Employing Artificial Immunology and Approximate Reasoning Models for Enhanced Network Intrusion Detection

Seyed A. Shahrestani

School of Computing and Mathematics
University of Western Sydney
Penrith Campus, Locked Bag 1797
PENRITH SOUTH DC NSW 1797
AUSTRALIA
seyed@computer.org

Abstract: - With the massive connectivity provided by modern computer networks, more and more systems are subject to attack by intruders. The creativity of attackers, the complexities of host computers, along with the increasing prevalence of distributed systems and insecure networks such as the Internet have contributed to the difficulty in effectively identifying and counteracting security breaches. As such, while it is critical to have the mechanisms that are capable of preventing security violations, complete prevention of security breaches does not appear to be practical. Intrusion detection can be regarded as an alternative, or as a compromise to this situation. Several techniques for detecting intrusions are already well developed. But given their shortcomings, other approaches are being proposed and studied by many researchers. This paper discusses the shortcomings of some of the more traditional approaches used in intrusion detection systems. It argues that some of the techniques that are based on the traditional views of computer security are not likely to fully succeed. An alternative view that may provide better security systems is based on adopting the design principles from the natural immune systems, which in essence solve similar types of problems in living organisms. Furthermore, in any of these methodologies, the need for exploiting the tolerance for imprecision and uncertainty to achieve robustness and low solution costs is evident. This work reports on the study of the implications and advantages of using artificial immunology concepts for handling intrusion detection through approximate reasoning and approximate matching.

Key-Words: - Intrusion detection, Natural immune system, Soft computing, Approximate reasoning.

1 Introduction

All Internet-based and intranet-based computer systems are vulnerable to intrusions and abuse by both legitimate users, who abuse their authorities, and unauthorized individuals. The personal computer and the Internet have become indispensable parts of everyday lives, while they are exceedingly vulnerable to even simple attacks. The vulnerability of some of these systems stems from the simple fact that they were never intended for a massive interconnection. At any case, with the rapidly increasing dependence of businesses and government agencies on their computer networks, protecting these systems from intrusions or at least the capability to detect intrusive behavior is crucial.

In general, any deliberate unauthorized attempt to access or manipulate information, or render a system unreliable or unusable is considered an intrusion attempt. An Intrusion detection system (IDS) is a tool that attempts to identify intrusive behavior 0. While the complexities of host computers are already making intrusion detection a difficult task, the increasing pervasiveness of networked-based systems and insecure networks such as the Internet has greatly increased the need for sophisticated approaches for intrusion detection.

An IDS assumes that an intruder’s behavior will be noticeably different from that of a legitimate user. It also assumes that many unauthorized actions are detectable. Two major approaches for detecting computer security intrusions in real time are misuse detection and anomaly detection. Misuse detection attempts to detect known attacks against computer systems. Anomaly detection uses knowledge of users’ normal behavior to detect attempted attacks. The primary advantage of anomaly detection over misuse detection methods is the ability to detect novel and unknown intrusion [2], [3].

Most of these systems are mainly dependent on knowledge based systems or input/output descriptions of the operations, rather than on deterministic models. Uncertainty is also a dominant feature of these systems. Uncertainties can be the result of lack of a comprehensive knowledge base, insufficiency or unreliability of data on the particular object under consideration, or stochastic nature of relations between the propositions used in the system [4]. In any of these systems, the need for exploiting the tolerance for imprecision and uncertainty to achieve robustness and low solution costs is evident. Consequently, approximate reasoning and handling intrusion detection through approximate matching can lead to more proficient ways of detecting intrusive behavior.

Another interesting and somehow different approach to intrusion detection is related to building computer immune systems as inspired by anomaly detection mechanisms in natural immune systems [5]. Such a system would have highly sophisticated notions of identity and protection that provides a general-purpose protection system to complement the traditional systems. The natural immune system tries to distinguish ‘self’ from the dangerous ‘other’ or ‘nonself’ and tries to eliminate the ‘other’ [6]. This can be viewed as a similar problem in computer security; where ‘nonself’ might be an unauthorized user, computer viruses or worms, unanticipated code in the form of Trojan horse, or corrupt data.

These issues are further discussed in the remainder of this paper. Next section gives an overview of the more traditional approaches to intrusion detection, along with their advantages and shortcomings. Section 3 demonstrates the importance of approximate reasoning and fuzzy intrusion detection. Section 4 gives an overview of intrusion detection approaches inspired by the natural immune systems and provides specific exemplary situations for utilization of such systems. The last section gives the concluding remarks.

