IST-Africa

IST-Africa 2017 Conference Proceedings
Paul Cunningham and Miriam Cunningham (Eds)
IIMC International Information Management Corporation, 2017
ISBN: 978-1-905824-56-4

Securing M-voting Using Cloud Intrusion Detection and Prevention System: A New Dawn

Dina MOLOJA[1], Noluntu MPEKOA[2]
[1]*Central University of Technology, Mothusi road, 9460, South Africa*
Tel: 057 910 3639, Fax: 086 619 5810, Email: mmoloja@cut.ac.za
[2] *Central University of Technology, Bloemfontein, 9301, South Africa*
Tel:051 507 3587, Email: nmpekoa@cut.ac.za

Abstract: Democracy has been transformed by the introduction of ICT, which is known as e-voting. E-voting is the use of computerised equipment to cast votes. M-voting is a subset of e-voting and is the use of mobile phones to cast a vote outside the restricted electoral boundaries. M-voting has a feature that is different from other e-voting solutions: the mobile-phone. Mobile-phones are pervasive; they offer connection everywhere at any time. However, using a fast growing device such as mobile-phone as a tool to cast a vote can raise questions when coming to its security. This paper designed and developed a security solution termed a cloud intrusion detection and prevention system which endeavours to secure the voters' mobile phone while casting their vote. The security system was developed using android version 6.0 for android phones and MySQL. Simulations were used to evaluate the system and results indicate that the proposed system is efficient, reliable and secure.

Keywords: Intrusion detection and prevention, M-voting, cloud computing, data security.

1. Introduction

New advances in Information and Communication Technology (ICT) have changed nearly every facet of our lives [12, 13]. Modern societies completely rely on ICT for business, work and leisure time activities, including in the area of voting [4, 6]. The development and widespread use of ICT is changing the way communities view voting processes and the way they vote [5]. Electronic voting (e-voting), as a new paradigm for voting is defined as the use of computers or computerized equipment to cast votes in elections [6, 11]. Mobile voting (M-voting) is the subset of e-voting and is the use of Global System for Mobile Communications that allows the voter to use their mobile phone to cast their vote regardless of their location [4, 5, 11, 13].

M-voting has a special feature that makes it different from other e-voting solutions: the mobile phone [4]. Mobile phones are pervasive and ambitious; they offer connection everywhere at any time. The mobile phone penetration is increasing every day, as the price of acquiring these devices becomes more and more affordable. It is hence these reason that m-voting has attracted a lot of attention from researchers and innovators [5, 11, 13]. Nevertheless, this paper is concerned with the increasing security threats against mobile phones, more especially during the voting process because security is a vital aspect in any voting system.

According to Dhaya and Poongodi [7], mobile phones face a wide range of security challenges including malicious threats and intrusions because they are gradually being used to store sensitive personal information. In addition, they can now be used as a tool to cast a vote during election [12, 13]. An intrusion against M-voting system may aim at violating either the secrecy or the integrity of the vote [4, 5, 12]. The past security solutions for mobile phones encountered several restrictions in practice. Many approaches are based on running a lightweight Intrusion Detection and Prevention System (IDPS) on the mobile phone. But such security solutions failed to provide effective protection as they are constrained by the limited memory, storage, computational resources, and battery power of the devices [19, 21]. Hence, our system incorporates cloud computing to solve the limited memory problems of mobile phones. Fundamentally, there is a need for a secure, effective and efficient security solution in order to offer successful implementation of M-voting in SA. In this paper, a Cloud Intrusion Detection and Prevention System (CIDPS) for M-voting in SA is developed. The system aimed at identifying any entity that attempts to compromise the confidentiality, integrity or availability of mobile voting devices for elections where all the electioneering stakeholders are expected to gain from the security of M-voting.

The paper is organized as follows: Section 2 discusses the developed security solution for M-voting. In section 3 the technologies and methods used to develop the system are discussed. Section 4 discusses how the system was developed. The results of the system are presented in section 5. Section 6 provides a summary of the researcher's achievements, further works and recommendations.

