ARTIFICIAL IMMUNITY USING CONSTRAINT-BASED DETECTORS

Haiyu Hou Jun Zhu Gerry Dozier
Department of Computer Science and Software Engineering
Auburn University
Auburn, AL 36849

ABSTRACT

In this paper, an artificial Immune system (AIS) is used to detect abnormality in a computer network in an effort to provide protection from illegal intrusion and unauthorized use. The problem of anomaly detection can be addressed as the problem of distinguishing self patterns from non-self patterns. The self patterns could be any variety of characteristics of a system or its users. Detectors of an AIS recognize non-self patterns and invoke an alarm. In this work, an AIS was used to monitor simulated TCP/IP traffic on a simulated broadcast local area network. Detectors, in the form of interval constraints, were used to develop a simple and effective AIS.

KEYWORDS: AIS, intrusion detection, self pattern, non-self pattern, constraint-based detector

INTRODUCTION

The immune system is critical in protecting the human body against the natural environment that contains a wide variety of pathogens in the form of viruses and bacteria [5]. Upon entering the body, these pathogens elicit immune system responses, during which antibody-mediated immunity kills and clears them from the body [5].

An artificial immune system (AIS) [4], inspired by the vertebrate immune system, is an automated system that can be used as the basis of a network security system. An AIS can monitor the traffic on a network and detect suspicious behavior [4]. An AIS can accomplish this by classifying normal traffic as self and abnormal traffic as non-self [1, 4].

Many researchers have investigated the effectiveness of AISs for computer security [1, 2, 4] and network security [3]. However these approaches have used binary coded detectors. In this paper, we propose an alternative network-based AIS that uses detectors that are coded as interval constraints. The remainder of this paper is as follows. In the next section, we briefly explain the rationale on our AIS, next discuss our implementation followed a discussion of our problem set. Next we present our results followed by our conclusions.

RATIONALE

The problem of anomaly detection can be addressed as distinguishing self patterns from non-self patterns[1]. The patterns could be a variety of characteristics of a system, such as the sequence of system calls [1, 3], the specification of a TCP/IP packet [4].

Our AIS monitors the TCP/IP traffic on a local area network (LAN). Each incoming or outgoing packet is broadcast to all the hosts on the LAN. For each packet, its source IP, destinations IP and port number on the internal host compose a data path triple (DP3) [4], which, as a pattern of the packet, is inspected by AIS; abnormal patterns are recognized as non-self. Non-self patterns will correspond to either a network intrusion by an external hacker, or misuse by an internal user.

An AIS contains immature, mature and memory detectors [4]. Mature and memory detectors are responsible detecting anomalous activity. The immature detectors are randomly generated and must go through an immunological process called negative selection [4], during which illegitimate detectors which bind to self DP3s are killed. Immature detectors that have survived the process of negative selection will bind to non-self DP3s and become mature detectors. A mature detector has a lifespan during which it must catch at least one non-self DP3; otherwise it dies. After a mature detector matches a non-self DP3, it becomes a memory detector that has a longer lifespan than mature detectors. If a memory or mature detector matches a DP3 transmitted on the network, an alarm message is sent the network administrator. This is similar to what is proposed in [4].

IMPLEMENTATION

Our implementation is as modified from [4]. For a TCP/IP packet, at least one internal host is involved in the connection. An IP address is composed of 4 bytes. For a class C LAN, the least significant byte is enough to identify an internal host, while the 4 bytes are all needed to specify the external hosts. The port on a host is identified by 2 bytes, which give 64k different numbers. To simplify the process, the port numbers is mapped into 69 categories according to [4]. Thus, a DP3 has 6 integers to specify a packet, with 4 integers for an external host IP address, 1 integer for the internal host IP address, and 1 integer for the port on the internal host. The first 5 integers range between 0 and 255, and the last integer ranges from 0 to 68. Another bit is included to indicate packet direction. If it's an incoming packet, this bit is set to 0, and it's set to 1 if the packet is outgoing.

Accordingly, our constraint-based detectors are composed of 6 pairs of integer intervals and 1 direction bit. If an interval covers the corresponding number, the detector matches this field. If the detector matches the direction bit and all the six fields of a DP3, the detector matches the DP3 completely. A match threshold is set to adjust the specificity or generality for detectors to match a DP3. The match is most specific if the threshold is set to 6, and most general if the threshold is set to 1.