2 Network Intrusion Detection Approaches

Typically, IDSs employ statistical anomaly and rule-based misuse models in order to detect intrusions. The detection in statistical anomaly model is based on the profile of normal users’ behavior. It will statistically analyze the parameters of the users’ current session and compares them to their normal behavior. Any significant deviation between the two is regarded as a suspicious session. As the main aim of this approach is to catch sessions that are not normal, it is also referred to as an ‘anomaly’ detection model. The second model is dependent on a rule-base of techniques that are known to be used by attackers to penetrate. Comparing the parameters of the users’ session with this rule-base carries out the actual act of intrusion detection. This model is sometimes referred to as a misuse detection model, as it essentially looks for patterns of misuse patterns known to cause security problems [2].

2.1 Statistical Detection

Statistical anomaly detection systems initiate the detection of the security breaches by analyzing the audit-log data for abnormal user and system behavior. These systems assume that such an abnormal behavior is indicative of an attack being carried out. An anomaly detection system will therefore attempt to recognizing the occurrence of ‘out of the ordinary’ events. For implementation purposes, the first step is concerned with building a statistical base for intrusion detection that contains profiles of normal user and system behavior. Based on that, these systems can then adaptively expand their statistical base by learning user and system behavior. This model of intrusion detection is essentially based on pattern recognition approaches, i.e. the ability to perceive structure in some data.

To carry out the pattern recognition act, the raw input data is pre-processed to form a pattern. As such, a pattern is an extract of information regarding various characteristics or features of an object, state of a system, etc. Patterns either implicitly or explicitly contain names and values of features, and if they exist, relationships among features. The entire act of recognition can be carried out in two steps. In the first step, a particular manifestation of an object is described in terms of suitably selected features. The second step, which is much easier than the first one, is to define and implement an unambiguous mapping of these features into class-membership space.

More specifically for intrusion detection purposes, the statistical analysis detects variation in a user's behavior by looking for significant changes in the session in comparison to user's behavior profiles or patterns. The profiles consist of the individual behavior in previous sessions and serve as a means for representing the expected behavior. Obviously, the information contents of the patterns that make up the profiles need to be dynamically updated. For intrusion detection purposes, various types of subjects may need to be considered and monitored. These may include users, groups, remote hosts, and overall target systems. Monitoring groups enables the detection system to single out an individual whose behavior significantly deviates from the overall average group behavior. Detection of system wide deviations in behavior that are not connected to a single user may be achieved by monitoring the target system. For instance, a large deviation in the number of system wide login attempts may be related to an intrusion.

To determine whether the behavior is normal or not, it is characterized in terms of some of its key features. The key features are then applied to individual sessions. While the features employed within different intrusion detection systems may vary substantially, they may be categorized as either a continuous or a discrete feature. A continuous feature is a function of some quantifiable aspect of the behavior such that during the course of the session its value varies continuously. Connection time is an example of this type of feature. This is in contrast to a discrete feature that will necessarily belong to a set of finite values. An example of such a feature is the set of terminal location. For each subject, the maintained profile is a collection of the subject's normal expected behavior during a session described in terms of suitably selected features.

The classification process to determine whether the behavior is anomalous or not, is based on statistical evaluations of the patterns stored as profiles specific for each subject. Each session is described by a pattern, usually represented as a vector of real numbers, consisting of the values of the features pre-selected for intrusion detection. The pattern corresponds to the same type of features recorded in the profiles. With the arrival of each audit record, the relevant profiles are solicited and their contents, the patterns they contain, are compared with the pattern vector of intrusion detection features. If the point defined by the session vector in the *n*-dimensional space is far enough from the points corresponding to the vectors stored in the profiles, then the audit record is considered anomalous. It can be noted that while the classification is based on the overall pattern of usage, the vector, highly significant deviations of the value of a single feature can also result in the behavior being considered as anomalous.

To be useful, the intrusion detection system must maximize the true positive rate and minimize the false positive rate. In most cases, but not all, achieving a very low false positive rate, that is a low percentage of normal use classified incorrectly as anomalous, is considered more crucial. This can be achieved by changing the threshold of the distance metric that is used for classifying the session vector. By raising this threshold, the false positive rate will be reduced while this will also lower the true positive rate, and hence fewer events are considered abnormal.

