Computer Immunology

by

Lyanne Wai-Yin Lau

Supervised by
Daniel Bradley

A thesis submitted to
The Department of Information Technology and Electrical Engineering
The University of Queensland

for the degree of

Bachelor of Information Technology (Hons)

October 2002

Declaration

I declare that the content of this thesis is my original work with the exclusion of references that have been appropriately acknowledged. In addition, this work has not been previously submitted for a degree at the University of Queensland or any other institutions.

Lyanne Wai-Yin Lau

Acknowledgements

I would like to thank my supervisor Daniel Bradley for his time and guidance throughout the year. As well as those who took the time to review previous drafts of this work.

Finally, many thanks to friends and family who have provided enormous support and encouragement during the course of the compilation of the thesis.

Abstract

Due to the vulnerability of existing computer systems, extensive research has been performed in Computer Immunology. Consequently, a radical method of securing the systems has been devised, namely the artificial immune system. This system, which is modeled on the biological immune system, is self-sufficient and minimizes the need for human intervention. Not only will the system be able to deal with computer viruses, it also detects the state of the system and configures programs if need be. To achieve this, efforts have been directed to studying how to define self and non-self, distributed detection, fault tolerance to attacks, responsive systems, and varied methods of implementation. This paper presents a summary of the published works in the domain of Computer Immunology.

Table of Contents

Abstract iii

List of Figures vi

List of Tables vi

1 Introduction 1

2 Background 2

2.1 Problems of current techniques 2

3 Basis of theory 4

3.1 Autonomy 4

3.2 Self and Non-Self 4

3.3 Multi-layered 5

3.4 Imperfect detection 5

3.5 Dynamically changing coverage 6

3.6 Distributability 7

4 Intrusion Detection System 8

4.1 Self and Non-self 8

4.1.1 Generation of the image of self and its Detectors 8

4.2 Detection Algorithm 9

4.2.1 String Detection 10

4.2.2 System Calls 13

4.2.2.1 Enumeration 13

4.2.2.2 Frequency-based 13

4.2.2.3 Data mining 14

4.2.3 Detectors with lifecycles 14

5 Distributed detection 16

5.1 Bayesian Methods 16

6 Multi-layered protection 18

6.1 Ineffectiveness of Firewalls 18

7 Taxonomy of Intrusions 19

8 Elimination 21

9 Future Directions 21

10 Conclusion 22

Bibliography 23

List of Figures

Figure 1 Representation of Self and Non-self ____ *8*

Figure 2 Generation of negative detectors ____ *9*

Figure 3 Matching with contiguous rule ____ *10*

Figure 4 Generation of Detectors ____ *12*

Figure 5 Monitoring of protected data ____ *12*

Figure 6 Lifecycle of a detector ____ *15*

List of Tables

Table 1 Computer misuse techniques. ____ *20*

1 Introduction

Computers, which are involved in every aspect of modern society, have become an essential part of our professional, social and economical lives, but their vulnerability is of increasing concern. Security flaws are commonplace in the computing world and the healing process almost inevitably involves human intervention. Although most flaws are caused by errors in the process of software engineering or unforeseeable mishaps, it is difficult to solve these problems by conventional methods such as formal verification. A radical way of constantly monitoring the system for newly disclosed vulnerabilities is required. The proposed solution to this is a computer that is self-sufficient. Not only will the system be able to correct itself from unknown problems, but it also ensures that the system is appropriately configured to the environment it operates in.

In order to devise such a system, many researchers have, for many years, tried to draw an analogy between computers and biological systems. They see that the biological equivalent for a similar task is the immune system and thus suggests the artificial immune system. An artificial immune system is a computer system that, modelled on the human body, is capable of monitoring its own state of health. Even though differences exist between it and its biological counterpart, the similarities that are present is compelling.

Such a system would require current operating systems to incorporate added protection scheme such as a stable definition of self, a scheme to detect and prevent dangerous behaviour, a method of eliminating disclosed intrusions, and the ability to remember information regarding new intruders to the system. Extensive research has been performed in this domain, focusing mainly in the area of intrusion detection systems and distributed detection, but it also touches upon the classification of different types of security concerns, methods of elimination, multi-layered protection, and system feedback.

The primary objective of an artificial immune system is to collect information about events occurring in a computer system, such as a viral attack, and then act accordingly. Other events of interests include those that violate a predefined security policy, such as attempts to affect the confidentiality, integrity or availability of a computer network.

2 Background

In the past few decades, computers have evolved from being non-existent to becoming a necessity in our daily lives. As they grow from simple single systems to current worldwide-networked systems, the importance of computer security has received more and more attention. The emergence of new viruses, coupled with the need to maintain a stable system, have become a major concern in the security paradigm. In order to solve this problem, researchers have proposed the adoption of a method that mimics the way the human body protects itself from external entities. Thus, we witness the emergence of artificial immune systems, i.e. computer systems that can survive on their own.

The majority of computer software nowadays is capable of self-diagnosis and repair, but it is restricted to problems that are known prior to the release of the software. Thus, frequent updates of management software are necessary. The ideal method of handling faults was first proposed by Kephart. Although it was initially intended for only virus detection and removal, the artificial immune system today extends protection to 'cleaning up waste products, repairing damage, security through checking and redundancy' [4], as well as other forms of intrusions.

As early as 1987, the relationship between computers and biological processes was made, starting with the term 'viruses' used by Adelman [10]. Later, Spafford made a similar comment by claiming that computer viruses are a form of artificial life [39]. Other authors have also investigated the analogy between epidemiology and the spread of computer viruses across networks [23].

2.1 Problems of current techniques

Currently, computer security problems are handled through human intervention although the degree of intervention varies with the type of problem that arises. In terms of a system management problem, there are management tools available in the market, such as Tivoli, OpenView, and Solstice and Host Factory [4]. However, these software packages are targeted at the more experienced users and require extensive knowledge of the system. Users of less experience could unintentionally lower the software's ability to protect the system or even fail to notice an error [4].