2. System Architecture

The M-voting security solution was implemented by developing a prototype system in android version 6.0 for android phones for backend and the database was implemented using MySQL. Android is the latest operating system used in most mobile phones and dominate most in the market shared, it is for these reasons we developed our system for android mobile phones.

The system was constructed from two parts: Sensor (client agent) and analyser (cloud analysis engine). Sensors collected data such as network traffic, log files and system trace files. Once data was collected was then forwarded to the cloud analysis engine where intensive intrusion scan happened. Cloud analysis engine are responsible for determining if there was intrusion among the data sent by the client agent. After an intrusion was detected the analysis engine's output was either an alarm or action.

The cloud analysis engine made use of a malware library to scan for intrusions. The client agent listened for notifications from the cloud analysis engine and warned the client if a threat is detected and also took the necessary precautions. The cloud analysis engine utilized a hybrid detection method where the signature method and anomaly based method worked in parallel. Moreover, if it happened that an attack was detected by the cloud analysis engine then it was compared with the known threats stored in signature database (SDB) and alerted the client agent in the case of matching according to signature detection method. On the other hand, if the match was not found to any of the signatures in the SDB and the behaviour is not a normal one, then the proposed model considered it as abnormal behaviour according to Anomaly detection method and also alerted the client agent and saved the signature of that attack as a new threat within the SDB.

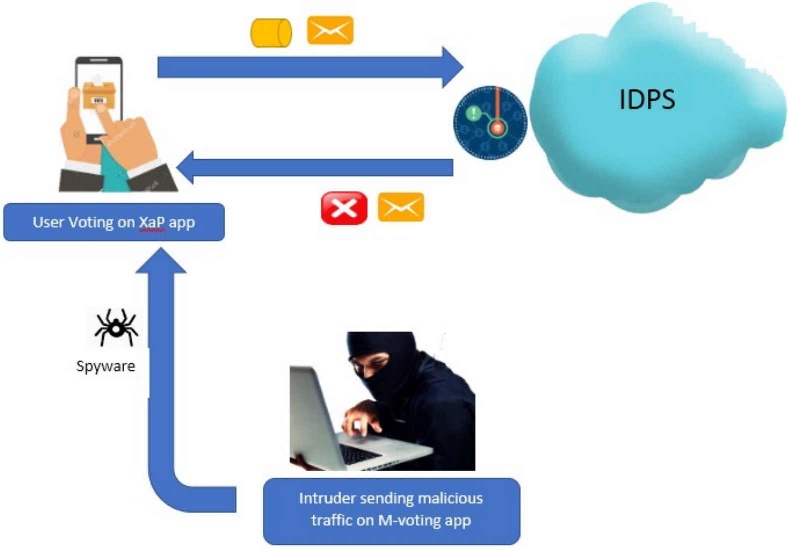


Figure 1: Cloud Intrusion Detection and Prevention System architecture

Figure 1 above illustrates the proposed architecture for cloud based intrusion detection and prevention system for M-voting in SA (CIDPS).

3. Technology Description

The security system was called Cloud Intrusion Detection and Prevention system (CIDPS) because it used cloud computing to solve the limited resources of mobile phones and Intrusion Detection and Prevention system to detect and prevent any intrusions directed to M-voting. A summary of the technologies that was used to develop CIDPS are listed below.

The main methodology used in this paper is the literature that was reviewed. Also, we used the three detection methods of IDPS, namely:

1. *Signature detection method*: This method is also known as Misuse detection method. Signature detection method is the content of the file dictionary of malware signature, an evidence of the recorded intrusions. This method need a huge database to store the detected malware signatures [9].
2. *Anomaly Based Detection*: This method is also known as behaviour detection method and it monitors the run time behaviour of the mobile application and compares the malicious or normal behaviour profiles to detect the malware [2, 9].
3. *Hybrid Detection Method*: According to Kaladevi and Sumitra [9] this method is a combination of both the signature and anomaly detection method. This method is used to overcome the limitations of both signature and behaviour based method [2,9]. A hybrid detection method was utilised in our proposed system.

