Waziri Victor Onomza* et al.
(IJITR) INTERNATIONAL JOURNAL OF INNOVATIVE TECHNOLOGY AND RESEARCH
Volume No.3, Issue No.2, February – March 2015, 1882 – 1896.

Development of Secure Plus Antivirus with the Artificial Immune System Model

WAZIRI VICTOR ONOMZA
School of Information and Communications Technology
Cyber Security Science Department
School of Information and Communications Technology
Federal University of Technology,
Minna, Niger State, Nigeria

JOHN ALHASSAN
School of Information and Communications Technology
Cyber Security Science Department
School of Information and Communications Technology
Federal University of Technology,
Minna, Niger State, Nigeria

MORUFU ALELERE
School of Information and Communications Technology
Cyber Security Science Department
School of Information and Communications Technology
Federal University of Technology,
Minna, Niger State, Nigeria

ABDULRAHMAN TUNDE
School of Information and Communications Technology
Cyber Security Science Department
School of Information and Communications Technology
Federal University of Technology,
Minna, Niger State, Nigeria

Abstract: **This paper is about Malware proliferation in the wide and the development of an Antivirus called Secure Plus. Malware is a generic name for malfunctioned program codes that could wreak destructive impacts on Information Technology critical infrastructures. These malware usually use various techniques to avoid being detected; usually they are encrypted using hybridized cryptographic algorithms. Malware may be detected using antivirus that can scan the database signatures already accumulated and stored by antivirus vendors in some server. These stored databases signatures can then be compared with zero-day malware through comparison with the benign software. The zero-day malware are of sophisticated program codes that can transmute into different transforming patterns; yet retain their portent functionalities attributes and are now of billion categories by deverse clones. This paper after over viewing the literatures on ground (and they are of large numerical numbers), attempts to make its contribution to the design and development of Antivirus that can detect those zero-day or metamorphic malware. This proposed Antivirus being developed is christened Secure Plus that applies the heuristic Artificial Immune System Algorithm for the design and development. The tested experimental outputs are provided as prove of the Secure Plus effectual functionality worthy of application but need further works through to detect malware proactively.**

Keywords: **Secure Plus, Polymorphism, Metamorphism, Metamorphism, artificial immune system**

INTRODUCTION

Classification of normal and abnormal files on a computer system is a very tedious task that is challenging the resourcefulness of computer security Professionals' workmanship. The growth of malware in various shapes and dimensions and their ability transmute are keeping antivirus vendors on their heels and making new computing security technologies to evolve by the day according to Moore's Law. With advent of new communicating technologies, so are the evolutions of sophisticated new generations of Malware (malicious malware) skills and propagation techniques in order to meet the surging nefarious braggadocios demands on the Internet globally and within seconds. These new investments by the malicious creators have introduced new terminological tools into the labour market that have graduated from simple oligopoly, polymorphism obfuscation to now more adaptive and devastating metamorphic malware. Of the three coinage abstractive developments of the malicious creatures, metamorphic malware is stealthier and comes in various morphed formations and like apparitions of the ancient times, they have defied being detected by various antivirus technological tools or keep the antivirus vendors on their heels like they are in physical warfare when they can change their antics on discovery of their techniques to congest further detection. This paper attempts to design and develop a new antivirus that we christened SecurePlus for computer worms that can be automated and developed through adaptive artificial immune systems just as in the immunological protecting systems of the vertebrate organisms through their self and nonself cells adaptive morphism. This development can be achieved by the application of the biologically inspired computational intelligent process; the artificial immune system (AIS) (Goldsby et al, 2002).

Malicious codes were first designed and developed in 1981 (malicious software now known with a generic name 'Malware" are program codes written by despicable experts) that could be grouped into various patterns according to their reprehensible functionalities and have been increasing fast since their first creation in the year aforementioned that now are running in billions by counts and having diversified malicious craftsmanship constructions. These morphed codes that are written by human creatures have been wreaking devastating influences on files and causing substantial destructions on operating systems, their accessories and networking technologies in the Internet. The programs are more intelligently surreptitious and shred than their creators; and like human spies, carry out to the letter their creators' biddings without failures unless detected and disinfected. Some of the malfunctioning Methods used by viruses are by attachments, substitution of program routines and various techniques such as junking are exponentially becoming more advanced through obstructive obfuscation manipulations, which are maliciously causing the scanning of systems and other communicating devices nutty and complications to detect (Deng et al, 2003).

1.1 Artificial Immune Systems (AIS)

In his research on AIS, Dasgupta, (1999) refers "Artificial immune systems as intelligent and adaptive systems that is inspired by the immune system toward real-world problem solving". Stephen and Forrest (2000) conceived the AIS as incorporate various immunological properties of natural immune systems that incled diversity, distributed computation, error tolerance, dynamic learning and adaptation and self-monitoring. Thus, the human immune system can be used as inspiration when developing algorithms to solve difficult computational problems due to the fact that it is a robust, decentralized, complex, and error-tolerant biological system; i.e. it possesses properties that make it ideal for certain application areas, such as computer intrusion detection and pattern recognition. The human system defensive mechanism is also well-studied within the realm of immunology, and is viewed as the most sophisticated of immune systems in nature. Although its precise function remains undetermined, it is postulated that it has two roles; to protect the body against invading micro-organisms (pathogens), and to regulate bodily functions (homeostasis) (Julie, et al,.2010).

The vertebrate immune system is composed of diverse sets of cells and molecules that work together with other systems (like neural and endocrine) for maintaining homeostatic state. The primary function of the immune system is to protect human bodies from infectious agents (such as viruses, bacteria and other parasites) commonly known as *pathogens*. Immune response is incited by the recognition of an associated molecule called *antigen.* Immune system usually works according to two mechanisms namely: Innate and Adaptive Immunity. Innate immunity is directed against general pathogens that enter the body while adaptive or acquired immunity allows launching an attack against any invader that innate system cannot digest. For more information about the immune systems, the reader is referred to (Castro and Zuben , 1999; Castro and Timmis, 2003; Timmis et al., 2004, Al-Enezi, et al. , 2010).

Innate immunity: Vertebrates are born with this immunity which plays a vital role in the initiation and regulation of immune responses. Specialized cells have evolved to recognize and bind to common molecular patterns of micro-organisms. By no means does it provide complete protection, as it is primarily static in nature (Castro and Zuben, 1999).

Adaptive immunity:That **t**he immune system is adaptive connotes that it can recognize and respond to previous and new infections. It also retains memory of the encountered infections that fasten future and rapid response when the body is attacked by the same or similar antigens. Therefore, AIS is directed towards specific invaders; either seen before or not previously encountered and gets modified by exposure to invaders. It mainly consists of lymphocytes (white blood cells, more specifically B and T type) that aid the process of recognizing and destroying specific substances, and are antigen-specific (Castro and Zuben , 1999). The adaptive immune construct consists of two recognized types that are useful in soft computing; they are Negative and Clonal Selections:

Clonal Selection

Clonal selection theory was proposed by Burnet (1959). The theory is used to explain basic response of adaptive immune system to antigenic stimulus. It establishes the idea that only those cells capable of recognizing an antigen will proliferate while other cells are selected against. Clonal selection operates on both B and T cells. B cells, when their antibodies bind with an antigen, are activated and differentiated into *plasma* or memory cells. Prior to this process, clones of B cells are produced and undergo somatic hyper mutation. As a result, diversity is introduced into the B cell population. Plasma cells produce antigen-specific antibodies that work against antigen. Memory cells remain with the host and promote a rapid secondary response (Castro and Timmis, 2003).

