Extended Thymus Action for Improving Response of AIS Based NID System against Malicious Traffic

M. Zubair Shafiq, Mehrin Kiani, Bisma Hashmi and Muddassar Farooq

***Abstract*— Artificial Immune Systems (AISs) are being increasingly utilized to develop Network Intrusion Detection (NID) systems. The fundamental reason for their success in NID is their ability to learn normal behavior of a network system and then differentiate it from an anomalous behavior. As a result, they can detect a majority of innovative attacks. In comparison, classical signature based systems fail to detect innovative attacks. Light Weight Intrusion Detection System (LISYS) provides the basic framework for AIS based NID systems. This framework has been improved incrementally, including incorporation of thymus action, since it was first developed. In this paper, we have extended the basic thymus action model, which provides immature detectors with multiple chances to develop tolerization to normal. However, AIS is prone to successful attacks by *malicious traffic* which appears similar to the normal traffic. This results in high number of false positives. In this paper, we present a mathematical model of malicious traffic for TCP-SYN flood based Distributed Denial of Services (DDoS) attacks. This model is used to generate different sets of malicious traffic. These sets are used for performance comparison of the proposed *extended thymus action* with the simple thymus action model. The results of our experiments demonstrate that the extended model has significantly reduced the number of false positives.**

I. Introduction

The basic aim of Intrusion Detection Systems (IDSs) is to detect malicious activities in computer systems. General Network Intrusion Detection (NID) systems can be classified into two major types namely signature-based NID systems and anomaly-based NID systems [10,11]. Signature-based schemes use expert systems, state-transition schemes and graphs. Signature-based NID systems look for attack signatures in network traffic and match them against a pre-configured set of intrusion signatures in a library [10]. If the incoming traffic matches a pattern in the existing library, an alarm is raised. A number of false alarms could also be raised if the matching algorithm exhibits deviation from exact signature matching [11].

In anomaly-based NID systems, a standard profile of the system is created, which describes normal or expected behavior. Deviation from this behavior would be labeled as "intrusive" or at least as "suspicious" [9,13]. Anomaly-based detection schemes usually use some form of statistical analysis for traffic modeling [10]. The traffic model can be built manually or by using machine-learning techniques. Less human effort is required if the profile has been created using machine-learning techniques. But it is more sensitive to innovative attacks or attacks whose signatures have not been stored in the system's library [10].

AISs are anomaly based systems but are inspired by Biological Immune System (BIS) and are being increasingly used for anomaly-based NID. BIS has the remarkable ability to distinguish *non-self* from *self*. Kim and Bently have done preliminary empirical work in utilizing AISs for NID systems [15-17]. Hofmeyr and Forrest [2] have done significant amount of research in effective implementation of AIS for network based anomaly detection (in LISYS). The idea of AIS has also been explored earlier in general area of computer security by the authors in [18-20]. Thus, AIS is extensively used as a general pattern learning system that is distributed, robust and dynamic [1,4].

A requirement of anomaly-based NID systems is to create a precise profile of self. This demands "exhaustive" training data. *False positives* could also exist in the system which is a major performance bottleneck. This defines the ratio of normal packets, falsely detected as malicious packets. The number of false positives is a relevent parameter that defines the performance of an AIS system. Our proposed extended thymus action model significantly reduces false positives. The major contributions of the work proposed in this paper are:

1) Modeling and implementation of *extended thymus action*;
2) Development of a mathematical model for malicious traffic which is used to generate different traffic attack sets;
3) Realization of our new model in OMNeT++ and its performance evaluation using different attack scenarios with varying degree of malicious activity. The results obtained through extensive experiments clearly demonstrate that the proposed model achieves lower number of false positives in attack scenarios as compared to the system not utilizing our model;
4) Development of a prototype TCP framework that can measure the number of successfully established TCP connections between clients and server. The framework differs between normal connection requests as *legal connections* and malicious connection requests as *illegal connection* requests. The results obtained from this framework clearly demonstrate that AIS with our model can establish the same number of legal connection requests as in case of a normal scenario. However, it rejects almost all illegal connection requests in an attack

M. Zubair Shafiq, Mehrin Kiani and Bisma Hashmi are the undergraduate students at the Department of Electrical Engineering, National University of Sciences & Technology, Rawaplindi, Pakistan (emails: zubairshafiq@ieee.org, kiani013@gmail.com, b.hashmi@gmail.com).

Muddassar Farooq is the Asst. Professor at the Department of Computer Engineering, National University of Sciences & Technology, Rawalpindi, Pakistan (email: muddassar.farooq@udo.edu)

scenario. This clearly shows the effectiveness of our model.

The rest of the paper is organized as follows. In section II and section III, we briefly outline fundamental principles of BIS and AIS respectively to make the paper self-contained for the readers. In section IV we present the implemented architecture of AIS for prevention of DDoS attacks. In section V, we present the results for performance comparison of AIS utilizing extended thymus action and simple thymus action.

