A DETECTOR GENERATING ALGORITHM FOR INTRUSION DETECTION INSPIRED BY ARTIFICIAL IMMUNE SYSTEM

Walid Mohamed Alsharafi and Mohd Nizam Omar
Inter Networks Research Laboratory, School of Computing, College of Arts and Sciences, Universiti Utara Malaysia, UUM Sintok, Malaysia
E-Mail: sharafi12@yahoo.com

ABSTRACT

Artificial immune system (AIS) allows us to inspire several ideas for the design of computer intrusion detection. The standard of negative selection algorithm (NSA), offered by Stephanie Forrest in 1994, is one of the most common mechanisms in AIS that applied in anomaly detection for the similarity of its basic idea. One of the most operational improvements in the standard of NSA is how to generate effective detectors which play a significant role in self and nonself discrimination in intrusion detection system (IDS). In this paper, we offer an improvement to a detector generating algorithm to generate effective detectors which leads to improve the standard of NSA, which in turn leads to improve the NSA based anomaly intrusion detection. Experimental results show that the improved algorithm able to generate more effective detectors and keeping the space and time complexities better than in the standard of NSA. This leads to detecting the real-time intrusion with less false negative.

Keywords: detector, detector generating algorithm, negative selection, intrusion detection.

INTRODUCTION

Intrusion detection is a significant component of information processing system protection. It provides an additional layer of defense against computer abuse after physical, authentication and access control (Dasgupta, 1999). The anomaly intrusion detection identifies the intrusion by the comparison between abnormal (i.e. anomaly) activities and the normal one of the protected system. The normal activities are maintained in the established normal profile which holds the model of the effective normal activity and detection methods. In order to resolve the problem in anomaly intrusion detection, there are some ways of Artificial Intelligence (AI) such as data mining, neural network, and artificial immune system (AIS). The AIS is a sub-field of computing inspired by the biological natural immune system. AIS gets us to inspire several ideas for solving intrusion problems by emulating the biological natural immune mechanism to discriminate "self" and "nonself": where "self" could be specified as many things, such as normal behavior, normal network traffic between computers and so on. One of the basic and most common approaches in AIS is the Negative Selection Approach (NSA), which is simple and easy to implement. It was employed in many studies, but mostly in anomaly detection for the similarity of its basic idea (Aziz, Azar, Hassanien and Hanafy, 2014). The standard of NSA has been offered in 1994. Since that time, a number of works have been proposed to improve the NSA standard so that it can be applied in designing real-time IDS. The algorithm proposed by Xian (Xian, 2009) has drawn our attention because of its simple idea of generating the effective detectors with less space and time complexities, comparing with the standard NSA, which run to design real-time NSA based anomaly IDS but, however, it costs a lot of false negative because the number of the generated effective detectors is quite small. Therefore, in this paper, we improve the work in (Xian, 2009) to generate more effective detectors for NSA based anomaly IDS with a low percentage of false negative as well as with less space and time complexities.

NEGATIVE SELECTION MECHANISM

Negative selection principle in biological immune system

The human immune system (IS) is a truly amazing constellation of responses to attacks from outside the body. It has many facets, a number of which can switch to optimize the response to these unwanted intrusions (Haldar and Ahmad, 2010). The system is unusually effective, most of the time. The interaction among the IS and several other systems and organs allows the regulation of the body, ensuring its stable operation. The effect of this mechanism is that the IS is capable of distinguishing between organism's self cells and nonself cells. This procedure identifies the principle of negative selection of the organism's cells. Negative selection allows only the existence of those cells that do not recognize self cells. The cells are produced and undergo a maturation process known as immune tolerance in the thymus gland and bone marrow respectively, after that they are permitted to convey constituent in an immune response. In immune tolerance, cells will die if they have matched with self cells, and if not, they will become mature cells, and function as the true immune competent cells.

Negative selection algorithm in artificial immune system

The process of detection anomaly intrusion in a computer system can be considered as the process of distinguishing "self" and "nonself" in the immune system.