Also, Intrusion detection systems (IDS) can be classified into different ways. Two of the major classifications are active and passive IDS:

1. *An active Intrusion Detection Systems (IDS)* is also known as Intrusion Detection and Prevention System (IDPS). Intrusion Detection and Prevention System (IDPS) is configured to automatically block suspected attacks without any intervention required by an operator. Intrusion Detection and Prevention System (IDPS) has the advantage of providing real-time corrective action in response to an attack [8,10].
2. *A passive Intrusion Detection Systems (IDS)* is a system that’s configured to only monitor and analyse network traffic activity and alert an operator to potential vulnerabilities and attacks. A passive IDS is not capable of performing any

protective or corrective functions on its own [1, 3]. An active IDPS was adopted in our proposed security solution.

Table 1: CIDPS technology description

CIDPS components	Available options	Chosen option	Description
Cloud analysis engine	Bro Snort TippingPoint X505	Snort	Snort is a libpcap-based [PCAP94] packet sniffer and logger that can be used as a lightweight network intrusion detection system (NIDS).
Client agent (on mobile phone)	Mobile sensor	Mobile sensor	Monitors, collects user, sensor inputs from the device interface in run time and sends them to the cloud analysis engine.
Cloud computing service model	Software as a Service Platform as a Service Infrastructure as a Service	Infrastructure as a Service	Infrastructure as a Service (IaaS) layer virtualises computing power, storage and network connectivity of the data centres, and offers it as provisioned services to consumers. Users can scale up and down these computing resources on demand dynamically. Typically, multiple tenants coexist on the same infrastructure resources. Examples of this layer include Amazon EC2, Microsoft Azure Platform.
Cloud platform	Amazon EC2 Microsoft Azure Google app engine	Microsoft Azure	Is a cloud storage system that provides customers the flexibility to store huge amount of data, and it can be stored for any duration. The individuality of the data that is stored with Microsoft Azure is that you can access the data anywhere and at any time.
Simulation tools used for evaluation	Genymotion AVD VMware	Genymotion	Is an Android emulator based on virtual box. The good thing is that it does not require the developer to install virtual box as it is bundled with the installer. Additionally, it can emulate specific devices and allows the developer to install/run/test apps on it, which makes it great for the developer to use it on a daily basis or just to test applications

4. System Design and Implementation

The security solution was developed such that it protects an M-voting system called XaP application. XaP application was developed for South Africa in South African context. The application was called XaP because it cuts (X) out the queuing time and all other inconveniences brought by traditional paper-based voting system, which makes voting fast and easy as possible [13]. The computational resources of mobile phones are limited; it is for this reason that the CIDPS incorporated cloud computing. Cloud computing is the practice of delivering computational services such as computer servers, databases, networking, software and more over the internet "cloud" [16]. Users and clients can submit a task to the service provider without possessing the required software or hardware [1].

The cloud analysis engine used the Snort IDPS. Snort is a signature based IDPS and it has a language to define new rules, it has an architecture making it possible to add new functionalities at the time of compilation. Snort is the most widely deployed intrusion detection and prevention technology worldwide.

The filtered traffic passed through the Snort attack detection engine to detect various active and passive attacks. With the help of Snort, we were able to analyse and monitor live traffic and capture packet flow along with anomalies in the network. Snort consist of the following four components illustrated in the figure below:

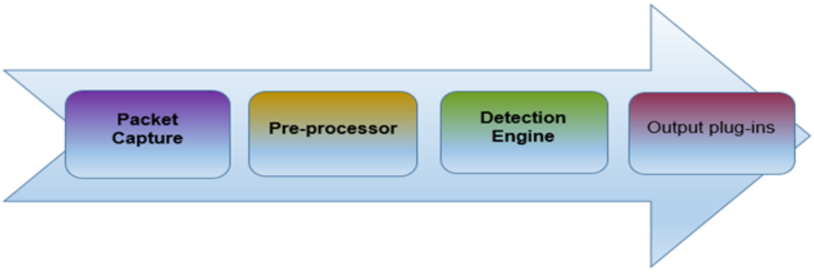


Figure 2: Snort components [15]

