, E.O., 2015. The effect of responsive web design on the user experience with laptop and Smartphone devices. *Journal of Technology*, *77*(4), pp. 41–47.

Ivankova, N.V., Creswell, J.W. & Stick, S.L., 2006. Using mixed-methods sequential explanatory design: From theory to practice. *Field methods*, *18*(1), pp. 3–20.

Jacobs, B. & Pieters, W., 2009. Electronic Voting in the Netherlands: from early Adoption to early Abolishment. In: Alessadro Aldini, Gilles Barthe, Roberto Gorrieri. *eds.*, *Foundations of security analysis and design.* pp. 121–144. Hong Kong, China: Springer Berlin Heidelberg.

Jiang, H., Zhang, G., Xie, G., Salamatian, K. & Mathy, L., 2013. Scalable high-performance parallel design for network intrusion detection systems on many-core processors. In: *Proceedings of the Ninth ACM/IEEE Symposium on Architectures for Networking and Communications ystems,* IEEE Press, October 2013, pp. 137–146. IEEE Press.

© Central University of Technology, Free State

Kalaichelvi, V. & Chandrasekaran, R.M., 2012. Design and Analysis of Secured Electronic Voting Protocol. *Journal of Engineering and Applied Sciences*, *7*(2), pp.143–147.

Karim, C., 2008. Biocep, towards a Federative, Collaborative, User-Centric, Grid-Enabled and Cloud- Ready Computational Open Platform. 2008 *Fourth IEEE International Conference on eScience*, pp.321–322,

Khamphakdee, N., Benjamas, N. & Saiyod, S., 2015. Improving Intrusion Detection System Based on Snort Rules for Network Probe Attacks Detection with Association Rules Technique of Data Mining. *Journal of ICT Research and Applications*, *8*(3), pp.234–250.

Khune, R.S. & Thangakumar, J., 2012. A cloud-based intrusion detection system for Android smartphones. *International Conference on Radar, Communication and Computing (ICRCC), IEEE, 2012,* pp. 180–184.

Kiayias, A., Korman, M. & Walluck, D., 2006. An internet voting system supporting user privacy. *Proceedings of the 22nd Annual Computer Security Applications Conference, IEEE, 2006,* pp. 165–174.

Kim, G., Lee, S. & Kim, S., 2014. A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. *Expert Systems with Applications*, *41*(4), pp. 1690–1700.

Kim, H., Smith, J. & Shin, K.G., 2012. Detecting energy-greedy anomalies and mobile malware variants. In: *Proceedings of the 6th international conference on Mobile systems, applications, and services,* ACM, June 2012, pp. 239–252.

Kitsing, M., 2014. Internet Voting in Estonia. *Proceedings of the 2014 Conference on Electronic Governance and Open Society: Challenges in Eurasia*, 2014, ACM,pp. 137–144.

Klein, E. 2010. eVOTZ: mobilizing voters. *InsideGNSS* , pp. 30-34.

Kolko, J., 2010. Abductive thinking and sense-making: The drivers of design synthesis. *Design Issues*, *26*(1), pp. 15–28.

© Central University of Technology, Free State

Kolpyakwar, A.A. & Bhute, Y., 2015. A Survey on Intrusion Detection System for Android Smartphone in a Cloud Environment. *International Journal of Advance Research in Computer Science and Management Studies*, *3*(6), pp. 12–17.

Kowalski, S. & Goldstein, M., 2006. Consumers' awareness of, attitudes towards and adoption of mobile phone security. In: *20th International Symposium on Human Factors in Telecommunication* 2008, pp. 20–23.

Kushwaha, A. & Kushwaha, V., 2011. Location-based services using android mobile operating system. *International Journal of Advances in Engineering & Technology*, *1*(1), pp.14–20.

Li, J., Ma, R. & Guan, H., 2017. Tees: An efficient search scheme over encrypted data on mobile cloud. *IEEE Transactions on Cloud Computing*, *5*(1), pp.126–139.

Li, Y., Dai, W., Ming, Z. & Qiu, M., 2016. Privacy protection for preventing data over collection in smart city. *IEEE Transactions on Computers*, *65*(5), pp.1339–1350.