II. Biological Immune System

Immune system is a defense mechanism present in all living beings. It resists attacks from foreign substances, by recognizing all the external substances (pathogens) as antigens (Ag). The ability of BIS to distinguish non-self cells from self cells protects the body from external antigens. In BIS, lymphocytes (white blood cells) specifically have a feature which enables them to distinguish non-self from self. The part of an antigen that is recognized by a lymphocyte is called an *epitope*. Antigens may have a variety of epitopes [4].

Lymphocytes have different antibodies (Ab) on their surface. An antigen binds only to that antibody with which it makes a good *match*. When a match is made, lymphocytes start producing antibodies to counter the antigens. The first encounter between a lymphocyte and a given antigen is called the primary response. All subsequent encounters constitute the secondary response. Secondary response is better than the primary response due to *affinity maturation* [12]. Affinity maturation is an adaptive process for lymphocytes. It consists of two sub-processes:

1) Clonal Proliferation
2) Somatic Hypermutation

In *clonal proliferation*, a lymphocyte undergoes multiple divisions resulting in more lymphocytes that can then bind to antigens during the secondary response. During primary response, lymphocytes undergo "high rate point mutation which helps in receptor editing for increasing the antibody diversity and affinity" [4]. This process is called *somatic hypermutation*.

An interesting situation arises when lymphocytes tend to recognize body's own cells as antigens. This may result in an auto-immune response against body's own cells. BIS caters for this response by employing negative selection [15,16]. Randomly created lymphocytes mature in the thymus. This process of maturation is called *tolerization*, as shown in Figure 1. Lymphocytes that detect body's own cells die a natural death (apoptosis) during maturation [3]. Only those lymphocytes mature, which have tolerization to self-antigens. Some self-antigens may still not be present in thymus so auto-immune response could be triggered against them. To prevent this, lymphocytes require a *costimulation* signal to get activated [1,2]. In the next section, we will show the mapping of these biological concepts into the different components and algorithms in an AIS framework.

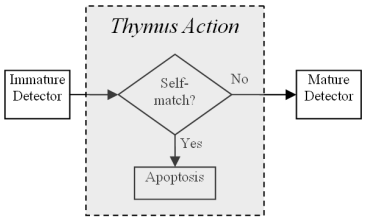

Fig. 1. Basic Thymus Action Model

III. Artificial Immune System

In [21], the generalized shape of a molecule in a *shape-space* is represented by an attribute string of length L. Therefore, an attribute string of length L can be regarded as a point m in n-dimensional shape-space, i.e. m ϵ U^n, where U represents the universal set of strings. A string may be represented by integers, binary numbers or even symbols. In most common shape-space models, peptide is modeled as a binary string (cardinality=2) of fixed length. Binary representation of peptide helps in preserving generality [21]. Therefore, if U represents the universal set containing all the strings then the necessary condition that must hold for this definition is,

$$U = S \cup N, S \cap N = \emptyset$$

where S represents self-set and N represents non-self set. Each antibody matches with that antigen, whose complement lies within a small surrounding region, characterized by a *cross reactivity threshold* [4]. Several matching techniques have been presented in [21] for matching a detector with the input string. Most relevant techniques are: hamming distance matching, manhattan distance matching, r-contiguous matching and euclidean distance matching. An *activation level* present in a detector should also increase by a given factor after every match. This level should reach a certain threshold in a fixed time interval to activate the detector. As proposed in [1], the activation level should be set to zero after the detector has been activated.

The errors produced by binary classifiers (AIS in this case) can be characterized into two groups: false positives and false negatives. False positive refers to a match made to a self cell and false negatives refer to a match not made to a non-self cell. In the next section we utilize these concepts of AIS for implementation of our extended thymus action model.

IV. Implementation of AIS for DDoS Prevention

In DoS attack, a malicious node can send additional control packets by spoofing the IP address of another node. As a result, this malicious traffic consumes the limited resources of a victim router/node so that it can no longer provide services to the legitimate nodes in the network. Defense schemes against such attacks include packet authentication, source identification and traffic filtering [22]. Packet authentication is not suitable for IP networks because it employs cryptographic signature based schemes that require a significant

amount of processing and bandwidth overheads [22]. Source identification uses link testing, packet logging or IP trace back schemes which require modification of existing routing infrastructure [22]. In comparison, traffic filtering does not suffer from the above-mentioned shortcomings of the other two techniques because it usually employs victim-end packet filtering. Therefore, for DoS attacks, this scheme is preferable over the others.

In DDoS attacks, a large number of control packets are simultaneously sent from multiple malicious nodes to a victim router/node so that the victim node and legitimate nodes can no longer communicate properly. Generally, malicious nodes in DDos attacks exploit shortcomings in networking protocols like Internet Control Message Protocol (ICMP). Relevant examples are SMURF attacks, Ping of Death attacks, TCP-SYN floods and UDP floods. In SMURF attack, ICMP echoes a request to the broadcast address with the victim's address as a source [5]. In Ping of Death attacks, ICMP packets with a payload of more than 64K are launched towards a victim node, which can crash the victim node running on earlier version of Windows and other operating systems [8]. In UDP floods, bandwidth is exhausted by sending a large number of bogus UDP packets [6]. In TCP-SYN floods, fake TCP connections are requested by malicious nodes by spoofing IP addresses of other nodes in the network [7].