We used Snort pre-processor to filter incoming client sessions to distinguish the beginning of the session. Pre-processor are engines that have the ability to give alerts, ignore or edit packages before they reach at the analysis engine running in the cloud [22]. Pre-processors that were built into Snort implemented the source code file “spp_phad.cpp” which was copied to the directory where “Snort.c” was installed and configured. We used #apt-get install snort as the command to install Snort. To grant Snort a full privilege to MySQL database, the below configurations were used:

```
#mysql>grant usage on snort.* to snort@localhost;
#mysql>
#mysql>set password for snort@localhost=PASSWORD(‘pwd4snort’);
#mysql>
#mysql>grant all privileges on snort.* to snort@localhost;
#mysql>
#mysql>flush privileges;
#mysql>exit
```

Figure 3: Snort privileges

Usually the IDPS is behind firewall and antiviruses in order to detect intrusions that firewalls have missed. Generally, our IDPS gives an extra protection layer to the defence strategy already installed in the mobile phone [14]. Figure 4 illustrates the placement and in-depth strategy of the security system where on the left are some of the possible threats from the internet endangering the overall security of M-voting network.

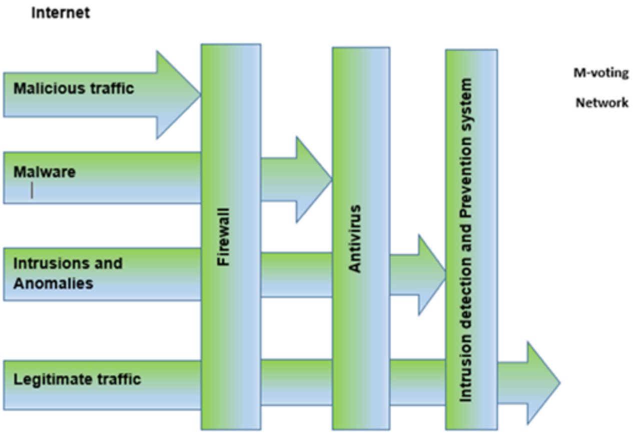


Figure 4: In-depth defence strategy

The basic idea in layered in-depth defence strategy is to enhance the overall security of the protected system.

5. Results

CIDPS security system was evaluated using accuracy, performance and timeliness as our evaluation criteria. We used Genymotion as our simulation tool to simulate the effectiveness of our system. Firstly, the voter had to download XaP application, once XaP is downloaded, the voter installs XaP (as depicted in Figure 5 and 6). The CIDPS checks the status of the mobile phone by scanning for malwares before XaP can used. Once the status has been confirmed, meaning there are no malwares found, a message will be shown to indicate that as depicted in Figure 6.

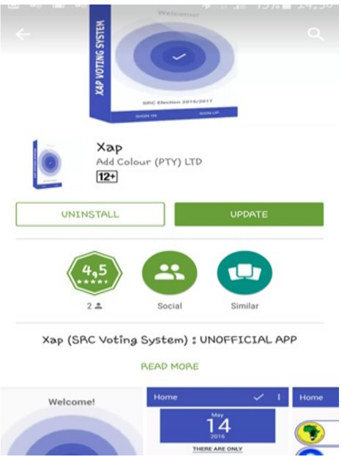
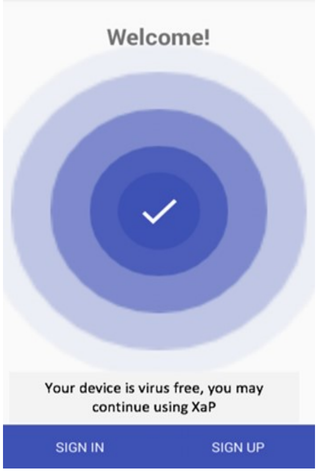


