Xiaojie LIU

Sichuan University, China

An immune based model for dynamic intrusion detection

***Abstract**. A new immune based model called AIBM for dynamic intrusion detection is proposed. AIBM uses a very small dynamic self set during the self tolerance for immature detectors, resulting in a higher efficiency in generating new mature detectors than traditional computer immune systems (CIS). Meanwhile, the self set can synchronize their variations with the real-network environment as time goes on, resulting in the dynamic evolution of self set, mature and memory detectors, offering more self-adaptation, and having a lower error rate than traditional CIS models.*

***Streszczenie.** Zaproponowano nowy model odporności do wykrywania dynamicznych włamań w systemach komputerowych. Układ oferuję auto-adaptację i ma mniejsze błędy niż tradycyjne układy CIS. (**Nowy model odporności do wykrywania dynamicznych włamań w systemach komputerowych**)*

Keywords: Artifical immune system, Network Intrusion detection, Dynamic evolution model
Słowa kluczowe: komputerowe systemy odpornościowe CIS

I. Introduction

Network security techniques [1-6] based on artificial immune system (AIS) have the features of diversity, self-adaptation and robustness, thus, they are considered a very important research direction in network security [1,7]. However, there are two major defects in the present computer immune systems (CIS) : One is that the self set is very large in size. As the cost for mature detector training is exponentially related to the size of self set [8], the efficiency of the traditional CIS models is very low. The other deficiency is that the definitions of self (normal network behaviors) and nonself (abnormal network behaviors) allow little change after they have been defined in many immune-based models or methods for network intrusion detection system(NIDS) [3,7,9], therefore, having few applications in the real network environment.

In this paper, a new immune based model called AIBM for dynamic intrusion detection is proposed. AIBM uses a very small dynamic self set during the self tolerance for immature detectors, resulting in a high efficiency in generating new mature detectors. Meanwhile, the definition of self and nonself in AIBM is dynamic. As time goes on, AIBM can add new self elements into, or eliminate the mutated ones from the self set, resulting in the dynamic evolution of self set, mature and memory detectors, offering more self-adaptation, and having a lower error rate than traditional CIS models.

II. Proposed Theoretical Model

Antigens (Ag, $Ag \subset D$, $D = \{0, 1\}^l$, $l \in N$) in our approach are fixed-length binary strings extracted from the Internet Protocol (IP) packets transferred in the network. Nonself patterns (*Nonself*) represent IP packets from a computer network attack, while self patterns (*Self*) are normal sanctioned network service transactions and non-malicious background clutter, such that $Self \cup Nonself = Ag$ and $Self \cap Nonself = \Phi$. Let B denote the intrusion detector set given by $B = \{< d, age, count >| d \in D \wedge age \in N \wedge count \in N\}$, where d is the antibody (antibody gene), age is the age of antibody d, $count$ (affinity) is the number of antigens matched by antibody d, and N is the set of natural numbers. B contains two subsets: mature detector (T_b) and memory detector (M_b), such that $B = M_b \cup T_b$, $M_b \cap T_b = \Phi$. A mature detector is a detector that is tolerant to self but not activated by antigens. A memory detector evolves from a mature detector matched enough antigens in its lifecycle. Therefore,

$T_b = \{x \mid x \in B \wedge \forall y \in Self \wedge (< x.d, y > \notin Match \wedge x.count < \beta)\}$

, and $M_b = \{x \mid x \in B,\ \forall y \in Self(< x.d, y > \notin Match \wedge x.count \geq \beta)\}$, where β(>0) represents the activation threshold, $Match = \{< x, y >| x, y \in D \wedge f_{r\_con}(x, y) = 1\}$, and $f_{r\_con}(x,y)$ is a r-contiguous bits matching function [3]. Let I_b denote the set of immature detectors given by $I_b = \{< d, age >| d \in D, age \in N\}$, which is used to generating mature detectors.

For the convenience depicting the relationship of two antigen elements in Ag, the relation $Consanguinity$ in Ag(including self & nonself elements) is introduced and given by:

$Consanguin\ ity = \{< x, y >| x, y \in Ag \wedge < x, y > \in Match\}$.

