Evolution Induced Secondary Immunity: An Artificial Immune System based Intrusion Detection System

Divyata Dal[1], Siby Abraham[2], Ajith Abraham[3], Sugata Sanyal[4] and Mukund Sanglikar[5]

[1]University Department of Computer Science, University of Mumbai, India
[2]Department of Mathematics, Guru Nanak Khalsa College, University of Mumbai, India
[3]Norwegian center Of Excellence, Q2S, Norwegian University of Science and Technology, Norway
[4]School of Tech. and Computer Science, Tata Institute of Fundamental Research, Mumbai, India
[5]University Department of Computer Science, University of Mumbai, India
divyata1711@gmail.com, sibyam@gmail.com, ajith.abraham@ieee.org, sanyal@tifr.res.in, masanglikar@rediffmail.com

Abstract

The analogy between Immune Systems and Intrusion Detection Systems encourage the use of Artificial Immune Systems for anomaly detection in computer networks. This paper describes a technique of applying Artificial Immune System along with Genetic algorithm to develop an Intrusion Detection System. Far from developing Primary Immune Response, as most of the related works do, it attempts to evolve this Primary Immune Response to a Secondary Immune Response using the concept of memory cells prevalent in Natural Immune Systems. A Genetic Algorithm using genetic operators- selection, cloning, crossover and mutation- facilitates this. Memory cells formed enable faster detection of already encountered attacks. These memory cells, being highly random in nature, are dependent on the evolution of the detectors and guarantee greater immunity from anomalies and attacks. The fact that the whole procedure is enveloped in the concepts of Approximate Binding and Memory Cells of lightweight of Natural Immune Systems makes this system reliable, robust and quick responding.

1. Introduction

An intrusion is defined as any set of actions that attempt to compromise the integrity, confidentiality or availability of a resource [1]. Intrusion Detection Systems (IDS) are developed to safeguard the computer network from these attacks.

In this paper, we attempt to use Artificial Immune (AIS) System and Genetic Algorithm (GA) to develop an IDS, which is host based and uses anomaly detection. The proposed technique monitors all incoming requests on the host and blocks it in case of an anomaly, thereby making the system reactive. We use AIS to develop both Primary and Secondary Immune Responses. Genetic Algorithm facilitates the evolution of Secondary Immune Response from the Primary Immune Response. The concept of lightweight, which is an important feature of Natural Immune System, has been incorporated during the Primary and Secondary Immune Responses. The uniqueness of this paper is the composite effect of Genetic Algorithm and Artificial Immune System to develop a Secondary Immune Response from Primary Immune Response, the effectiveness of which is validated by experimented results.

The different approaches to IDS have been narrated in various papers [2][4][10][14][15]. Li [3] applies GA in IDS by modeling network connection information as 57 gene chromosomes with hexadecimal representation. The novel approach of Artificial Immune System was developed to overcome the weaknesses of Network-based IDS's. An Artificial Immune System framework called LISYS introduced by Forrest et al. [5] is specialized for the problem of Network Intrusion Detection. It uses a 49-bit compressed representation of TCP SYN packets introduced by Hofmeyr [1][5][6][7]. Hofmeyr and Forrest [5] discuss a secondary response, which is similar to our work. They achieve the Secondary Response in two steps. In the first step, the response of the mature detectors has been provided with an extended lifetime during the training stage. In the second step, a human intervention not a system intervention, leads to the formation of a memory detector.

This paper is organized as follows: Section 2 discusses the system we have implemented. Section 3 analyzes the experimental results and finally Section 4 presents the concluding remarks of this work.

2. IDS-EVOLUSIRS

2.1. System Overview

Intrusion Detection System – Evolutionary Secondary Immune Response System (IDS-EVOLUSIRS) is developed in four stages:

1) Data Conversion
2) Generation and Training of Detectors
3) Intrusion Detection
4) Memory Cell Identification

Embedded in each stage is the concept of lightweight of the Natural Immune System. It has been implemented as follows:

Approximate Binding. A single detector is capable of detecting any number of intrusions as long as the affinity of the detector-intrusion binding is above a particular threshold, thus enhancing the detection capability of the intrusion detection system [11].