The most common form of DoS attacks is TCP-SYN flood attacks. In these attacks, a malicious node floods the victim node, running a TCP server, by sending TCP-SYN packets with forged source addresses (a.k.a. *IP Spoofing*) [22]. Consequently, the server allocates resources for the request. The connection state is maintained till timeout. The server then sends back SYN_s+ACK_{c+1} (see Figure 2) packets while waiting for ACK_{s+1} packet, which never comes (see Figure 3). In DDoS version, the victim node running TCP server is flooded with SYN packets from various malicious nodes with spoofed IPs that have no common pattern. As a result, the resources of the victim node are exhausted resulting in denial of service to legitimate nodes, which are running TCP clients.

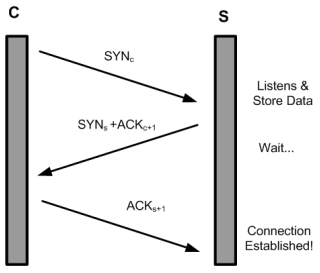

Fig. 2. Three Way TCP Connection Handshake

In [2], Hofmeyr and Forrest proposed a primitive architecture to cater for TCP-SYN flood based attacks. Their basic framework implemented as LISYS derived inspiration from ARTIS. In the next section, we briefly review their architecture and also present the improvements such as somatic

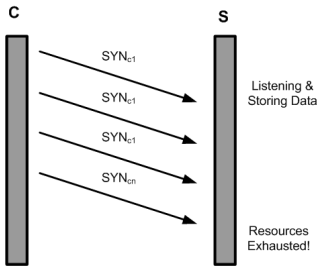

Fig. 3. TCP-SYN Floods

TABLE I
PEPTIDE REPRESENTATION

Bits	Bit-Length	Field Description
1-8	8	Least Significant Byte of Server's IP
9-40	32	Client's IP
41	1	Flag
42-49	8	Port Number

hypermutation (proposed and implemented in [1]) and our proposed extended thymus action. It is to be noted that the notion of thymus was not implemented in LISYS and was presented in [1].

A. Detector

A detector plays the combined role of both a lymphocyte and an antibody in AIS. It detects non-self antigens and acts against them. A population of various types of detectors, based on *least activation rule* [1], is stored in the memory.

B. Peptide Representation

Peptide representation consists of the following parameters:

- String Length (N)
- Cardinality (m)
- Fields

The selection of string length and fields depends on the type of information to be stored in an antigen. Cardinality is chosen to be two (m=2) because binary representation of IP addresses are used. DDoS prevention requires encoding of TCP-SYN packets. Hofmeyr proposed 49-bit string format for TCP-SYN packets that contains the fields given in Table I [2]. This scheme has shown better results as compared to other schemes [1].

C. Matching Methods

The system is tested using malicious traffic. Hamming and r-contiguous distances are compared with a reactivity threshold r such that $0 \leq r \leq l$. The Hamming distance (D) is defined as:

$$D = \sum_{i=1}^{L} \delta \;\; where\ \delta = \begin{cases} 1 & \text{if } \text{Ab}_i\text{==}\text{Ag}_i \\ 0 & \text{else} \end{cases}$$

r-contiguous distance is the same as hamming distance except that it looks for contiguous bit positions. A simple algorithm for L-bit *contiguous matching* is given in Algorithm 1.

Algorithm 1- Contiguous Distance Matching
Distance$_{max}$=0
Distance=0
for (First to Last Bit)
if (δ==1) **then**
Distance++
else
if (Distance$_{max} \leq$ Distance)
Distance$_{max}$ = Distance
end if
Distance=0
end else
end for
return Distance$_{max}$

D. Costimulation

Costimulation is provided by the AIS when the number of different matches increases above a particular threshold value. As a result, the response time of AIS to detect attacks and accordingly discard malicious packets is increased. This slow startup significantly helps in reducing *false positives* [1,2]. Figures 4 and 5 depict the effect of changing costimulation threshold on the percentage of packets dropped (due to matching between self and non-self strings) for hamming and r-contiguous matching techniques respectively. We can conclude from these figures that increasing costimulation threshold requires greater number of matches for dropping a packet which is helpful in reducing the number of false positives. However, on the other hand costimulation threshold should not be set too high because it may significantly increase the time to detect an attack. Therefore, we selected a costimulation threshold of 20 in our model.