In the light of this thought, Stephanie Forrest lead a research group in New Mexico University and proposed the immune negative selection algorithm in 1994 (Forrest, Perelson, Allen and Cherukuri, 1994). In essence of this algorithm is to generate a set of detectors which do not recognize "self" but distinguish the "nonself" (viruses, worms, Trojan horses, spyware, unauthorized access, etc.) from "self" (protected data files, authorized users, etc.) (Forrest, Perelson, Allen and Cherukuri, 1994). This algorithm comprises two phases (Forrest, Hofmeyr, Somayaji and Longstaff, 1996): censoring and monitoring. The censoring stage serves for the generation of mature detectors which monitor the system being protected for changes. The algorithm builds a lot of competent detectors in the following steps (see Figure-1):

Step 1: define a set of self (S). The data being protected is viewed as a string. The string is split into several l-length substrings. The set of self S consists of several substrings.

Step 2: generate a set of random candidate detectors (Ro). They are also l-length strings and generate in some probability analytical ways.

Step 3: generate a set of competent detectors (R). Strings from Ro that match self are eliminated. Strings that do not match any of the strings in S become members of the detector collection R. This step is called censoring.

Step 4: monitor the changes of self. This is accomplished by continually choosing one detector in R and testing to find out if it matches with strings in S. If the self string matches one of the detector strings, a change would happen in S. Those changes are done probably for intrusion, virus or misusing. This step is called monitoring.

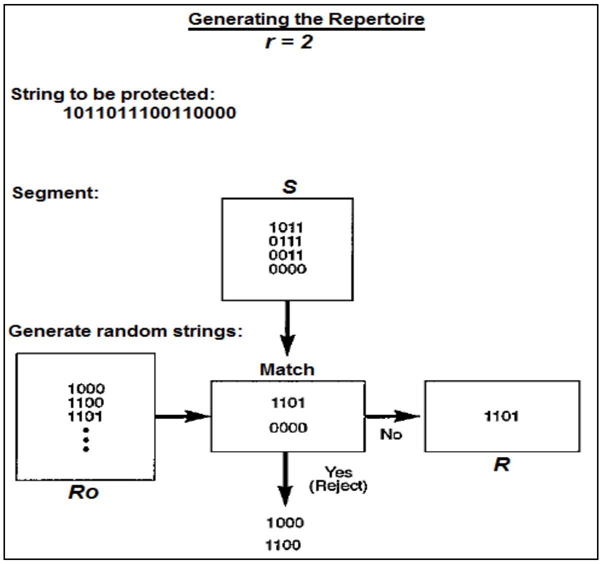


Figure-1. The generation of the repertoire (i.e. the competent detectors) by using the alphabet (0, 1) with r = 2 (Forrest, Perelson, Allen and Cherukuri, 1994).

ARTIFICIAL IMMUNE NEGATIVE SELECTION ALGORITHM FOR THE ANOMALY INTRUSION DETECTION

Here, we propose an improvement to the algorithm of detector genrating offered in (Xian, 2009) which can generate effective detectors in NSA based anomaly intrusion detection. This algorithm is better than the detector generating algorithm which offered by Forrest in 1996 on the speed and efficiency of generating detectors (Xin, 2009). The terms related to this algorithm and to our proposed improvement are defined in Table-1.

Table-1. Definition of terms used in the algorithm.

Terms	Definitions
l	Length of string (including self strings, nonself strings and detectors)
r	Matching threshold
m	Alphabet size (supposed m=2 (i.e. using the alphabet (0,1))
S	Self string, length=r
N	Size of string (or detector) space = 2^l
N_S	Size of self data
N_{NS}	Size of nonself data ($N_{NS} = N - N_S$)
N_R	Size of competent detectors
N_H	Size of holes (see Compute the "Holes" section)
N_D	Size of detected nonself ($N_D = N_{NS} - N_H$)
P_f	Probability of failing to detect nonself (P_f = 1- (Detected / N_{NS}))

The Rule of r-contiguous bits matching

The r-contiguous bits matching rule states that the two character strings are matching if both strings have at least r bit continuous identical beginning from a particular position. (Ayara, Timmis, de Lemos, de Castro and Duncan, 2002). In Figure-2: if m =2, l =6, r =3, we can consider S1 =110100 and S2 =100101 are matching.