Given $X \subseteq Ag$, $\forall x, y \in X$, $< x, y > \in Consanguinity$, X is called a *consanguinity class* generated by *Consanguinity*. If X is a consanguinity class and there is no relation *Consanguinity* between any element in $Ag - X$ and the element in X, then X is called a *maximal consanguinity class*.

It is obvious that the relation *Consanguinity* is reflexive and symmetrical, but not transitive. Let $< x, y > \in Consanguinity$, i.e., there is a relation *Consanguinity* between two antigen elements x and y, then we know that x and y have similar genes.

Suppose that each element in Ag is a point in a two-dimensional space. For any $x, y \in Ag$, if $< x, y > \in Consanguinity$, then there is an edge between x and y. Thus all elements in Ag can form a graph, which is called *consanguinity graph*. For convenience, directed edges are replaced by undirected ones and the closed curve from a vertex to itself is ignored when drawing the *consanguinity graph*.

According to the definition of *maximal consanguinity class*, we have: an isolated point in the consanguinity graph is a maximal consanguinity class; all points in a maximal complete sub-graph form a maximal consanguinity class; the two points of an edge, which is not in a maximal complete sub-graph, also form a *maximal consanguinity class*.

As Fig. 1 shows, the *maximal consanguinity classes* are: $\{b\}$, $\{a, c\}$, $\{a, h\}$, $\{c, e, f\}$, $\{c, d, f, g\}$.

Given $\pi = \{A_1, A_2, ..., A_n\}$, $Ag_1 = Ag$, $Ag_i = Ag - \bigcup_{1 \leq j < i \leq n} A_j$, and $\pi^i = \{X^i_1, X^i_2, ..., X^i_k\}$, where π^i is the set of *maximal consanguinity classes* in Ag_i, $A_i \in \{x \mid x \in \pi^i, |x| = \max_{1 \leq t \leq k}(|X^i_t|\}$, i.e., A_i is a *maximal consanguinity class* in Ag_i , which has maximum elements.

$Ag = \bigcup_{1 \le i \le n} A_i$, we call π the *maximal consanguinity genealogy*. According to the definition of π , we have $A_i \cap A_j = \Phi$, where $1 \le j < i \le n$. The above description proves that the *maximal consanguinity genealogy* in *Ag* is a partition of *Ag*.

Since all the elements in a *maximal consanguinity class* have similar genes, therefore, we can use one element to represent the whole set. Let $A_i^{gene} = d(d \in A_i)$ denote the gene of A_i. Given $\pi^{gene} = \{A_1^{gene}, A_2^{gene}, ..., A_n^{gene}\}$, we call π^{gene} as the *gene sequence* of the *maximal consanguinity genealogy* in *Ag*. It is obvious that A_i^{gene} and π^{gene} can be regarded as the basic characteristics of A_i and *Ag*, respectively.

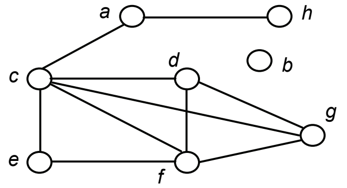

Fig. 1. Consanguinity graph.

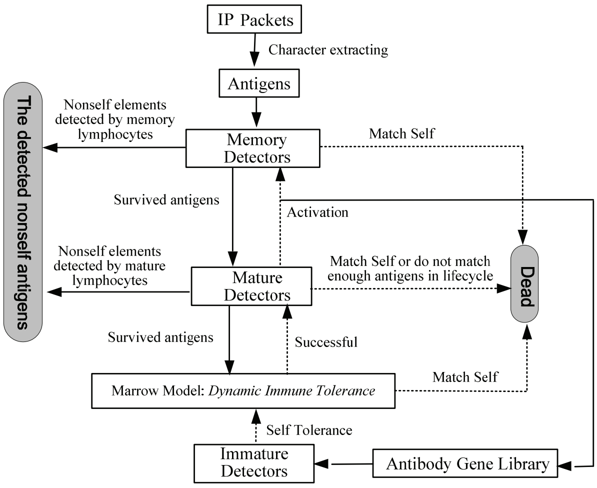
IP Packets
Character extracting
Antigens
The detected nonself antigens
Nonself elements detected by memory lymphocytes
Memory Detectors
Match Self
Survived antigens
Activation
Nonself elements detected by mature lymphocytes
Mature Detectors
Match Self or do not match enough antigens in lifecycle
Dead
Survived antigens
Successful
Match Self
Marrow Model: Dynamic Immune Tolerance
Self Tolerance
Immature Detectors
Antibody Gene Library