In case of viral attacks, they are commonly picked up by virus scanners and destroyed. However, this is only limited to known viruses and minor variants of them. In terms of previously unknown viruses, they are handled in a much more complicated manner. First of all, a sample of the newly disclosed virus is sent to an international group of virus collectors, such as anti-viral software vendors, who then analyse the structure and behaviour of the virus, and define a signature for identification purposes. Subsequently, virus collectors will put together a patch and release it to the public. Finally, the anti-virus software will then be updated adding the signature and the patch to their database.

Other anti-virus techniques include activity monitors and integrity management systems [21]. Activity monitors are programs that monitor system activities and alert users to activities that are commonly associated with viruses. Integrity management systems on the other hand, monitor the system for suspicious changes to files. The two techniques mentioned above are capable of detecting new methods of attack but are incapable of accurately identifying the location and nature of the attacker. Therefore, each of them alone is insufficient for an artificial immune system but could be used as a component of it.

3 Basis of theory

One of the key factors that human beings have been able to survive evolution is the amazing ability of the human body's immune system. It efficiently distinguishes intruders from the diverse range of normal cells types in the body and consequently eliminating the intruders. Being an adaptive system, it is capable of expanding its inborn capabilities through experience. This ability to recognise and remember unknown foreigners is well worth studying.

3.1 Autonomy

In the present state, computers are incapable of self-maintenance. With the enormous number of errors that could possibly occur and the speed at which new viruses are introduced, system administrators and other computing professionals find maintaining a healthy system increasingly difficult. Nature, on the other hand, has the capability of self-healing which is comparable and is greater in complexity [4].

An autonomous system is one that is designed with flexibility and is able to adapt to the ever-changing world. Potentially, an artificial immune system will be able to recognise known intruders, eliminate them, as well as learn about unknown intruders. The learning process consists of determining what is and what is not an intruder, figuring out how to recognise and eliminate it, and remembering how to recognise it for future use [21].

3.2 Self and Non-Self

For a successful intrusion detection system, its ability to identify processes between self and non-self is essential. Traditionally, self is defined as the internal cells and molecules of the body, whereas non-self is considered as all elements that do not belong to 'self', i.e. all foreign substances including viruses and bacteria. In the biological system, there exist foreign organism detectors known as antigens. These, when bound with other cells, initiate an immune response to acknowledge lymphocytes to destroy the bounded foreign cells.

The concept of identifying self from non-self can be applied in the context of computer security but its implementation in an artificial system is far more complex than in its biological counterpart. Although many solutions to the design of an intrusion detection system can be found in nature, the solution to this particular problem has not yet been found. The major difficulty lies in the different structures of 'self' in the two systems. While all biological systems are constructed by proteins and remain virtually unchanged throughout its lifetime, the structure of a computer system is constantly changing and both self and non-self constitutes of 1s and 0s.

To solve this problem, two methods of implementing the self and non-self detection has been introduced [11]:

1. Misuse intrusion detection and anomaly intrusion detection
2. Misuse intrusion detection, through known patterns of intrusion.

3.3 Multi-layered

Similar to the human body, an artificial immune system would benefit from the use of a multi-layered protection scheme. Nature's different layers of protection include the skin, physiological barriers, the innate immune system, the adaptive immune responses, and the excretion system.

Being the outermost layer, the skin is able to effectively block water and water-soluble substances from both leaving and entering the body, thus limiting the types of substances that can penetrate the skin. The second line of defence is the physiological barriers that maintain the body at a certain pH level and temperature, within homeostatic limits, that is not habitable for some foreign organisms. The third level of protection is the innate immune system and the adaptive immune system. They ward off harmful and disease-causing organisms, and stalk and eliminate particular foreign substances. The last layer of protection is the body's excretion system. The importance of this component cannot be overlooked due to it ability to cleanse the body of waste products and unwanted substances which would suffocate the body's cells [4].

The existing equivalent to the skin layer in the digital world is the firewall. Acting as the first line of defence, it filters unwanted data and admits only relevant information. The next line of defence that is widely available currently is the innate immune system, as represented by anti-viral software. By referring to a database that contains the definitions and solutions for all the known viruses, anti-virus software is able to track down the files which are infected and eliminate the intruders. Finally, waste management, a component of computer management systems that parallels the biological excretion system, scanning for errors that might exist in the data.

3.4 Imperfect detection

The principle of imperfect detection is commonly practised in nature. The biological immune systems' chief detectors — lymphocytes — not only recognise and attack specific invaders such as bacteria and viruses, they also detect normal cells that are defected, either by viruses or mutations. These lymphocytes have the ability to not only identify specific foreigners, but also foreigners unknown to the body. As a result, an artificial immune system requires a less specific recognition system.

In nature, antigens, a type of signalling mechanisms, have the ability to detect more than one type of foreign organisms. Thus, not all antigens are perfectly matched by a pre-existing detector in nature. This is also the most preferable choice for computer systems as detectors can single out not only designated intruding processes (intruders) but also intruding processes (intruders) that perform similar tasks. However, while the structure (the job) of lymphocytes is less specific but has the ability to identify a wider range of intruders, it has the drawback of being less competent to identify an exact intruder.

By employing this feature, precarious functions that are common among different methods of intrusions will alert the system to an attack. If the performed function is not hazardous, it then leads to the maturation process of lymphocytes. This is a process where immature lymphocytes are trained under the direction of thymic hormones. Those of which have the sharpest ability to identify foreign antigens survive. However, only a small percentage of lymphocytes pass this test; others that bind strongly and mount an attack against itself are vigorously weeded out and destroyed. In the digital world, a similar process of learning is required for the detection process to differentiate between the self and non-self.