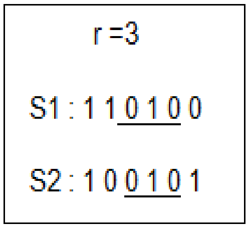


Figure-2. R-contiguous bits matching rule.

Detectors generating algorithm

` We use the idea of the algorithm proposed in (Xin, 2009) to get a more efficient algorithm of detector generating.

If m, l and N_S are given, then the competent detectors (N_R) can be generated. First, we give the following definition:

C: a character array of the length l, it is used to keep every new generated detector tentatively.

Si: a substring of C, starting from the position i and its length is r ($1 \leq i \leq l - r + 1$).

The improved detector generating algorithm is presented as a flow diagram in Figure-3.

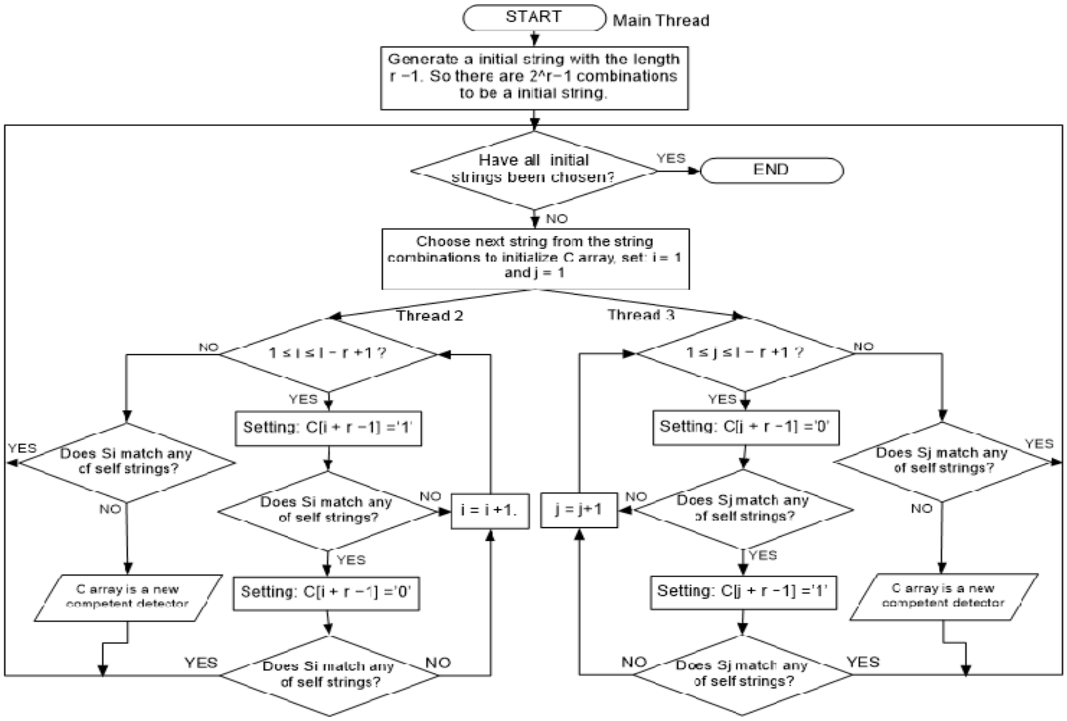
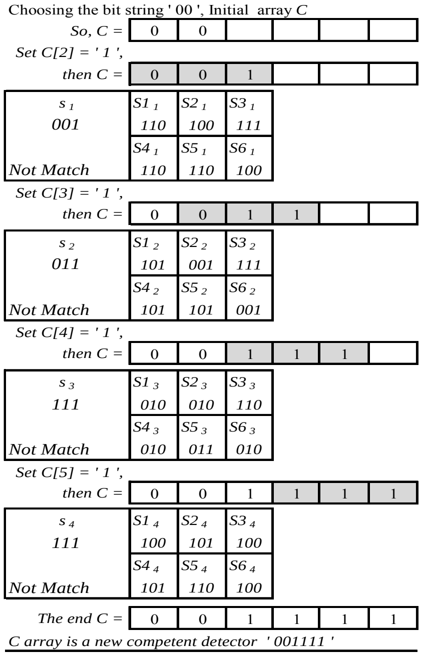