Memory Cells. A detector stores information about previously detected intrusions. Storing these as memory cells enables the system to respond quickly when the same intrusions are encountered in the future [11][12].

Gene Expression. Detector diversity can be maintained by generating a vast number of detectors from new combinations of segments stored as memory cell, ensuring the effective detection of a wide variety of intrusions [9][12].

The concept of Approximate Binding is incorporated in the training phase of detectors, whereas the Memory Cells concept has been implemented in the detection stage of the system. The concept of Gene Expression has not yet been implemented in our system, but we plan to incorporate it in the next phase of our system.

Table 1. Representation of fields

Name of the Field	Minimum and Maximum Value	Binary Strings Length
Destination IP Address	0.0.0.0 - 255.255.255.255	38 bits
Source IP Address	0.0.0.0 - 255.255.255.255	38 bits
Dest. Port No	0 – 65535	16 bits
Duration	0 – 999 seconds	10 bits
Protocol	0 – 65535	16 bits
Source Port No	0 – 65535	16 bits

2.2. Data Conversion

The IDS-EVOLUSIRS uses two datasets self and non-self. The data is taken from the DARPA dataset, which is a DARPA/MIT Lincoln Laboratory off-line intrusion detection evaluation data set [13]. The following fields have been considered in the order mentioned in [3].

1. Destination IP Address
2. Source IP Address
3. Destination Port Number
4. Duration
5. Protocol
6. Source Port Number

The requests are converted into binary strings of length 134 by concatenating the fields in the order mentioned, padding it with zeros wherever necessary. The maximum value of each of the fields along with the length of their equivalent binary strings is listed in Table 1:

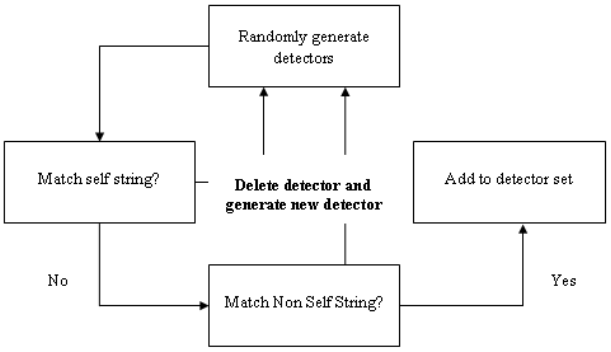

Figure 1. Negative selection

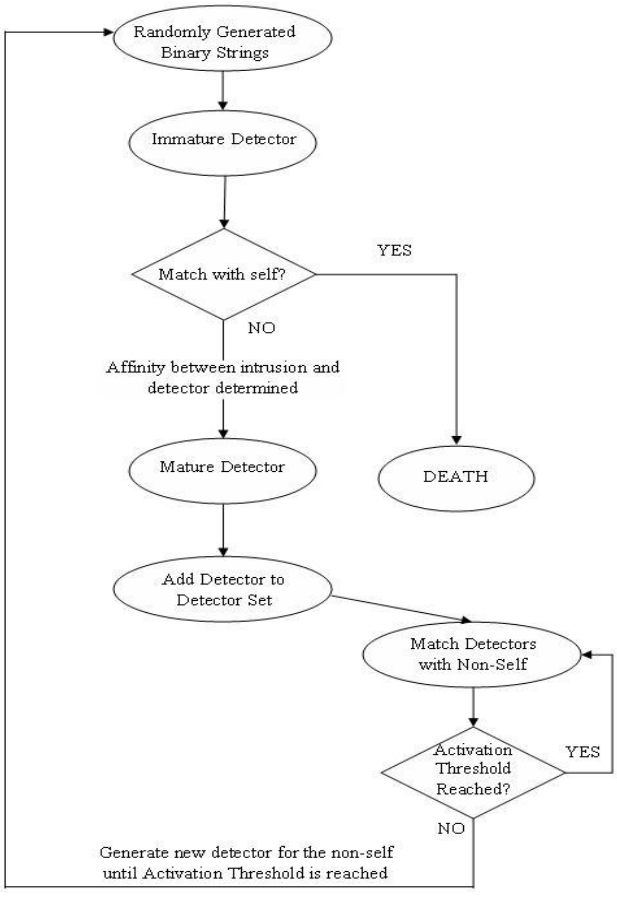

Figure 2. Generation and training of detectors

2.3. Detector Generation

The detectors are represented as a set of randomly generated binary strings that are trained to differentiate between the self and non-self connections. Training is

done using the Negative Selection algorithm with the intention of refining the detector set against the self and the non self (intrusions). The refinement procedure, as shown in Figure 1, uses a variation of the pattern matching algorithm known as the r-Contiguous bits algorithm. It discards any detectors matching the self, thereby generating a new detector in its place.

The output is then trained against the non-self connections using the same algorithm. The number of contiguous bits ('r') matching the detector determines the *fitness* of that detector, which in turn determines the amount of affinity between the detector and the anomaly string. Once the detector has been presented to all the self and non-self connections, it forms the "Mature Detector Set", as shown in Figure 2, and is not subject to further change. This detector set is used in the Primary Response of the IDS-EVOLUSIRS.

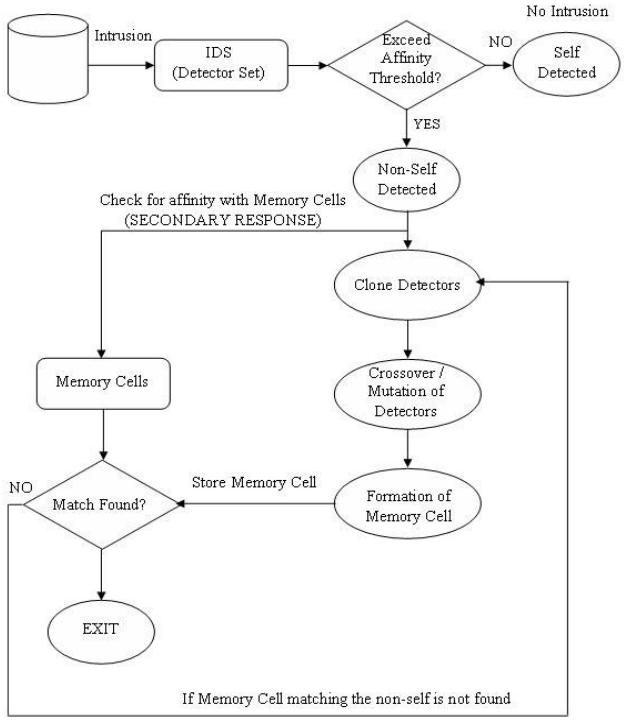

Figure 3. IDS-EVOLUSIRS

2.4. Intrusion Detection

Once the training of the detectors is complete, it is now ready to face a real time connection. On facing a typical request from an anonymous and external node, the system evaluates this request using the concept of the fitness value to determine whether the request is an anomaly or not. Higher the fitness value of the request, greater is the possibility that the incoming request is an anomaly. In case of IDS-EVOLUSIRS, this idea has been extended. If a match is found at 13 contiguous locations, we classify it as a hit, 13 contiguous locations being specific to our system. But, the system is activated only if 3 or more sister detectors are activated by the request. Thus, a request matching 3 or more detectors at 13 or more contiguous locations classifies it as an anomaly as shown in Figure 3.

2.5. Memory Cell Identification

The adaptive and evolutionary property of Genetic algorithms has been used to evolve the highly fit sister detectors activated when an anomaly has been encountered. The genetic operators – selection, cloning, crossover and mutation - have been used for this purpose.