Fig. 2. The Framework of AIBM

Fig. 2 illustrates the framework of *AIBM*, where the new immature detectors have to experience a self tolerance period: the detectors will be eliminated if it matches any self antigen (negative selection). The immature detectors that survived in self tolerance period will evolve into mature ones, there the mature detectors have a fixed lifecycle: the detectors will be eliminated if they do not match enough antigens or match self elements in their lifecycle; they will be activated if they get enough antigens. However, the activated detectors will be eliminated if they do not receive costimulation, i.e., false positive error, there the detected antigens are self elements. Meanwhile, the detectors will evolve into memory ones with the help of costimulation, there the detected antigens are sure nonself elements. The memory detectors have an infinite lifecycle, and will be activated as soon as they match an antigen. However, the memory detectors will be eliminated if they match self element, i.e., the false positive error happens.

In the process of intrusion detection, the antibody genes of the activated mature detectors, i.e., new intrusions are found, will be saved into *Antibody Gene Library* and used as dominant inheritability genes, so it is possible that better genes of a new antibody may be generated through some evolutionary strategies (e.g., gene edit, genetic operator).

AIBM serves to classify an input set (*Ag*) into self (Ag_{Self}) and nonself ($Ag_{Nonself}$) by *B* within δ steps. In each step, a fixed amount of antigens are selected from *Ag* to form *sAg* for detection. The antigens classified into Ag_{Self} are used as self elements for the self tolerance of immature detectors. Since the Ag_{Self} is dynamic, so does the self tolerance, thus, two detectors may exist in the system simultaneously: one is tolerant of a certain self element, but the other is not. This problem is solved by co-stimulation.

A. Self Evolution

(1) $$Self(t) = \begin{cases} \{x_1, x_2, ..., x_n\} & t = 0 \\ Self(t-1) & t \bmod \delta \ne 0 \\ Self(t-1) \cup Self_{new}(t) - & \\ Self_{variation}(t) - Self_{dead}(t) & t > 0 \wedge t \bmod \delta = 0 \end{cases}$$

(2) $$Self_{new}(t) = \pi^{gene}(Ag_{self}(t-1))$$

(3) $$\begin{aligned} Self_{variation}(t) = \{x \mid x \in Self(t-1), \\ \exists y \in B(t-1)\ (f_{check}(y,x) = 2 \wedge f_{costimulation}(x) = 0)\} \end{aligned}$$

(4) $$Self_{dead}(t) = \begin{cases} \phi, & |Self(t-1)| + |Self_{new}(t)| \\ & -|Self_{variation}(t)| \le L \\ \{\text{the } |Self_{new}(t)| - & \\ |Self_{variation}(t)| \text{ earliest} & \textit{otherwise} \\ \text{elements in } Self(t-1)\}, & \end{cases}$$

Equation (1) simulates the dynamic evolution of self set, where $x_i \in D(i \ge 1, i \in N)$ is the initial self element defined by the security administrator, $Self_{variation}$ is the nonself set (mutated self elements) newly defined by the security administrator at time *t*. $Self_{new}$ is the new self set collected from the network in last period, where $\pi^{gene}(Ag_{self}(t-1))$ is the *gene sequence* of the *maximal consanguinity genealogy* in *Agself* (*t*-1) (the detected self elements) at time *t*-1. Instead of the whole $Ag_{self,}$ only its gene sequence, which is typical characters of A_{gself}, is collected. As the size of gene sequence of Ag_{self} is much smaller than that of Ag_{self} itself, thus, the size of *Self* will not be increased rapidly.