Lin, W.C., Ke, S.W. & Tsai, C.F., 2015. CANN: An intrusion detection system based on combining cluster centers and nearest neighbors. *Knowledge-based systems*, *78*, pp. 13–21.

Lo, C.C., Huang, C.C. & Ku, J., 2010. A cooperative intrusion detection system framework for cloud computing networks. In: *Parallel Processing Workshops (ICPPW), 2010 39th International Conference on Parallel Processing Workshop, IEEE, 2015,* pp. 280–284.

Luo, J.Z., Jin, J.H., Song, A.B. & Dong, F., 2011. Cloud computing: architecture and key technologies. *Journal of China Institute of Communications*, *32*(7), pp. 3–21.

Magomelo, M., Mavhemwa, M.P. & Ndumiyana, D., 2013. Effectiveness of mobile voting as a supplementary method to ballot casting: Case of Bindura University of Science Education. *International Journal of Emerging Trends and Technology in Computer Science*, *2*(6), pp. 161–172.

Mahmood, R., Esfahani, N., Kacem, T., Mirzaei, N., Malek, S. & Stavrou, A., 2012. A white box approach for automated security testing of Android applications on the

© Central University of Technology, Free State

cloud. In: *7th International Workshop on Automation of Software Test (AST),* IEEE, 2012, pp. 22–28.

Malisa, L., Kostiainen, K & Capkun, S., 2017. Detecting mobile application spoofing attacks by leveraging user visual similarity perception. In: *Proceedings of the Seventh ACM Conference on Data and Application Security and Privacy*, March, pp. 289–300.

Malkawi, M., Khasawneh, M., Al-Jarrah, O. & Barakat, L., 2009. Modelling and Simulation of a Robust E-voting System. *Communications of the IBIMA*, *8*(26), pp.198–206.

Manzalini, A. & Crespi, N., 2015. SDN and NFV for Network Cloud Computing: A Universal Operating System for SD Infrastructures. In: *2015 IEEE Fourth Symposium on Network Cloud Computing and Applications (NCCA),* IEEE, June 2015, pp. 1–6.

Marforio, C., Jayaram Masti, R., Soriente, C., Kostiainen, K & Čapkun, S., 2016. Evaluation of personalized security indicators as an anti-phishing mechanism for smartphone applications. In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems,* ACM, May 2016, pp. 540-551.

Mehmood, Y., Habiba, U., Shibli, M.A. & Masood, R., 2013. Intrusion detection system in cloud computing: Challenges and opportunities. In: *2nd National Conference on Information Assurance (NCIA), IEEE*, December 2013, pp. 59–66.

Mitchell, R. & Chen, I.R., 2014. A survey of intrusion detection techniques for cyber-physical systems. *ACM Computing Surveys (CSUR)*, *46*(4), p. 55.

Modi, C., Patel, D., Borisaniya, B., Patel, H., Patel, A. & Rajarajan, M., 2013. A survey of intrusion detection techniques in the cloud. *Journal of Network and Computer Applications*, *36*(1), pp. 42–57.

Modi, C.N. & Acha, K., 2017. Virtualization layer security challenges and intrusion detection/prevention systems in cloud computing: a comprehensive review. *The Journal of Supercomputing*, *73*(3), pp. 1192–1234.

© Central University of Technology, Free State

Moon, D., Pan, S.B. & Kim, I., 2016. Host-based intrusion detection system for secure human-centric computing. *The Journal of Supercomputing*, *72*(7), pp. 2520–2536.

More, D., Shaikh, M., Awaskar, M., Ghongde, S., Wattamwar, S. & Tadpelliwar, D., 2015. Mobile voting system. *International Journal of Emerging Technology and Advanced Engineering*, *5*(1).

Mpekoa, N., 2014. *A Model of Mobile Phone Voting System for South Africa*. Masters dissertation, Tshwane University of Technology.

Mpekoa, N. & Van Greunen, D., 2016. Factors affecting successful implementation of m-voting in South Africa. In: *2016 International Conference on Information Society (i-Society), IEEE,* October 2016, pp. 57–62.

Mpekoa, N. & Van Greunen, D., 2017. E-voting experiences: A case of Namibia and Estonia. In: *IST-Africa Conference (IST-Africa), IEEE,* May 2017, pp. 1–8.