When an anomaly is encountered, the sister detectors activated as a result is called the set of "Activated Detectors", which are candidates for memory cells. Then, the genetic operator of selection is applied to determine which of these detectors should be cloned. The cloning threshold is set by the following formula:

$$\text{Cloning Threshold} = \frac{\text{Sum of fitness of all the detectors}}{\text{Total number of detectors}}$$

Those activated detectors having a fitness value greater than or equal to the cloning threshold undergo the cloning. The number of clones to be generated for the candidate detectors is determined by the following formula:

$$\text{Number of Clones} = \text{Int}\{\text{Fitness of detector} * 10 / \text{Total Fitness}\}$$

Once the process of cloning is complete, the clones and the remaining activated detectors together form the set of "Winner Detectors".

Subjecting these *Winner Detectors* to the genetic operators of *Mutation* and *Crossover* facilitates the evolution of these detectors. After a substantial number of generations, the detector with fitness value greater than all the *Winner Detectors* is treated as a "Memory Cell".

3. Experimental Results

The dataset used for the evaluation of our Intrusion Detection System is the 1998 DARPA Intrusion Detection Evaluation Data Set. Although this dataset is quite old, it is nevertheless widely used to evaluate intrusion detection systems. The 'tcpdump.list' files have been used for training as well as testing of the system.

3.1. Experimental Setup

As stated in the Offline Evaluation Plan of the DARPA Dataset, of the seven weeks of the 1998 dataset, the first six weeks of data are used as training. This comprises of 30,000 self records and 165 non self. The seventh week of data is used to test the performance of the IDS size of which is 5000 with 4019 self and 981 non-self. The entire set of data is converted into binary strings and a set of 100 binary strings is randomly generated to represent potential detectors, each with the same length (134 bits) as the data in the training dataset.

The Negative Selection algorithm used to train the detectors uses the r-Contiguous bits algorithm to refine the data against the non-self data set. The fitness 'r' of a detector is defined as the number of contiguous matching bits of the strings. A number of different values of 'r' have been tried. We have observed that for any value less than or equal to 12, even the self data matches the detectors. Further, for any value greater than or equal to 14, all the non-self data fails to match the detectors, and the "Mature Detector Set" is not formed. In either case, the Negative Selection algorithm fails. Thus, we have hit upon a unique value of 'r' equal to 13, where IDS-EVOLUSIRS is successful. The condition of activation of a detector is that it must match at 13 or more contiguous locations. Thus we have a single detector, which is capable of detecting any number of intrusions as long as the affinity of the detector-intrusion binding is above this threshold value 13. This implements the concept of approximate binding. However, for the recognition of a string as non-self, after experimental results, we decided that a minimum of 3 detectors must be activated by the request. Therefore, an incoming request is classified as an anomaly only when it matches at least 3 detectors, at at least 13 or more contiguous locations. The detectors are generated till these two conditions are met.

During training of the detector set, if a non-self (anomaly) fails to match 3 or more detectors, the non-self is classified as a hole. The proposed solution to this problem is to randomly generate detectors till the anomaly matches at least 3 detectors, therefore overcoming the problem of holes.

Once training is complete, the system is now ready to face real time requests. If a incoming request is classified as an anomaly, the detectors are activated, constituting the "Activated Detector Set" which undergoes the genetic operations of selection, cloning, crossover and mutation. After experimental analysis, the probabilities of mutation and crossover have been fixed to 0.3 and 0.7 respectively. A random number is generated, and depending on its value, the selected detector or detectors undergo mutation or crossover respectively. Mutation is performed by randomly selecting a detector from the activated set and deliberately complimenting the bits between two randomly selected locations of the detector. Two-Point-Crossover is performed on any two randomly selected detectors by swapping the bits between two randomly selected crossover points.