δ (>0) is called the self evolution period. The self evolution period δ means that the self set evolves periodically (δ): the *Self* keeps stable in the period of δ , but will be changed after the end of evolution period δ , where the elements in $Self_{variation}$ will be eliminated, and the new self elements ($Self_{new}$) collected from the network in last period and survived during the δ steps detection, will be supplemented. However, the earliest self elements ($Self_{dead}$) will be eliminated when the size of self set exceeds the threshold *L*, thus, the size of self set will not be increased unlimitedly, therefore, the self tolerance for immature detectors can be carried out in a high efficient way (see sec.*D*. Immune Tolerance , for detail).

$f_{check}(y,x)(y \in B, x \in Ag)$ is used to classify antigens as either self or nonself: if *x* is matched and does not belong to *Self*(*t*-1), then *x* is sure a nonself antigen, and 1 is returned; if *x* is matched and belongs to *Self*(*t*-1), then *x* may be a nonself antigen (needs co-stimulation), and 2 is returned; however, if *x* is not matched, then *x* is sure a self antigen, and 0 is returned. $f_{costimulation}(x)(x \in Ag)$ simulates the co-stimulation in a biological immune system. If *x* is a self antigen, then $f_{costimulation}(x)$=1, else $f_{costimulation}(x)$=0, which is usually a response from network-security administrator.

B. Antibody Gene Library

$$(5)\quad Agd(t)=\begin{cases}\{d_1,d_2,\ldots,d_k\} & t=0\\ Agd(t-1)\cup Agd_{new}(t)-Agd_{dead}(t) & t\geq 1\end{cases}$$

$$(6)\quad Agd_{new}(t)=\bigcup_{x\in T_{clone}(t)}\{x.d\}$$

$$(7)\quad Agd_{dead}(t)=\bigcup_{x\in M_{dead}(t)}\{x.d\}$$

where $d_i \in D(i=1,\ldots,k)$ is the initial antibody gene, Agd_{new} is the set of antibody genes of mature detectors which are activated by antigens at time t (i.e., some new intrusions have been detected, save the corresponding antibody genes, see section *E*. mature-detector evolution, for details), and Agd_{dead} is the set of antibody genes of memory detectors which match self antigens at time t (i.e., a false-positive error happens, see section *F*. memory-detector evolution , for details). Agd_{new} is used as dominant inheritability genes, so it is possible that better genes of a new antibody may be generated through some evolutionary strategies (e.g., gene edit, genetic operator). However, the antibody genes, which are confirmed as wrong genes and do not satisfy the requirement of the current network any more, need to be deleted from the antibody-gene library. The antibody-gene library Agd is used to generate antibodies of immature detectors in a marrow model (see section *D*. immune tolerance, for details).

C. Antigen Evolution

$$(8)\quad Ag(t)=\begin{cases}Self(0) & t=0\\ Ag(t-1)-Ag_{nonself}(t) & t>0, t \bmod \delta \neq 0\\ Ag_{new}(t) & t>0, t \bmod \delta = 0\end{cases}$$

$$(9)\quad \begin{aligned}Ag_{nonself}(t)=\{x \mid x\in sAg(t-1),\\ \exists y\in B(t-1)((f_{check}(y,x)=2\wedge\\ f_{costimulation}(x)=0)\vee f_{check}(y,x)=1)\}\end{aligned}$$

$$(10)\quad Ag_{self}(t)=\begin{cases}Ag(t) & t=0, t \bmod \delta \neq 0\\ Ag(t-1) & t>0, t \bmod \delta = 0\end{cases}$$

Where $sAg(t)\subset Ag(t)$, $|sAg(t)|=\eta * |Ag(t)|$, $t\geq 0$, sAg is selected from Ag randomly in the proportion of η (detective coefficient, 0<η≤1). $Ag_{nonself}$ is the set of nonself antigens detected by the detectors at time t, where $Ag(0)$ = $Self(0)$ indicates that at this time the model is try to do the job of self tolerance for the newly generated immature detectors and produce new mature detectors (see Section *D*. “Immune Tolerance”). Note, δ here is also called antigen update period, indicating that Ag is replaced by the new antigen set (Ag_{new}) every δ steps. In each antigen update period, the detected nonself antigens are deleted from Ag, then, the remaining antigens in Ag are classified into self elements (Ag_{Self}).