Negative Selection

Negative selection is a mechanism that protects the body against self-reactive lymphocytes. It utilizes the immune system's ability to detect unknown antigens while not reacting to the self-cells. During the generation of T-cells, receptors are made through a pseudo-random genetic rearrangement process. Then, they undergo a censoring process in the thymus, called the negative selection. In this process, T-cells that react against self-proteins are destroyed and only those that do not bind to self-proteins are allowed to leave the thymus. These matured T-cells then circulate throughout the body performing immunological functions and protecting the body against foreign antigens (Somayaji et al., 1997).

Immune Networks Theory

The immune Network theory has been introduced by Jerne (1974). The main idea was that the immune system maintains an idiotypic network of interconnected B cells for antigen recognition. These cells interconnect with each other in certain ways that lead to the stabilization of the network. Two B cells are connected if the affinities they share exceed a certain threshold, and the strength of the connection is directly proportional to the affinity they share (Castro and Timmis, 2003)

1.2 Motivation

As information technology keeps on evolving, present day society faces the problem of information security from malicious software. More so, if the problems of already known malware are not resolved quickly and, the confidentiality, integrity, availability information that forms the nitty-gritty of information Security will not also be assured; that will lead to the entire loss of the gains so far achieved in telecommunication technologies globally. This alone, necessitates the development of protective and proactive software that is capable of mitigating the influence of these malware to the barest optimal minimum is highly demanded.

With this and other similar cases which are not mentioned in this project, we are motivated to center our research light on artificial immune system for script virus detection and disinfecting, which we believe could go a long way in solving some of the above mentioned problems of malware havoc on computer systems. Ultimately, it is important to learn from must be cornered and ameliorated with once and for all.The biological phenomena inspirations like the AIS are being considered as the basic tool for this inspirations that could be applied to meet the innovative solutions in order to protect systems from software pathogens or malicious software.

For more information about the immune systems, the reader can refer to (Castro and Zuben , 1999; Castro and Timmis, 2003; Timmis et al., 2004, Al-Enezi, et al. , 2010). However, figure 1.1 illuminates the effectual function activities of the AIS.

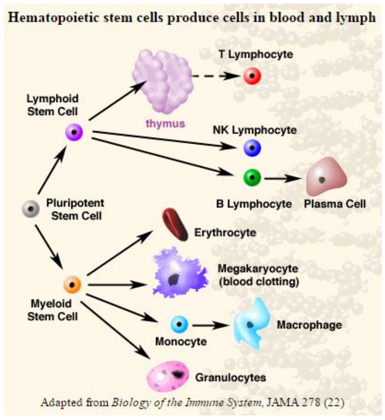


Figure1.1: Depicts the development of Immune System (see the website)

1.2 Motivation

As information technology keeps on evolving, present day society is facingsome security problems of information safety from malicious software developers and their clients. More so and to put it straight on the face, if theproblems of already known malware are not resolved quicklythe issues of confidentiality, integrity andavailability ofinformationhangs in the balance ofnuts and boltsantics of black hackers worldwide;and so to sum it all, that will lead to the entire loss of the gains so far achieved in telecommunication technologies globally. This alone, necessitates the development of protective and proactive design and development of software that iscapable of lickingthese malware threats at their bulbs, if not, to the minimaloptimal control.

In consequence of the above details and ultimately too, it is important to make some contribution inorder to protect systems from software pathogens or malicious software.

1.2.1 Malware

Malware detection isthe act of investigating into computer viruses and other malware sources by evaluating the string of bits or bytes that are encoded in the virus or malware programs. These chains of strings are known as the fingerprint of the virus which a identified as the signature of the virus. The methods or models used in detecting viruses are signature based detection using antivirus.

There have been massive techniques developed to detect malware activities in the wide Bits (2011). This is exemplified by loading distributing executable files on the Internet which would present a risk to overall security of the system, Carrera and Erdelgi (2004). As noted by the two authors in the immediate oforestated, malware programmers have the techniques of installing hidden malicious codes in an innocuous remote files or application surreptitiously with the cardinal aim of threatening the existing system.In another development by Islam, et al. (2011), they examined the risk of downloading more than 450, 000 files of which they found approximately 18% malicious codes programs. They went further to investigate various techniques for malware on whether malware different codes could yield the same results. With overwhelming amazement, they discovered that there were many cases where forensic tools were unable to detect the malware contents of the infected files.

For want of space, the reader could learn more on malware from the following authors referred(Daniel. 2006, Ed Skoudis,2004Sean and John: the Craft of System Security, Peter Szor, 2005), John Viega, 2009, John Viega, 2008).

RELATED WORKS: ARTIFICIAL IMMUNE SYSTEM

AIS is a heuristic technique, in the context of computing, Heuristics techniques refer to the process of finding solutions to problem by trial and error or by rules that are only loosely defined. This paper reviews the literatures which are related to the design of antivirus development techniques. In the application of AIS for the development,

The concepts of innate and adaptive biological immune systems are used as direct physical models for developing virus pattern recognition, computer immunological memory, and autonomic virus patch software. Given the evolving business environment where malicious software threats (e.g. worms, viruses, infectious agents) are becoming commonplace, the development of virally immune self-healing or self-defending information systems networks appears to hold some promise.

In their novel seminar research works, (Al-Enezi, et al., 2010), present some prominent researches that have been recently covered in them alware study.

The Clonal Selection principle is the whole process of antigen recognition, cell proliferation and differentiation into memory cell (Burnet, 1959). Several artificial immune algorithms have been developed imitating the clonal selection theory. Castro and Zuben (2002) proposed a clonal selection algorithm named CLONALG for learning and optimization, CLONALG generates a population of N antibodies, each specifying a random solution for the optimization process. During the iteration stages, some of the best existing antibodies are selected, cloned and mutated in order to construct a new candidate population. New antibodies are then evaluated and certain percentage of the best antibodies is added to the original population. Finally a percentage of worst antibodies of previous generation are replaced with new randomly created ones.

In their research paper,Rouchen et al. (2003) presented Immunity Clonal Strategy algorithm (ICS) thatincluded Immunity Monoclonal Strategy Algorithms (IMSA) and Immunity Polyclonal Strategy Algorithm (IPSA) are introduced. ICS is used to solve multi-objective optimization task.In furtherance of the research literature,Zuo and Li (2003) proposed a Chaos Artificial Immune Algorithm (CAIF) that is applied forsolving function optimization problems. It uses chaotic variable which performs local search and explore the solution space.

Garrett (2004) was also specific in introducing an Adaptive Clonal Selection (ACS) algorithm as a modification of CLONALG. He suggests some parametric modifications to the CLONALG based on an analysis of the operators for selecting the amount of mutation and number of clones to overcome the drawbacks of CLONALG such as the several parameters used and the binary representation. An Adaptive Immune Clonal Strategy Algorithm (AICSA) ia another proposal that was proposed for solving numerical optimization problems in Liu et al (2004). This proposal assigns dynamically the immune memory unit and antibody population according to the Ab-Ab and Ab-Ag affinities. It also integrates the local search with the global search algorithms.

Yu and Hou (2004) in their contributions presented an improved clonal selection algorithm based in CLONALG algorithm. It is a learning operator that was introduced to enhance the learning mechanism of CLONALG and to improve the detection efficiency. Campels et al. (2005) further went another staggering step to proposethe Real-Coded Clonal Selection Algorithm (RCSA) for electromagnetic design optimization. This proposal suggests some parametricmodifications to the clonal selection algorithm to enable the treatment of real valued variables for optimization problems. This development has some features such as the number of clones, mutation range and the fraction of the population selected by each generation. In their rightful postulate to further enhance the application of AIS,Cutello et al. (2005) reworked on immunological algorithm which introduced the continuous global optimization problems named OPT-IA. They propounded that the main features of the proposed algorithm are the following: the cloning operator that explores the neighbourhood at each point within the search space; The inversely proportional hypermutation operator used in the algorithm where the number of mutations are inversely proportional to the fitness value; and finally, the aging operator is used to remove the oldest candidate solution from the current populations to introduce diversity and avoid local minima during the search process.