Neyeloff, J.L., Fuchs, S.C. & Moreira, L.B., 2012. Meta-analyses and Forest plots using a Microsoft excel spreadsheet: step-by-step guide focusing on descriptive data analysis. *BMC Research Notes*, *5*(1), p. 52.

Okediran, O.O., Olabiyisi, S.O., Omidiora, E.O. & Ganiyu, R.A., 2011. A Survey of Remote Internet Voting Vulnerabilities. *World of Computer Science and Information Technology Journal (WCSIT), 1*(7), pp. 297–301.

Olusola, O.O., Olusayo, O.E., Olatunde, O.S. & Adesina, G.R., 2012. A review of the underlying concepts of electronic voting. *Information and Knowledge Management*, *2*(1), pp. 8–20.

Östlund, U., Kidd, L., Wengström, Y. & Rowa-Dewar, N., 2011. Combining qualitative and quantitative research within mixed method research designs: a methodological review. *International Journal of Nursing Studies*, *48*(3), pp. 369–383.

Paikaray, J., Mohapatra, S. & Rath, S.K., 2015. A New Approach toward Locating ERP Components on Cloud Computing Architecture. *Journal of Computer Sciences and Applications*, *3*(6), pp. 143–146.

© Central University of Technology, Free State

Papamartzivanos, D., Damopoulos, D. & Kambourakis, G., 2014. A cloud-based architecture to crowdsource mobile app privacy leaks. In: *Proceedings of the 18th Panhellenic Conference on Informatics, ACM,* 2014, pp. 1–6).

Patel, A., Taghavi, M., Bakhtiyari, K. & JúNior, J.C., 2013. An intrusion detection and prevention system in cloud computing: A systematic review. *Journal of Network and Computer Applications*, *36*(1), pp. 25–41.

Persson, M., Sundell, A. & Öhrvall, R., 2014. Does Election Day weather affect voter turnout? Evidence from Swedish elections. *Electoral Studies*, *33*, pp. 335–342.

Pickering, C. & Byrne, J., 2014. The benefits of publishing systematic quantitative literature reviews for PhD candidates and other early-career researchers. *Higher Education Research & Development*, *33*(3), pp. 534–548.

Portokalidis, G., Homburg, P., Anagnostakis, K. & Bos, H., 2010. Paranoid Android: versatile protection for smartphones. In: *Proceedings of the 26th Annual Computer Security Applications Conference, ACM,* 2010, pp. 347–356.

Rahimi, M.R., Ren, J., Liu, C.H., Vasilakos, A.V. & Venkatasubramanian, N., 2014. Mobile cloud computing: A survey, state of art and future directions. *Mobile Networks and Applications*, *19*(2), pp. 133–143.

Raja, I., 2013. A Cloud-based Intrusion Detection Forensic Analysis on Smart Phones. *Journal of Global Research in Computer Science*, *4*(4), pp. 204–207.

Ranjini, K., Kanthimathi, A. & Yasmine, Y., 2011. Design of adaptive road traffic control system through unified modelling language. *International Journal of Computer Applications*, *14*(7), pp. 36–41.

Rashida, S.Y., 2013. Hybrid architecture for distributed intrusion detection system in wireless network. *International Journal of Network Security & Its Applications*, *5*(3), p. 45.

Rassam, M.A., Maarof, M.A. & Zainal, A., 2012. A survey of intrusion detection schemes in wireless sensor networks. *American Journal of Applied Sciences*, *9*(10), p. 1636.

© Central University of Technology, Free State

Ravale, U., Marathe, N. & Padiya, P., 2015. Feature selection based hybrid anomaly intrusion detection system using K-means and RBF kernel function. *Procedia Computer Science*, *45*, pp. 428–435.

Reichertz, J., 2007. *Abduction: The logic of discovery of grounded theory.* London: Sage.

Roesch, M., 1999. Snort: Lightweight intrusion detection for networks. *Lisa*, *99*(1), pp. 229–238.

Rubner, G., 2012. mbclick – An Electronic Voting System that Returns Individual Feedback. In: *Seventh International Conference on Wireless, Mobile and Ubiquitous Technology in Education (WMUTE)*, IEEE, 2012, pp. 221–222.

Shabtai, A. & Elovici, Y., 2010. Applying behavioral detection on android-based devices. *Mobile Wireless Middleware, Operating Systems, and Applications*, 2010, pp.235–249.