D. Immune Tolerance (Marrow Model)

$$(11)\quad I_b(t)=\begin{cases}\{x_1,x_2,\ldots,x_\xi\} & t=0\\ I_{tolerance}(t)-I_{maturation}(t)\cup I_{new}(t) & t\geq 1\end{cases}$$

$$(12)\quad \begin{aligned}I_{tolerance}(t)=\{y \mid y.d=x.d, y.age=x.age+1,\\ x\in(I_b(t-1)-\{x \mid x\in I_b(t-1),\\ \exists y\in Self(t-1),(f_{r\_con}(x,y)=1)\})\}\end{aligned}$$

$$(13)\quad I_{maturatiion}(t)=\{x \mid x\in I_{tolerance}(t), x.age>\alpha\}$$

$$(14)\quad I_{new}(t)=\{y_1,y_2,\ldots,y_\xi\}$$

Equation (11) simulates the lymphocyte growth in the marrow, where x_i=<d,0> $(d\in D, 1\leq i\leq \xi)$ is the initial immature detector generated randomly. $I_{tolerance}$ is the set of survived immature detectors in I_b(t-1) after one step of tolerance process. Immature detectors need undergo α (≥1, tolerance period) steps of tolerance processes and then evolve into mature ones. $I_{maturation}$ is the set of immature detectors which have undergone α steps of tolerance processes and evolved into mature ones at time t. I_{new} is the set of new immature detectors generated at time t.

The generation of I_{new} is based on Agd (shown in section *B*. antibody gene library), where the key step is to generate the antibodies of immature detectors. The newly generated antibodies of immature detectors are usually composed of two parts: some antibodies are generated randomly, the others are derived from Agd, where the deriving methods include gene edit, genetic algorithm, etc.

Our proposed marrow model is based on the dynamic self tolerance, where the system can add new self elements into and remove mutated ones from $Self(t)$ at any time. The model has a good adaptive ability where the process of self tolerance is dynamic. The newly added self antigens will make the model generate new mature detectors which tolerate those new self elements. However, the mature detectors, which generated before these new self elements added into $Self(t)$, may not tolerant of these new self elements. Thus, two different detectors may exist: one is tolerant of a certain antigen, but the other is not. Competition between these detectors is arbitrated by the external system (co-stimulation, please refer to $f_{costimulation}$.

In a traditional CIS model, the procedure of training immature detectors is to do a matching calculation for each detector with all the self elements. As it is proved that the cost for mature detector training is exponentially related to the size of self set [8], thus, the efficiency of generating mature detectors is very low. However, different from the traditional CIS models, the self set is replaced by the gene sequences of self, as the size of gene sequence of self set is much smaller than that of self set itself, so the training cost of AIBM for generating mature detectors is much lower that trainditional CIS models.

E. Mature-Detector Evolution

$$(15)\quad T_b(t)=\begin{cases}\phi & t=0\\ T_b^{'}(t)\cup T_{new}(t)-T_{memory}(t)-T_{dead}(t) & t\geq 1\end{cases}$$

$$(16)\quad T_b^{'}(t)=T_b^{''}(t)-P(t)\cup T_{clone}(t)$$

$$(17)\quad \begin{aligned}T_b^{''}(t)=\{y \mid y.d=x.d, y.age=x.age+1,\\ y.count=x.count, x\in T_b(t-1)\}\end{aligned}$$

$$(18)\quad \begin{aligned}P(t)=\{x \mid x\in T_b^{''}(t),\exists y\in sAg(t-1),\\ (f_{check}(x,y)=1\vee(f_{check}(x,y)=2\wedge f_{costimulation}(y)=0)\}\end{aligned}$$

Equation (15) depicts the lifecycle of the mature detectors. All mature detectors have a fixed lifecycle (λ). If a mature detector matches enough antigens (≥β) in its lifecycle, it will evolve to a memory one (T_{memory}). However, the detectors will be killed and replaced by newly generated mature detectors (T_{new}) if they do not match enough antigens in their lifecycle. T_{dead} is the set of detectors that have not match enough antigens (≤β) in lifecycle λ or classified self antigens as nonself (i.e., false-positive error) at time t. $T_b^{'}$ simulates that the mature detectors undergo one step of evolution. $T_b^{''}$ indicates that the mature detectors are getting older. P depicts the set of mature detectors whose antibodies match nonself antigens.