3.5 Dynamically changing coverage

Another method for maintaining the flexibility of the artificial immune system is by adapting a feature of the natural immune system where coverage of possible intruders is random and constantly changing. No full set of detectors ever exist in the human body due to the huge number of different antigenic determinants, estimated to be 10 million, that the body can respond to.

Depending on how recent a particular type of intruder has been spotted in the human body, the amount of lymphocytes targeted specifically at this intruder will vary. If the intruder has been recently found in the system, a comparatively large number of lymphocytes will be dedicated to it. For possible intruders that have not attacked the body for a period of time, the human body is patrolled by a randomly selected subset of detectors. To ensure that random selection will not affect the efficiency of the immune system (i.e. degrading it), the subset of detectors are constantly changing through cell death and reproduction. When this random selection is implemented on the artificial immune system, it can greatly enhance the flexibility of the system, allowing for less resource intensive usage.

3.6 Distributability

One of the most important concepts that an artificial immune system should learn from its natural counterpart is the ability to monitor abnormal behaviour in a network of computers. Lymphocytes in nature, for instance, not only detect and respond to specific stimuli, they also perform the task of distributable so as to maintain a healthy system. To do so, detection is conducted not only in one location, but also throughout the body. These lymphocytes, which amount to approximately 10^{12} in the human body, patrol the entire organism determining whether the cells they come in contact with are part of the system or not.

To maintain this distributability in nature, each of the lymphocytes operates individually. Different types of lymphocytes identify different types of intruders. When a lymphocyte comes in contact with other cells, it tries to identify whether the cell is the type of cell (an intruder) that it has been assigned to detect. When intruders are identified, an alarm signal is sent out throughout the body to alert the system to beware of the intruder.

This distributability feature also ensures robustness as on the occasion that one or more lymphocytes are destroyed, the remaining system is still in operation. Each of the components of the system operates locally and individually, thus enabling global protection without the need for a central controlling system.

4 Intrusion Detection System

The concern for protecting computers from viruses, unauthorised users and other forms of unanticipated behaviour has raised research in the domain of intrusion detection systems (IDS). It is an autonomous system that aims at securing computer systems. The operation of the system can be separated into two phases, namely defining self and non-self along with generating negative detectors, and monitoring the system.

4.1 Self and Non-self

Immunologists have long-perceived the study of immunology as the study of differentiating internal cells from other external entities. Parallel to nature anomaly detection systems "can be generally viewed as the problem of learning to distinguish *self* from *other*" [12]. *Self is* defined as legitimate users, data and other normal behaviour of the system. Conversely, *other* is conceived to be unauthorised users, damaged data, redundancy, viruses and other security breaches.

4.1.1 Generation of the image of self and its Detectors

The task of defining the boundary between self and non-self can be a complex process. Each string of bits is a member of a set of either self or non-self, which are mutually exclusive [19]. Upon the introduction of a new string, it is labelled as either normal or anomalous depending on the boundary previously defined. Any faults in the determination of the boundary arise in errors, such as false positives and false negatives. A false negative error results when the string is anomalous but considered normal, while a false positive is a normal string that is deemed unsafe (Figure 1).

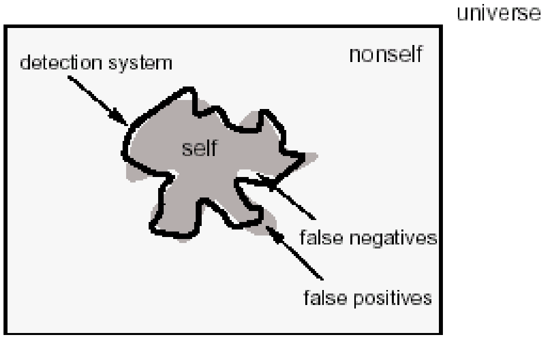

Figure 1 Representation of Self and Non-self [18]

Following the formation of an unambiguous boundary is the task of generating detectors for the non-self. As can be seen in figure 2 [19], possible negative detectors are randomly generated. If they cover any part of self, they are discarded. The process of generating negative detectors continues until a full set is formed, similar to the way lymphocytes are trained under the direction of thymic hormones in nature. The way in which these negative detectors are utilised then depends on the type of detection algorithm used (further discussed in Section 4.2).

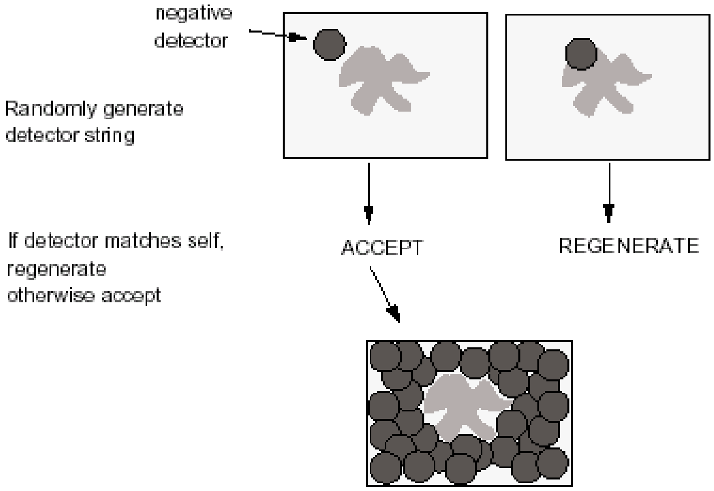
negative detector
Randomly generate detector string
If detector matches self, regenerate otherwise accept
ACCEPT
REGENERATE

Figure 2 Generation of negative detectors [18]

4.2 Detection Algorithm

As researches in the domain of Intrusion Detection Systems (IDS) grows in number and in intensity, various algorithms have been suggested to perform the task. Based on nature's thymic negative selection strategy, these algorithms can be classified into two techniques: misuse detection and anomaly detection, and into two types of systems: host-based and network-based. It thus results in four major classes of systems: host-based misuse detection system, network-based misuse detection system, host-base anomaly detection system and network-based anomaly detection system. [5]

The misuse detection approach detects intrusions through the use of signatures of known system vulnerabilities [24]. Anomaly detection, on the other hand, utilises databases built upon system activities during standard operation, to distinguish behaviours that deviate from the norm. The two approaches compared, misuse detection is usually considered as more trustworthy as a system because of its lower rate of false negatives [5]. However, its major shortcoming lies in its limited ability to identify previously unknown attacks.