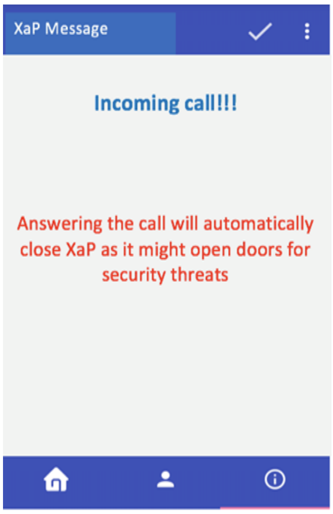
Figure 5: XaP installation



Your device is virus free, you may continue using XaP

Figure 6: Malware scan

The voter signs in if they are already registered to vote or signs up if it is for the first time using XaP (as depicted in Figure 7). The CIDPS was designed in such a way that in detects and prevents attacks only when the mobile device is using the XaP application. Once the mobile device is used for any other applications, it is open to threats and attacks. XaP messages are used to communicate the status of the mobile device with the mobile user as shown in Figure 8 below.



Incoming call!!!

Answering the call will automatically close XaP as it might open doors for security threats

Figure 7: XaP login

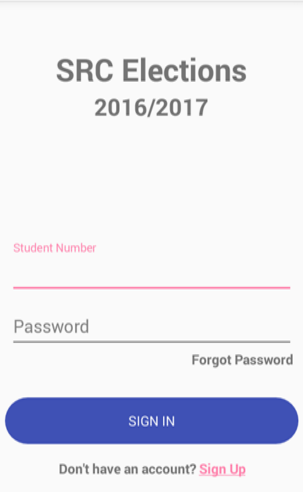


Figure 8: XaP message

Once the login is successful, the voter is allowed to use XaP to cast their vote. The CIDPS monitors the status of the mobile phone from the time the voter opens XaP until the application is closed.

5.1 The Snort Database Used Three Separate Signatures

Scenarios were established to verify the effectiveness of the proposed CIDPS. A large sample of attacks was used to test the proposed system. These attacks covered different classes of attack types and contained exploits for both newly discovered as well as older well-known vulnerabilities. The Snort rules downloaded were the most recent updated rules from the official snort portal [https://www.snort.org/downloads/#rule-downloads].

Table 2: Evaluation results for Snort

Attacks	Results
Baseline attacks	100%: detected
Mutated attacks	40%: detected
Evasion techniques	60%: evated

- *Baseline attacks*
  - The baseline attack results represent the ability of the CIDPS to correctly detect the baseline attack when the exploit was not subjected to any mutation technique. In all cases, the malicious packets were identified as malicious and an alert was logged to the alerts file.
- *Mutated attacks*
  - The mutated attack results indicate whether the CIDPS was able to detect the baseline attack and all the mutations of the same attack attempted during the experiment. In the first experiment, 40% were detected and 60% were evaded.
- *Evasion technique*
  - Evasion techniques are key techniques that could enable the mutated exploits to evade detection.

6. Conclusion and Recommendations

M-voting is an application that guaranteed the civic right of the citizen through providing new service that ensured availability, privacy and secrecy to the voter. However, security is one of the vital aspects of voting process. Consequently, using pervasive technology like mobile phones to cast a vote could be one of the most significant threats to the citizen participation in M-voting if its security challenges are left unattended. It is for this reason that a security solution called Cloud Intrusion Detection and Prevention system was developed to protect the confidentiality and integrity of M-voting application during the voting procedure.

Evaluations were done using simulations, this can be taken forward by implementing the security system in real life using real equipment and networks. Unfortunately, not a lot of companies and even government are willing to test M-voting for any elections. If an opportunity like that could be given to the researcher, the proposed security solution including the XaP application can be improved to such an extent where it can even be used during national elections.

In future, we hope that the security system can be tested on any voting application not only on XaP application and can also be developed such that it is able to protect any mobile device using any operating system not only Android operating system.

References

[1] B. Azuan, G. Norbik, M. Sameer and F. Hasan. Danger Theory Based Hybrid Intrusion Detection System for Cloud Computing. International Journal of Computer and Communication Engineering, (2013): 650-653.