To increase speed and to reduce misclassification error, particularly when the number of classes, for instance the number of users, is large or not known, some suggestions have been made for grouping of classes. For example, patterns can be mapped into a generalized indicator vector, based on their similarities. This vector is then used in conjunction with a standard search tree method for identification purposes. Another method first computes a similarity measure-based on distance metric- between each pattern and every other pattern and merges close samples with each other. Yet, another proposed method is to find a pattern prototype, a typical example of certain classes, and use that for establishing the category of a new pattern before comparing it with other exemplars of that category to recover its specific identity, see [3] for details.

2.2 Rule-based Misuse Detection

Obviously, attempting to detect intrusions based on deviations from expected behaviors of individual users has some difficulties. For some users, it is difficult to establish a normal pattern of behavior. Therefore, it will be easy for a masquerader to go undetected as well. Alternatively, the rule-based detection systems are based on the understanding that most known network attacks can be characterized by a sequence of events. For implementation purposes, high-level system state changes or audit-log events during the attacks are used for building the models that form the rule bases. In Rule-based misuse detection model, the IDS will monitor system logs for possible matches with known attack profiles [2]. Rule-based systems generate very few false alarms, as they monitor for known attack patterns.

There is another situation for which statistical anomaly detection may not be able to detect intrusions. This is related to the case when legitimate users abuse their privileges. That is, such abuses are normal behavior for these users and are consequently undetectable through statistical approaches. For both of these cases, it may be possible to defend the system by enforcing rules that describe suspicious patterns of behavior. These types of rules must be independent of the behavior of an individual user or their deviations from past behavior patterns. These rules are based on the knowledge of past intrusions and known deficiencies of the system security. In some sense, these rules define a minimum standard of conduct for users on the host system. They attempt to define what can be regarded as the proper behavior that its breaches will be detected. Most current approaches to detecting intrusions utilize some form of rule-based analysis. Expert systems are probably the most common form of rule-based intrusion detection approaches; they have been in use for several years [3].

The areas of KBS, expert systems, and their application to intrusion detection have been and still are a very active research area. Among the very important aspects of the KBSs, are their knowledge bases and their establishment. This area and related subjects may be considered as a field by itself, referred to as ‘knowledge engineering’. Knowledge engineering is the process of converting human knowledge into forms suitable for machines, e.g. rules in expert systems. Some examples of an interdisciplinary approach based for knowledge engineering in computer security systems are described in [5].

For successful intrusion detection, the rule-based subsystem contains knowledge about known system vulnerability, attack scenarios, and other information about suspicious behavior. The rules are independent from the past behavior of the users. With each user gaining access and becoming active, the system generates audit records that in turn are evaluated by the rule-based subsystem. This can result in an anomaly report for users whose activity results in suspicious ratings exceeding a pre-defined threshold value.

Clearly, this type of intrusion detection is limited in the sense that it is not capable of detecting attacks that the system designer does not know about. To benefit from the advantages of both approaches, most intrusion detection systems utilize a hybrid approach, implementing a rule-based component in parallel with statistical anomaly detection. While in general, the inferences made by the two approaches are independent or loosely coupled. The two subsystems share the same audit records with different internal processing approaches. There are arguments and ongoing research in tightening the two together in the hope of achieving a reduced false-positive rate of anomaly detection and eliminating the possibility of multiple alarms [8].

3 Tolerance for Imprecise Intrusion Detection and Approximate Reasoning

Hybrid systems that are claimed to combine the advantages of both statistical and rule-based algorithms, while partially eliminating the shortcomings of each one, are also devised. In general, such systems will use the rule-based approach for detection of previously encountered intrusions and statistical anomaly detection algorithms for checking new types of attacks. An example of this general approach is based on utilization of neural networks that are trained to model the user and system behavior, while the anomaly detection consists of the statistical likelihood analysis of system calls [15]. Another approach is based on state transition analysis [16]. It attempts to model penetrations as a series of state changes that lead from an initial secure state to a target compromised state. A case based reasoning approach to intrusion detection, which alleviates some of the difficulties in acquiring and representing the knowledge is presented in [17]. A data-mining framework for adaptively building intrusion detection models is described in [18]. It utilizes auditing programs to extract an extensive set of features that describe each network connection or host session, and applies data mining approaches to learn rules that accurately capture the behavior of intrusions and normal activities.