4.2.1 String Detection

One of the earliest proposed methods that differentiates self from non-self is string detection as proposed in the paper titled ‘Self-Nonself’ Discrimination in a Computer’, by Professor Stephanie Forrest and her colleagues at the University of New Mexico. Focusing on the detection of “unauthorised use of computer facilities, guaranteeing the integrity of data files and preventing the spread of computer viruses’ [12], the technique employs detectors generated from the system, following constant monitoring the system by comparing protected data, i.e. self, with its detectors.

When comparing data, the string matching technique is employed, where a string, *s*, is matched with a string detector, *d*. Rules capable of performing this action include Hamming distance, edit distance or *r*-contiguous bit [19]. The matching of strings *d* and *s*, by using *r*-contiguous bits rule, defines that d and s should have the identical bits in at least *r*-contiguous locations. Theoretically, *r*, which can be any number, is a threshold that determines the specific number of a particular detector sets (Figure 3).

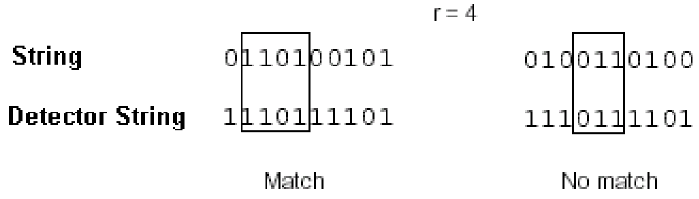

Figure 3 Matching with contiguous rule [19]

The major objective of this algorithm is to detect changes to protected data and to become aware of addition of new data. Phase I of the algorithm involves the generation of the detector set. First of all, protected data are presented as strings of bits which are segmented into equal size portions. A random string is generated from the system and then used to check against the protected data. If the random string does not match the protected data, it is deemed as part of the detector set (Figure 4). Otherwise, it is disregarded and another random string is generated. This process continues until all data are compared with the protected data and a complete set of detectors is generated.

Phase II is the monitoring process. It consists of continuously matching a random string from current system behaviour with a random string from the detector set in the background. Indication of a match in the system infers a non-self within the system (Figure 5).

Major benefits of this algorithm include: [38]

- Protection for multiple sites. This can be achieved by ensuring each copy of the detection algorithm is unique, as well as its probabilistic detection feature. These characteristics minimise the chances of intrusion at multiple sites even when one site had been successfully attacked into.
- A robust system. The algorithm proposed is more robust than current systems, in the sense that it detects foreign activities rather than looks for just specific patterns of intrusion, such as signatures
- A small set of detectors. The smaller the set of detectors, the higher the probability of identifying random changes to the original data set.
- More economical. The cost of checking data is relatively cheap, in terms of time and space, when comparing with the cost of generating signatures,. Signatures are expensive to generate computationally and usually require more than one for each data set.

However, there is a major drawback in the proposed algorithm. While detection of modified and added data can be easily detected, data which is deleted from the self can escape detection effortlessly.

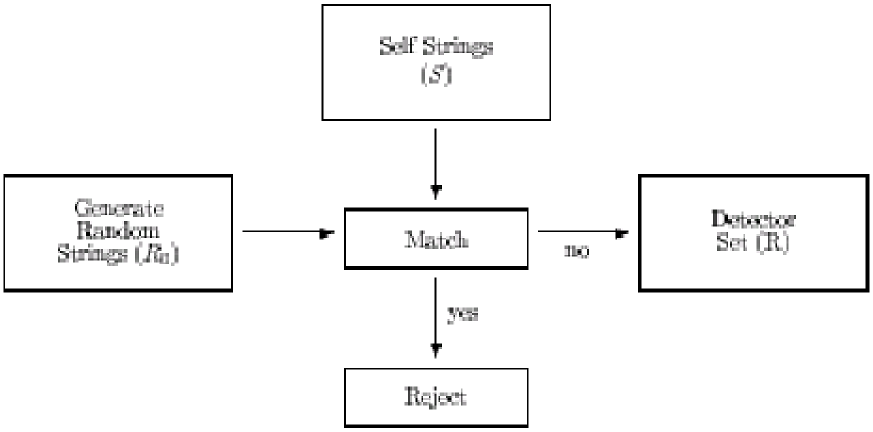

Figure 4 Generation of Detectors [12]

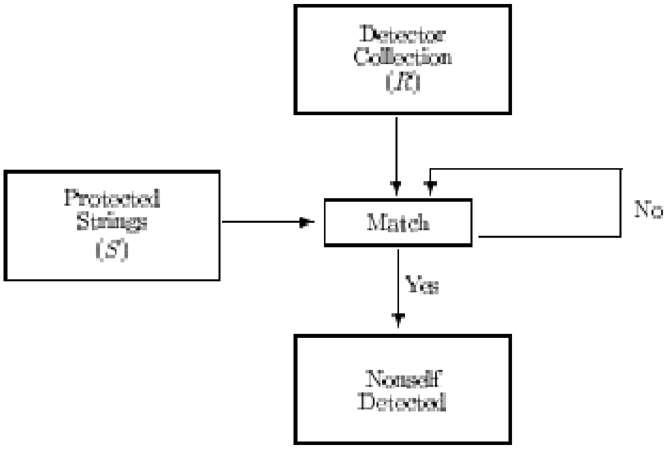

Figure 5 Monitoring of protected data [12]

4.2.2 System Calls

Another approach to detecting intrusions in computer systems is by monitoring sequences of system calls. This approach was first proposed in 1996 by S. Forrest's team. Since then, numerous researches have focused in this area, including numerous articles cited in [40], such as [11], has proven that the normal behaviour of a program can be 'characterised by local patterns' [40]. Any sequence that is irregular is considered to be suspicious.