START
Main Thread
Generate a initial string with the length r −1. So there are 2^r−1 combinations to be a initial string.
Have all initial strings been chosen?
YES
END
NO
Choose next string from the string combinations to initialize C array, set: i = 1 and j = 1
Thread 2
Thread 3
1 ≤ i ≤ l − r + 1 ?
1 ≤ j ≤ l − r + 1 ?
NO
YES
Setting: C[i + r −1] ='1'
Setting: C[j + r −1] ='0'
Does Si match any of self strings?
Does Sj match any of self strings?
i = i +1.
j = j+1
Setting: C[i + r −1] ='0'
Setting: C[j + r −1] ='1'
C array is a new competent detector
Does Si match any of self strings?
Does Sj match any of self strings?

Figure-3. The improved detector generating algorithm.

Figure-4 shows the example of generating process of a new detector '001111' using the Thread 2 of the improved detector generating algorithm shown in Figure-3.



Choosing the bit string ' 00 ', Initial array *C*

So, C =	0	0				

Set C[2] = ' 1 ',

then C =	0	0	1			

s_1 *001* *Not Match*	*$S1_1$* *110*	*$S2_1$* *100*	*$S3_1$* *111*
	$S4_1$ *110*	*$S5_1$* *110*	*$S6_1$* *100*

Set C[3] = ' 1 ',

then C =	0	0	1	1		

s_2 *011* *Not Match*	*$S1_2$* *101*	*$S2_2$* *001*	*$S3_2$* *111*
	$S4_2$ *101*	*$S5_2$* *101*	*$S6_2$* *001*

Set C[4] = ' 1 ',

then C =	0	0	1	1	1	

s_3 *111* *Not Match*	*$S1_3$* *010*	*$S2_3$* *010*	*$S3_3$* *110*
	$S4_3$ *010*	*$S5_3$* *011*	*$S6_3$* *010*

Set C[5] = ' 1 ',

then C =	0	0	1	1	1	1

s_4 *111* *Not Match*	*$S1_4$* *100*	*$S2_4$* *101*	*$S3_4$* *100*
	$S4_4$ *101*	*$S5_4$* *110*	*$S6_4$* *100*

The end C =	0	0	1	1	1	1

C array is a new competent detector ' 001111 '

Figure 4: generating process of the detector ' 001111 '

The Example in Figure-4: when we set m = 2, l = 6, r = 3 and self collection includes 6 strings (i.e. $N_S = 6$): S1= 110100, S2= 100101, S3=111100, S4=110101, S5=110110 and S6=100100, the algorithm generates 2^{r-1} = 4 possible bit strings: ' 00 ', ' 01 ', ' 10 ' and ' 11 ', and then use them to initialize successively character array C. For example, if we choose the bit string ' 00 ', then Figure-4 shows how to generate the detector ' 001111 ' using the Thread 2 of the flowchart in Figure-3. The Thread 3 will generate another detector which is '000000'. Figure-5 shows the string space map that results from both Thread 2 and Thread 3 of the flowchart with the same above-mentioned sets (i.e. r =3, l = 6, m = 2 and N_S = 6).