The self set is composed of DP3s that represent normal activity on the network. It is reasonable to assume that the DP3s of the self set can be divided into separate sub self sets because the connections are between LANs and all the external hosts on a single

LAN have the same network address. For example, if 5 LANs have hosts connecting to a sixth LAN, then the self set of this sixth LAN may be classified into 5 clusters (or sub-self sets).

EXPERIMENT SET-UP

A simulation of 10 hosts each running an AIS was studied. Each of the AISs contained 100 constraint-based detectors. The self set had a size of 2000 DP3s, which were randomly generated and evenly spread into a number of clusters. There were several variables in setting up the experiment, including match threshold, number of self-clusters, and the lifespan of immature detectors. The lifespan of mature detectors was set to 100,000, which means it must match a non-self DP3 within 100,000 DP3 transmission on the network, or it dies and is replaced by a randomly generated immature detector. The lifespan of a memory detector was set to 200,000.

The match threshold took on values between 1 and 6. The number of self-clusters took on values from the set {1, 2, 4, 5, 8, 10, 20, 25, 40, 50, 80, 100}. The lifespan of for immature detectors was selected from the set {10000, 15000, 20000}. For each possible set of parameters the resulting network security system was trained and tested 30 runs. A non-self set, consisting of 200 DP3s, was created by randomly generated. The self and non-self sets for each of the 30 runs were mutually exclusive.

The experiment was divided into two stages. In the first stage, the training was carried out by randomly picking self-patterns from the self set and feeding them to the AISs. The training stage lasted for 150,000 iterations which equivalent to transmitting 150,000 DP3s across the simulated LAN. False positives were counted for the last 29,000 iterations in the first stage. In the testing stage, non-self patterns, or simulated attacks, were fed to the system to test the efficiency of detection. The number non-self DP3s identified by the system was summed up, and the summation divided by 200 in order to compute the detection rate. This was repeated nine additional times with nine other non-self sets being fed into the system, which resulted in 10 detection rates. The overall detection rate was calculated as the average of these 10 detection rates.

RESULTS

Average detection rates over various system settings are shown in Figures 1 - 3. From these figures, it can be seen that the detection rate went down when the number of clusters increased. However, for match thresholds of 2 and 3, the detection rates remained close to 1.0 until the cluster number increased beyond 40. The detection rate dropped because the increasing in self-clusters means more diversity in the self patterns, which requires more learning time and more memory resources as well. It’s reasonable to predicate that longer the training time and the larger size of the detector set the better the detection rate when working with a larger number of self-clusters.

It can also be observed that for the best systems (those with a match threshold of 3 or 4 in Figures 1-3) a longer lifespan for immature detectors causes a decreased the detection rate. The reason is that the longer the immature detector lifespan, the greater chance for the immature detector to match a self-pattern during negative selection. This lessens the chance of an immature detector surviving negative selection and developing into a mature detector. This, in turn, reduces the total number of mature detectors in the system. Although the resulting system may have fewer mature detectors, with a longer immature detector lifespan, the mature detectors are less likely to bind to self patterns which reduces the number of false positives generated. The issue of false positives will be discussed later.