All of the three methods described below utilises sequences of system calls to identify the differences between normal and intrusive behaviour. They all encompass a series of learning techniques, which is required by the system to develop its set of 'normal' processes.

4.2.2.1 Enumeration

Enumeration is a method for differentiating self and non-self [11]. First of all, each sequence of normal behaviour is recorded. Subsequently, the system is scanned for mysterious sequences. The types of sequences that are most suited for this method are ones which are contiguous and of the same length. In addition, the sequences are preferably stored in a tree structure in order to allow for easy access. This enables the database of normal behaviour to be compact, yet computationally efficient, and thus speeds up the comparison process.

As presented in [11], the database of self is complied by "sliding windows of size $k + 1$ across the trace of system calls and records which calls follow which within the sliding window" [11]. Using the same sliding window algorithm, these normal patterns are then used to check for the reliability of new traces. The result of checking is determined by how the sequences of system calls differ from the *self* database.

4.2.2.2 Frequency-based

Another method of intrusion detection using system calls is frequency-based methods, which measure the frequency each sequence occurs. Suggested by Bhangoo and Helman [17], the frequency of each sequence is ranked by the number of occurrences in both normal circumstances and during intrusions. Thus, those sequences that repeatedly occur in an intrusion or are less prevalent during normal operation are classified as untrustworthy. However, as not all normal sequences can be predetermined, assumptions are made to choose a 'frequency distribution for abnormal sequences' [40]. A variety of methods can be used when choosing this distribution, with the easiest being a distribution that is compiled by assuming that abnormal distribution is uniform.

The determination of an intrusion is defined by counting the number of deviations from the norm which exceeds the threshold level. In such a case, an alert is triggered [5].

4.2.2.3 Data mining

To maintain a more refined database of self-sequences, it was suggested by Lee and colleagues [29] that instead of obtaining a collection of all normal behaviour, patterns of normal behaviour could be used in its place. By simply learning the underlying features of what is considered to be normal, it will enable the learning process to easily generalise the different patterns that could occur in a system even with insufficient training data. During the monitoring process, system behaviour is checked against the database of normal behaviour patterns. Sequences that do not match the normal patterns are ruled as abnormal, and thus treated with care.

4.2.3 Detectors with lifecycles

Proposed by Forrest et al. [8], is a method of detection which is closely modelled upon nature. Detectors of the biological world are short-lived in order to ensure a greater coverage of possible invaders (see Section 3.x). Similarly, detectors in the artificial world can be trained to work the same way. At any particular time, only a partially random set of detectors are on alert. These detectors, resembling their biological counterpart, also go through the lifecycle process. Death fall upon them with a probability of p_{death} of death after a period of time in service. On its destruction, another randomly generated detector is created to take up its role as a protector of the system (Figure 6).

One of the benefits of a dynamic detector population is the system's ability to evolve the set of self. Throughout the lifetime of a computer system, the boundary of self and non-self constantly changes. The up to date images of self, however, will always be reflected in the detector set due to the constant generation of new detectors and destruction of old detectors.

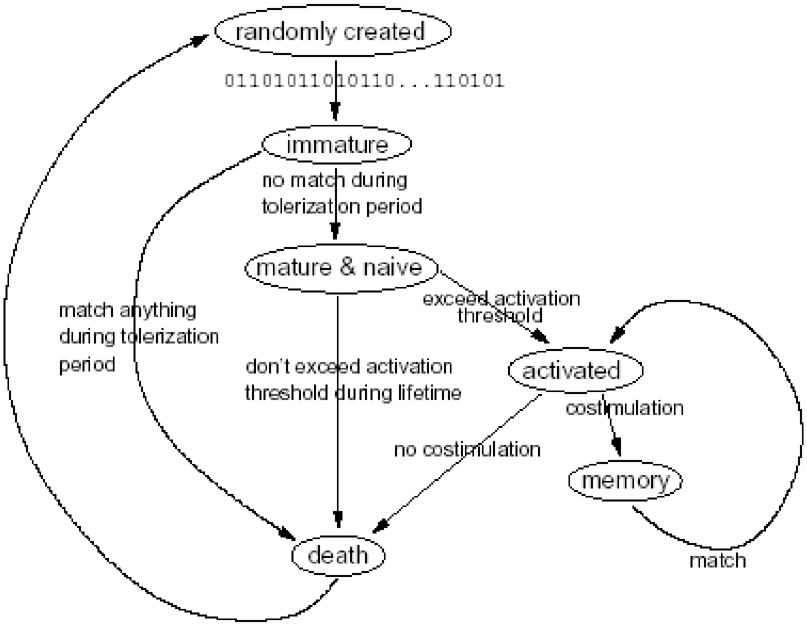

Figure 6 Lifecycle of a detector [18]

5 Distributed detection

Due to its distributed nature, the biological immune system is capable of survival even though part of the system have been invaded. Due to the individuality of lymphocyte detector cells, each of them is responsible for protecting self against a small portion of non-self. This causes a reduction in the coverage at any given site, but in return it provides a sufficient amount of system-wide coverage [10].