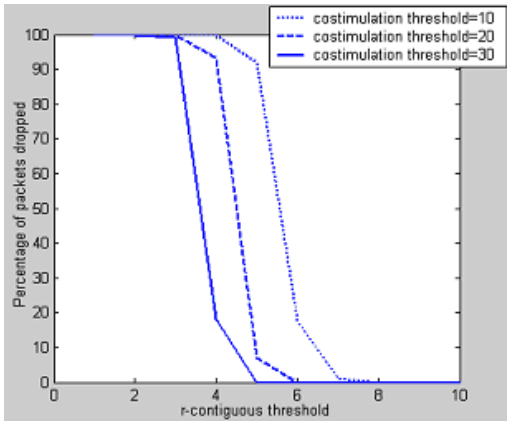

Fig. 4. Effect of changing costimulation threshold on percentage of packets dropped (r-contiguous matching)

E. Tolerization (Extended Thymus Action)

In section II, we highlighted the role of thymus in developing self-tolerization i.e. lymphocytes may not detect self-antigens. Thymus notion was presented in [2] but was not explicitly modeled in LISYS. In [1] the results of the implementation of thymus model for tolerization were presented. In [14], the authors also used self-tolerization principle

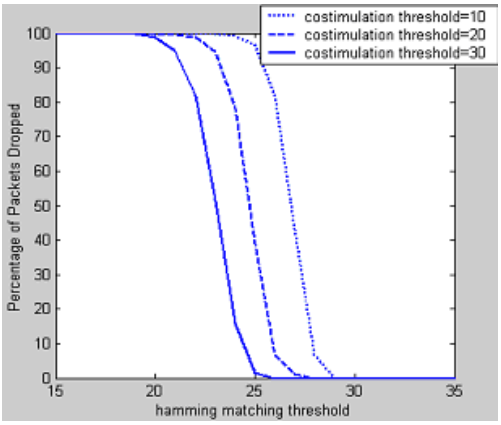

Fig. 5. Effect of changing costimulation threshold on percentage of packets dropped (hamming matching)

(using negative selection) to detect several types of network intrusions.

Now, we present an extension to the simple thymus model. The system is tuned for tolerization to self-data using *extended thymus action*. A randomly generated set of detectors is evolved for multiple generations in thymus. Negative selection principle is used to tolerize detectors to *self* [3]. If *'Generations == 0'*, and detector still has not developed tolerization for self-data, then the detector undergoes programmed cell death, also known as apoptosis. Otherwise detector fields are mutated randomly and are checked for self-match till 'Generations==0' or self-match does not occur (see Figure 7). The number of self-matches converge to zero for *Generations*$\gg$*1*. Figure 6 shows the convergence of detectors for different hamming thresholds. One can easily conclude that increasing the number of generations beyond 20 has little impact on the performance of the system.

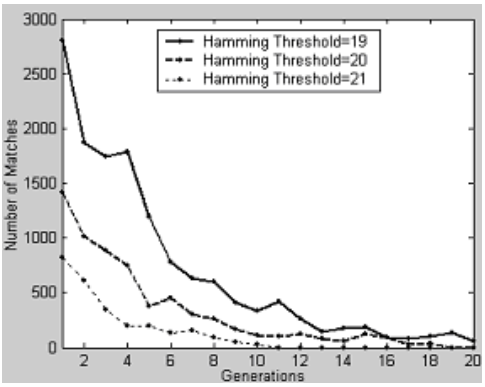

Fig. 6. Increased tolerization to self for multiple generations

In Figure 5, we need to select a suitable value of the hamming threshold that results in the optimum performance of our AIS. It is evident from the figure that a large value (about 25) of the hamming threshold significantly reduces the percentage of packets dropped. Therefore, we studied the impact of hamming distances threshold on true positive rate

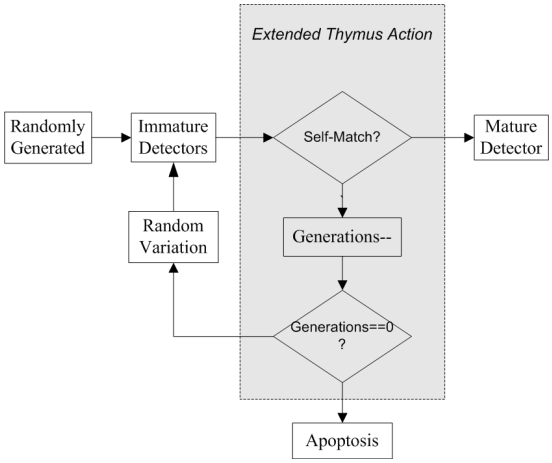

Fig. 7. Extended thymus action

and false positive rate. The results of this study are depicted in Figure 8. The important outcome of this study is that we need to maintain a hamming threshold between 20-23 for an optimum performance that results in 100% true positive rate and 0% false positive rate.

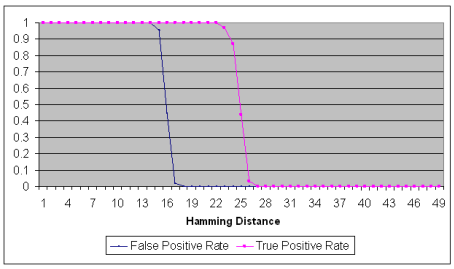

Fig. 8. Effect of changing Hamming Threshold on False Positive Rate and True Positive Rate

The same process was repeated for r-contiguous matching to select a suitable thershold value of 5. In section V, we compare extended thymus action with simple thymus action, and show that extended thymus action helps in reducing *false positives* due to increased self-tolerization.