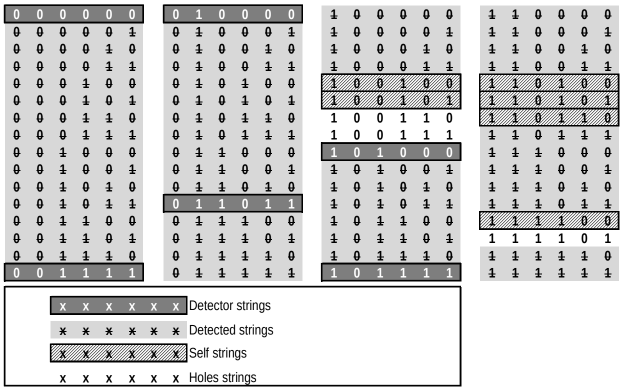


Figure 5: Example of string space map that shows results of applying the offered algorithm when set: *r* =3, *l* = 6, *m* = 2 and *Ns* = 6.

The time complexity of the algorithm is decided by the time to generate 2^{r-1} possible combinations and the time to extend detector by each combination. So the time complexity is $o(2^{r-1}(l - r + 1)N_S)$. The space complexity is determined by the size of the array C. So the space complexity is o(l).

The complexity of space and time of this algorithm is the same as in the algorithm proposed in (Xian, 2009). However the efficiency of our algorithm is better because its ability to generate more competent detectors (N_R) with less holes (N_H) which leads to decrease the probability of failing to detect nonself (P_f) significantly.

Compute the "Holes"

In accordance with the above-mentioned matching rule and self collection strings S, for some nonself strings, called “holes”, it is impossible to generate valid detectors, i.e. the hole string will never match any of the generated detectors. When the self collection has two strings like this: beginning with some assigned place and having the same numbers in more than r -1 adjacency bit, there are at least two holes which can’t be detected by any detector. Unfortunately the holes can not be avoidable. But too many holes mean that there are too many nonselfs which can not be detected, and there are also too much intrusion that can’t be detected (Xin, 2009).

RESULTS AND DISCUSSIONS

The Table-2 shows experimental results based on the detector generating algorithm.

Table-2. Experimental results

Self Collection Example	l	N	N_S	N_{NS}	r	N_R	N_H	N_D	P_f
						(a)			
						(b)			
S1=1101, S2=1001, S3=1111, S4=0101	4	16	4	12	2	1	4	8	0.333
						1	4	8	0.333
					3	3	3	9	0.250
						5	2	10	0.167
S1=1101, S2=1001, S3=1111, S4=0101 S5=1100, S6=1010			6	10	2	1	3	7	0.300
						1	3	7	0.300
					3	2	4	6	0.400
						3	1	9	0.100
S1=110100, S2=100101, S3=111100, S4=110101, S5=110110, S6=100100	6	64	6	58	3	3	18	40	0.310
						6	3	55	0.052
					4	8	26	32	0.448
						15	8	50	0.138
					5	13	29	29	0.500
						28	5	53	0.086
S1=110100, S2=100101, S3=111100, S4=110101, S5=110110, S6=100100, S7=110111, S8=100111			8	56	3	3	16	40	0.286
						4	8	48	0.143
					4	8	24	32	0.429
						14	12	44	0.214
					5	12	30	26	0.536
						24	7	49	0.125
S1=10010010 S2= 10010101, S3= 11010010 S4= 11010101, S5= 11011010 S6= 11110001, S7= 11011110 S8= 10011001	8	256	8	248	5	12	145	103	0.585
						25	50	198	0.202
					6	29	132	116	0.532
						58	24	224	0.097
					7	64	120	128	0.484
						128	8	240	0.032
S1=10010010 S2= 10010101, S3= 11010010 S4= 11010101, S5= 11011010 S6= 11110001, S7= 11011110 S8= 10011001, S9= 00001111 S10= 01010011			10	246	5	10	142	104	0.577
						22	46	200	0.187
					6	29	130	116	0.528
						58	28	218	0.114
					7	64	118	128	0.480
						128	10	236	0.041

In Table-2 each of the columns N_R, N_H, N_D and P_f depicts the values resulted from two different experiments (a) and (b), where the experiment (a) represents the algorithm proposed in (Xin, 2009), whereas the experiment (b) represents the algorithm with the improvement which we offer to get a better detector generating algorithm regarding the efficiency.