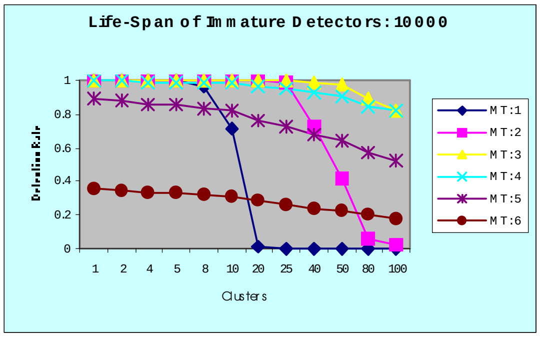

Figure 1. Detection rate with lifespan of immature detector set to 10000

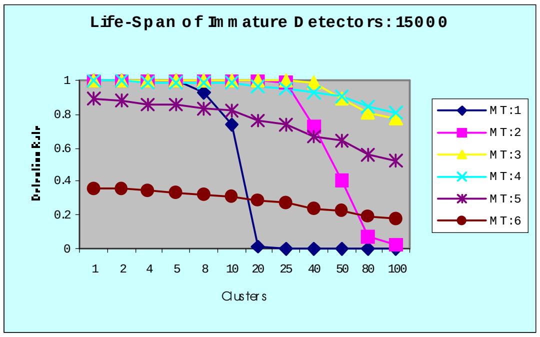

Figure 2. Detection rate with lifespan of immature detector set to 15000

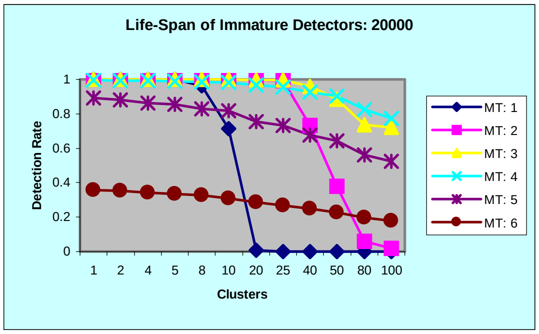

Figure 3. Detection rate with lifespan of immature detector set to 20000

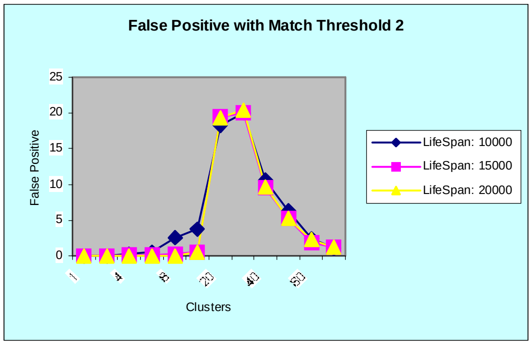

Figure 4. False positives when the matching threshold was 2

In Figures 1-3, one can see that the match threshold is more important than the lifespan of immature detectors. It can be seen in figures that match thresholds of 3 and 4 were consistently the better performers. Lower threshold for a detector to match to a pattern means a relaxed constrain satisfaction, thus the immature detector got greater possibility to match to some pattern. This leads to less amount of mature detector with the same logic related above. On the other hand, high threshold leads to more mature detectors, however, these detectors have more trouble to match a pattern, even if it's an obvious non-self pattern. So there's a balance between the two, at which highest detection rate takes place.

The number of self clusters also affects the setting of match threshold. If the number of self-clusters was below 25, 2 might be an ideal choice because the detection rate was kept very close to 1.0; however, it would be more advisable to choose 3 if the self patterns fall into more clusters because the detection rate with 3 as match threshold was kept over 0.95 even the cluster number reached 50 if the lifespan of immature detector was set to 10000.

The relationship between false positives and match threshold are shown in Figures 4 - 8. Match threshold of 1 resulted in 0 false positives regardless of the number of self clusters. It can be seen that increasing clusters increased the diversity of the self DP3s, which resulted in more false positives. This basically matches with the results as shown in Figures 1 - 3, except that when the match threshold was set to 2 there was a peak in the number of false positives for unknown reasons. From Figure 1 it can be seen that for match threshold of 2, the detection rate was kept very close to 1.0 until the number of clusters increased to 25, when it began to drop, and it was at this point that highest false positive rate occurred. When the cluster number exceeded 25, the detection rate dropped and so did the number of false positives. This was because with more diversity in the self DP3s, the system had fewer mature detectors and more difficulty matching a DP3, no matter it would be a false positive in the training phase, or it was a true intrusion in the testing phase.

It was also observed that different settings on match threshold had a significant effect on the number of false positives generated, except for match threshold of 1. When the match threshold was set to 3 or 4, the system had the highest false positive rate, at which point the system also gave the highest detection rate, and the false positive dropped when the match threshold increased to 5 and 6. If the system is capable of recognizing more DP3s, it also potentially generates more false positives. So, there's a trade-off between the detection rate and the accuracy of the detection. If the security of the network system were more concerned, higher detection rate would be more desired over the accuracy while false alarms are not much trouble, then a shorter immature detector lifespan and a match threshold of 3 or 4 would be suitable.

The size of the self set was set to 2000 for the results reported here, however, we also tested the system with self set of varied size, and it turned out the self set size did not noticeably affect the system effectiveness.