In any of these algorithms and approaches to intrusion detection, the need for exploiting the tolerance for imprecision and uncertainty to achieve robustness and low solution costs is evident. This is in fact, the guiding principle of soft computing and more particularly approximate reasoning and fuzzy logic [19]. The subject of soft computing is the representation of imprecise descriptions and uncertainties in a logical manner. Many IDSs are mainly dependent on knowledge bases or input/output descriptions of the operation, rather than on deterministic models. Inadequacies in the knowledge base, insufficiency or unreliability of data on the particular object under consideration, or stochastic relations between propositions may lead to uncertainty. Uncertainty refers to any state of affair or process that is not completely determined. In rule-based and expert systems, lack of consensus among experts can also be considered as uncertainty. In addition, humans acting as administrators, security expert and the like, prefer to think and reason qualitatively, which leads to imprecise descriptions, models, and required actions.

Zadeh introduced the calculus of fuzzy logic as a means for representing imprecise propositions in a natural language as non-crisp fuzzy constraints on a variable [20]. This is ‘vagueness’: a clear but not precise meaning. That is to say, fuzzy logic started to cover vagueness, but turned out to be useful for dealing with both vagueness and uncertainty. The use of fuzzy reasoning in expert systems is naturally justifiable, as imprecise language is the characteristic of much expert knowledge. In crisp logic, propositions are either true or false, while in fuzzy logic different modes of qualifications are considered.

For any type of the intrusion detection algorithm, some points need to be further considered. In rule-based expert systems, administrators or security experts must regularly update the rule base to account for newly discovered attacks [2]. There are some concerns about any system that relies heavily on human operators or experts for knowledge elicitation. For instance, humans, in the course of decision making and reaching a conclusion, might use variables that are not readily measurable or quantifiable. Humans might articulate non-significant features. This, among other reasons, can lead to the establishment of inconsistent, from one expert to another, rule bases. In addition, the system will be slower than what it should be as some of the rules that make up the knowledge base are of secondary importance. Broadly speaking, experts' knowledge is necessarily neither complete nor precise. For these reasons, it is highly desirable to have systems and algorithms that acquire knowledge from experiential evidence automatically.

The statistical-anomaly detection algorithm will report ‘significant’ deviations of a behavior from the profile representing the user’s normal behavior. While the significant usually refers to a threshold set by the system security officer, in practice it can be difficult to determine the amount that a behavior must deviate from a profile to be considered a possible attack. In particular, as it will be discussed in the next section, for distributed anomaly detection based on the concepts in natural immune system, it is in fact advantageous to be able to carry out approximate detection.

4 Artificial Immunology Based Intrusion Detection

This section gives a brief overview of an interesting and somehow different approach to intrusion detection. The design objective for this approach is related to building computer immune systems as inspired by anomaly detection mechanisms in natural immune systems. The analogy between computer security problems and biological processes was suggested as early as 1987, when the term ‘computer virus’ was introduced [9]. But it took some years for the connection between immune systems and computer security to be eventually introduced [10], [5]. This view of computer security can also be of great value for implementing other intrusion detection approaches, for instance see [11] and [12].

In the immune system, the intrusion detection problem is viewed as a problem of distinguishing self, for instance legitimate users and authorized actions, from nonself or intruders. To solve this problem, detectors that match anything not belonging to self are generated. The method relies on a large enough set of random detectors that are eventually capable of detecting all nonself objects. While these systems show several similarities with more traditional IDSs, they are more autonomous. Such systems present many desirable characteristics [6]. In particular, it needs to be noted that the detection carried out by the immune system is approximate; the match between antigen or foreign protein, and receptor, surface of the specialized cells in the immune system, need not be exact. This will allow each receptor to bind to a range of similar antigens and vice versa. Noting the cited works, these concepts and ideas are further discussed in the remainder of this section.

One of the main motivations behind these approaches is that the traditional view of computer security is not likely to be able to claim complete victory in this battle. This is mainly related to the fact that the key assumptions of the traditional view are all false in practice. Such assumptions include:

- Security policy can be explicitly specified,
- Programs can be correctly implemented, and
- Systems can be correctly configured;

Computers are dynamic systems; manufactures, users, and system administrators constantly change the system state. Formal verification of such a dynamic system is not practical. Without a formal verification many of the tools such as encryption, access control, audit trails, and firewalls all become questionable. In turn, this means that perfect implementation of a security policy is impossible, resulting in imperfect system security.