The above process of applying the genetic operators continues till a detector having a fitness value greater than all those in the "Winner Detector Set" is generated. This detector then becomes a *Memory Cell*, and is stored in a separate file aloof from the population of the detectors. This memory cell is used to generate Secondary Immune Response, should a similar anomaly attack the system in the future. Detector diversity has been maintained by creating a memory cell from the fittest detectors ("*Winner Detectors"*) formed as a result of the process of Genetic algorithm. During this process, the detectors mutate and crossover, exchanging effective detector fragments, resulting in the formation of a memory cell, which has a higher fitness value than the *"Winner Detectors"* participating in this process. This memory cell, therefore, ensures more effective detection. In this manner we implement the concept of *Memory Cells* in the system.

Table 2. Content of distinct memory cells

Dest IP	Src IP	Dst Port	Conn Duration	Proto-col	Src Port	MC Found?	MC Fit-ness
208.239 003..255	202.072. 001. 077	138	0:0:1	207/ u	138	NO	28
172..016 112..050	202.072. 001.077	7	0:0:1	eco/i	7	NO	37
202..072 001..077	172..016 112..050	7	0:0:1	ecr/i	7	NO	22
172..016 112.050	010..000 001.020	514	0:0:1	syslo g	514	NO	32
172.016 114.050	202..049 .244.010	143	0:0:1	imap	0	NO	38

3.2. Experimental Discussions

Distinct memory cells. On performing the experiment using the "Mature Detector Set" on the test data, comprising of both self as well as non self connections, we conclude that the system was able to correctly classify all the test data. The outcome of the experiment was to have a set of memory cells of size 66. (This is subject to slight variance each time the experiment is carried out owing to the random nature of genetic operators)

The Table 2 summarizes the Primary Immune Response, listing the first 5 detectors saved as memory cells formed during exposure of the system to the test data. A "NO" in column 7 of the table indicates the formation of a new memory cell. On the other hand, if a previously seen intrusion is encountered, it triggers the Secondary Immune Response, resulting in a memory cell detecting it, indicated by a "YES" in column 7. The formation of memory cells was the result of the use of Genetic algorithms for the evolution of the *Winner Detectors*. Without the use of Genetic algorithms in AIS, we would only have a history of past attacks, and only attacks completely resembling the known attacks could have been prevented.

Connections with same fitness value. During experiments, it occurred to us that it was possible to have memory cells with the same fitness values, but different connections. Figure 4 shows the fitness values and the corresponding number of distinct connections of 66

distinct memory cells formed during the test period. The problem of having the same fitness value with distinct connections has been addressed by considering matching of contiguous locations in addition to the usual comparison of the fitness values. This helps in detecting the already seen attacks in a much more reliable way.

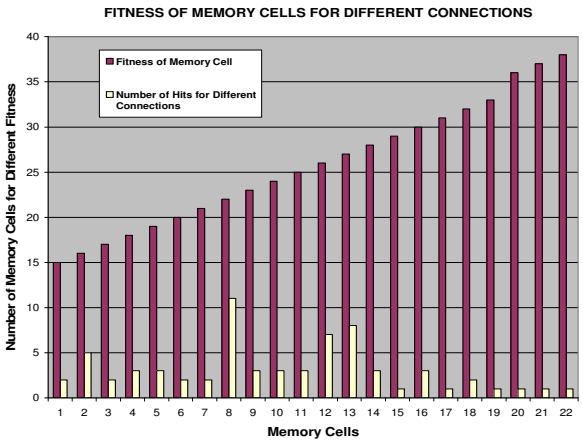

Figure 4. Number of connections with the same fitness

Dependence of memory cells on a particular field of the connection. From Table 2, we conclude that the fitness of a memory cell would vary each time the experiment is run and that no specific field of the anomaly is said to be important in the formation of a new memory cell. However, it has been observed that, when an anomaly is encountered for which a memory cell is already present, most of the times, the *Destination IP* and *Source IP* remain unchanged. But, the number of memory cells formed for each *Source IP* and *Destination IP* address varies.