F. Affinity Maturation

Once a detector detects a non-self antigen (malicious TCP-SYN packet), it reports this to a Central Alarm System (CAS). The detected packet is dropped only if CAS provides *secondary costimulation*. The original detector undergoes affinity maturation. Affinity maturation is an important part of AIS for improving immune response (*acquired immunity*). Affinity maturation consists of the following two processes:

1) Clonal Proliferation: A detector undergoes multiple replications during clonal proliferation which results in the formation of large population of the original detectors.

2) Somatic Hypermutation: A detector undergoes random mutation so as to differentiate themselves from multiple clones. This is to increase their diversity against a particular type of attack. Only those detectors are left to mature whose affinity with the detected packet is greater than or equal to an earlier affinity. In [1], the authors have shown that somatic hypermutation improves the secondary response of AIS against malicious traffic.

V. Results

This section describes the experiments performed in order to systematically study the impact of our *extended thymus action* in the AIS. We used the OMNeT++ simulator to simulate a DDoS attack on a targeted router. The attack scenario was simulated on NSFNet. Note that Router-n is represented as rte[n], where n is the id of the router (see Figure 9). All routers, except Router-5 and Router-13, simultaneously sent normal TCP-SYN packets to Router-2. In addition, Router-5 and Router-13 sent the malicious TCP-SYN packets to Router-2 which in this case is the victim router. This experiment did validate the operational accuracy of our AIS by showing that it can filter the malicious traffic (launched by Router-5 and Router-13) from the legitimate traffic.

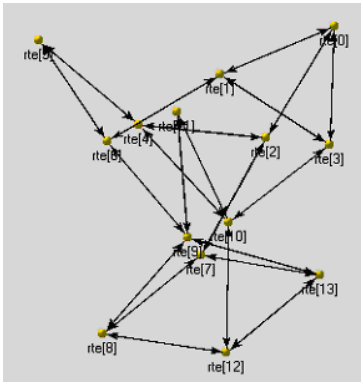

Fig. 9. NSFNet in OMNeT++

One of the major problems faced by anomaly based IDSs (including AIS based IDSs) is their inability to distinguish malicious traffic from normal traffic especially when a malicious node tries to deceive the system by launching specialized traffic patterns that differ only slightly from normal traffic patterns. This phenomenon is also called *slight deception*. In order to generate such attack traffic patterns, we developed a generic mathematical model of malicious traffic.

Before we move to details of mathematical model, let us revisit the TCP-SYN representation format by Hofmeyr [2]. It is a 49-bit representation, in which a 32-bit field refers to the TCP client's IP. From TCP server's point of view we can represent traffic as bands of different IP addresses. So, we may plot traffic against Most Significant Byte (MSB) of a TCP client's IP. Let us consider a simple case of single band traffic (for example, 132 in Figure 10). Then malicious traffic for this band can be modeled using normal distribution with $\mu = 132$. The extent up to which modeled traffic matches actual traffic depends on σ. Normal distribution, also known as Gaussian distribution is given by the equation below:

$$f(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right)$$

σ represents the standard deviation and μ represents the mean of the curve.

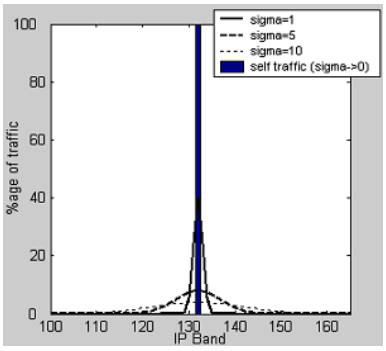

Fig. 10. Generated malicious TCP-SYN traffic for single band self-traffic

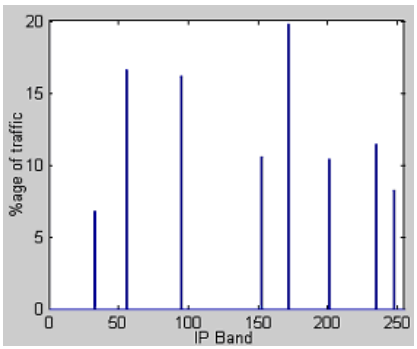

Fig. 11. Multiple band TCP-SYN self-traffic

But in real network scenarios, traffic received by a TCP server consists of multiple IP bands (see Figure 11 for self traffic plot). In Figure 11, the MSB of client's IP (x-axis) varies between 0 and 255. In order to develop a mathematical model for malicious traffic consisting of multiple bands, we have used multiple normal distributions. The sum of these weighted normal distributions is denoted by an overall modeling function (M), represented mathematically as:

$$M(x,\sigma,\mu_1,...,\mu_n,\omega_1,...,\omega_n) = f_1^{'}(x,\sigma,\mu_1,\omega_1) + ... + f_n^{'}(x,\sigma,\mu_n,\omega_n)$$

Here ω_n represents the weight of the n^{th} band, which corresponds to the percentage contribution by the band to total malicious traffic. μ_n represents the central position of the n^{th} band. Malicious traffic is modeled simply by using different values of σ . Note that $f^{'}$ represents weighted normal distribution,

$$f_k^{'} = \omega_k f_k,\ where\ 0 \leq \omega_k \leq 1\ and \sum_{k=1}^{n} \omega_k = 1$$

We have plotted the malicious traffic for multiple-bands by varying different values of σ in Figure 12.