Another adaptive clonal algorithm was proposed in Bian and Qiu (2006) for optimal phasor measurement unit (PMU) placement. It adjusts the number of the cycle supplement population and the probabilities of hypermutation and recombination operators of the CLONALG algorithm. These modifications can enhance the optimization process and help to avoid the locally optimal traps. Cutello et al. (2006) in making improvement to their AIS research introduced an improved version of OPT-IA called opt-IMMALG. The main modifications in this algorithm are the replacement of the binary string representation by a real-coded one and the introduction of a new inversely proportional hyper mutation operator.

In the eyes of Gong et al. (2007a) a presention, an improved clonal selection algorithm based on CLONALG with a novel mutation method, self-adaptive chaotic mutation were introduced. The main modifications are that the new algorithm adopts the logistic chaotic sequence to generate the initial antibody population, while the hypermutation adopts self-adaptive chaotic mutation. In the same conference proceeding, Gong et al. (2007b) offered a Differential Immune Clonal Selection algorithm (DICSA) and was proposed to solve the global optimization problems. It combines the clonal selection theory and differential evolution and employs three operators: a clone operator, a differential mutation crossover mutation and a standard selection operator.

A parallel clonal selection algorithm for solving the Graph Coloring Problem presented in Dabrowski and Kubale (2008). It uses an island model where every processor works on its own pool of antibodies to improve the performance. Lu and Zhichun (2008) proposed a Clonal Chaos Adjustment Algorithm (CCAA) for Multi-modal Function Optimization. In order to enhance the global convergence performance of CLONALG, it takes advantages of the ergodic and dynamic properties of chaos system, and introduces the chaotic search mechanism into the CLONALG to improve its search efficiency.

Many other clonal selection based algorithms have been introduced in the literature mostly with the intend of solving global optimization problems. Examples of these algorithms include: Jiao and Li (2005), Li et al. (2005), Jin et al. (2006), Xiu-li and Yu-qiang (2006), Halavati et al. (2007), He and Jian (2007), Hu et al. (2007), Chen (2007), Zhang et al. (2007), Li et al. (2008), Qiao et al. (2008) and Yang et al. (2008).

2.1 Some Experimental Details on AIS Based on Malware Detection

More than a decade ago (Harmer et al, 2002) used biological strategies to develop a self-adaptive distributed based defense immune system. Furthermore,the cooperated authors designed a prototype of the system and implemented it using java programming language. The main aim of their work is the ability to identify the presence of malicious software patterns inside a huge set of existing self-patterns. They also pointed out that, the Antivirus and intrusion detection systems at that time could overpower detecting viruses, which was one of the major aims of their works. Their prototype implementation gives a better way of detection, identification, and elimination of viruses. However, before that time Hofmeyr (1999), proposedsome excellent work on virus classification using AIS. His algorithm assumed the detection and classifying of malware based on input pattern of the IP packages. He assumed mathematically a set of finite discrete space X that might conceived as consisting of binary bits and represented this finite set as $X \in \{0,1\}^l$ or $X = \{0,1,...,255\}^{\frac{l}{8}}$, where 0 representsnonself and I, self while l is the order of the finite space.Hofmeyr (ibid) in furthering the development of the classification malware with the benign software, opined the assumption of of two subsets of X, namely the self $S \subseteq X$ and the nonself $N \subset X$ such that the union of the two set is $S \cap N = X$ .and $S \cap N = \varnothing$.

For virus detection, the nonself proposition represents malicious viral codes while the self represents the is the representative of the legitimate benign programs. For the Intrusion Detection System, the nonself is construed as the Internet Packets (IPs)from the computer network attack, while self patterns are normal sanctioned network service transactions and non-malicious background cluster.

Hofmeyre went in detail with the task of of the detection algorithm as a process of classicatication of the input pattern data $I \in X$ as either self or nonself. Therefore for an input string $I:\{0.1\}^{l}$, a detector set

$$D: D = \{\sim_1, \sim_2, \ldots, \sim_i\} \qquad (2.1)$$

where,

$$\sim = \{0,1\}^{k},\ k \in l;\ i \in N, \qquad (2.2)$$

a matching threshold $\in$, the classification as self and nonself can be actualized as follow:

$$(f, \in, I, D) = \begin{cases} Malicious: \ f((I, \sim) \geq 1 - \in \\ Benign, \quad Otherwise \end{cases}$$
(2.3)

This detection methodology if properly constituted can generate two types of errors that are stated in the following definitions:

Type I, or false positive errors; and

Type II, or false negative errors

Definition, a false-positive error $₵^{+}$ occurs when a member of the self set S is incorrectly classified as malicious. Conversely, a false-negative $₵^{-}$ is the classification of a member of the nonself set N as benigh. This can be represented mathematically as

$$(I \in S \cap match(f, \in, \operatorname{Im} D) \\ = malicious) \rightarrow ₵^{+} \qquad (2.4)$$

$$(I \in N \cap match(f, \in, \operatorname{Im} D) \\ = malicious) \rightarrow ₵^{-} \qquad (2.5)$$

Fig 2.2 is used in the detection of self and nonself patterns within a larger potentially larger set of existing self-patterns. The algorithm used static string method for generating random strings.

In furthering of the literature revisions, (Rui and Ying, n.a) proposed a virus detection system (VDS) based on artificial immune system, the system starts by generating a detector set from virus files in the database, the authors applied negative selection and clonal selection to a detector set in order to eliminate autoimmunity detectors and increase the diversity of the detector set in the non-self-space respectively. They used affinity vectors of the training set and the testing set to train and test the classifiers respectively.VDS compares the detection rates using three classifiers, k-nearest neighbor (KNN), RBF networks and SVM when the length of detectors is 32 $- bit$ and 64 $- bit$. The experimental results show that the proposed VDS has a strong detection ability and good generalization performance. Below is a table 2.1 for the proposed system showing the number of files in each dataset.

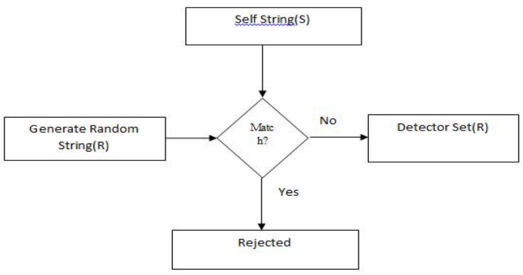


Fig 2.1 The proposed Negative selection Algorithm

Their experiment result from table 2.1 shows that there are 284 benign files in total i.e. the sum of benign files for both the training set and the testing set(71+213=284), with 78MB in total, and 3547 virus files i.e.(885+2662=3547) with 7.8MBin the data set (DS). Another dataset contains 208 benign files is used for negative selection, totally 189MB. All the benign files are system files or well-known programs with extensions .exe.The DS is randomly divided into different percentages of training set and testing set as shown in table 2.1, there is no overlap in the two sets.

Table 2.1 Result of the first virus experiment

	Training Set		Testing Set	
Data sets	Benign files	Virus files	Benign files	Virus files
Dataset1	71	885	213	2662
Dataset2	142	1773	142	1774
Dataset3	213	2662	71	885

In this paper,(Tony et al, 2004), used n-grams analysis to detect new malicious codes automatically. The authors also stated their aim for coming up with this method, which is that, the current commercial antivirus systems that used signature based detection method in most cases detect virus after it has caused damage to the system. They also pointed out that, signature method of virus detection performed poorly when it comes to detecting new viruses or zero-day malware. Their system is capable of classifying good file from malicious file. Ultimately, one of their major contributions is the use of large dataset as compared to the previous ones.