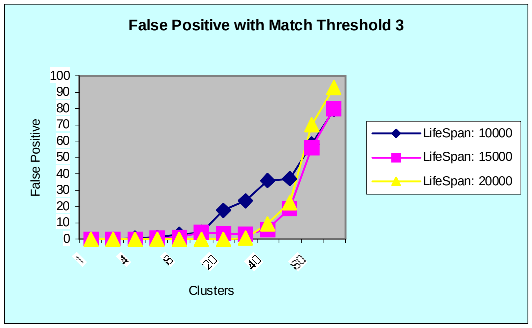

Figure 5. False positives when the matching threshold was 3

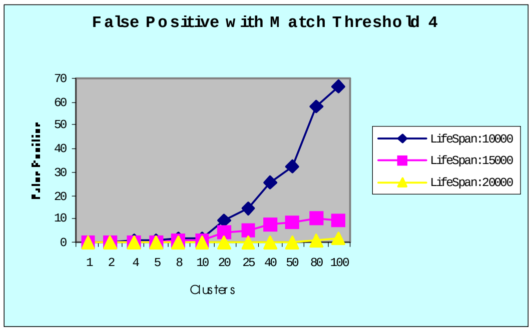

Figure 6. False positives when the matching threshoold was 4

CONCLUSION

We proposed an alternative approach to build an simple and efficient AIS. Constraint-based detector is easier to understand compared with binary string based detector [3], which facilitates the system security officer to analyze the network traffic when intrusion happens.

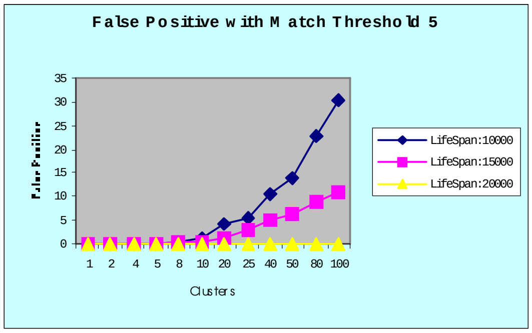

Figure 7. False positives when the matching threshold was 3

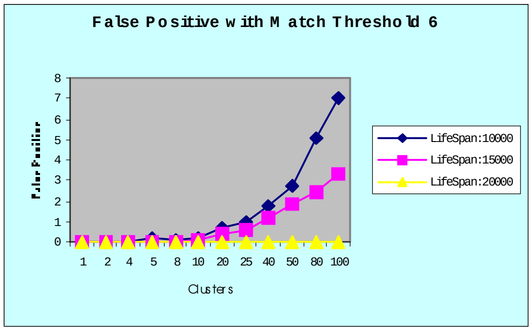

Figure 8. False positives when the matching threshold was 4

The diversity of the self patterns decides the difficulty of the detection problem, which significantly influences the effectiveness of the AIS. The match threshold decides the specificity or generality of the match between detectors and patterns. A high match threshold results in more mature detectors that match patterns more specifically, while a low match threshold results in fewer mature detectors that match patterns more generally. The detection rate might be low if the match threshold is either too high or too low. The lifespan of an immature detector may also affect the number of mature detectors generated during the training phase. The longer the lifespan of immature detectors the fewer number of mature detectors there may be. This can lead to a lower detection rate.

Effectiveness and accuracy are two important requirements for a good intrusion detection system. If the diversity of the self patterns were low with respect to the size of the AIS, the system would be both effective and accurate, giving a high anomaly detection rate and low false positive rate. However, if the self patterns showed high diversity with respect to the size of the AIS, there would be a trade-off between effectiveness and accuracy, one of which might be sacrificed for the other.

REFERENCES

1. S. Forrest, S. Hofmeyr, A. Somayaji and T. Longstaff, “A Sense of Self for Unix Processes”. *Proceedings of IEEE Symposium on Security and Privacy*, 120 –128 (1996).
2. P. Harmer and G. Lamont, “An Agent Based Architecture for a Computer Virus Immune System”. *Proceedings of the Genetic and Evolutionary computation Conference*, (2000).
3. S. Hofmeyr, A. Somayaji and S. Forrest, “Intrusion Detection using Sequences of System Calls”. *Journal of Computer Security* 6: pp. 151-180 (1998).
4. S. Hofmeyr and S. Forrest, “Immunity by Design: An Artificial Immune System”. *Proceedings of the Genetic and Evolutionary Computation Conference*, (1999).
5. D. Krogh, “The Immune System: Defending the Body from Invaders”. *Biology*, ISBN: 0023668911 (2000).