Shabtai, A., Tenenboim-Chekina, L., Mimran, D., Rokach, L., Shapira, B. & Elovici, Y., 2014. Mobile malware detection through analysis of deviations in application network behavior. *Computers & Security*, *43*, 2014, pp. 1–18.

Shahbazi, M., Mcafee, Inc., 2013. *Mobile data security system and methods*. U.S. Patent 8,495,700.

Singh, J. & Thapar, E.V., 2012. Intrusion Detection System in Wireless Sensor Network. *International Journal of Computer Science and Communication Engineering*, *1*(2), pp. 76–80.

Soldani, D. & Manzalini, A., 2015. Horizon 2020 and beyond: on the 5G operating system for a true digital society. *IEEE Vehicular Technology Magazine*, *10*(1), pp. 32–42.

Sommers, J., Yegneswaran, V. & Barford, P. 2004. A framework for malicious workload generation. In: *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, ACM, 2004, pp. 82–87.

© Central University of Technology, Free State

Suo, H., Liu, Z., Wan, J. & Zhou, K., 2013. July. Security and privacy in mobile cloud computing. In: *9th International Wireless Communications and Mobile Computing Conference (IWCMC),* , IEEE, July 2013, pp. 655–659.

Thakur, S., Olugbara, O.O., Millham, R., Wesso, H.W. & Sharif, M., 2014. Transforming voting paradigm – the shift from inline through online to mobile voting. In: *2014 IEEE 6th International Conference on Adaptive Science & Technology (ICAST)*, IEEE, 2014, pp. 1–7.

Thomas, J.R., Silverman, S. & Nelson, J., 2005. *Research methods in physical activity, 7E*. United States: Human Kinetics

Titova, S. & Talmo, T., 2014. Mobile Voting Systems for Creating Collaboration Environments and Getting Immediate Feedback. *International Journal of Mobile and Blended Learning*, *6*(3), pp. 18–34.

Torrance, H., 2012. Triangulation, respondent validation, and democratic participation in mixed methods research. *Journal of Mixed Methods Research*, *6*(2), pp. 111–123.

Wang, B., Zheng, Y., Lou, W. & Hou, Y.T., 2015. DoS attack protection in the era of cloud computing and software-defined networking. *Computer Networks*, *81*, 2015, pp. 308–319.

Wang, G., Hao, J. & Huang, L., 2010. A new approach to intrusion detection using artificial neural networks and fuzzy clustering. *Expert Systems with Applications*, *37*(9), pp. 6225–6232.

Warrier, G.B. 2010. *Cell Phone based voting system.* Kerala.

White, J.S., Fitzsimmons, T. & Matthews, J.N., 2013. Quantitative analysis of intrusion detection systems: Snort and Suricata. *SPIE Defense, Security and Sensing*, United States, May 2013, 8757, pp. 12.

Wurm, S., Tomasik, M.J. & Tesch-Römer, C., 2010. On the importance of a positive view on ageing for physical exercise among middle-aged and older adults: Cross-sectional and longitudinal findings. *Psychology and Health*, *25*(1), pp. 25–42.

© Central University of Technology, Free State

Xing, T., Huang, D., Xu, L., Chung, C.J. & Khatkar, P., 2013. Snortflow: An openflow-based intrusion prevention system in cloud environment. In: *2013 Second Research and Educational Experiment Workshop (GREE),* Salt Lake City, UT, USA, IEEE, 20–22 March 2013, pp. 89–92.

Zhang, H., Jiang, H., Li, B., Liu, F., Vasilakos, A.V. & Liu, J., 2016. A framework for truthful online auctions in cloud computing with heterogeneous user demands. *IEEE Transactions on Computers*, *65*(3), pp. 805–818.

Zissis, D. & Lekkas, D., 2011. Securing e-Government and e-Voting with an open cloud computing architecture. *Government Information Quarterly*, *28*(2), pp. 239–251.

Zonouz, S., Houmansadr, A., Berthier, R., Borisov, N. & Sanders, W., 2013. Secloud: A cloud-based comprehensive and lightweight security solution for smartphones. *Computers & Security*, *37*, 2013, pp. 215–227.

© Central University of Technology, Free State