Feng, (2007), in his contribution stated reasons why some antivirus software are prone to attacks. At the end, he proffers some lasting solution to those problems. One important term he pointed out was “Detection Bypass”, he stated that “Detection Bypass is a problem of relatively low severity. It is more important for a server side Anti-virus than desktop Antivirus software”. According to his model, the attack is usually achieved when attackers manipulate zip file containing Trojan and then sends it as an attachment to the victims system. He also noted that in most cases, after successful manipulation of the zip file, the Antivirus software is usually unable to parse the zip file which normally would result to false negative output.

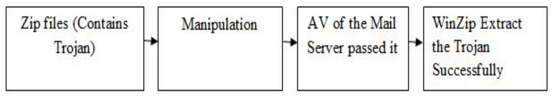

Zip files (Contains Trojan)
Manipulation
AV of the Mail Server passed it
WinZip Extract the Trojan Successfully

Fig 2.2 Detection Bypass.

(Wang et al, 2002) in their paper used inspiration from the natural immune system and proposes a system which is based on matching in three layers to detect many types of viruses. Their proposed system was designed in three layers, i.e., top, middle, and bottom. In the top layer, a matching was applied in order to improve information loss from the system. In the middle layer, new storage method is applied in order to keep track of the important signatures on the system.

2.2 Literature Review Conclusion

Based on all the literatures reviewed so far on the area of artificial immune system for viruses detection and disinfection, it is observed that most of the programmers used advanced method of virus detection, such as heuristic analysis, fuzzy logic and some machine learning algorithms. Their methods, despite its advancement and cost, usually result to false positive and false negative. These issues and many other issues not actually mention in this project, are some of the challenges they faced. Ultimately, this project tries to avoid the above mentioned problems by limiting the scanning to a specific extension, which is the “.exe”,”.doc” and script file extension.

METHODOLOGY

The focus of this research project and in this section in particular, is to conduct a design and development of the antirus based-on the Artificial Immune System (AIS) algorithm for malware classification and its disinfection using VB.net and C++. The designed secure coding could be used in the development of the system antivirus which we shall a symbolic coinage as **SecurePlus**.

3.1 Analysis of Current Virus Detection Systems

In brief, we present the traditional detection methods being utilized as anomaly detection system to recognize malicious codes that could be divided into three major categories: Behavior-based, Data-based and the Emulator (Smith, 2001), which are defined as follows:

***(i)Behavior-based**:* This monitors the behavioral-patterns of the malicious code as it executes sequentially on the application programming interface (API) dynamically. These methods at first construct outlines the legitimate operations of the monitored programs. During the detection process, any system call sequence or argument that does not comply with the previously generated normal profiles is regarded a sign that the system is compromised. The corresponding program will be stopped and then classified as a malicious program or malware. In most cases, the malicious programs may be identified when the computer is already damaged. Thus many methods use virtual environments called sandbox to simulate real system environment where the unclassified programs are running. However, Malware developers have developed more effective algorithms that could evade this virtual process.

(ii)Data-based: Data-based methods can detect virus before they are executed; they utilize the binary data extracted from the program files. The traditional methods extract signatures from virus samples scanners and compare these signatures with unclassified files to determine whether they are virus or not. These methods were effective in the past. Nowadays, Malware programmers have gone miles ahead of this solution. Many sophisticated malware are being designed by the days and cannot be captured or identified by the signature based principle.Moreso, malware writters now use hybridization encryptions to encrypt their codes or make the viral codes invisible for detection.

(iii)Virtual Memory Machine: With the emergence of unknown malware known as zero-day malware that uses the sophisticated disguise methods such as obfuscation, polymorphic and metamorphic designed processes, the Virtual Memory Machine techniques is the common technique that is being used to detect these embedded malware techniques using emulators. The Virtual Machine uses the emulating processes to expose the decrypted malware as the encrypted malicious code decrypts and executes sequentially in the sandbox

3.1.1 Drawbacks of the Current Virus Detection System

The problems faced with the current malware detecting algorithms are outlined below:

(i) Behavior Based: The drawback of these methods is that it is not cost effective and it has much strenuous effort to build a less-error sandbox; also the sandbox cannot be simulated on the same environment as does real operating system. Although these approaches have produced promising results, they can produce high rates of false positive (a process that rejects a benign output while it is supposed to be accepted), an issue which has yet to be resolved.

(ii) Database Method*:* This technology cannot detect more advanced and sophisticated codes such as polymorphic and metamorphic codes. The encryptedoligamorphic malware are thosemalware that are encrypted and could download their overheaddecryptor headers after a while. The encrypted codes of oligamorphic malwarecan morph themselves. Database signature antivirus can only dictate known previous based codes that are stored in a given antivirus server database. To meet the new development stage design, the antivirus has to be retrained regularly to capture the new decryptors and incorporate them into its database.

(iii)Virtual Memory Machine: This technique is not practical for high level language written viruses. Also, the emulation-based virus detection is considered as quite slow when it analyzes on the virtual platform. Despite this new development on the platform of virtual machines as Intrusion Detection System (IDS), malware detectors still are able to develop efficient programs that can evade detection on such sandboxes.

The front-end of the proposed system would be designed using VB.net programming language, while the back-end and the file system component would be design using C++.

VB.net was chosen for the design and implementation of some of the components of the antivirus software because an Anti-virus program needs to execute much closer to the Operating System which makes it the best language to get something productive. The front-end of the application (the GUI that the user utilizes to configure scans and view report) is achieved using visual basic programming language. The back-end virus scanning service is implemented using C language. It was chosen due its functions needed to achieve basic Anti-virus task. The language also has built-in classes and functions that enable an actual scanning of the operating system files.

3.2Artificial Immune System Architecture Algorithm

Due to space constraint, we do not present the design of the architecture in deatil; but instead, we present some few details meeting the missing abstraction of our model through the algorithm as yielded in the italics:

The virus detection algorithms as used by the artificial immune system are outlined below and shown in italic:

Scan the System files

Load Virus Signatures from the database

Compare file content with signature

If signature is contained in file goto 5 else goto 6

Show Virus Alert Message

Goto step 1

End

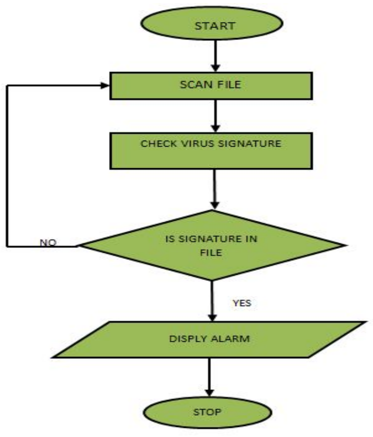


Fig 3.1 Flowchart for the virus detection system

Figure 3.1 is the abstractive deduction of the structural algorithm givenin a flow chart structure:

Fig 3.2 belowshows the various menus of the entire system, it gives a clear view of the various menus located in the various components of the software that is presented as a tree graph; the main window, as seen in fig 3.2, is the first interface that is presented to the user when the our developed antirus is started. Other menus comprises of the sub-sections which also lists other options to the user.

Fig 3.3 below depicts how user can interact and use the various component of the developed antivirus software. The use-case also confirms how a particular aspect of the software is extended to the other various aspects of it. As shown in the same figure 3.3, the user scans component using the SecurePkus antivirus that links to other operation that can be performed under the scan menu.