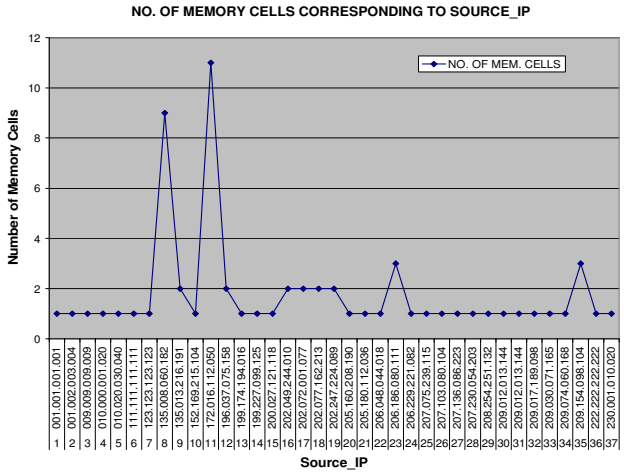

Figure 5. Number of memory cells corresponding to Source_IP

As Figures 5 and 6 illustrate, the number of Distinct Destination IP is less than the number of Distinct Source IP. This means that the Destination IP remains more or less constant over a number of Source IP.

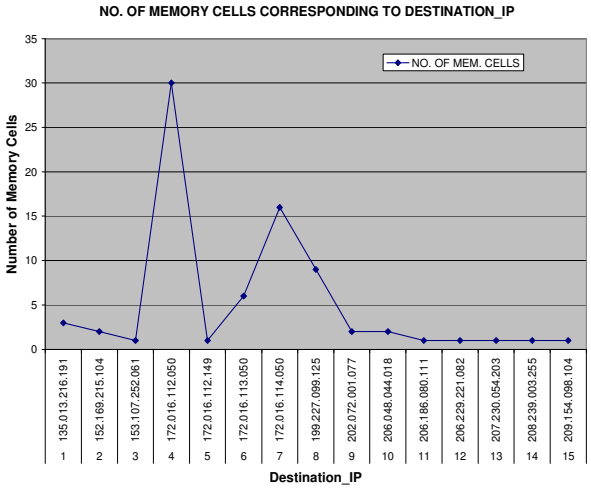

Figure 6. Number of memory cells corresponding to Destination_IP

Number of distinct memory cells. From experimental results, we observe that the DISTINCT memory cell that has the maximum number of hits is most of the times the one with fitness value equal to the MEAN fitness value. The MEAN fitness value is calculated by taking half the sum of the minimum fitness value and the maximum fitness value of the memory cells of the set. This can be observed from Figure 7, which shows that the detector with fitness value 22 is the one with the maximum number of hits.

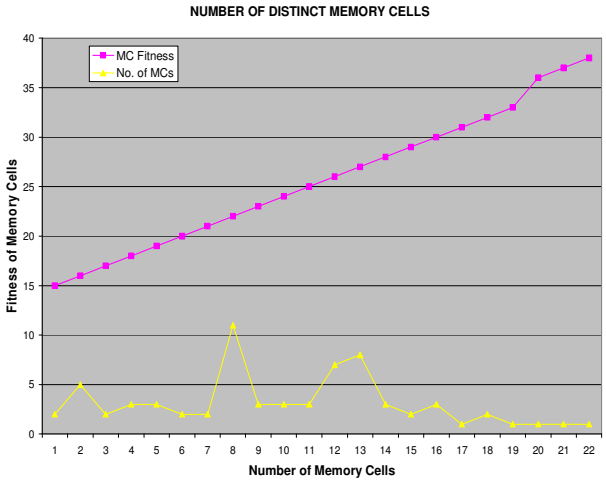

Figure 7. Number of distinct memory cells

The experimental results report the achievement of a superior anomaly detection rate, which was possible by the unique procedure involving GA and AIS. The trained

detectors could detect all the anomalies during the Primary Immune Response and the winner detectors could classify all the previously seen requests as anomalies during the Secondary Immune Response because of the GA facilitating the working of the AIS.