Similarly, a computer network can defend itself against harm by locating sensors at traffic directing devices, such as switches and routers, as they separate a network into segments. Currently, networks are protected against external access by firewalls, boarder routers and gateways. These walls limit the number of entry points into a network. Therefore, defensive mediums are normally positioned at the boundary points. These sensors enable the system to focus its attention on local traffic that is either going in or out of the particular host. The trade-off to being more knowledgeable of local resources is a restricted view of the wider community. Furthermore, by monitoring network traffic, sensors are dependent solely on network protocols, rather than depending on operating systems.

As stated in [19], to maintain a scalable and efficient intrusion detection system, sensors are best equipped with a superset of the system's definition of self. This will solve the problem of accuracy and consistency caused by employing different sets of self over different hosts on a network.

A restrictive view of a segment of a network, however, is insufficient when trying to understand the purpose of the attackers. In order to gain a wider perspective, and to encapsulate the whole community without using a central controlling system, information must be gathered from other distributed systems. The gathered information will be used to further develop the profile of an individual attacker's actions. [5]

5.1 Bayesian Methods

As developed by Boyd, the central ideas of decision-making are observe, orient, decide and act, a reoccurring sequence known as OODA [41]. Knowledge creation lies within the second stage, the orient stage, where data is correlated to determine the possible actions of the attacker.

The data correlation process is complex. It consists of four refinement stages, namely data, object, situation, and meaning and process [5]. The initial stage of data refinement filters noise out from the collected raw data, resulting in information only of interest, which is then grouped into related events. The object refinement stage, is the process where information is standardised to include synchronized time and a common format such as the DARPA Common Intrusion Detection Framework (CIDF) [27].

Subsequently, the situation refinement stage groups the objects into sets based on common attributes or related behaviours. For example, a mass number of attacks originating from one IP address, or similar attacks from varied IP addresses in a short period of time [5]. Finally, meaning and process refinement is applied to the objects and sets to determine their possible future actions. The derived information is then added to the current knowledge base.

A possible method of carrying out situation refinement is through the use of Bayesian Multiple Hypothesis Tracking (BMHT) algorithm (41). This algorithm has the ability to track targets with confidence. Essentially, it generates a list of possibilities that could explain the data examined. These possibilities are then evaluated for the probability of the hypotheses, singling out one that is most probable.

6 Multi-layered protection

The idea of a layered approach to computer security has long existed before the concept of computer immunology. Currently, the multi-layered protection approach is already implemented in most network systems and is considered one of the most effective methods of preventing unauthorised access. A very simple example of this is the use of a firewall on a network to serve as the first line of defence, followed by an antivirus program focusing on internal operations. While firewalls are able to restrict access to the internal network and deliver only information that is considered acceptable, they are, as with all other layers, not fully reliable. Antivirus programs behind the firewalls are therefore necessary to add confidence to the data being processed.

More comprehensive than the above mentioned firewall-antivirus program combination is a multi-layered system that combines the following components:

1. Security policy of your organization
2. Host system security
3. Auditing
4. Router security
5. Firewalls
6. Intrusion detection system
7. Incident response plan

Although each of the layers can be implemented independently to defend a system, protection against invaders is optimal when all the layers are implemented together as a full security system providing multiple layers of protection.

The very first layer is a document which identifies the security concerns in an organization. Layers two to five are methods of controlling the information flow, thus preventing harmful or unnecessary data to enter the internal system. The last two layers handle problems that arise within the system, by utilising the intrusion detection system it discloses abnormal behaviour, and the incident response plan dictates the actions taken when the IDS has detected a fault.

6.1 Ineffectiveness of Firewalls

It is commonly believed that to secure a computer network, a firewall is most suited for the task. However, it is not a complete solution to the computer security problem. The major duty of a firewall is to restrict network traffic to what is desired by network managers, and not to prevent an attack from occurring once traffic has flown through its walls. In addition, all traffic between each of the host in and internal network are not interfered, thus merely implementing a firewall cannot solve internally aroused security problems.

7 Taxonomy of Intrusions

Following the detection of an intrusion, the next step towards its elimination is to determine the type of attack, namely the identifying and classification process. Unlike the natural immune system which recognises the type of viruses or bacteria on discovery, the intrusion detection systems discussed in section 4 are only capable of distinguishing non-self from self, but incapable of classifying the detected non-self.

The task of categorising exposed non-selves lies in the taxonomy of intrusion. This process further open the door to a wealth of extra information such as statistics on intrusion and discovery of new patterns [30]. One method of classification, proposed by Neumann and Parker, utilises a nine class hierarchy (See Table 1). According to an examination of this method, by Lindqvust and Jonsson [30], where the classes are organised in the form of the physical world, hardware, software and other forms of computer abuse, the outcome/result is "well-founded and… covers most of the known techniques", and also has an "inherent grading of the classes from external attacks to authorized users misusing their privileges". However, it does have its weaknesses, such as ambiguity in classifying password related attacks, dilemma in determining the proper class of misuse techniques — bypassing intended controls (NP5) or passive misuse of resources (NP7), and difficulty in classifying human behaviour [30].

Various other methods of classification have been proposed/suggested in the past years, some of which include the work of Lackey, Brinkley and Schell, Kumar and Axelsson. Lackey, in 1974, basing his classification system on genuine examples of system penetration, derived six categories of penetration techniques. Recent works performed in this area include Brinkley and Schell's six resource-oriented computer misuse classes, Kumar's classification system that is based on commonalities left behind by intruders, and Axelsson's proposal of a "taxonomy of system characteristics".

Each of the classification schemes mentioned above are not flawless and do not fulfil every aspect of what an ideal taxonomy should do. Collaborated from articles focused on the topic of classification, the following are properties that define an effective taxonomy:

- Description. Every category should have a clear and unambiguous definition of what it should contain. [30]
- Explanation. The taxonomy should provide clues to the possible cause of the problem identified. [2]
- Mutual exclusion. All categories should be mutually exclusive of each other as well as exhaustive to provide unambiguous sets that will cover all the possible problems that could affect the target system. [30].
- Internal and external distinction. If successful, the taxonomy should be able to differentiate between attacks that require internal access to those performed by external entities [30].