$T_{clone}(t)=\{y \mid y.d=x.d, y.age=x.age, y.count=x.count+1,$ $x \in P(t)\}$, which depicts the clone process of mature detectors, which is simplified by just adding matching count by 1. In the mature-detector lifecycle, the inefficient detectors on classifying antigens are killed through the process of clone selection. However, the efficient detectors on classifying antigens will evolve to memory ones. Therefore, similar antigens representing abnormal network behaviors can be detected quickly when they intrude the system again.

F. Memory-Detector Evolution

$$M_b(t)=\begin{cases}\phi & t=0\\ M_b(t-1)-M_{dead}(t)\cup T_{memory}(t) & t\geq 1\end{cases}\qquad(19)$$

$$M_{dead}(t)=\{x \mid x\in M_b(t-1),\ \exists y\in Ag(t-1)(f_{check}(x,y,t)=2\wedge f_{costimulation}(y)=1)\}\qquad(20)$$

Equation (19) depicts the dynamic evolution of M_b, where T_{memory} is the set of newly generated memory detectors. A memory detector will be deleted if it matches a known self antigen (M_{dead}, i.e., false-positive error).

III. Simulations and experimental Results

The experiment was carried out in the Laboratory of Computer Network and Information Security at Sichuan University. A total of 40 computers in a network were under surveillanced. An antigen was defined as a fixed length binary string (l=256) composed of the source/destination IP address, port number, protocol type, IP flags, IP overall packet length, TCP/UDP/ICMP fields, and etc. The task aimed to detect network attacks. The size of self set is set to 75 (L=75). The r-contiguous bits matching rule was used to compute the affinity between antigens and antibodies (r=8) [3,9]. The size of initial self set n is randomly set to 40, and the number of newly generated immature detectors ξ=10. 100 IP packets were captured from network each time, and, they were transformed into antigen format to be processed by the detection system. The detective coefficient η was randomly set at 0.8. The network was attacked by 20 kinds of attacks, such as Syn Flood, Land, Smurf, Teardrop, …, etc, the proportion between self and nonself packets was 9:1, i.e., there was one nonself packet among 10 packets. The experimental results were evaluated by TP rate (the true positive rate, the nonself detection probability) and FP rate (the false positive rate, the probability of the self antigens being detected by mistake).

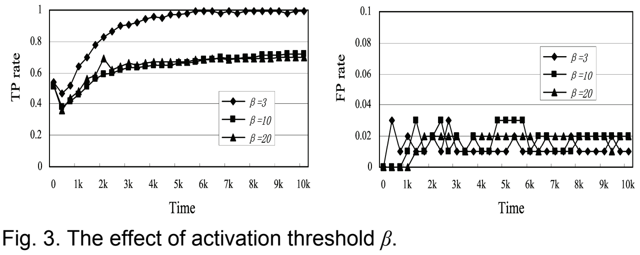

Fig. 3. The effect of activation threshold β.

Fig. 3~6 illustrate the effect of the parameters of α, δ, λ and β, and, Fig. 7 shows a satisfied result obtained in the experiments.

In contrast to the previous works on immune-based models or methods for NIDS [3,6-9], whose definitions of self and nonself allow little change after they have been defined. However, our proposed method has a dynamic evolution model for the definitions of self and nonself, and thus, offers more self-adaptability. To test the effectiveness of our proposed model, the corresponding comparison experiments were performed, with Exhaustive algorithm [9,10], proposed by Forrest et al, selected as the opponent. The Exhaustive algorithm is a typical one among the algorithms used in the traditional CIS, which has a strong impact on the design of CIS.