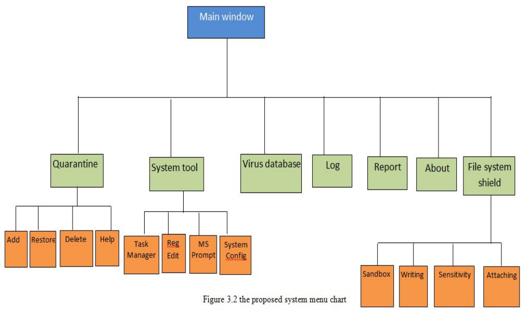


Figure 3.2 the proposed system menu chart

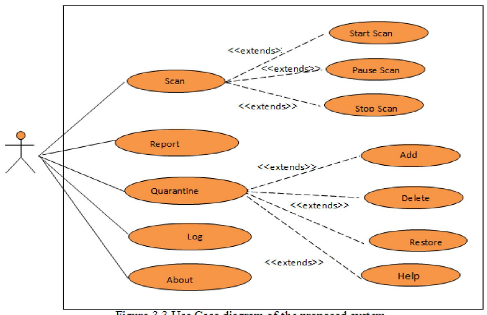


Figure 3.3 Use Case diagram of the proposed system

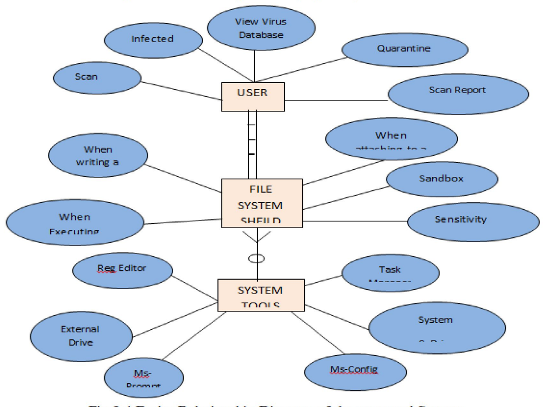


Fig 3.4 Entity Relationship Diagram of the proposed System

3.3Virus FileTable Structure:

Table 3.1 shows the database information about the file affected by Virus, its name, location and details. This database file will contain the information about the virus details.

Table 3.1 Virus Files Table Structure

S/NO	FIELD NAME	DESCRIPTION	FIELD SIZE	DATA TYPE
1.	ID	Unique identification number	30	Number
2.	Virus	Information of Virus	50	Text

Table 3.2 shows the information about the files in the quarantine cubicle. The information, as viewed in the same table that includes the name of the file, and the details of the file. This database file will contain the information about the files in quarantine

Table 3.2 Quarantine Files Table Structure

S/NO	FIELD NAME	DESCRIPTION	FIELD SIZE	DATA TYPE
1.	ID	Unique identification number for the contact	30	Integer
2.	Name	Name of the virus	50	Text
3.	File Extension	File extension of the infected file.	50	Text
4.	Type	Type of the virus	50	Text
5.	Location	File location of the infected file	50	Text

3.4 Choice of programming language

VB.net was chosen for the design and implementation of some of the components of the antivirus software due to its simplicity and robust built-in functions that provides the capability of executing commands much closer to the Operating System, especially in the aspect of reliable virus scanning. The front-end of the application (the GUI that the user utilizes to configure scans and view report) is achieved using visual basic programming language. The back-end virus scanning service is implemented using C language. It was chosen due its simplicity and functions needed to achieve basic Anti-virus task.

3.4.1 Requirement of the Existing System

In computing, it is believed that for an application to run on a system effectively, it needs minimum software and hardware component that it will require in order for the system to run efficiently. The software running can be hindered in the present malware which can adversely affect this efficiency.

3.4.2 Software Requirement

Software are applications that runs on a system, they can only be seen but not cannot be touched. The software requirements for this application are

1. Windows XP Operating System or Higher Microsoft Windows OS

2. Visual Basic 6 Runtime Components

3. Microsoft Access

3.4.3 Hardware Requirement

Hardware, in computer world refers to the physical components that make up a computer system, they are usually connected together to create a complete system.

The minimum hardware requirements for the system are:

1. Screen Resolution Recommended (1280 X 800) Minimum of (1280 X 270)

2. 256MB Random Access Memory (RAM Size)

3. 128 MB Hard Disk Drive

4. Pentium 3 Central Process Units

EXPERIMENTAL OUTPUT

Experimental output or implementation is a realization of a technical specification of algorithm as a program, software component, or other computer system through computer programming and deployment. This sectionis targeted towards the implementation of Artificial Immune System for the development of Virus Detection and disinfection as planned in section 3.

The implementation of the Artificial Immune System for script virus detection and elimination was done through the utilization of both the Microsoft Visual Basic Rapid Application Development tool and the Microsoft Access Database System. Visual Basic was used to implement the programming logic of the system while the Access database system provided the auxiliary storage unit for the information about the system.

THE EXPERIMENTAL RESULTS

In this section, the Artificial Immune System for script virus detection and elimination user interfaces are presented along with the specifications and requirements necessary to run the system in its intended area of application. The analysis of data not considered as our concern is to develop an antivirus that works. All our testing are presented dialogues figures

4.1 Testing the New System

System testing is the testing of a complete and fully integrated software product. Usually software is only one element of a larger computer based system. Ultimately, software is interfaced with other software/hardware systems. System testing involves a series of different tests whose sole purpose is to Due to space constraint and the desire to show the all captions images and texts fonts completely, we cannot display the visualizations of all diagrams as experimentedfor the implementation of SecurePlus antivirus. However, Fig 4.2 is the interface used to view the reports on the logging information performed by the developed SecurePlus. It displays details of the encrypted virus in hexadecimal. The interface, Fig 4.2, is used to view the signatures contained in the system's database.

Fig 4.3 below allows the user to perform some operations on system tasks such as launching the task manager, loading the command prompt, loading the Reg Edit tool and checking the system configuration.

exercise the full computer based system.During the testing of the proposed system, the following were targeted:

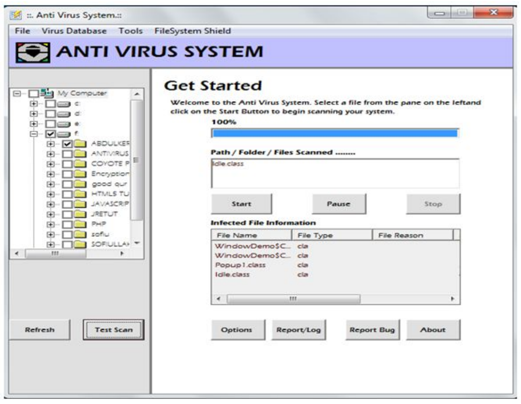
1. The fully integrated software applications including the external computer peripherals devices, were tested in order to check how components interact with one another and with the system as a whole (This is also called End to End scenario testing)

2. Verification through thorough testing of every input in the application to check for desired outputs.

3. Testing of the user's experience with the proposed system application.

4. Building of detailed test cases and test suites that test each aspect of the application as seen from the outside without looking at the actual source code.

System Screenshots:



Welcome to the Anti Virus System. Select a file from the pane on the leftand click on the Start Button to begin scanning your system.