4.1 Natural and Artificial Immune Systems

A better computer security system may be achieved by adopting the design principles from the natural immune systems, which solve similar type of problems but with radically different approaches from those used in traditional computer security. Such a system would have highly sophisticated notions of identity and protection that provides a general-purpose protection system to complement the traditional systems. The natural immune system tries to distinguish ‘self’ from the dangerous ‘other’ or ‘nonself’ and tries to eliminate the ‘other’. This can be viewed as a similar problem in computer security; where ‘nonself’ might be an unauthorized user, computer viruses or worms, unanticipated code in the form of Trojan horse, or corrupt data.

The natural immune system provides defense at many levels. The first barrier to infection is the skin. The second level is a physiological barrier, where pH, temperature, and similar conditions cause inappropriate living environments for some of the foreign organisms. The innate immune system and the adaptive immune response will handle those foreign organisms that pass these barriers and enter the body. The innate immune system primarily consists of circulating scavenger cells that ingest extra cellular molecules and material. The adaptive immune response is also called the ‘acquired immune response’, as it is the immunity that is adaptively acquired during the life of the organism. This is the most sophisticated system that also provides the most potential for computer security.

The adaptive immune system is essentially a distributed detection system primarily consisting of white blood cells, or lymphocytes. Lymphocytes circulate through the body and act as small detectors. They are viewed as negative detectors, because they recognize nonself patterns, ignoring self patterns. To achieve their tasks, the surfaces of lymphocytes are covered with receptors. Detection occurs when molecular bonds between a pathogen and receptors are formed. The strength of the bond is dependent on how complementary the molecular shape is with respect to the receptor. It also depends on electrostatic surface charge between the pathogen and the receptor. Detection is approximate allowing a lymphocyte to bind with several different types of structurally related pathogens.

The required diversity of lymphocyte receptors is achieved by generating them through a genetic process that introduces huge amounts of randomness. The randomness on the other hand, can result in production of lymphocytes that detect self instead of nonself. To provide tolerance of self, lymphocytes mature in an organ called thymus through which most self proteins circulate. While maturing, if any lymphocyte binds to these self proteins, they will be eliminated.

Lymphocytes are typically short-lived and are continually replaced by new ones, with new randomly generated receptors. In this way the coverage provided by the immune system over time increases; the longer a pathogen is present in the body the greater is the chance of its detection as it will encounter a greater diversity of lymphocytes. Additionally, through learning and memory, protection is made more specific. If the immune system detects a pathogen that it has not encountered before, it undergoes a primary response. During this process, it learns the structure of the specific pathogen through evolving a set of lymphocytes with high affinity for that pathogen.

Obviously, the coverage of the immune system is not complete. It is interesting to note that due to the uniqueness of the immune systems, the degree of vulnerability of any individual to a given pathogen is different from that of any other individual. The result of this diversity of immune systems across a population is a great improvement of the chances for the survival of the population as a whole. Based on this very brief overview and the discussions in the previous sections, it is easy to see that the natural immune system has many features that are desirable to be implemented in a computer security system. Some of them can be summarized as:

- The protection system is multi-layered and non-monolithic in the sense that there is no periphery in which all activity is trusted.
- The detection system is massively parallel and truly distributed in functioning.
- The system is autonomous; no centralized control to initialize the detection or manage the response.
- Individual components are disposable and unreliable, yet the system as a whole is robust.
- Detection of previously encountered infections is quick with aggressive response against them.

- Novel intrusions are detected through a variety of adaptive mechanisms.

- Each copy of the detection system is unique; the diversity means that pathogens that evade one immune system are not necessarily able to evade other immune systems.

- The immune system provides a dynamically changing coverage. Maintaining a set of detectors large enough to cover the space of pathogens is impossible, so the system maintains a continually changing detector repertoire, which circulates throughout the body.

- The detection is imperfect allowing for increased flexibility in allocation of the resources.

These features can be viewed as guidelines for design of computer security systems inspired by natural immune system. That is, utilizing some techniques that are directly related to some mechanism from immunology can incorporate some of these features. For some others, new algorithms may need to be developed. The primary emphasis is not on mimicking the natural immune system, but rather capturing those aspects that can help in building a robust adaptive distributed computer security system. A specific example may be helpful in demonstrating how some of these ideas can be implemented in the computer security area.