[2] AH. Bhat, P. Sabyasachi, and Debasish Jena. "Machine learning approach for intrusion detection on cloud virtual machines." International Journal of Application or Innovation in Engineering & Management (IJAIEM) 2.6 (2013): 56-66.

[3] M.A. Aydın, A. Halim Zaim, and K. Gökhan Ceylan. "A hybrid intrusion detection system design for computer network security." Computers & Electrical Engineering 35.3 (2009): 517-526.

[4] U.O. Ekong, and C. K. Ayo. "The Prospects Of M-Voting Implementation In Nigeria." 3GSM & Mobile Computing: An Emerging Growth Engine for National Development (2007): 172-179.

[5] U. Ekong and V. Ekong. "M-voting: a panacea for enhanced e-participation." Asian Journal of Information Technology 9.2 (2010): 111-116.

[6] R. Gibson, A. Römmele, and S. Ward, eds. Electronic democracy: mobilisation, organisation and participation via new ICTs. Routledge, 2004.

[7] R. Dhaya and M. Poongodi. "Mobile Virus Prevention Techniques: A Survey Perspective." International Journal of Innovative Research in Computer and Communication Engineering 2.1 (2014): 1980-1985.

[8] K.K. Hausman, S.L. Cook, and T. Sampaio. Cloud Essentials: CompTIA Authorized Courseware for Exam CLO-001. John Wiley & Sons, 2013.

[9] P. Kaladevi and P. Sumitra. "A study on Intrusion Detection Systems in Cloud Computing." database 2.1 (2016).

[10] S. Kale & V. Bhosale. Revised Approach for Smartphone Security Using Cloud and Android Applications. International Journal of Advanced Research in Computer Engineering & Technology, 2015; 3723-3729.

[11] K. Ok, V. Coskun, and M.N. Aydin. "Usability of mobile voting with NFC technology." Proceedings of IASTED international conference on software engineering. 2010.

[12] A. Khelifi, Y. Grisi, D. Soufi, D. Mohanad and P.V.S. Shastry. "M-Vote: a reliable and highly secure mobile voting system." Information and Communication Technology (PICICT), 2013 Palestinian International Conference on. IEEE, 2013.

[13] O.P. Kogeda, and N. Mpekoa. "Model for A Mobile Phone Voting System for South Africa." 2013 Conference. 2013.

[14] C.C. Kotkar, and P. Game. "Exploring Security Mechanisms to Android Device." International Journal of Advanced Computer Research 3.4 (2013): 216.

[15] G.V. Nadiammai and M. Hemalatha. "Handling intrusion detection system using snort based statistical algorithm and semi-supervised approach." Research Journal of Applied Sciences, Engineering and Technology 6.16 (2013): 2914-2922.

[16] M. La Polla, F. Martinelli, and D. Sgandurra. "A survey on security for mobile devices." IEEE communications surveys & tutorials 15.1 (2013): 446-471.

[17] I. Raja, "A Cloud-based Intrusion Detection Forensic Analysis on Smart Phones." Journal of Global Research in Computer Science 4.4 (2013): 204-207.

[18] M.P.K. Shelke, M.S. Sontakke, and A. D. Gawande. "Intrusion detection system for cloud computing." International Journal of Scientific & Technology Research 1.4 (2012): 67-71.

[19] D. Vaghela. "Analysis and Security Testing of Android System by Malware Detection in Networks." International Journal of Engineering Sciences & Research Technology 1.4: 274-278.

[20]W. Voorsluys, J. Broberg, and R. Buyya. "Introduction to cloud computing." Cloud computing: Principles and paradigms (2011): 1-44.

[21] S. Zonouz, A. Houmansadr, R. Berthier, N. Borisov, W. Sanders "Secloud: A cloud-based comprehensive and lightweight security solution for smartphones." Computers & Security 37 (2013): 215-227.

[22]SnortUsersManual2.6.1;3December2006.<www.snort.org/docs/snort_manual/2.6.1/snort_manual.pdf>