Path / Folder / Files Scanned ....

idle.class

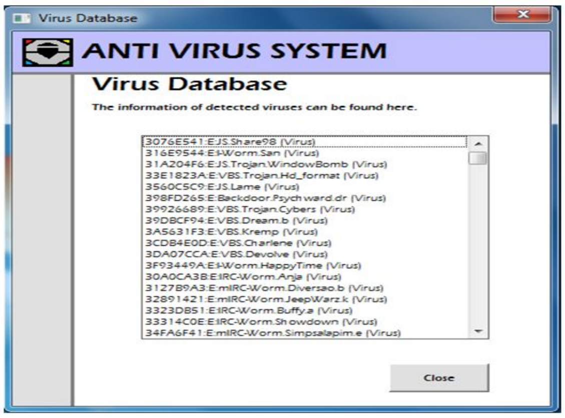
Infected File Information

File Name	File Type	File Reason
WindowDemo$C...	cla	
WindowDemo$C...	cla	
Popup1.class	cla	
Idle.class	cla	

Figure 4.1 Main Window

Sandbox Window: Fig 4.4, gives user the privilege to activate the sandbox feature within the application. By enabling the sandbox feature, i.e. choosing the option "enable auto sandbox", the software is able to simulate the behavior of the file contained in the storage devices in order to examine their behavior before attacking the system. The interface of SecurePlus in Fig 4.5 enables the user to specify the type of files to be scanned at the moment they are run or executed. It is recommended that the user checks all the boxes for maximum security. Checking the boxes "scan programs when executing" and "scan scripts when executing" will ensure that all programs and scripts will be scanned at the moment they are run to ensure they are clean and will not cause any damages to your computer or your data. If the box "Scan libraries (DLLs) when loading" is checked, all DLL files will be scanned when they are loaded. This may result in some applications starting more slowly, but SecurePlus will significantly increase the security of the system.

SecurePlus can also test the Sensitivity of the Window displays in Fig 4.6 below; one can adjust the basic sensitivity, which determines how thoroughly files are scanned, and also the heuristic sensitivity is exhibited:

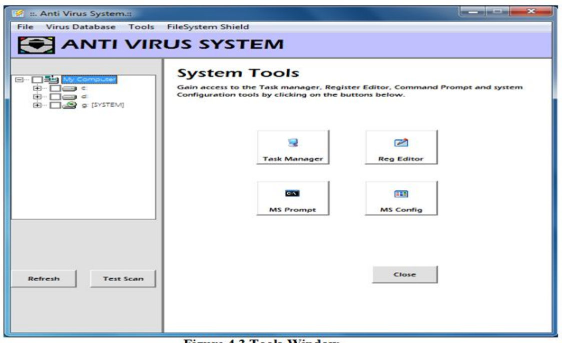


Virus Database

The information of detected viruses can be found here.

```
3076E541:E:JS.Share98 (Virus)
316E9544:E:I-Worm.San (Virus)
31A204F6:E:JS.Trojan.WindowBomb (Virus)
33E1823A:E:VBS.Trojan.Hd_format (Virus)
3560C5C9:E:JS.Lame (Virus)
398FD265:E:Backdoor.Psych ward.dr (Virus)
39926689:E:VBS.Trojan.Cybers (Virus)
39DBCF94:E:VBS.Dream.b (Virus)
3A5631F3:E:VBS.Kremp (Virus)
3CDB4E0D:E:VBS.Charlene (Virus)
3DA07CCA:E:VBS.Devolve (Virus)
3F93449A:E:I-Worm.HappyTime (Virus)
30A0CA3B:E:IRC-Worm.Anja (Virus)
312789A3:E:mIRC-Worm.Diversao.b (Virus)
3289142 1:E:mIRC-Worm.JeepWarz.k (Virus)
3323DB51:E:IRC-Worm.Buffy.a (Virus)
33314C0E:E:IRC-Worm.Showdown (Virus)
34FA6F41:E:mIRC-Worm.Simpsalapim.e (Virus)
```

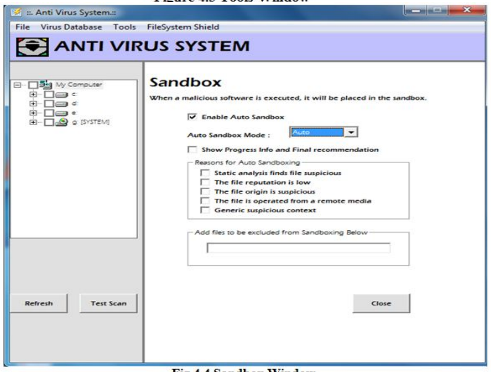
Figure 4.2 Virus signatures Database Window.



System Tools

Gain access to the Task manager, Register Editor, Command Prompt and system Configuration tools by clicking on the buttons below.

Figure 4.3 Tools Window



Sandbox

When a malicious software is executed, it will be placed in the sandbox.

☑ Enable Auto Sandbox

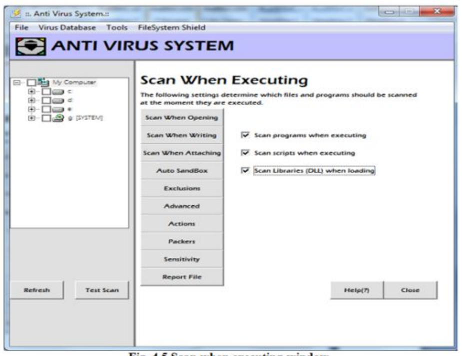
Auto Sandbox Mode : Auto

☐ Show Progress Info and Final recommendation

Reasons for Auto Sandboxing
- ☐ Static analysis finds file suspicious
- ☐ The file reputation is low
- ☐ The file origin is suspicious
- ☐ The file is operated from a remote media
- ☐ Generic suspicious context

Add files to be excluded from Sandboxing Below

Fig 4.4 Sandbox Window

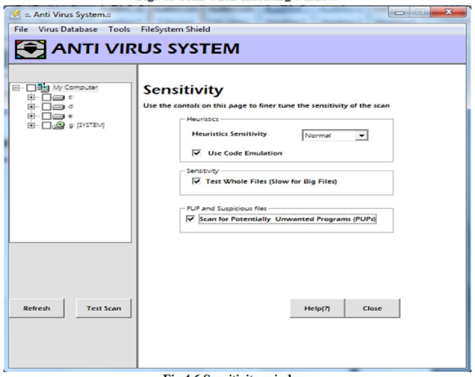


Scan When Executing

The following settings determine which files and programs should be scanned at the moment they are executed.

- ☑ Scan programs when executing
- ☑ Scan scripts when executing
- ☑ Scan Libraries (DLL) when loading

Fig. 4.5 Scan when executing window



Sensitivity

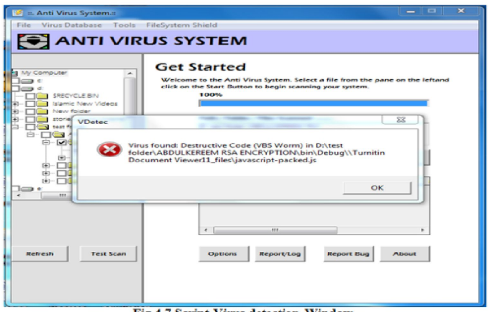
Use the contols on this page to finer tune the sensitivity of the scan

Heuristics
Heuristics Sensitivity: Normal
☑ Use Code Emulation

Sensitivity
☑ Test Whole Files (Slow for Big Files)

PUP and Suspicious files
☑ Scan for Potentially Unwanted Programs (PUPs)

Fig 4.6 Sensitivity window



Get Started

Welcome to the Anti Virus System. Select a file from the pane on the leftand click on the Start Button to begin scanning your system.