4.2 Network Intrusion Detection

With fundamental differences between living organisms and computer systems, it is far from obvious how the natural immune systems can be used as models for building competent computer intrusion detection systems. While some of the described ideas have been implemented and reported in the relevant literature, many of the appealing parts are still at their theoretical stages. In this part, expanding on the works reported in [13], the outline of the artificial immune system in the context of a specific application area is presented. The specific problem considered here, is related to protecting a local area network (LAN) from network-based attacks. A LAN has the convenient property that every node on the network segment can see every packet passing through the LAN.

In this domain, ‘self’ is defined as the set of normal pair-wise connections between computers, at the TCP/IP level. This includes connections between two computers in the LAN as well as connections between a computer in the LAN with an external computer. Each connection is defined in terms of its data-path triple consisting of: the source IP address, the destination IP address, and the service or port by which the computers communicate. This information is compressed to a single 49-bit string that unambiguously defines the connection. Self is therefore the set of normally occurring connections observed over time on the LAN. In a similar way, nonself is a set of connections, using the same 49-bit presentation, with the difference being that nonself consists of those connections that are not normally observed on the LAN. Note that the nonself set is potentially enormous.

A single bit string of 49 bits and a small amount of state also represent each detector cell. In effect this will represent the receptor region on the surface of a lymphocyte. This region detects and binds to foreign material through the recognition process. There are many ways for carrying out the recognition process, some of which have been outlined in the earlier parts of this report. For instance, production rules, neural networks, or string matching approaches can implement the detection or recognition. In string matching, detector *d* and string *s* will match through some matching rules, Hamming distance or r-contiguous bits being examples of such rules.

The detectors are grouped into sets, one set per machine or host on the LAN. With the broadcast assumption, each detector set is constantly exposed to the current set of the connections on the LAN. The detector uses this set as a dynamic definition of the self. Note that the observed connections in a fixed time-period are analogous to the lymphocyte being exposed to a set of proteins in thymus over some period of time. Within each detector set, new detectors are created randomly and asynchronously on a continual schedule. These new detectors remain immature for some period of time, during which they have the possibility of matching any current network connection. If the detector matches any connection while it is immature, it is deleted. This is similar to negative selection process in the natural immune system.

A potential problem with this approach is that a nonself packet arriving during negative selection can cause immature detectors to be wrongly eliminated. However, by noting that nonself packets are rare, there are probably other mature detectors available for their detection. This is a small loss of efficiency, because of deleting valid detectors, but the function is preserved.

The lifecycle of a detector is summarized in Figure 1, which is adapted with some variations and extensions from [13]. As can been seen from that figure, detectors that survive the initial phase are promoted to 'mature detectors'. Each mature detector is a valid one that acts independently. If a mature detector matches a sufficient number of packets, an alarm is raised. Note that a detector must match a number of times before it is activated. This is referred to activation threshold, which is implemented to lower the false positive rate of the detection system. Here, false positives arise if the system is trained on an incomplete description of the self, and then the detectors encounter new but legitimate patterns. Through activation threshold implementation, the system is capable of tolerating such legitimate patterns, but still detects abnormal activity.

A mature detector is considered to be a 'naïve detector' before it goes through a further learning phase. At the end of this phase, if the detector has failed to match a packet it is deleted. On the other hand, if it has matched a sufficient number of nonself packets, it becomes a 'memory detector' with an extended lifetime. Memory detectors have a lower threshold of activation, thus implementing a 'secondary response' that is more sensitive to previously seen nonself strings. Although these memory detectors are desirable, a large fraction of naïve detectors must always be present. This is because the naïve detectors are necessary for the detection of novel foreign packets; i.e. they are needed for anomaly detection.

5 Concluding Remarks

In terms of networked systems, any action that attempts to compromise the integrity, confidentiality, or availability of a resource is defined as an intrusion. These security breaches can be considered in two main categories of misuse intrusions and anomaly intrusions. As misuse intrusions follow well-defined patterns, they can be detected by performing pattern matching on audit-trail information. Anomalous intrusions are detected by observing significant deviations from normal behavior. Anomaly detection can also be performed using other mechanisms, including neural networks, machine learning classification techniques, and approaches that are based on design concepts inspired by biological immune systems. Anomalous intrusions are harder to detect, mainly because they do not show fixed patterns of intrusion. Therefore, for this type of intrusion detection approaches that are based on approximate reasoning are more suitable. A system that is based on a combination of the alertness of a computer program, artificial immunology, and human-like capabilities in handling imprecision and adaptive pattern recognition has obvious advantages.