Table 1. Computer misuse techniques [30]

Class	Description
NP1 External misuse	Generally nontechnological and unobserved, physically separae from computer and communication facilities, for example visual spying.
NP2 Hardware misuse	a) Passive, with no (immediate) side effects. b) Active, with side effects.
NP3 Masquerading	Impersonation; playback and spoofing attacks etc.
NP4 Setting up subsequent misuse	Planting and arming malicious software.
NP5 Bypassing intended controls	Circumvention of existing controls or improper acquisition of otherwise denied authority.
NP6 Active misuse of resources	Misuse of (apparently) conferred authority that alters the system or its data.
NP7 Passive misuse of resources	Misuse of (apparently) conferred reading authority.
NP8 Misuse resulting from inaction	Failure to avert a potential problem in a timely fashion, or an error of omission, for example.
NP9 Use as an indirect aid in committing other misuse	a) As a tool in planning computer misuse etc. b) As a tool in planning criminal/unethical activity.

8 Elimination

Presented by [21] is an approach for correcting a system after the detection of viruses or other forms of intruders. When an invasion is identified, it will be checked against the database of previously known intruders. If an exact match is found, the intruder will be handled in the same manner as it predecessors, as the same method has been proven to be successful. On the occasion where an exact match cannot be found, the system will try to locate an approximate match in order to gain clues to how the intruder can be defeated.

If the intruder is entirely new to the system, it will be captured by decoy programs. A signature that identifies the intruder will be generated and used for referencing in the future. In addition, the intruder is analysed to determine where and how the system has been affected, so as to further resolve the situation. Subsequently, information gathered concerning the new intruder, such as its signature and method of returning the system to a stable state, is recorded into the database and sent to neighbouring systems. Being an autonomous system, the artificial immune system ideally should be able to perform these tasks without external influence.

9 Future Directions

To date, the different techniques of identifying suspicious behaviours in computers have proven to be quite successful. However, the development of anomaly detection strategies is by no means complete, and it is believed that there can be more effective methods to determine any external behaviour in a system. At present, most approaches to monitoring a system are based on random detection, such as random generation of detectors or random matching of auditing of behaviour. This is the method the biological immune system employs to achieve the detection purpose. A disadvantage of this technique is its low degree of specificity. While random checking is sufficient in nature, it is insufficient for the digital world. What is required is a more specific approach that can detect external entities as they enter the system.

Other features of the immune system that require further research include the classification of intrusions and methods of removing the intruders. As stated in [30], the classification of intrusion techniques are still imperfect. There is currently no classification scheme that can systematically categorise all security breaches without ambiguity. Furthermore, studies on methods of destroying intruders are far from extensive.

10 Conclusion

An artificial immune system is a valuable addition to the security of computer systems. Due to its autonomy, it allows not only security personnel, but also the average computer users to tackle the mass amount of security flaws that exist in our systems.

At present, extensive research has focused on intrusion detection, i.e. techniques related to the separation of normal system activities from abnormal suspicious behaviours. A variety of approaches to achieving this goal have been proposed. Experiments under controlled conditions have shown that while all of the approaches are able to recognise most common intrusion types and some less common intrusions types, some approaches do it more efficiently than others.

The summary of works performed in the domain of computer immunology, as presented in this thesis, is by no means exhaustive. On the topic of intrusion detection systems alone, Mè and Michel [32] was able to compile a bibliography consisting of over 600 references. Research in the classification of intrusion types and elimination of intruders, on the other hand, has far less resources.

In summary, it can be said that technologies for an artificial immune system currently exist but they are fragmented and imperfect. Extension effort will be required to build a truly operational immune system: one that contains an intrusion detection system which has an efficient method of classifying the intrusions, the ability to operate in a distributed network, the provision of multi-layered protection, and the ability to eliminate intruders.

Bibliography

[1] P. Ammann, S. Jajodia, McCollum, C.D. Blaustein, B.T. “Surviving Information Warfare Attacks on Databases”, *1997 IEEE Symposium on Security and Privacy*, Oakland (1997).

[2] S. Axelsson, “Intrusion Detection Systems: A Survey and Taxonomy”, http://citeseer.nj.nec.com/axelsson00intrusion.html, 2000.

[3] S. Axelsson, “Research in Intrusion-Detection Systems: A Survey”, http://citeseer.nj.nec.com/axelsson98research.html, 1998.

[4] M. Burgess. “Computer Immunology”, *Proceedings of the Twelfth Systems Administration Conference*, p282–298, December 6-11 1998, Boston, Massachusetts.

[5] D. Burroughs, L. Wilson, G. Cybenko, “Analysis of Distributed Intrusion Detection System Using Bayesian Methods”, *21st IEEE International Performance, Computing, and Communications Conference*, Phoenix (2002)

[6] S. Cheung, K. Levitt, “Protecting Routing infrastructures for Denial of Service Using Cooperative Intrusion Detection”. In *Proceedings of the New Security Paradigms Workshop, Langdale (1997).*

[7] F. Cohen. “Computer Viruses”, Computers and Security, 1987.

[8] D. Dasgupta and S. Forrest. “Artificial Immune System in Industrial Applications”, *Accepted for presentation at the International conference on Intelligent Processing and Manufacturing Material (IPMM)*. Honolulu, HI (July 10-14, 1999).

[9] S. Forrest and S.A. Hofmeyr. "Immunology as information processing", In Design Principles for the Immune System and Other Distributed Autonomous Systems, edited by L.A. Segel and I. Cohen. Santa Fe Institute Studies in the Sciences of Complexity. New York: Oxford University Press (2001).