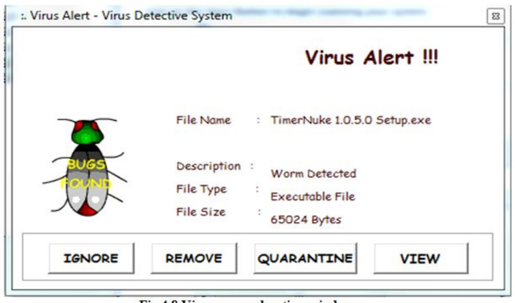
100%

Virus found: Destructive Code (VBS Worm) in D:\test folder\ABDULKEREEM RSA ENCRYPTION\bin\Debug\\Turnitin Document Viewer11_files\javascript-packed.js

Fig 4.7 Script Virus detection Window

In fig 4.7, if a script virus is detected, the system would presents the user with a new window interface within the original window that shows the type of virus and the file

Fig 4.8 presents users with options for the script virus such as ignore, remove, quarantine and view. Here, the usefulness of the ignore option is that if the user observed that the detected virus is not a real virus, it can be ignored easily affected by the script virus. In fig 4.7, a script virus is detected on a java script file, i.e. ".js file".


:. Virus Alert - Virus Detective System

Virus Alert !!!

File Name : TimerNuke 1.0.5.0 Setup.exe

Description : Worm Detected

File Type : Executable File

File Size : 65024 Bytes

Fig 4.8 Virus removal option window

UMMARIES AND CONCLUSION

Artificial immune systems draw inspiration from the natural immune system in order or solve computer security problems. We have in this project, applied some Artificial immune system models to solve virus detection problem**.** We have described and analyzed already existing computer systems designed with inspiration from the biological immune system. We have also reviewed several literatures related to the area of Artificial Immune System for virus detection and elimination.

Ultimately,we applied some of the models to solve virus detection and elimination too.

RECOMMENDATIONS

Malwares cannot be totally stopped, but it can be reduced to the barest minimum, therefore it is necessary to make recommendations that will guide computer users on the effect these malwares. We therefore provide some recommendation to web users, home users and for further research.

6.1 Recommendations to web users

(i) Never click on links or attachment in an email that you are not sure whether is from a trusted source. If you think the e-mail looks suspicious. It probably is. It never hurts to send an email to verify that it is legitimate.

(ii) If you use an e-mail retrieving program, disable image previews. Email applications like outlook, Thunderbird, and others often automatically load attachments to your convenience, but this takes away you ability to decide whether or not file is safe to open. Check your references to disable these settings.

(iii) Be wary of files with double extensions such as ".txt.vb" or"jpg.exe" as a default setting windows, windows often hides some common file extensions, meaning that a program like plaintext.exe will appear to you as simply paint. Double extension exploit this by hiding the second, dangerous extension and reassuring you with the first, safe extension- which is utterly meaningless to your computer; your system only recognizes the extension to the extreme right and run the file as sch.

6.2 Recommendations to home users

(i) Install consistent and effective Anti-virus software on your computer system. By Installing an Antivirus program and keeping it up to date can assist protect your computer against viruses.

(ii)Turn on user account control (UAC), this is especially important when changes are to be made to your computer system that require higher-level permission, UAC inform users and gives them the privilege to accept any change that takes place on the computer system.

(iii)Before you explore or open any flash drive on your computer, scan it first.

6.3 Recommendations for future research

The analysis discussed on this project shows how to come up with software that is capable of detecting and disinfecting script viruses from storage devices. However, in view of the urgent need for current and updated script virus detection and elimination, we recommend that a similar software but with a more capability, such as updating database signature automatically be developed. Such software should also have the capability of examining files before they are executing such as the use of a better sandbox.

CONCLUSION

From the inspiration and techniques from the biological immune system and already published papers on the area of natural immune system, we have designed an application for script virus detection and elimination. Just as the biological immune system's capability of differentiating between self and non-self, we have also applied those features to come up with this application. Ultimately, the software was able to detect and remove script virus from storage devices.

REFERENCES

[1]. Abou-Assaleh T., N. Cercone, V. Keˇselj, and R. Sweidan. (2004). "Ngram-based Detection of New Malicious Code."In Proceedings ofthe 24th Annual International Computer Software and ApplicationsConference (COMPSAC 2004), Hong Kong.

[2]. Aickelin, U., Bentley, P., Cayzer, S., Kim, J., and McLeod, J., (2003). Danger theory: "The link between AIS and IDS". In Proc. of the Second International Conference on Artificial

Immune Systems (ICARIS-03), pages 147–155,

[3]. Al-Enezi .R.,Abbod M.F. Alsharhan S., 2010. Artificial Immune Systems – Models, Algorithms And Applications, IJRRAS 3 (2) , May 2010, pp118-131

[4]. BITS (2011) 'Malware Risks and Mitigation Report' ITS/The Financial Services Roundtable 2011,

[5]. Burnet F.M., 1959. The Clonal Selection, Theory of Acquired Immunity. Cambridge University Press

[6]. Carrera,E. &Erdelyi. (2004). 'Digital genome mapping: Advanced binary malware analysis', Virus, Bulletin, pp. 175-186.

[7]. Castro, D., L., Timmis, J., Artificial Immune Systems: A New Computational Intelligence Approach. Springer, London (2002).Member IEEE.

[8]. Castro L. de and Zuben F., 1999. Artificial Immune Systems: Part I – Basic Theory and Applications. TR – DCA 01/99.

[9]. Cutello V., Narzisi G., Nicosia G., and Pavone M., 2005.An Immunological Algorithm for Global Numerical Optimization. Artificial Evolution: 7th Int. Conference, Evolution Artificielle, EA 2005, October 26-28, Lille, France, Springer, LNCS 3871:284-295.

[10]. Daniel Bilar, Intro To Malware, CS342 slides, Oct. 6, 2006

[11]. Dasgupta, D., and Michalewicz, Z., (1997) Evolutionary algorithms in engineeringapplications. New York: Springer-Verlag, Inc.,

[12]. Dabrowski J. and Kubale M., 2008.Computer Experiments with a Parallel Clonal Selection Algorithm for the Graph Coloring Problem.IEEE International Symposium on Parallel and Distributed Processing (IPDPS 2008), 14-18 April, Miami, FL, USA, pp.1-6.

[13]. Dasgupta D., 1997. Artificial neural networks and artificial immune systems: Similarities and differences. In IEEE International Conference on Systems, Man and Cybernetics, Orlando, FL, pp. 873–878.

[14]. Ed Skoudis, Malware: Fighting Malicious Code. Pearson Edcation, 2004.

[15]. Garrett S., 2004. Parameter-Free Adaptive Clonal Selection. Congress on Evolutionary Computation (CEC2004), 19-23 June, Volume: 1, pp.: 1052- 1058.

[16]. Gong M., Jiao L., Zhang L., and Ma W., 2007a. Improved Real-Valued Clonal Selection Algorithm Based On A Novel Mutation Method.Proceedings of 2007 International Symposium on Intelligent Signal Processing and Communication Systems Nov.28-Dec.1, Xiamen, China.

[17]. Gong M., Zhang L., Jiao L. and Ma W., 2007. Differential Immune Clonal Selection Algorithm.Proceedings of 2007 International Symposium on Intelligent Signal Processing and Communication Systems Nov.28-Dec.1, Xiamen, China.

[18]. Harmer, P.K., Williams, P.D., Gunsch, G.H. and Lamont, G.B. (2002), "An artificial immune system architecture for computer security applications", IEEE Transactions on Evolutionary Computation, Vol. 6 No. 3, pp. 252-80

[19]. Hofmeyr, S. A., (19990: "An immunological model of distributed detection and its application to computer security," Ph.D. dissertation, Univ. New Mexico, Albuquerque, NM, 1999.

[20]. Hu J., Guo C., Li T. and Bu R., 2007.A Mutation-Classified, Parameter-Dynamic Immunological Algorithm for Global Optimization. Proceedings of the 2007 American Control Conference, NY, USA, July 11-13, page(s): 546-551.