There is an urgent need for further work on exploring the ways that artificial intelligence techniques can make the intrusion detection systems more efficient. Additionally, the capabilities of fuzzy logic in using the linguistic variables and fuzzy rules for analyzing and summarizing the audit log data need to be investigated. More specifically, intelligent approaches that utilize the protection concepts of natural immune systems to capture those aspects that can help in building a robust adaptive distributed computer security system need to be further investigated.

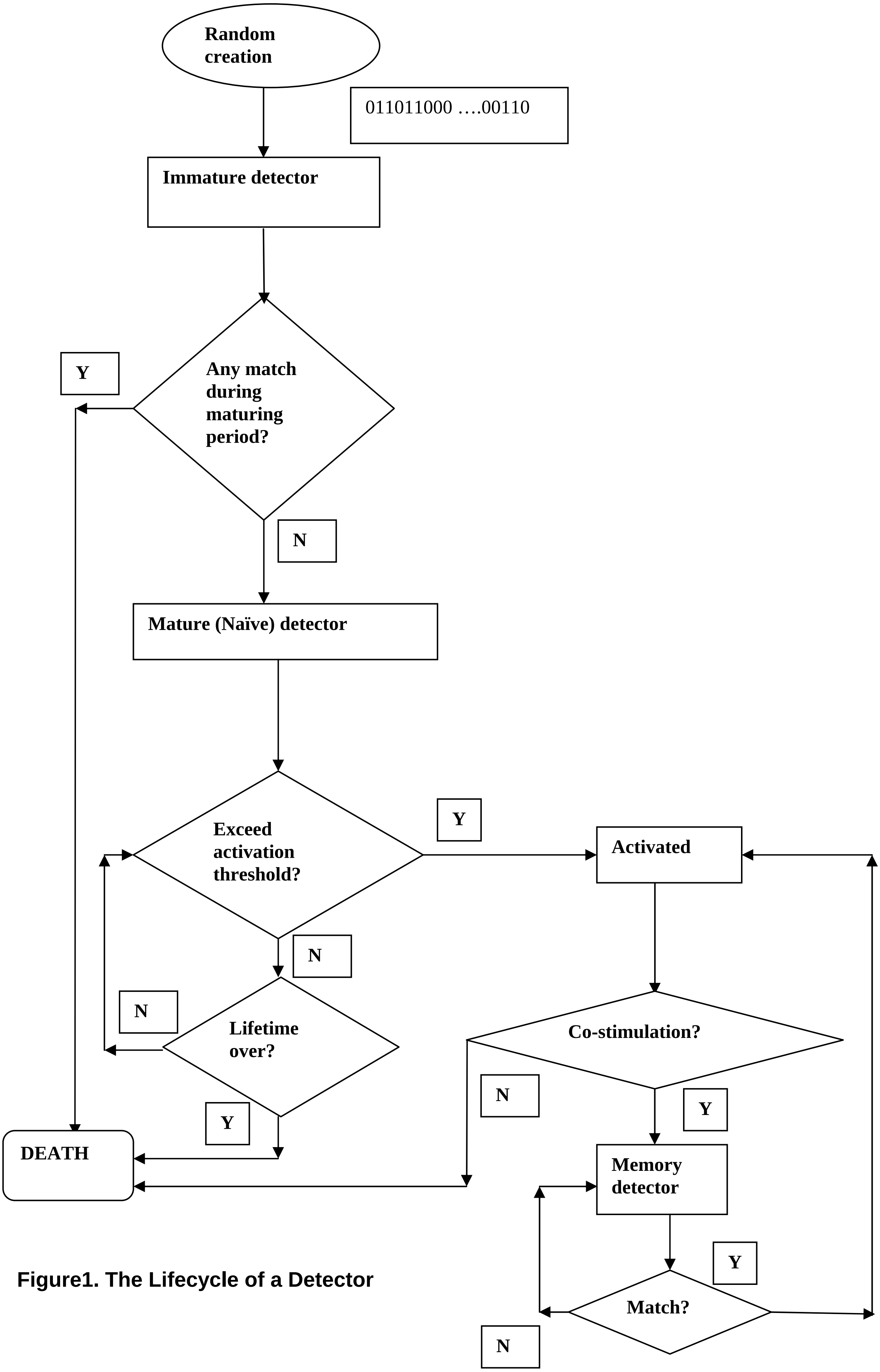

Figure1. The Lifecycle of a Detector

References:

[1] L. Zhuowei, A. Das, and Z. Jianying Zhou, “Theoretical basis for intrusion detection,” *Proc. Sixth IEEE SMC*, 2005, pp184-192.