[10] S. Forrest, S. Hofmeyr, and A. Somayaji. “Computer immunology”, *Communications of the ACM* Vol. 40, No. 10, pp. 88-96 (1997).

[11] S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Longstaff. "A Sense of Self for Unix Processes", *Proceedings of 1996 IEEE Symposium on Computer Security and Privacy* (1996).

[12] S. Forrest, A.S. Perelson, L. Allen, R. and Cherukuri. “Self-nonself discrimination in a computer”, *In Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy*, Los Alamitos, CA: IEEE Computer Society Press (1994).

[13] S. Forrest, A. S. Perelson, “Genetic Algorithm and the immune system”, Parallel Problem Solving from Nature”, 1991.

[14] A. Ghosh, J. Wanken, F. Charron, “Detecting Anomalous an Unknown Intrusions Against Programs”, http://citeseer.nj.nec.com/ghosh98detecting.html, 1998.

[15] S. Greenwald, “Discussion Topic: What is the old Security Paradigm?”, In *Proceedings of the New Security Paradigms Workshop,* Charlottesville (1998).

[16] J. Hale, S. Shenoi, “Catalytic Inference Analysis: Detecting Inference Threats due to Knowledge”, *1997 IEEE Symposium on Security and Privacy*, Oakland (1997).

[17] P. Helman, J. Bhangoo, “A Statistically Based System for Prioritzing Infromation Exploration Under Uncertainty”, IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 1997.

[18] S. Hofmeyr and S. Forrest. "Architecture for an Artificial Immune System", Evolutionary Computation 7(1), Morgan-Kaufmann, San Francisco, CA, pp. 1289-1296 (2000).

[19] S. Hofmeyr, and S. Forrest. “Immunizing Computer Networks: Getting All the Machines in Your Network to Fight the Hacker Disease” (1998).

[20] J. Hughes, T. Aura, M. Bishop, “Using Conservation of Glow as a Security Mechanism in Network Protocols”, *2000 IEEE Symposium on Security and Privacy*, Berkeley (2000).

[21] J. Kephart. “A Biologically Inspired Immune System for Computers”, *Artificial Life IV: Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Things*, MIT Press, 1994.

[22] J. Kephart, G. Sorkin, M. Swimmer, and S. White. “Blueprint for a Computer Immune System” *Virus Bulletin International Conference* in San Francisco, California, (October 1-3, 1997).

[23] J. Kephart, S. R. White, D. M. Chess, “Computers and Epidemiology”, IEEE Spectrum

[24] J. Kim, P. Bentley, “The Human Immune System and Network Intrusion Detection”, http://citeseer.nj.nec.com/kim99human.html, 1999.

[25] C. Ko, “Logic Induction of Valid Behaviour Specifications for Intrusion Detection”, *2000 IEEE Symposium on Security and Privacy*, Berkeley (2000).

[26] C. Ko, M. Ruschitzka, K. Levitt, “Execution Monitoring of Security-Critical Programs in a Distributed Systems: A Specification-Based Approach”, *1997 IEEE Symposium on Security and Privacy*, Oakland (1997).

[27] W. Lee, R. Nimbalkar, K. Yee, S. Patil, P. Desai, T. Tran, S. Stolfo, “A Data Mining and CIDF Based Approach for Detecting Novel and Distributed Intrusions”, RAID 2000, 2000.

[28] W. Lee, S. Stolfo, K. Mok, “A Data Mining Framework for Building Intrusion Detection Models”, *1999 IEEE Symposium on Security and Privacy*, Oakland (1999).

[29] W. Lee, S. J. Stolfo, “Data Mining Approaches for Intrusion Detection”, Proceedings of the 7th USENIX Security Symposium, 1998.

[30] U. Lindqvist, E. Jonsson, “How to Systematically Classify Computer Security Intrusions”, *1997 IEEE Symposium on Security and Privacy*, Oakland (1997).

[31] U. Lindqvist, P. Porras, “Detecting Computer and Network Misuse Through the Production”, *1999 IEEE Symposium on Security and Privacy*, Oakland (1999).

[32] L. Mé, C. M. Sup’elec, “Intrusion Detection: A Bibliography”, http://citeseer.nj.nec.com/484682.html, 2001.

[33] C. Meadows, “Three Paradigms in Computer Security”, In *Proceedings of the New Security Paradigms Workshop, Langdale* (1997).

[34] W. Murray, “The Application of epidemiology to computer viruses”, Computers and Security, 1988.

[35] L. Nunes de Castro, F. José Von Zuben, “Artificial Immune Systems: Part II A Survey Of Applications”, http://citeseer.nj.nec.com/nunesdecastro00artificial.html, 2000.

[36] P. Pal, F. Webber, R. Schantz, J. Loyall, R. Watro. W. Sanders, M. Cukier, J. Proudler. “Survival by Defence-Enabling”. In *Proceedings of the New Security Paradigms Workshop,* Cloudcroft (2001)

[37] R. F. Smith. “Why you need LANguard S.E.L.M and how to use it on your network”, GFI Software LTD, 2002

[38] A. Somayaji, S. Hofmeyr, and S. Forrest. “Principles of a Computer Immune System”, *1997 New Security Paradigms Workshop*, pp75-82, ACM (1998).

[39] E. Spafford, “Computer Viruses – a form of artificial life?”, Artificial Life II, p727-745, 1992.

[40] C. Warrender, S. Forrest, B. Pearlmutter, “Detecting Intrusions using System Calls: Alternative Data Models”, *1999 IEEE Symposium on Security and Privacy*, Oakland (1999).

[41] E. Waltz, “Information Warfare: Principles and Operations”, Artech House, Norwood, 1998.

[42] Computer Immune System, http://www.cs.unm.edu/~immsec/ (current 20th April 2002)

[43] Computer Immunology @ Oslo University College, http://www.iu.hio.no/~mark/research/immune/ (current April 20th 2002)