[21]. Jiao L. and Li Y., 2005.Quantum-Inspired Immune Clonal Optimization.The 2005 IEEE International Conference on Neural Networks and Brain, 13-15 Oct., pp. 461--466.

[22]. Lu H. and Zhichun M., 2008.A Clonal Chaos Adjustment Algorithm for Multi-modal Function Optimization.Proceedings of the 27th Chinese Control Conference, July 16-18, Kunming, Yunnan, China.

[23]. Frederick B. Cohen, A Short Course on Computer Viruses. John Wiley, 1994.

[24]. John Viega, The Myths of Computer Security: What the Computer Security Industry Doesn't Want you to Know. O'Reilly Media, 2009.

[25]. Islam, N.,Anand, R., Jaeger, T.,&Rao, J.R. (2009). 'A flexible security system for using Internet content',Software, IEEE, Vol. 14, No.5, pp. 52,59. Retrieved from

[26]. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=605931&isnumber=13290

[27]. John Viega,"Why Anti-Virus Sucks, and How to Fix It", Talk to Harvard's Center for Research on Computation and Society (CRCS), Wed. Oct. 8, 2008 (see video of talk at http://crcs.seas.harvard.edu/2008/09/24/wednesday-october-8-2008-john-viega-on-tbd/

[28]. Jung, W., K., (2002):Integrating Artificial Immune Algorithms for Intrusion Detection, Department of Computer Science, University College London, July 30, 2002

[29]. Matching rule," 1st International Conference on Artificial Immune Systems (ICARIS) (J. Timmis and P. J. Bentley, eds.), (Canterbury, UK), pp. 99 106, University of Kent at Canterbury Printing Unit, Sep.

[30]. Peter Szor, The Art of Computer Virus Research and Defense. Addison Wesley, 2005.

[31]. Paul, K., H., Paul, D., Williams, G., H., Gunsch, and Gary B. L., (2002): "An Artificial Immune System Architecture for Computer Security Applications". IEEE Transactions on Evolutionary Computation, Vol. 6, No. 3, June 2002

[32]. Rui C., and Ying T., (n.a): "A Virus Detection System Based on Artificial Immune System". Member IEEE

[33]. Qiao P., Wang T. and Su J., 2008.An Improved Clone Selection Immune Algorithm. Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security, edited by Belur V. Dasarathy, Proc. of SPIE Vol. 6973, 69730O, (2008) , DOI: 10.1117/12.772464.

[34]. Qiao Y. and Jianping Y., 2006. AINIDS: An Immune-Based Network Intrusion Detection System. Proc. of SPIE vol. 6241, 62410U, April 18

[35]. Sean Smith and John Marchesini, The Craft of System Security, Chapter 6: Implementation Security.

[36]. Singh, S.,(2002) "Anomaly detection using negative selection based on the r-contiguous

[37]. Ruochen L., Haifeng D. and Licheng J., 2003.Immunity Clonal Strategies. Proceedings of Fifth International Conference on Computational Intelligence and Multimedia applications (ICCIMA'03), 27-30 Sept. , pp. 290- 295.

[38]. Somayaji A., Hofmeyr S., and Forrest S., 1997.Principles of a Computer Immune System. In: Proceedings of the Second New Security Paradigms Workshop, pp. 75–82.

[39]. Symantec Antivirus Remote Stack Buffer Overflow Vulnerability

[40]. http://www.securityfocus.com/bid/18107/references

[41]. Timmis J.,, Hone, A. , Stibor, T, Clark E., (2008), Theoretical advances in artificial immune systems, www.elsevier.com/locate/tcs

[42]. Uwe, A., Julie G.,S., Jamie, T., (n.a): "Immune System Approaches to Intrusion Detection - A Review". School of Computer Science, University of Nottingham, UK Xiu-Li P. and Yu-Qiang F., 2006. Solving Competitive Facilities Location Problem with the Clonal Selection Algorithm", International Conference on Management Science and Engineering (ICMSE apos;06), Volume , Issue , 5-7 Oct., Page(s):413 – 417.

[43]. http://www.biology.arizona.edu/immunology/tutorials/immunology/page2.html

[44]. Yang J., Sun L., Lee H., Qian Y. and Liang Y., 2008. Clonal Selection Based Memetic Algorithm for Job Shop Scheduling Problems. Journal of Bionic Engineering, vol. 5, pp. 111-119.

[45]. Zhenhe, G., Zhengkai, L., Ying, T., Ling Z., (2006) "An NN based Malicious Executables Detection Algorithm Based on Immune Principles", pp.5-6.

[46]. Zuo X. and Li S., 2003.The Chaos Artificial Immune Algorithm and Its Application to RBF Neuro-Fuzzy Controller Design. IEEE International Conference on Systems, Man and Cybernetics, Volume 3, Issue , 5-8 Oct., Page(s): 2809 - 2814. Cheering

BIBLIOGRAPHY

	Dr. Victor OnomzaWaziri is an Associate Professor in the Department of Cyber Security Science, Federal University of Technology, Minna-Nigeria. He was the Pioneer Head of Cyber Security Science from 2009-2014; until in July 2014. He acquired his PhD in Applied Mathematics in the area of Computational Optimization in Wave diffusion Equations; 2004, from the Federal University of Technology, Minna-Nigeria. He has a Postdoctoral Certificate in Computer Science from the University of Zululand in South Africa; 2007. Other areas of his researches include Modern Cryptography, Data Mining and Machine Learning that involves Intelligent Soft Computing, Network Security; Malware Detection with concern in zero-day Malware, General Cyber Security Science and Big Data Analytics with indepth focus on Fully Homomorphic Encryption Schemes. In most cases, Matlab, Maple and Mathematica are the bases for his accessory in modeling and Simulations in Modern Cryptographic analyses. He has published many papers in reputable Journals at both International and Local Scenes. He Lectures various courses in the Department of Cyber Security Science that include Cryptography, Network Security, Clouds Security, Data Mining, Computational Theory, Automata and Programming Languages
	Dr. J. K. Alhassan was born at Ganmu-Alhaeri, in Kwara State, Nigeria on 9th January, 1974 and obtained Bachelor of Technology in Mathematics/Computer Science, at Federal University of Technology, Minna, Niger State, Nigeria in 2000. Then Master of Science in Computer Science, at University of Ibadan, Nigeria in 2006, and Doctor of Philosophy in Computer Science, at Federal University of Technology, Minna, Niger State, Nigeria in 2014. The major field of study is computer science. He carried out part of his PhD research at United Institute of Informatics Problems, National Academy of Sciences of Belarus (UIIP NASB) Minsk, Republic of Belarus. He is currently the Ag. Head, at the Department of Cyber Security Science, Federal University of Technology, Minna, Niger State, Nigeria. He has published twelve journal articles and four conference proceedings. His research interest includes Artificial Intelligence, Data Mining, Internet Technology, Database Management System, Software Architecture, Machine Learning, Human Computer Interaction and Computer Security. Dr. Alhassan is a member of Computer Professionals Registration Council of Nigeria (CPN).
	Dr. IsmailaIdris is with the Deparment of Cyber Security Science. He obtain his Bachelor degree with Federal University of Technology, Minna. M.Sc. with university of Ilorin and PhD degree with University of Teknologi Malaysia. His research interest are Information Security, Data Mining, Machine Learning, Evolutionary Algorithm.
	AbdulkadirAdamu was born 16th dec, 1988 Institution: Federal University of Technology Minna, State of Origin: Niger L.G.A: Bida Department:Cyber Security Science Phone Number:08066337186, Year Of Graduation: 2013 Qualification: B.tech Cyber Sec