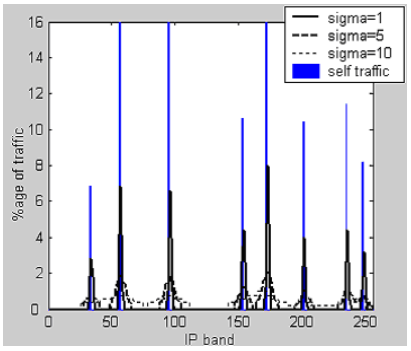

Fig. 12. Generated malicious TCP-SYN traffic for multiple band self-traffic

TABLE II
TRAFFIC SETS USED IN SIMULATIONS

Traffic Set Type	σ	No. of Strings	Unique Strings
Self-Set	-	7,000	4,200
Attack Set # 1 (TS1)	1	215,000	1,634
Attack Set # 2 (TS2)	10	215,000	4,988
Attack Set# 3 (TS3)	20	215,000	6,063
Attack Set# 4 (TS4)	30	215,000	6,665
Attack Set# 5 (TS5)	infinity	215,000	19,823

We generated a number of malicious traffic sets (TS) by using different values of standard deviation. The methodology used to generate self traffic is crucial since AIS requires a learning phase in which it tunes its detectors on the basis of self-traffic. In our case self-traffic consists of a batch of 7000 packets from all routers. It is assumed that self-traffic used for training does not contain any malicious packets. Five malicious (attack) sets were used to evaluate the performance of the system. The details of different self-traffic and malicious traffic sets are given in Table II.

In order to demonstrate the deceptiveness of these malicious traffic sets we carried out ROC (Receiver Operating Characterizes) analysis of AIS (without extended Thymus Action). The ROC curves for all the attack sets are plotted in Figure 13. Every curve is obtained by varying the hamming threshold. It is evident from the figure that AIS works as a perfect classifier for TS5 *($\sigma \rightarrow$ inf., random spoofing)* at point (0,1). From TS4→TS1 (as σ decreases) the performance of AIS degrades gradually. For TS1 the performance of AIS is even worse than the random guess line (y = x). This is because this attack set generates deceptive traffic that appears to be quite similar to normal traffic. This results in higher false positive rate for a given true positive rate.

Figure 14 presents the performance comparison of the system without/with our *extended thymus action*. Extended thymus action (see section IV) was utilized to evolve detectors on the basis of self traffic for multiple generations. Each attack lasted for 1000 seconds and we repeated each experiment 10 times. The reported results are an average of the values obtained from 10 independent runs. The performance parameters analyzed for extended thymus action include false positives and false negatives. This is because if false positive rate is reduced then it is expected that false negative rate may increase

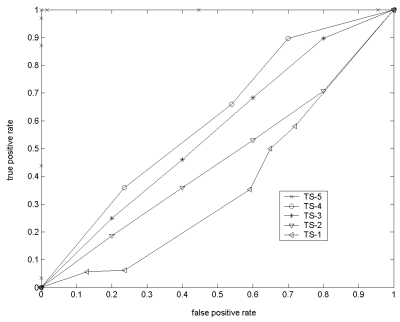

Fig. 13. ROC Curves of AIS (without Extended Thymus Action) for different Attack Sets

due to loss of sensitivity (decrease in true positive rate). This outcome is ofcourse undesirable. The above mentioned performance parameters are defined for TCP-SYN flood based DDoS attacks as:

$$FalsePositiveRate = \frac{\#\,of\ normal\ packets\ detected}{\#\,of\ packets\ detected}$$

$$FalseNegativeRate = \frac{\#\,of\ malicious\ packets\ not\ detected}{\#\,of\ packets\ not\ detected}$$

It is evident from figures 14 and 15 that our proposed model has reduced the false positive rate for all attack sets (*see Table III for exact values*). As expected, false negative rate has not increased. It is due to increased tolerization of detectors to normal traffic without loss of sensitivity. This clearly demonstrates the merits of the new approach. The experiments were done using hamming and contiguous distance matching techniques. Our proposed model showed its effectiveness for r-contiguous as well as hamming distance matching technique. Figure 16 shows the ROC curve for AIS with extended thymus action. Comparison with Figure 13 clearly demonstrates the improvement in the response of AIS, against malicious traffic, due to extended thymus action model.