It is clear from the results in the Table-2 that the sizes of competent detectors (N_R) in experiment (b) almost double comparing with the corresponding sizes in the experiment (a). This lead to that the sizes of the detected nonself (N_D) in experiment (b) almost double comparing with the corresponding sizes in the experiment (a) and this lead in turn to that: 1) the sizes of holes (N_H) in experiment (b), is less comparing with the corresponding sizes in the experiment (a). 2) The probability of failing to detect non-self (P_f) in experiment (b) gradually decreases comparing with the corresponding values in the experiment (a). As it was mentioned in the previous section, too many holes mean that there is too much intrusion that can’t be detected which means too much false negative reported by the anomaly IDS. With this in mind and through the analysis of the results in the Table-2, it is clear that the anomaly IDS that would be developed using our improved algorithm will report less false negative than those reported by the anomaly IDS which would be developed using the algorithm proposed in (Xin, 2009). This is because, as we explained above, the size of holes resulted by our improved algorithm is quite less comparing with the corresponding size of holes resulted using the algorithm proposed in (Xin, 2009).

CONCLUSION AND FUTURE WORK

In this paper, we offered an improved and efficient algorithm for generating effective detectors needed to improve the performance of the standard NSA, which in turn can be used in computer intrusion detection design. We tested the algorithm to study: (1) its detector generating efficiency. (2) non-self detecting efficiency. The results demonstrate both, the efficiency of the detector generating and the efficiency of non-self detecting. Besides that, the improved algorithm keeps the space and time complexities better than the standard of NSA. This leads to detecting the real-time intrusion with less false negative. Therefore, our future work is to investigate the degree to which the proposed algorithm is able to integrate with the anomaly IDS architecture proposed in our previous work (Omar & Alsharafi, 2013) to enhance and improve the efficiency of the anomaly IDS model. We also plan to use the standard of NSA and the proposed detector generating algorithm to investigate whether is it possible to invent what we will call Abnormal Profile as a new concept in anomaly IDS beside the well known Normal profile concept. The expected abnormal profile will contains the effective detectors, generated by the NSA, which take on a significant role in self and nonself discrimination for anomaly IDS.

REFERENCES

Ayara M., Timmis J., de Lemos R., de Castro L. N. and Duncan, R. (2002). Negative selection: How to generate detectors. In Proceedings of the 1st International Conference on Artificial Immune Systems (ICARIS). 1: 89-98. Canterbury, UK: [sn].

Aziz A. S. A., Azar A. T., Hassanien A. E. and Hanafy S. E. O. 2014. Negative Selection Approach Application in Network IDSs. arXiv preprint arXiv: 1403.2716.

Dasgupta D. 1999. Immunity-based intrusion detection system: a general framework. In: Proc. of the 22nd NISSC. 1: 147-160.

Forrest S., Perelson A. S., Allen L. and Cherukuri R. 1994. Self-nonself discrimination in a computer. In 2012 IEEE Symposium on Security and Privacy. IEEE Computer Society. pp. 202-202.

Forrest S., Hofmeyr S. A., Somayaji A. and Longstaff T. A. 1996. A sense of self for unix processes. In Security and Privacy, 1996. Proceedings., IEEE Symposium on pp. 120-128.

Haldar C. and Ahmad R. 2010. Photoimmuno-modulation and melatonin. Journal of Photochemistry and Photobiology B: Biology. 98(2): 107-117.

Omar M. N. and Alsharafi W. M. 2013. A Normal Profile Updating Method for False Positives Reduction in Anomaly Detection Systems. In: The Second International Conference on Informatics Engineering and Information Science (ICIEIS2013). The Society of Digital Information and Wireless Communication. pp. 182-187.

Xin D. S. R. L. 2009. The anomaly intrusion detection based on immune negative selection algorithm. Granular Computing, 2009, GRC'09. IEEE International Conference on 2009.