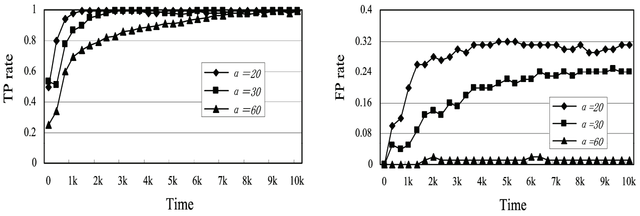

Fig. 4. The effect of tolerance period α.

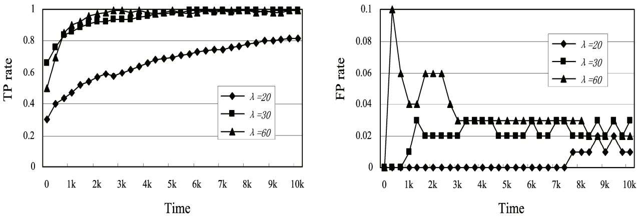

Fig. 5. The effect of lifecycle λ.

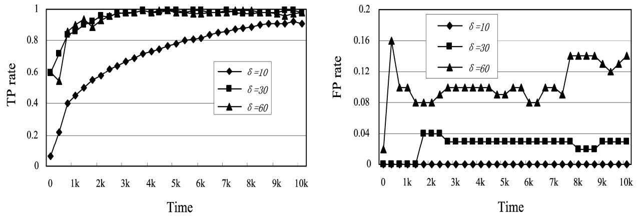

Fig. 6. The effect of updating cycle δ.

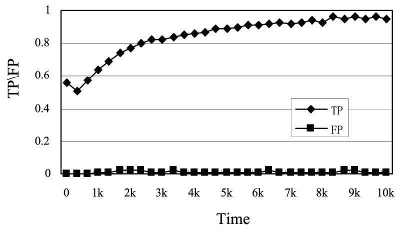

Fig. 7. The satisfied TP and FP, where $\alpha=\delta$=50, β=5, λ=40.

Fig. 8 illustrates the FP rates for both the Exhaustive Algorithm and AIBM, where 40 of each 100 packets are self antigens, and half of them are newly defined (e.g., another 20 ports are now opened to provide more services). Since the Exhaustive Algorithm cannot alter the self elements when the training phase has been completed, the detectors generated by the algorithm will not be tolerant of the newly defined self antigens, which is the reason why it has a higher FP rate than AIBM.

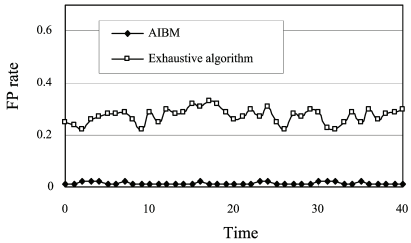

Fig. 8. FP rates for both the Exhaustive algorithm and AIBM, where 40 of each 100 packets are self antigens, and half of them are newly defined.

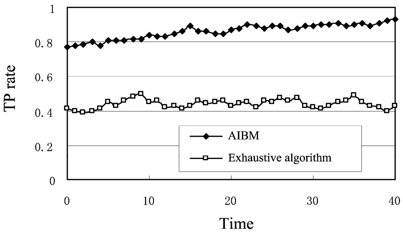

Fig. 9. TP rates for both the Exhaustive algorithm and AIBM, where 80 of each 100 packets are nonself antigens, and half of them are new defined.

Fig. 9 illustrates the TP rates for both the Exhaustive Algorithm and AIBM, where 80 of each 100 packets are nonself antigens, and half of them were self antigens before, but are nonself behaviors now. That is, 40 ports in the system are now closed and do not provide any more services. Since the Exhaustive Algorithm cannot deal with mutated self antigens, it has a lower TP rate than AIBM.