[2] B. Mukherjee, L. Heberlein, and K. Levitt, “Network intrusion detection,” *IEEE Network*, pp. 26 – 41, May-June 1994.

[3] D. Denning, “An intrusion-detection model,” *IEEE Trans. Software Engineering*, Vol. 13, No. 2, pp. 222, February 1988.

[4] S. Shahrestani, H. Yee, and J. Ypsilantis, “Adaptive recognition by specialized grouping of classes,” in *Proc. 4th IEEE Conference on Control Applications*, 1995, pp. 637-642.

[5] J. Kephart, “A biologically inspired immune system for computers,” in *Artificial Life: Proc. International Workshop on the Synthesis and Simulation of Living Systems*, R. Brooks and P. Maes, Eds. Cambridge, MA: MIT Press, 1994.

[6] S. Forrest, S. Hofmeyr, and A. Somayaji, “Computer Immunology,” *Communications of the ACM*, Vol. 40, pp. 88-96, 1997.

[7] K. V. Rajkumar, V. Vaidehi, S. Pradeep, N. Srinivasan, and M. Vanishree, “Application Level IDS using Protocol Analysis,” *Proc. IEEE International Conf on Signal Processing, Communication, and Networking*, 2007, pp. 355-359.

[8] W. Yang, W. Wan, L. Guo, and L. Zhang, “An Efficient Intrusion Detection Model Based on Fast Inductive Learning,” *Proc. IEEE International Conf on Machine Learning and Cybernetics*, 2007, pp. 3249-3254.

[9] F. Cohen, “Computer viruses,” *Computers and Security*, Vol. 22, pp. 22-35, 1987.

[10] S. Forrest, A. Perelson, L. Allen, and R. Cherukuri, “Self-nonself discrimination in a computer,” *Proc. IEEE Symposium on Research in Security and Privacy*, 1994.

[11] S. Forrest, A. Somayaji, and D. Ackley, “Building diverse computer systems,” *Proc. 6th Workshop on Hot Topics in Operating Systems*, 1997.

[12] H. Xie, and Z. Hui, “An intrusion detection architecture for Ad-hoc networks based on artificial immune systems,” *Proc.7th International Conf. Parallel and Distributed Computing*, 2006, pp.1-4.

[13] S. Hofmeyr, S. Forrest, “Immunity by design: An artificial immune system,” *Proc. Genetic and Evolutionary Computation Conference*, 1999, pp. 1289-1296.

[14] Z. Yu-Fang, X. Zhong-Yang, and W. Xiu-Qiong, “Distributed intrusion detection based on clustering,” *Proc. IEEE International Conf on Machine Learning and Cybernetics*, 2005, pp. 3279-3283.

[15] D. Endler, “Intrusion detection: Applying machine learning to Solaris audit data,” *Proc. 14th Computer Security Applications Conference*, 1998, pp. 268 – 279.

[16] K. Ilgun, R. Kemmerer, and P. Porras, “State transition analysis: a rule-based intrusion detection approach,” *IEEE Trans. Software Engineering*, Vol. 21, pp. 181 – 199, March 1995.

[17] M. Esmaili, B. Balachandran, R. Safavi-Naini, and J. Pieprzyk, “Case-based reasoning for intrusion detection,” *Proc. 12th Computer Security Applications Conference*, 1996, pp. 214 – 223.

[18] L. Wenke, S. Stolfo, and K. Mok, “A data mining framework for building intrusion detection models,” *Proc. IEEE Symposium on Security and Privacy*, 1999, pp. 120 – 132.

[19] L. A. Zadeh, “Soft computing and fuzzy logic,” *IEEE Software*, Vol. 11, no. 6, pp. 48-56, Nov. 1994.

[20] L. A. Zadeh, “Fuzzy Sets,” *Information and Control*, Vol. 8, pp. 338-353, 1965.