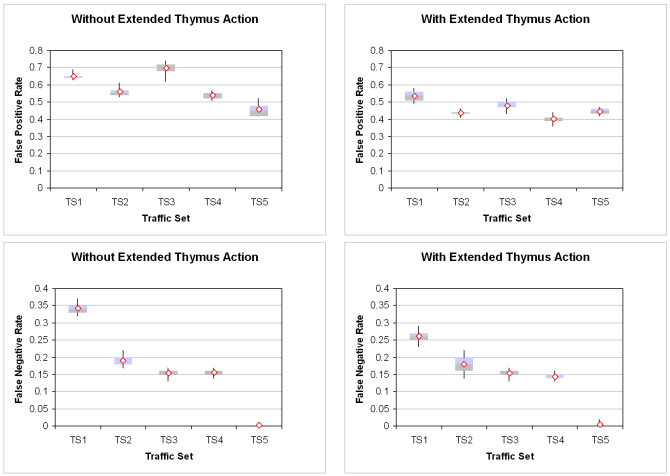

Fig. 14. Performance Comparison without/with Extended Thymus Action for r-Contiguous distance

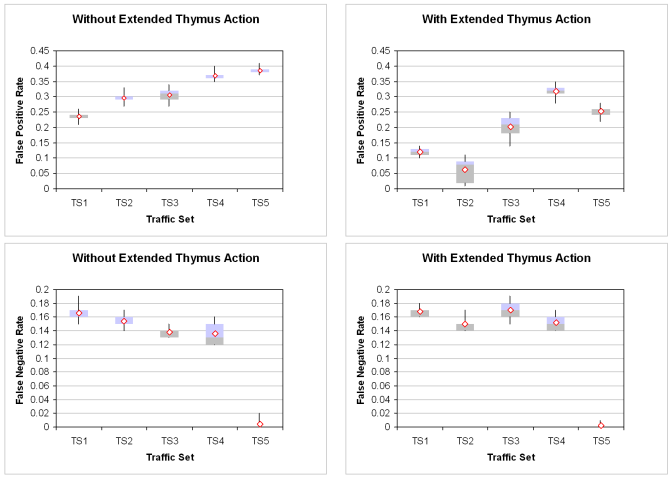

Fig. 15. Performance Comparison without/with Extended Thymus Action for Hamming distance

TABLE III
PERCENTAGE REDUCTION IN FALSE POSITIVES DUE TO EXTENDED THYMUS ACTION

Matching Method	TS1	TS2	TS3	TS4	TS5	Avg.
Contiguous	17.5%	22.1%	31.3%	25%	9%	21%
Hamming	49.1%	79.7%	34%	0%	34.7%	39.5%

We also tested our system under different attack scenarios by generating new traffic sets in which severity of the attacks varied. Degree of severity was enhanced by reducing the time interval between two subsequent TCP connection requests. In TS1 TCP connection requests were launched, on the average, every 220 msec, while till TS6 this interval was gradually reduced to 75 msec. Note that a self node, if get hacked, may flood the victim node with TCP-SYN packets without spoofing its IP. Such an attack scenario is catered by setting a threshold on inter-arrival time of TCP-SYN connections from a particular IP. This threshold was set to 50 msec for our system. Figure 17 shows that AIS minimizes the percentage of illegal connections when attack severity is increased incrementally from TS1 to TS6. But in comparison a network without AIS completely failed in inhibiting the establishment

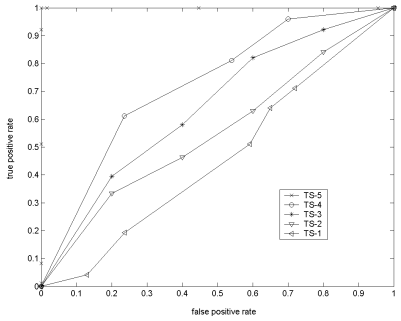

Fig. 16. ROC Curves of AIS with Extended Thymus Action

of illegal TCP connections. One can easily see in Figure 18 that a network with our AIS framework module restores the percentage of legal connections up to the same level when it was not experiencing any malicious traffic. The percentage of legal connections under normal network traffic is less than 100% in TS2 to TS6 because the routers dropped TCP-SYN packets in their overflowed buffers due to congestion.

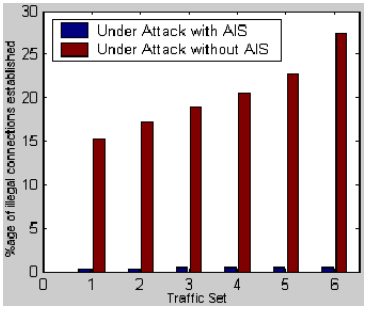

Fig. 17. AIS minimizes the percentage of illegal connections

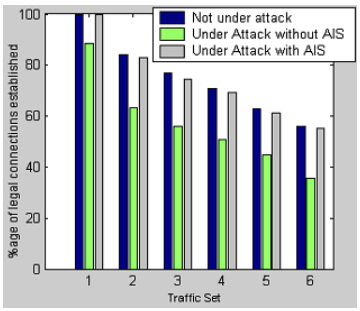

Fig. 18. AIS restores the percentage of legal connections

VI. Conclusion & Future Work

In the previous sections, we have presented the review of architecture of AIS for DDoS attacks. We have presented the mathematical model of deceptive malicious traffic which is used to generate attack traffic sets of TCP-SYN floods. We have provided the ROC analysis of the response of our AIS for these traffic sets. More sophisticated SYN-flood based attacks appear similar to the legitimate traffic. Therefore, it becomes difficult for a victim-end filter to distinguish malicious traffic from normal traffic. We present our proposed extended thymus action model. The performance comparison of extended thymus action model with simple thymus action model is also presented, which clearly shows significant reduction in the number of false positives (averaging up to 40% for hamming distance matching and 21% for r-contiguous distance matching) without affecting other performance parameters.