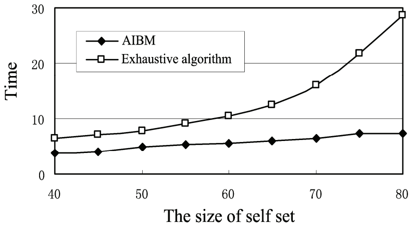

Fig. 10. The mature detector generating efficiency for both the Exhaustive algorithm and AIBM, where the number of generated mature detectors is fixed, e.g., 20.

Fig. 10 illustrates the mature detector generating efficiency with different size of self set for both the Exhaustive Algorithm and AIBM. In AIBM, the training set for mature detectors is just only formed by the gene sequences of self, which is much smaller than the orignal self set itself. Thus, the number of self elements used in the immune tolerance process is very small and will not exceed a threshold L (see equation (1) and (4) for details). Thus the time used to generate a fixed number of mature detectors is stable. However, the Exhaustive Algorithm uses all self elements to train the immature detectors. Therefore, it takes much more time than AIBM to generate mature detectors when the size of self set increases.

IV. Conclusion

In a real-network environment, the roles of self and nonself may exchange at times, e.g., the network administrator may open or close some ports to provide or forbid some specified services (e.g., port 80 is opened or closed to provide or forbid the WWW service), as a result, some network activities, which were forbidden before, are permitted now, and *vice versa*. Thus, a dynamic model for the normal and malicious network activities is needed to depict the evolution of self and nonself.

However, almost all the reported immune based intrusion detection systems use a static way for self description, which lack self-adaptation and cannot fit to the complicated networks, causing a higher false-negative and false-positive rates. Moreover, the static self definition often encounters an inevitable problem in the real-network environment: the training cost of mature detectors is exponential in the size of self set.

In this paper, a new immune-based model, which is called AIBM, for dynamic intrusion detection is proposed. In AIBM, a new method for naturally and dynamically defining self is proposed. The self elements are collected naturally from the network, meanwhile, the mutated self elements will be eliminated. Therefore, the self set can synchronize their variations with the real-network environment as time goes on. And the concept of dynamic tolerance for immature detectors is thus advanced.

As a result, AIBM can use a very small dynamic self set during the self tolerance for immature detectors, resulting in a high efficiency in generating new mature detectors and having a lower error rate than traditional CIS models, thus, the problem that the time cost for training mature detector is exponential in the size of self set is avoided.

This work was supported by the National Natural Science Foundation of China under Grant No. 60873246, 61173159, the Cultivation Fund of the Key Scientific and Technical Innovation Project of Ministry of Education of China under Grant No. 708075.

REFERENCES

[1] Klarreich E. Inspired by Immunity. Nature, vol. (415) (2002) 468-470

[2] B. Hanson. Sensing Worms, Science, 330(6011): 1589, 2010

[3] Li, T.: Computer Immunology. Publishing House of Electronics Industry Beijing (2004).

[4] Albert R, Jeong H, Barabasi A L. Attack and error tolerance of complex networks. Nature, vol. (406) (2002) 378-382.

[5] F. R. Chang. Is Your Computer Secure? Science, 325(5940): 550-551, 2009

[6] Perelson A S, Weisbuch G. Immunology for physicists. Rev Mod Phys, 1997, 69(4):1219-1263

[7] L. N. De Castro and J. I. Timmis, Artificial Immune Systems as a Novel Soft Computing Paradigm, Soft Computing Journal, 2003, 7(8): 526-544.

[8] D'haeseleer, P., Forrest, S., An immunological approach to change detection: algorithm, analysis and implication, In: IEEE Symposium on Research in Security and Privacy, Oakland, CA, IEEE Computer Society Press, 1996, 110-119.

[9] Forrest S, Perelson A S. Self-nonself discrimination in a computer. In Proc. IEEE Symposium on Security and Privacy, Oakland (1994) 202-213.

[10] Timmis J, Bentley P J. Negative selection: how to generate detectors. In Proc. of the 1st International Conf. on Artificial Immune Systems, University of Kent at Canterbury (2002) 89-98

***Authors**: Associate prof. Xiaojie Liu, School of Computer, Sichuan University, P.R.China, No.24, South Section 1, Yi Huan Road, 610065, E-mail:* liuxiaojie@scu.edu.cn.