4. Conclusions

This paper illustrated the use of memory cells for developing a Secondary Immune Response in an Intrusion Detection System. Our work differs from other Intrusion Detection Systems in that it has encapsulated Secondary Immune Response by incorporating the use of memory cells that enable faster detection of an already encountered anomaly. Moreover, it is observed that the nature of memory cells is highly random and is dependent on the evolution of the detectors using genetic operators. This is possible because of the evolving nature of Genetic algorithms and the adaptability induced by Artificial Immune System. This random nature of memory cells makes the system less predictive and enhances the detection capability of the system to trap similar anomalies.

In our paper, we have provided a local solution for the problem of holes. In the future, we intend to provide a global solution for the same. Although our detector set was trained against a sufficiently diverse set of samples, it is possible to encounter a non-self pattern that cannot be detected by the existing detector set. Ultimately, this problem of holes can be handled if we have an expression code for anomalies. Thus, our future work will focus on development of such Gene Expressions, which will tackle the problem of holes globally.

References

[1] Steven Andrew Hofmeyr, "An Immunological Model of Distributed Detection and its Application to Computer Security" University of New Mexico. May 1999.

[2] Ajith Abraham, Crina Grosan, Carlos Martin-Vide, "Evolutionary Design of Intrusion Detection Programs", International Journal of Network Security, Vol.4, No.3, PP.328–339, Mar. 2007.

[3] Wei Li, "Using Genetic Algorithms for Network Intrusion Detection", Proceedings of the United States Department of Energy Cyber Security Group 2004 Training Conference, Kansas City, Kansas, May 24-27, 2004, CD ROM Proceedings, 8 pages

[4] J. Balthrop, F. Esponda, S. Forrest, M. Glickman, "Coverage and Generalization in an Artificial Immune System", Proceedings of the Genetic and Evolutionary Computation Conference, Pages: 3 – 10, 2002.

[5] S. Hofmeyr, S. Forrest, "Immunity by Design: An Artificial Immune System", In: Proceedings of the Genetic and Evolutionary Computation Conference, vol. 2, pp. 1289-1296.

[6] S. A. Hofmeyr and S. Forrest, "Architecture for an artificial immune system," Evolutionary Computation, vol. 7(1), pp. 45–68, 2000.

[7] A. Somayaji, S. A. Hofmeyr, and S. Forrest. "Principles of a computer immune system", *In Proceedings of the Second New Security Paradigms Workshop*, 1997.

[8] R. A. Goldsby, T. J. Kindt, B. A. Osborne, and W. H., Freeman. Kubi "Immunology", W. H. Freeman and Co., 5th ed edition, 2002.

[9] Tizard, I. R., "Immunology: Introduction", 4th Ed, Saunders College Publishing, 1995.

[10] Jungwon Kim and Peter Bentley, "The Human Immune System and Network Intrusion Detection", 7th European Congress on Intelligent Techniques and Soft Computing (EUFIT '99), Aachen, Germany, September 13- 19.

[11] Paul, W. E., 1993, "The Immune System: An Introduction", in *Fundamental Immunology* 3rd Ed., W. E. Paul (Ed), Raven Press Ltd.

[12] J. Balthrop, F. Esponda, S. Forrest, M. Glickman, "Coverage and Generalization in an Artificial Immune System", Proceedings of Genetic and Evolutionary Computation Conference (GECCO) 2002.

[13] DARPA/MIT Lincoln Laboratory off-line intrusion detection evaluation data set:

http://www.ll.mit.edu/IST/ideval/ (accessed on April 10, 2008)

[14] Sandhya Peddabachigari, Ajith Abraham, Crina Grosan and Johnson Thomas, Modeling Intrusion Detection System Using Hybrid Intelligent Systems, Journal of Network and Computer Applications, Elsevier Science, Volume 30, Issue 1, pp. 114-132, 2007.

[15] Yuehui Chen and Ajith Abraham and Ju Yang, Feature Deduction and Intrusion Detection Using Flexible Neural Trees, Second IEEE International Symposium on Neural Networks (ISNN 2005), Lecture Notes in Computer Science Vol. 3498, J. Wang, X. Liao and Zhang Yi (Eds.) Springer Verlag, Germany, pp. 439- 446, 2005.