The proposed model was tested for r-contiguous and hamming distance matching techniques. We also investigated our AIS model under different degrees of attack severity. AIS successfully reduced the percentage of illegal connections established and restored the percentage of legal connections to the same level when the system was not under attack. It would be interesting to investigate the performance of other matching techniques such as r-chunk matching and variations of hamming distance matching (e.g. Rogers and Tanimoto, R&T matching) against deceptive traffic sets. In future, we also want to compare the cost-to-performance ratio of our model with signature based IDS.

References

[1] Martin Thorsen Ranang, *An Artificial Immune System Approach to Preserving Security in Computer Networks*, Master Thesis, Faculty of Information Technology, Mathematics and Electrical Engineering (IME) at the Norwegian University of Science and Technology (NTNU), June 2002.

[2] Steven A. Hofmeyr and S. Forrest, *Architecture for an Artificial Immune System*, Evoloutionary Computation Journal, pp. 443-473, 2000.

[3] L.N. de Castro, M. Ayara, J. Timmis, R. de Lemos, Ross Duncan, *Negative Selection: How to Generate Detectors*, Computing Laboratory, University of Kent at Canterbury, U.K. and Research Group Self Service Strategic Solutions, U.K.

[4] Leandro N. de Castro, *An Introduction to Artificial Immune Systems*, ICANNGA, April 2001, Prague.

[5] CERT Advisory CA-1998-01: *Smurf IP Denial-of-Service Attacks*, URL http://www.cert.org/advisories/CA-1998-01.html

[6] CERT Advisory CA-1996-01: *UDP Port Denial-of-Service Attacks*, URL http://www.cert.org/advisories/CA-1996-01.html

[7] CERT Advisory: *TCP SYN Flooding & IP Spoofing Attacks*, URL http://www.cert.org/advisories/CA-1996-21.html

[8] *D-o-S Attack via ping*, URL http://www.cert.org/advisories/CA-1996-26.html, 1997.

[9] *Intrusion detection systems: Protocol Anomaly Detection*, White Paper - Symantec Enterprise Security.

[10] Debar H., Dacier M. and Wepsi A., *Towards a taxonomy of intrusion detection systems*, Comp. Networks, pp. 361-378.

[11] Stefan Axelsson, *Intrusion Detection Systems: A Survey and Taxonomy*, Department of Computer Engineering, Sweden, 2000.

[12] P. Marrack and J. Kappler. *How the Immune System Recognizes Invaders*. In Scientific American, vol. 263, number 3, pp. 48-55, September 1993.

[13] H. S. Vaccaro and G. E. Liepins, *Detection of anomalous computer session activity*, IEEE Symposium on Security and Privacy, pp. 280-289, 1989.

[14] D. Dasgupta and F. Gonzalez, *An Immunity-Based Technique to Characterize Intrusions in Computer Networks*. In the Journal IEEE Transactions on Evolutionary Computation, Volume:6, Issue:3, Page(s):281-291, June, 2002.

[15] Kim J. & Bently P. J., *Investigating the Roles of Negative Selection in an AIS for NID*, IEEE Transactions of Evolutionary Computing, Special Issue on AIS, 2001.

[16] Kim J. & Bently P. J., *Evaluating Negative Selection in an Artificial Immune System for Network Intrusion Detection*, Proc. Of Genetic and Evolutionary Computation Conference, pp. 1330-1337.

[17] Kim J. & Bently P. J., *Towards an Artificial Immune System for Network Intrusion Detection: An investigation of Clonal Selection with a Negative Selection Operator*, Proc. of the Congress on Evolutionary Computation, pp. 1244-1252, 2001.

[18] S. Forrest, S. Hofmeyr, and A. Somayaji, *Computer immunology*, Communications of the ACM, pp. 88-96, 1997.

[19] S. Forrest, S. A. Hofmeyr, and A. Somayaji, *A sense of self for unix processes*, In Proceedings of the 1996 IEEE Symposium on Research in Security and Privacy, Los Alamitos, CA, 1996. IEEE Computer Society Press.

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

[21] Leandro N. de Castro and Jonathan Timmis, *Artificial Immune Systems: A New Computational Intelligence Approach*, Springer, 2002.

[22] Tao Peng, *Defending against Distributed Denial of Services Attacks*, Ph. D. dissertation, Department of Electrical and Electronics Engineering, University of Melbourne, April, 2004.